IDtrust 2008
7th Symposium on Identity and Trust on the Internet
Program and Proceedings

Notes

Transportation

There will be a shuttle leaving the Gaithersburg Holiday Inn at 8:00 a.m. Tuesday morning to travel to NIST. The shuttle will leave at 8:15 a.m. Wednesday and Thursday. The shuttle will return to the hotel at the end of the sessions on Tuesday and Wednesday. There will not be shuttle service the afternoon of Thursday.

Wireless

802.11b Wireless access points will be available for SSH, IPSEC, HTTP, DNS, FTP, POP, IMAP, and SMTP connectivity. Only WPA1 access will be provided, and users must sign NIST's Visitor Network Access Agreement with regard to security patches, anti-virus software, etc. NIST's Visitor Network Access Agreements are available in the registration area.

Blogging

Participants and observers are encouraged to use the tag "idtrust2008" when blogging about the symposium.

Tuesday, March 4, 2008 - Full Day

8:00 Bus Departs from Gaithersburg Holiday Inn for NIST
8:30 - 9:00

Registration and Continental Breakfast

9:00 - 9:10 Welcome and Opening Remarks

Program Chair: Kent Seamons, *Brigham Young University* (Slides: ppt)

9:10 - 10:00 Keynote Talk I

Identity Interoperability, Standards, and the State of Adoption

(Presentation slides: ppt)

Dan Blum, *Sr. VP and Principal Analyst, Burton Group*

10:00 - 10:30 Break
10:30 - 12:00 Session 1 - Technical papers - Identity Management
Session Chair: Carl Ellison, *Microsoft*

A Client-Side CardSpace-Liberty Integration Architecture
(Presentation slides: ppt)

Waleed Alrodhan, *University of London*
Chris Mitchell, *University of London*

Identity Protection Factor (IPF)
(Presentation slides: pdf odp)

Arshad Noor, *StrongAuth*

OpenID Identity Discovery with XRI and XRDS
(Presentation slides: ppt)

Drummond Reed, *Cordance*
Les Chasen, *NeuStar*
William Tan, *Neustar*

12:00 - 12:15 Break
12:15 - 1:00 Keynote Talk II
Identity and Policy for Security, Trust and Privacy
(Presentation slides: pdf)
Rakesh Radhakrishnan, *Chief Identity Integration Architect, Sun Microsystems, Inc.*

1:00 - 2:00 Lunch
2:00 - 3:30 Session 2 - Panel: Open Reputation Management Systems
Panel Moderator: Abbie Babir, *Nortel* (Slides: pdf ppt)

Drummond Reed, *Cordance Corporation* (Slides: ppt)
Tony Nadalin, *IBM*
Chris Hagenbuch, *SafeTSpace* (Slides: ppt)
Rakesh Radhakrishnan, *Sun Microsystems* (Slides: pdf)

3:30 - 4:00 Break
4:00 - 5:30 Session 3 - Technical papers - Access Control in Open Systems
Session Chair: Carl Ellison, *Microsoft*

A Content-Driven Access Control System
(Presentation slides: pdf ppt)

Jessica Staddon, *PARC*

Philippe Golle, *PARC*
Paul Rasmussen, *PARC*
Martin Gagne, *U.C. Davis*

Secure Roaming with Identity Metasystems

(Presentation slides: ppt pdf)

Long Nguyen Hoang, *Helsinki University of Technology*
Pekka Laitinen, *Nokia Research Center*
N. Asokan, *Nokia Research Center*

Secure Communication for Ad-Hoc, Federated Groups

(Presentation slides: pdf ppt)

Ludwig Seitz, *Swedish Institute of Computer Science*
Andreas Sjöholm, *Axiomatics and Swedish Institute of Computer Science*
Babak Sadighi, *Axiomatics and Swedish Institute of Computer Science*

5:30 Bus Departs for Gaithersburg Holiday Inn
6:00 Social Gathering and Dinner Buffet - Gaithersburg Holiday Inn

---

Wednesday, March 5, 2008 - Full Day

8:15 Bus Departs from Gaithersburg Holiday Inn for NIST
8:30 - 9:00
Registration and Continental Breakfast

9:00 - 9:15 Welcoming Remarks
OASIS and the IDtrust Member Section: John Sabo, *CA, Inc.* (Slides: ppt)

9:15-9:45 Session 4 - Technical papers - Public Key Infrastructure I
Session Chair: Bill Burr, *NIST*

User-Centric PKI

(Presentation slides: pdf ppt)

Radia Perlman, *Sun Microsystems*
Charlie Kaufman, *Microsoft*

9:45 - 10:00 Break
10:00 - 11:00 Session 5 - Panel - Federations Today and Tomorrow
Ken Klingenstein, *Internet2* (Slides: ppt)
Patrick Harding, *Ping Identity* (Slides: pdf)

11:00 - 11:30 Break

11:30 - 1:00 Session 6: Panel: Liberty Alliance Identity Assurance Framework: Advancing Common Levels of Trust, Certification, Accreditation and Business Rules

Panel Moderator: Peter Alterman, *National Institutes of Health* (Slides: ppt)

Douglas Pelton, *Wells Fargo* (Slides: ppt)

Lena Kannappan, *FuGen Solutions, Inc.* (Slides: pdf)

Jan Riis, *Lakeside A/S* (Slides: ppt)

1:00 - 2:00 Lunch

2:00 - 3:30 Session 7: Technical Papers - Public Key Infrastructure II

Session Chair: Andrew Regenscheid, *NIST*

Public Key Superstructure "It's PKI Jim, But Not As We Know It!"

(Presentation slides: ppt)

Stephen Wilson, *Lockstep Consulting*

Audit and Backup Procedures for Hardware Security Modules

(Presentation slides: odp)

Túlio Cícero Salvaro de Souza, *UFSC*

Jean Everson Martina, *University of Cambridge*

Ricardo Felipe Custódio, *UFSC*

Securing the core with an Enterprise Key Management Infrastructure (EKMI)

(Presentation slides: pdf odp)

Arshad Noor, *StrongAuth*

3:30 - 4:00 Break

4:00 - 5:00 Session 8: Technical Papers - Practice & Experience: Health Care

Session Chair: Scott Rea, *Dartmouth College*

A Federation of Web Services for Danish Health Care

(Presentation slides: ppt)

Esben Dalsgaard, *Digital Health Denmark (SDSD)*

Kåre Kjelstrøm, *Silverbullet A/S*

Jan Riis, *Lakeside A/S*

Security and Privacy System Architecture for an e-Hospital Environment

(Presentation slides: pdf ppt)

Kathryn Garson, *University of Ottawa*

Carlisle Adams, *University of Ottawa*

5:00 - 5:30 Session 9: RUMP Session

Session Chair: Neal McBurnett, *Internet2*

Impromptu Rump Session. Sign-ups will be taken prior to the session by Neal McBurnett.

Privacy View of Systems Engineering

(Presentation slides: ppt)

David Weitzel, *Mitre*

Safeguarding Digital Identity

(Presentation slides: ppt)

Bruce Bakis, *Mitre*

Update on XML Signature, XML Security

(Presentation slides: ppt pdf)

Frederick Hirsch, *Nokia*

Wireless Access using an Identity Provider

(Presentation slides: ppt)

Kent Seamons, *Brigham Young University*

Vehicle Infrastructure Integration (VII): Trusting Your Car to Be Anonymous

James L. Fisher, *Noblis*

5:30 Bus Departs for Gaithersburg Holiday Inn
Dinner (on your own)
8:00 Birds-of-a-Feather Sessions

Gaithersburg Holiday Inn, Washingtonian Room

Thursday March 6, 2008 - Half Day

8:15 Bus Departs from Gaithersburg Holiday Inn for NIST
8:30 - 9:00

Registration and Continental Breakfast

9:00-11:00 Session 10 - Identity and Access Control in the Enterprise using OASIS Security Standards

Panel Moderator: Hal Lockhart, *BEA Systems, Chair, Oasis Technical Committees (SAML, XACML)* (Slides: ppt)

Anil Saldhana, *Red Hat, Member, Oasis Technical Committees (SAML, XACML)* (Slides: ppt)
Anthony Nadalin, *IBM, Member, Oasis Technical Committees (SAML, XACML)* (Slides: ppt)
Andreas Sjöholm, *Axiomatics, Oasis Technical Committee (XACML)* (Slides: ppt)
Sunil Madhu, *Securent (Cisco), Oasis Technical Committee (XACML)* (Slides: ppt)

11:00 - 11:30 Break
11:30 - 12:00 Session 11 - Invited Talk

OpenID: Current Status and Challenges

(Presentation slides: ppt pdf)
George Fletcher, *Chief Architect, Identity Services, AOL*

12:00-12:30 Wrap up

See Also

This workshop is part of the IDtrust Symposium Series

- 2010: 9th Symposium on Identity and Trust on the Internet (IDtrust 2010)
- 2009: 8th Symposium on Identity and Trust on the Internet (IDtrust 2009)
- 2008: 7th Symposium on Identity and Trust on the Internet (IDtrust 2008)
- 2007: 6th Annual PKI R&D Workshop
- 2006: 5th Annual PKI R&D Workshop
- 2005: 4th Annual PKI R&D Workshop
- 2004: 3rd Annual PKI R&D Workshop
- 2003: 2nd Annual PKI Research Workshop
- 2002: 1st Annual PKI Research Workshop

IDtrust 2008

March 4-6, 2008
National Institute of Standards and Technology
Gaithersburg, MD

7th Symposium on Identity and Trust on the Internet (IDtrust 2008) Identity and Trust Infrastructures

Kent Seamons
Brigham Young University
Program Chair

IDtrust 2008

March 4-6, 2008
National Institute of Standards and Technology
Gaithersburg, MD

- 7th Symposium on Identity and Trust on the Internet (IDtrust)
  (Previously the PKI R&D Workshop)
- Our new name reflects the expanding scope and interests of this community
- Researchers and practitioners from industry, government, and academia

Sponsors

- National Institute of Standards and Technology (NIST)
- Internet2
- OASIS IDtrust Member Section
- Federal PKI Policy Authority

Program Committee

- Kent Seamons, *Brigham Young University (chair)*
- Peter Alterman, *National Institutes of Health*
- Abbie Barbir, *Nortel*
- Jim Basney, *NCSA*
- David Chadwick, *University of Kent*
- Joe Cohen, *Forum Systems*
- Carl Ellison, *Microsoft*
- Stephen Farrell, *Trinity College Dublin*
- Richard Guida, *Johnson & Johnson*
- Peter Gutmann, *University of Auckland*
- Russ Housley, *Vigil Security, LLC*
- Himanshu Khurana, *NCSA*
- June Leung, *FundSERV*
- Neal McBurnett, *Internet2*
- Bob Morgan, *University of Washington*
- Clifford Neuman, *University of Southern California*
- Arshad Noor, *StrongAuth*
- Eric Norman, *University of Wisconsin*
- Tim Polk, *NIST*
- Scott Rea, *Dartmouth College*
- Andrew Regenscheid, *NIST*
- John Sabo, *Computer Associates*
- Ravi Sandhu, *Univ. of Texas at San Antonio and TriCipher*
- Krishna Sankar, *Cisco Systems*
- Stefan Santesson, *Microsoft*
- Frank Siebenlist, *Argonne National Laboratory*
- Sean Smith, *Dartmouth College*
- Ann Terwilliger, *Visa International*
- Von Welch, *NCSA*
- Stephen Whitlock, *Boeing*
- Michael Wiener, *Cryptographic Clarity*

Thank You!

Special Thanks

- Steering Committee Chair
  Neal McBurnett, Internet2
- Local Arrangements Chair
  Sara Caswell, NIST
- Dee Schur, OASIS
- General Chair
  Ken Klingenstein, Internet2

Technical Program

- Technical Paper sessions (peer reviewed)
  - Identity Management
  - Access Control in Open Systems
  - PKI
  - Practice and Experience: Health Care
- Submissions up 50% over last year
- Each paper received 4 reviews on average
- Some papers received shepherding
  - Thank you authors and PC members
- Published in the ACM Digital Library as part of the ACM International Conference Proceedings Series

Technical Program

- Panel Sessions
  - Reputation Management
  - Federations Today and Tomorrow
  - Liberty Alliance Identity Assurance Framework
  - Identity and Access Control in the Enterprise
- Invited Talk – OpenID Status and Challenges
  - George Fletcher
    Chief Architect, Identity Services, AOL

Keynote Speakers

Dan Blum
Senior VP and Principal Analyst
Burton Group

Rakesh Radhakrishnan
Lead Architect
Sun Microsystems

RUMP Session

- Short Work-In-Progress Talks
  - Wed afternoon
  - Submit an abstract
  - 5 minute presentations (subject to change)
  - Contact: Neal McBurnett
    - neal@bcn.boulder.co.us

Courtesy: http://www.flaminghotideas.co.uk/library_travel.htm

Last Minute Instructions - Speakers

- Speakers please contact your session chairs in advance
  - At the beginning of the break before your session
- An electronic copy of each presentation should be given to Neal for the web site (ppt/odp/pdf)

Social Gathering and Dinner Buffet

- Tuesday, Gaithersburg Holiday Inn, 6 PM

Bird-of-a-Feather

- Propose a topic for informal Birds-of-a-Feather sessions
  - Signup to attend a session that interests you
- Wednesday, 8 PM
  Room available at the Holiday Inn

Looking to the Future

- Please make plans now to submit a technical paper for next year
  - Submission deadline will be in the fall (October)
- Complete a survey at the conclusion of the workshop – your feedback is important to us!

Enjoy the Workshop

- The success of the workshop is in your hands
  - Participate!
  - We welcome *outrageous* comments and questions

Identity Interoperability, Standards, and the State of Adoption

Presented for the ID Trust 2008
March 4, 2008

Dan Blum

Senior VP, Principal Analyst

Security and Risk Management Strategies

Burton Group

dblum@burtongroup.com

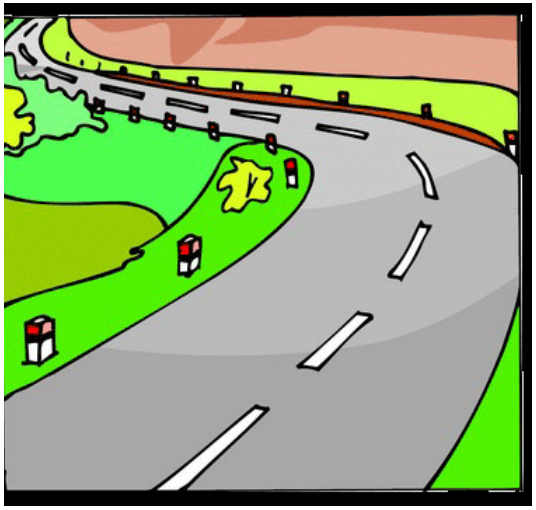
Identity Interoperability

Thesis

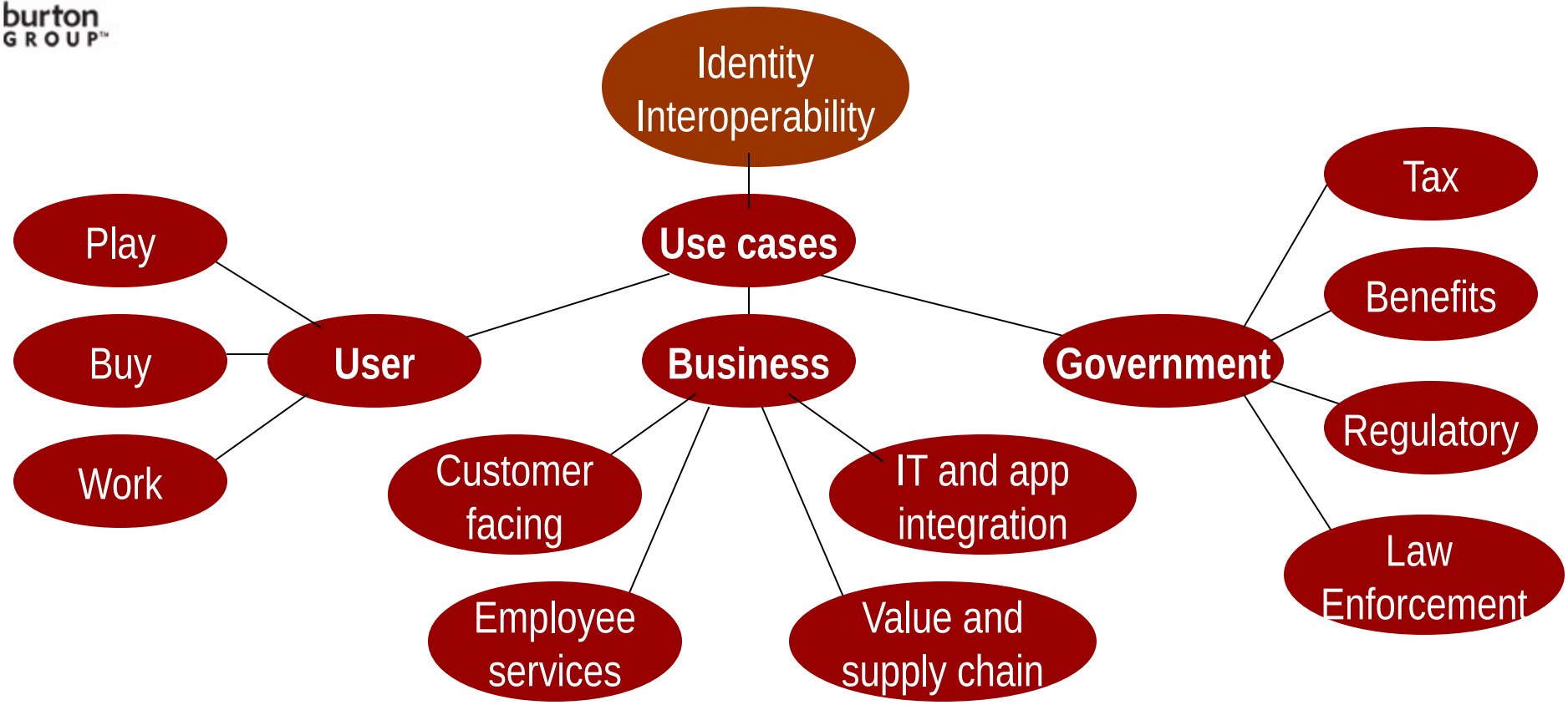
- Identity is key to Internet applications and information security but interoperability remains an issue
- PKI, federated identity and user-centric identity are
  - Three generations of standards
  - All targeting identity interoperability
  - Overlapping and related, but catering to different communities
- We're early on the adoption curve, have unsolved issues
- Standards cannot yet automate business relationship establishment, trust or authorization
- Nonetheless, there are many opportunities for deployment

Roadmap for Presentation

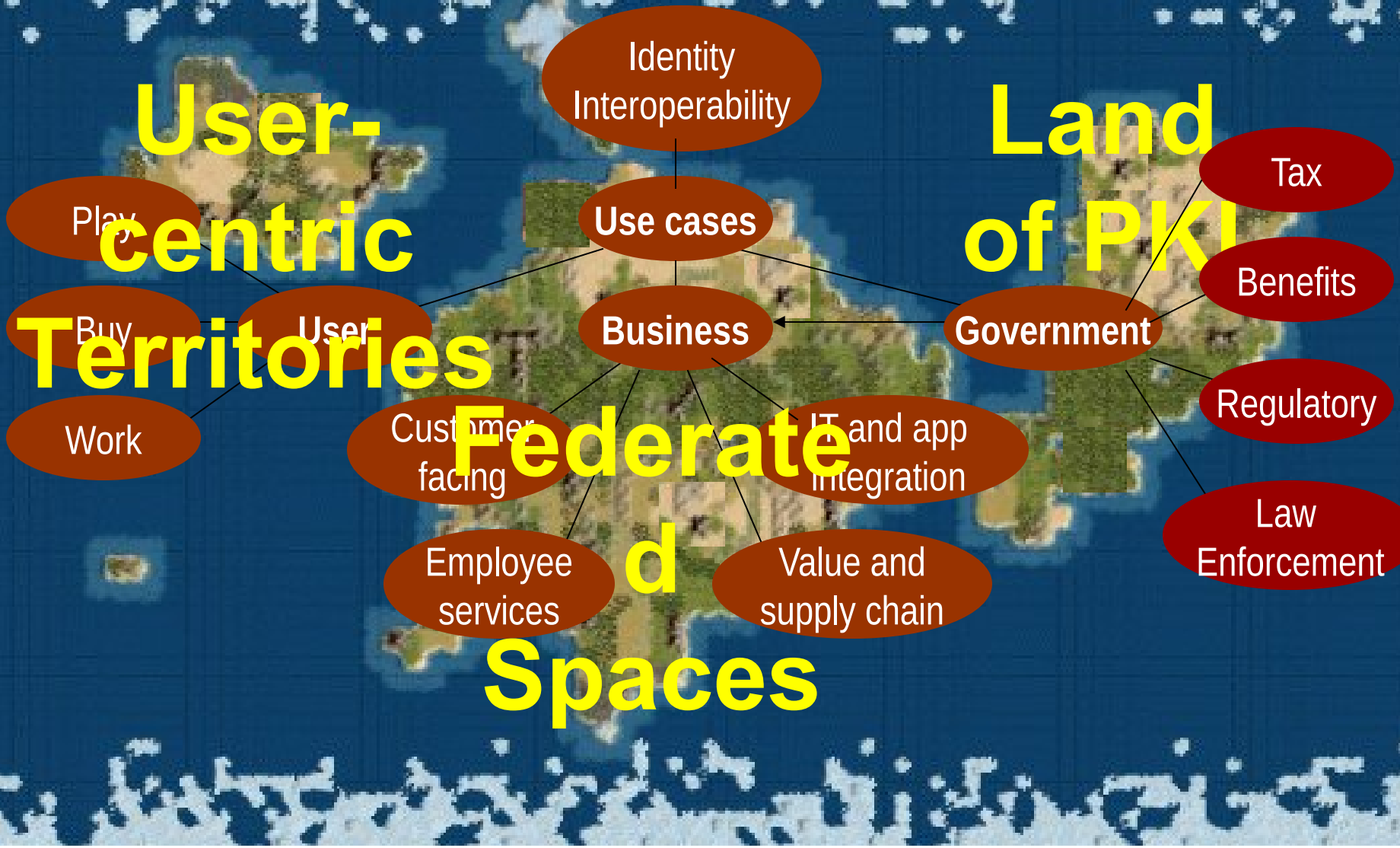
- ***Discuss interoperability use cases, and adoption***
- Evaluate key standards
- Conclusion and recommendations



Mapping the Use Cases



Use Cases – Seem like Islands?

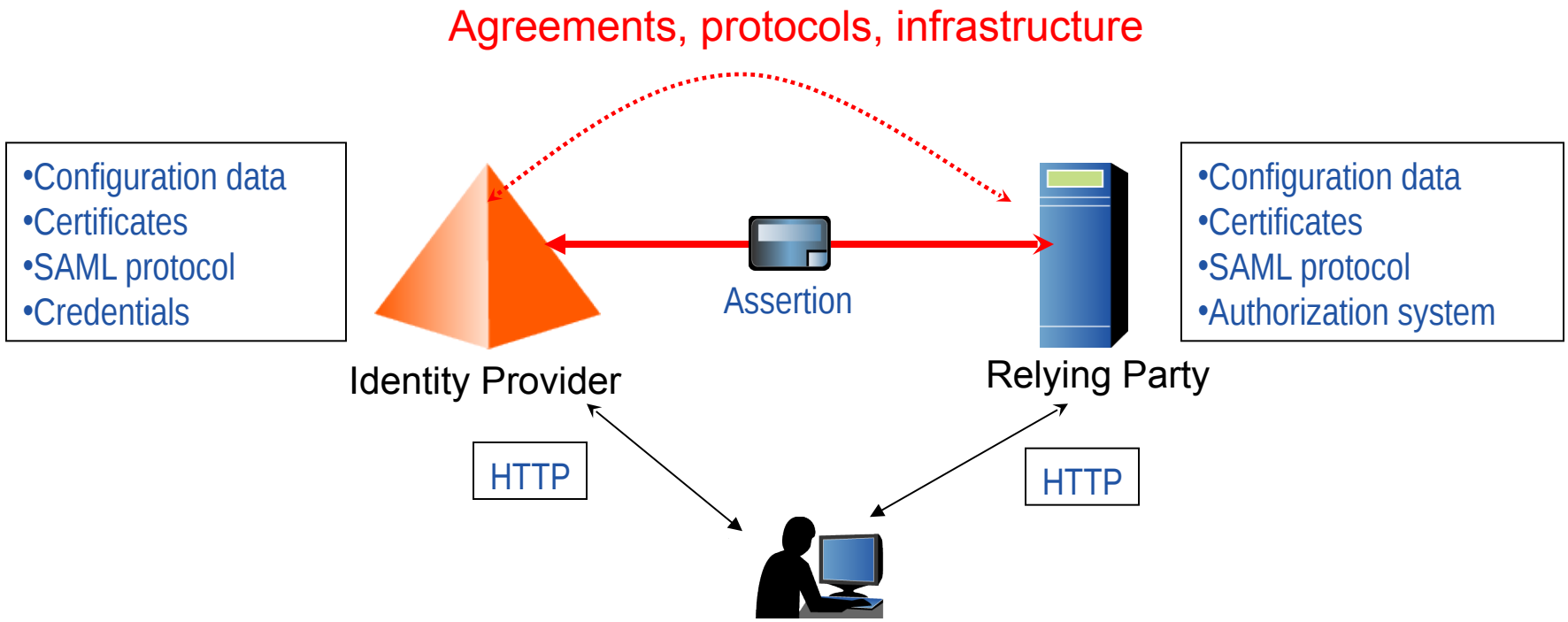


Federated Spaces

What is federated identity?

- *Agreements, standards, and technologies that make identity and entitlements portable across domains*

Basic model



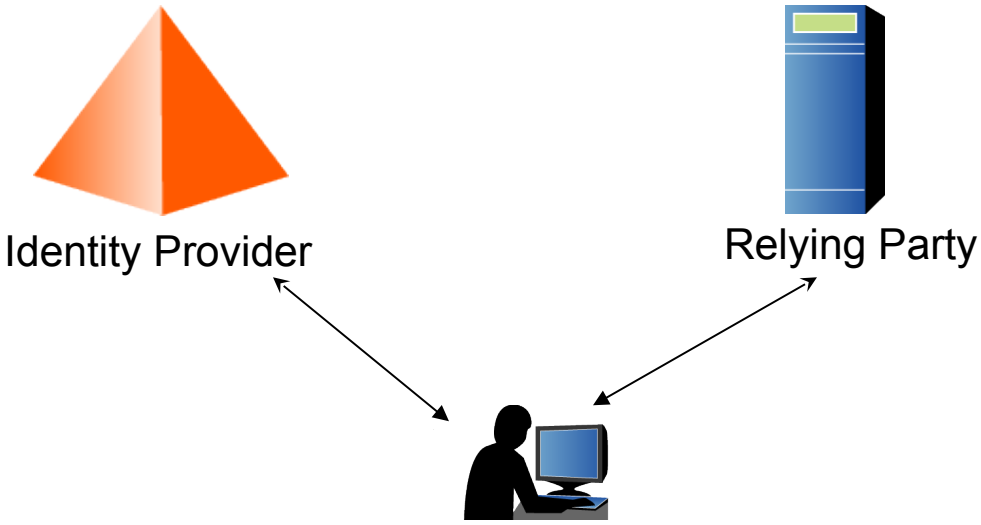
User-Centric Identity

What is user-centric identity?

- *User brokers or controls identification and authentication process to relying parties*

Basic model

A priori agreements may not be necessary



User takes some of the responsibility

- User must be present (or further issues of broker service addressed)
- Client software must be installed
- IDP must implement a token service
- Technical and privacy policies must map from user, IDP, RP

State of the Market

Federated identity adoption is expanding

- Enterprises with many federation partners, trying to scale
- Multi million user deployments underway
- Community federations: higher education, telecommunications, automotive, government, pharmaceutical, financial services, petroleum…

Many commercial products support SAML 2.0

- Vendors have moved quickly to support converged standard; still some growing pains
- Enterprises implementing multiprotocol scenarios

Yet divergence continues - between federated and user-centric spaces, and between some of the vendors

State of the Market

Adoption of federated identity in business growing, but not universal

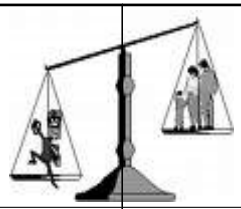
- Deployment barriers
  - Standards and tools that work for 10s or 100s of partners still to hard to scale when deploying 1000s of partners
  - Partners not ready, or unwilling
  - Difficult for small business to deploy
  - Application service providers don't focus on identity
  - Relative high cost for some implementers
  - Concerns about risk, liability, and audit
  - Difficult to build business agreements

Trust is the elephant in the road

Federated Identity Adoption

Incentives must outweigh disincentives

Incentives	Disincentives
• Transactional revenue	• Technical complexity
• New business opportunities	• Deployment cost
• Lower admin costs	• Trust issues



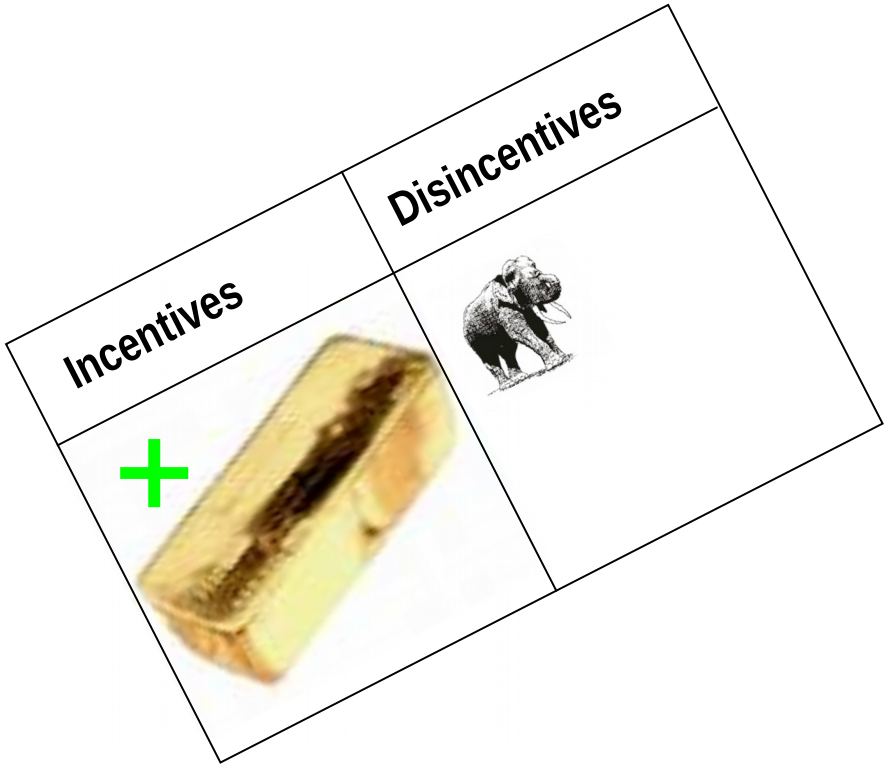
Federated Identity Adoption

Where strong affinities exist federations form

- Pair-wise federations (many)
- Hub and spoke federations (collections of multiple pair-wise)
- Communities of interest (some)

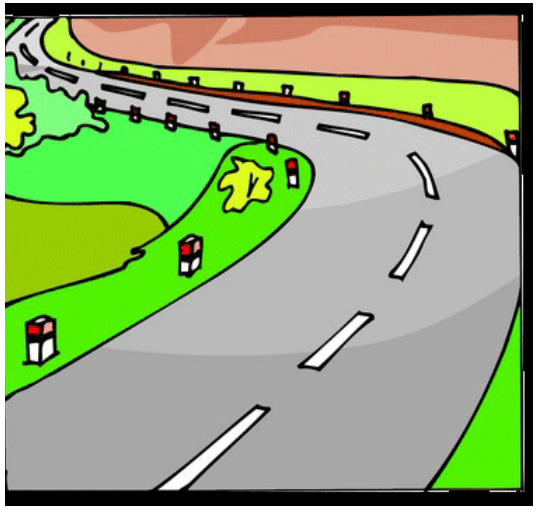
Employee services

Value and supply chain



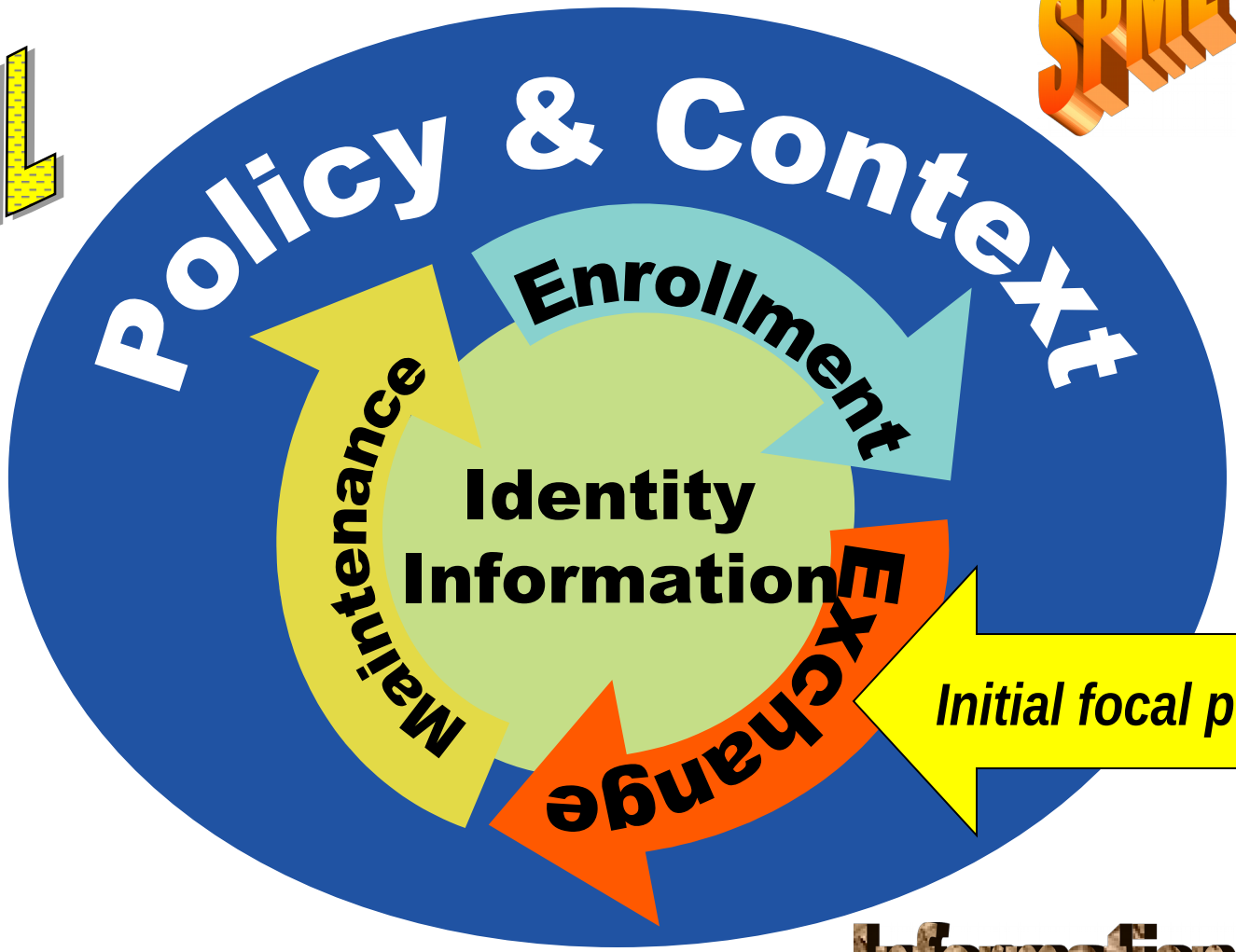
Roadmap for Presentation

- Discuss interoperability use cases, and adoption
- ***Evaluate key standards***
- Conclusion and recommendations



Where do the Standards Fit?

Identity Service



Evaluating the Standards

Exchange alone can be a complicated problem

Standards are complex, with multiple interoperability points

- Protocol
- Schema
- Profiles
- Core features
- Optional features

It's a fairly significant task to test them

- Especially the full functionality for SAML 2.0, Information Cards or PKI

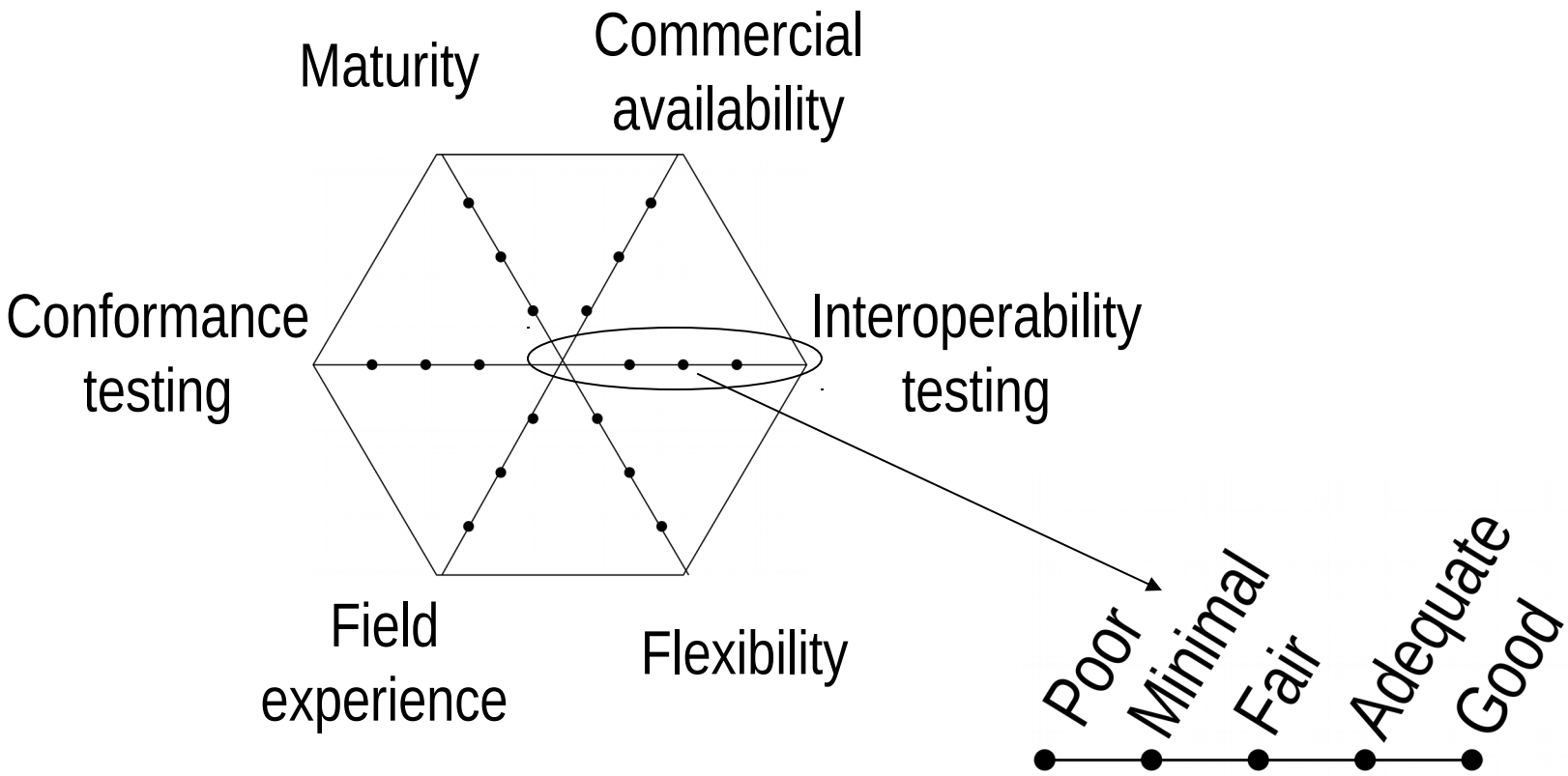
Evaluating the Standards

Ingredients that impact interoperability confidence level

- Maturity of standard
  - How long has it been available?
- Commercial availability
  - Is it available from several vendors?
- Conformance testing
  - Is there a conformance testing program?
- Interoperability testing
  - Is there formal interoperability testing?
- Field experience
  - How many production deployments exist?
- Flexibility
  - How much can you do with it?

Scoring the Standards

Developing a score card

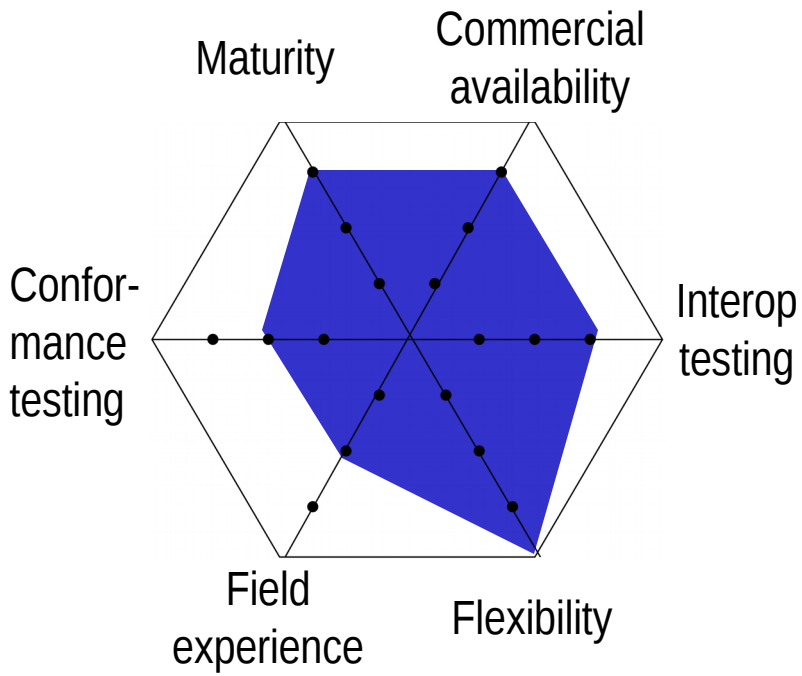


SAML – What its for

- Open standard for browser-based federation
- Integrated in Web services security standards and Information Cards
- SAML 2.0 – powerful, complex and coming - replacing SAML 1.x
- Use SAML as the baseline standard for business federations

How it scores

(Feb 2008)



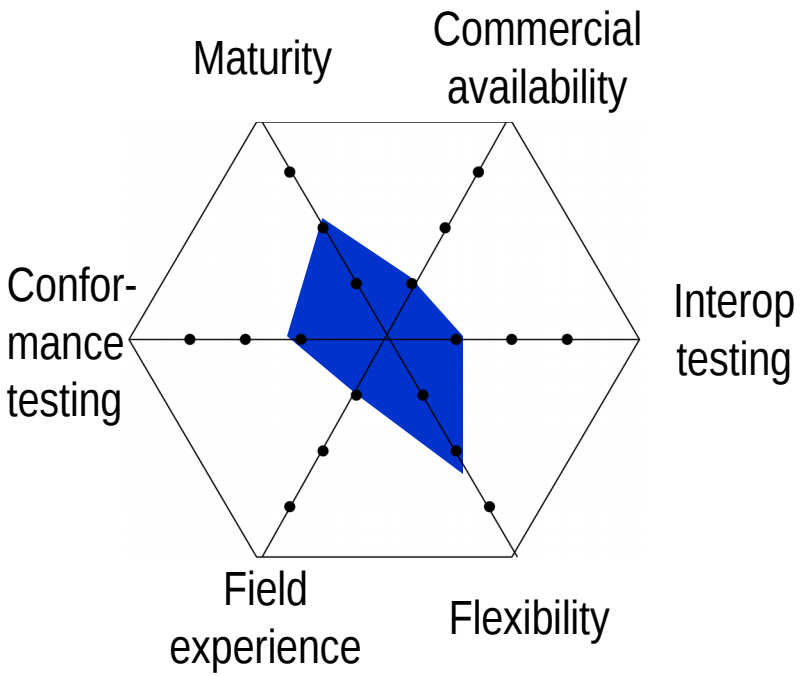
XACML

XACML – What its for

- Express security policies and access rights
- Designed to work with SAML
- Use XACML when building entitlements engines
- Use XACML if context is bounded and you want to express policies in an interoperable, automated manner

How it scores

(Feb 2008)



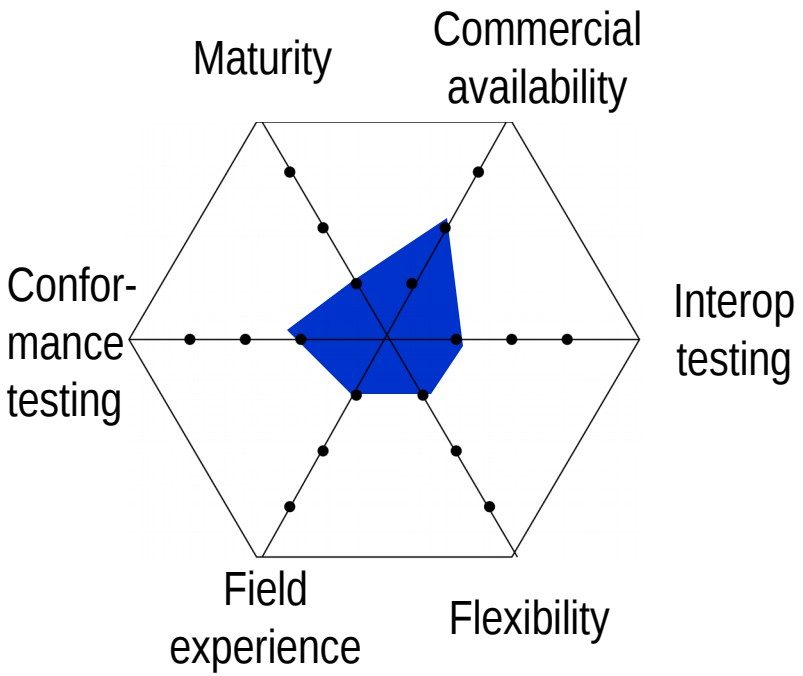
SPML

SPML – What its for

- Provision accounts among applications, organizations in structured, automated way
- Consider SPML when large volumes of accounts must be held at multiple federated identity sites
- And account information is complex and federations will be stable over time

How it scores

(Feb 2008)



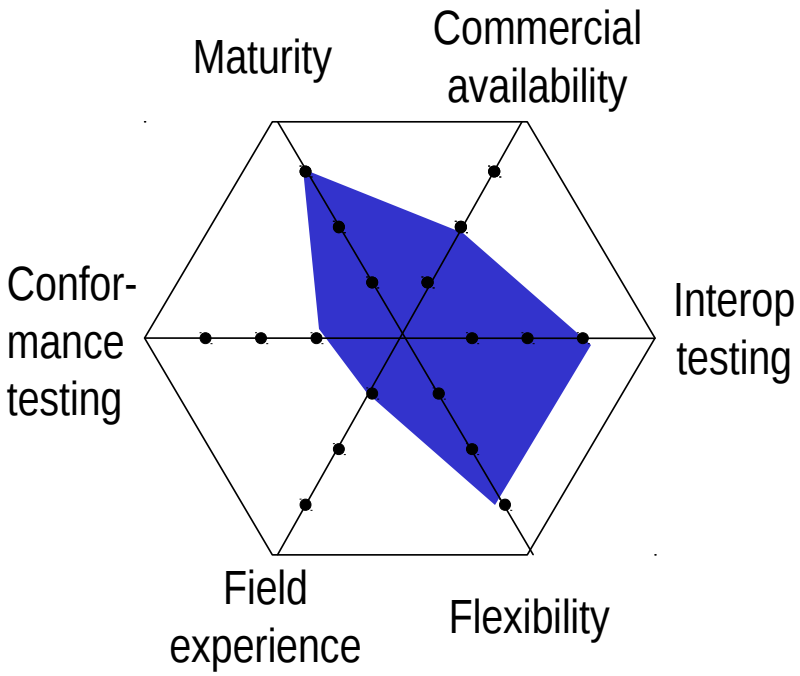
Information Cards

Information Cards – What its for

- User centric identity specifications
- Leverages WS-Trust security token service (STS), and other WS-* specifications
- Implemented in Windows Vista CardSpace and Higgins Open Source project
- Consider Information Cards for user-centric pilots, projects or architectures

How it scores

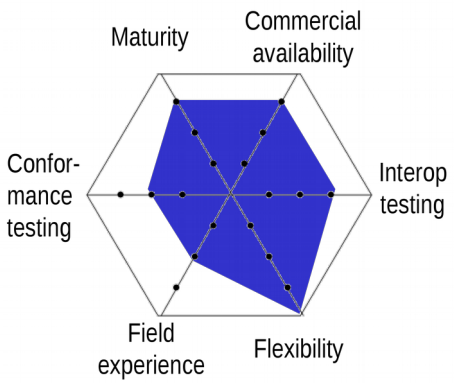
(Feb 2008)



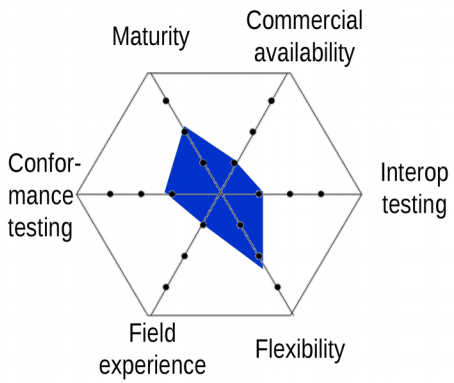
How Identity Standards Score

Summing up standards (Feb 2008)

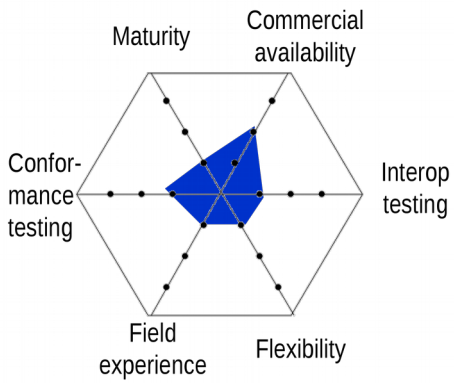
SAML



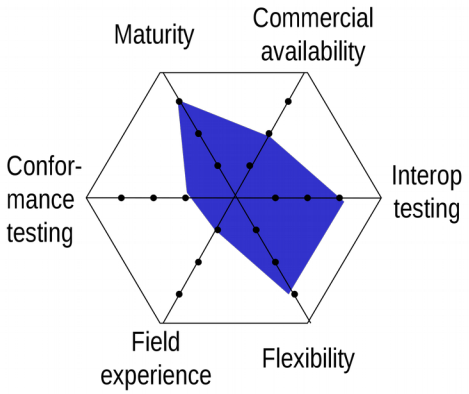
XACML



SPML



Information Cards



Are they delivering the promise of interoperability?

SAML

Yes

XACML

Not yet
(Extends SAML)

SPML

Not yet
(Ahead of its time)

Information Cards

Looks promising

Other Standards or Specifications

OpenID

- Popular user-centric system, allows chaotic expansion
- No trust model, insecure protocol, no id assurance

WS-Federation passive profile

- Required for Microsoft environment
- Unhelpful, duplicates SAML functionality

Liberty Alliance

- Morphed into deployment forum, ceded much of its standards work to OASIS
- Its most important specifications now part of SAML 2.0
- Identity Assurance Framework & Concordia initiatives promising

Authorization

Most identity interoperability work emphasizes authentication, and exchanges

Authorization is an even more difficult problem

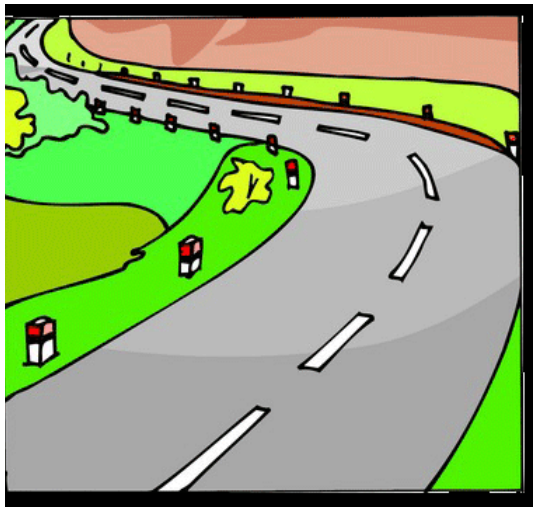
- Tied to business context: Environment, risk, liability
- Tools cannot establish or maintain business relationships, trust, define roles or initial agreements
- Tools useful for yes/no decisions, passing roles or attributes

Need up front work to bound the context

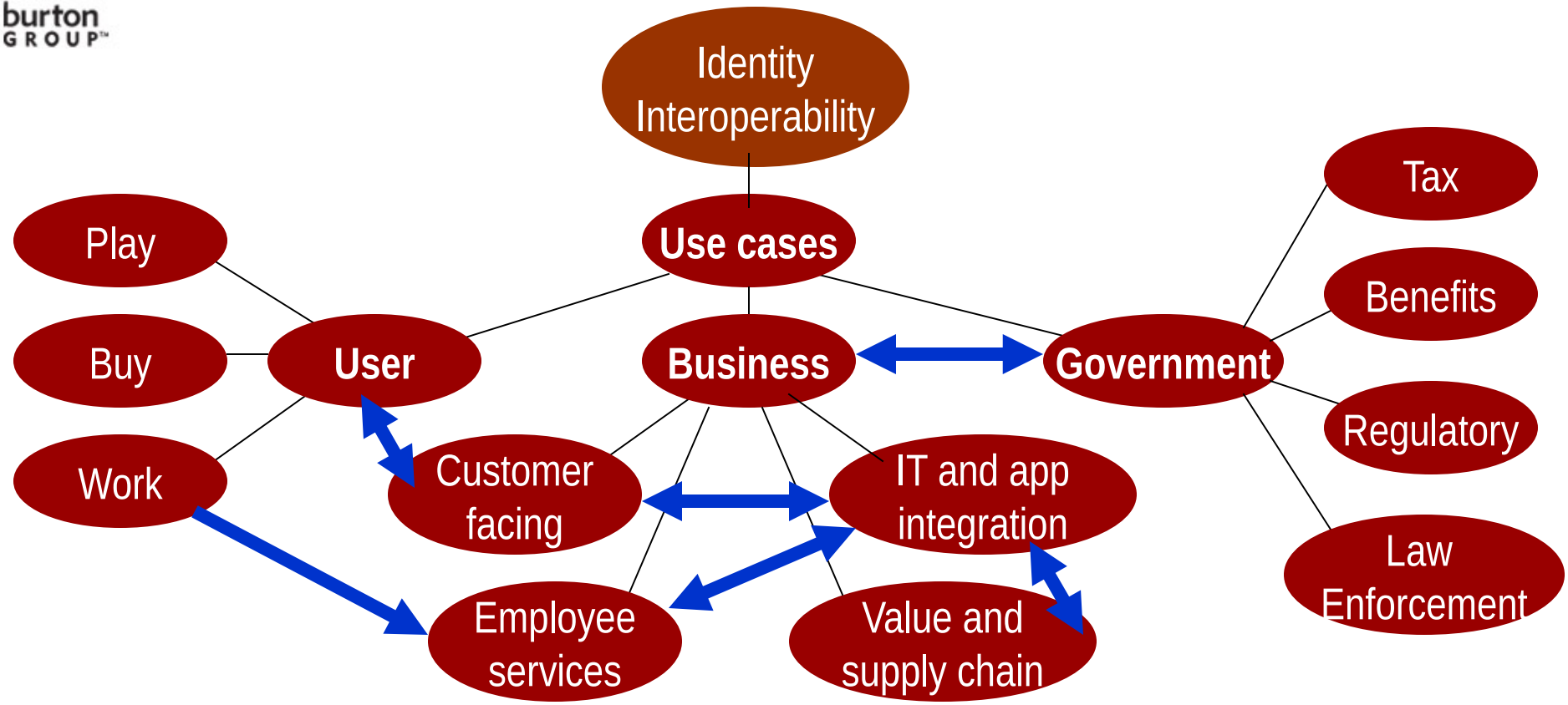
Then see what can be automated

Roadmap for Presentation

- Discuss interoperability use cases, and adoption
- Evaluate key standards
- ***Conclusion and recommendations***



Use Cases are not Islands



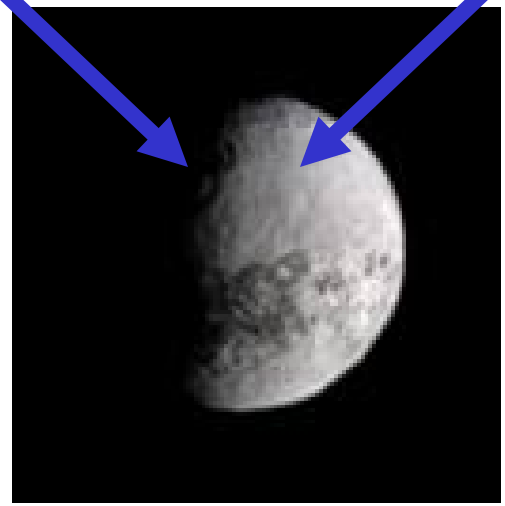
Convergence

How and when will we see ubiquitous identity interoperability?

On the dark side:

We don't really know yet; too many hard unsolved problems

- Business models
- Trust models
- Scalability of deployment



On the bright side:

We have standards and products for many tactical opportunities

- New apps
- Improve convenience
- Cost savings
- Increase assurance

Identity System Standards

Recommendations

- Don't force fit all projects into a consolidated solution
- Tailor deployments for each IT tier (organizational, regional, functional)
- Stick to patterns that work – enterprise portal, value chain, industry hub
- Pick the right standard(s) for the task
  - But be prepared for issues that may arise with new standards
- Keep the solution as simple as possible
- Develop and document implementation procedures
  - Encourage 'best practices'
  - Makes the process repeatable

References

Burton Group *Identity and Privacy Strategies*

- Let's Get Logical: The Convergence of Physical Access Control and Identity Systems
- Federation Products 2008
- Picking the Right Federation Product for the Job
- Federation's Future in the Balance: Teetering Between Ubiquity and Mediocrity
- Information Card Landscape
- Comparing SAML, WS-Trust, and OpenID
- Business and Legal Issues in Federation
- Federated Identity Technical Position
- SAML 2.0: Convergence Point for Browser-Based Federation

A Client-side CardSpace-Liberty Integration Architecture

Waleed A. Alrodhan
Royal Holloway, University of London
Egham, Surrey, TW20 0EX
United Kingdom
W.A.Alrodhan@rhul.ac.uk

Chris J. Mitchell
Royal Holloway, University of London
Egham, Surrey, TW20 0EX
United Kingdom
C.Mitchell@rhul.ac.uk

ABSTRACT

Over the last few years, many identity management schemes, frameworks and system specifications have been proposed; however these various schemes and frameworks are typically not interoperable. In this paper we propose an approach to enable interoperation between two of the most prominent identity management schemes, namely the Liberty Alliance Project scheme (specifically the ID-FF LEC Profile) and the Microsoft CardSpace (formerly known as InfoCard) scheme. This integration should enhance interoperability by enabling users to make use of identity management systems even if the system participants are using different schemes. The main advantages and disadvantages of the proposed integration model are also investigated.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and protection

General Terms

Identity Management

Keywords

CardSpace, Liberty, Federation, Integration

1. INTRODUCTION

It has become common for Internet users to access multiple independent systems in a single working session, and hence users have a need for multiple digital identities. However, managing these digital identities can be a complex and difficult job, and to solve this dilemma a number of Identity Federation systems have been proposed and deployed. These systems are typically not interoperable, which makes it difficult to use them in open environments such as the Internet.

This paper proposes an approach to address this problem. Specifically, it proposes a method to enable interoperation between the Liberty Alliance Project scheme and the Microsoft CardSpace scheme.

The remainder of this paper is organised as follows. Section 2 provides an overview of the Liberty Alliance Project and the Microsoft CardSpace identity management system. Section 3 presents the proposed integration model. In section 4 we provide an operational analysis of the proposed integration model, section 5 contains a brief review of related work, and section 6 concludes the paper.

IDtrust '08, March 4-6, 2008 Gaithersburg, MD. Copyright 2008 ACM 1978-1-60558-066-1 ...$5.00.

2. THE LIBERTY ALLIANCE PROJECT AND MICROSOFT CARDSPACE

This section provides a brief introduction to the two identity management architectures for which interoperation is enabled, namely the Liberty Alliance Project scheme and Microsoft CardSpace.

2.1 The Liberty Alliance Project and the ID-FF Single Sign-On Profiles

The Liberty Alliance Project (www.projectliberty.org) is an industry collaboration that was started in December 2001 by 16 major companies, including Sun, GM, United Airlines, and France Télécom. This collaboration now involves more than 150 members, including government agencies, companies, banks and universities. According to the project website, there are more than 400 million Liberty-enabled identities and clients across the world.

The Liberty Alliance Project (henceforth abbreviated to Liberty) aims to build open standard-based specifications for federated identity, provide interoperability testing, and to help provide solutions to identity theft. Liberty also aims to establish best practices and business guidelines for identity federation.

Figure 1 shows the general Liberty model, which is essentially a single sign-on model [2]. In this model, a principal (or user) can federate its various identities to a single identity issued by an identity provider, so that the user can access services provided by service providers belonging to the same circle of trust by authenticating just once to the identity provider. This, of course, relies on a pre-established trust relationship between the identity provider and every service provider in the circle of trust. The model provides a level of pseudonymity and unlinkability via the use of pseudonyms instead of real identifiers in the communications between the identity provider and the service providers, and this enhances user privacy. In the example shown in figure 1, the principal has federated its identities within two distinct circles of trust, which results in the user having two

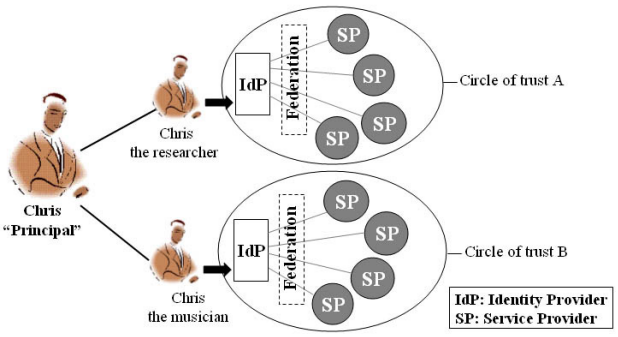


Figure 1: The Liberty model.

identities, one for circle of trust A and the other for circle of trust B. It merits mentioning that these two identities could also be federated, but this would require a pre-established trust relationship between the identity providers.

As shown in figure 2, the Liberty implementation specifications are divided into three frameworks: the Identity Federation Framework (ID-FF) [12], the Identity Web Services Framework (ID-WSF) [11] and the Service Interface Specifications (ID-SIS) [5] and [6]. In this paper, we focus on the ID-FF.

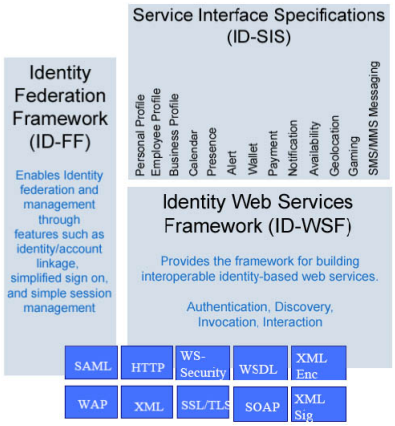


Figure 2: The Liberty Frameworks.

The ID-FF provides approaches for implementing the federation and single sign-on model, and the techniques needed for that model, including session management and identity/account linkage. In order to realise this federation model, a set of profiles is required (so called ID-FF Liberty profiles). An ID-FF Liberty profile may best be defined as the combination of message content specifications and message transport mechanisms for a single client type (that is, user agent) [3].

There are many types of ID-FF Liberty profile, including Single Sign-On and Federation Profiles, Register Name Identifier Profiles, Identity Federation Termination Notification Profiles, Single Logout Profiles, Identity Provider Introduction, NameIdentifier Mapping Profile and NameIdentifier Encryption Profile. In this paper we are primarily concerned with the Single Sign-On and Federation Profiles. The currently defined Single Sign-On profiles are the Artifact profile, the Browser POST profile and the Liberty-enabled client and proxy (LEC) profile [3].

2.2 Microsoft CardSpace

CardSpace is the name for a Microsoft WinFX software component that is built on the concept of the "identity metasystem". This identity metasystem is designed to comply with the Laws of Identity, as promulgated by Microsoft [1]. The metasystem provides a way to represent identities using claims, and a means to bridge technology and organisational boundaries using claims transformations [7]. The CardSpace identity management architecture is designed to provide the user with control over his digital identities in a user friendly manner, and to tackle identity management security problems such as breaches of privacy and identity theft, with no single identity authority control. CardSpace works with Internet Explorer browsers (CardSpace plug-ins for browsers other than Microsoft Internet Explorer can also be developed, such as the Firefox Plug-in[1]).

The concept behind CardSpace is relatively simple; it is based on the identification process we experience in the real world when using physical identification cards. In the CardSpace system, an identity provider issues virtual identification cards (named InfoCards) to users, who can later use them to identify themselves to any service provider who trusts this identity provider. It merits mentioning here that the InfoCard is stored in the user's machine and does not contain any security-sensitive information.

CardSpace is an identity federation model that essentially facilitates a single sign-on service, although it is not clear from the published Microsoft papers and documents how single sign-on sessions are handled within the metasystem. The CardSpace metasystem makes use of Web Services (WS-*) protocols to achieve its objectives. Note that most of these protocols make use of a *Security Token Service* [4].

2.3 Message Flows within the Liberty ID-FF LEC Profile and the CardSpace Framework

Before describing the proposed integration model, we briefly describe the message flows within both the Liberty ID-FF LEC profile and the CardSpace framework. There are three main roles within both identity management architectures:

1. The Identity Provider or Identity Issuer (IdP), which issues the identity to the user;
2. The Service Provider (SP), as referred to in the Liberty specifications, or the Relying Party (RP), in Microsoft terminology; the SP or RP needs to identify the user before providing services to him/her;
3. The Principal or User.

It is important to mention that, in order for a principal to employ Liberty federation using the LEC profile, the principal must possess a Liberty-Enabled browser in order to handle and understand the messages sent and received. To make the browser Liberty-Enabled, the principal needs to install certain java components on the machine; such components can be downloaded freely from the Internet (e.g. the SecureID ID-FF 1.1 and ID-FF 1.2 Java Toolkits[2], and the

[1]http://xmldap.blogspot.com/2006/05/firefox-identity-selector.html

Sun FederationSPAdapter[3]).

Figure 3 shows the message flow within the ID-FF LEC profile, if we assume that the client has already been authenticated by the IdP. Note that the Liberty-Enabling component must be installed on the Principal's PC prior to performing the protocol. The main steps in the protocol are as follows:

1. **User Agent → SP** : Service Request (HTTP Request with Liberty Enabled Header)
2. **SP → User Agent** : Authentication Request + "optionally" an IdP List
3. **User Agent or User** : Selects the IdP to be used
4. **User Agent → IdP** : SAML-Assertion Request
5. **IdP → User Agent** : SAML-Assertion Response
6. **User Agent → SP** : Authentication Response + SAML-Assertion (within an HTML Form)
7. **SP → User Agent** : Service Granted!

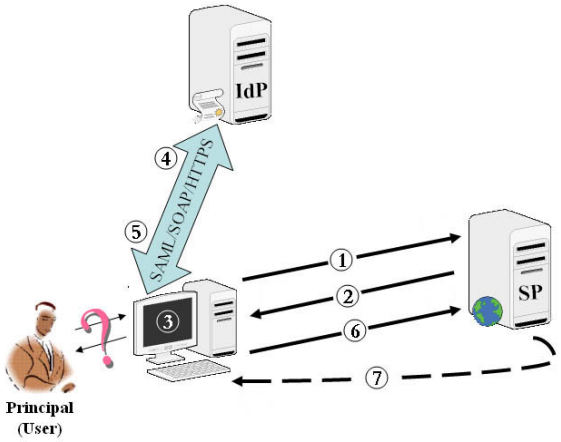


Figure 3: The ID-FF LEC profile message flow.

The SAML messages in steps 4 and 5 are bound to SOAP, which is carried over an SSL connection to provide confidentiality (integrity is preserved using an XML-Signature). As shown in the figure, how the IdP is selected in step 3 is not specified in the Liberty specifications.

Figure 4 shows the message flow within the CardSpace framework. As shown in the figure, there is an additional component here, namely the *Security Token Service* (henceforth abbreviated to STS), which is responsible for the security policy and token management within the IdP (and optionally within the RP). The User Agent here is essentially an CardSpace enabled browser (also called the *Service Requestor*). Note that the CardSpace component must be installed on the Principal's PC prior to step 3 of the protocol. The main steps in the protocol are as follows:

1. **User Agent → RP** : HTTP GET Login HTML Page Request
2. **RP → User Agent** : HTML Login Page + CardSpace Tags (XHTML or HTML object tags)

[2]http://www.sourceid.org
[3]http://docs.sun.com/source/819-4682

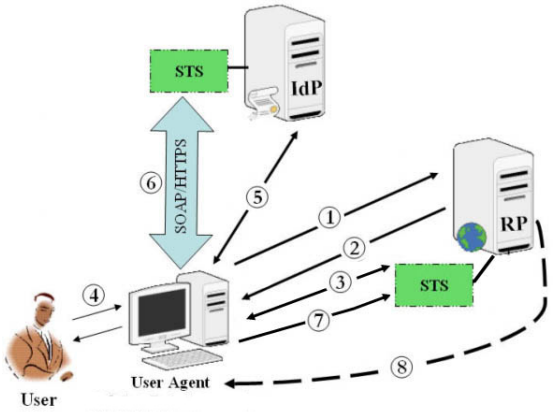


Figure 4: The MS-CardSpace profile message flow.

3. **User Agent ↔ RP-STS** : User Agent retrieves Policy via *WS-SecurityPolicy*
4. **User Agent ↔ User** : User picks an InfoCard
5. **User Agent ↔ IdP** : User Authentication
6. **User Agent ↔ IdP-STS** : User Agent retrieves security token via *WS-MetadataExchange* and *WS-Trust*
7. **User Agent → RP-STS** : User Agent presents the security token via *WS-Trust*
8. **RP → User Agent** : Welcome, you are now logged in!

Unlike the ID-FF LEC profile, in Step 4 of the CardSpace message flow the user is presented with the InfoCards that can be used to identify himself to that particular RP, and picks one of them. The WS-MetadataExchange and WS-Trust messages in step 6 are transported using SOAP, which is carried via an SSL connection to provide confidentiality (integrity is preserved using an XML signature). If the RP does not have an STS server, the messages in steps 3 and 7 will be carried using HTTP over an SSL/TLS channel.

In the remainder of this paper we assume that all SPs and IdPs support at least one of the two identity architectures (either Liberty or CardSpace, or both). In such a situation, Windows users will face one of the following two scenarios:

- *Scenario A*: The Identity Provider/Identity Issuer supports both identity management architectures (Liberty and CardSpace). In this case the IdP perform the difficult task of maintaining two different identity architectures, thereby providing flexibility to users when they federate their identities with the SPs or RPs. Users are able to access services provided by Service Providers regardless of which identity management architecture they adopt.
- *Scenario B*: The Identity Provider/Identity Issuer supports only one identity management architecture (either Liberty or CardSpace). In this case, users are only able to access Service Providers using the identity management architecture supported by their Identity Issuer or IdP.

In Scenario B, Windows users have a compatibility problem if the Relying Party is CardSpace-Enabled while their IdP is Liberty-Enabled, or vice versa. Table 1 shows the applicability of the identity management systems in all of the

possible scenarios that might occur; L.E. stands for Liberty Enabled, and CS.E. stands for CardSpace Enabled. The (✓) sign indicates that there is no compatibility problem, whereas the (×) sign indicates the opposite.

Table 1: The applicability of the identity management architectures.

User Agent	L.E. IdP L.E. SP	CS.E. IdP CS.E. RP	L.E. IdP CS.E. RP	CS.E. IdP L.E. SP
Unenhanced	×	×	×	×
L.E.	✓	×	×	×
CS.E.	×	✓	×	×
L.E. and CS.E.	✓	✓	×	×

3. THE INTEGRATION MODEL

There is a noticeable similarity between the message flow within the ID-FF LEC profile and the message flow within the CardSpace framework. This similarity can be exploited in order to integrate the two identity management architectures. This section presents the motivation for this integration proposal, and describes the proposed integration model.

3.1 Motivation

Liberty is currently the leading identity management architecture for identity federation, and it has gained the acceptance of a number of technology-leading companies and organisations. By contrast, the CardSpace metasystem currently only works on the Windows operating system (this seems likely to continue, at least in the near future), and is deployed freely with Windows Vista, with backwards compatibility with Windows XP. Given the wide use of Windows, CardSpace is likely to have a significant impact despite this restriction. Thus, enabling integration between these two systems is likely to be of significant benefit to a wide range of service providers and users. It seems that interoperation between Liberty and CardSpace can be achieved by integrating the frameworks.

The CardSpace identity management architecture consists of two parts:

1. *The user agent supporting components*: i.e. the service requestor and the identity selector. These components are responsible for managing the cards and communicating with other parties in the model. It appears that these components could be integrated with the Liberty ID-WSF services; however, how this might be achieved is beyond the scope of this paper.
2. *The Identity Framework*: i.e. the message flow and the rules for communication between the parties. This framework is similar to the Liberty ID-FF LEC profile, in which the user agent is "Liberty-enabled" in the same way as the user agent is "CardSpace-enabled" in the CardSpace framework.

3.2 Integrating the two schemes

In the integration model we propose, we introduce an identity management architecture adaptor that is both Liberty-enabled and CardSpace-enabled. We propose placing this adaptor in the user's machine. This gives the user the ability to use any IdP, and access a service provided by any RP or SP. This means that, in the presence of such an adaptor, we can change the last two entries in the fifth row of table 1 to (✓) instead of (×), which implies that such a user would have the ability to make use of an identity system regardless of the identity management architecture adopted by the RP or the SP. That is, this integration model is designed to resolve the incompatibility situation that may occur if the IdP is Liberty-enabled and the RP is CardSpace-enabled, or vice versa.

Before describing the proposed integration model, we outline a major difference between the scope of the Liberty ID-FF and the CardSpace framework. The CardSpace framework's main goal is to support the authentication of a user by a trusted third party (i.e. an IdP) and the provision of assertions by the trusted third party to the SP that claims regarding attributes of the user are correct. By contrast, in Liberty, the ID-FF is designed to achieve Identity Federation by asserting to the SP that the user has been successfully authenticated by a trusted third party (i.e. the IdP) using an authentication method, and no claims (i.e. user attributes) are involved in the authentication process. Nevertheless, asserting user attributes by a trusted third party can be achieved using the Liberty protocols; however, this falls under the Liberty ID-WSF, which is a different framework, as discussed earlier in this paper.

The *identity management architecture adaptor* proposed here is a piece of software installed on the user's machine which understands both the Liberty and CardSpace frameworks, and their message flows and formats. The adaptor's main job is to interpose itself between IdPs and SPs/RPs adhering to different identity management architectures, in order to translate particular messages generated by one party to the other. We consider operational details of the identity management architecture adaptor later in this section.

Before presenting the integration model and its message flow, we note the following restrictions on its operation:

1. Given that the Liberty specifications are based only on SAML-Assertion tokens, only SAML-Assertion tokens are permitted within the integration model, i.e. other security tokens supported by CardSpace (e.g. Kerberos v5 tickets) are not permitted. The SAML-Assertion tokens will simply be forwarded from the IdP to the SP/RP.
2. For the proof of rightful possession of the security token, only asymmetric techniques are permitted. In the asymmetric proof technique, the IdP inserts the public key of the User Agent in the security token, so that the User Agent can use its private key to prove rightful possession of the token to the RP (or SP). There is a clear resemblance between the CardSpace *asymmetric* proof-key technique [8], and SAML *holder-of-key* confirmation method [9]; hence, mapping between these techniques is viable using a relatively simple process.
3. For a CardSpace-enabled RP and a Liberty-enabled IdP, the identity management architecture adaptor will discard any token freshness restrictions requests imposed by the RP, since the Liberty-enabled IdP may not be capable of understanding them.

In the CardSpace framework, the claims to be asserted in the RP Security Policy are represented as attributes of an Attribute Statement, that is contained within SAML-

Assertion requests and responses exchanged before the security token can be retrieved from the IdP. However, in the Liberty ID-FF, the IdP does not expect any SAML Attribute Statements in the *AuthenticationRequestEnvelope*, i.e. the SOAP envelope that carries the authentication requests. This presents a problem that must be solved before the Identity Management Architecture Adaptor can convert the relevant messages. We propose two possible solutions to this problem.

1. We could convert the claims in the CardSpace Security Policy into attributes within a SAML Attribute Statement in the *AuthenticationRequestEnvelope*. However, for this to work, we must ensure that the IdP is able to process such statements in order to assert them in the *AuthenticationResponseEnvelope*. However, this solution goes outside the Liberty standards, and would potentially require the Liberty-Enabling software component to be modified.
2. Alternatively, we could make the integration model only accept CardSpace Security Polices with no claims listed, and hence the RP that issues the polices will only require an assertion that the user has been authenticated by a trusted third party. However, this solution will severely impact on the usability of the integration model.

Figure 5 shows the message flow for the integration model in the case where the IdP is CardSpace-enabled and the SP is Liberty-enabled. Note that the Liberty-Enabling component must be active on the Principal's PC prior to performing the protocol, and the CardSpace component must be active on the Principal's PC prior to step 3 of the protocol. Here, the message flow would be:

1. **User Agent → SP** : Service Request (HTTP Request with Liberty Enabled Header)
2. **SP → User Agent** : Authentication Request + "optionally" an IdPs List
   - [The identity management architecture adaptor converts the Liberty `Authentication Request`, received from the SP, into a CardSpace `RP Retrieved Security Policy`, and forwards it to the CardSpace-enabling component]
3. **User Agent ↔ User** : User Picks an InfoCard
4. **User Agent ↔ IdP** : User Authentication
5. **User Agent ↔ IdP-STS** : User Agent retrieves security token via *WS-MetadataExchange* and *WS-Trust*
   - [The identity management architecture adaptor converts the CardSpace `Retrieved Security Token`, received from the IdP-STS, into a Liberty `Authentication Response`, and forwards it to the Liberty-enabling component]
6. **User Agent → SP** : Authentication Response + SAML-Assertion (within the HTML Form)
7. **SP → User Agent** : Service Granted!

Figure 6 shows the message flow for the integration model in the case where the IdP is Liberty-enabled and the RP is CardSpace-enabled. Note that the Liberty-Enabling component must be active on the Principal's PC prior to step 4 of the protocol, and the CardSpace component must be active on the Principal's PC prior to step 3 of the protocol. Here, the message flow would be:

1. **User Agent → RP** : HTTP GET Login HTML Page Request
2. **RP → User Agent** : HTML Login Page + CardSpace Tags (XHTML or HTML object tags)
3. **User Agent ↔ RP-STS** : User Agent retrieves Policy via *WS-SecurityPolicy*
   - [The identity management architecture adaptor converts the CardSpace `RP Security Policy`, received from the RP, into a Liberty `Authentication Request`, and forwards it to the Liberty-Enabling component]
4. **User Agent or User** : Selects the IdP to be used
5. **User Agent → IdP** : SAML-Assertion Request
6. **IdP → User Agent** : SAML-Assertion Response
   - [The identity management architecture adaptor converts the Liberty `Authentication Response`, received from the IdP, into a CardSpace `IdP-STS Retrieved Security Token`, and forwards it to the CardSpace component]
7. **User Agent → RP-STS** : User Agent presents the security token via *WS-Trust*
8. **RP → User Agent** : Welcome, you are now logged in!

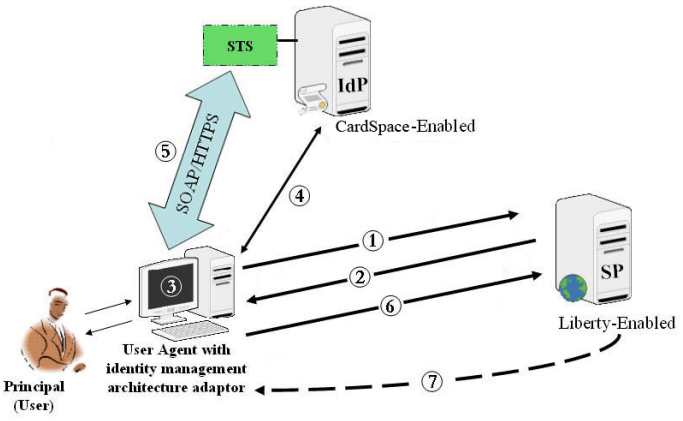


Figure 5: Message flow within the integration model (a).

As described above, the identity management architecture adaptor interposes itself between the IdP and the SP/RP in order to translate messages at certain stages of the protocol run. The identity management architecture adaptor must be able to make four types of message conversion, including two token forwarding operations. Figure 7 shows the message types that need to be converted by the identity management architecture adaptor, along with the respective message formats.

Thus, the identity management architecture adaptor software must perform two types of task:

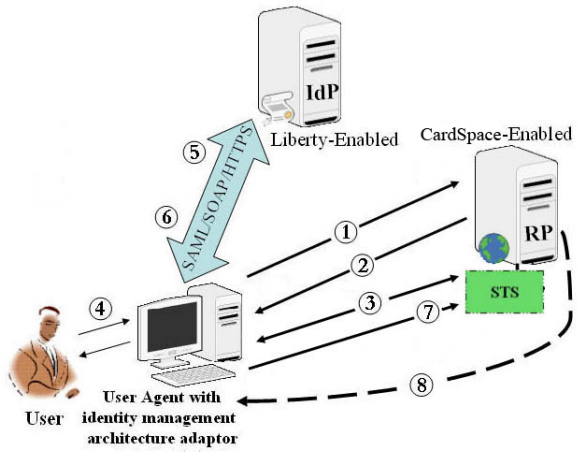


Figure 6: Message flow within the integration model (b).

- Convert the formats of four types of message.
- Forward the converted messages to either liberty enabling or CardSpace software components.

For the first task, implementing the required message conversions should not be difficult, since all the messages formats are open and published (XML definitions). However, the second task is not so straightforward, since it requires a well-defined set of APIs in order for the identity management architecture adaptor software to communicate with the Liberty-Enabling and CardSpace components. The precise details of the operation of the adaptor will therefore depend on how these components are implemented. Indeed, it is possible that the adaptor could be integrated into the respective components.

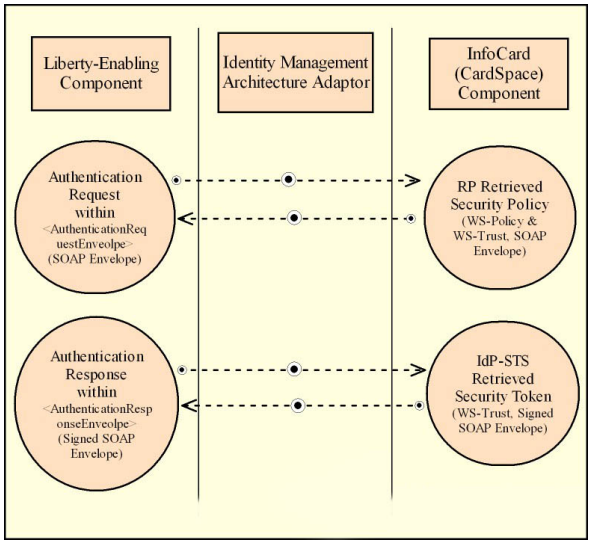


Figure 7: Message format conversions.

4. AN ANALYSIS OF THE INTEGRATION MODEL

The integration model takes advantage of the similarity between the ID-FF LEC profile and the CardSpace framework, and this should help to reduce the effort required for full system integration. Moreover, the proposed integration model is designed to be implemented without the need for technical cooperation between Microsoft and Liberty.

Implementing such a model might be non-trivial task, but the benefits could be significant. If interoperability between leading identity management systems is not supported, then this could be a major obstacle to the global adoption of such schemes.

The proposed model is designed to integrate two frameworks with somewhat different scopes. This difference in scope has given rise to the most serious obstacle facing this integration model, namely the dilemma of translating the CardSpace Security Policy claims. Neither of the two solutions proposed in this paper are ideal, since they either involve reducing the applicability of the scheme, or making modifications to the core of the Liberty-Enabling software component.

One potential limitation of the proposed model is that the user agent must be both CardSpace-enabled and Liberty-enabled. However, Windows Vista user agents (i.e. browsers) will, by default, be CardSpace-enabled, and to make the browsers Liberty-enabled will simply require the installation of certain java scripts on user machines; as a result this does not seem to be a major issue. Another possible limitation of the proposed model is the added restrictions on the token type, encryption and freshness requests; these restrictions will prevent the users from utilising certain features offered by CardSpace. However, these restrictions only affect the CardSpace framework because, from the Liberty perspective, token handling will remain the same [10].

A further possible limitation is the necessity for interactions between the adaptor and the CardSpace metasystem; because of the closed nature of CardSpace, this might not be straightforward, unless a well-defined set of APIs is publicly available. Finally, use of the proposed integration model will result in a delay at the user system while the identity management architecture adaptor performs the necessary conversions; however, any such delay is likely to be very small.

5. RELATED WORK

Recently, the Bandit[4] project has developed an open source integration system between Liberty and CardSpace (a demo is available at the project's web site). Although the source code is provided, it seems that there is no published specification of the integration system, which makes it difficult to discover exactly how the Bandit scheme works.

One obvious difference between the integration scheme we propose in this paper and the Bandit integration scheme relates to the location of the integration adapter. Unlike the scheme discussed above, in Bandit the integration adapter is placed on the RP (or SP), not on the user machine. We believe that placing the integration adapter on the user machine increases the usability of the scheme. It is much simpler for a user to deploy an integration scheme, and it seems likely that many well-known service providers will only sup-

[4]http://www.bandit-project.org

port a single identity management system for operational and commercial reasons.

6. CONCLUSIONS

In this paper we have proposed a model enabling integration of Liberty and CardSpace. This integration model takes advantage of the similarity in the message flows between the Liberty ID-FF LEC framework and the CardSpace framework. The proposed integration model is based on a client-side identity management architecture adaptor that converts the format of messages within the message flows of the two schemes. We have also presented an analysis of the main limitations and benefits of our proposed integration model.

The model is designed to integrate two frameworks with somewhat different scopes, and this has caused certain technical problems, in particular in translating the CardSpace claims. In this paper we have suggested possible solutions for such problems.

We believe that enabling interoperation between the two most prominent identity federation architectures could be of major benefit to both the users and producers of these systems.

7. REFERENCES

[1] K. Cameron. The laws of identity, May 2005. Microsoft Corporation.

[2] K. Cameron and M. B. Jones. Design rationale behind the identity metasystem architecture, February 2006. Microsoft Corporation.

[3] S. Cantor, J. Kemp, and D. Champagne (editors). Liberty ID-FF bindings and profiles specification — 1.2-errata-v2.0, 2004. Liberty Alliance Project.

[4] M. B. Jones. A guide to supporting InfoCard v1.0 within web applications and browsers, March 2006. Microsoft Corporation.

[5] S. Kellomai (editor). Liberty ID-SIS employee profile service specification — version: 1.0, 2003. Liberty Alliance Project.

[6] S. Kellomai (editor). Liberty ID-SIS personal profile service specification — version: 1.0, 2003. Liberty Alliance Project.

[7] Microsoft Corporation. A technical reference for InfoCard v1.0 in windows, August 2005.

[8] Microsoft Corporation and Ping Identity Corporation. A guide to integrating with InfoCard v1.0, August 2005.

[9] R. Monzillo, C. Kaler, A. Nadalin, and P. Hallem-Baker (editors). Web Services Security: SAML Token Profile 1.1, February 2006. OASIS Standard Specification, OASIS Open.

[10] P. Thompson and D. Champagne (editors). Liberty ID-FF implementation guidelines — version 1.2, 2004. Liberty Alliance Project.

[11] J. Tourzan and Y. Koga (editors). Liberty ID-WSF web services framework overview — version: 1.1. Liberty Alliance Project.

[12] T. Wason (editor). Liberty ID-FF architecture overview — version: 1.2. Liberty Alliance Project.

A Client-side CardSpace-Liberty Integration Architecture

Waleed A. Alrodhan
Royal Holloway, University of London
Information Security Group

7th Symposium on Identity and Trust on the Internet (IDtrust 2008)

Agenda

- Introduction
- Liberty Alliance Project (ID-FF LEC profile)
- Microsoft CardSpace
- Integrating the two schemes
- Analysis

Introduction

- It has become common for Internet users to access multiple independent systems in a single working session.
- Hence, users need multiple digital identities.
- There are many solutions. (e.g. Federation, User Centric, CardSpace, etc.)
- Interoperability?

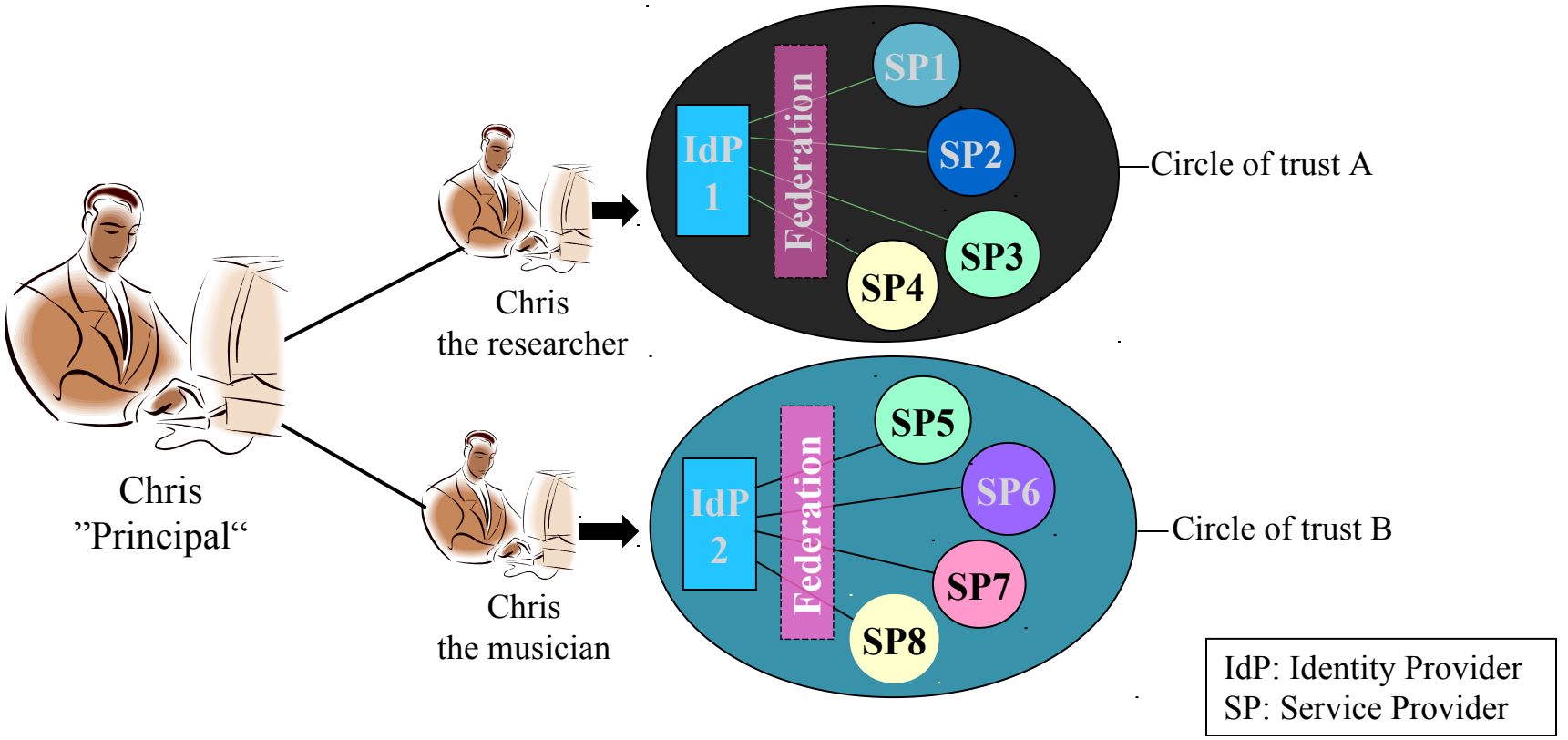
Liberty Alliance Project

Introduction

- An industry collaboration, started in December 2001.
- Liberty aims to build open standard-based specifications for federated identity, provide interoperability testing, and to help provide solutions to identity theft.
- There are more than 40 million liberty-enabled identities and clients across the world (LAP, 2005).

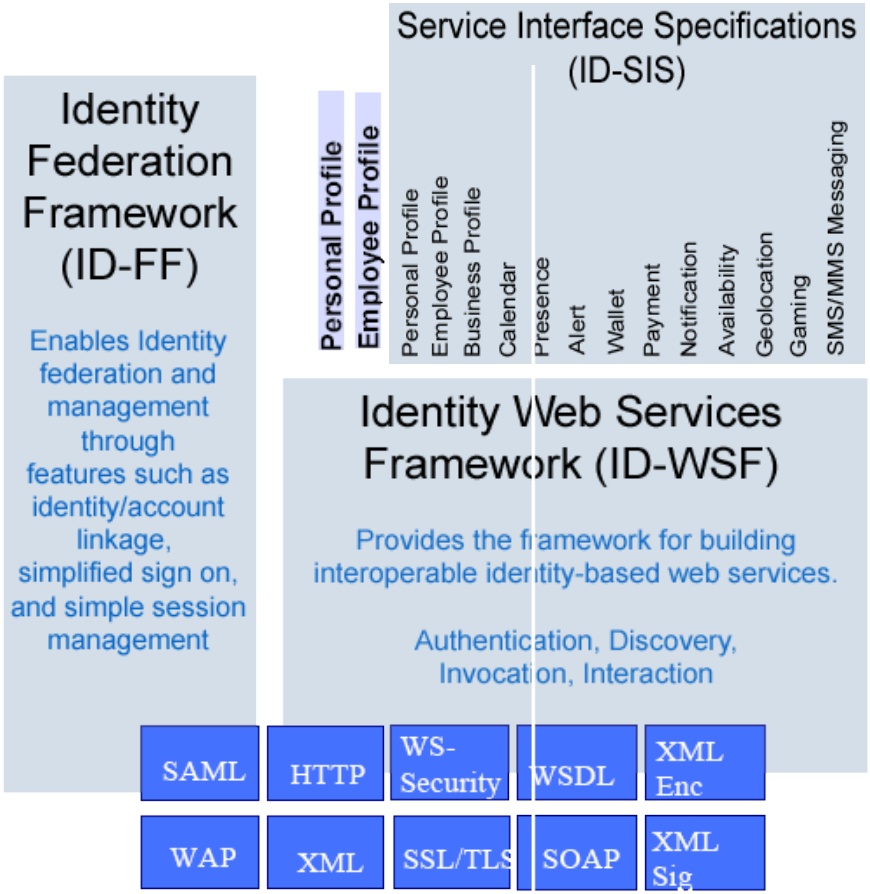
Liberty Alliance Project

- The Basic Federation Model



Liberty Alliance Project

- The Specifications



The charts in this slide are taken from the Liberty Alliance Project website

Liberty Alliance Project

The ID-FF Liberty Profiles

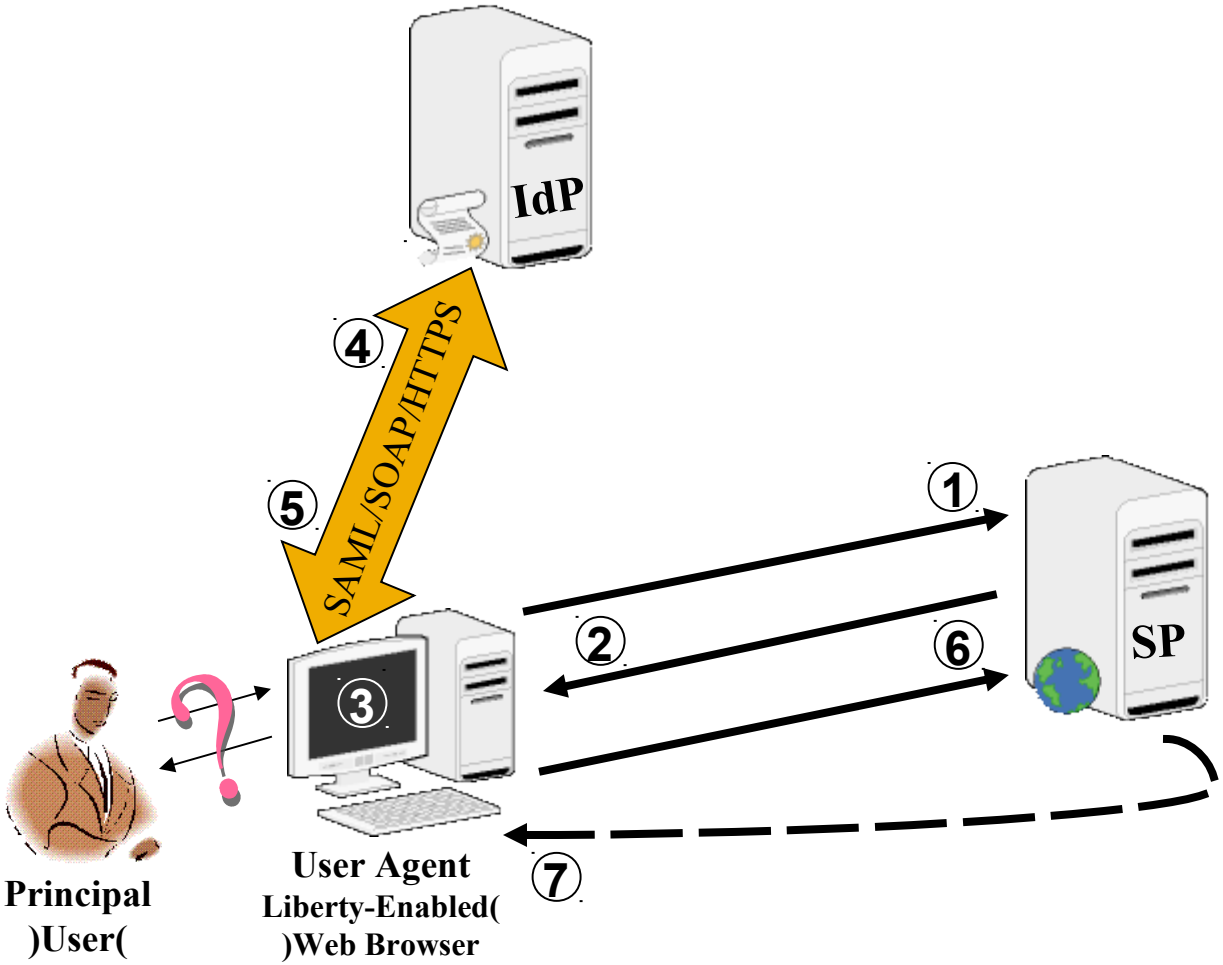
- "The combination of message content specification and message transport mechanisms for a single client type (that is, user agent) is termed a Liberty profile."
  (Liberty Alliance Project - Liberty ID-FF Bindings and Profiles Specification)

- There are many Profiles:
  - Single Sign-On and Federation Profiles, Register Name Identifier Profiles, Identity Federation Termination Notification Profiles, Single Logout Profiles, Identity Provider Introduction, NameIdentifier Mapping Profile, NameIdentifier Encryption Profile

- There are three Single Sign-On and Federation Profiles:
  1. Artifact profile
  2. Browser POST profile
  3. Liberty-enabled client and proxy profile

Liberty Alliance Project

- Liberty-Enabled Client and Proxy Profile



It is assumed that the client has already been Authenticated by the IdP

User Agent → SP: Service Request .1
)HTTP Request with Liberty Enabled Header(

: **SP → User Agent**. 2
+Authentication Request
optionally" an IdPs List"

:**User Agent OR User**. 3
Obtaining IdP

:**User Agent → IdP**. 4
SAML-Assertion Request

:**IdP → User Agent**. 5
SAML-Assertion Response

:**User Agent → SP**. 6
+Authentication Response
SAML-Assertion (within the HTML Form)
)Redirect, HTTP (HTML Form) POST(

:**SP → User Agent**. 7
!Service Granted

Microsoft CardSpace

Introduction

- WinFX software component that is built on the concept of the “identity metasystem”.
- Designed to provide the user control over his digital identities in a user friendly manner, and to tackle problems such as privacy breaching and identity theft, with no single or central identity authority control.

Microsoft CardSpace

Introduction II

- Currently deployed with Windows Vista. (works with multiple browsers)
- The identity is defined as a set of claims, where the claim is an assertion of the truth of something.
- Based on the identification process we experience in the real world when using physical identification cards.
- Laws of Identity.

Microsoft CardSpace

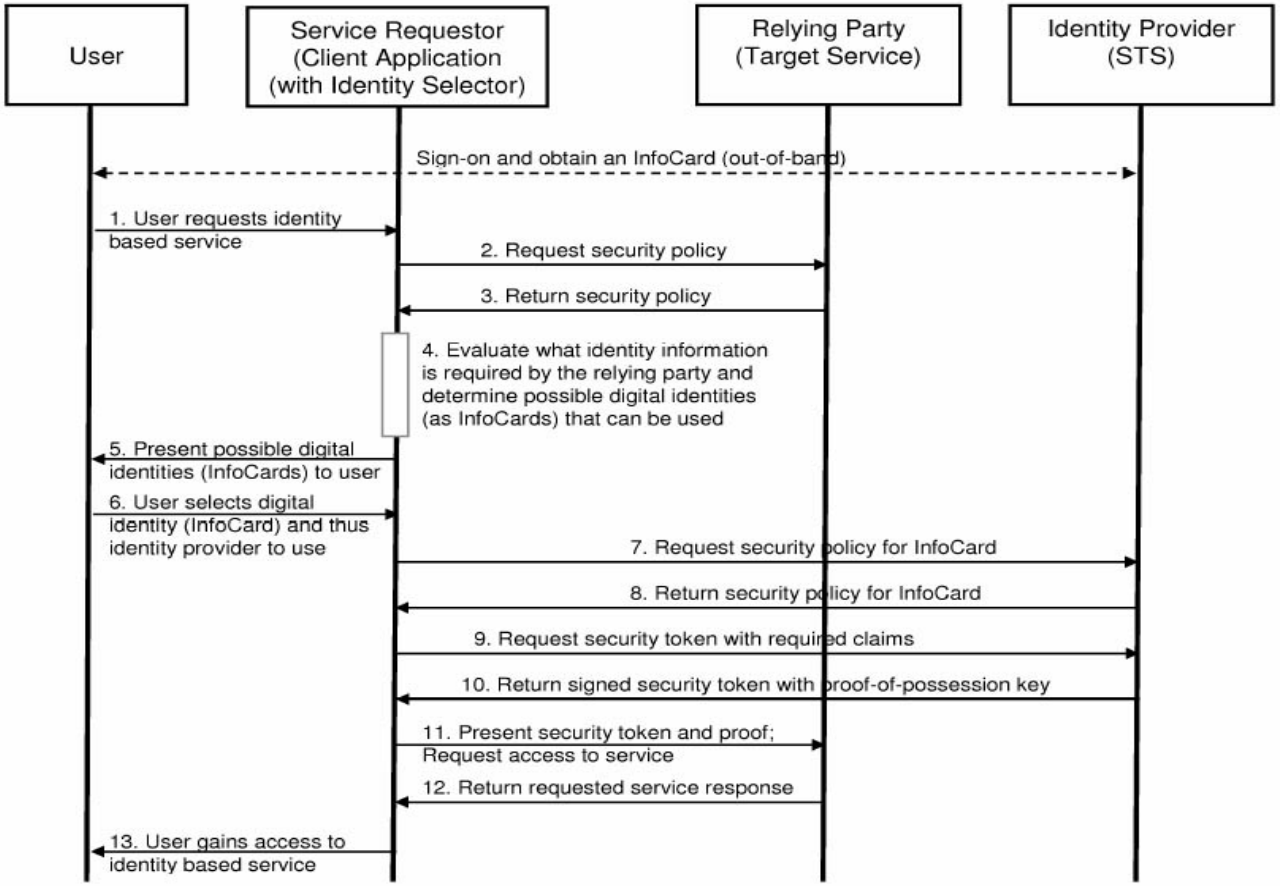
- InfoCard Example

```
<InfoCard
xmlns="http://schemas.microsoft.com/ws/2005/05/identity"
xmlns:wsa=”http://schemas.xmlsoap.org/ws/2004/08/addressing”
xmlns:wsp=”http://schemas.xmlsoap.org/ws/2002/12/policy”
xml:lang=”en-us”>
<InfoCardReference>
<CardId>1234abcd</CardId>
</InfoCardReference>
<CardName>Royal Holloway Student Card</CardName>
<CardImage MimeType=”image/gif”> ... </CardImage>
<IssuerName>Royal Holloway</IssuerName>
<TimeIssued>2008-03-04 T00:30:05Z</TimeIssued>
<TokenServiceReference>
<TokenService>
<wsa:EndpointReference>
<wsa:Address>http://www.rhul.ac.uk/sts</wsa:Address>
<wsid:Identity>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>...</ds:X509Certificate>
</ds:X509Data>
>ds:KeyInfo</
>wsid:Identity</
>wsa:EndpointReference</
>UserNamePasswordAuthenticate<
>Username>Waleed</Username<
>UserNamePasswordAuthenticate</
>TokenService</
>TokenServiceReference</

>ic:InfoCardPolicy<
>SupportedTokenTypes<
”/>TokenType URI=”urn:oasis:names:tc:SAML:1.0:assertion<
>SupportedTokenTypes</
>SupportedClaims<
”>SupportedClaim URI=”http://.../ws/2005/05/identity/claims/givenname<
>DisplayTag>First Name</DisplayTag<
>SupportedClaim</
”>SupportedClaim URI=”http://.../ws/2005/05/identity/claims/surname<
>DisplayTag>Last Name</DisplayTag<
>SupportedClaim</
>SupportedClaims</
 />RequireAppliesTo<
>ic:InfoCardPolicy</
>InfoCard</
```

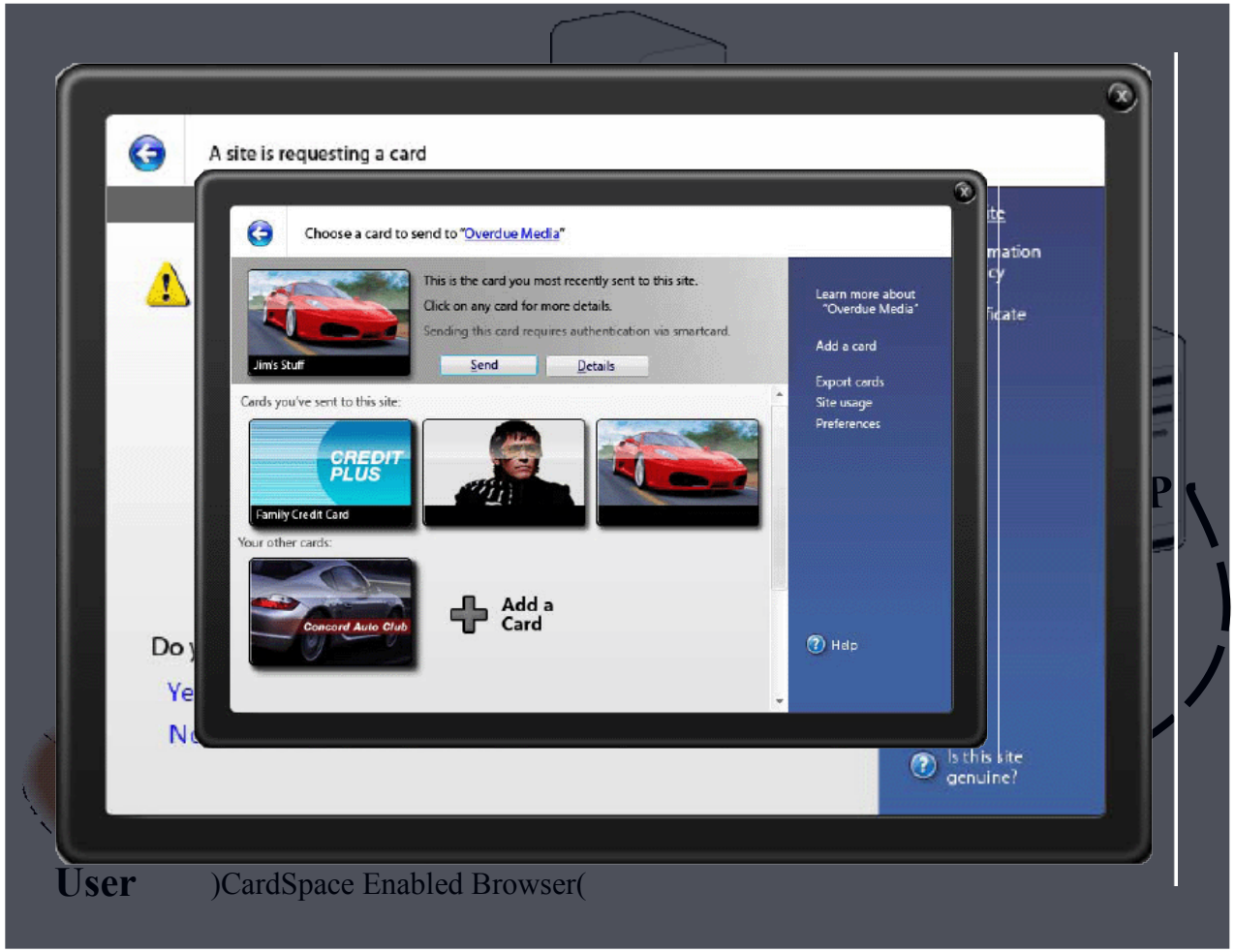
Microsoft CardSpace

- The Message Flow



Microsoft CardSpace

- The Basic Model



User Agent → RP: HTTP GET .1
Login HTML Page Request

: **RP → User Agent**. 2
HTML Login Page + CardSpace Tags
) XHTML or HTML object tags(

3. **User Agent↔ RP-STS**:
User Agent retrieves Policy via WS-SecurityPolicy

4. **User Agent ↔ User**:
User Picks an InfoCard

5. **User Agent ↔ IdP**:
User Authentication

6. **User Agent ↔ IdP-STS**:
User Agents retrieves security token Via WS-MetadataExchange and WS-Trust

:**User Agent → RP-STS**. 7
User Agent presents the security token via WS-Trust

:**RP → User Agent**. 8
!Welcome, your are now logged in

Integrating the two schemes

- Why?

	L.E. **IdP** L.E. **SP**	CS.E. **IdP** CS.E. **RP**	L.E. **IdP** CS.E. **RP**	CS.E. **IdP** L.E. **SP**
Ordinary User Agent	✗	✗	✗	✗
L.E. User Agent	✓	✗	✗	✗
CS.E.User Agent	✗	✓	✗	✗
L.E. & CS.E.User Agent	✓	✓	This can !be changed ✗	This can !be changed ✗

Integrating the two schemes

- How?

- CardSpace architecture consists of two parts:
  1. The user agent supporting components. (ID-WSF?)
  2. The identity framework. (ID-FF LEC)

- Different scopes?

Integrating the two schemes

- The Identity management architecture adaptor
  - A piece of software installed on the user's machine which understands both the Liberty and CardSpace frameworks, and their message flows and formats.
  - The main job is to interpose itself between IdPs and SPs adhering to different identity management architectures, in order to translate particular messages generated by one party to the other.
- Assumptions
  - IdP-IdP integration is out of scope.
  - In case of L.E. IdP & CS.E. RP, we assume that there is a pre-established trust relationship. (Pseudonyms, CardSpace Ref./InfoCard ID)

Integrating the two schemes

- Restrictions
  - Only SAML tokens.
  - No end-to-end encryption. (secure channels)
  - Only *Asymmetric* proof of rightful possession. (*holder-of-key*)
  - In case of CS.E. RP & L.E. IdP, token freshness requests are discarded.

Integrating the two schemes

- How to represent the claims?

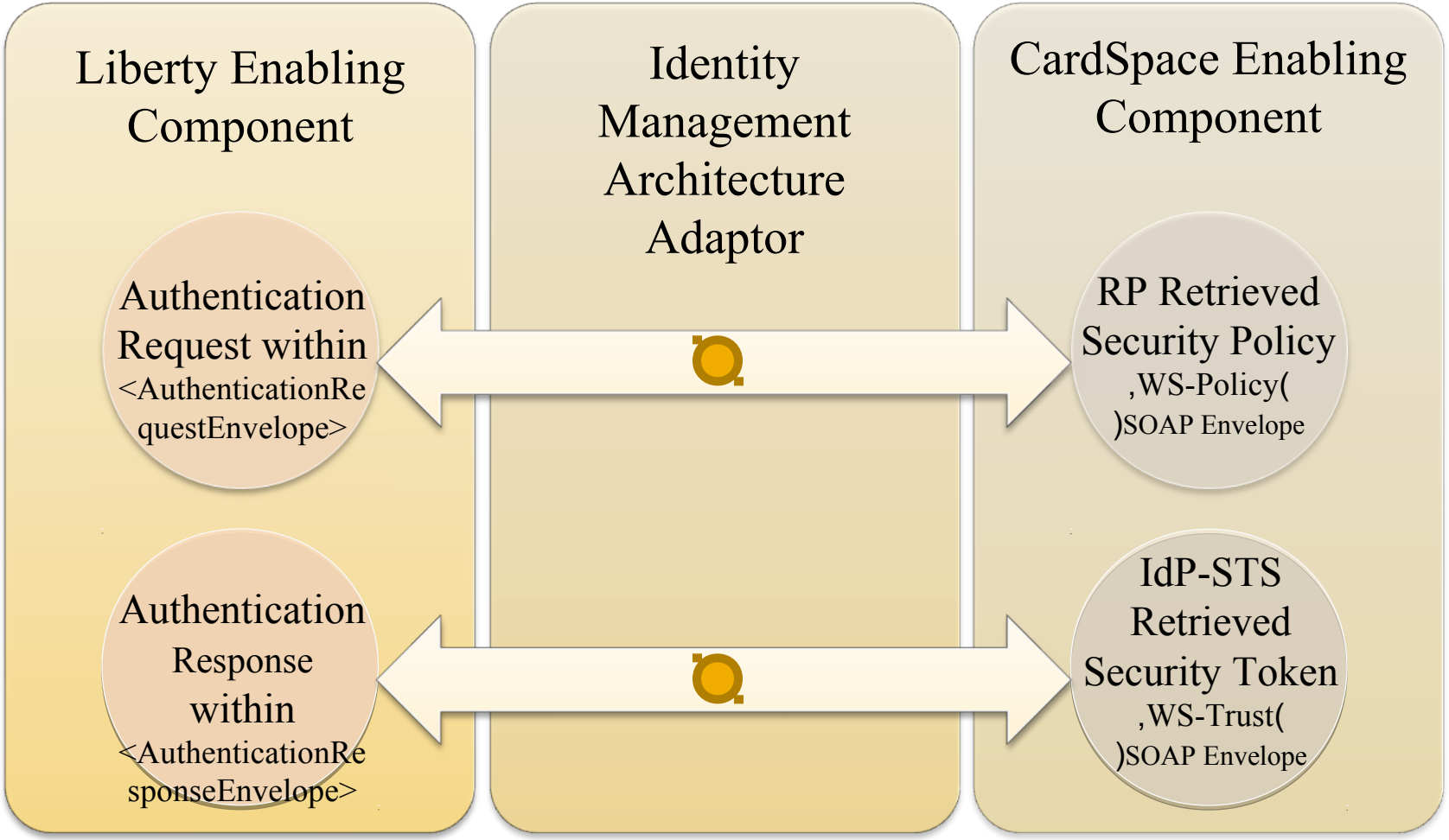
- SAML *Attribute Statement*.

(Requires some modifications to the Liberty enabling component)

- Authentication with no claims.

(severely impact on the usability of the integration)

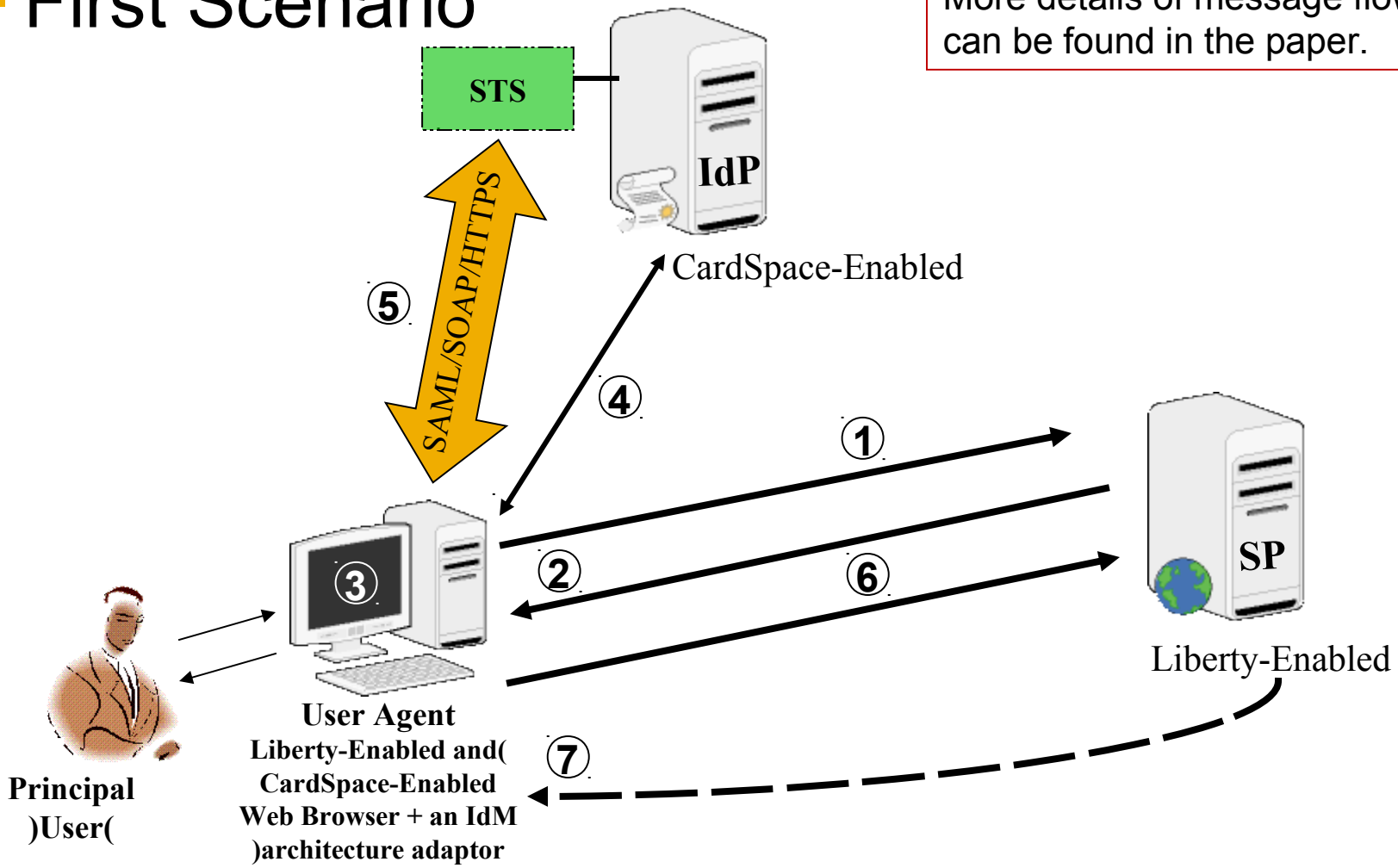
Integrating the two schemes



Integrating the two schemes

- First Scenario

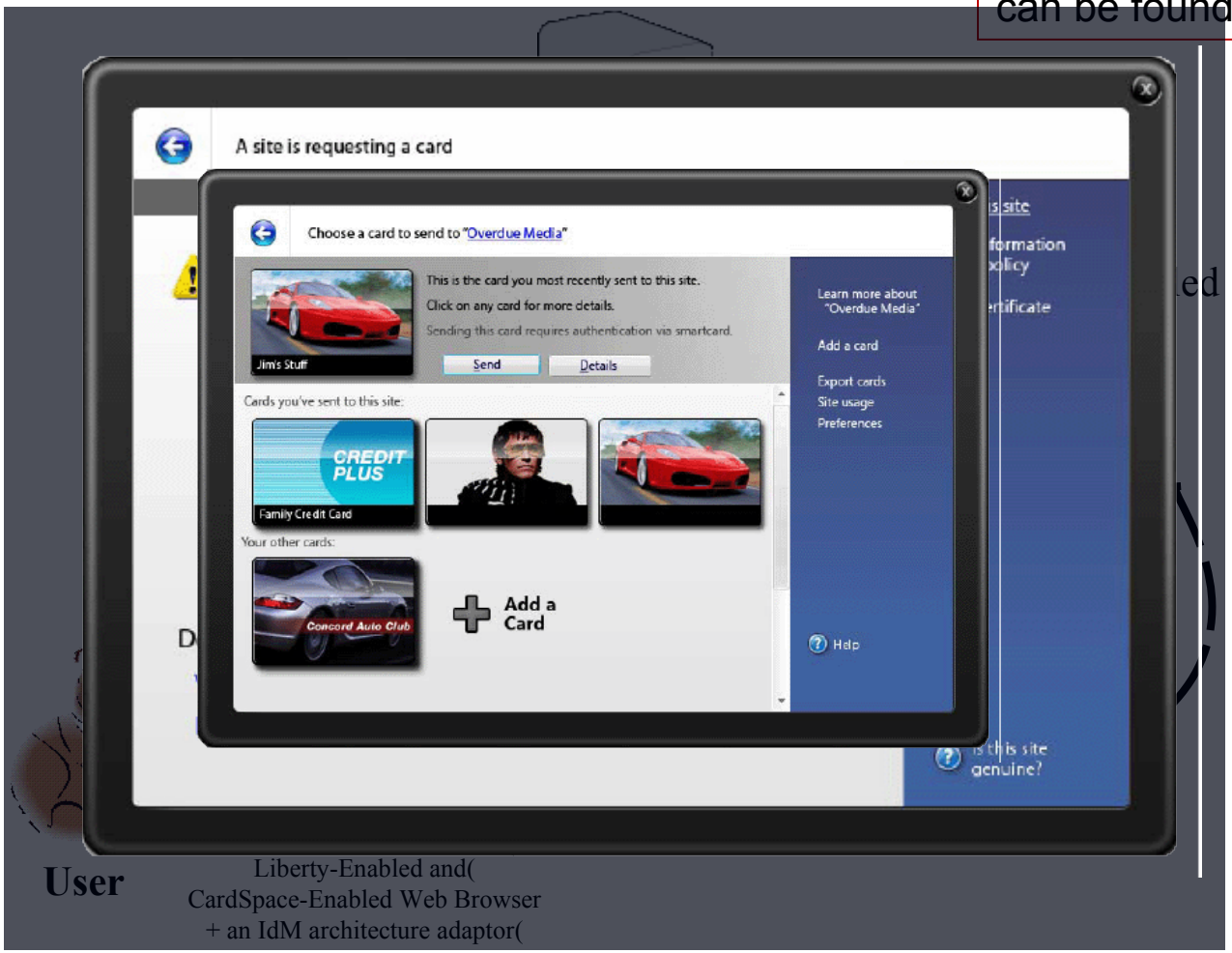
More details of message flow can be found in the paper.



Integrating the two schemes

- Second scenario

More details of message flow can be found in the paper.



Integrating the two schemes

Analysis

- The proposed integration model is designed to be implemented without the need for technical cooperation between Microsoft and Liberty, however, Implementing such a model is non-trivial task.
- CardSpace and the Liberty ID-FF are designed to somewhat different scopes.
- User-agents still need to be CardSpace and Liberty enabled.
- There is no end-to-end encryption.

Integrating the two schemes

Analysis II

- There are restrictions on the token type, encryption and freshness requests.
- Interaction with CardSpace enabling component. (APIs)
- Delay?
- Bandit project.

ThankYou!


Copyright 2003 by Randy Glasbergen.
www.glasbergen.com
GLASBERGEN
"No, nobody stole my identity. I just sort of misplaced it after I got married and had children."

Identity Protection Factor (IPF)

Arshad Noor
StrongAuth, Inc.
550 Lakeside Drive, Suite 10
Sunnyvale CA 94085
arshad.noor@strongauth.com

ABSTRACT

Since the dawn of computing, operating systems and applications have used many schemes to identify and authenticate entities accessing resources within computers. While the technologies and schemes have varied, there appears to have been little attempt to classify them based on their ability to resist attacks from unauthorized entities.

With the proliferation of identity management technologies in the market today, it is becoming increasingly difficult to assess and compare them with each other. As the threat level continues to rise on the internet, and regulations governing information technology continue to grow, risk managers need more objective mechanisms to assign risk to their systems so they may apply appropriate mitigating controls.

This paper attempts to describe a classification scheme that will permit the comparison of seemingly different identification and authentication (I&A) technologies on the basis of their vulnerability to attacks. With a better understanding of related authentication technologies, companies can determine the appropriate technology to use for mitigating authentication risks.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and protection – *authentication.*

General Terms

Management, Security, Standardization.

IDtrust '08, March 4-6, 2008 Gaithersburg, MD

Keywords

Access Control
Asymmetric key
Authentication
Identification & Authentication
Identity Protection Factor (IPF)
Identity Management
Shared-secret
Symmetric key

1. INTRODUCTION

User ID/Passwords, One-Time Password (OTP) tokens, biometrics, smartcards, Network Information Services (NIS) aka Yellow Pages, Kerberos, Secure Socket Layer (SSL), Lightweight Directory Access Protocol (LDAP), Security Assertion Markup Language (SAML), OpenID, CardSpace - these are just a small sample of the dizzying array of identification and authentication (I&A) technologies the computer industry has created to address varying business and security requirements in the Identity Management (IdM) space.

However, except for the rather simplistic "what-you-know, what-you-have and what-you-are" method of classification, there has been little attempt to create a formal classification scheme for I&A technologies on the basis of their resistance to attack. While most technologists have an intuitive understanding of the relative merits of each I&A technology, and security practitioners are more than likely to have a deeper understanding of it, the lack of a formal classification method is indicative of the immaturity of the profession.

This paper attempts to introduce such a classification scheme. It is intended to be a first step within a process that will, hopefully, lead to more research and consequently, a validation of this scheme or the creation of a better one. Besides bringing clarity to this segment of security management, the benefits of such a classification will lead to better risk-management of systems.

1.1 Organization of Paper

This paper begins by describing some of the problems in the area of authentication technologies in Section 2. In Section 3, an overview of the IPF Scale and the technologies that make up the scale is provided. It also explains why this particular model makes sense. In Section 4, the characteristics, strengths and vulnerabilities of each level of the IPF Scale are presented. Section 5 compares this concept with some well-known frameworks and papers on the subject. Section 6 identifies where more research needs to be conducted in this area and the paper finally concludes in Section 7.

1.2 Some Definitions

Before delving into the problems of authentication technologies, it is useful to establish this paper's definition of identification and authentication, so that the proper context is established for discussion.

Within the context of computerized information systems, **Access Control** – the discipline of restricting access to computer resources – is defined to consist of three distinct processes[1]:

i. **Identification** – where a resource claims (or is identified through other means) a specific and unique identifier. Depending on the resource making the claim, the identifier may be a User ID, a Distinguished Name of a digital certificate, a Fully-qualified Domain Name, an Internet Protocol address, etc. *NIST Special Publication 800-63*[2] refers to this resource as "Claimant", which this paper will use from time-to-time.
ii. **Authentication** – where the claimed identifier is verified by the access control mechanism, through some means. Depending on the resource, the verification process may consist of using technologies such as Passwords, Digital Signatures, Reverse DNS lookups, etc. A successful verification deems the resource to be "authenticated".
iii. **Authorization** – where the privileges associated with an authenticated identity are determined. This is the final step of determining whether the claimant may be granted access to another restricted resource.

Similar definitions of Identification and Authentication are presented by the authors of "*Who goes there?: Authentication Through the Lens of Privacy*"[3].

When discussing identity protection, this paper restricts itself only to the Authentication process part of Access Control.

2. PROBLEMS IN AUTHENTICATION

With the introduction of time-shared computers in large corporations and government sectors in the latter part of the 20th century, the use of the User ID/Passwords as the basis of authentication in restricting access to computerized resources, was a very reasonable control mechanism. Given the controlled physical access to computing devices and the closed architectures of computers of the time, it was fairly difficult for attackers to compromise computing resources.

With the explosive growth of the personal computer and Local Area Network (LAN) based computing since the late eighties, and of the internet and the World Wide Web since the mid-nineties, computing resources are now available at even the remotest corners of the planet. Simultaneously, in an attempt to take advantage of the internet and the WWW, companies are racing to transform their business practices - and as a consequence, their computing infrastructure - to deliver personal and business services to their customers over the internet.

As a result, not only are hitherto closed computing systems opening up to the internet, but an unprecedented number of users are now connecting to these computing systems through web-portals from all parts of the globe.

Even though there have been many advances in the field of Identification & Authentication (I&A) technology in the intervening period, most companies - including banks and other firms providing financial services - web-enabling their service delivery have chosen to rely on the ubiquitous User ID/Password as the means of identifying and authenticating users.

This has resulted in a multitude of problems:

i. Consumers of services are forced to register with a multitude of web-sites and acquire new User IDs and Passwords to avail these services. As a result, the average consumer now has more than a dozen credentials, if not more. Most users tend to reuse passwords for multiple credentials, thereby increasing the risk to more valuable accounts;
ii. In order to achieve the quickest and widest adoption of their web-enabled service, businesses use the path of least resistance and require users to only choose a User ID and Password to register for availing the service. While this has the desired effect in the short-term, the long-term problems are just now starting to surface as businesses grapple with the problems of identity-theft;
iii. Knowledgeable , but unethical, computer professionals have recognized opportunities to gain financially by stealing User ID/Passwords and taking advantage of the products and services available over the internet in the name of legitimate users;
iv. The number and types of attacks that attempt to compromise end-user computers and their accounts have grown tremendously over the last ten years – phishing, pharming, key-stroke loggers, root-kits, cross-site

scripting (XSS), SQL injection – these are just a small sample of the types of attacks that have surfaced since the advent of the WWW;

The availability of more sophisticated authentication mechanisms have not made a huge difference towards managing risk in most companies. In an attempt to cut costs and to ease the process of registration, businesses have avoided using the more resilient authentication mechanisms in favor of the User ID/Password.

To make matters worse, in an attempt to simplify the process of authenticating to web-sites, new schemes for authentication are being hatched by the technology industry. The Liberty framework[4], OpenID[5], CardSpace[6] are some examples of "federation" where a single, or a small group of identity-service providers (IdSP) manage the I&A processes, while service web-sites - also called "Relying Parties" in this context - are provided assertions by the IdSP about user-identities. While this has the benefit of off-loading the I&A process to IdSP from the service-provider's point-of-view, unless the underlying authentication technology used by the IdSP was strong, the consolidation of user-credentials at the IdSP could lead to a larger compromise for service-providers.

Finally, in another example of the immaturity of the computing industry, the fact that companies that have suffered a breach are not required to report compromises in a technical manner to some authority, similar to the National Highway Traffic Safety Administration (NHTSA) for automobiles, or the Federal Aviation Administration (FAA) for airlines, makes it impossible for the industry to compile statistical data that would provide insights to risk and mitigation techniques that work. In a telling example of the failure of legislation, the "Breach Disclosure" laws of 37+ US states do not require compromised companies to provide *any technical information* about the compromised infrastructure or the mechanics of the compromise, thus disabling the entire industry from learning from another company's misfortune.

3. IDENTITY PROTECTION FACTOR

Just as the medical industry has coined the term "Sun Protection Factor (SPF)" as a measure of the ability of sunscreen lotions to block the sun's harmful rays from burning human skin[7], this author introduces the term **Identity Protection Factor (IPF) as a measure of the ability of an I&A technology to resist attack from unauthorized entities**.

The IPF uses a numerical scale ranging from zero (0) to ten (10) to indicate the relative effectiveness of the I&A technology to protect credentials, with higher numbers indicating a greater ability to resist attack.

Note: One assumption this paper makes is that the risk of compromise to an I&A technology is based on the client-server architecture using a network for the transport of credentials. This is the typical scenario for the vast majority of systems in use today. However, the IPF can also be used to rate I&A technologies where no network is used for authentication. Examples of the latter are computer operating systems authenticating users against a local database of credentials, or an application authenticating users locally against its own credential database.

A second assumption this paper makes about secret-based authentication is that the shared secret between the human user and the system is not maintained in plaintext on the system.

3.1 IPF Scale

The eleven (11) layers of the IPF Scale are:

IPF	Description
0	No identification or authentication
1	**Shared-secret** based authentication on a local system, or a network **without** any network encryption
2	**Shared-secret** based authentication **with** network encryption
3	**Multiple shared-secret** based authentication **without** an external token, but with network encryption
4	**Asymmetric-key** based authentication with **Private Key in a file**
5	**Multiple shared-secret** based authentication with **external token** and network encryption
6	Asymmetric-key based authentication with **Private Key** generated and stored **on cryptographic hardware token** and using **keyboard** for authentication to token
7	Asymmetric-key based authentication with Private Key generated and stored on cryptographic hardware token and using an **external PIN-pad** for authentication to token
8	Asymmetric-key based authentication with Private Key generated and stored on cryptographic hardware token using an external PIN-pad and **being physically present at the machine** where the resource exists and where authentication is performed

9	Asymmetric-key based authentication with Private Key generated and stored on hardware cryptographic token, using an external PIN-pad, being physically present at the machine where authentication is performed and using **M of N control** for authentication to token
10	Non-existent/Unknown

3.2 A one-dimensional linear scale?

Given the diversity of business, security, operational and user requirements, the IPF Scale begs the question – why a linear scale on a single dimension? Risk-management decisions are a lot more complex than can be plotted on a one-dimensional scale, so how can security professionals and business managers be expected to make a complex decision on authentication technology based on a one-dimensional scale? Chung and Neuman identify multiple dimensions in establishing the *Strength of Security* of network protocols[26]. However, the current paper has chosen to focus on only a single dimension for the reason explained below.

It is the contention of this author, that while it is true that many variables determine the final outcome of computing infrastructures - cost, ease-of-use, operational complexity, availability, ubiquity, etc. - authentication technologies have only one single over-riding factor that truly matters: their ability to resist attacks! If an authentication technology is compromised, nothing else matters to the business at that point. (*Note: At the time of writing this paper, news reports have appeared in the computing press about the $7.9B loss of Societe' Generale to have been caused by poor password management of internal trading systems at this bank[8]).*

An analogy in the field of civil engineering serves as a telling example: while civil engineers do pay attention to factors such as cost, operational complexity and aesthetics, only a single feature truly matters when constructing a bridge – structural integrity of the design and materials used to construct the bridge, and the ability of the bridge to carry its load given the adverse conditions the bridge may be exposed to in its environment. The collapse of the Tacoma Narrows Bridge ("Galloping Gertie") in November 1940[9] is still used as a case-study in the field of Civil Engineering, to teach engineering students the importance of focusing on design. The more recent collapse of the interstate bridge on 35W, in Minneapolis in the state of Minnesota, US in August 2007[10] serves as another example of what truly matters when all is said and done.

It is the position of this paper, that regardless of what factors a company might take into account when determining the authentication technology to use for a computer system, the only factor that truly matters, at the end of the day, is the authentication technology's ability to resist attacks. If an authentication technology with a given IPF rating is not matched appropriately with the risk that a business wishes to assume in a given computer system, it is only the randomness of attacks that prevents the computer system from being certainly compromised.

4. CHARACTERISTICS OF THE IPF

Following are the technological characteristics of each layer of the IPF Scale, with examples of real-world technologies, and potential attacks against them.

4.1 IPF 0

There are business situations where a computing device does not require human users to identify and authenticate themselves to avail services from the device. Kiosks in public facilities, such as airports and museums, are common examples of these business situations.

While the application providing services to the public does operate with the privileges of a User ID known to the underlying operating system, for the purposes of our classification system, we assume this application has an IPF of Zero (IPF 0) because it does not prompt the human user - within the context of the application - to identify and/or authenticate themselves. Such applications are typically built without any credential databases for authentication.

Note: However, within the context of the operating system in which this application executes, I&A technology with a higher IPF rating is assumed to be in force.

I&A technologies with a rating of IPF 0 are assumed to provide no credential risk-mitigation benefits.

4.2 IPF 1

I&A technologies with an IPF of One (IPF 1) are defined as those using a shared-secret based authentication mechanism *without* any network encryption, such as SSL, TLS, IPSec or message-level security, to protect the credential.

IPF 1 technologies are different from IPF 0 in that they add an authentication mechanism to IPF 0 technologies to increase their degree of resistance to attacks. All other characteristics of the application and/or system remain the same as IPF 0.

The primary example of an I&A technology with IPF 1 is the ubiquitous User ID and Password being transported to the server over a protocol such as the Hyper-text Transfer Protocol (HTTP), or a User ID and Password being authen-

ticated against a local file or database on the machine where the credential is presented.

I&A technologies with a rating of IPF 1 are capable of being compromised by any of the following forms of attacks:

- Dictionary attacks against the password file;
- Attacks on the password using Rainbow tables;
- Snooping of network traffic for credentials;
- Keystroke loggers to capture the credentials;
- Phishing attacks that prompt the legitimate user to provide their credentials to the attacker;

Given the nature of IPF 1 I&A technologies – the use of a shared secret to authenticate the user - an attacker can compromise a credential without the knowledge of the credential-owner or server, and usurp the identity of the legitimate user. So, a compromise to IPF 1 I&A technologies can go undetected for long durations after the compromise itself has occurred. This paper identifies this characteristic feature of I&A technologies as being "**compromise-blind**".

4.3 IPF 2

I&A technologies with an IPF of Two (IPF 2) are defined as those using a shared-secret based authentication mechanism with some form of network encryption, such as SSL, TLS, IPSec or message-level security, to protect the credential.

IPF 2 technologies are different from the prior level in that they add network and/or message-level security to increase their degree of resistance to attacks. All other characteristics of the application and/or system remain the same as IPF 1.

I&A technologies with a rating of IPF 2 are identical to I&A technologies with ratings of IPF 1, with one exception: they are not susceptible to compromise through snooping of network traffic, thus giving them a higher IPF rating. Otherwise, all other characteristics and vulnerabilities remain the same.

Other examples of I&A technologies with a rating of IPF 2 would be those based on biometrics alone. On the surface, while biometric I&A technologies appear to use an external authenticator – fingerprint, iris scan, etc. - the reading is ultimately translated into a shared-secret - the template. Biometrics-based I&A technologies also have their own unique attacks that render them susceptible to compromise[11], [12], [13].

IPF 2 I&A technologies, like IPF 1 technologies, are compromise-blind.

4.4 IPF 3

I&A technologies with an IPF of Three (IPF 3) are defined as those combining multiple shared-secret based authentication mechanisms. Because multiple shared-secrets are used to authenticate an identity, this allows such I&A schemes to have a higher IPF rating.

IPF 3 technologies are different from the prior level in that they add a second shared-secret credential to increase their degree of resistance to attacks. All other characteristics of the application and/or system remain the same as IPF 2.

Examples of I&A technologies with IPF 3 are those that combine a User ID and Password with:

- Non-electronic One-Time Password (OTP) tokens – such as from a sheet of paper with predesignated OTPs;
- The selection of a predesignated graphic from an array of graphics;
- Predesignated answers to specific questions;
- Biometrics-based technology;

In all cases, the second authentication credential is also a shared secret that must be sent to the authenticator as part of the authentication process.

From a vulnerability point of view, IPF 3 I&A technologies need to be compromised by multiple types of attacks. The User ID/Password part of the authentication set is capable of being compromised by the same attack techniques as IPF 1 or IPF 2 technologies, while the second shared-secret of IPF 3 I&A technologies is susceptible to phishing attacks[14].

IPF 3 I&A technologies – regardless of the number of shared-secrets used to authenticate the identity - are compromise-blind.

4.5 IPF 4

I&A technologies with an IPF of Four (IPF 4) do not use shared secrets for authentication. They use a form of authentication based on asymmetric cryptographic keys – more popularly known as Public Key cryptography.

IPF 4 technologies are drastically different from the prior level in their use of public-key cryptography to increase their degree of resistance to attacks. All other characteristics of the application and/or system remain the same as IPF 3.

An example of an I&A technology with IPF 4 is the X.509 digital certificate using the Client Authentication protocol

from SSL or TLS[15]. Another example is the Secure Shell (SSH) Protocol when using public-keys[16].

Given the nature of public-key cryptography, the network communication between the client, where the credential is presented, and the server where it is authenticated, is encrypted or cryptographically transformed to prevent replay-attacks; this also eliminates attacks from network-snooping. Public-key cryptography also makes it possible to authenticate a user without having to send the Private Key to the authenticator; merely proof of possession of the Private Key is sufficient to authenticate the user.

The defining characteristic of this IPF 4 technology is that the Private Key of the client's asymmetric key-pair is stored in a file. This file is on the client machine's file-system or an external drive accessible as part of the client machine's file-system with standard ownership privileges.

While the Client Authentication protocol of SSL/TLS and the Public Key Authentication protocol of SSH is robust, the primary vulnerability in this technology is that the file containing the Private Key – typically called a **Cryptographic Keystore** - can be compromised by malware through:

- Dictionary attacks that guess the password/PIN protecting the Cryptographic Keystore; and/or
- Keystroke loggers that capture the password/PIN protecting the Cryptographic Keystore;

Once compromised, the attacker can establish a new SSL/TLS session with the server – either from the legitimate user's PC or from the attacker's own PC if they have copied the Cryptographic Keystore file to their machine - while assuming the identity of the legitimate user. Neither the legitimate user, nor the server would know that the credential had been compromised. Thus, this I&A technology using this specific implementation at IPF 4 is compromise-blind.

4.6 IPF 5

I&A technologies with an IPF of Five (IPF 5) are identical to IPF 3 - i.e. authentication is based on multiple shared secrets – with one exception: in addition to the User ID/Password credential, they use an *external* electronic One-Time Password (OTP) token to generate a shared-secret which is valid for a very short duration[17].

While IPF 5 technologies drop back to shared-secrets for authentication, they are different from prior levels in that they add an external hardware token to increase their degree of resistance to attacks. All other characteristics of the application and/or system remain the same as IPF 4.

While public-key cryptographic authentication systems are generally considered to be superior to shared-secret based authentication systems (because no secrets are shared between the client and the server), I&A technologies with a rating of IPF 4 are easier to attack than systems with a rating of IPF 5.

Nonetheless, external OTP tokens are susceptible to a phishing attack where the attacker can setup a website that mimics the legitimate server site, and then prompts the legitimate user for their User ID, Password and the OTP secret. Upon receiving these, the attacker uses these values to immediately authenticate to the legitimate server site while displaying an error message to the legitimate user. Thus, an attacker can compromise IPF 5 I&A technologies without direct access to the token, and without compromising the client or server machines.

However, because the window of opportunity is extremely short, and the legitimate user must succumb to a phishing attack first, the attacker will have a harder time compromising an IPF 5 technology than ones with lower IPF ratings.

IPF 5 I&A technologies are partially compromise-blind. If the external OTP token is stolen or missing, the legitimate user can suspect their credential may get compromised unless they notify appropriate authorities to take corrective action. However, until an Administrator disables that specific OTP credential, the server remains compromise-blind.

4.7 IPF 6

I&A technologies with an IPF of Six (IPF 6) is the first level on the IPF Scale that provides a significant ability to resist attacks against compromise of the credential. Not only does it use X.509 digital certificates with the SSL/TLS protocol (or Public Key with the SSH protocol), but it also uses a cryptographic hardware token – rated at **FIPS 140-2 Level 2** (or above). The token is used to generate and store the asymmetric key-pair on, thus eliminating an attacker's ability to copy the Private Key from the client machine[18].

In this implementation of an IPF 6 technology, there is little scope for an attacker to compromise the credential from a remote location:

- A dictionary attack (or a Rainbow table attack) is not feasible since there is no password database on the server to attack;
- A keystroke logging attack does not serve much purpose, since the the physical hardware token is necessary to complete the Client Authentication protocol in SSL/TLS or the Public Key Authentication protocol in SSH;

- Network traffic is encrypted by the nature of this protocol, so an attacker would not learn anything from snooping the network;
- Even if the attacker managed to steal the physical token, launching an attack on the token to access the Private Key will be useless, since most token implementations lock up the token after a small number – typically 3-5 - of incorrect attempts, thus rendering the token useless until the token is unlocked by a Security Officer of the legitimate user's company;

However, there is a possibility that an attacker – once having gained access to the legitimate user's client machine, and with significant knowledge of the client and server applications, could launch a social-engineering attack with software of his/her creation. The attack software could prompt the legitimate user for the password/PIN to the token; and if the legitimate user typed in his/her password or PIN – assuming this to be a legitimate request from an application on their PC – this would give the attacker access to the token.

For this attack to work, the attacker must have more than the average attacker's knowledge about hardware tokens, the SSL, TLS or SSH protocols, the look & feel of the client application that interacts with the hardware token, and finally, a great deal of knowledge of the server application to be able to manipulate it remotely.

Slightly less complex – but equally compromising - attacks might result in the legitimate user signing objects that he/she did not intend to sign.

IPF 6 I&A technologies are not compromise-blind, since the legitimate user would be aware of the loss of a hardware token (if it is external), or would be prompted for a PIN to the token. However, until an Administrator disables a specific cryptographic hardware token's credential, the server remains compromise-blind.

4.8 IPF 7

In an IPF 6 I&A technology, if the cryptographic hardware token that stored the Private Key were embedded on the motherboard of the client machine as in a Trusted Platform Module (TPM)[19], or when an external cryptographic token is inserted into the client machine through some port, the Private Key to the legitimate user's credential would become accessible to software on the machine.

Should the attacker, using a keystroke-logger, capture the PIN or pass-phrase to the cryptographic hardware token, they would be able to compromise the legitimate user's credential and establish an authenticated session to the server.

I&A technologies with an IPF of Seven (IPF 7) differ from IPF 6 in that they use external PIN pads - or other physical authentication devices - that are hard-wired to the cryptographic hardware token. This allows the legitimate user to authenticate to the token directly without using the keyboard of the client machine for the authentication process, thereby increasing technologies at this level to resist attacks better than technologies at IPF 6.

IPF 7 I&A technologies are not compromise-blind, since an attacker would not only have to have physical access to the cryptographic hardware token, but also manipulate the hard-wired connections between the authentication input device and the cryptographic token. Servers continue to remain compromise-blind until an Administrator disables a specific credential.

4.9 IPF 8

I&A technologies with an IPF of Eight (IPF 8) differ from IPF 7 in that they require the physical presence of the legitimate user at the machine where the protected resource is being accessed. This is the same machine where the user presents his/her credential and where the application performs the authentication.

By having the legitimate user be present at the machine that will perform the authentication, the machine operators can be assured that there has been no physical tampering of the connections between the cryptographic hardware token and the input device presenting the authentication credential. This difference distinguishes IPF 8 technologies from those at IPF 7 and adds to the technology's ability to resist attacks.

However, IPF 8 technologies are susceptible to compromise by a knowledgeable insider, someone who has unfettered access to the server.

IPF 8 I&A technologies are not compromise-blind.

4.10 IPF 9

I&A technologies with an IPF of Nine (IPF 9) go above and beyond IPF 8 by adding the requirement that no single individual may gain access to the server individually, and that a quorum of legitimate users must be present and authenticated to gain access to the server resource.

This is typically implemented as an M of N control, where M is a subset of N, but represents a majority to establish a quorum. M and N are both odd numbers. Examples of an M of N control is when 3 of 5, 4 of 7, etc. legitimate users are required to authenticate to the server to access the resource.

By requiring such a control, implementers reduce the risk of a sole insider-attack on the resource. While it is still possible to compromise the resource through collusion of legitimate insiders, operators of such an infrastructure must manage that risk through adequate process controls.

IPF 9 I&A technologies are not compromise-blind.

4.11 IPF 10

In keeping with typical scales, this author believes that I&A technologies with an IPF of Ten (IPF 10) - implying perfection - do not exist. There is no such thing as perfect security, and consequently there is no perfect identification and authentication technology.

5. COMPARISON WITH OTHER FRAMEWORKS

There are a number of frameworks and concepts that have been created in the field of identity management. The IPF is compared to some of these frameworks/concepts to place the IPF in perspective.

5.1 The Liberty Alliance Framework

In response to an effort by Microsoft to consolidate User ID's and Passwords through a service called Passport, Sun Microsystems and 33 other companies created a consortium called the Liberty Alliance to provide an alternative to Microsoft's Passport technology[20].

The Liberty Alliance framework supports many authentication technologies - User ID/Password, OTP, X509 digital certificates, etc. - and uses Security Assertion Markup Language (SAML) to federate credentials through an identity service provider (IdSP). No matter how many web-service providers federate their identity management services to the IdSP, the end-user must still authenticate to the IdSP before a SAML assertion can be created to send to the web-service provider.

It is at the point of authentication at the IdSP that the IPF rating would come into play. The IdSP could offer different classes of identity management services based on the IPF ratings of the authentication technology used which would allow the web-service providers to manage the risk of their computer systems based on the IPF ratings they choose to accept in the SAML assertions.

So, the Liberty framework and the IPF Scale are complementary, each providing a different benefit to web-service providers and consumers.

5.2 Oracle's Identity Governance Framework

Oracle and 7 other companies, in November 2006, founded the Identity Governance Framework (IGF) to address the governance of identity-related information across enterprise IT systems[21]. The project is now subsumed under the Liberty Alliance Identity Framework[22].

Most applications currently depend on tightly-coupled application programming interfaces (API) to repositories of information that includes the attributes of credential owners. For example, a Human Resource application has a need to lookup various attributes - such as Social Security Number, tax-related information, medical information, benefits information, etc. - of personnel whose credentials are stored in the HR database. The HR application may use one of many APIs – Java Database Connectivity (JDBC), Open Database Connectivity (ODBC), Lightweight Directory Access Protocol (LDAP), Simple Object Access Protocol (SOAP), etc. - to access this information depending on what type of repository holds the attributes. However, once the application is built to access a specific repository, they usually have little flexibility in dealing with the repository schema, or changes in policies with respect to the identity attributes.

The IGF allows the creation of loosely-coupled systems to reference such identity attributes without hard-coding them into applications, using XML-based protocols, Client applications use the *Client Attribute Requirements Markup Language (CARML)*[23] to specify their requirements for identity attributes, while the service providers use the *Attribute Authority Policy Markup Language (AAPML)*[24] to indicate the attributes they serve and the policies under which they serve up the attributes.

Even though the framework does provide a means to create loosely-couple applications, it still requires users to authenticate to some credential verifier before the user can use the application. While the IGF framework is not explicit in its documentation about what forms of authentication are required, given that the framework is now part of the Liberty Alliance, it can be safely assumed that the Liberty-supported authentication technologies will be supported by the IGF. Therefore, web-service providers can choose to define IGF policies, using AAPML, that serve up different levels of attribute data based on the IPF rating of the authentication technology used by the end-user.

So, once again, the IGF and the IPF Scale are complementary, providing two completely different benefits and services to web-service providers and consumers.

5.3 NIST Special Publication 800-63

The National Institute of Standards and Technology (NIST) published Special Publication 800-63[2] that defines four *Levels of Assurance (LoA)* for electronic authentication

with Level 1 being the lowest and Level 4 the highest level of assurance. The NIST publication has some overlap with the IPF Scale.

Where they are similar is that they both focus on authentication technologies and their abilities to resist attacks. However, the NIST publication's authentication LoA becomes confusing because it combines secret sharing schemes with asymmetric key schemes within the same level. It is this paper's contention that due to the nature of asymmetric key-based authentication schemes at IPF-6 or above, they provide significantly better protection of credentials than secret sharing schemes. Thus, this author believes that the IPF Scale provides better clarity with respect to authentication technology than the NIST LoA framework.

Secondly, the NIST LoA framework is not sufficiently granular to distinguish between implementations of a specific authentication technology. As this paper described earlier, even when using an asymmetric key-based authentication technology (which is FIPS 140-2 Level 2/3 approved) it is possible to differentiate the protection factor rating into at least 4 factors - IPF-6 through IPF-9 - based on how the credential owner is authenticated to the cryptographic token. The NIST LoA collapses such differences into a single Level 4. It is this author's contention that the IPF Scale provides better risk-management with lesser ambiguity to implementers.

Where they differ are, the NIST publication factors in registration processes and identity proofing in addition to authentication technologies in determining the LoA, while the IPF focuses only on the authentication technology. The LoA is designed to provide a business-level assurance regarding claimants' identities and is broader in scope than the IPF Scale.

This author concedes that there is value in providing this level of business assurance regarding a claimants' identity, but believes that that a more useful scheme might be to create a quantitative *Identity Proofing Score* that assigns a quantitative value to various degrees of identity-proofing, and then combine it with a more granular IPF rating to arrive at a more granular Level of Assurance.

This author, additionally, believes that an LoA must take more factors into consideration when determining a level; some factors that need to be factored are:

- The operational practices of the verifier's infrastructure; a Relying Party can have little assurance in a Level 3 or 4 credential if the operations of the Registrar or the Verifier have weaknesses that are unknown until after a breach is discovered;
- The security state of the client machine from which the claimant is authenticating to the Verifier. Once again, a Relying Party can have little assurance in a claimants' credential if the machine from which they're using the token has been compromised;

More work needs to be done in this area to determine the optimal way to combine these factors.

5.4 Microsoft's CardSpace

Microsoft introduced an identity meta-system, dubbed CardSpace, to simplify the management of computer-based identities[25]. Using standards such as WS-Security, WS-SecurityPolicy, WS-Trust, WS-MetadataExchange, SOAP, XML and SAML, CardSpace allows an end-user to submit a *Security Token* provided by an Identity Provider (IdP), to a Relying Party (RP) instead of a credential.

When an end-user needs to authenticate to an RP's site to access a secured resource, the user is presented with the option of submitting a security token in addition to other traditional forms of authentication supported by the RP. If the user chooses the option to submit a security token, they are given the choice of selecting a card from their local CardSpace environment on their client PC.

Unless the card they selected was created by a local "self-issued identity provider" on their PC, the user is redirected to make a request to a third-party IdP for the security token. The IdP, after having authenticated the requester, generates a security token - that is digitally signed and encrypted if there is sensitive information embedded in it - which is used by the end-user to submit to the RP. The RP after having verified the token, makes an authorization decision to allow/disallow access to the secured resource by the end-user.

CardSpace, in its first iteration, supports four methods of having end-users authenticate to the third-party IdP. These are:

1) User ID/Password;
2) Kerberos tickets;
3) X.509 digital certificates from either soft (file-based) or hard-tokens;
4) SAML security tokens created by the "self-issued identity provider"

The IPF Scale and CardSpace are complementary. CardSpace has effectively created four levels of authentication to the IdP, which is more coarse-grained than the IPF Scale. While this may be acceptable to many RP's, this author believes that it is not granular enough to manage risk effectively.

While more analysis is required to determine this, it should be possible to define an element in the WS-SecurityPolicy document created by RP's regarding the IPF rating of authentication that is acceptable to the RP. The IdP, upon receiving this security policy from the RP, can authenticate the end-user using an authentication credential with that IPF rating, and then assert if the end-user was successfully authenticated at that IPF rating in the security token generated by the IdP.

5.5 Higgins – Open source identity framework

Higgins is an open source identity framework from the Eclipse project, with a goal towards integrating identities and profile information across multiple sites and applications. Using standards such as WS-Trust, SAML, LDAP, OpenID, etc., it allows end-users to manage their identities and associated attributes using "*i-cards*". On the surface, it appears t be the "open-source" equivalent to CardSpace, but Higgins inter-operates with CardSpace as one of the identity registries.

To a large degree, the mechanics of Higgins are similar to CardSpace. And to the same degree, the IPF Scale is complementary to Higgins too. Just as CardSpace redirects the end-user to an IdP for authentication and to acquire a security-token from a "*security token service*", Higgins also supports a *Token Service* that allows end-users to authenticate to an IdP and generate a security token that can be handed to the RP. Since Higgins supports WS-SecurityPolicy, WS-Trust and WS-Security too, it is conceivable that the same element definitions for CardSpace that define an IPF rating, can be used within the Higgins framework.

5.6 Comparison summary

As can be seen from comparing IPF to various identity frameworks in this section, the IPF quantifies a unique aspect of authentication technology that makes it possible to complement and integrate with almost all identity management frameworks, while adding unique value to the field of risk management.

6. FURTHER RESEARCH

This is a first attempt at assigning a numerical rating to I&A technologies so they may be compared to each other in mitigating risks of compromise. There are many areas that this author believes requires further research:

- Validation of the assumptions of this model. Are there benefits and attacks that have been overlooked?
- Validation of the granularity of this model. Would a model that has more granularity – say from zero (0) to one-hundred (100) serve the community better?
- Identification of probabilities for compromises of I&A technologies with specific IPF ratings, based on historical breach data;
- Establishment of a repository with IPF values of known I&A technologies;
- Investigation about the possibility of creating an international database of breaches, with sufficient technical detail to assist researchers and practitioners on how to improve computer security;
- A methodology for assessing the risk of a given application system, and how implementers of the application may choose an I&A technology with a specific IPF to mitigate the credential risk.

7. CONCLUSION

There is a plethora of identification and authentication (I&A) technologies available to the information technology (IT) community today. With the exception of the "something-you-know, something-you-have and something-you-are" classification scheme, there has been no methodology based on the risk of compromise to credentials, to assist implementers of systems in choosing appropriate I&A technology to address their business risk. The Identity Protection Factor (IPF) rating is an attempt to create such a classification scheme.

Covering the gamut of shared-secret based I&A technologies and asymmetric-key cryptography based solutions that incorporate the use of cryptographic hardware tokens, this paper presents the IPF Scale and ranks known I&A technologies against this scale on the basis of their protection levels.

Using the IPF Scale and IPF ratings of individual I&A technology products, implementers of information systems will have a means to assess the relative strengths of I&A technologies and their ability to resist attacks to the credential.

The paper concludes that there is no perfect I&A technology, and that further research is necessary to validate the assumptions and granularity of this model, to create an IPF repository of products, and most importantly – to determine the probability of a compromise of each IPF layer based on historical breach data. This information will become crucial towards reducing identity-based risks in the future.

REFERENCES

[1] "Information Security for Lawyers and Law Firms" - Sharon Nelson, David Isom, John Simek, 2006, ISBN 1590316630

[2] NIST Special Publication 800-63 – Electronic Authentication Guideline, William Burr, Donna Dodson, Tim Polk, April 2006 - http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf

[3] “Who goes there?: Authentication Through the Lens of Privacy” – Stephen Kent, Lynette Milett, 2003, ISBN-10: 0-309-08896-8

[4] Liberty Alliance - http://projectliberty.org/liberty/specifications__1

[5] OpenID - http://openid.net/

[6] Microsoft CardSpace - http://msdn2.microsoft.com/en-us/library/aa480189.aspx

[7] Sun Protection Factor (SPF) - http://en.wikipedia.org/wiki/Sun-screen#Sun_protection_factor

[8] Poor password Management may have led to bank meltdown – InfoWorld, February 2008 - http://www.infoworld.com/article/08/02/04/Poor-password-management-may-have-led-to-bank-meltdown_1.html

[9] “Galloping Gertie Collapes November 7, 1940” - http://www.wsdot.wa.gov/TNBhistory/Connections/connections3.htm

[10] Interstate 35W Bridge Collapse website - http://www.dot.state.mn.us/i35wbridge/index.html

[11] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino, "Impact of Artificial Gummy Fingers on Fingerprint Systems," Proceedings of SPIE Vol. #4677, Optical Security and Counterfeit Deterrence Techniques IV, 2002.

[12] How to hack biometrics - http://www.theinquirer.net/en/inquirer/news/2005/07/30/how-to-hack-biometrics

[13] Attacks on Biometric Systems: A Case Study in Fingerprints - http://biometrics.cse.msu.edu/Publications/SecureBiometrics/UludagJain_BiometricAttacks_SPIE04.pdf

[14] Phishing attack targets one-time passwords - http://www.theregister.co.uk/2005/10/12/outlaw_phishing/

[15] The TLS Protocol Version 1.0 - http://www.ietf.org/rfc/rfc2246.txt

[16] SSH Authentication Protocol - http://www.ietf.org/rfc/rfc4252.txt

[17] A One-Time Password System - http://tools.ietf.org/html/rfc2289

[18] Security requirements for cryptographic modules - http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[19] Trusted Platform Module FAQ - https://www.trustedcomputinggroup.org/faq/TPMFAQ/

[20] Alliance forms against Microsoft Passport – USA Today, December 2001 - http://www.usatoday.com/tech/news/2001/12/20/anti-passport-alliance.htm

[21] Oracle Identity Governance Framework (IGF) - http://www.oracle.com/technology/tech/standards/idm/igf/index.html

[22] Liberty Alliance Identity Governance - http://www.projectliberty.org/index.php/liberty/strategic_initiatives/identity_governance

[23] Client Attribute Requirements Markup Language (CARML) - http://www.oracle.com/technology/tech/standards/idm/igf/pdf/IGF-CARML-spec-03.pdf

[24] Attribute Authority Policy Markup Language (AAPML) - http://www.oracle.com/technology/tech/standards/idm/igf/pdf/IGF-AAPML-spec-08.pdf

[25] Introducing Microsoft CardSpace - http://msdn2.microsoft.com/en-us/library/aa480189.aspx

[26] Modelling the Relative Strength of Security Protocols, 2nd Workshop on Quality of Protection, Oct 30, 2006, Alexandria, VA, USA - http://www-scf.usc.edu/~hochung/papers/qop18-chungneuman.pdf

Identity Protection Factor (IPF)

Arshad Noor
StrongAuth, Inc.
arshad.noor@strongauth.com

Today's Goals

- What is IPF?
- What is the IPF Table?
- Description of IPF Levels

The Problem?

UserID-Passwords

CardSpace

LDAP

One-time Password Tokens

Smartcards

Biometrics

NIS/NIS+

KERBEROS

OpenID

SAML

Higgins

Liberty

SSL/TLS with Client-Auth

IGF

What is IPF?

"**Identity Protection Factor (IPF) is a measure of the ability of an I&A technology to resist attack from unauthorized entities.**"

Why 1 dimension?

- **Resistance to attack**
- What about
  - Cost?
  - Ease-of-use?
  - Convenience?
  - Deployment issues
  - Integration issues
- Answer: "Galloping Gertie"

IPF Table

IPF	Description
0	No identification or authentication
1	**Shared-secret** based authentication on a local system, or a network **without** any network encryption
2	**Shared-secret** based authentication **with** network encryption
3	**Multiple shared-secret** based authentication **without** an external token, but with network encryption
4	**Asymmetric-key** based authentication with **Private Key in a file**
5	**Multiple shared-secret** based authentication with **external token** and network encryption
6	Asymmetric-key based authentication with **Private Key** generated and stored **on cryptographic hardware token** and using **keyboard** for authentication to token
7	Asymmetric-key based authentication with Private Key generated and stored on cryptographic hardware token and using an **external PIN-pad** for authentication to token
8	Asymmetric-key based authentication with Private Key generated and stored on cryptographic hardware token using an external PIN-pad and **being physically present at the machine** where the resource exists and where authentication is performed
9	Asymmetric-key based authentication with Private Key generated and stored on hardware cryptographic token, using an external PIN-pad, being physically present at the machine where authentication is performed and using **M of N control** for authentication to token
10	Non-existent/Unknown

IPF-0

- NO identification or authentication *
- Example: Self-service Kiosks

* *However, the software is assumed to be executing with the computing environment (and privileges) of a user authenticated with a credential at a higher IPF level.*

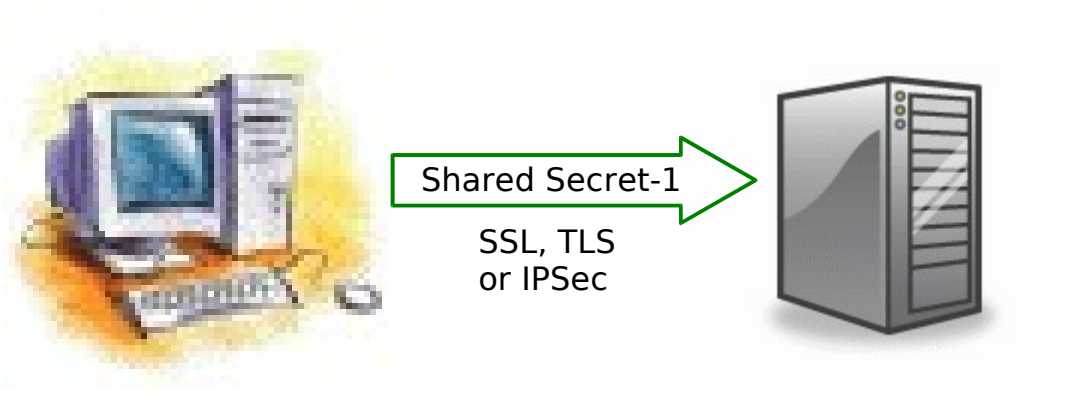
IPF-1

- **Shared-secret** based authentication on a local system, or a network **without** any network encryption

IPF-1 Features

- Adds authentication credential to IPF-0
- Can be compromised by:
  - Dictionary attacks, network snooping, shoulder-surfing, keystroke loggers, phishing, social-engineering, rogue employee
- Compromise-blind
- Example: UserID-Password

- **Shared-secret** based authentication **with** network encryption

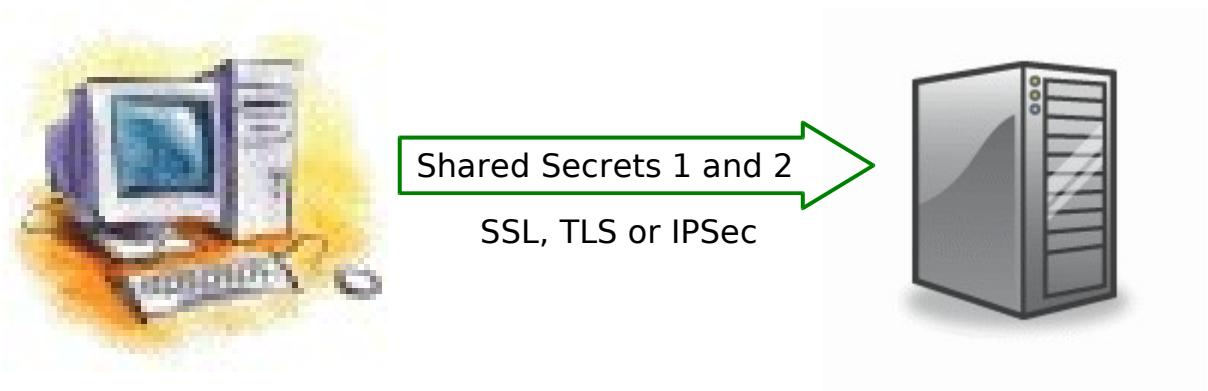


IPF-2 Features

- Adds network encryption – such as SSL, TLS or IPSec - to IPF-1
- Can be compromised by:
  - Dictionary attacks, ~~network snooping~~, shoulder-surfing, keystroke loggers, phishing, social-engineering, rogue employee or technology-specific attacks (for biometrics)
- Compromise-blind
- Examples:
  - UserID-Password
  - Biometrics that convert reading to a template

IPF-3

- **Multiple shared-secrets** based authentication **without** an external token, but **with** network encryption



IPF-3 Features

- Adds another shared-secret to IPF-2
- Can be compromised by multiple attacks:
  - Dictionary attacks, ~~network snooping~~, shoulder-surfing, keystroke loggers, phishing, social-engineering, rogue employee and technology-specific attacks (for biometrics)
- Compromise-blind
- Examples:
  - UserID-Password with biometric
  - UserID-Password with image-selection
  - UserID-Password with answer to question

- **Asymmetric-key** based authentication with **Private Key in a file**

IPF-4 Features

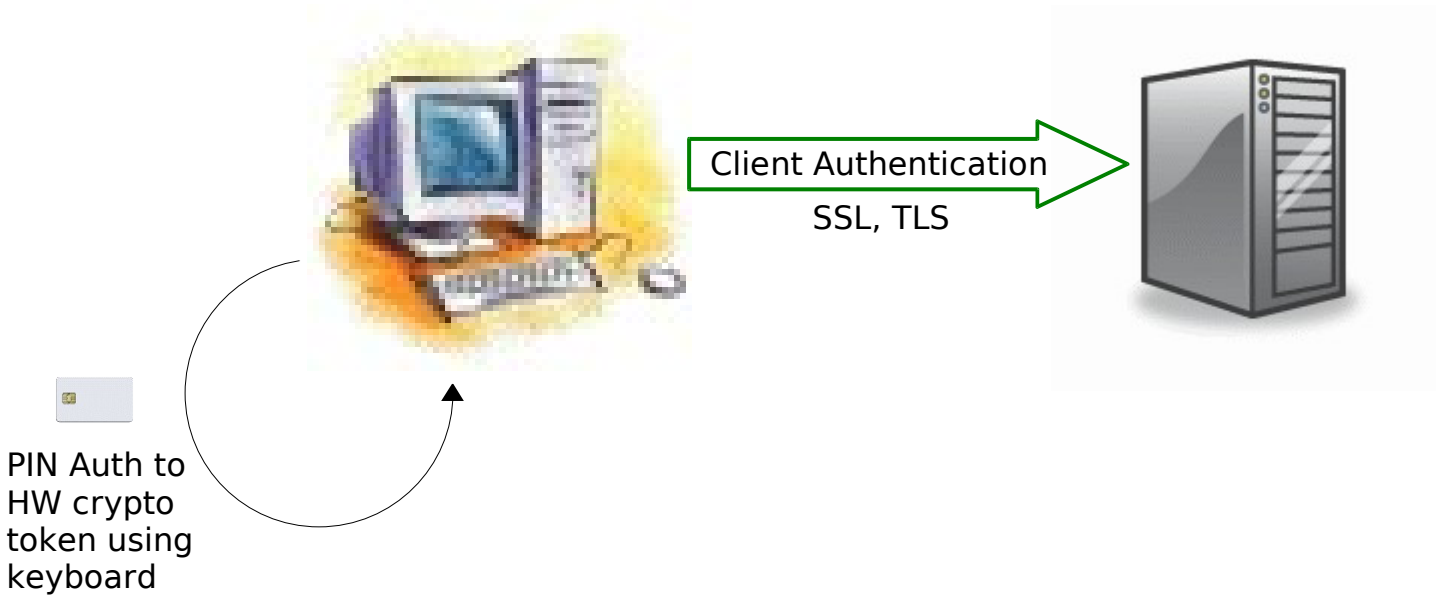
- Uses asymmetric cryptography – does not use a shared secret
- Uses a file-based cryptographic keystore
- Compromised by copying keystore AND:
  - Dictionary attacks, keystroke loggers
- Compromise-blind
- Examples:
  - X509 digital certificate with Private Key in a file
  - Public/Private key-pair with Private Key in a file

- **Multiple shared-secrets** based authentication **with** an external token and **with** network encryption

IPF-5 Features

- Adds external hardware token for second shared-secret to IPF-3
- Can be compromised by multiple attacks:
  - Dictionary attacks, ~~network snooping~~, shoulder-surfing, keystroke loggers, phishing, social-engineering, rogue employee or technology-specific attacks (for biometrics)
- Partially compromise-blind; token loss/theft is immediately detectable
- Examples:
  - OTP token with UserID-Password
  - OTP token with biometric

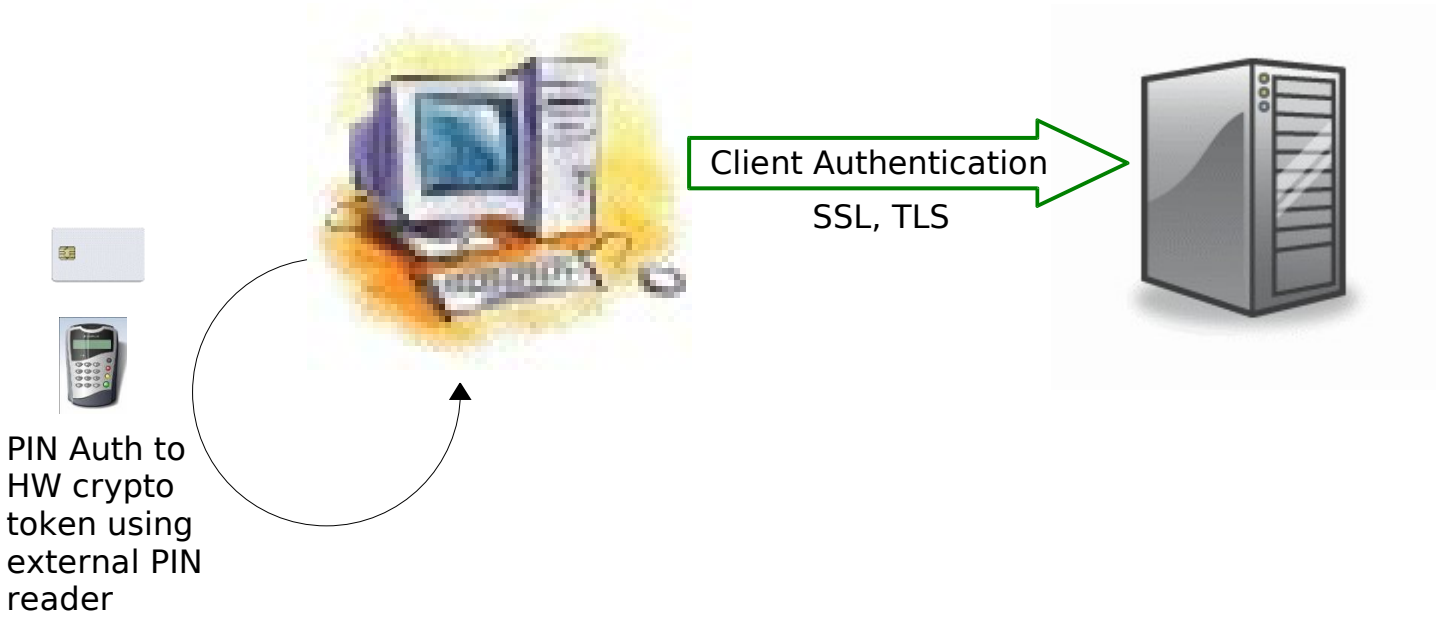
- Asymmetric-key based authentication with **Private Key** generated and stored **on cryptographic hardware token** and using **keyboard** for authentication to token



IPF-6 Features

- Asymmetric cryptography – no shared secret
- Adds external hardware cryptographic keystore to IPF-4
- Compromised by:
  - Social-engineering attack
  - Keystroke-logging AND theft of token
- Client is not compromise-blind; but server can be until client certificate is revoked
- Examples:
  - X509 digital certificate with Private Key on token

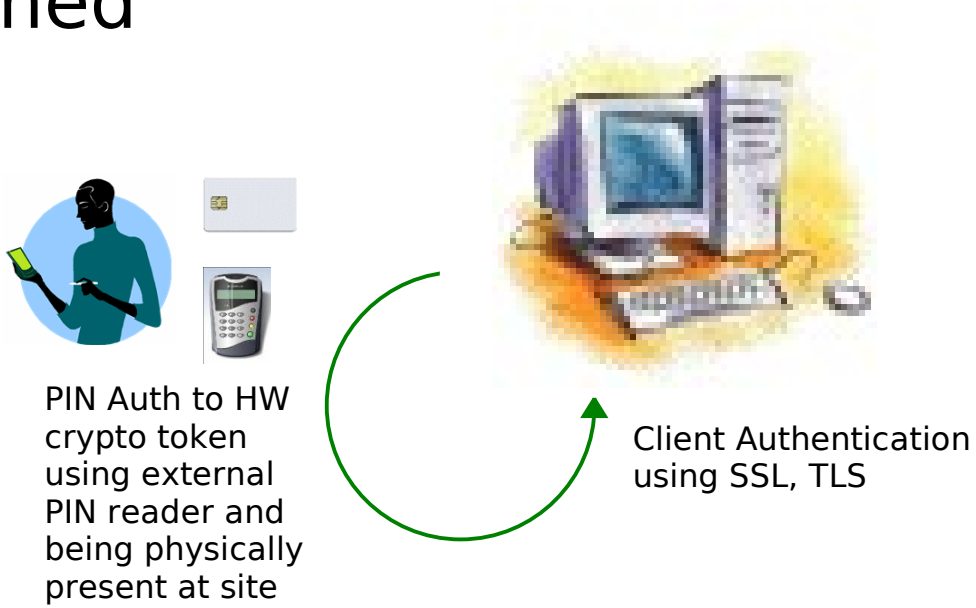
- Asymmetric-key based authentication with Private Key generated and stored on cryptographic hardware token and using an **external PIN-pad** for authentication to token



IPF-7 Features

- Adds external PIN reader to IPF-6
- Compromised by: Social-engineering attack?
- Client is NOT compromise-blind; but server can be until client credential is disabled
- Example:
  - X509 digital certificate with Private Key on token with external PIN reader

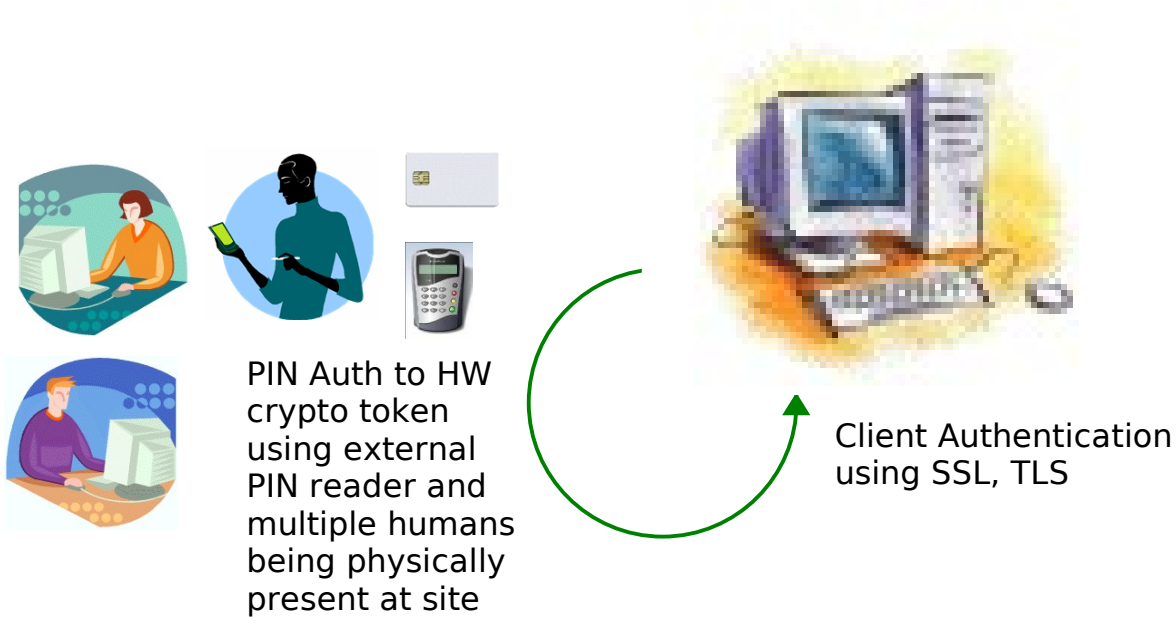
- Asymmetric-key based authentication with Private Key on cryptographic hardware token using an external PIN-pad and **being physically present at the machine** where the resource exists and authentication is performed



IPF-8 Features

- Adds requirement to be physically present at the authentication site, to IPF-7
- Compromised by:
  - Rogue employee
- NOT compromise-blind
- Example:
  - X509 digital certificate with Private Key on token with external PIN reader and requiring physical presence at authentication site

- Asymmetric-key based authentication with Private Key on hardware token, using an external PIN-pad, being physically present at the machine where authentication is performed and using **M of N control** for authentication to token



IPF-9 Features

- Adds requirement of multiple humans to be physically present at the authentication site, to IPF-8
- Compromised by:
  - Collusion of rogue employees
- NOT compromise-blind
- Example:
  - X509 digital certificate with Private Key on token with external PIN reader and requiring physical presence of multiple humans at authentication site

IPF-10

- Does not exist; there is no perfect authentication mechanism

Other frameworks

- No conflict with Liberty (including Identity Governance Framework), CardSpace or Higgins: they all require authentication using some credential at some point which can have an IPF rating
- Some overlap with NIST 800-63
  - Has a broader focus – Level of Assurance - which must look at process controls
  - Mixes technology and process controls – might be useful to define an *Identity Proofing Score* and combine it with IPF to create a compound value

Next steps

- Validation of model
  - Are these levels sufficient? Or do we need more granularity?
- Probabilities of compromises of I&A technologies with specific IPF ratings based on historical breach data
  - Database of (anonymized) breaches with sufficient technical data to assist researchers
- Repository of IPF ratings for technologies
- Model for risk-assessment model and use of specific IPF technologies to manage risk

Conclusion

- Questions?
- Contact Information
  - www.strongauth.com
  - www.strongkey.org
  - info@strongauth.com
  - (408) 331-2000

OpenID Identity Discovery with XRI and XRDS

Drummond Reed
Cordance Corp.
3020 Issaquah-Pinelake RDF #74
Sammamish WA 98075
+1.206.618.8530
drummond.reed@cordance.net

Les Chasen
Neustar, Inc.
46000 Center Oak Plaza
Sterling VA 20166
+1.571.434.5474
les.chasen@neustar.biz

William Tan
Neustar, Inc.
46000 Center Oak Plaza
Sterling VA 20166
+1.571.434.5400
william.tan@neustar.biz

ABSTRACT
The work examines the identity discovery problems that needed to be addressed by the OpenID 2.0 protocol in order to enable a user-centric Internet identity layer. The paper illustrates how the OASIS XRI and XRDS specifications were applied to help solve these identity discovery challenges. The work also considers interoperable identity discovery for other Internet identity frameworks such as SAML, Information Cards, and the Higgins Project, and recommends future work.

Categories and Subject Descriptors
C.2.4 [**Computer-Communications Networks**]: Distributed systems – *distributed databases.* D.4.6 [**Operating Systems**]: Security and protection. H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *user-centered design.* H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia – *architectures, navigation, user issues.* K.6.5 [**Management of Computing and Information Systems**]: Security and protection – *authentication, unauthorized access.* K.8.3 [**Management and Maintenance**]: Security and protection.

General Terms
Design, Security, Human Factors, Standardization, Verification.

Keywords
User-centric identity, identity discovery, XRI, Extensible Resource Identifier, identifier, resolution, XRDS, Extensible Resource Descriptor Sequence, OpenID, Yadis, SAML, information card, i-card, Higgins Project.

1. INTRODUCTION
In enterprise identity management frameworks, the context of an identity being asserted is generally known, or can be discovered directly via mechanisms specified in the framework. But when the context is the Internet as a whole, this approach is no longer viable.

IDtrust '08, March 4-6, 2008, Gaithersburg, MD, USA.

Internet identity management frameworks such as SAML [1], Shibboleth [2], Liberty Alliance [3], Information Cards [4], Higgins [5], and OpenID [6] all must deal with the "identity discovery problem".

Of these frameworks, the one most distinguished by how it handles discovery is OpenID. Due to its origin as a means of combating blog spam, the premise of OpenID is that a user may assert their identity by using their own identifier—for example the URL of their blog, home page, social network profile, or any other web resource the user controls. An OpenID relying party (RP) can then discover from that identifier the user's OpenID provider (OP) and initiate OpenID authentication.

This approach is unquestionably "user-centric" because it gives the user complete control over the identifier they use and therefore the set of OPs that an RP may query. (The issue of whether an RP will accept authentication from the user's selected OP selected is out-of-scope for the OpenID protocol.) However even with this very direct approach, there were still identity discovery challenges that needed to be solved in the steps up from OpenID Authentication 1.0 [7] in 2005 to OpenID Authentication 2.0 [8] in 2007. In this paper we explore these challenges and show how the OASIS XRI and XRDS specifications were employed to overcome them. We also look at how XRI and XRDS are being used by other Internet identity management frameworks, and conclude by suggesting future work.

2. IDENTITY DISCOVERY CHALLENGES IN OPENID
The basic OpenID authentication scenario is that a user logs into an RP site by entering their OpenID identifier rather than a conventional RP-specific username. The RP then resolves the OpenID identifier to discovery the user's OP. In OpenID 1.0, the only supported identifier was a URL, and resolution was either directly to the user's OpenID server, or to an HTML page containing a meta tag with the URL of the OpenID server.

Early OpenID usage raised the following challenges.

2.1 Service Description
As soon as OpenID began to evolve into V1.1 [9], there arose the need not just to discover the user's OpenID service endpoint, but to describe its capabilities (if nothing more than whether it supported OpenID V1.0, V1.1, or both). In addition, the OpenID protocol was designed to be extensible so it can include transfer other useful identity information, such as attributes or authorizations. For example, the Simple Registration extension

(SREG) [10] was developed to automatically transfer the most frequently-request attributes needed for site registration. Thus OPs needed the ability to describe whether they supported SREG or other OpenID extensions.

In addition, OpenID 1.0 was not the only URL-based authentication protocol; it was actually preceded by Lightweight Identity (LID) [11]. Users who wanted their OpenID identifier to work with either protocol needed a way to describe that it supported both capabilities.

These requirements were all difficult to fulfill with HTML link tags; they called for a more generalized, standardized service description mechanism.

2.2 OpenID Recycling

In conventional username/password identity schemes, if a user account is abandoned, the RP can delete the credential and reassign the username to another user. With OpenID, this option is not available since the RP does not assign the user's identifier—that choice is up to the user.

This introduces the "OpenID recycling problem": if a user loses control of their OpenID identifier (for example, if the domain name registration for their URL expires), a new registrant of that URL can gain access to the same private resources as the previous registrant because the new registrant can point the URL to their own choice of OP.

This problem still exists even if the OpenID identifier is assigned by a service provider who can control reassignment, because service providers with large namespaces cannot afford to permanently lock up names within that namespace.

2.3 Resolution Integrity and Trust

From a security standpoint, the weakest link in the OpenID protocol is the discovery stage. This is not only where an untrustworthy RP will focus a phishing attack, but also where resolution of a standard HTTP URL may be hijacked. The use of an HTTPS URL provides significant (but not complete) protection from these attacks.

However OpenID cannot make HTTPS resolution the default due to implementation and usability challenges—it is often not feasible for an individual to obtain an SSL certificate, and outsourced Web hosting services do not always support HTTPS infrastructure. So although HTTPS is recommended, a user must explicitly have it provisioned, then request to use it at every OpenID RP by typing their fully qualified HTTPS URL (not a shortcut like "username.provider.com").

2.4 Privacy and Non-Correlation

Another widely recognized privacy implication of the OpenID 1.x protocol is that it required users to share the same OpenID identifier (or one of a small set they control) with every site. While in some cases this is desirable for cross-site attribute and reputation management, in many other cases such correlation keys are neither necessary nor desirable. In fact other Internet identity frameworks including Liberty Alliance and Information Cards have gone out of their way to avoid introducing such keys.

2.5 Extensibility

Lastly, OpenID architects and users recognized that if OpenID is to develop into a generalized framework for user-centric Internet identity, it must be extensible to a wide variety services that may be associated with an OpenID identifier. These services should all share a common, interoperable description format, including the ability for services—with the user's permission—to communicate and interact with each other on the user's behalf.

Again, these were not the purposes for which HTML header link tags were designed. Clearly a more robust but still lightweight solution is needed.

3. ADDRESSING THESE CHALLENGES WITH XRI AND XRDS

At the same time OpenID 1.1 was evolving, the XRI Resolution 2.0 specification [12] was under development at the XRI Technical Committee at OASIS. The previous 1.0 version was a generalized identity discovery framework developed for use with XRIs (Extensible Resource Identifiers)—abstract identifiers designed for network-, domain-, and application-independent resource identification. XRIs essentially serve the same function for URIs (and any other form of network address) that domain names serve for IP addresses.

However because XRI resolution was based on HTTP(S) and XML, there was nothing to prevent the resolution protocol from being generalized to work with URLs as well as XRIs. This became a key design goal of XRI Resolution 2.0, and lead to the following features being incorporated into identity discovery for OpenID Authentication 2.0.

3.1 Service Endpoint Discovery with XRDS Documents

XRI infrastructure uses the XRDS (Extensible Resource Descriptor Sequence) format for discovery documents. By contrast with DNS, which describes resources using binary resource record types, XRDS documents are a simple, easily extensible XML format for describing the capabilities of any XRI-, IRI-, or URI-identified resource in a manner that can be consumed by any XML-aware application (or non-XRI aware browsers via a proxy resolver).

Figure 1 is an example of an XRDS document describing the resource identified by an XRI for the user of a telephone number, `xri://(tel:+1-201-555-0123)*home` . (This particular XRI illustrates the ability of XRI syntax to include identifiers from other namespaces, essentially acting as an "XML for identifiers".)

Figure 1. An example XRDS document

```
<XRDS xmlns="xri://$xrds" ref="xri://(tel:+1-201-555-0123)*home">
    <XRD xmlns="xri://$xrd*($v*2.0)" version=”2.0”>
        <Query>*home</Query>
        <Status code="100"/>
        <ServerStatus code="100"/>
        <Expires>2005-05-30T09:30:10Z</Expires>
        <ProviderID>xri://(tel:+1-201-555-0123)</ProviderID>
        <LocalID>*residence</LocalID>
        <EquivID>https://example.com/example/resource/</EquivID>
        <CanonicalID>xri://(tel:+1-201-555-0123)!1234</CanonicalID>
        <CanonicalEquivID>
         xri://=!4a76.c2f7.9033.78bd
        </CanonicalEquivID>
        <Service>
            <ProviderID>
             xri://(tel:+1-201-555-0123)!1234
            </ProviderID>
            <Type>xri://$res*auth*($v*2.0)</Type>
            <MediaType>application/xrds+xml</MediaType>
            <URI priority=”10”>http://resolve.example.com</URI>
            <URI priority=”15”>http://resolve2.example.com</URI>
        </Service>
        <Service>
            <ProviderID>
             xri://(tel:+1-201-555-0123)!1234
            </ProviderID>
            <Type>xri://$res*auth*($v*2.0)</Type>
            <MediaType>application/xrds+xml;https=true</MediaType>
            <URI>https://resolve.example.com</URI>
        </Service>
        <Service>
            <Type>http://openid.net/signon/1.0</Type>
            <URI>http://example.com/openid/</URI>
            <LocalID>https://example.com/example/resource/</LocalID>
        </Service>
        <Service>
            <Type match="null" />
            <Path select="true">/media/pictures</Path>
            <MediaType select="true">image/jpeg</MediaType>
            <URI append="path" >http://pictures.example.com</URI>
        </Service>
    </XRD>
</XRDS>
```

By requesting an XRDS document (MIME type `application/xrds+xml`) when resolving an OpenID identifier, an OpenID RP can easily determine the OpenID service endpoints associated with a user's identifier. Each service endpoint, described by the `<xrd:Service>` element, can be identified using one or more `<xrd:Type>` elements. This element accepts a URI, IRI, or XRI to identify the service type. This makes the XRDS format extensible by any specification or service provider without the need for a central type registry.

In addition to advertising the service types it supports, each service endpoint can include a set of URIs representing concrete network endpoints at which this service is available. Redundant network endpoints can be expressed by using more than one `<xrd:URI>` element. Priority among multiple URIs for the same service endpoint (or multiple service endpoints of the same type) is expressed using a priority attribute with the same semantics as in DNS [13]. Elements with equal priority are selected randomly, which can be used to achieve round robin behavior [14].

3.2 Preventing OpenID Recycling with Persistent XRI I-Numbers

The OpenID recycling problem reflects a generic issue in resource identification: the fact that semantic identifiers—the identifiers people find easiest to remember and use—are often the least persistent. The reason is the evolutionary nature of semantics—people constantly change names, addresses, and service providers. This runs directly contrary to identity management policies that depend on a persistent binding between an identifier and the resource it represents (user, device, application, domain, etc.)

This problem is minimized in enterprise contexts because identifier reassignment policies can be tightly enforced. Unfortunately such policies are not feasible at Internet scale. The

domain name secondary market, for example, exists for the very purpose of transferring domain name registrations.

One solution has been to create separate Internet registry and resolution infrastructure for persistent identifiers. The IETF URN (Uniform Resource Name) [15] and Handle [16], [17], [18] specifications were developed for this purpose. However because they lack the human usability of DNS, their adoption has largely been limited to digital artifact registries and DRM systems where identifier non-reassignability is an absolute requirement.

A primary motivation for development of the XRI specifications at OASIS was solving this usability problem by providing a uniform syntax and resolution protocol for *both* reassignable and persistent identifiers. In XRI parlance these known as *i-name*s and *i-number*s. XRI resolution makes it very efficient for each reassignable i-name to have a synonymous persistent i-number that is discovered in the same resolution call. For example, in Figure 1, the local i-name **`*home`**, shown in the `<xrd:Query>` element is synonymous with the i-number **`xri://=!4a76.c2f7.9033.78bd`**, shown in the `<xrd:CanonicalEquivID>` element.

This architecture enables XRI registry infrastructure such as that implemented by XDI.org [19] to enforce policies requiring each i-name registration to have a synonymous i-number, and for i-numbers to never be reassigned, as in URN or Handle registries.

However, synonym assertions must be verified before they can be trusted. XRI Resolution 2.0 specifies two automated verification methods: a) confirming that the synonyms were assigned by the same XRI authority, or b) if they were assigned by different authorities, confirming that they are authorized synonyms by checking for the existence of an `<xrd:EquivID>` reference between the two XRDS documents.

The result is a deep structural solution to the OpenID recycling problem. Although OpenID Authentication 2.0 does not require XRIs, it specifies that when an XRI i-name is used as an OpenID identifier, after discovery the RP must use the synonymous canonical i-number as the user's claimed identifier, and that this synonym must be verified [20]. Since this i-number will never be reassigned, both the registrant and the RP are protected from future reassignment of the i-name. This also enables XRI registries to safely reassign i-names to new registrants by pairing them with a new persistent i-number.

It should be noted that for backwards compatibility, the OpenID Authentication 2.0 specification also supports the ability for an OpenID service provider to add a fragment to a URL in order to distinguish the current user of that URL from a previous or future user [21]. However this solution to OpenID recycling works only for service providers who strictly control their entire URL namespace. For other URLs, transfer of the domain will also transfer control of URL fragments. Furthermore, reassignment of the base URL to a new registrant terminates the ability of the previous registrant to assert that identity because a fragment cannot be resolved. XRI i-numbers do not have this limitation; they can continue to be used indefinitely without regard to the reassignment of any i-names with which they have previously been associated.

3.3 Automatic Trusted Resolution with XRI I-Names

XRI architecture includes two options for secure resolution. The first is to require HTTPS for each request in the resolution chain. For example, the i-name **@cordance*drummond** requires two XRDS document requests: 1) query the @ registry for ***cordance** (* is assumed after the @), 2) query the **@cordance** registry for ***drummond**. In HTTPS secure resolution, each resolution query must use an HTTPS service endpoint, or else resolution fails.

The second option is for each XRDS document in the resolution chain to include a signed SAML assertion that resolvers can verify via public key information discovered in the previous XRDS. The two options are not exclusive; indeed the specification recommends that SAML trusted resolution be used in conjunction with HTTPS trusted resolution to ensure confidentiality.

RPs using OpenID Authentication 2.0 can advantage of this capability by automatically resolving all XRIs using at least the HTTPS trusted resolution protocol. It is relatively easy for XRI authorities to comply with this requirement because XRI is an abstraction layer for URIs; thus an XRI resolution service endpoint only needs to be provisioned with one SSL certificate, no matter how many XRI identifiers it hosts. XDI.org, for example, mandates HTTPS support for the XRI global registry and resolution infrastructure it oversees [22].

The result is that unlike URLs, all XRI i-names used as OpenID identifiers can automatically default to secure resolution, without the need for a user to type a special prefix.

3.4 Anti-Correlation with Pairwise Identifiers

The premise of OpenID 1.x was that users would share one globally unique URL (or one of a presumably small set of URLs) with RPs. This stood in stark contrast to other Internet identity frameworks such as Liberty Alliance ID-WSF and Information Cards, which go to great lengths to use pairwise identifiers so they introduce no new correlation handles at the protocol level.

OpenID Authentication 2.0 addressed this issue by adding support for "directed identity"—a term for the use of pairwise-unique identifiers coined by Microsoft Chief Identity Architect Kim Cameron [23]. This was accomplished by adding the new service endpoint type "http://specs.openid.net/auth/2.0/identifier_select". When a user enters an OpenID identifier resolving to a service endpoint of this type (typically by entering the URL or i-name of their OpenID provider, rather than their own OpenID identifier), the RP knows it must ask the OP for the user's identifier. The OP can then offer the user the choice of using one of their existing OpenID identifiers, or having the OP generate a pairwise-unique identifier for this specific relationship. In fact the user need not know or remember this identifier as the OP can store and automatically use it in future logins to the same RP.

This directed identity feature works with both URLs and XRIs, however by assigning XRI i-numbers in the OP's own XRI delegation space, OPs can take advantage of their persistence and security features discussed above.

3.5 Extensibility to New Services

Much of the market interest in OpenID lies in its larger potential to serve as a framework for many user-centric identity services, all keyed off a user's OpenID identifier(s). To do this, these services need to share a common discovery mechanism that enables different services to interoperate on the user's behalf.

An example is the new OAuth 1.0 protocol, released by the OAuth community in October 2007 [24]. OAuth might be called "OpenID for applications", i.e., it enables a user to delegate to a website or application the ability to access the user's private resources—without the user needing to reveal their actual credentials.

OAuth 1.0 assumes that OAuth providers and consumers are configured manually. However the OAuth community quickly recognized the need for automated discovery of OAuth service endpoint URIs and other configuration metadata. By December 2007 they had published the first draft of OAuth Discovery 1.0 [25]. This specification makes extensive use of XRDS architecture, and specifically the ability for it to be extended by: a) new service type URIs, IRIs, or XRIs from any namespace, b) new XML elements and attributes from other XML namespaces, and c) new trust models based on existing XRDS elements such as `<xrd:ProviderID>` and `<xrd:LocalID>` and/or extension elements.

4. INTEROPERABILITY WITH OTHER INTERNET IDENTITY FRAMEWORKS

Although we have focused on the relevance of XRI and XRDS to OpenID discovery, these technologies are equally applicable to other Internet identity frameworks. In fact they may play a key interoperability role, as discussed in this section.

4.1 SAML

The OASIS SAML specifications include authentication flows very similar to OpenID except for the initial discovery steps [26]. So it is not surprising that they can be adapted to use the same XRDS discovery mechanism as OpenID 2.0. The only difference is the use of a SAML authentication service endpoint. This flow was demonstrated by Pat Patterson of Sun at Internet Identity Workshop in December 2006 [27].

This flow can be further enhanced to provide automated discovery of the SAML metadata [28] necessary to interact with the SAML service provider. By including an XRI as the value of the `<xrd:ProviderID>` element in the SAML authentication service endpoint, an RP can use XRI trusted resolution to resolve this identifier and obtain another XRDS with service endpoint(s) advertising the location of the service provider's SAML metadata documents (which should also be retreived using HTTPS).

4.2 Information Cards

The information card architecture implemented by Microsoft CardSpace, the Higgins Project, and others takes a different approach to identity discovery. First an RP publishes a machine-readable policy description of the claims they require for authentication/authorization to a Web resource. When the user browses that page, their identity selector client reads the policy and presents the user with a choice of the information cards they have (if any) that satisfy it. If the claims are not "self-issued", but come from a third-party identity provider ("managed"), the card itself contains the metadata necessary for the selector to send an authentication token to the provider and obtain a security token bearing the claims, which it then passes to the RP.

In this architecture, an RP does not need to discover a service endpoint for the identity provider directly; all interactions are handled through the selector client. This has clear privacy advantages. However one drawback is that it does not provide the RP with an addressable network endpoint for further discovery or interaction with the user. This has led to proposed "OpenID Information Cards" [29]—standard information cards issued by a user's OP and conveying a security token containing the user's authenticated OpenID identifier. RPs accepting OpenID information cards can then invoke OpenID 2.0 discovery to locate other identity services for the user.

4.3 The Higgins Project

The Higgins Project is a protocol-independent open-source Internet identity framework designed to integrate identity, profile, and relationship information across multiple heterogeneous systems. Started by Parity and including IBM, Novell, Oracle, and Google as contributors, Higgins achieves interoperability via three primary framework elements:

1. *The Higgins Data Model*—a uniform identity data model based on RDF and OWL [30].
2. *Context providers*—Higgins components that implement the Higgins data model to provide a common API for access to any identity data store, from an LDAP directory to an XML document [31].
3. *I-cards*—a consistent user interface metaphor for all identity interactions, regardless of the underlying protocols or token types. I-cards are essentially synonymous with "information cards", but broader in function because they include card types not defined by Microsoft [32].

In the Higgins data model, every identity subject in a context that exposes a Higgins API is addressable with the combination of a *ContextId* and a local *SubjectId*. For interoperability, Higgins required ContextIds to be in a form that enables automated discovery of Higgins context provider configuration metadata, while at the same time supporting the native identifier types that may be used across a very wide variety of Higgins contexts.

The Higgins Project was able to satisfy these requirements with the XRI/XRDS 2.0 framework. First, it lets them express ContextIds as filenames, URLs, or XRIs [33]. Second, it lets them define a small set of Higgins service endpoint types and extension elements for expressing Higgins context provider configuration metadata in XRDS documents [34]. The result is that any Higgins component can resolve the ContextId portion of a Higgins address, discover the Higgins configuration metadata in the XRDS document, and perform automatic configuration.

XRI and XRDS also help enable a new of i-card called a "relationship card" or "r-card". R-cards are intended not just for one-time attribute exchange, but for ongoing, user-permissioned data sharing relationships. To support this functionality, r-card

metadata includes an XRI provisioned by the r-card issuer. An identity selector accepting this r-card can resolve this XRI to discover the service endpoint(s) for the r-card data sharing protocol(s) spoken by both the selector and the RP. The appropriate protocol can then be used to synchronize updates to r-card data, such as a change-of-address for a magazine or mailing list subscription.

Initial r-card implementations use an early version of the XDI (XRI Data Interchange) data sharing protocol under development at the OASIS XDI Technical Committee [35]. Although the final XDI 1.0 specifications are not expected until later this year, XDI is well suited to sharing data in the Higgins Data Model because it too uses an RDF graph model—one in which all nodes are addressable using XRIs. Higgins r-cards correspond to XDI *link contracts*—XDI graphs that express the policies governing usage, synchronization, redistribution, and retention of XDI data. Higgins clients can resolve the XRI in an r-card to an XRDS document to discover an XDI service endpoint and request the associated link contract to set up a data sharing relationship.

5. CONCLUSION AND FUTURE WORK

This paper has shown that performing secure, privacy-protecting identity discovery is a challenge even for a discovery-oriented protocol like OpenID. OpenID 2.0 was able to meet these challenges by taking advantage of the abstract resource identification and discovery features of the OASIS XRI and XRDS framework. Other Internet identity frameworks including SAML, Information Cards, and the Higgins Project have also been able to support new features and address interoperability issues using XRI and XRDS.

However we are still a long ways from a fully generalized and interoperable identity discovery layer for the Internet. For this level of abstraction, XRI and XRDS are at the same stage DNS was twenty years ago, and must mature under usage just as DNS did. Key areas of future work include:

- *Caching and scalability testing.* XRI authority servers and resolvers need the same high-performance XRDS caching as DNS nameservers and resolvers.
- *Proxying.* XRI 2.0 includes basic support for proxy resolvers that offload the work of XRI resolution and XRDS parsing to another web server. Proxy resolvers are attractive both for simplicity and performance reasons, but special attention must be paid to security, privacy, and caching requirements.
- *PKI integration.* While XRI 2.0 includes basic support for signed SAML assertions and a simple mechanism for key discovery, it has the potential to become a much more robust and generalize framework for key distribution and management. The XRI Technical Committee recently joined the OASIS IDtrust Member Section to further explore this area.
- *Reputation.* After basic location and configuration metadata, the type of discovery metadata in greatest demand is reputation. In a context as large as the Internet, where relationships are often dynamic and traditional trust cues and metrics may not be available, reputation is an essential ingredient, as sites like eBay and Slashdot have shown. It is especiallly relevant to an identity discovery layer because that layer must be able to both *support* and *consume* reputation services. This is another area of focus of the OASIS IDtrust Member Section, and may spawn a new Technical Committee in that section by mid-2008.

6. REFERENCES

[1] S. Cantor, J. Kemp, R. Philpott, E. Maler, 2005. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. http://www.oasis-open.org/committees/security.

[2] The Shibboleth Project, 2007. Internet2/MACE. http://shibboleth.internet2.edu/.

[3] Liberty Alliance Project Specifications, 2007. The Liberty Alliance Project. http://www.projectliberty.org/liberty/specifications__1

[4] A. Nanda, 2007. Identity Selector Interoperability Profile V1.0. Microsoft Corporation.

[5] Higgins Project Charter, 2005, Eclipse Foundation. http://www.eclipse.org/higgins/higgins-charter.php.

[6] OpenID Specifications, 2007. OpenID Foundation. http://openid.net/developers/specs/.

[7] B. Fitzpatrick, 2005. OpenID Authentication 1.0. OpenID Foundation. http://openid.net/specs/specs-1.0.bml.

[8] B. Fitzpatrick et al, 2007. OpenID Authentication 2.0. http://openid.net/specs/openid-authentication-2_0.html.

[9] B. Fitzpatrick, D. Recordon, 2006. OpenID Authentication 1.1. OpenID Foundation. http://openid.net/specs/openid-authentication-1_1.html.

[10] J. Hoyt, J. Daugherty, D. Recordon, 2006. OpenID Simple Registration 1.1. OpenID Foundation. http://openid.net/specs/openid-simple-registration-extension-1_0.html.

[11] J. Ernst, 2005. Lightweight Identity (LID). Netmesh Corporation. http://lid.netmesh.org/.

[12] G. Wachob et al, 2007. Extensible Resource Identifier (XRI) Resolution 2.0 Committee Draft 02. OASIS XRI Technical Committee. http://docs.oasis-open.org/xri/2.0/specs/cd02/xri-resolution-V2.0-cd-02.pdf.

[13] Ibid, Section 4.3.3.

[14] Ibid, Section 4.3.3.

[15] R. Moats, 1997. URN Syntax. Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 2141. http://www.ietf.org/rfc/rfc2141.txt.

[16] Sam Sun, Larry Lannom, Brian Boesch, 2003. Handle System Overview. Internet Engineering Task Force (IETF) Request for Comments (RFC) 3650. HDL= http://hdl.handle.net/4263537/4060

[17] Sam Sun, Sean Reilly, Larry Lannom, 2003. Handle System Namespace and Service Definition. Internet Engineering Task Force (IETF) Request for Comments (RFC) 3651. HDL= http://hdl.handle.net/4263537/4068

[18] Sam Sun, Sean Reilly, Larry Lannom, Jason Petrone, 2003. Handle System Protocol (Ver 2.1) Specification. Internet

Engineering Task Force (IETF) Request for Comments (RFC) 3652. HDL= http://hdl.handle.net/4263537/4086

[19] S. Blackmer et al, 2006. XDI.org Global Services Specifications V1.0. XDI.org. http://gss.xdi.org.

[20] Ibid [8]. Section 7.3.2.3.

[21] Ibid. Section 11.5.1.

[22] Ibid [19], Section 5.3

[23] K. Cameron, 2005. The Laws of Identity. Microsoft Corporation.

[24] Atwood, M et al, 2007. OAuth Core 1.0. OAuth.net. http://oauth.net/core/1.0/

[25] Hammer-Lahav, E., 2007. OAuth Discovery 1.0 Draft 1. OAuth.net. http://oauth.googlecode.com/svn/spec/discovery/1.0/drafts/1/spec.html.

[26] Hodges, J., 2007. Technical Comparison: OpenID and SAML, Draft 6. http://identitymeme.org/doc/draft-hodges-saml-openid-compare-06.html

[27] Patterson, P, 2006. YADIS/XRI Identifier Resolution with SAML 2.0. http://blogs.sun.com/superpat/entry/yadis/xri_identifier_resolution_with_saml

[28] Cantor, S. et al, 2005. Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard.

[29] Hardt, D; Bufu, J., 2007. OpenID Information Cards 1.0 – Draft 01. Sxip Identity Corporation. https://openidcards.sxip.com/spec/openid-infocards.html

[30] Higgins Project, 2007. The Higgins Data Model. http://wiki.eclipse.org/Higgins_Data_Model

[31] Higgins Project, 2007. Context Providers. http://wiki.eclipse.org/Context_Provider

[32] Higgins Project, 2007. I-Cards. http://wiki.eclipse.org/I-Card

[33] Higgins Project, 2007. ContextId. http://wiki.eclipse.org/ContextId

[34] Higgins Project, 2007. Context Discovery. http://wiki.eclipse.org/Context_Discovery

[35] Reed, D., Sabadello, M., 2008. The XDI RDF Model. OASIS XRI Technical Committee. http://wiki.oasis-open.org/xdi/XdiRdfModel

OASIS

OpenID Discovery Using XRI and XRDS

IDtrust Symposium, March 4-6, 2008

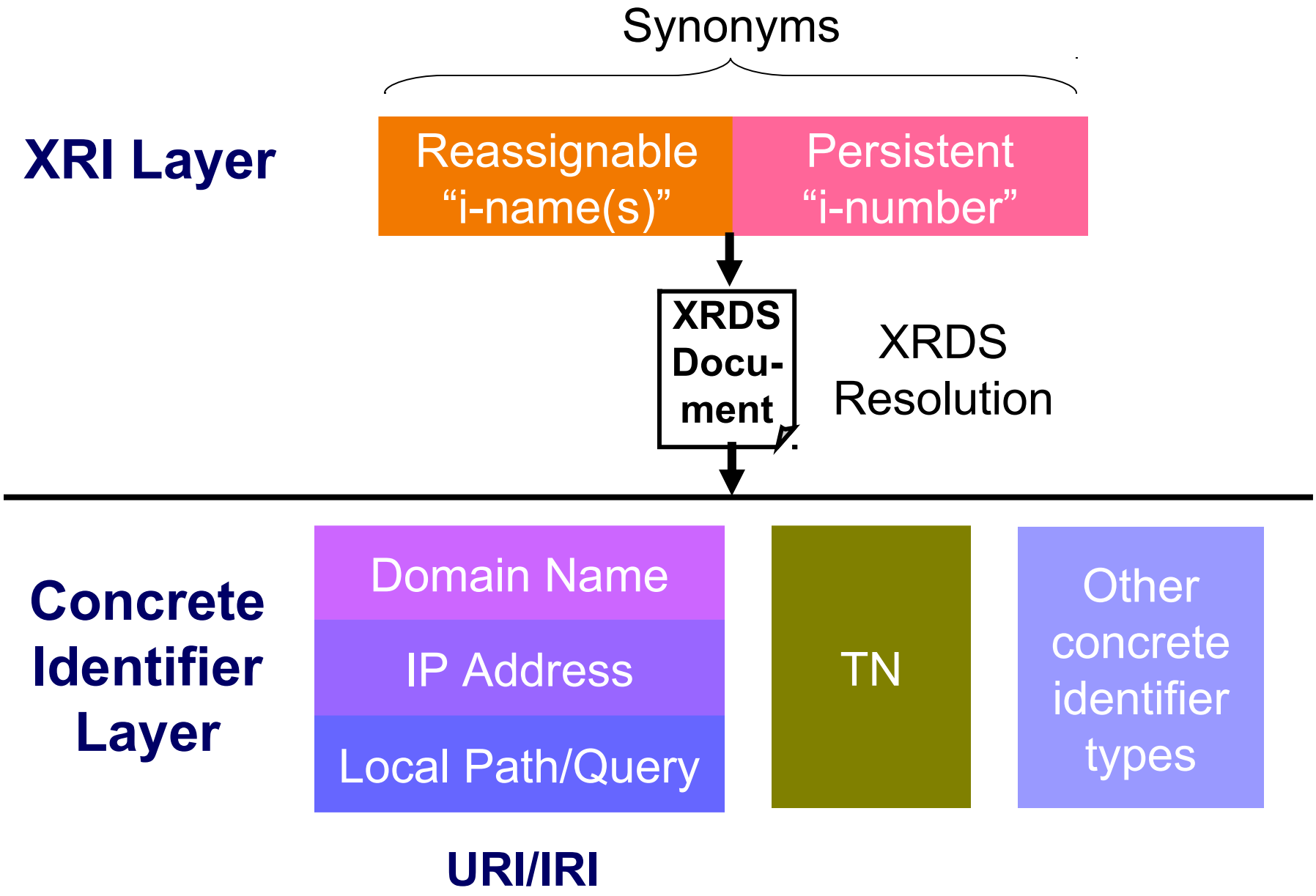
Drummond Reed, Cordance
Les Chasen, NeuStar
William Tan, NeuStar

Overview

- The OASIS XRI and XRDS specifications played a key role in identity discovery for OpenID 2.0
- We'll explain the five key discovery challenges they helped solve
- We'll suggest potential interoperability with other identity protocols/frameworks

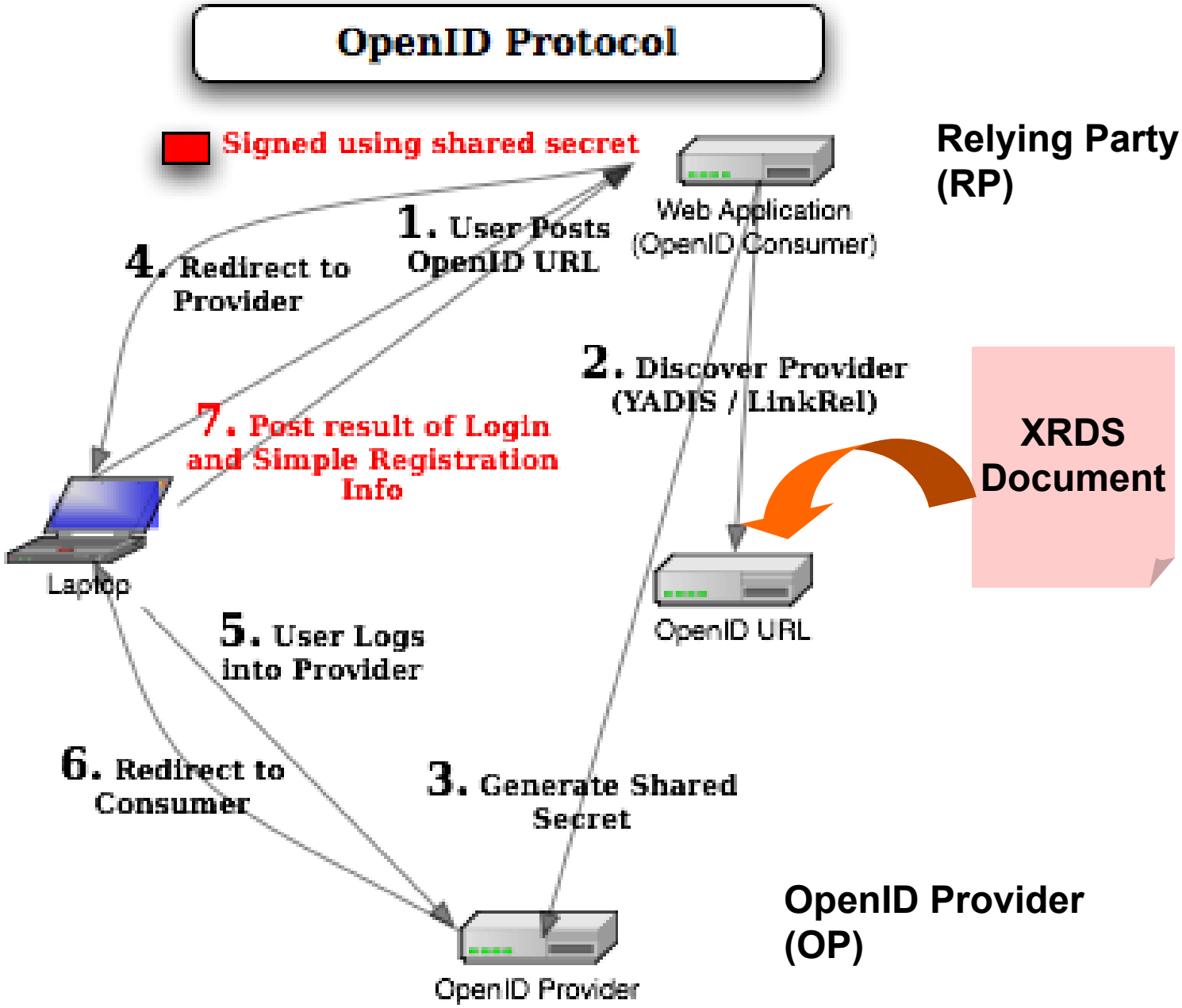
What is XRI (Extensible Resource Identifier)?

- An OASIS Technical Committee
  - Started January 2003
- An open standard language for abstract structured identifiers
  - Identifiers that are independent of domain, application, protocol, or language
  - Identifiers that resolve to other identifiers
- “XML for identifiers”


Synonyms
XRI Layer
Reassignable "i-name(s)"
Persistent "i-number"
XRDS Docu-ment
XRDS Resolution
Concrete Identifier Layer
Domain Name
IP Address
Local Path/Query
TN
Other concrete identifier types
URI/IRI

What is OpenID?

- An open community specification for user-centric Internet authentication
  - Based on the concept that users have their own globally-resolvable identifier and OpenID authentication service
- Prime use case: eliminate the need for separate usernames and passwords for different websites


OpenID Protocol
Signed using shared secret
Web Application
(OpenID Consumer)
Relying Party
(RP)
1. User Posts
OpenID URL
4. Redirect to
Provider
2. Discover Provider
(YADIS / LinkRel)
7. Post result of Login
and Simple Registration
Info
XRDS
Document
Laptop
OpenID URL
5. User Logs
into Provider
6. Redirect to
Consumer
3. Generate Shared
Secret
OpenID Provider
OpenID Provider
(OP)

Evolution from OpenID 1.x to 2.0

- OpenID 1.0 “hardwired” a URL to an OpenID identity server
- This was very rigid and not extensible
- As the OpenID 2.0 tent grew, it needed a more flexible and robust discovery layer

The challenges for OpenID 2.0 identity discovery

- Service description
- OpenID recycling
- Resolution integrity and trust
- Privacy and non-correlation
- Extensibility

Challenge #1: Service description

- Describe what versions of OpenID an OpenID identifier supports
- Enable redundant, prioritized OpenID provider endpoints
- Describe what other authentication protocols may be available (e.g., LID, SAML)

Service description: the solution

- XRDS (Extensible Resource Descriptor Sequence) documents
- The XML analog of DNS resource records
- Very simple set of elements describing
  - Synonyms for an identifier
  - Service endpoints for an identifier
  - Expiration and trust verification metadata

```
<XRDS xmlns="xri://xrds">
  <XRD xmlns="xri://xrd*($v*2.0)">
    <Query>*example</Query>
    <Expires>2005-05-30T09:30:10Z</Expires>
    <ProviderID>xri://=</ProviderID>
    <CanonicalID>xri://=!7c4.58ff.7c9a.e285</CanonicalID>
    <Ref>xri://@!2017.cd67.94c8.023!c83d</Ref>
    <Service>
       <Type>xri://$res*auth*($v*2.0)</Type>
       <URI>http://res.example.com/=!1234.5678.a1b2.c3d4/</URI>
     </Service>
    <Service>
       <Type>http://openid.net/openid/1.1</Type>
       <Type>http://openid.net/openid/2.0</Type>
       <Path>+openid
       <URI>http://authn.example.com/openid/</URI>
    </Service>
    </XRD>
</XRDS>
```

Challenge #2: OpenID recycling

- With usernames/passwords usernames can be recycled
  - The service provider controls the binding with the credential
- With OpenID, that's no longer true
  - The user controls the binding to the credential
  - Losing control of the identifier = losing control of the credential

Challenge #2: OpenID recycling

- Service providers with large name-spaces can't afford to assign names once and lock them up forever
  - Examples: AOL, Yahoo
- DNS names are inherently recyclable – an entire industry exists to serve the secondary domain name market

OpenID recycling: the solution

- Synonyms
  - Support the binding of a recyclable identifier with a non-recyclable synonym
  - Authenticate based on the persistent synonym
  - Treat the recyclable identifier as only a temporary handle for the persistent synonym

OpenID recycling: the solution

- Persistent synonyms is a primary raison d'être for XRI
  - XRI distinguishes between reassign-able "i-names" and persistent "i-numbers" at the syntax level
  - XRDS documents provide automated synonym mapping
  - XRI Resolution 2.0 includes automated synonym authorization verification

```
<XRDS xmlns="xri://xrds">
  <XRD xmlns="xri://xrd*($v*2.0)">
    <Query>*example</Query>
    <Expires>2005-05-30T09:30:10Z</Expires>
    <ProviderID>xri://=</ProviderID>
    <CanonicalID>xri://=!7c4.58ff.7c9a.e285</CanonicalID>
    <Ref>xri://@!2017.cd67.94c8.023!c83d</Ref>
    <Service>
       <Type>xri://$res*auth*($v*2.0)</Type>
       <URI>http://res.example.com/=!1234.5678.a1b2.c3d4/</URI>
     </Service>
    <Service>
       <Type>http://openid.net/openid/1.1</Type>
       <Type>http://openid.net/openid/2.0</Type>
       <Path>+openid
       <URI>http://authn.example.com/openid/</URI>
    </Service>
    </XRD>
</XRDS>
```

Challenge #3: Resolution integrity/trust

- OpenID could not specify HTTPS resolution for all OpenID URLs
  - Too many users do not have access to HTTPS certs or infrastructure
  - Thus the default had to be HTTP
  - This forces users with HTTPS URLs to have to type the entire string, e.g., **https://my.openid.identifier.tld**

Resolution integrity/trust: the solution

- As abstract identifiers, XRIs always map to concrete service endpoints
- XRI resolution offers three trusted modes:
  - HTTPS, SAML, or both
- Thus all XRI i-names can use HTTPS resolution as the default
  - No need for users to know/do anything

Challenge #4: Privacy & non-correlation

- OpenID 1.x assumed users would share the same identifier(s) with every RP
- Violates the Fourth Law of Identity:
  - *A universal identity system must support both "omni-directional" identifiers for use by public entities and "unidirectional" identifiers for use by private entities, thus facilitating discovery while preventing unnecessary release of correlation handles.*

Privacy & non-correlation: the solution

- Directed identity
  - Users can enter the URL or XRI of their identity provider
  - The discovered XRDS doc contains a directed identity service endpoint
  - The RP redirects the user to their OP to select their identifier
  - The OP can also generate a pairwise unique “per relationship” identifier

Privacy & non-correlation: the solution

- Directed identity supports means OpenID 2.0 satisfies the Fourth Law
- It is the only mode some large service providers currently support
  - Yahoo
- Ideally users will have a choice of whether to use a public or directed identifier

Challenge #5: Extensibility

- OpenID is a framework for user-centric identity services
- RPs need to be able to discover what OpenID extension specs an OP supports
  - SREG, AX, PAPE (more coming)
- The discovery format itself needs to be extensible

Extensibility: the solution

- XRDS documents
  - Service types are declared using URIs, IRIs, or XRIs – anyone can extend
  - Multiple types can be declared for the same service endpoint
  - Elements can be added from any XML namespace
  - XRDS documents can redirect or refer to other XRDS documents

Extensibility: the solution

- Example: OAuth
  - “OpenID for services/applications”
  - Allows users to authorize a website or application to access protected resources without providing their credentials directly
  - OAuth Discovery uses XRDS extensibility

```
<XRDS xmlns="xri://$xrds">
  <XRD xmlns:oauth="http://oauth.net/discovery/1.0" xmlns="xri://$xrd*($v*2.0)">
    <Query>http://api.example.com/</Query>
    <Expires>2007-12-31T23:59:59Z</Expires>
    <oauth:RequestParameterMethods>
       <oauth:Method>AUTH-HEADER</oauth:Method>
       <oauth:Method>POST-BODY</oauth:Method>
       <oauth:Method>URL-QUERY</oauth:Method>
    </oauth:RequestParameterMethods>
    <oauth:RequestSignature>
       <oauth:Method>HMAC-SHA1</oauth:Method>
    </oauth:RequestSignature>
    <Service>
       <Type>http://oauth.net/core/1.0/endpoint/request</Type>
       <URI>https://api.example.com/session/request</URI>
       <oauth:HttpMethod>POST</oauth:HttpMethod>
       <oauth:RequestSignature append="head">
       <oauth:Method>PLAINTEXT</oauth:Method>
       </oauth:RequestSignature>
    </Service>
...
```

Interoperability with other identity frameworks

- SAML
- Information Cards
- Higgins

SAML

- OpenID can use SAML!
  - Shown by Pat Patterson at the Internet Identity Workshop in December 2006
  - Same discovery steps, similar protocol flow, just using SAML tokens
  - Can also use XRDS documents for automated discovery of SAML metadata

Information Cards

- Information cards can carry discoverable OpenID identifiers
- XRDS discovery is not used in the information card flow
- But sharing an OpenID claim can enable the RP to do XRDS discovery on other identity services

Higgins

- Higgins needed a solution for cross-domain context discovery
- Higgins resolves a URL or XRI to an XRDS document to discover:
  - The service endpoint URI(s) for the context
  - The Higgins context configuration metadata needed to open the context

```
<XRDS xmlns="xri://$xrds">
  <XRD xmlns="xri://$xrd*($v*2.0)">
    <Query>*mycontext</Query> <Status code="100"/>
    <Expires>2999-01-01T00:00:00.000Z</Expires>
    <ProviderID>xri://@</ProviderID>
    <LocalID priority="10">!12345</LocalID>
    <CanonicalID priority="10">@!12345</CanonicalID>
    <Service priority="10"
xmlns:hconf="http://higgins.eclipse.org/Configuration">
      <Type>$context+jdbc</Type>
      <Type match="default" />
      <URI>jdbc:postgresql://192.168.1.102/mydatabase</URI>
      <hconf:Configuration xmlns="http://higgins.eclipse.org/Configuration"
       xmlns:hconf="http://higgins.eclipse.org/Configuration">
        <SettingHandlers>
          <SettingHandler Type="xsd:string" Class="java.lang.String"
           Handler="org.eclipse.higgins.configuration.xml.StringHandler"/>
        </SettingHandlers>
        <Setting Name="TestContext" Type="htf:map">
          <Setting Name="username" Type="xsd:string">dbuser</Setting>
          <Setting Name="password" Type="xsd:string">dbpass</Setting>
        </Setting>
      </hconf:Configuration>
```

Future work

- Caching and scalability testing
- Proxying
  - Performance optimization
  - Integration with authority servers
- PKI integration
- Reputation discovery

Conclusions

- OpenID may or may not become an Internet-wide authentication standard
- But OpenID identity discovery model has already proved broad utility
- XRDS resolution provides a common discovery format for URLs and XRIs
- It can provide an interoperable foundation for Internet identity layer

Contact us

- Drummond Reed, Co-Chair, XRI TC
  - http://xri.net/=drummond.reed
  - drummond.reed@cordance.net
- Les Chasen, NeuStar, Editor, XRI TC
  - http://xri.net/=les.chasen
  - les.chasen@neustar.biz
- William Tan, NeuStar, Editor, XRI TC
  - http://xri.net/=wil
  - william.tan@neustar.biz

- **Learn** through the IDtrust Knowledgebase of educational materials and background on the standards
- **Share** news, events, presentations, white papers, product listings, opinions, questions, and recommendations through postings, blogs, forums, and directories.
- **Collaborate** with others online through a wiki interface

http://idtrust.xml.org

Identity & Policy
(for Security, Privacy and Trust)

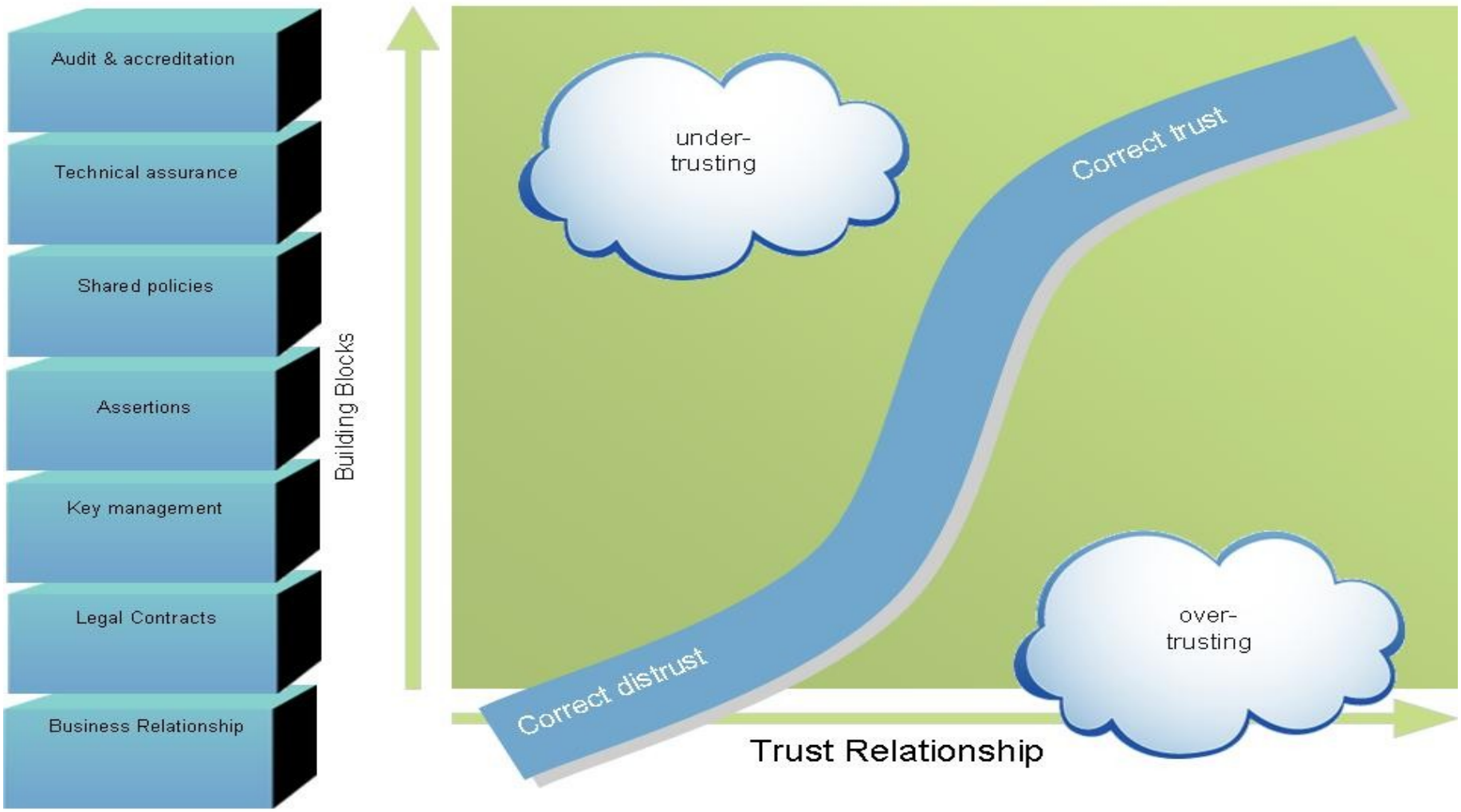
March, 4th 2008
NIST
Identity and Trust Symposium

Rakesh Radhakrishnan
Chief Identity Integration Architect (Telco)
Sun Microsystems, Inc.

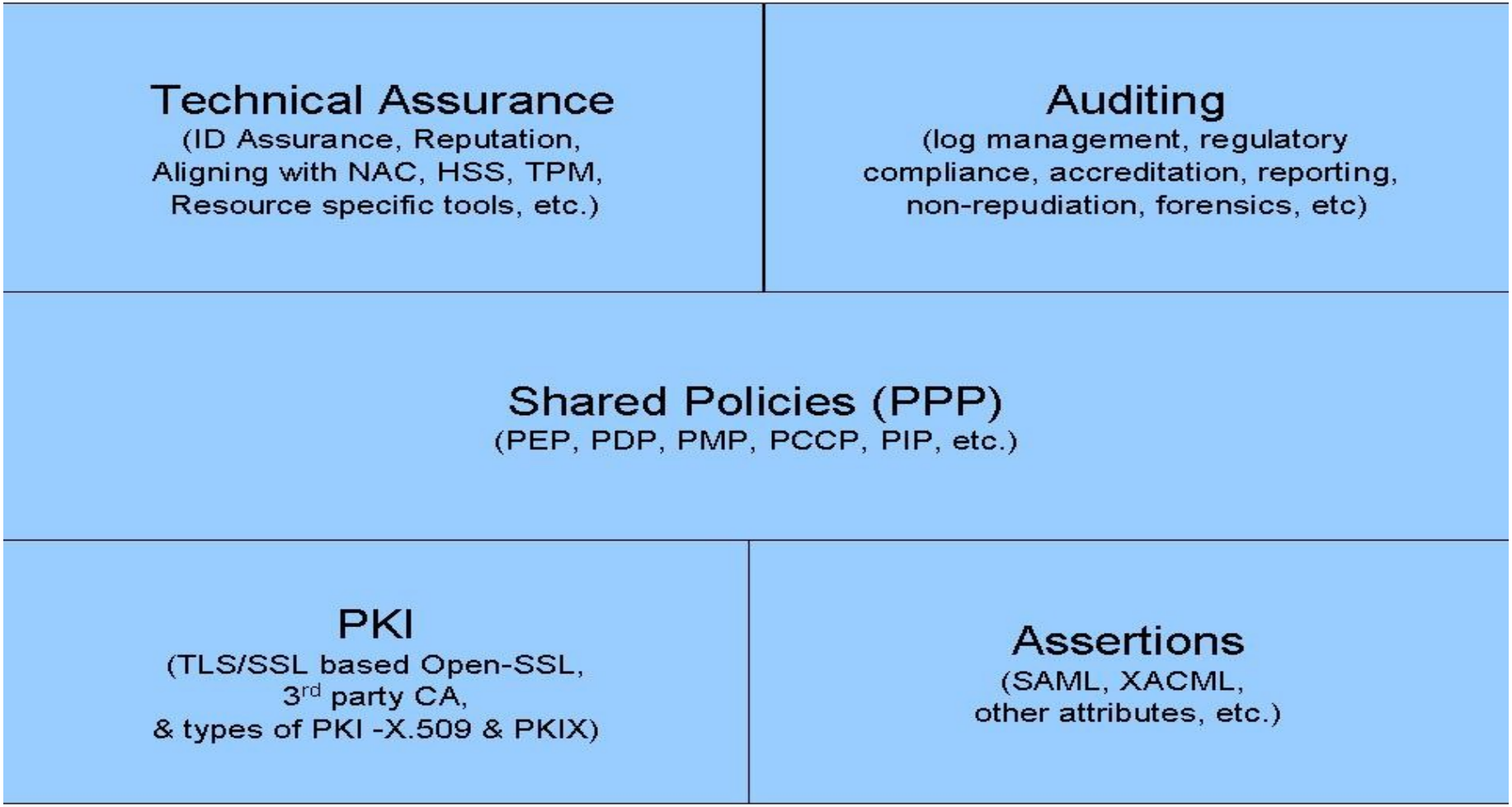
Agenda

- Vertical Integration of Identity Systems (mapping identifiers and aggregating meta-data)
- Policy based Security Service invocation (SaaS)
- Pervasive Policy Paradigm
- FAM Policy System Architecture
- Policy Orchestration (papers + POC/Pilots and Prototypes)

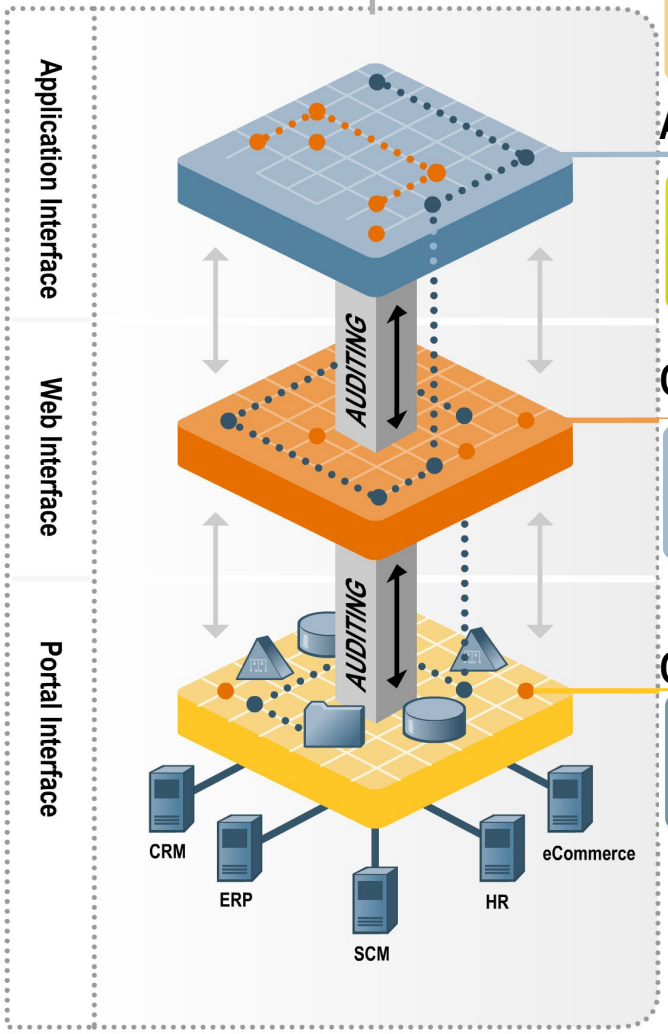
Identity and Trust



Identity and Trust



Vertical Integration (Identity & Security)



User & Device Centric IDS

User ID & Profile
Device ID & Profile
User & Device specific Policies

Access & Sensor Network IDS

AM Agents for Wifi, WiMAX, BPL, Cable head end, xDSL, RFID/EPC & more

Core & Federated Network IDS

FM integration with OAM, NG IN, HSS, HLR, NAC, FW & more

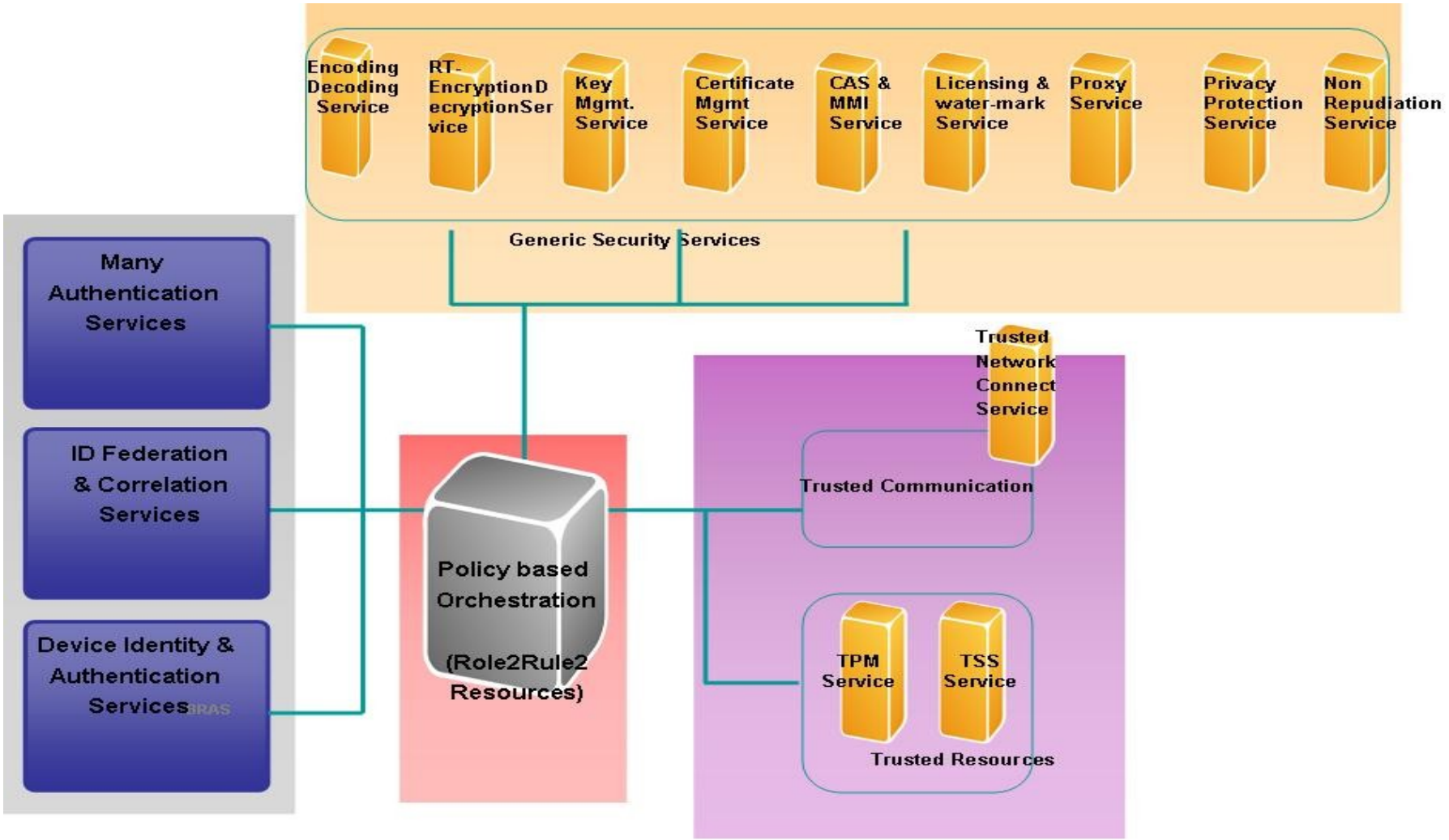
Content & Service Centric IDS

Integration with Service Registry Repository, ESB, DRM, Service specific Policies, & more.

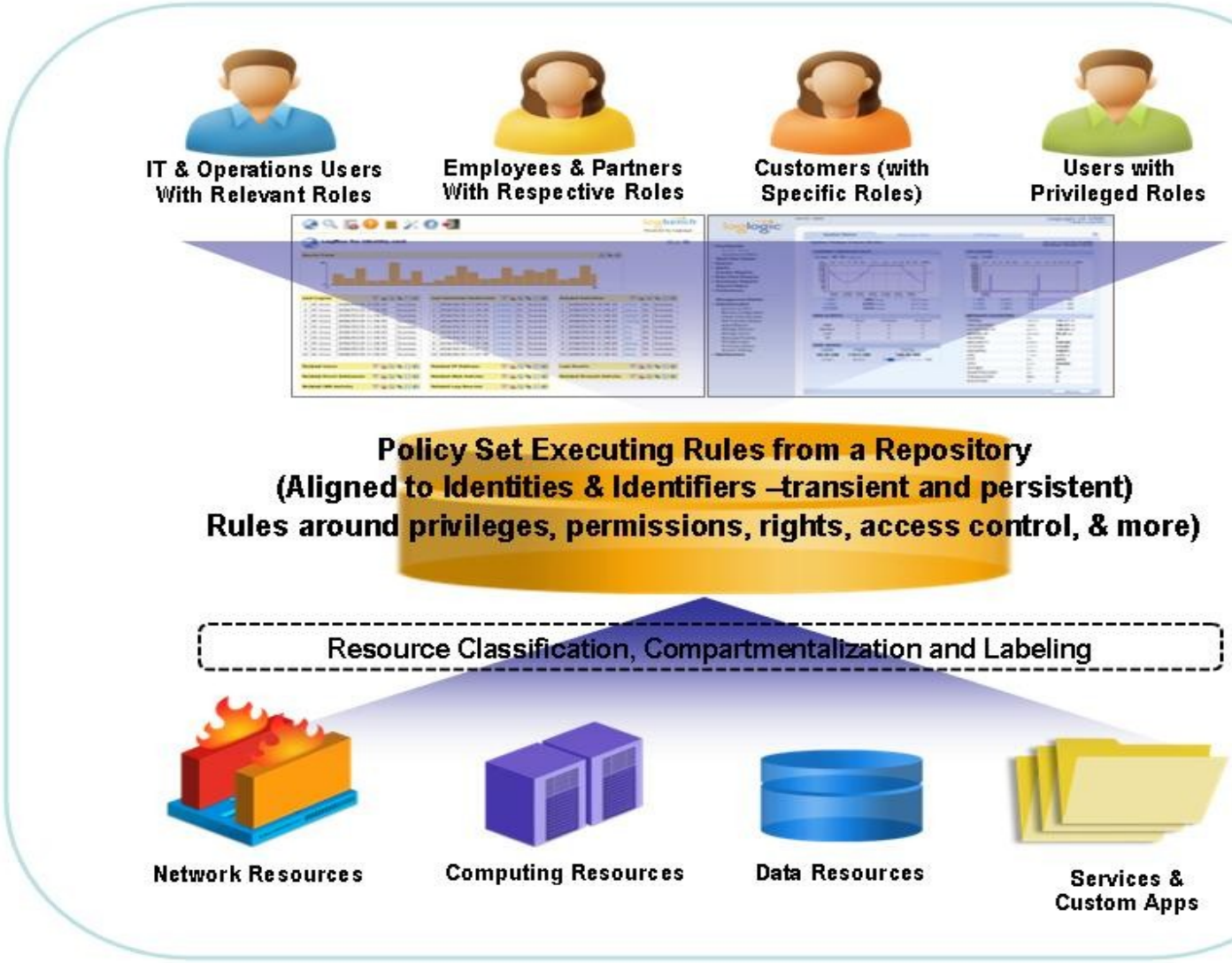
Policy based Security Services (SAAS)

- Authentication Services (many Authentication types/contexts)
- Policy Services (Rule Management Services)
- Federation Services (CDSSO, COT, interlinked Federation)
- Session Services (distributed sessions – network facing and service facing)
- Logging Services (end to end)
- Token Management Services (token table, token transfers, etc.)
- Repository Services (agnostic to repository technology)
- Key Management Services (certificates, algorithms, enc/dec)
- Identity Reputation Services (history of transactions)
- Identity Assurance Services (NIST/Liberty)
- Identity and Trusted Computing (Resource labeling+TPM)
- Identity Privacy Management Services (icons)
- Role Management Services (full life cycle of roles)
- Identity Context Services (location, presence, etc.)
- Identity Mobility Services (roaming, distributed sessions, etc.)
- Identity Service Management Services (service provisioned to entities)
- Identity DRM Services (disintermediation)
- Identity Audit and Compliance Services (reporting)

Policy based Security Services (SAAS)

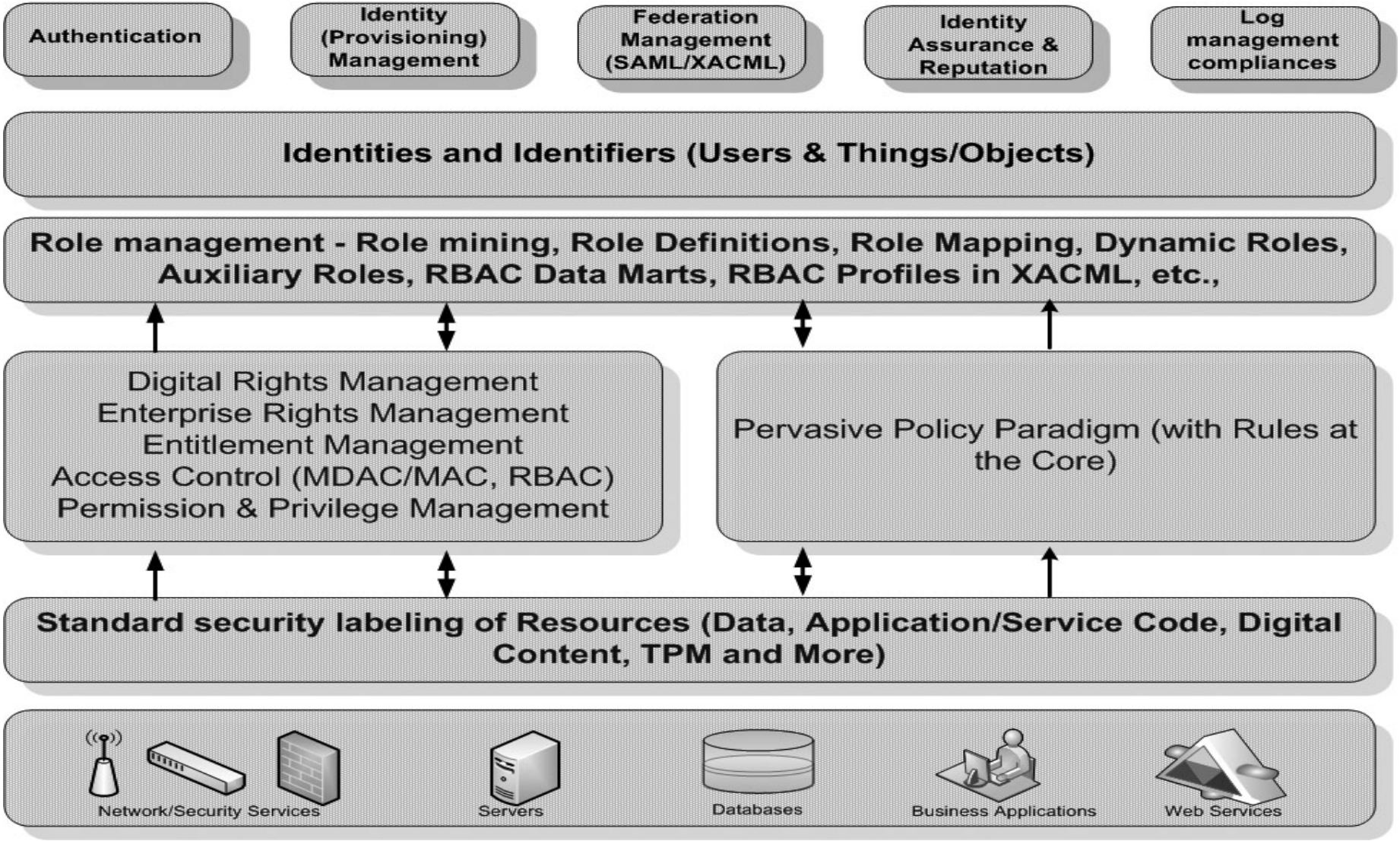


Pervasive Policy Paradigm

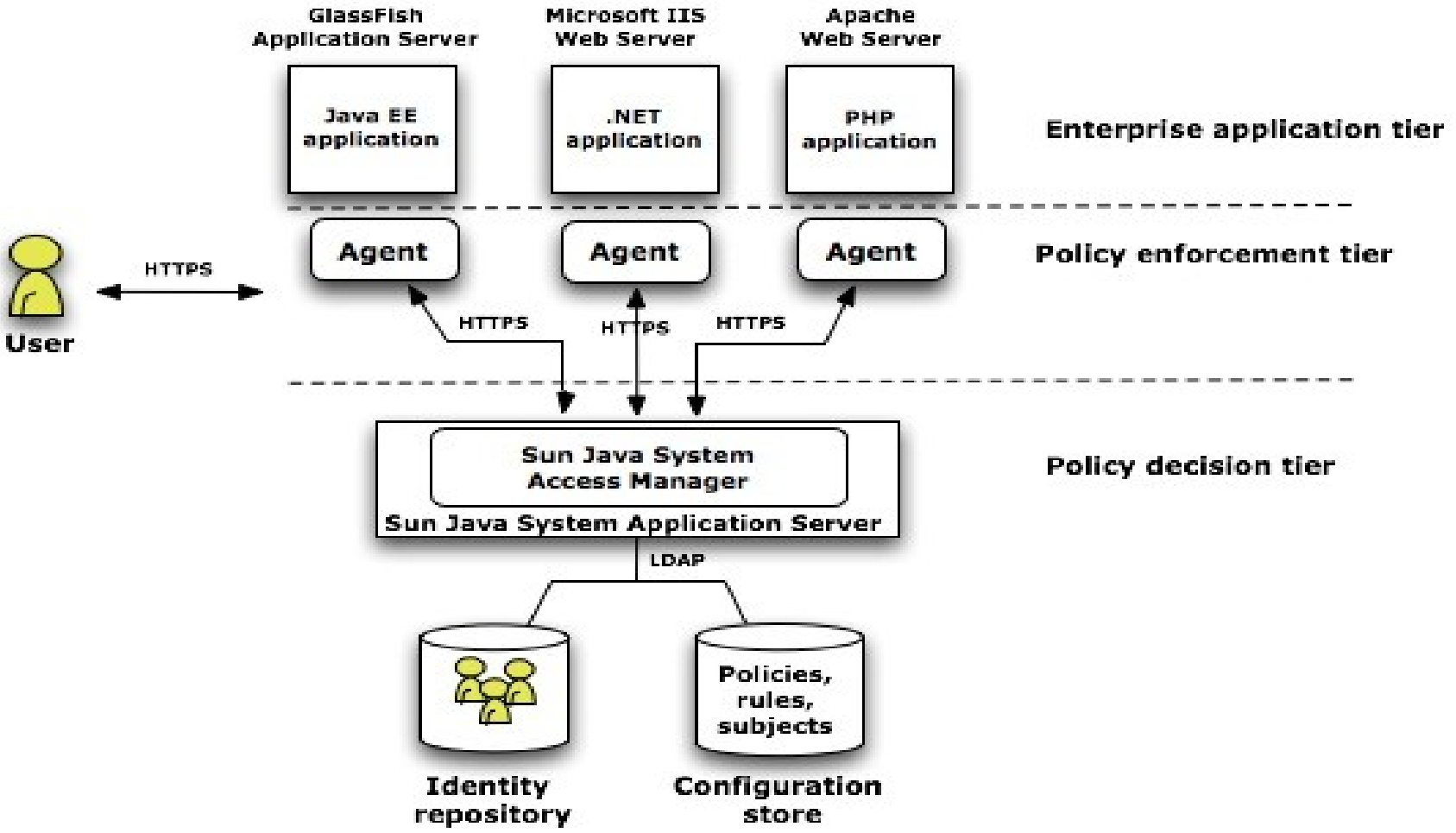


Rules that ensures appropriate security services are leveraged (encryption, certificate/ keys, token mgmt, scrambling, licensing, reputation mechanism, protocols, label mgmt, platform security modules (TPM, TNC, TC), etc.

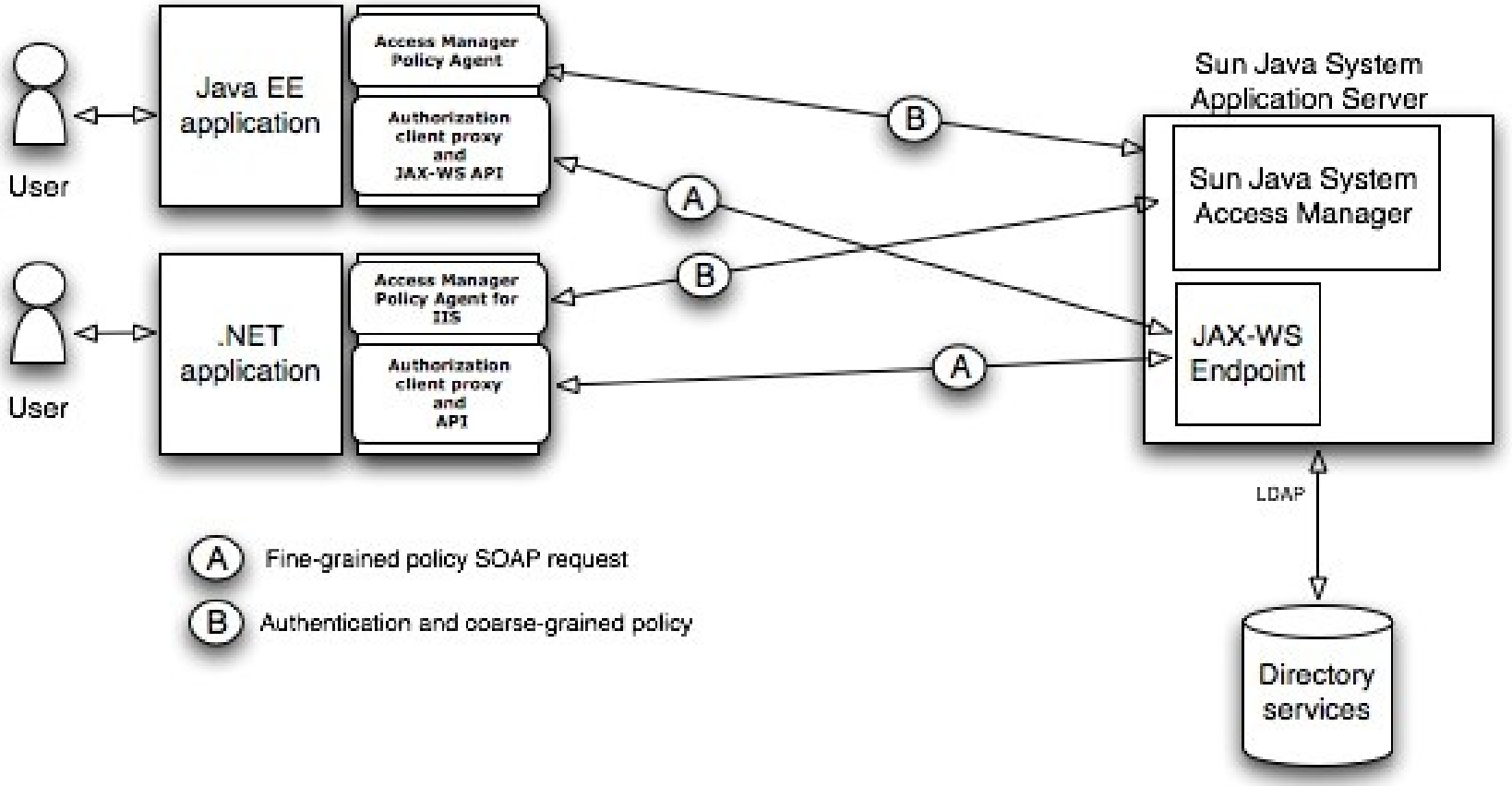
Pervasive Policy Paradigm



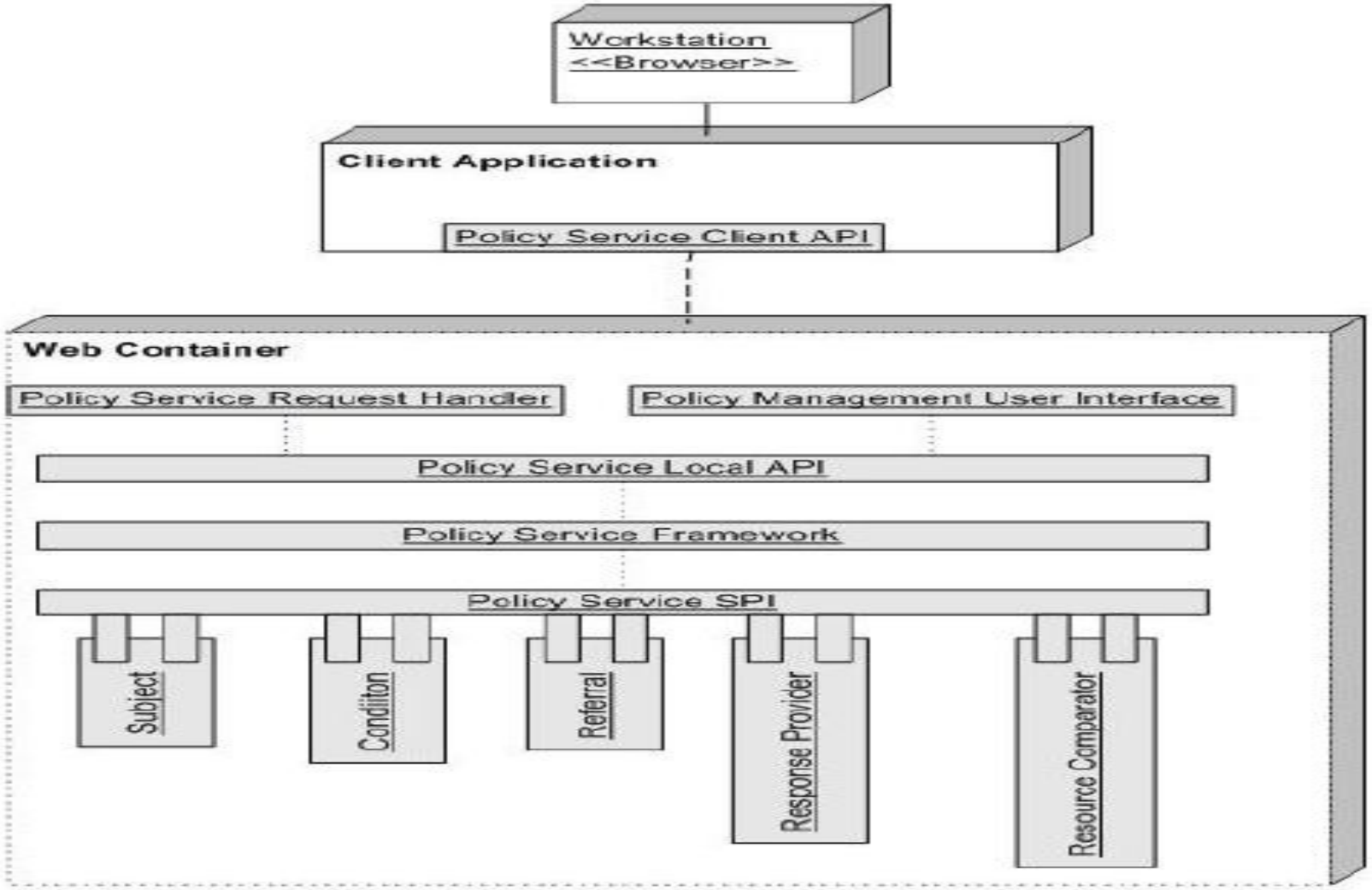
Pervasive Policy Paradigm



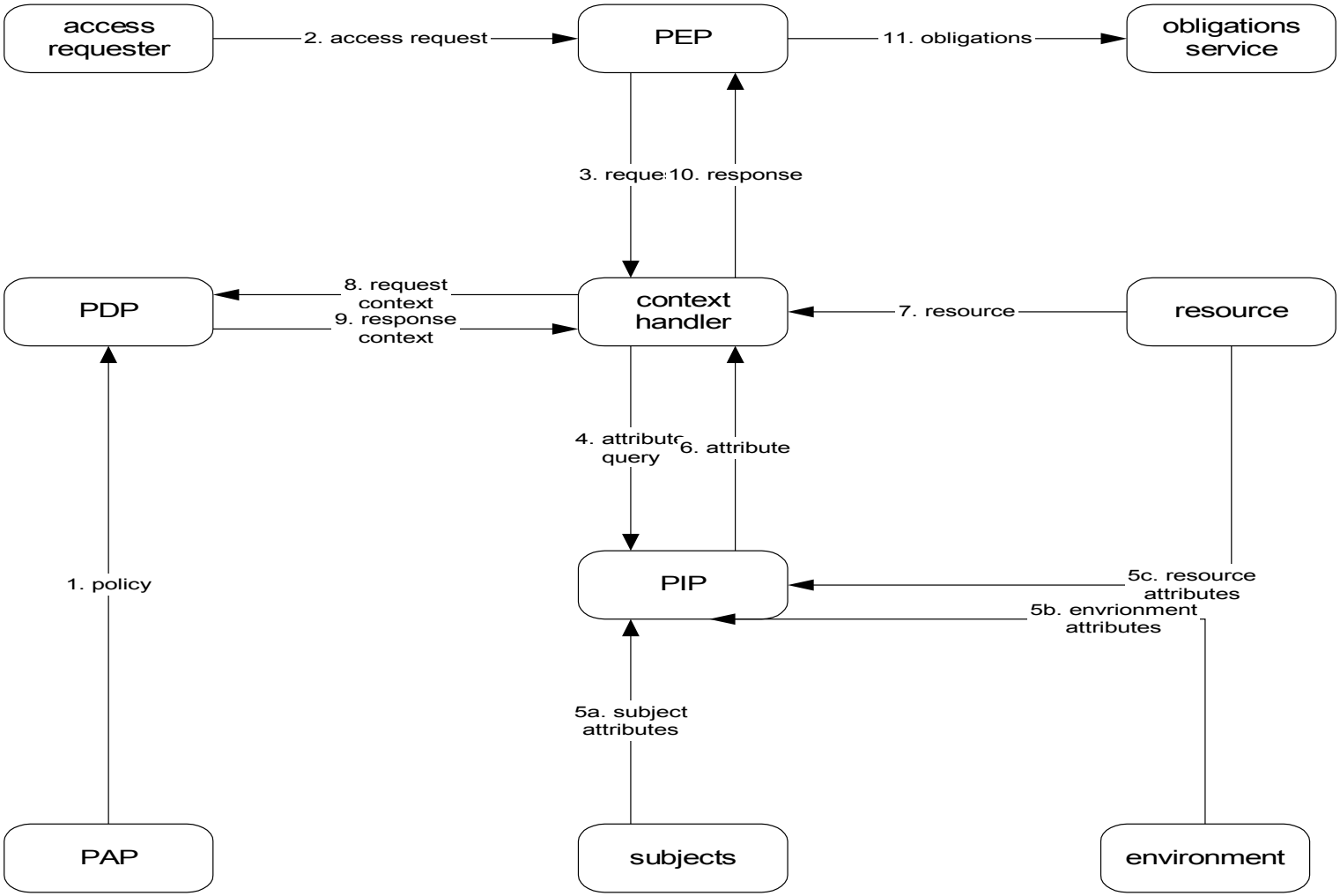
Pervasive Policy Paradigm



Policy System for SOA (FAM 8.x Architecture)



Policy System for SOA (FAM 8.x Architecture)



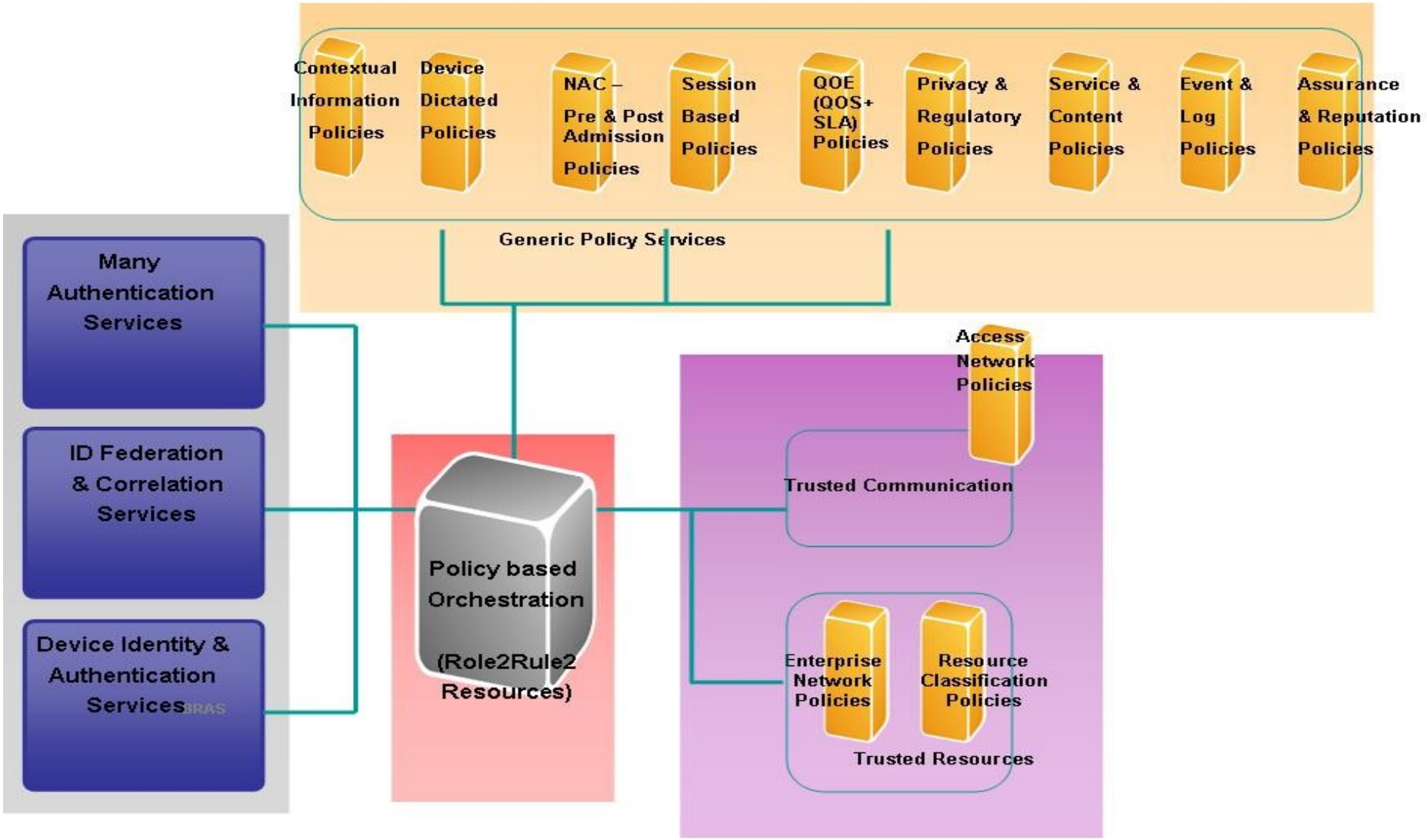
Pervasive Policy Paradigm

- To provide a method for combining individual rules and policies into a single policy set that applies to a particular decision request
- To provide a method for flexible definition of the procedure by which rules and policies are combined
- To provide a method for dealing with multiple subjects acting in different capacities
- To provide a method for basing an authorization decision on attributes of the subject & resource
- To provide a method for dealing with multi-valued attributes
- To provide a method for basing an authorization decision on the contents of an inf resource
- To provide a set of logical and mathematical operators on attributes of the subject, resource and environment
- To provide a method for handling a distributed set of policy components, while abstracting the method for locating, retrieving and authenticating the policy components
- To provide a method for rapidly identifying the policy that applies to a given action, based upon the values of attributes of the subjects, resource and action
- To provide an abstraction-layer that insulates the policy-writer from the details of the app env
- To provide a method for specifying a set of actions that must be performed in conjunction with policy enforcement

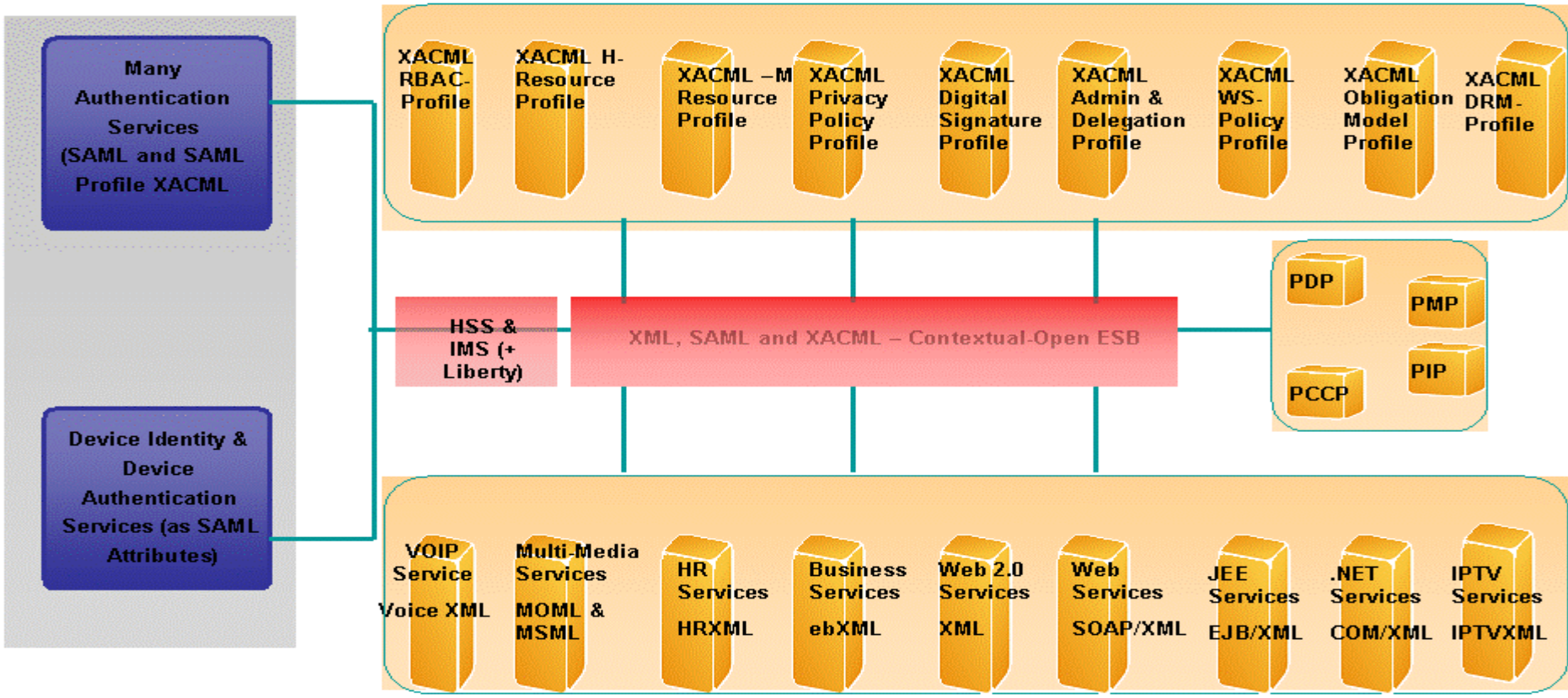
Pervasive Policy Paradigm

- Identity enabled Derived Device Policies
- Identity enabled Access Networks Policies
- Identity enabled QOS/QOE Policies
- Identity enabled Session Specific Policies
- Identity enabled Privacy Preservation Policies
- Identity enabled Service Security Policies
- Identity enabled Content Control Policies
- Identity enabled Enterprise Network Policies
- Identity enabled Regulatory Requirement Policies
- Identity enabled Event Log Policies
- Identity enabled Contextual Policies
- Identity enabled Policy Assurance (with PCCP)
- Policy Orchestration

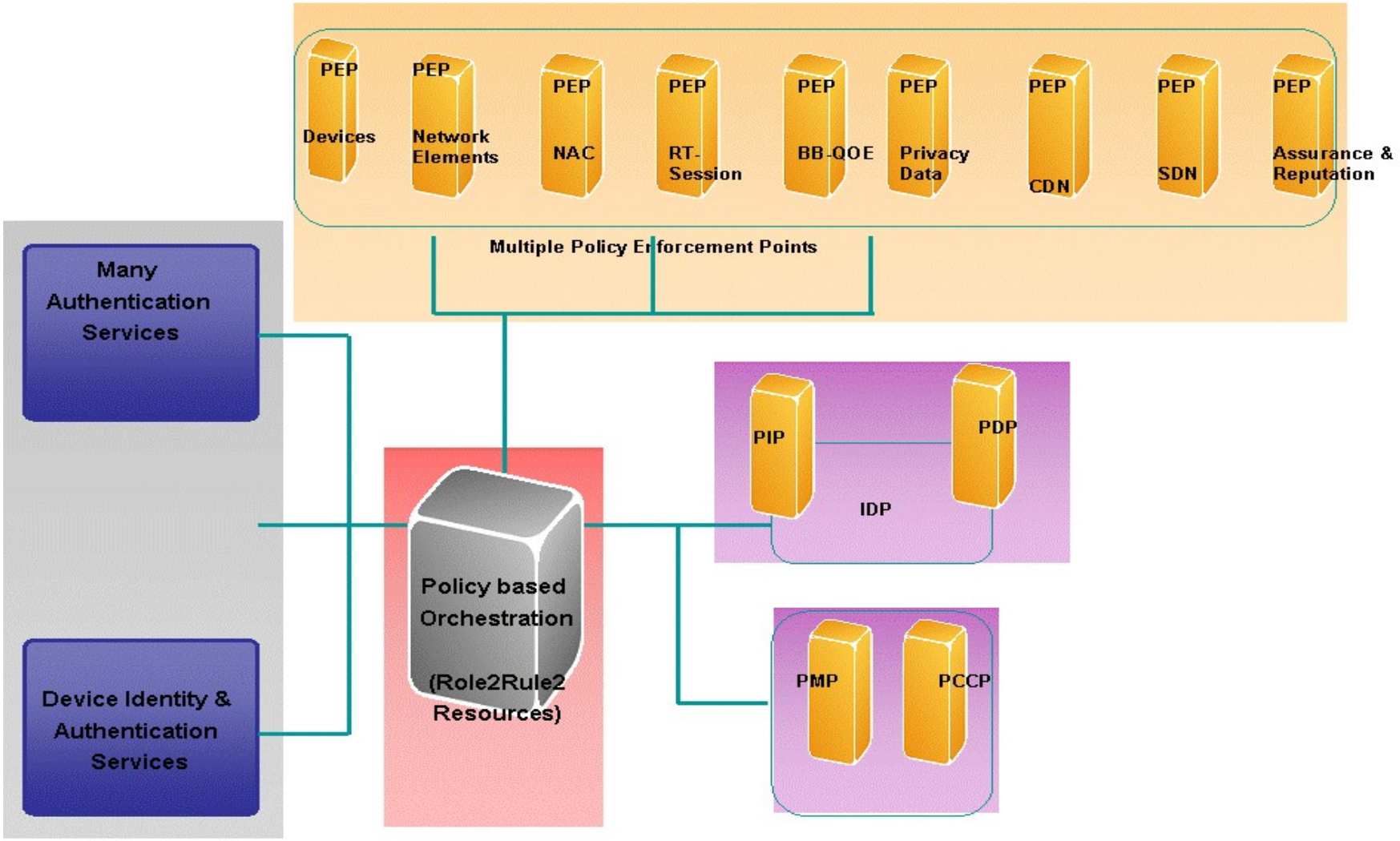
Policy Orchestration for Control & Alignment



Policy Orchestration using XML & XACML



Policy Orchestration for Control & Alignment



100+ XACML Papers & Series

- NIST Paper on Device & Log Policies (PDA or any Mobile IP Devices)
- NCSA Paper on Policies for VO (PIP and PCCP)
- Dr. Mouli's papers on RBAC/XACML, Privacy Policies, Enterprise Policies and Policy Inference (4 papers)
- Papers on Contextual Policies and Mobile Agents
- Joint papers with ISV, NEP & Industry Forum
  - Book 1: Identity and Security (06/07)
  - Book 2: Identity and Policy (06/08)
  - Book 3: Identity and SOA -09 (context, mobility, Enterprise SOA, etc.)
  - Book 4: Identity and Trust -09 (TCG, TPM, COT, TNC, etc.)
  - Book 5: Identity and eGov 10 (Assurance and Reputation)

100+ POC, Pilot and Proto-types

- Cisco, Juniper, Alcatel, Nokia/Siemens and many more NEP's integrate with FAM for NAC policies
- Trust Digital and I-Ovation like ISV's for Device policies
- TAZZ & Bridge-water for BB 4G Networks for IMS, IPTV policies
- Kabira and Telcordia for RT-Charging Policies (session based policies)
- True-baseline, Cisco and Juniper for QOS policies
- Layer 7 Technologies for Service policies
- Reactivity and Securent for Enterprise Network policies
- Log Logic for Event and Log Policies
- CDN policies (projects)
- OSS/BSS TMF Co-op and policy orchestration
- I-pass for Network facing policy orchestration
- Sun ESB and FAM for service policy orchestration

What did we learn?

- XACML is great for abstraction of policies and Attributes (ABAC) -service orchestration automatically invokes polices
- Policies themselves require operator or user defined workflow (e.g., device + NAC + user level authN -iPass)
- Continuity in end point security (event based Policies)
- RT Policy Checking and Certification is very important (PCCP) requires orchestration
- Policy Information Point can leverage XDI (PIP) and can share context via an Open ESB
- Use-full for aligning Service SLA with Network QOS
- Changes is Environment (code RED, breaches in the network, etc.) forces dynamic policy work-flows
- Alignment to Audit Rules and Log based Policies
- Session Specific policies invoked when highly Security Sensitive Service is invoked.

What did we learn?

- Stage 0 – Registration is Key for Success – this includes identity provisioning, role management and rule definitions, workflow definitions, etc.
- Do not forget things like -Support multiple password policies for a user, depending on the targets on which he is having account etc.
- RBAC is a project by itself
- POC and Demo's when followed up with Real Projects require Executive Sponsors, Resources (SI, Partners), and a methodology
- Most projects are JEE environments with Heterogeneous platforms
- Not all XACML profiles will be leveraged for initial stages of project – resource profile with request response works well for most use cases (H and M), RBAC profile is extremely power full, XACML –WS policy has been used, privacy policy exchange tested, QOS policies trialed.
- Plan for acquisitions and mergers impacting project

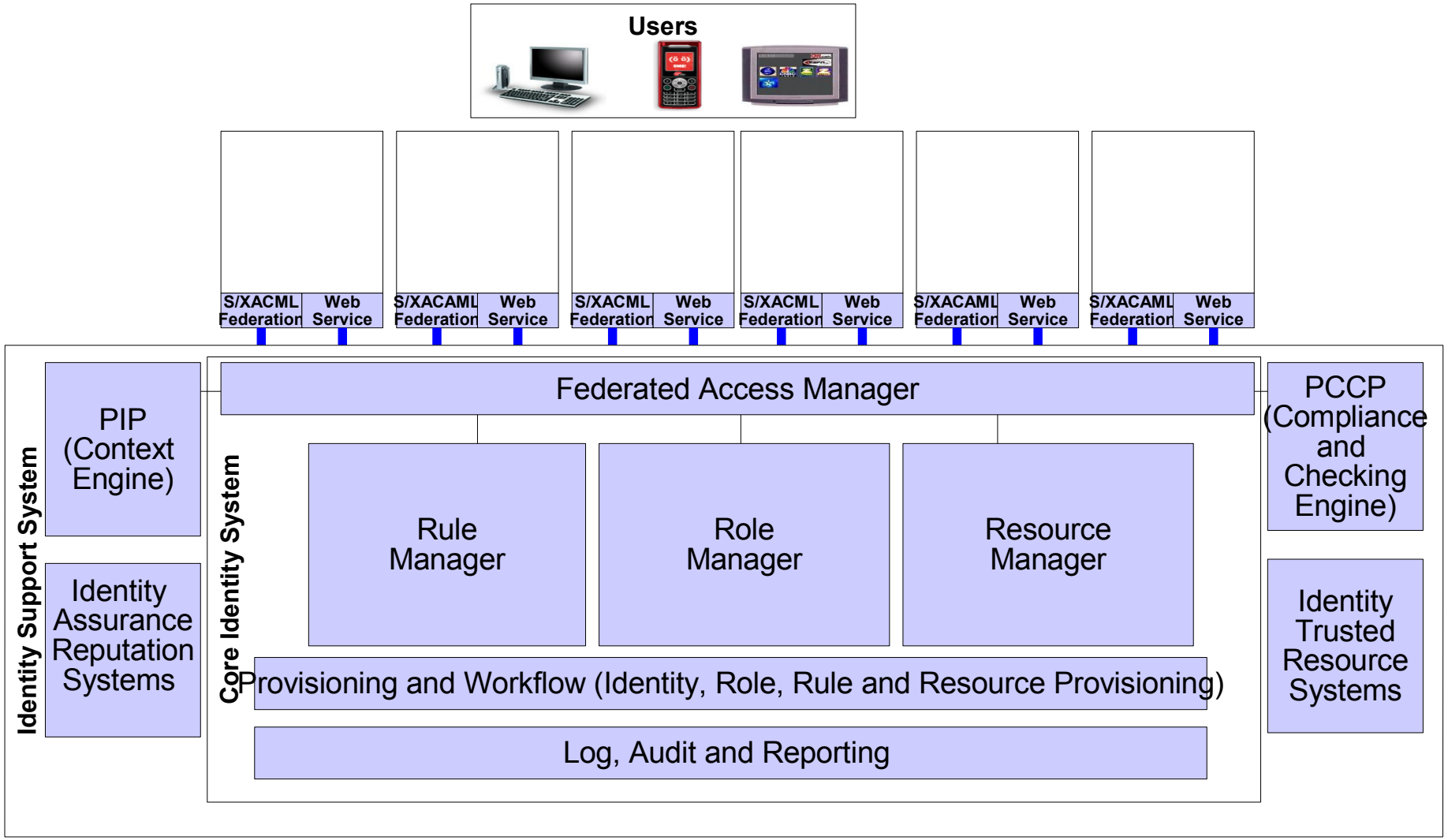
What did we learn?

- Policy Orchestration works – not all use cases require end to end orchestration – but will require a subset of policies to be orchestrated – NAC policy, Device Policy and AuthN policy for example – with the Req-Resp model between multiple PEP and PDP's
- PIP integration is basically Secure Exchange of Context Data and Profiles (location, presence, preference, etc.)
- Policies are integral part of Identity Assurance
- PCCP – Policy Compliance and Checking points external to a PMP will be required for ongoing continuous validation of legitimacy of policy with inference and other techniques
- XACML –DRM ? Abstraction from multiple DRM techniques
- Delegation and Administration profile in a PMP (3rd party)
- Obligation (3rd party)

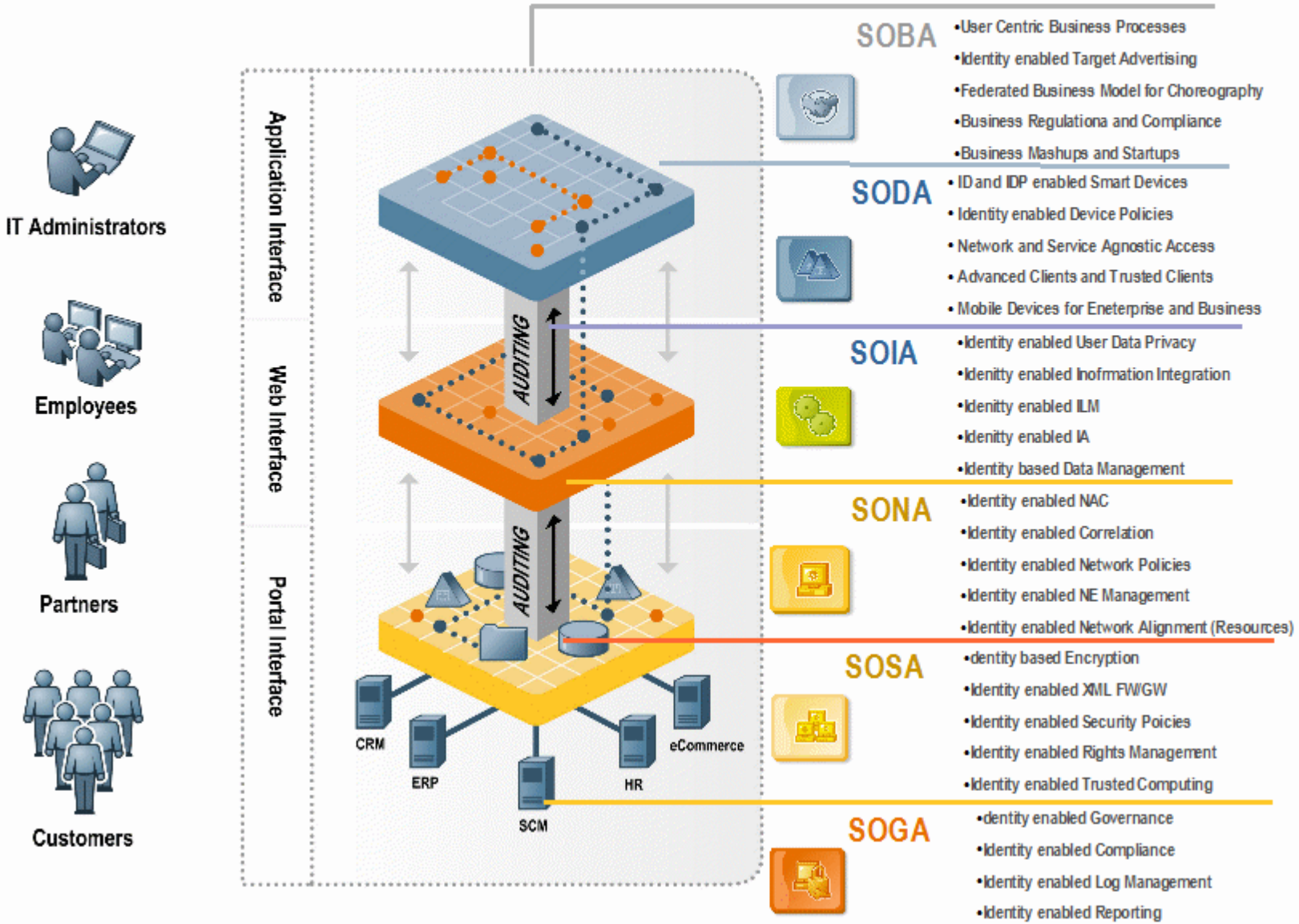
What did we learn?

- Its not easy – The tools are there – but it requires extensive discipline and Project Management
- Each project is driven by specific domain level requirements (Privacy and Audit or QOE)
- Partners are extremely important for creating an Eco System
- POC and Proto-type in a Lab is a different project from Production (Architecture Assessment helps and Pilots as well)
- Leverage Speciality integration partners
- Address scope with phases (1 to 5)
- Create Policy Building Blocks (with XACML abstraction)
- Align with Role Management, Resource Management and SOA projects
- Policy Infrastructure is key for ID Governance

An evolving Identity System



Identity System – An Axel for Alignment



Thank You !!!

rakesh.radhakrishnan@sun.com

http://www.network-identity.com

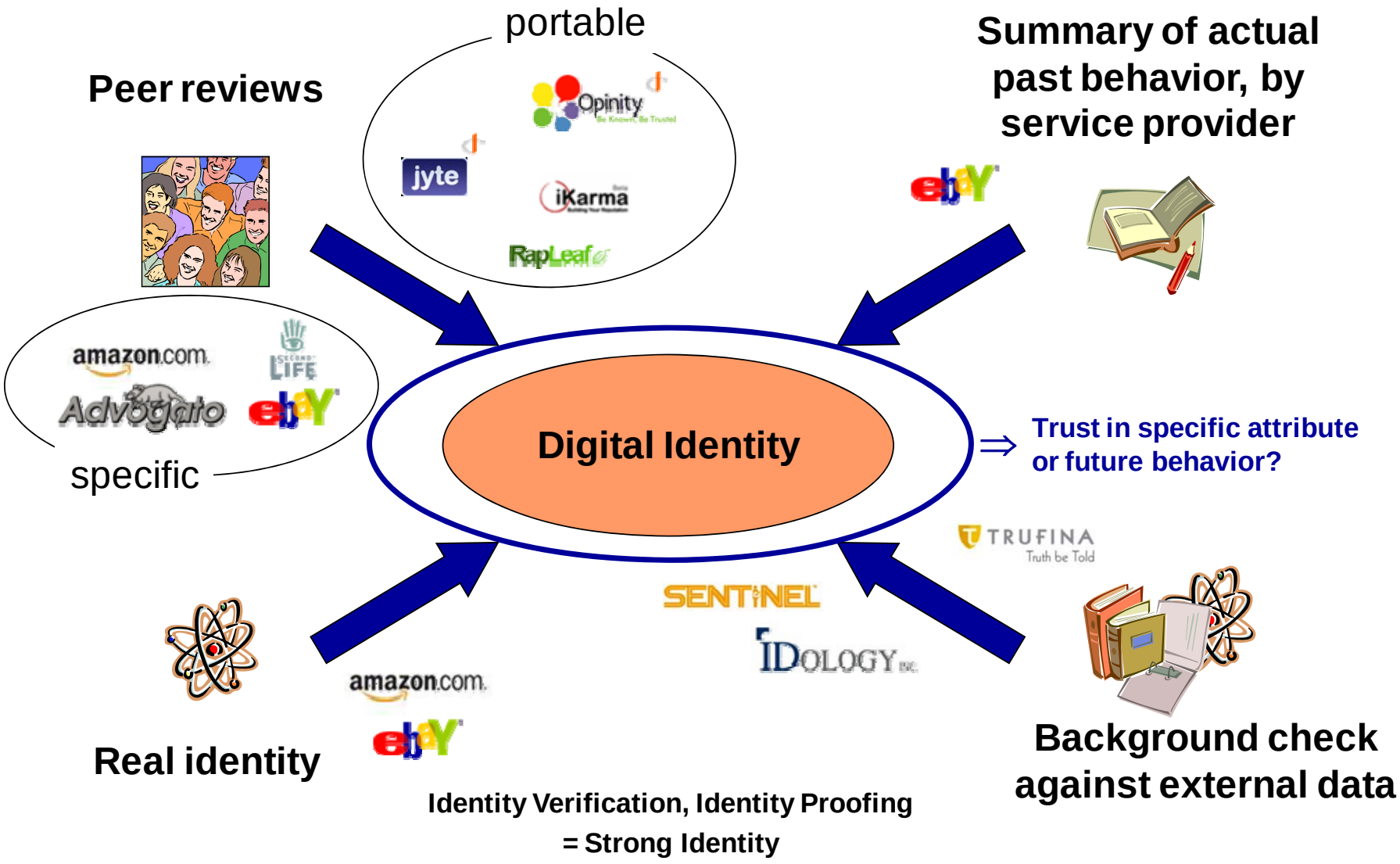
OASIS

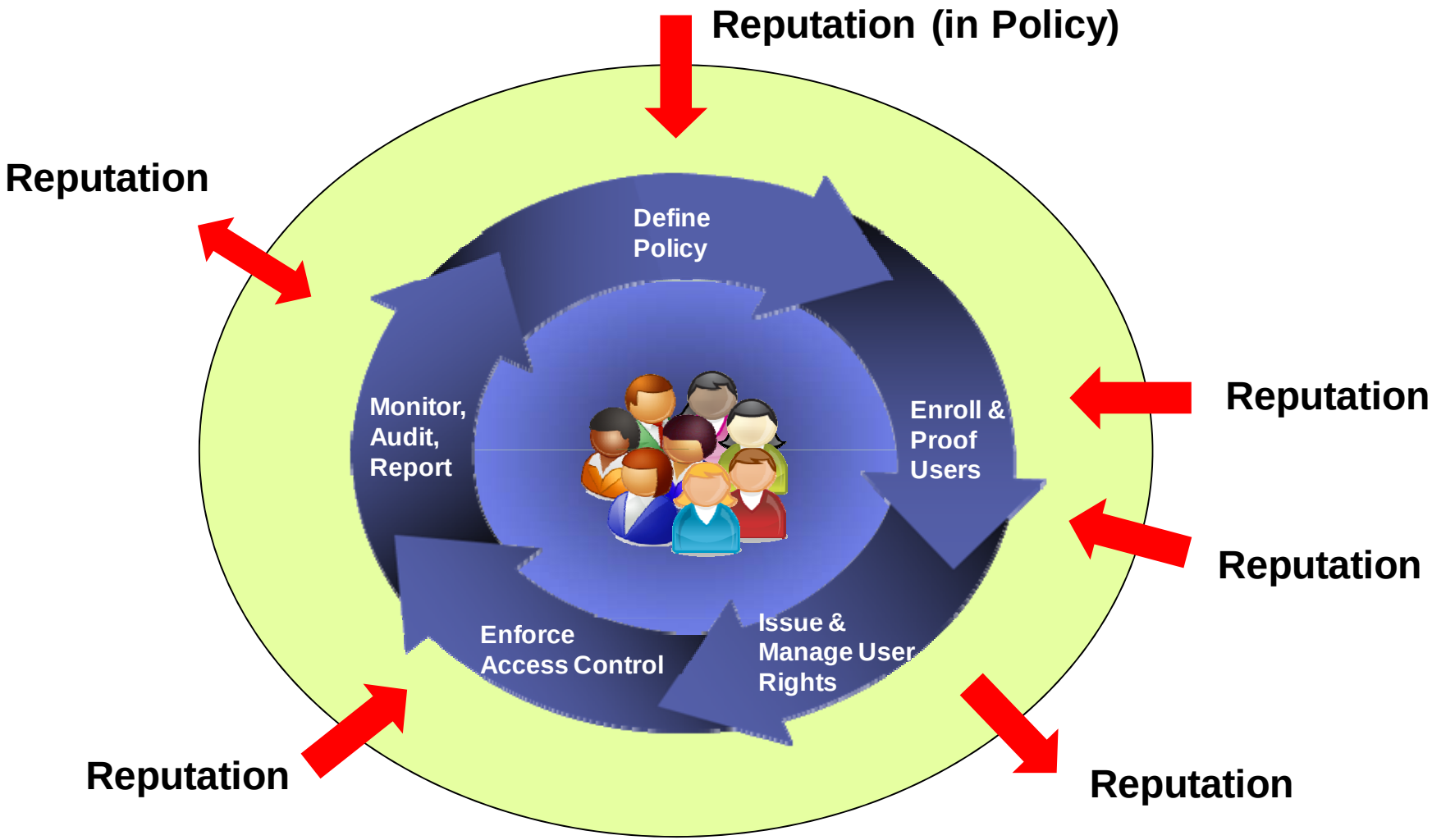
Open Reputation Management Systems

OASIS

ORMS

- Develop scenarios for reputation management
  - Reputation of individuals, business partners, services processes, possibly even data
- Develop reference model and reputation technology
  - Flexible reputation data model
  - Framework and protocol/s for exchanging and porting reputation data
  - Evaluation algorithms for mapping reputation to risk / risk levels
  - Support for privacy, multiple identities, identity resolution


portable
Peer reviews
Summary of actual past behavior, by service provider
Opinity
jyte
iKarma
RapLeaf
amazon.com
Advogato
eBay
specific
Digital Identity
⇒ Trust in specific attribute or future behavior?
TRUFINA
Truth be Told
SENTINEL
IDOLOGY
amazon.com
eBay
Real identity
Background check against external data
Identity Verification, Identity Proofing
= Strong Identity


Reputation (in Policy)
Reputation
Define Policy
Enroll & Proof Users
Issue & Manage User Rights
Enforce Access Control
Monitor, Audit, Report
Reputation
Reputation
Reputation
Reputation

Thoughts

- Need anti-use cases
- Need to consider protections and fail-safe mechanism / policy to prevent cascading impact... whether accurate, in error or malicious... kind of like the circuit breaker in the stock market
- A "reputation score" should likely be computed within a situation / context with an expiration, rather than via static assignment. This has more to do with ensuring the right context, rather than assuming that scoring results would be highly variable.

Thoughts

- Humans and entities, roles and personas.... which is to be considered when Jim uses a credit card at a store? Jim is a homeowner or Jim is the sole proprietor of a construction company.
- Contexts and aggregations .... what info is relevant ? Jim the citizen and Jim the sole proprietor are both bad drivers; Jim the citizen pays taxes, but Jim the sole proprietor can't pay some bills
- Attribute weighting and order of precedence ... is it ever more or less important to.... be a good driver, to pay your bills, to receive a reference letter from a priest, to be a democrat / republican, etc.?

Thoughts

- Constraints and Degrees of Freedom... likely better to define independent dimensions of reputation and then look for trends in deviation for dimensions rather than to develop an aggregate value too early in the process
- Threats and vulnerabilities... playing the reputation system; use of surrogates, cascading impact; homogeneity of analyses
- Reputation "qualities" .... freshness / stallness, trend (derivative or other)

OASIS

OpenID Provider Reputation

IDtrust Symposium, March 4-6, 2008

Drummond Reed, Cordance

OpenID has a crying need for reputation services!

- The OpenID authentication protocol has no inherent trust model
- Any OpenID user can register with any OpenID provider (OPs)
- This leaves a wide open question for OpenID relying parties (RPs):

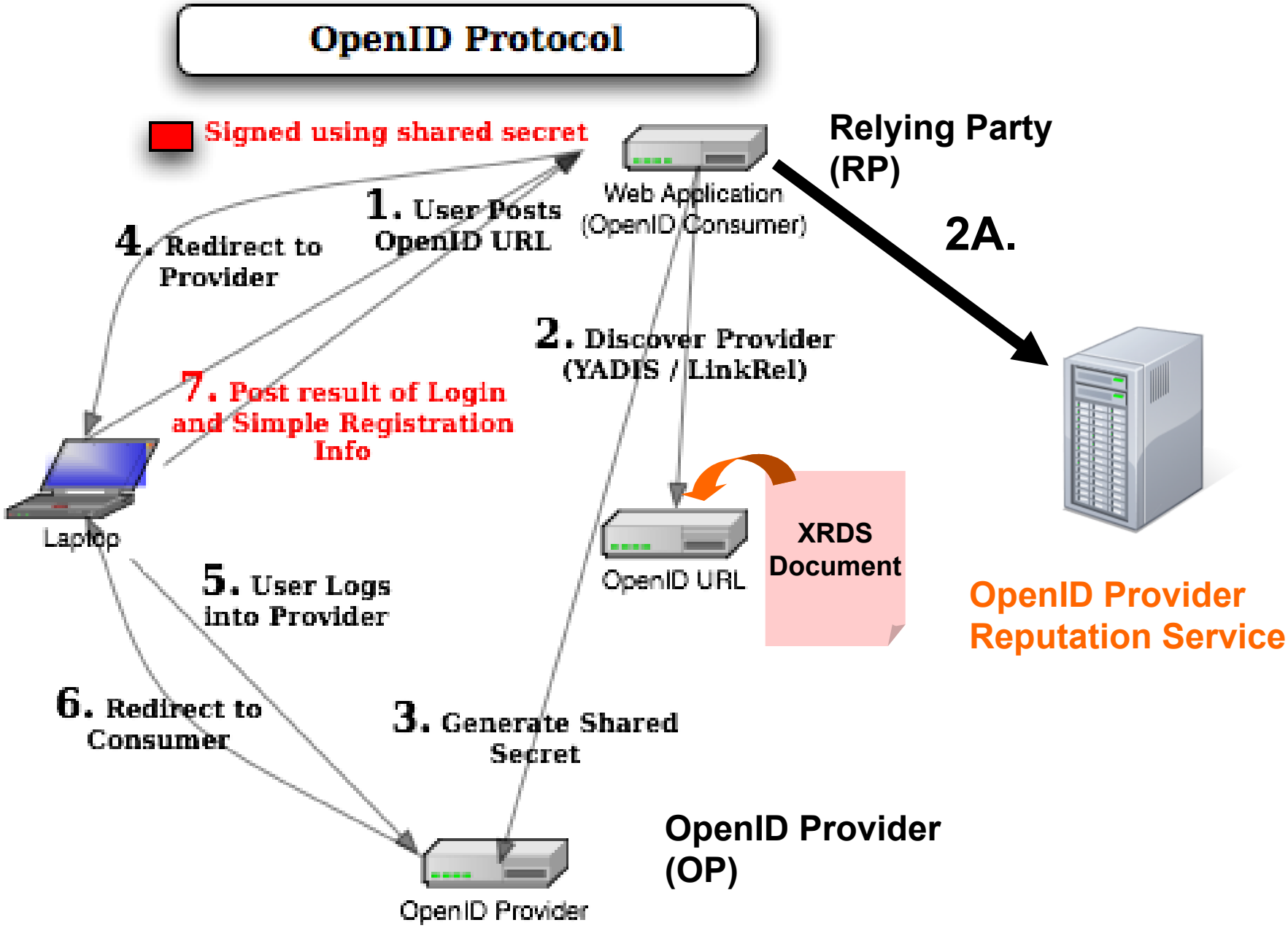
Which OpenID Providers should we trust???

The evidence

- The largest OpenID providers in the world...
  - Yahoo
  - AOL
  - Google Blogger
- ...do not accept OpenID logins
- They only *issue* OpenIDs

The solution the OpenID community prefers...

- ...is not a top-down federation model, but an organic reputation network model
- This fits the open, scalable, organic nature of OpenID
- It would also map easily into the OpenID protocol flow


OpenID Protocol
Signed using shared secret
Web Application
(OpenID Consumer)
Relying Party
(RP)
1. User Posts
OpenID URL
2A.
4. Redirect to
Provider
2. Discover Provider
(YADIS / LinkRel)
7. Post result of Login
and Simple Registration
Info
Laptop
OpenID URL
XRDS
Document
OpenID Provider
Reputation Service
5. User Logs
into Provider
6. Redirect to
Consumer
3. Generate Shared
Secret
OpenID Provider
(OP)
OpenID Provider

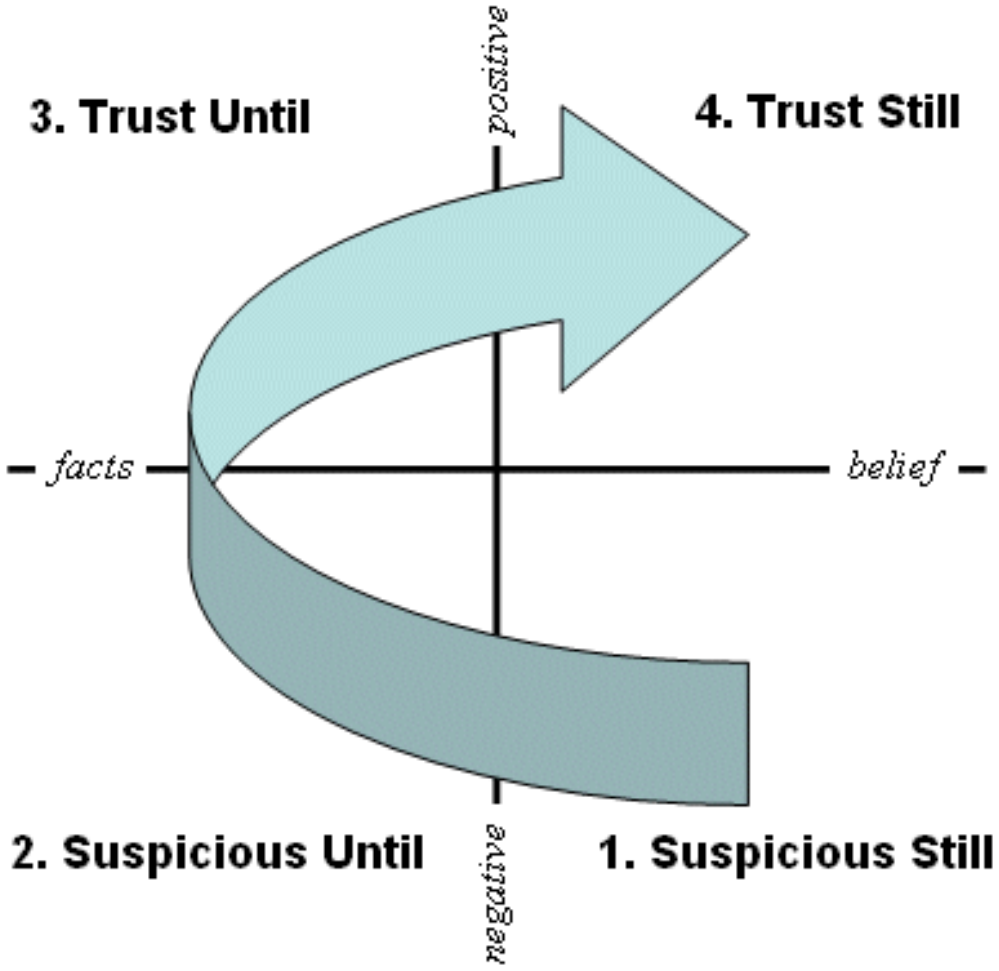
Trust, a dictionary definition

- Real-world or Social: The concept of social trust can be obtained from dictionaries, such as Merriam Webster: "
- 1 : assured reliance on the character, ability, strength, or truth of someone or something
- 2 a : dependence on something future or contingent : HOPE b : reliance on future payment for property (as merchandise) delivered CREDIT
- 3 a : a property interest held by one person for the benefit of another b : a combination of firms or corporations formed by a legal agreement; especially : one that reduces or threatens to reduce competition
- 4: (1) : a charge or duty imposed in faith or confidence or as a condition of some relationship (2): something committed or entrusted to one to be used or cared for in the interest of another b : responsible charge or office c : CARE, CUSTODY <the child committed to her trust>"

Basic Trust Models

- **Suspicious still**. Don't ever trust anyone, even after they have done something nice.
- **Suspicious until**. Don't trust anyone until they prove themselves.
- **Trust until**. Trust people until they screw up.
- **Trust still**. Trust people even after they make mistakes, sometimes even when they hurt you.

Models in Practice



Implementing Trust Models

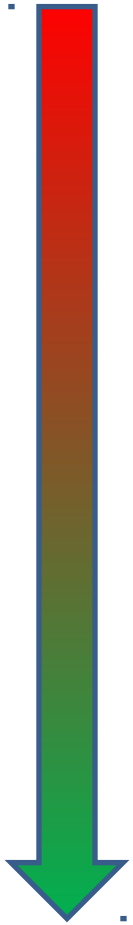
- NSA: "a trusted system or component is one with the power to break one's security policy“
- X.509: "Generally, an entity can be said to "trust" a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects…”
- ABA Digital Signature Guidelines (ABADSG) I: trust is not defined per se, but indirectly, by defining "trustworthy systems" (or, systems that deserve trust) …”
- PGP: even though PGP uses the word trust extensively, such as in web-of-trust…

Confusing the Implementation…

- Public Key Infrastructures
- PGP Web of Trust
- Trust Frameworks- Liberty, Bayesian, WS-Trust and others
- Policy and Governance
- Trust Metrics -karma, feedback
- Reputation

Identity Verification

- Personal Consent
- Parental Consent
- Peer Verification
  (Distributed Trust)
- Delegated Administration
  (Financial Institution)



- I am Chris
- This is my child Chris
- Based on what I know of Chris, that is Chris
- Based on Chris's breeder documents, or derivatives, this is Chris

Trust in the Veracity of Identity

A basic explanation of a Trust and Reputation system

- Peers store opinions on their experiences; they store an opinion about the people they interact with and the systems they are on.
- Peers share their opinions providing recommendations for people and systems.
- A peer’s opinion can be weighted based on how much the querying peer trusts them.
- The aim of the system is to eliminate malicious peers and systems

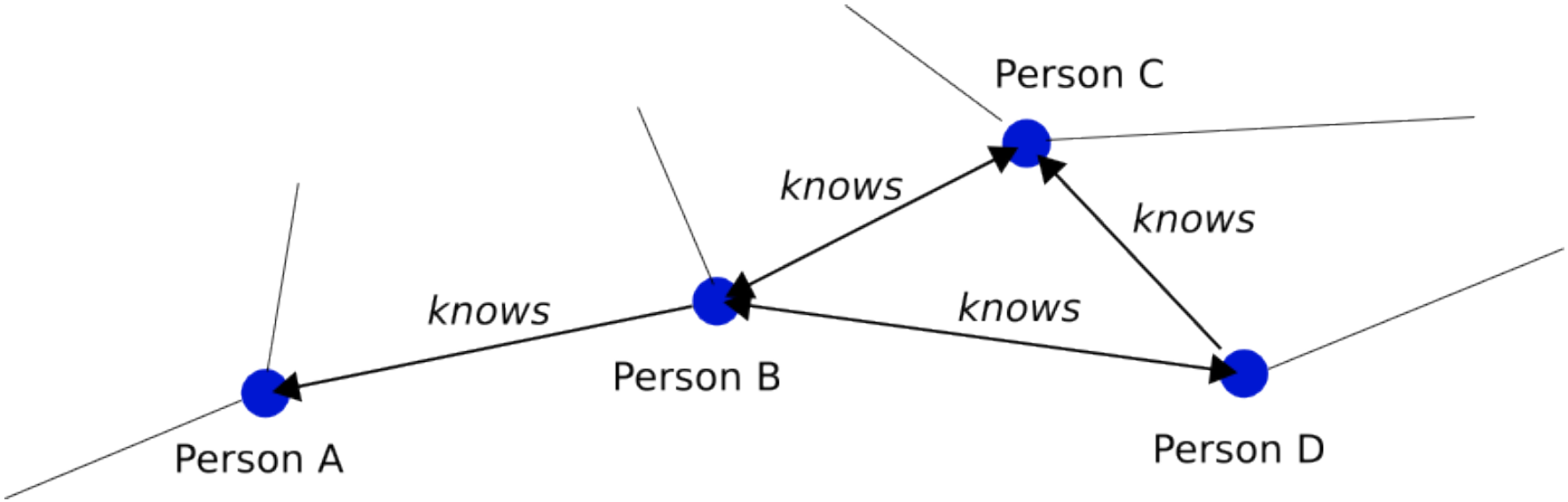
An Example

- David has two friends John who is a mechanic and Scott who is a Doctor.
- David trusts Scott with a medical complaint but not to fix his car and respectively, trusts John to fix his car but not to diagnose a medical condition.
- So in the context of fixing a car John is trustworthy, but Scott is untrustworthy.
- What one peer may consider a good file is not what another peer would consider good. For instance peer A's priority in a good file is its content regardless of its quality.

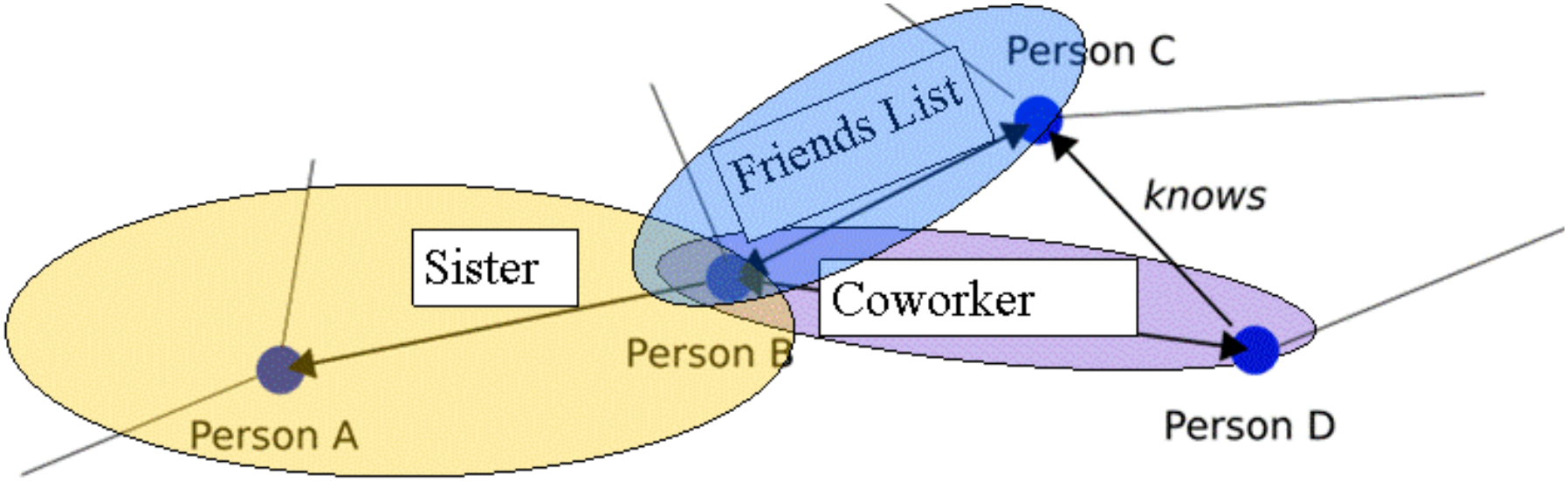
The Doppelganger effect

- People want multiple identities it;
  - Use a pen/pseudo name
  - Anonymity
- People have identities within different contexts;
  - Family
  - Work
  - Online Friends

A FOAF Example



A Layer Deeper

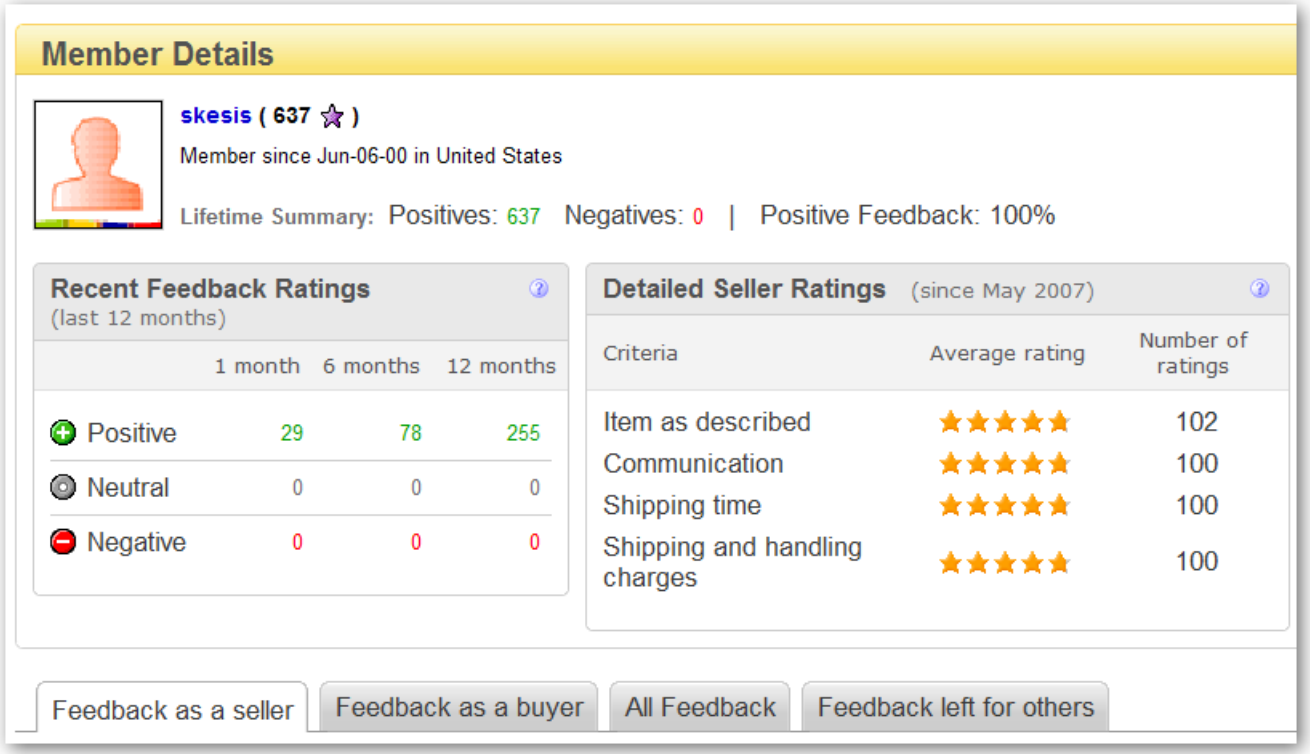


Trust and Reputation are based on context of interaction

The Need for Common Data Models

- How to describe the trust among persons from one site to another?
- Do interactions on one site affect another… would it be helpful to know?
- What commonality exists and how can they become transparent?
- How can privacy be preserved?

I am a Power Seller on ebay



Member Details

skesis (637 ★)

Member since Jun-06-00 in United States

Lifetime Summary: Positives: 637 Negatives: 0 | Positive Feedback: 100%

Recent Feedback Ratings
(last 12 months)

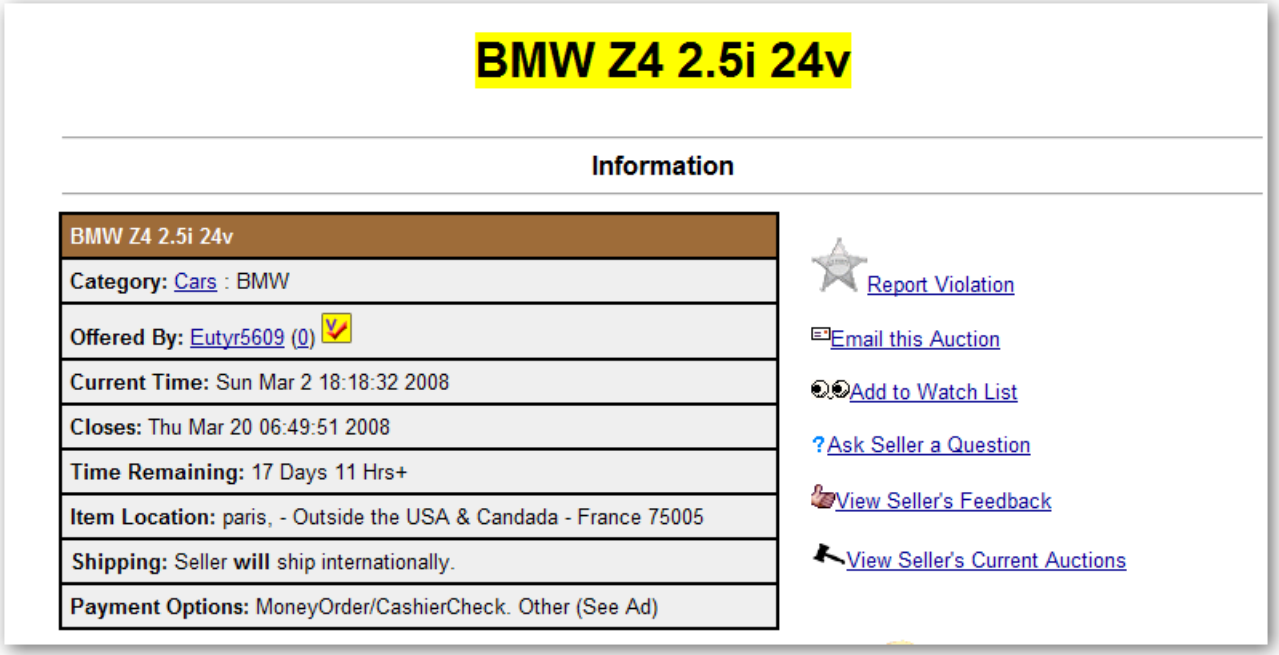
	1 month	6 months	12 months
Positive	29	78	255
Neutral	0	0	0
Negative	0	0	0

Detailed Seller Ratings (since May 2007)

Criteria	Average rating	Number of ratings
Item as described	★★★★★	102
Communication	★★★★★	100
Shipping time	★★★★★	100
Shipping and handling charges	★★★★★	100

Feedback as a seller | Feedback as a buyer | All Feedback | Feedback left for others

Would You Buy From Me on Auction Fire?



BMW Z4 2.5i 24v

Information

BMW Z4 2.5i 24v
Category: Cars : BMW
Offered By: Eutyr5609 (0)
Current Time: Sun Mar 2 18:18:32 2008
Closes: Thu Mar 20 06:49:51 2008
Time Remaining: 17 Days 11 Hrs+
Item Location: paris, - Outside the USA & Candada - France 75005
Shipping: Seller **will** ship internationally.
Payment Options: MoneyOrder/CashierCheck. Other (See Ad)

Report Violation

Email this Auction

Add to Watch List

Ask Seller a Question

View Seller's Feedback

View Seller's Current Auctions

The Need for Transparency

- How do my actions follow me?
- How can trust be transferred, or reputation be borrowed from one site to another?
- Can other commercial data enrich my reputation – credit score, etc?

Identity & Policy
(for Open Reputation)

March, 4th 2008
NIST
Identity and Trust Symposium

Rakesh Radhakrishnan
Chief Identity Integration Architect Telco)
Sun Microsystems, Inc.

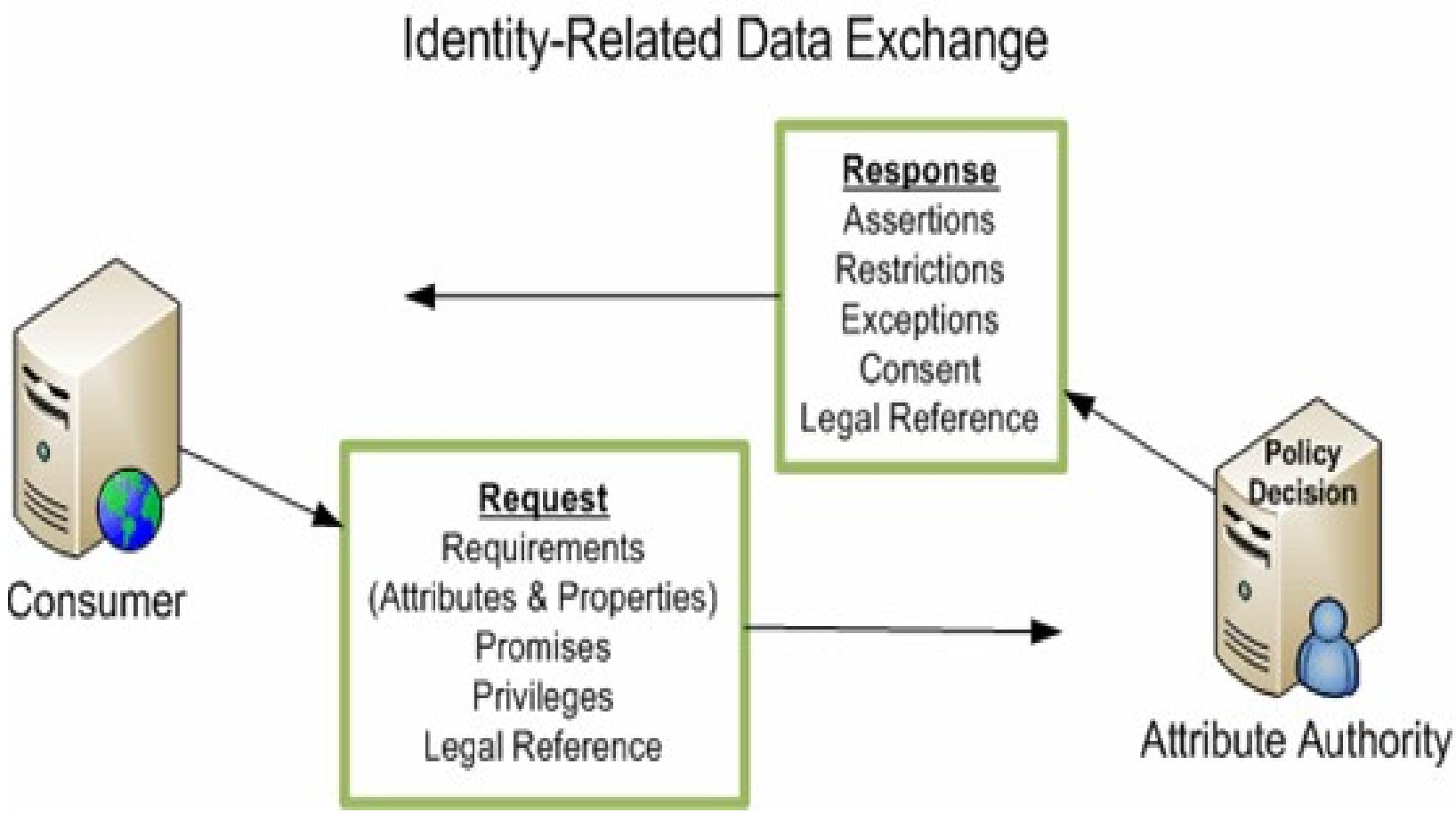
Agenda

- Reputation Data explained
- Need for ID Governance for ORMS
- ID Assurance for ORMS
- XDI and XACML/AAPML
- Policy based Reputation Context

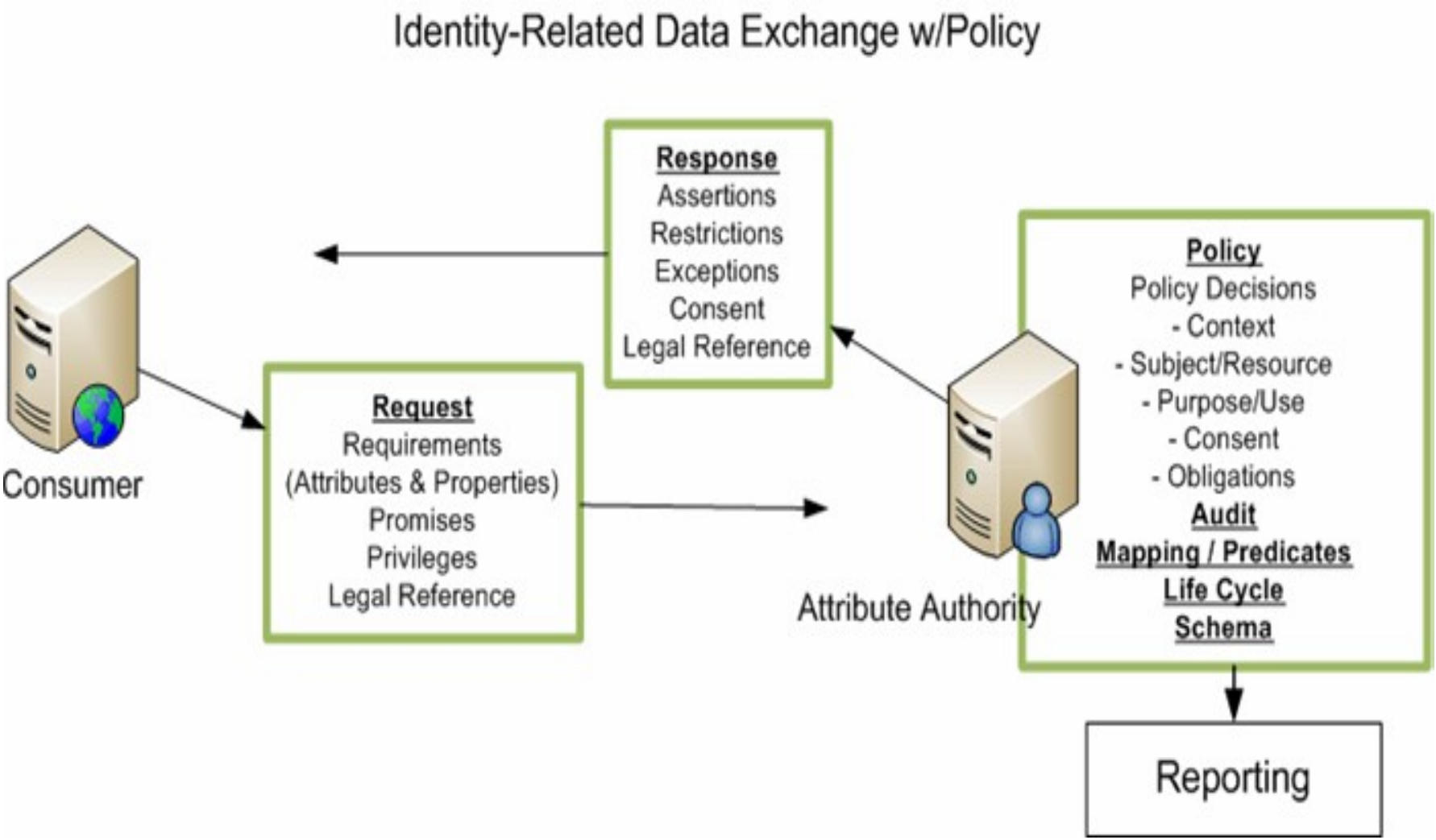
Reputation

- Reputation: A specific characteristic or trait ascribed to a person or thing
- Security Sensitive (stakeholder owning and managing the data- TRW for Credit Reports, External Entity rating Doctors, etc.)
- Reputation aids in Rating Services (credit rating, lawyer rating, device rating, etc.)
- Needs a Refutation process (by the entity concerned)
- Needs to align with Privacy and other concerns
- Common Representation and Interpretation of Scores
- XDI - a single pointer for multiple reputation data sources

Liberty Alliance Identity Governance



Liberty Alliance Identity Governance



Liberty Alliance Identity Governance

- Declarative Syntax
- Which client may specify attribute requirements
- Providers of Identity related Attributes to express policy on the usage of information
- Align with Privacy and other concerns
- PPEL (preference expression language)
  - Strict
  - Cautious
  - Moderate
  - Flexible
  - Casual

Identity Assurance and ORMS

- IA Levels
  - Little of No confidence
  - Some confidence
  - High confdence
  - Very High confidence
- IA Level's aligned to Impact Levels

Identity Assurance and ORMS

- Mapping IA level required when sharing – Reputation Data – trivial reputation data, security sensitive reputation data, etc.
- Financial Reputation for Loan might require higher levels of IA, etc.
- Reputation Data can lead to higher or lower IA levels.
- Assurance and Reputation – critical to managing Identity Lifecyles
- Reputation Data and IA level linked to Specific Services and COT that deliver such services

XDI and XACML

- Integration of Policy based and Reputation based -Trust Systems
  - Policy based -structured organisation environment
  - Reputation based – unstructured user community
  - Reputation-based Trust can be formalized by relations between Trustors, Trustees, Actions and Trust Levels (policies)
  - The two combined improves Trust Management
    - (paper from Europe in 2006)
- Potential Synergies -
  - XDI mapping to multiple Reputation Data Sets
  - XACML mapping to multiple Data Specific policies

Policies based Reputation Context

- Policy Orchestration and ABAC – leverages reputation context for the delivery of contextual services
  - Driving Record and Ratings -for a given individual and specific periodic analysis of matching automobile insurance providers – may be offered as a service -that complies with privacy preference policies of defined by the user.
  - Doctor Reputation and Rating -for a given city and specific area of specialization (autism and PDD) – may be offered as a service -that complies with the NAB (national autism board) with (policies for) the ratings validated by parents of autistic kids in area

Thank You !!!

rakesh.radhakrishnan@sun.com

http://www.network-identity.com

A Content-Driven Access Control System

Jessica Staddon[1] Philippe Golle[1] Martin Gagné[2*] Paul Rasmussen[1]

[1]Palo Alto Research Center
{staddon, pgolle, rasmussen}@parc.com

[2]University of California at Davis
gagne@cs.ucdavis.edu

ABSTRACT

Protecting identity in the Internet age requires the ability to go beyond the identification of explicitly identifying information like social security numbers, to also find the broadly-held attributes that, when taken together, are identifying. We present a system that can work in conjunction with natural language processing algorithms or user-generated tags, to protect identifying attributes in text. The system uses a new attribute-based encryption protocol to control access to such identifying attributes and thus protects identity. The system supports the definition of user access rights based on role or identity. We extend the existing model of attribute-based encryption to support threshold access rights and provide a heuristic instantiation of revocation.

Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption

General Terms

Security

Keywords

Access control, attribute-based encryption, secret sharing, inference control, revocation.

1. INTRODUCTION

Identity protection is about more than protecting social security numbers, passport numbers and other values assigned for the explicit purpose of identification. Recent research has shown that even broadly held attributes, when taken together, can be identifying. For example, [18, 27] demonstrate that the triple of gender, zip code and date of birth is unique to a large fraction of individuals in the U.S. population and [28] shows that even characteristics like profession and nationality can be identifying. Hence, identity protection requires the ability to control access to an individual's attributes, even those attributes an individual may share with many others.

We propose a system that protects identity and other sensitive information by controlling access to an individual's attributes through encryption. Attributes can be extracted from text with natural language processing tools such as [17] or through the manual association of attributes by user tagging. Inference detection algorithms such as [28] can determine what sets of attributes allow sensitive information, such as identity, to be inferred. Attributes can be of varied granularity. For example, we might want to group all social security numbers into an "SSN attribute" since it is likely that any SSN should be protected. In contrast, we may not need to protect all names, so it might make sense to represent each name as its own attribute and only control access to the ones deemed sensitive.

Once attributes have been identified, our system uses a novel protocol for attribute-based encryption to offer protection against sensitive inferences based on those attributes. Passages in unstructured documents and fields in forms are encrypted based on their attributes, and users are assigned a decryption key according to the attributes they are allowed to access. The encrypted document can then be released and shared publicly with the assurance that no one will be able to recover identifying, or otherwise sensitive, information, without the proper access rights (in the form of a decryption key).

With this ability to define fine-grained access rights based on document content, the system can ensure users are only able to access information about certain individuals. For example, a user authorized to view articles about a particular former CIA agent Valerie Plame Wilson, can be given the capability to decrypt any article provided it mentions at least three of the following four organizations, as the presence of any three is strong evidence that the text is about Valerie Wilson: Brewster, Jennings and Associates (a CIA front company), London School of Economics (one graduate school she attended), College of Europe (another graduate school she attended) and the CIA (her employer).

Alternatively, the system can protect all identities by preventing the user from decrypting enough information to infer identity. For example, an employee of a mortgage company who arranges property appraisals might be given a decryption key that allows them to view the property address and a contact number for the applicant, but not identifying infor-

*This work was done while interning at the Palo Alto Research Center.

IDtrust '08, March 4-6, 2008 Gaithersburg, MD.

mation like the applicant's social security number or enough demographic attributes to infer identity.

We have implemented a research prototype of our system. Our implementation is based on interviews conducted with professionals who routinely deal with sensitive data in their work. The implementation allows the system administrator to define access rights for different users of the system by the user's identity or role. However, it goes beyond traditional role-based access control (RBAC) systems (see, for example, [14]) by tying enforcement of those access rights to document content, as opposed to the categorization of a document. For example, while with RBAC it may be natural to block access to all performance appraisals, with our system it is easier to distinguish between Bob's access to Mary's performance appraisal and Bob's access to his own performance appraisal. In addition, our system provides an efficient approach to protecting sensitive content because it allows a single (encrypted) version of a document to be circulated within an organization while ensuring that each user will only be able to view the portions of the document they are authorized to see.

Overview. This paper is organized as follows. In section 2, we survey related work. In section 3 we describe various use cases for our technology based on field interviews we conducted. We describe our model in section 4. Our novel attribute-based encryption protocol is presented in section 5 and our prototype system is discussed in section 6. Our model of attribute-based encryption with revocation and our heuristic construction are in section 7. We conclude in section 8.

2. RELATED WORK

Our system is related to work in a number of different areas, each of which we discuss in turn.

Attribute-Based Encryption. Our encryption protocols are forms of attribute-based encryption. The notion of attribute-based encryption (ABE) is introduced by Sahai and Waters in [25]. In follow-on work, Goyal, Pandey, Sahai and Waters [19] extend the model of [25] to accommodate general monotone access rights and provide a novel secret sharing-based scheme for supporting general access rights whose security relies on standard BDDH.

We further extend the model of [19] by allowing for user revocation. In addition, our use of ABE for identity protection, and other undesired inferences stemming from document content, is new. Our protocols are designed with this application in mind. In particular, in section 5.1 we present a scheme that leverages an understanding of topic detection algorithms in the natural language processing community to control access to text about a targeted individual.

Revocation and Broadcast Encryption. Revocation schemes and broadcast encryption schemes exist for both the symmetric key and public key settings (see, for example, [16, 23, 12, 8]). In either setting, the goal is to distribute keys to users so that given any target subset of users, it is possible to encrypt a message so that exactly those users can decrypt the message. In our setting, users have previously assigned access rights and based on these, it may not be necessary to revoke arbitrary subsets of users. The natural adaption of revocation and broadcast encryption schemes to our setting leads to high overhead. In particular, if we replace a user in the revocation scheme with a conjunction of attributes, and users store the keys assigned by the revocation scheme to each of their conjunctions, this may lead to high communication overhead ($O(2^m)$, where m is the total number of attributes).

Commercial Offerings. There are a number of commercially available products that provide identity protection by finding, and optionally, protecting, identifying information in documents. The vast majority of these focus on pattern matching explicitly identifying information like social security numbers and account numbers (see, for example, [29, 24]). The product offered by MortgageGrader [21] goes a step further by protecting race and other attributes to protect against discrimination, not identification. The only commercial technology of which we are aware with even the potential to protect against identification through the association with broadly held attributes is the technology of AreteQ [1]. However, their main application appears to be protecting sensitive military information such as weapons development and they do not appear to offer encryption as a mode of protection.

3. USE CASES

Prior to developing our encryption algorithms and research prototype, we conducted interviews with financial sector and legal professionals who routinely deal with sensitive data. These interviews suggested a number of use cases for content-driven access control that we describe now in turn.

Mortgage Lending. In order to comply with privacy legislation (e.g. the Gramm-Leach-Bliley Act) and to support separation of duty mandates in Sarbanes-Oxley, lending companies need to decompose processing of financial information into several roles and control the financial information that is accessible by employees based on their role. For example, a lender might provide access to a loan application for the purposes of evaluating the credit of the applicant and appraising the property for which the loan is sought. The credit evaluator needs identifying information such as the applicant's social security information and name, but does not need the property address, whereas the appraiser may only need the address. With our system, the application can be encrypted once and the different keys assigned to the credit evaluator and the appraiser ensure that each person will be able to view only the information they need.

Mergers and Acquisitions. When a company is interested in being merged or acquired, it needs to make its financial data available to potential "suitors" for review. The current approach to this involves the construction of a virtual "dataroom" which houses all the relevant records and to which suitors are given access. The difficulty is that it is desirable to parameterize the extent of the access according to the stage of the negotiations. A new suitor may be only given coarse financial information whereas a suitor who is deemed to be more serious might be allowed to see names of clients and more detailed records. To achieve this fine-grained access control today it is necessary to sanitize the documents and records differently for the various suitors. With our system, the data can be encrypted once, and the suitor's decryption key determines what data they can access.

LEGAL. In the United States there is a broad trend toward storing court records online. These records contain a wide array of identifying information including social security numbers, names, and addresses, some of which may be redacted upon request (see, for example, Florida statutes [15]). A potentially more efficient approach would be to encrypt the identifying fields and passages and ensure through the decryption keys that only authorized users can access the identifying content in the online record.

4. MODEL

As discussed in section 1, we present encryption schemes that take as input documents (or document regions) on which automated or manual content analysis has been performed to generate tags that reflect the meaning of the document. For example, the tags may be keywords in the document, or metadata associated with the document such as the name of the document's author or the date the document was created. Following the terminology introduced by Sahai and Waters in [25],we refer to these tags as *attributes*, and our encryption protocol as an attribute-based encryption (ABE) protocol.

Let $\mathcal{W} = \{w_1, \ldots, w_m\}$ represent the set of all attributes. These attributes are boolean variables that take the value true in documents or portions of documents associated with the attribute, and take the value false otherwise. For example, let w_i represent the attribute 'John Smith'. The variable w_i takes the value true in portions of documents that involve 'John Smith', and false everywhere else.

Access rights to a document can be represented formally as a monotone boolean formula over the variables $w_1, \ldots, w_m$ and their negations $\overline{w_1}, \ldots, \overline{w_m}$. (In what follows, we denote the boolean operator AND by $\wedge$ and OR by $\vee$). A user can access all documents or portions of documents whose assignment of values to the boolean variables $w_1, \ldots, w_m$ satisfy the boolean formula that expresses the user's access rights. For example, if a user has rights to all documents that contain both the attributes w_1 and w_2, but not w_3, this is represented by the formula: $w_1 \wedge w_2 \wedge \overline{w_3}$.

More generally, we represent the access rights of a user in disjunctive normal form (DNF), i.e. as a disjunction of conjunctions of the variables $w_1, \ldots, w_m$ and their negations. We refer to the conjunctions that make up the disjunctive formula as "subformulas" of the user's access rights. For example, the conjunction $(w_1 \wedge w_2)$ is a subformula of the access right $(w_1 \wedge w_2) \vee (w_1 \wedge w_3) \vee (w_4)$.

Let $T \subset \mathcal{W}$, and let σ be a boolean formula over the variables $w_1, \ldots, w_m$ and their negations. We say that T *satisfies* σ if σ is satisfied when the variables in T are set to true and the variables in the complement of T are set to false.

DEFINITION 4.1. *An ABE scheme consists of four algorithms*[1]*:*

Setup$(\lambda, \mathcal{W})$ *takes as input a security parameter λ and a set of attributes $\mathcal{W} = \{w_1, \ldots, w_m\}$, and outputs the public parameters and a master key, (pub, mk). The set $\mathcal{W}$ must be part of the public parameters.*

KeyGen(mk, σ) *takes as input a master key mk and a monotone boolean formula σ over the variables $w_1, \ldots, w_m$ and their negations $\overline{w_1}, \ldots, \overline{w_m}$, and outputs secret parameters d_σ.*

Encrypt(mk, M, T) *takes as input a master key mk, a message M, and a subset $T \subset \mathcal{W}$ of the attributes that take the value* true *in the message M, and outputs a ciphertext C.*

Decrypt(T, d_σ, C) *takes as input a set of attributes $T \subseteq \mathcal{W}$, a secret key d_σ corresponding to a boolean formula σ and a ciphertext C, and outputs a message or a special symbol $\perp$.*

such that if T satisfies σ, then we have

$$\textbf{Decrypt}(T, d_\sigma, \textbf{Encrypt}(mk, M, T)) = M.$$

Informally, an ABE scheme is secure if a user can only decrypt portions of documents that satisfy the boolean formula that describes the access rights of the user. Formally, we define the security of an ABE scheme through the following game between an adversary and a challenger:

Setup: the challenger runs the **Setup** algorithm, gives the public parameters to the adversary and keeps the master key to himself.

Phase 1: the adversary adaptively issues queries $\sigma_1, \ldots, \sigma_m$, where each σ_i is a boolean formula over the variables in $\mathcal{W}$ and their negations. The challenger responds by running the **KeyGen** algorithm and gives the secret key d_{σ_i} corresponding to σ_i to the adversary. In addition, the adversary adaptively issues encryption requests for a message M and attribute set T. The challenger responds by running **Encrypt** and gives the resulting ciphertext C to the adversary.

Challenge: the adversary outputs two equal length messages M_0 and M_1, and a subset T^* of $\mathcal{W}$ such that T^* satisfies none of the boolean formulas σ_i issued in Phase 1. The challenger picks a bit $b \xleftarrow{R} \{0, 1\}$, encrypts $C \leftarrow \textbf{Encrypt}(mk, M_b, T)$ and outputs C.

Phase 2: the adversary adaptively issues queries $\sigma_{m+1}, \ldots, \sigma_n$ such that T^* does not satisfy σ_i for $m + 1 \leq i \leq n$. The challenger answers these queries as in Phase 1.

Guess: the adversary outputs a bit b'.

We define the advantage of the adversary in attacking the scheme as

$$Adv_A(\lambda) = \left| Pr[b = b'] - \frac{1}{2} \right|$$

The ABE scheme is secure if the adversary's advantage is negligible. In this paper, we prove the security of the ABE scheme proposed in section 5 in the selective-set model (following [9, 10, 6]). Specifically, we assume that the set T^* is given by the adversary at the beginning of the game, before he receives the public key. We note that schemes secure against selective attacks are also secure against adaptive attacks, with a loss of 2^m (where m is the number of keywords) in the efficiency of the reduction.

The proof of security of our ABE scheme is based on the Bilinear Decisional Diffie-Hellman (BDDH) assumption. We

[1]A similar definition was independently proposed in [19].

briefly describe BDDH here, referring the reader to [7] for additional information.

Bilinear Decisional Diffie-Hellman Let G_1 and G_2 be groups of prime order q, with an admissible bilinear map (see [7]) $\hat{e} : G_1 \times G_1 \rightarrow G_2$, and let g be a generator of G_1 and $h = \hat{e}(g, g)$. The BDDH problem is to distinguish 4-tuples of the form (g^a, g^b, g^c, h^{abc}) and (g^a, g^b, g^c, h^d), where a, b, c, d are random elements of the set $\{1, \ldots, q-1\}$. We say that a polynomial time adversary $\mathcal{A}$ has advantage ϵ in solving the BDDH problem if $|Pr[\mathcal{A}(g^a, g^b, g^c, h^{abc}) = \mathsf{true}] - Pr[\mathcal{A}(g^a, g^b, g^c, h^d) = \mathsf{true}]| > \epsilon$.

5. AN ABE SCHEME BASED ON SECRET SHARING

To leverage secret sharing (see, for example [26]) to achieve ABE, we choose a master secret and associate shares of the secret with each of the attributes that make up the user's access right. The shares are user-specific to prevent unauthorized users gaining access to documents through collusion, and are stored in encrypted form to ensure that they can only be accessed when a document with the appropriate attributes is received. If a document has a set of attributes that satisfies a user's access rights, then the user is able to reconstruct the document encryption key (a function of the master secret), and thus, decrypt the document. A key point is that as part of the decryption process the shares are randomized to be specific to the particular document, so that the ability to access one document doesn't imply the ability to access others. To summarize, this construction takes as input an arbitrary secret sharing scheme that realizes a user's access rights and consists of the following two components:

- Generate shares of a master secret: Associate with each literal in a Boolean formula, σ, that describes the user's access rights, a share of the master secret, where the sets of shares capable of reconstructing master secret is as indicated by σ.
- Encrypt shares based on their corresponding attribute: Each share, s, is associated with a literal, W_s, such that W_s must be present in a document, or document region, in order for the user to be able to recover a randomized version of s.

To demonstrate this construction yields a secure ABE, we provide an instantiation of this scheme using the secret sharing scheme of [2] and prove it secure using the BDDH assumption. Following this we introduce an example using another approach to secret sharing that reduces storage for certain access rights.

In this instantiation, the user access rights are described by boolean formulas σ on the attributes, represented by a rooted tree[2] in which each internal node is either an AND or an OR gate, and the leaves are keywords. We say that a leaf w_i is satisfied by a set of attributes T if $w_i \in T$, an AND node is satisfied if all its children are satisfied and an OR node is satisfied if one of its children is satisfied. The tree σ is satisfied if the root is satisfied. If user U is given access σ_U, then he should be able to read every document D whose set of attributes T_D satisfies σ_U.

[2]We often abuse notation and denote the tree by σ, as well.

Setup$(\lambda, \mathcal{W})$.

Say $\mathcal{W} = \{w_1, \ldots, w_m\}$. Let $\mathbb{G}$ and $\mathbb{G}'$ be groups and let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}'$ be an admissible bilinear map (see [7]). Select a random generator $g \in \mathbb{G}$ and a random integer $S \in \{0, \ldots, |\mathbb{G}| - 1\}$. Compute $h_0 = e(g, g)^S$, and select $g_1, \ldots, g_m \overset{R}{\leftarrow} \mathbb{G}$. The values $g_1, \ldots, g_m$ are associated with attributes $w_1, \ldots, w_m$. The public key is $(\mathbb{G}, g, h_0, g_1, \ldots, g_m)$ and the master key is $mk = S$.

KeyGen(mk, σ).

First assign the secret value S to the root of the tree. Then, values are assigned to all the nodes in the tree recursively as follows:

- if an OR gate is assigned secret value s, assign the secret value s to all its children.
- if an AND gate with k children is assigned secret value s, generate $k-1$ random values $s_1, \ldots, s_{k-1}$ in the set $\{1, \ldots, |\mathbb{G}| - 1\}$ and set $s_k = s - \sum_{i=1}^{k-1} s_i \bmod |\mathbb{G}|$ and assign a secret value s_i to each child.

When this is done, a key is associated with each leaf of the tree: a leaf with keyword w_i assigned secret value s is associated with a key $(d_0 = g^r, d_1 = g^s \cdot g_i^r)$ where $r \overset{R}{\leftarrow} \{1, \ldots, |\mathbb{G}| - 1\}$ (different r for each leaf). The secret key d_σ associated with σ is the set of secret keys associated with all the leaves of σ.

Encrypt(mk, T, M).

Select $r' \overset{R}{\leftarrow} \{0, \ldots, |\mathbb{G}| - 1\}$.
Compute $u = g^{r'}$, $v_j = g_j^{r'}$ for $w_j \in T$ and $w = h_0^{r'} \cdot M$.
Return $C = (u, \{v_j\}_{w_j \in T}, w)$.

Decrypt(T, d_σ, C).

Say $C = (u, \{v_i\}_{w_i \in T}, w)$.
For each leaf in σ that is satisfied by T, associate $h = e(u, d_1) \cdot e(v_i, d_0)^{-1}$ to the leaf, where (d_0, d_1) and w_i are respectively the secret key and the keyword associated with the leaf (note that $e(u, d_1) \cdot e(v_i, d_0)^{-1} = e(g, g)^{rs}$ where $r = \log_g u$ and s is the value associated with the leaf by the `KeyGen` algorithm). Then, associate a group element with each node in σ that is satisfied by T in a bottom-up fashion as follows:

- if h is associated with one child of an OR node, associate h with the OR node as well.
- if $h_1, \ldots, h_k$ are associated with each of the k children of an AND node, associate $h = \prod_{i=1}^{k} h_i$ to the AND node.

At the end of this process, the value $h = e(g, g)^{rS} = h_0^r$ will be associated with the root of σ (where $r = \log_g u$). We can then compute $M = w \cdot h^{-1}$.

Theorem 5.1. *If the Bilinear Decision Diffie-Hellman problem is hard, then the scheme above is a secure ABE scheme in the selective-set model.*

Proof. Let $\mathcal{A}$ be an adversary that can obtain advantage ϵ against the scheme. We show how to use this adversary to build an adversary $\mathcal{B}$ that breaks the Bilinear Decision Diffie-Hellman problem.

Given $(g, A = g^{\alpha}, B = g^{\beta}, C = g^{\gamma}, Z)$, algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ on a set of attributes S. $\mathcal{A}$ then outputs a set of attributes T^* that he wishes to attack.

Algorithm $\mathcal{B}$ produces the public key as follows:
Select $\omega_1, \ldots, \omega_m$ and ρ at random in $\{1, \ldots, |\mathbb{G}| - 1\}$.
For each $w_i \in T^*$, compute $g_i = g^{\omega_i}$.
For each $w_i \in S \setminus T^*$, compute $g_i = B \cdot g^{\omega_i}$.
Compute $h_0 = e(g, g)^{\rho} \cdot e(A, B)^{-1}$ (so, implicitly, $S = \rho - \alpha\beta$).

Algorithm $\mathcal{B}$ then runs algorithm $\mathcal{A}$ with the public key it just produced, and answers $\mathcal{A}$'s queries as follows:

– **Encryption queries.** Those can be answered simply by running the encryption algorithm with the public key.

– **KeyGen queries.** Say $\mathcal{A}$ issues a key extraction for the formula σ. Note that T^* cannot satisfy σ, otherwise the query would not be allowed. $\mathcal{B}$ can compute the secret key corresponding to σ as follows.
First, assign the temporary key (A, g^{ρ}) to the root, which 'implicitly' assign the secret value S to the root. Remember that $\rho = S + \alpha\beta$. We show how $\mathcal{B}$ can compute the secret keys of all the leaves of a tree rooted at a node that is not satisfied by T^* using only an 'implicit' description of its secret value s through $(\hat{d}_0 = g^{\alpha}, \hat{d}_1 = g^{s+\alpha\beta})$:

- if the tree is rooted at an OR node, then all its children are unsatisfied by T^*, so we can implicitly give them the secret value s by assigning the temporary key $(\hat{d}_0, \hat{d}_1)$ to each of them and computing the secret keys for each recursively.

- if the tree is rooted at an AND node with k children, then at least one of its child c is not satisfied by T^*. Select $k-1$ random values $s_1, \ldots, s_{k-1} \xleftarrow{R} \{1, \ldots, |\mathbb{G}|-1\}$ and assign a secret value s_i to each child except for c. The secret keys for the subtree rooted at these nodes can now be computed using the `KeyGen` algorithm. Then, implicitly pass the secret value $s - \sum_{i=1}^{k-1}$ to c by assigning it the temporary key $(\hat{d}_0, \hat{d}_1 \cdot \prod_{i=1}^{k-1} g^{-s_i})$ and the secret keys for the tree rooted at c are computed recursively.

- if the tree is rooted at a leaf with keyword w_i, then by design, $w_i \notin T^*$ because the tree is rooted at an unsatisfied node. Select a random value $\hat{r} \in \{1, \ldots, |\mathbb{G}| - 1\}$ and put $r = \hat{r} + \alpha$. Then the secret key associated to this leaf is $(g^r, g^{s+rw_i}) = (A \cdot g^{\hat{r}}, \hat{d}_1 \cdot A^{\omega_i} \cdot B^{\hat{r}} \cdot g^{\hat{r}\omega_i})$.

– **Challenge query.** When $\mathcal{A}$ issues two messages M_0, M_1 on which he wishes to be tested, $\mathcal{B}$ first selects a random bit $b \in \{0, 1\}$, computes the ciphertext $(C, \{C^{\omega_i}\}_{w_i \in T^*}, (e(C, g)^{\rho} \cdot Z^{-1}) \cdot M_b)$ and returns it to $\mathcal{A}$. Note that $e(C, g)^{\rho} \cdot Z^{-1} = e(g, g)^{\gamma S} e(g, g)^{\alpha\beta\gamma} Z^{-1}$ so if (g, A, B, C, Z) is a Bilinear Diffie-Hellman tuple, this is just $e(g, g)^{\gamma S} = h_0^{\gamma}$ as it should be, otherwise, it's just a random element in the group.

Eventually, algorithm $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{B}$ outputs 1, otherwise, $\mathcal{B}$ outputs 0.

If (g, A, B, C, Z) was a Bilinear Diffie-Hellman tuple, then $\mathcal{A}$ will guess the correct bit with probability $1/2 + \epsilon$, otherwise, the distribution of $(e(C, g)^{\rho} \cdot Z^{-1}) \cdot M_b$ is independent of M_b, therefore, the adversary cannot guess the correct bit with probability more than $1/2$. Thus, $\mathcal{B}$ correctly determines whether (g, A, B, C, Z) is a valid Bilinear Diffie-Hellman tuple with probability $1/2 + \epsilon/2$. □

5.1 A Variant for Threshold Access Rights

As discussed earlier, our ABE protocol leverages automated content analysis in order to achieve content-driven access control. In high security applications such as government, it is important to have control over the number of "false positives" output by the content analysis. For example, if a government analyst is granted access to all documents about "Karl Rove" and the "leak", it may be important to ensure that the rights the analyst receives don't inadvertently allow access to other documents about Rove, or documents pertaining to other leaks. A common approach to topic detection in content analysis is to identify a set of keywords corresponding to a topic [11]. To reduce false positives, the user will receive "threshold" access rights that require ℓ out of k keywords to be present in order for the user to be able to decrypt the document. We briefly present a variant of the previous scheme that uses a different secret sharing mechanism in order to reduce user storage for threshold access rights.

Setup$(\lambda, \mathcal{W})$.
The same as before.

KeyGen(mk, σ).
Let S be the master secret, and let $W_1, \ldots, W_k$ represent the attributes in σ. Recall that user U has access to any document or document region with ℓ attributes in the set, $W_1, \ldots, W_k$. Let $q(x)$ be a polynomial of degree $\ell - 1$ over $\{1, \ldots, |\mathbb{G}| - 1\}$ and let $a_1, \ldots, a_{\ell} \in \{1, \ldots, |\mathbb{G}| - 1\}$ be distinct elements. Finally, let $r \xleftarrow{R} \{1, \ldots, |\mathbb{G}| - 1\}$. The secret key, d_{σ}, is $g^r, g^{q(a_1)} \cdot g_1^r, \ldots, g^{q(a_k)} \cdot g_k^r$, where g_i, $i = 1, \ldots, k$ are defined as before.

Encrypt(mk, T, M).
The same as before.

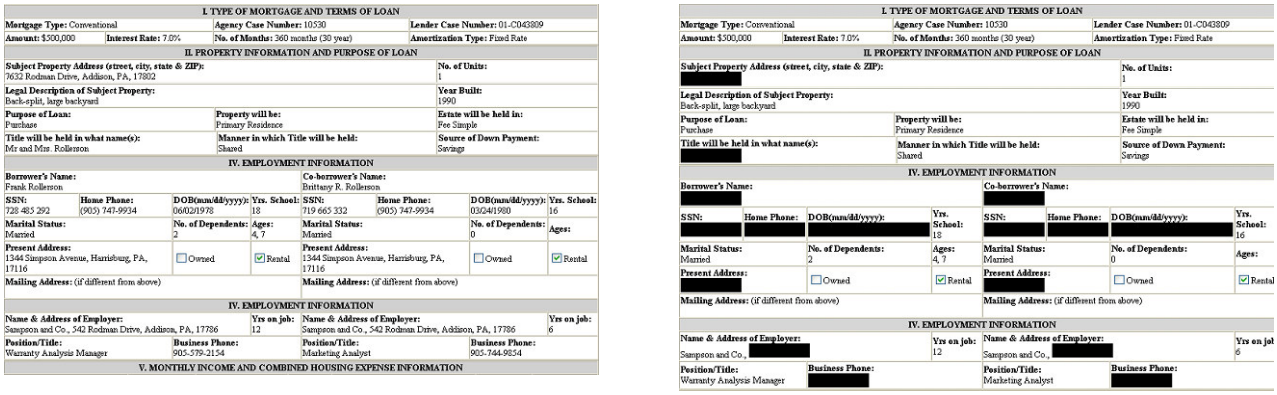
Decrypt(T, d_{σ}, C).
If $W_i \in \{W_1, \ldots, W_k\} \cap T$, the user computes $e(g^{q(a_i)} \cdot g_i^r, g^{r'})/e(g_i^{r'}, g^r) = h^{r' q(a_i)}$. If $\{W_1, \ldots, W_k\} \cap T$ has at least ℓ elements, the user recovers $h^{r'S}$ using polynomial interpolation.

This variant improves on the efficiency of the general construction in that user storage is on the order of the number of attributes in σ (as opposed to the number of subformulas in σ). The proof of security can be adapted to this variant.

Finally, we note that a novel secret sharing-based scheme for threshold access rights appears in [25]. The mechanics of our scheme are a bit different from theirs and these differences allow the above variant to be easily extended to permit user revocation as described in section 7.

6. PROTOTYPE SYSTEM

We have implemented a prototype of our content-driven access control system in Java. Our system assumes that document regions (e.g. paragraphs or fields in forms) are tagged according to the attributes present (e.g. names, phone numbers, etc.). The tags indicate the attributes present and



I. TYPE OF MORTGAGE AND TERMS OF LOAN

Mortgage Type: Conventional | Agency Case Number: 10530 | Lender Case Number: 01-C043809
Amount: $300,000 | Interest Rate: 7.0% | No. of Months: 360 months (30 year) | Amortization Type: Fixed Rate

II. PROPERTY INFORMATION AND PURPOSE OF LOAN

Subject Property Address (street, city, state & ZIP): 7632 Rodman Drive, Addison, PA, 17802 | No. of Units: 1
Legal Description of Subject Property: Back-split, large backyard | Year Built: 1990
Purpose of Loan: Purchase | Property will be: Primary Residence | Estate will be held in: Fee Simple
Title will be held in what name(s): Mr and Mrs. Rollerson | Manner in which Title will be held: Shared | Source of Down Payment: Savings

IV. EMPLOYMENT INFORMATION

Borrower's Name: Frank Rollerson				Co-borrower's Name: Brittany R. Rollerson			
SSN: 728 485 292	Home Phone: (905) 747-9934	DOB(mm/dd/yyyy): 06/02/1978	Yrs. School: 18	SSN: 719 665 332	Home Phone: (905) 747-9934	DOB(mm/dd/yyyy): 03/24/1980	Yrs. School: 16
Marital Status: Married		No. of Dependents: 2	Ages: 4, 7	Marital Status: Married		No. of Dependents: 0	Ages:
Present Address: 1344 Simpson Avenue, Harrisburg, PA, 17116		☐ Owned	☑ Rental	Present Address: 1344 Simpson Avenue, Harrisburg, PA, 17116		☐ Owned	☑ Rental
Mailing Address: (if different from above)				Mailing Address: (if different from above)			

IV. EMPLOYMENT INFORMATION

Name & Address of Employer: Sampson and Co., 542 Rodman Drive, Addison, PA, 17786	Yrs on job: 12	Name & Address of Employer: Sampson and Co., 542 Rodman Drive, Addison, PA, 17786	Yrs on job: 6
Position/Title: Warranty Analysis Manager	Business Phone: 905-579-2154	Position/Title: Marketing Analyst	Business Phone: 905-744-9854

V. MONTHLY INCOME AND COMBINED HOUSING EXPENSE INFORMATION

I. TYPE OF MORTGAGE AND TERMS OF LOAN

Mortgage Type: Conventional | Agency Case Number: 10530 | Lender Case Number: 01-C043809
Amount: $300,000 | Interest Rate: 7.0% | No. of Months: 360 months (30 year) | Amortization Type: Fixed Rate

II. PROPERTY INFORMATION AND PURPOSE OF LOAN

Subject Property Address (street, city, state & ZIP): [redacted] | No. of Units: 1
Legal Description of Subject Property: Back-split, large backyard | Year Built: 1990
Purpose of Loan: Purchase | Property will be: Primary Residence | Estate will be held in: Fee Simple
Title will be held in what name(s): [redacted] | Manner in which Title will be held: Shared | Source of Down Payment: Savings

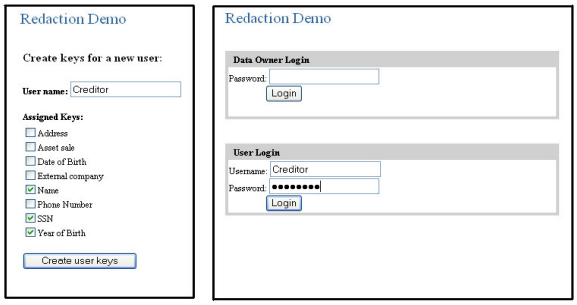
IV. EMPLOYMENT INFORMATION

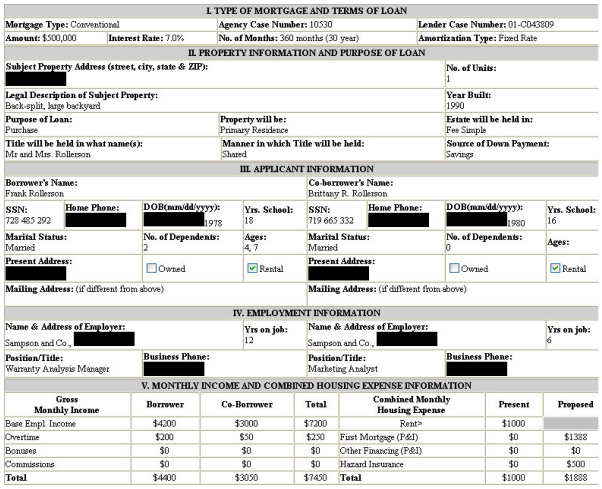
Borrower's Name: [redacted]				Co-borrower's Name: [redacted]			
SSN: [redacted]	Home Phone: [redacted]	DOB(mm/dd/yyyy): [redacted]	Yrs. School: 18	SSN: [redacted]	Home Phone: [redacted]	DOB(mm/dd/yyyy): [redacted]	Yrs. School: 16
Marital Status: Married		No. of Dependents: 2	Ages: 4, 7	Marital Status: Married		No. of Dependents: 0	Ages:
Present Address: [redacted]		☐ Owned	☑ Rental	Present Address: [redacted]		☐ Owned	☑ Rental
Mailing Address: (if different from above)				Mailing Address: (if different from above)			

IV. EMPLOYMENT INFORMATION

Name & Address of Employer: Sampson and Co., [redacted]	Yrs on job: 12	Name & Address of Employer: Sampson and Co., [redacted]	Yrs on job: 6
Position/Title: Warranty Analysis Manager	Business Phone: [redacted]	Position/Title: Marketing Analyst	Business Phone: [redacted]

Figure 1: These two screenshots from our prototype system show an unencrypted mortgage application on the left and the same document, with ciphertexts represented by black boxes, on the right. The ciphertexts are generated with our attribute-based encryption scheme, using the attributes, address, name, SSN, phone number and date of birth.





I. TYPE OF MORTGAGE AND TERMS OF LOAN

Mortgage Type: Conventional | Agency Case Number: 10530 | Lender Case Number: 01-C043809
Amount: $300,000 | Interest Rate: 7.0% | No. of Months: 360 months (30 year) | Amortization Type: Fixed Rate

II. PROPERTY INFORMATION AND PURPOSE OF LOAN

Subject Property Address (street, city, state & ZIP): [redacted] | No. of Units: 1
Legal Description of Subject Property: Back-split, large backyard | Year Built: 1990
Purpose of Loan: Purchase | Property will be: Primary Residence | Estate will be held in: Fee Simple
Title will be held in what name(s): Mr and Mrs. Rollerson | Manner in which Title will be held: Shared | Source of Down Payment: Savings

III. APPLICANT INFORMATION

Borrower's Name: Frank Rollerson				Co-borrower's Name: Brittany R. Rollerson			
SSN: 728 485 292	Home Phone: [redacted]	DOB(mm/dd/yyyy): [redacted]1978	Yrs. School: 18	SSN: 719 665 332	Home Phone: [redacted]	DOB(mm/dd/yyyy): [redacted]1980	Yrs. School: 16
Marital Status: Married		No. of Dependents: 2	Ages: 4, 7	Marital Status: Married		No. of Dependents: 0	Ages:
Present Address: [redacted]		☐ Owned	☑ Rental	Present Address: [redacted]		☐ Owned	☑ Rental
Mailing Address: (if different from above)				Mailing Address: (if different from above)			

IV. EMPLOYMENT INFORMATION

Name & Address of Employer: Sampson and Co., [redacted]	Yrs on job: 12	Name & Address of Employer: Sampson and Co., [redacted]	Yrs on job: 6
Position/Title: Warranty Analysis Manager	Business Phone: [redacted]	Position/Title: Marketing Analyst	Business Phone: [redacted]

V. MONTHLY INCOME AND COMBINED HOUSING EXPENSE INFORMATION

Gross Monthly Income	Borrower	Co-Borrower	Total	Combined Monthly Housing Expense	Present	Proposed
Base Empl. Income	$4200	$3000	$7200	Rent>	$1000	
Overtime	$200	$50	$250	First Mortgage (P&I)	$0	$1388
Bonuses	$0	$0	$0	Other Financing (P&I)	$0	$0
Commissions	$0	$0	$0	Hazard Insurance	$0	$500
Total	$4400	$3050	$7450	Total	$1000	$1888

Figure 2: The left side of this figure shows the administrator window for defining the Creditor's access rights at the top and the Creditor logging into the system at the bottom. The right side of the figure shows the mortgage application as it appears to the creditor, the creditor can view SSNs and years of birth, but other sensitive information remains encrypted.

their precise location in the document. Location is important because the user interface of the prototype represents the encrypted regions as black bars. As much as possible, we size the black bars in a manner that is independent of the length of the text they represent, to resist attacks such as those described in [20].

Since our focus is on content protection, we will not discuss the creation of tags in detail, but we note that there exist well-known tools for identifying and tagging sensitive information in documents. For example, sensitive content may be tagged with tools for extracting patterns and entities automatically from documents [17], in combination with user interface tools for reviewing and manually annotating documents [3, 4].

Our encryption tools assume a simple XML encoding to embed tags in documents. In the demo prototype depicted below, the XML encodings indicating the locations of the sensitive attributes in the document are generated manually.

Our access-control system allows a data owner to encrypt sensitive portions of documents, and to define access rights that allow users selective access to encrypted content. A region can be determined to be sensitive by its associated tags. To illustrate how our tool might be used we discuss two scenarios. The first concerns processing a mortgage application; the second deals with a company that is attempting to satisfy investors while protecting sensitive internal information. We describe each in turn and provide screenshots of the tool's output.

Residential Loan Processing. A typical mortgage application is shown in the figure on the left side of figure 1. The application contains several fields with potentially sensitive information, specifically, the applicants' names, SSNs, addresses, phone numbers and dates of birth. These types of information form the attributes in our attribute-based encryption protocol, that is, a user authorized to view the phone number attribute will be able to decrypt any phone number in the encrypted mortgage application. Figure 1 shows the unencrypted mortgage application on the left and

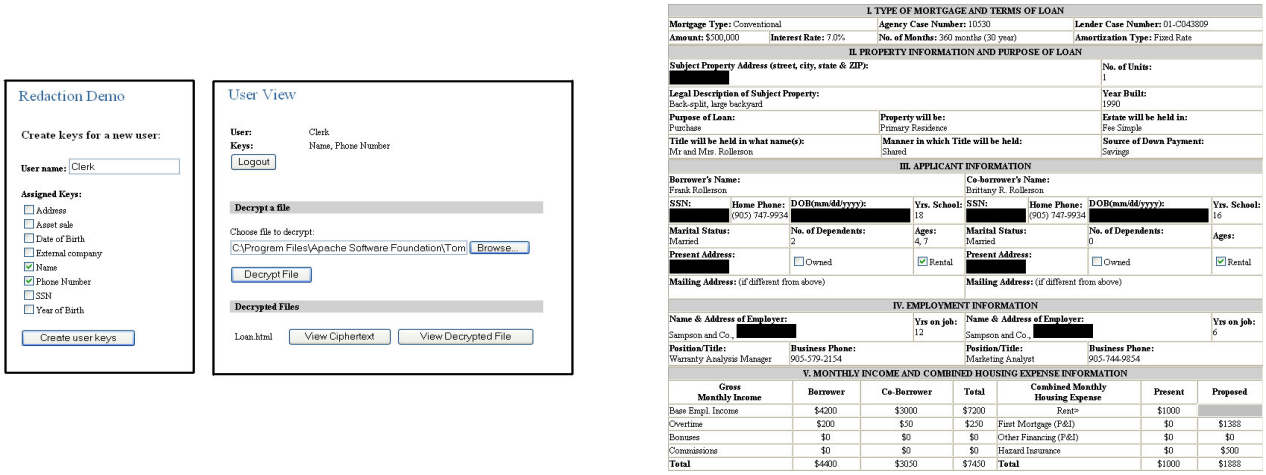


Figure 3: The left side of this figure shows the administrator window for defining the Clerk's access rights at the top and the Clerk selecting the mortgage application to decrypt below. The right side of the figure show the mortgage application as it appears to the Clerk. The Clerk can view names and phone numbers, but other sensitive information remains encrypted.

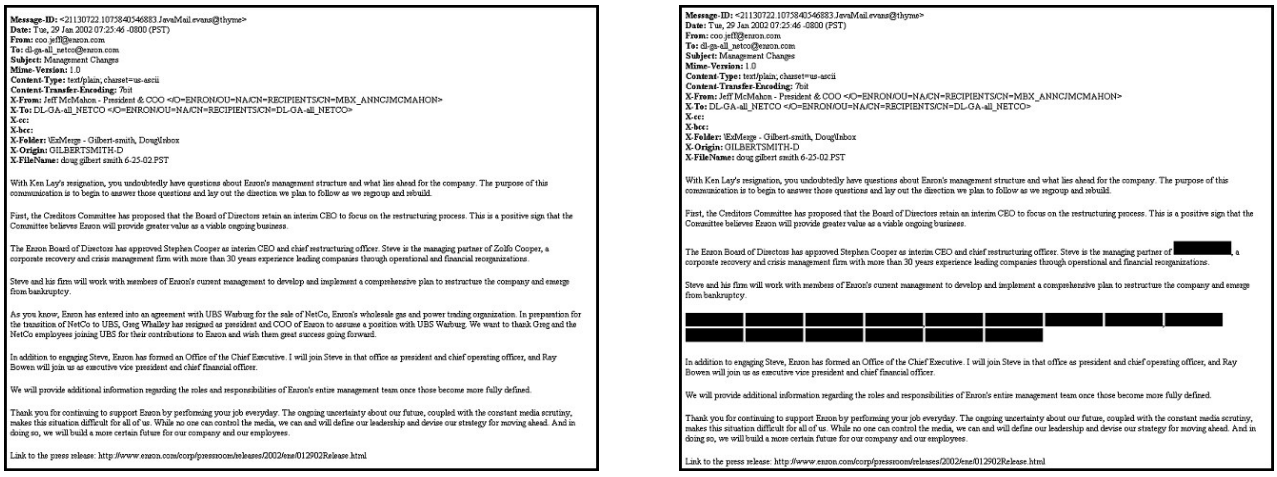


Figure 4: The left side of this figure shows the original Enron email in unencrypted form and the right side shows the same email but with external companies and a passage about an asset sale encrypted, as these are potentially sensitive.

the encrypted application (using the scheme of section 5) on the right. The encrypted fields in the form are depicted as black boxes. Note that any two fields corresponding to the same type of attribute contain a rectangle of the same size to give resistance to attacks that leverage the length of the redaction "bar" to infer the information removed [20].

We consider two users and demonstrate how the system ensures each will be able to decrypt the information they need in the encrypted application, while other sensitive information remains encrypted. First, we define a creditor who needs the applicants' names, social security numbers and year of birth to run a credit check. The left side of figure 2 shows the window in which the administrator selects the attributes to which the creditor should have access and creates the corresponding decryption key for the creditor. The term "keys" is used in the window because we found it more intuitive to users to think of each attribute having an associated key, as opposed to a decryption key being associated with an access right, as we describe in section 4.

On the left side of figure 2 we also show the creditor logging in, and on the right side of the figure we show the creditor's view of the document. The creditor is able to decrypt the SSNs, names and year of birth, but other potentially sensitive information such as addresses and phone numbers, remains in encrypted form.

Second, we consider a clerk whose job is to follow up with applicants after loan approval for quality assurance purposes. The clerk needs to know the phone numbers and names of the applicants, but doesn't need other potentially sensitive information such as SSNs and addresses. Figure 3 shows the clerk user being created and given the necessary access rights on the left, and the clerk decrypting the loan application with their assigned decryption key on the right. The clerk is able to decrypt phone numbers and names but not dates of birth, SSNs and addresses.

Protecting Corporate Identity. An email from the

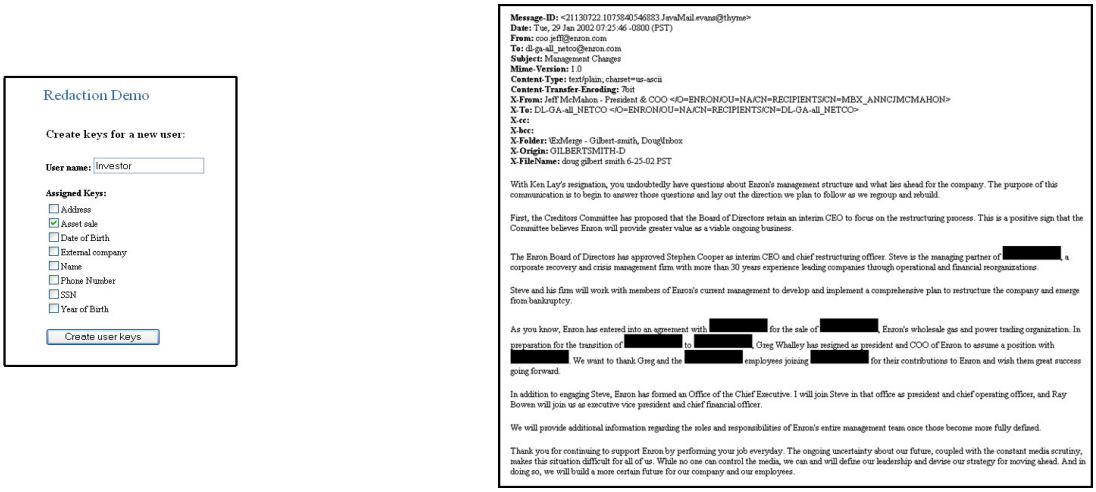


Figure 5: The left side of the figure shows an investor being given access to asset sale information. The right side shows that the investor can view asset sale information although actual company names remain encrypted.

Enron corpus [13] appears in figure 4 on the left side. This email provides information about an Enron asset sale that might be important for a company to share with investors as proof that the company is taking the right steps toward viability. However, the email also mentions several external partners who might not have given permission for their names to be released. The right side of figure 4 shows this information protected, the company names have been encrypted and a paragraph that contains information on an asset sale and the purchasing company has been encrypted. Note that the black boxes indicating ciphertexts all have the same size to resist attacks like those in [20].

In figure 5 we see an investor user being created and being given the right to access information about the asset sale (but not external company names)[3]on the left, and we see the email as it is viewed by this investor on the right.

Finally, we make some comments about the performance of our code. We have implemented a research prototype and have not optimized for speed. The slow part of our implementation involves pairing computations (using Miller's original algorithm). Steps that don't rely heavily on pairing computations are relatively fast. For example, generating a user decryption key for 2 attributes averaged 200 ms on an iBook G4. In the current prototype, decrypting content can be quite slow, taking several seconds, however, we expect that leveraging recent improvements in computing pairings (see, for example, [5]) will improve the performance of our prototype.

7. EXTENDING THE ABE MODEL TO USER REVOCATION

Although revocation is likely to be needed less in the document access setting that we consider than in the pay-television setting that motivates much of the existing broadcast encryption and cryptographic revocation schemes, there are certain to be instances in which it's necessary to block the access of rogue users to new documents introduced to the system. Because this may be a rare event we choose not to add user revocation to the existing schemes by assigning users additional keys based on revocation schemes such as [22] and encrypting the document ciphertexts according to such a revocation scheme. Rather, we propose a heuristic extension to the scheme of section 5 that incurs no additional user storage.

We begin by defining our efficient ABE scheme with user revocation. An ABE scheme with user revocation consists of the same four algorithms as a standard ABE scheme with the following differences. The first difference is that we assume that the number of attributes $m = |\mathcal{W}|$ is even, and that every document is tagged with a set T of exactly $|T| = m/2$ attributes. We justify these assumptions by noting that every attribute W_i has a negation $\overline{W_i}$. If all attributes and their negations are included in $\mathcal{W}$, the set $\mathcal{W}$ is of even size m and every document is naturally tagged with $m/2$ attributes (for each attribute, either the attribute itself, or its negation).

Setup$(\lambda, \mathcal{W}, \mathcal{U}, v)$ takes as input a security parameter λ, a set of attributes $\mathcal{W} = \{W_1, \ldots, W_m\}$ where m is even, a set of users $\mathcal{U} = \{U_1, \ldots, U_n\}$ and an upper-bound v on the number of users who can be revoked. The algorithm **Setup** outputs the public parameters and a master key, (pub, mk). The set $\mathcal{W}$ and $\mathcal{U}$ must be part of the public parameters.

KeyGen(mk, U_i, σ) takes as input a master key mk, a user $U_i \in \mathcal{U}$, and a monotone boolean formula σ over the variables $W_1, \ldots, W_m$ and their negations $\overline{W_1}, \ldots, \overline{W_m}$, and outputs secret parameters $d_{U_i,\sigma}$.

Encrypt(mk, M, T, R) takes as input a master key mk, a message M, a subset $T \subset \mathcal{W}$ of the attributes that take the value true in the message M such that $|T| = |\mathcal{W}|/2$, and a set $R \subseteq \mathcal{U}$ of revoked users such that $|R| = v$ (for simplicity, we assume that there are always exactly v revoked users; the introduction of dummy

[3]Note that NetCo, an organization that was part of Enron, is protected along with the external company names. With additional content analysis it might be possible to distinguish NetCo from the external companies.

users allows for revocation of fewer than v "real" users). The algorithm **Encrypt** outputs a ciphertext C.

Decrypt$(T, d_{U_i,\sigma}, C)$ takes as input a set of attributes $T \subseteq \mathcal{W}$ such that $|T| = |\mathcal{W}|/2$, a secret key $d_{U_i,\sigma}$ corresponding to a boolean formula σ for user U_i and a ciphertext C, and outputs a message or a special symbol $\perp$.

such that if T satisfies σ and $U_i \notin R$, then we have

$$\textbf{Decrypt}(T, d_{U_i,\sigma}, \textbf{Encrypt}(mk, M, T, R)) = M.$$

In other words, user U_i can decrypt all documents whose attributes satisfy the access rights of U_i, provided U_i's right to decrypt these documents were not revoked.

7.1 Security Definition

Formally, we define the security of an ABE scheme with user revocation through the following game between an adversary and a challenger:

Setup: the challenger runs the **Setup** algorithm, gives the public parameters to the adversary and keeps the master key to himself.

Phase 1: the adversary adaptively issues queries of the following types:

- **Key query:** (σ, U_i), where σ is a boolean formula over the variables in $\mathcal{W}$ and their negations and $U_i \in \mathcal{U}$. The challenger responds by running the **KeyGen** algorithm and gives the secret key $d_{U_i,\sigma}$ corresponding to σ for user U_i to the adversary.
- **Encryption query:** (M, T, R), where M is a message, $T \subset \mathcal{W}$ is a subset of the attributes such that $|T| = |\mathcal{W}|/2$, and $R \subseteq \mathcal{U}$ is a subset of users such that $|R| \leq v$. The challenger responds by computing **Encrypt**(mk, M, T, R) and giving the resulting ciphertext to the adversary.

Challenge: the adversary outputs two equal length messages M_0 and M_1, a subset T^* of $\mathcal{W}$ such that $|T^*| = |\mathcal{W}|/2$ and a subset R^* of $\mathcal{U}$ with the following property: if T^* satisfies one of the key queries (σ, U_i) issued in Phase 1, then $U_i \in R^*$.

The challenger then picks a bit $b \xleftarrow{R} \{0, 1\}$, encrypts $C \leftarrow \textbf{Encrypt}(mk, M_b, T^*, R^*)$ and outputs C.

Phase 2: the adversary adaptively issues key and encryption queries as in Phase 1, such that all queries (σ, U_i) satisfy the following property: if T^* satisfies (σ, U_i), then $U_i \in R^*$. The challenger answers these queries as in Phase 1.

Guess: the adversary outputs a bit b'.

We define the advantage of the adversary in attacking the scheme as in section 4, and say that the ABE scheme with user revocation is secure if the adversary's advantage is negligible.

7.2 An ABE Scheme With User Revocation

Our efficient ABE scheme with user revocation is formally defined below. It is a heuristic extension of the scheme of section 5 and is inspired by [23].

Setup$(\lambda, \mathcal{W}, \mathcal{U}, v)$.

Say $\mathcal{W} = \{w_1, \ldots, w_{2m}\}$ and $\mathcal{U} = \{U_1, \ldots, U_n\}$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}'$ be an admissible bilinear map between groups $\mathbb{G}$ and $\mathbb{G}'$. Select a random generator $g \in \mathbb{G}$ and let $h = e(g, g)$. Select a random integer $S \in \{0, \ldots, |\mathbb{G}| - 1\}$. Select a random polynomial p of degree $m + v$ such that $p(0) = S$. Define $g_i = g^{p(i)}$ for $1 \leq i \leq 2m$. Finally, choose a random value $u_i > 2m$ for user U_i such that $u_i \neq u_j$ for all $i \neq j$. The public parameters are $(\mathbb{G}, g, n, m, v)$, and the master key is $mk = (h, (g_1, \ldots, g_{2m}), (u_1, \ldots, u_n), p)$.

KeyGen(mk, U_i, σ).

The key generation is exactly as described in section 5, except that we assign the secret value $g^{p(u_i)}$ to the root of the tree.

Encrypt(mk, M, T, R).

Select $r \xleftarrow{R} \{0, \ldots, |\mathbb{G}| - 1\}$.
Compute $u = g^r$, $v_j = g_j{}^r$ for $w_j \in T$, $\mu_j = g^{rp(u_j)}$ for $U_j \in R$ and $w = h^{rS} \cdot M$.
Return $C = (u, \{v_j\}_{w_j \in T}, \{\mu_j\}_{U_j \in R}, w)$.

Decrypt$(T, d_{U_i,\sigma}, C)$.

Say $C = (g^r, \{v_i\}_{w_i \in T}, \{\mu_i\}_{U_i \in R}, w)$.
Exactly as in the scheme described in section 5, user U_i recovers the value $g^{rp(u_i)}$ associated with the root of the tree of the formula that describes U_i's access rights. Provided C does not revoke user U_i, user U_i now has $m + v + 1$ values of the form $g^{rp(x)}$ for distinct values x. By polynomial interpolation, U_i can compute $h^{rp(0)} = h^{rS}$ and recover $M = w \cdot h^{-rS}$.

In the interest of space, we note simply that the variant of section 5.1 can be modified in the same way to achieve revocation.

8. CONCLUSION

We propose a system to protect identity and other sensitive information by controlling access to an individual's attributes through encryption.

Our system encrypts not only sensitive personal information, but also groups of personal attributes which may indirectly allow for the inference of a person's identity, even though none of the attributes is directly sensitive. Personal attributes are encrypted with an attribute-based encryption scheme, which allows for efficient, fine-grained access control based on the content of documents. In addition, we described a heuristic extension of our encryption scheme which supports revocation of access rights.

We have implemented and tested our identity protection scheme. Our prototype implementation demonstrates the usefulness of our scheme in financial and corporate applications. Our tool is broadly applicable for identity protection in other application areas as well, such as the medical and legal domains.

9. REFERENCES

[1] AreteQ. http://www.areteq.com

[2] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Advances in Cryptology – CRYPTO '88*, LNCS 403, 27-35, 1989.

[3] E. Bier, E. Ishak and E. Chi. Entity Workspace: an evidence file that aids memory, inference and reading. *Intelligence and Security Informatics, 2006*

[4] E. Bier and E. Chi. Entity quick click: rapid text copying based on automatic entity extraction. *CHI 2006.*

[5] I. Blake, V. Murty and G. Xu. Refinements of Miller's algorithm for computing Weil/Tate pairing. *Cryptology ePrint Archive, report 2004/065.*

[6] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. *Advances in Cryptology – Eurocrypt 2004.*

[7] D. Boneh and M. Franklin. *Identity based encryption from the Weil pairing.* In *SIAM J. of Computing*, Vol. 32, No. 3, pp. 586-615, 2003.

[8] D. Boneh, C. Gentry and B. Waters. Collusion resistant broadcast encryption with short cipertexts and private keys. *Advances in Cryptology – Crypto 2005.*

[9] R. Canetti, S. Halevi and J. Katz. A forward-secure public key encryption scheme. *Advances in Cryptology – Eurocrypt 2003.*

[10] R. Canetti, S. Halevi and J. Katz. Chosen-ciphertext security from identity based encryption. *Advances in Cryptology – Eurocrypt 2004.*

[11] F. Chen, A. Farahat and T. Brants. Multiple similarity measures and source-pair information in story link detection. *Human Language Technology Conference, North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2004)*; 2004 May 2-7; Boston; MA; USA. East Stroudsburg PA: ACL: 2004; 313-320.

[12] Y. Dodis and N. Fazio. Public-key broadcast encryption for stateless receivers. *ACM CCS Workshop of Digital Rights Management, 2002.*

[13] Enron Email Dataset. http://www.cs.cmu.edu/ enron/

[14] D. Ferraiolo and R. Kuhn. Role-based access control. *Proceedings of the 15th National Security Conference*, 1992.

[15] The 2007 Florida Statutes, 119.0714 Court files, court records; official records.

[16] A. Fiat and M. Naor. Broadcast encryption. *Advances in Cryptology– Crypto '93*, pp. 480-491.

[17] GATE: General Architecture for Text Engineering. http://gate.ac.uk/

[18] P. Golle. Revisiting the uniqueness of simple demographics in the US population. In *WPES 2006.*

[19] V. Goyal, O. Pandey, A. Sahai and B. Waters. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *ACM CCS, 2006.*

[20] D. Lopresti and A. L. Spitz. Quantifying information leakage in document redaction. In *HDP '04.*

[21] MortgageGrader. http://www.mortgagegrader.com

[22] D. Naor, M. Naor and J. Lotspeich. Revocation and tracing schemes for stateless receivers. *Advances in Cryptology – Crypto 2001.*

[23] M. Naor and B. Pinkas. Efficient trace and revoke schemes. *Financial Cryptography, 2000*, pp. 1-20.

[24] Redact-It. http://www.redact-it.com

[25] A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *Advances in Cryptology – Eurocrypt 2005*, pp. 457-473.

[26] D. Stinson. *Cryptography: Theory and Practice*, CRC Press, 1995.

[27] L. Sweeney. Uniqueness of simple demographics in the US population. LIDAPWP4. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA, 2000.

[28] J. Staddon, P. Golle and B. Zimny. Web-based inference detection. In *USENIX Security 2007.*

[29] Vontu. http://www.vontu.com

A content-driven access control system

Jessica Staddon, PARC
Philippe Golle, PARC
Martin Gagne, U. C. Davis
Paul Rasmussen, PARC

March 2008

parc
Palo Alto Research Center

Whose birthday is it?

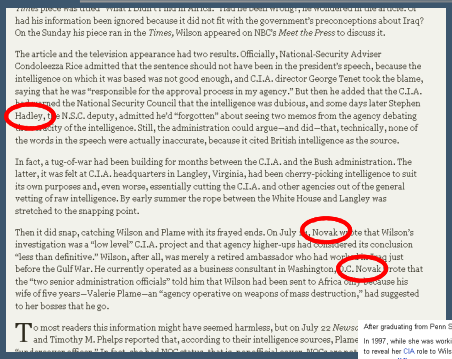
Date of birth
+
Gender
+
Location
=
Identity

[Sweeney 2000, Golle 2006]

Attributes are sensitive

parc
Palo Alto Research Center

Tie access rights to attributes



had his information been ignored because it did not fit with the government's preconceptions about Iraq? On the Sunday his piece ran in the *Times*, Wilson appeared on NBC's *Meet the Press* to discuss it.

The article and the television appearance had two results. Officially, National-Security Adviser Condoleezza Rice admitted that the sentence should not have been in the president's speech, because the intelligence on which it was based was not good enough, and C.I.A. director George Tenet took the blame, saying that he was "responsible for the approval process in my agency." But then he added that the C.I.A. had informed the National Security Council that the intelligence was dubious, and some days later Stephen Hadley, the N.S.C. deputy, admitted he'd "forgotten" about seeing two memos from the agency debating the veracity of the intelligence. Still, the administration could argue—and did—that, technically, none of the words in the speech were actually inaccurate, because it cited British intelligence as the source.

In fact, a tug-of-war had been building for months between the C.I.A. and the Bush administration. The latter, it was felt at C.I.A. headquarters in Langley, Virginia, had been cherry-picking intelligence to suit its own purposes and, even worse, essentially cutting the C.I.A. and other agencies out of the general vetting of raw intelligence. By early summer the rope between the White House and Langley was stretched to the snapping point.

Then it did snap, catching Wilson and Plame with its frayed ends. On July [illegible] Novak wrote that Wilson's investigation was a "low level" C.I.A. project and that agency higher-ups had considered its conclusion "less than definitive." Wilson, after all, was merely a retired ambassador who had worked [illegible] just before the Gulf War. He currently operated as a business consultant in Washington. [illegible] Novak wrote that the "two senior administration officials" told him that Wilson had been sent to Africa only because his wife of five years—Valerie Plame—an "agency operative on weapons of mass destruction," had suggested to her bosses that he go.

To most readers this information might have seemed harmless, but on July 22 *Newsw* and Timothy M. Phelps reported that, according to their intelligence sources, Plame

After graduating from Penn State, Plame was briefly married to her college boyfriend Todd Sesler, who was also accepted into the CIA training program but decided to not pursue it.[1]

In 1997, while she was working for the CIA, Plame met former Ambassador Joseph C. Wilson IV "at a reception in Washington ... at the residence of the Turkish Ambassador."[10] Una to reveal her CIA role to Wilson on their first date, initially she told him that she was an energy trader in Brussels, and he thought that she was "an up-and-coming international executive."[14] After they began dating and became "close," Plame revealed her employment with the CIA to Wilson (Wilson, *Politics of Truth* 242).[14] They were married on April 3, 19 Plame's second marriage and Wilson's third (Wilson, *Politics of Truth* 273).

Professionally and socially, she has used variants of her name. Professionally, while a covert CIA officer, she used her given first name and her maiden surname, "Valerie Plame." Sinc leaving the CIA, as a speaker, she has used the name "Valerie Plame Wilson," and she is referred to by that name in the civil suit that the Wilsons brought against former and current government officials, *Plame v. Cheney*.[2] Socially, and in public records of her political contributions, since her marriage in 1998, she has used the name "Valerie E. Wilson."

At the time that they met, Wilson relates in his memoir, he was separated from his second wife Jacqueline, a former French diplomat; they divorced after 12 years of marriage so that h could marry Plame. His divorce had been "delayed because I was never in one place long enough to complete the process," though he and she had already been living separate lives a the mid-90s.[15] Plame and Wilson are the parents of twins, Trevor Rolph and Samantha Finnell Diana, born in January 2000. From his first marriage (1973-1996), to Susan Dale Otchis Wilson is also the father of another set of twins (also a boy and a girl), Sabrina Cecile and Joseph Charles, who were born in 1975.

Prior to the disclosure of her classified CIA identity, Valerie and Joe Wilson and their twins lived in the Palisades, an affluent neighborhood of Washington, D.C., on the fringe of Georgetown.[16] After she resigned from the CIA following the disclosure of her covert status, in January 2006, they moved to Santa Fe, New Mexico.[16][17]

Career

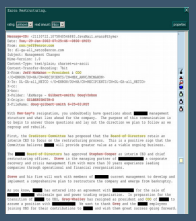
Soon after graduation from Penn State, Plame moved to Washington, D.C.,[18] where she worked at a clothing store, biding her time, while awaiting the results of her application to the Central Intelligence Agency (CIA).[2][19] She was accepted into the 1985-86 CIA officer training class, beginning her training for what became a twenty-year career with the Agency.[9] Although the CIA will not release publicly the specific dates from 1985 to 2002 when she worked for it, due to security concerns,[17][20] Special Counsel Patrick Fitzgerald affirmed that Plame "was a CIA officer from January 1st, 2002...forward" and that "her association with the CIA was classified at that time through July 2003."[18] At many times in Ms. Plame's career put her life in profound danger. Due to the nature of her clandestine work for the CIA, many details about Plame's professional career are still classified, but it is documented that she

Document attributes include the document's *content*

Both documents talk about Plame and Wilson. Leak document talks about Novak and Hadley too.

Our approach

- Automated entity extraction to identify names, places, etc.
  - text is tagged based on content
- Sensitivity identification based on the entities
  - May involve topic detection



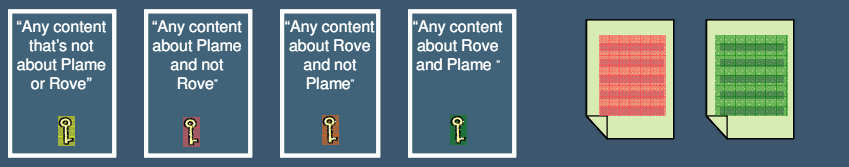
- Document encryption based on content of document
  - A form of attribute-based encryption

This talk

parc
Palo Alto Research Center

First attempt

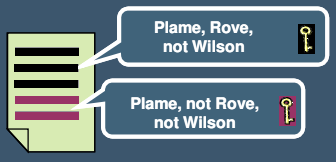
- Associate a key with each possible user access right
  - User stores a single key corresponding to their access right
  - Encrypt content with every key corresponding to a satisfying access right
  - Low user storage, *high document overhead*



parc
Palo Alto Research Center

A second attempt…

- Associate a key with each set of tags
  - Encrypt document region with key corresponding to tags that are and aren't in region
  - User stores all keys corresponding to sets of tags satisfied by their access rights
  - Low document overhead, *high user storage*

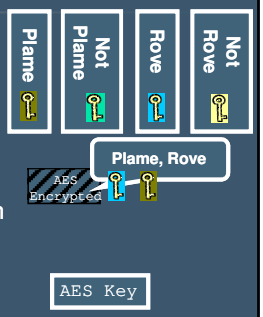


User with access to "all content that's not about Wilson" stores:
…

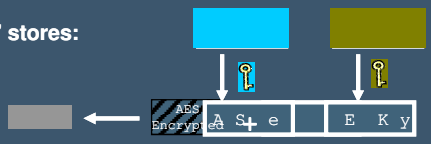
parc
Palo Alto Research Center

Overview of our approach

- Create a key for each *tag*
- The encryption of a document region is:
  1. Encryption of the text under a randomly selected symmetric key (e.g. AES)
  2. Keys corresponding to tags associated with region
- Users store AES key encrypted under tag keys
  - If a region doesn't have the right tags, the user won't be able to recover the AES key and so can't decrypt the region



User with access right "Plame and Rove" stores:



parc
Palo Alto Research Center

Some of the missing details…

- Randomize the process to make it work more than once!
  - Document region ciphertext includes *randomized* versions of keys
  - AES keys are randomly generated from a base AES key
  - What users learn from decrypting 1 document region has no impact on their ability to decrypt a 2nd region
    - Regions must have the right tags in order for a user to decrypt

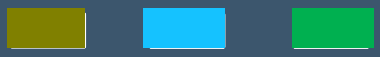
parc
Palo Alto Research Center

A small example: Set-up

- Analyst has permission to read anything pertaining to Plame leak
  - Can read any document pertaining to at least 2 of "Plame", "Rove", and "Novak"
- Initialization:
  - Groups G and H, each of prime order
    - Generators g and h, respectively
  - Bilinear map e(.,.): e(g,g)=h
  - a1, a2, a3, r, r1, r2, r3, random elements in {1,…,|G|-1}
  - Q(x) a polynomial of degree 1
- Let D be a document about Plame and Rove (not Novak)

parc Palo Alto Research Center

Encryption & Decryption



- Analyst's key: g^r, $g^{Q(a1)+r(r1)}$, $g^{Q(a2)+r(r2)}$, $g^{Q(a3)+r(r3)}$
- Encryption of D:
  - $g^{r'}$
  - $g^{r'r1}$
  - $g^{r'r2}$
  - $E_K(D)$ where $K=h^{r'Q(0)}$

- Sketch of Decryption:
  1. $e(g^r, g^{r'ri})=h^{rr'(ri)}$
  2. $e(g^{Q(ai)+r(ri)}, g^{r'})=h^{r'(Q(ai)+r(ri))}$
  3. From 1 & 2, recover $h^{r'Q(ai)}$, for i=1, 2, and use polynomial interpolation to recover $K=h^{r'Q(0)}$

parc Palo Alto Research Center

What have we achieved?

- Fine-grained, content-driven access control
- Encryption overhead grows with the # tags
  - Not with the number of access rights
- User storage grows with the complexity of the access rights
  - Not with the number of access rights
- Secure provided Bilinear Decisional Diffie-Hellman is hard
- Prototype implementation
  - Defines access rights in terms of categories of information
  - Addresses, DOBs, SSNs, Phone Numbers, Company Names, etc.

parc
Palo Alto Research Center

Extensions

- Can implement any access right expressible as a Boolean formula
  - E.g. (Plame & Wilson & Novak) or (Plame & Wilson & Libby)
- Attributes can be metadata (e.g. user-generated tags) in addition to extracted entities
- Supports revocation as a heuristic extension
  - Revoked user can't access new content that matches their old access rights
  - Idea: Adapt scheme in spirit of broadcast encryption scheme of Naor-Pinkas 2000

parc
Palo Alto Research Center

Other use cases

- Mortgage applications
  - Encrypt the fields of the application according to content type
  - Appraiser can decrypt address, but not SSNs
  - Credit checker can decrypt SSNs and not addresses
  - A single encrypted copy of the document can be maintained
- Mergers and Acquisitions
  - Encrypt company records for “virtual data rooms”
  - Partners access rights change as negotiations progress
    - Can decrypt more and more data

parc
Palo Alto Research Center

Conclusion

- Even seemingly innocuous attributes can be sensitive when taken together
- To preserve privacy, access control should be in terms of the attributes of the content
- Our encryption protocol supports attribute-based access rights

parc
Palo Alto Research Center

Thanks!

parc
Palo Alto Research Center

Secure Roaming with Identity Metasystems

Long Nguyen Hoang
Helsinki University of Technology
Finland
silver@cc.hut.fi

Pekka Laitinen
Nokia Research Center
Finland
pekka.laitinen@nokia.com

N. Asokan
Nokia Research Center
Finland
n.asokan@nokia.com

ABSTRACT

The notion of identity metasystem has been introduced as the means to ensure inter-operability among different identity systems while providing a consistent user experience. Current identity metasystems provide limited support for secure roaming: by "roaming" we refer to the ability of a user to use the same set of identities and credentials across different terminals. We argue that in order to support different types of roaming, the identity metasystem client should be structured as a set of distributable components. We describe such distributed client-side software architecture and how that architecture is implemented by adapting Novell's Bandit project. We use our implementation to demonstrate how credentials are stored in a trusted device in the form of a mobile phone but can be used on less trusted terminals in the form of PCs.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection, Authentication.

General Terms

Management, Design, Security, Human Factors.

Keywords

Identity metasystem, Mobility, Roaming.

1. IDENTITY METASYSTEMS

As more and more transactions are carried out over the Internet, the need for easy-to-use and secure authentication mechanisms becomes increasingly evident. This has resulted in a number of identity systems intended to save the users from having to create, manage, and use various passwords for different service providers. Each of these systems uses different protocols and requires different types of user interaction. More recently, the notion of Identity Metasystem [1] has gained ground. The main purpose of identity metasystem is to put an abstract identity management layer on the Internet to allow existing identity systems based on various technologies to inter-operate with each other while providing a consistent user experience regardless of which identity system is used.

Technically, identity metasystem introduces the concept of an "information card" modeled after a business card, license, badge, etc. In general, an information card is a digital representation of user identity. A card may be managed or self-issued. Managed cards are issued by trusted authorities known as identity providers and contain only metadata describing how to get security tokens from those identity providers. Secret credential needed when generating authentication token are managed by those identity providers. Self–issued cards, on the other hand are managed by users themselves with the help of the local identity system on their devices. Secret credential information needed to generate security tokens for self-issued cards should be persistently stored on user devices themselves (or be provided by the user every time). This leads to the issue of roaming. To illustrate the issue, let us consider the following example scenario:

Alice has a number of information cards on her home PC. She uses them to access her web-based emails as well as for other services. When she decides to go on a trip, she loads her information cards on her mobile phone before leaving. When Alice arrives at the station, she notices some kiosk PCs that she can use to access the Internet. Alice wants to check her mail. Since she is unsure whether the kiosks are trustworthy, she does not want to transfer all of her self-issued cards to the kiosk machine. Yet, she would prefer to use the kiosk for reading email because it has a convenient display and keyboard. Alice starts up the identity metasystem on the kiosk as well as on her phone. They discover each other and establish a secure Bluetooth [10] connection. With a single click on her phone, Alice allows her information cards to appear on the identity selector user interface (UI) on the kiosk. When Alice attempts to sign in to her e-mail account, her phone prompts her for authorization instead of asking her to username/password on the terminal. She confirms on her phone and her mails can then be read on the kiosk as normal. When she finishes her reading, she leaves the kiosk and the Bluetooth connection terminates. Her information cards, as a consequence, disappear from the kiosk's identity selector UI.In this example, although Alice is willing to trust the kiosk as the means to read her e-mail while she is physically present at the kiosk, she does not want to reveal the keys that can be used by malicious software on the kiosk to access her mail after she has left.

The most well-known identity metasystems so far are Microsoft CardSpace [2][3][4][5] and the open-source Bandit project [6]. Both CardSpace and Bandit allow the possibility to *export* information cards from one device and import them into another but the export/import operation results in the transfer of the entire

IDtrust '08, March 4-6, 2008 Gaithersburg, MD.

information card, including the secret credentials that are part of self-issued cards. Currently these systems are unable to support the usage scenario described above. To support such scenarios, the identity metasystem functionality should be split up into parts so that some can stay on a trusted device like the mobile phone, while the others can be migrated to a less trusted terminal temporarily. In this paper, we show that there are different ways to partition the functionality of an identity metasystem thereby allowing two or more devices to co-operate in providing the identity metasystem client functionality. Therefore we argue that in order to realize such scenarios, the identity metasystem client should be structured as a collection of distributable components.

In this paper, we propose a distributed software architecture for identity metasystem clients. We also study different ways to distribute client functionality among two or more devices. We then describe how we have implemented our architecture using the Bandit client implementation where one part of the client is running on a PC and the other on a mobile phone.

The rest of this paper is structured as follows. In Section 2, we give a brief review of the identity metasystems architecture in general and its current realizations. In Section 3, we describe our distributed architecture and discuss different ways to partition the client functionality between two devices In Section 4 we describe the technical details on how we have adapted the Bandit identity metasystem to our architecture to support secure roaming. Finally, in Section 6, we discuss possible extensions and conclude.

2. IDENTITY METASYSTEMS

2.1 General Concept

The identity metasystem model emphasizes the roles of three key agents: the **relying party** (RP), the **identity provider** (IdP) and the **identity selector system** (ISS). A RP is any system capable of authenticating users based on their information cards, it can be a program, a web server, or any other service. The user is required to prove their identity before accessing to the resource. The IdP, as in Kim Cameron's vision [7], issues information cards to user and provides authentication service. Finally, the identity selector is the system that is responsible for handling messages between an RP and an IdP, and providing a secure mechanism for user to manage their information cards. Figure 1 expresses a simple authentication process in identity metasystem. Whenever a user wants to access a service (e.g., a web page), the RP offering the service first sends the security requirements on user authentication it expects to be satisfied (which is stated in RP's policy). On the user's terminal, the identity selector interface processes these policies and automatically displays appropriate information cards that satisfy the policies. The user can choose one of these cards. It should be noted that, conceptually, an information card contains nothing but metadata saying only where and how to retrieve the identity information. Once the user selects a card, the identity selector facilitates the authentication of the user to the identity provider (which requires an additional authentication mechanism to be specified in the information card. For example, this authentication may be based on a username/password or a digital signature and certificates) and sends a request for security token based on RP's policies. The IdP verifies the request and issues a security token in return. To complete the authentication process, the security token is passed to the RP through the identity selector system to prove the user's identity to the service.

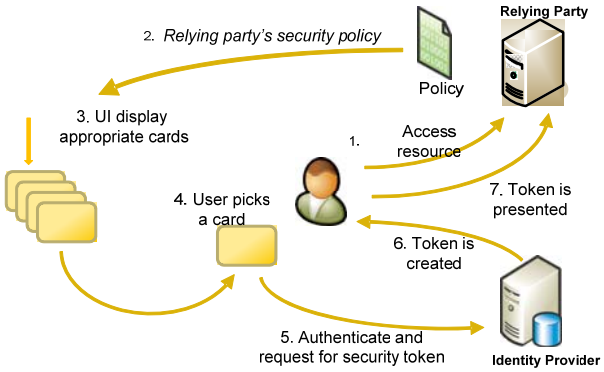


Figure 1: Identity metasystem authentication process

2.2 Microsoft CardSpace

Microsoft CardSpace [2][5], previously known as InfoCard, is the first commercial implementation of the identity metasystem concept and the one with the most widespread deployment since it is shipped as part of Windows Vista platform. Figure 2 depicts the system architecture of CardSpace. Services or applications on the platform trigger CardSpace system via an "*activator*" (*infocardapi*). The core part of the system, the CardSpace service handles all token and management requests, and invokes other services such as user interface or low-level storage. However, in the current version 1.0, secure roaming is not possible: in order to use the same information cards on multiple machines, the end user has to *export* his information cards to a file (with a *.crds* extension) and then *import* it into the CardSpace system on another machine. Although import/export is conceptually simple, it involves moving all data associated with the card, including sensitive data like credential secrets of self-issued cards. If Alice in our example scenario in Section 1 uses CardSpace, she has to remember to remove all the cards including the secret data from the kiosk before she leaves. Moreover, even if she remembers to do this, she has no way of knowing if her cards and credentials had been copied before they were removed from the terminal.

2.3 Novell Bandit project

Bandit project [6] is an open-source, cross-platform implementation of the identity metasystem concept. Technically, Bandit consists of a set of loosely-coupled identity components that provide identity services while ensuring both interoperability and collaborations between agents. Figure 3 shows the high-level system architecture of Bandit. On the top is the management user interface, called *DigitalMe*. The main ISS module handles all requests. Bandit does not only specify and implement the identity selector as CardSpace does but also provides a common framework for easily integrating any digital identity management system to Bandit using Higgins framework [10]. Bandit supports the possibility of using multiple storage providers where information cards and other data used by the client can be stored persistently. Currently it allows the storage provider to be the local file system, or a Bluetooth-connected storage device, or an online database. As an open source project, Bandit constitutes an excellent platform for further development.

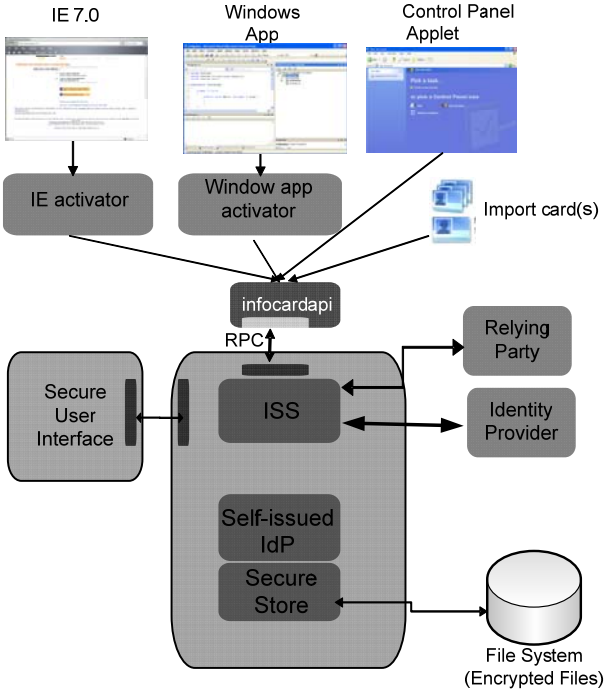


Figure 2: Microsoft CardSpace architecture [5]

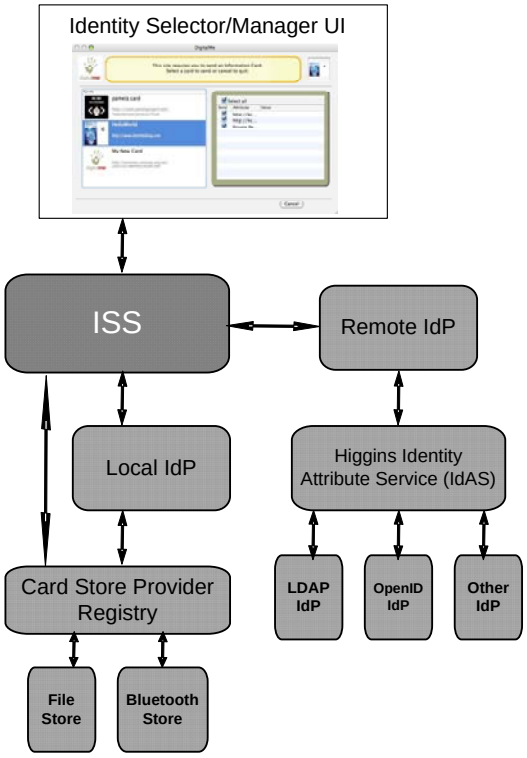


Figure 3: Novell Bandit architecture [6]

3. ARCHITECTURE OF IDENTITY METASYSTEMS

3.1 General Architecture

There is no definitive identity metasystem architecture. In general, identity metasystems are expected to follow the principles set out by the "laws of identity" [7]. As we saw in Section 2, different realizations of the architecture have been implemented. Fortunately, those realizations have similar characteristics. Figure 4 depicts our own definition of identity metasystem architecture. The architecture consists of loosely-coupled identity components with interfaces between them:

Relying Party: the relying party can be considered a "remote" component in a whole identity metasystem. In general, a relying party is any system hosting some services which require authentication before being accessed. In the context of identity metasystem, a relying party first publishes its security policy (using [11][12]) then authenticates the user based on a security toke-claim assertion.

Identity Selector: The role of the identity selector is to provide a consistent identity management tool for end users. This component also operates as a contextual connector between relying parties and identity providers such that the end user can choose which identity provider should be used to authenticate the end user to the particular relying party. Technically, the identity selector requests a security token from the identity provider and then passes that issued token to the relying party according to the information card chosen by the end user.

Self-issued IdP, Remote IdP: The identity provider is responsible for managing user's digital identities in the form of information cards and issuing security tokens on demand. Remote IdP is maintained by a well-known, trusted organization while self-issued IdP is managed locally (in the operating system) by end users themselves. As the identity provider maintains all the user credentials, it must be secured. For example, the self-issued IdP can be protected using end user's local trusted machine.

IdP Authentication: In order to obtain a security token from an IdP, the identity metasystem must first authenticate to the IdP that is specified in the chosen information card. Authentication mechanisms can be based on self-issued card token or other existing authentication technologies such as username/password, Kerberos [13], or Public Key Infrastructure (PKI)[14], depending on IdP's security policy. In general, this component provides an abstract way to authenticate an end user to an identity provider.

Trigger: The trigger is a "bridge" between identity metasystem - aware applications and the identity selector system. Whenever an end user accesses a service that supports the identity metasystem, the trigger is used to activate the authentication process by first collecting the relying party's policy, then activating the identity selector. At the end of the process, the trigger dispatches the security token returned from the identity selector to the relying party.

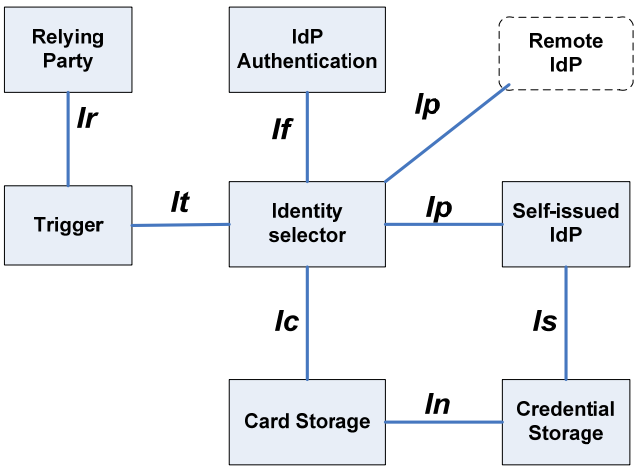


Figure 4: System components and interfaces

Card Storage: Responsibilities of the card storage include information card manipulation, card enumeration as well as filtering possible information cards based on what type identity information is required based on RPs' policies.

Credential Storage: The credential storage maintains user's credentials including personal information, certificates and other critical data. In practice, the credential storage and the card storage are typically combined. For example with self-issued cards, metadata and private user information are stored together in one XML document.

Every identity component is aware of other components. For example, component A may send a request to a specific component B (unicast) or a group of components (multicast) or all (broadcast). Here a middleware becomes useful because it provides a seamless mechanism for inter-component communication. We will discuss more about this in Section 4.

In distributed systems, components should communicate via well-defined interfaces. In Table A1, we list some common operations for each interface between the identity components in an identity metasystem. Figure A1 shows a simplified sequence diagram of identity component interaction for the scenario described in Section 1. The low-level connections between the identity components are assumed to be established beforehand (implicit setting, configuration setting, or dynamic discovery) and secure enough. If the transport layer is not secure enough then the security semantic needs to be improved in the application protocol layer between the distributed identity components which may be future topics. Interface ***If*** and ***Ip*** are special in that their binding protocols are decided on demand: security policy of identity provider decides the binding protocol of ***Ip*** and ***If***. For example, if an IdP states in its policy that it requires username/password authentication, the identity selector must establish an SSL connection to the IdP, capture user's input from ***If*** and send those credentials through a secure channel over ***Ip***. Alternatively, the authentication to the IdP can done using a smart card (***If***) without having to use SSL connection. Binding protocols for other interfaces are unspecified, they can be RPC [15] calls or SOAP [16] requests or any custom-defined messages.

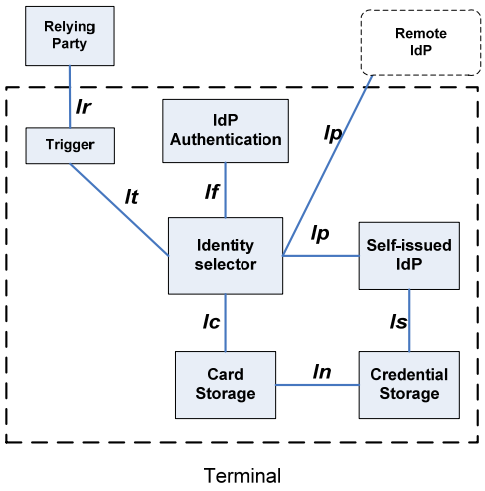


Figure 5: Configuration 0 – all in terminal

3.2 Distribution of Identity Components

An identity metasystem, which supports secure roaming, is expected to support various use cases in run time. For example, the end user may use his trusted device as a simple external storage device or a full-featured device including the user interface interaction to establish the connection to the terminal where service is consumed. The terminal, in return, should adapt its functionality to different settings according to client profile set on the trusted device. This yields to a technical issue that identity metasystem components, not only within a system but also across distributed systems, should be able to seamlessly cooperate to complete a particular task. We call these use cases "configurations". In this section, we list out six typical configurations together with their strong points and drawbacks.

Configuration 0 - "all in terminal": This configuration is the simplest configuration in which all identity components are packed in the terminal side. This is also the starting point for every identity metasystem. Because user's information is kept in one place, having the same identity profile on multiple machines securely is not possible. Configuration 0 is on the level of roaming that is currently supported by CardSpace, i.e., export/import functionality of cards. Although CardSpace allows user to export all his cards onto an external device and then import them from that device, CardSpace cannot directly use cards that are on an external device.

Configuration 1 - "external storage": In this configuration, the trusted device provides credential storage service and card storage service. It is more like an external database such as an USB stick or a smartcard. The advantage of this configuration is that the trusted device does very little processing. It is noted that communicating channels between the host and the device must be protected as the self-issued IdP in the terminal accesses directly to the credential storage component running on the trusted device. However, to gain better performance, we can rely on the security of the transport protocol being used in ***Is*** as long as it is secure enough.

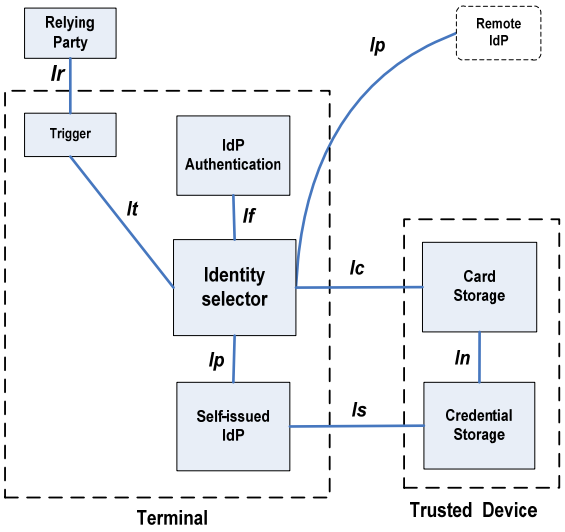


Figure 6: Configuration 1 – External storage

Configuration 2 - "mobile identity provider": This configuration extends the functionality of the trusted device in configuration 1. The device can now operate as a self-issued identity provider. The self-issued IdP component running on the trusted device is responsible for issuing security tokens when self-issued information cards are used. The identity selector on the terminal can also use managed information cards and request for security tokens from remote identity providers over the network as usual. This configuration is considerably more secure than configuration 1 since the interfaces between the terminal and the trusted device are ***Ip*** and ***Ic*** and the data transmitted over them is not that sensitive. Furthermore, user's credentials always stay in the trusted device as the credential store component is only accessed locally by the self-issued IdP component that is also residing in the trusted device. One tradeoff is the increase in the resource consumption in the trusted device when issuing the security tokens (including parsing the request from the identity selector component, retrieving corresponding data, generating/signing/encrypting the security token before sending it back to the identity selector component).

Configuration 3 - "mobile identity selector": In this case, the trusted device supports built-in identity selector feature. The end user browses and uses services on the terminal but performs all the authentication related operations on his trusted device. Whenever he is asked to prove his identity, the trigger component on the terminal activates the identity selector user interface on the trusted device so that he can choose one of his information cards. This gives advantage and convenience to the end users because they perform all security related operations on their trusted devices (including the authentication related user interface ***If***). This is also a benefit when the terminal itself does not have a proper display, e.g., auto ticket seller or electronic door access.

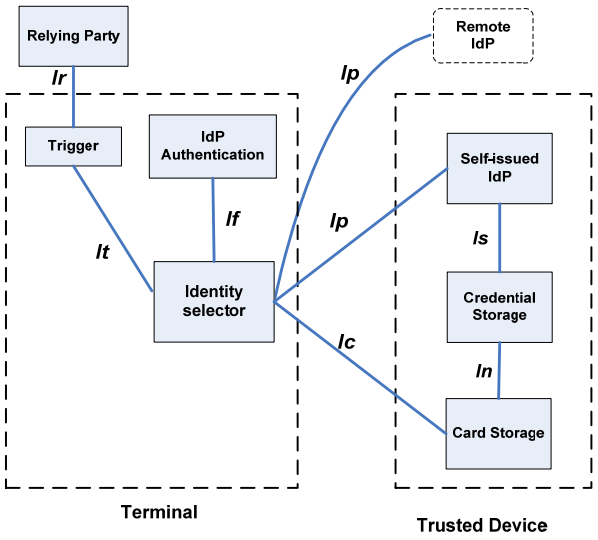


Figure 7: Configuration 2 – mobile identity provider

Configuration 4 - "all in trusted device": Next natural step is to have all the components in the trusted device. In this case, the user experience is the same as in configuration 0 except that now everything is done on the trusted device instead of the terminal. The difference between configuration 3 and 4 is that the end user both uses and authenticates to services in the trusted device, and this actually converges to the case where an identity metasystem is supported in the trusted device. Although this configuration is the most secure in the context of mobile identity metasystem, it is not applicable for every device because of their limited capabilities (storage space, memory, processing power, etc.).

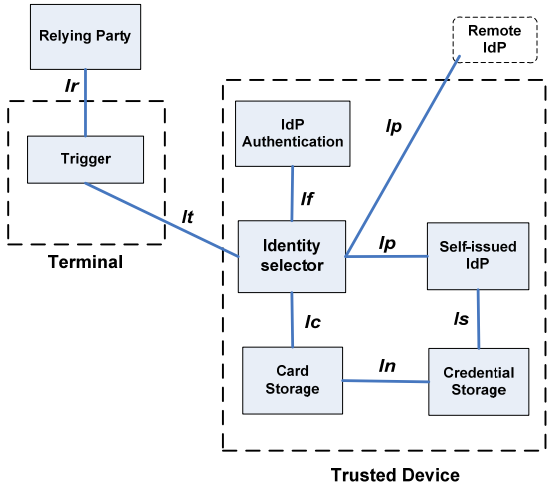


Figure 8: Configuration 3 – mobile identity selector

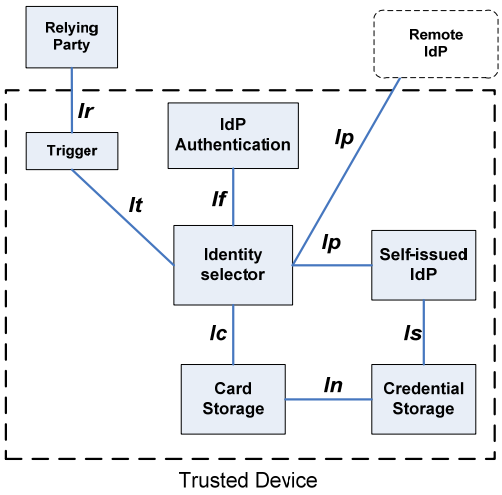


Figure 9: Configuration 4 – all in trusted device

Configuration 5 - "external service operator": This is a special configuration which is not possible with current technologies but we still describe it because it might be an interesting case in future. In this scenario, all of user's essential data (personal and managed information cards, user credentials, and profile settings) are stored on a network server; the user device only operates as an access terminal for the end user. We assume that there is an external service operator providing personal identity management services via (mobile) network. For example, when an end user is accessing a service using his trusted device, the device displays his information cards which have been loaded on demand from a remote server. In the case, if the end user loses his trusted device, he just goes to his personal identity management service, logs in, and locks his profile usage for the lost device. The lost device poses no threat as there is no sensitive information stored on it.

4. IMPLEMENTATION

In the configurations listed in Section 3 there are two connected systems: one on a terminal and one on a trusted device; each system contains some of the identity components (e.g. identity selector, credential storage) and each of those components are expected to cooperate seamlessly regardless of whether they are located in the same platform or distributed across different systems. For example, authentication process in section 2.1 may be executed using any of the configurations listed in section 3.2 in which one identity component should be aware of others components. We have implemented the middleware for enabling seamless communication between the identity components. The result is that the main target **secure roaming of identity** becomes possible (with configurations 2, and 3).

4.1 High Level Architecture

The main idea is based on the Web Services model [17]. One identity component hosted by an identity system is considered a "service" of that system, and it can be discovered by other identity components. Components/services communicate using SOAP messages so that they can be implementation language and platform independent. In this section we propose an extension to Novell's Bandit project which supports configurations specified in Section 3.2. In our work, we adapt the implementation of -

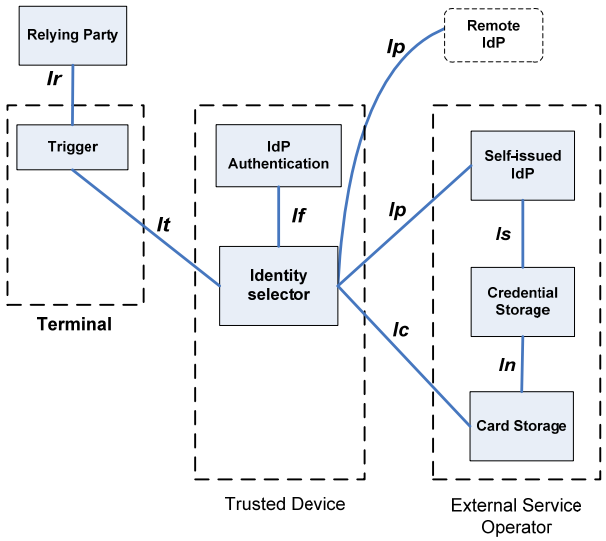


Figure 10: Configuration 5 – external service operator

identity components from the Bandit project to our middleware. Open-source identity components from other identity metasystem implementation can also fit into our middleware. Figure 11 shows the high-level architecture of our middleware. It is noted that so far only WS-Trust [18] is supported. Other methods, OpenID [19] for example, can be added to the system as an extension which can be considered in future study.

Connectivity Endpoint: Connectivity endpoint is used for transport-level messaging. It is also used for service broadcasting and discovery. Our middleware can support various types of endpoints (Bluetooth, USB, IP, IrDA, etc.). Each endpoint is controlled by a *Session Manager*.

Session Manager: This module handles the state of the each session. It maintains a list of connected target devices and their service profiles. A service profile contains information such as the unique component container identifier, the negotiated configuration, and the transport media being used. In addition, as a gateway for all incoming/outgoing messages, *Session Manager* can be a firewall, only allowing a certain set of configured operations from certain sources. For example in configuration 2, the trusted device allows only operations defined for ***Ip*** and ***Ic*** interfaces sent from the terminal.

Repository Service: This module maintains a lookup table of references to identity components. The content of the lookup table is updated when two systems negotiate and configure their service profiles to work together or when a system disconnects. The purpose of the repository service is for dynamic communication in the case where we do not want to work with fixed, implicit configurations.

Component Container: This module wraps identity components that are running on the local system. Each Component Container is marked with a unique identifier so that one identity component can be identified uniquely among connected systems. For example one identity component can be addressed using naming convention as follows:

"*<Container identifier> - <Component name>*".

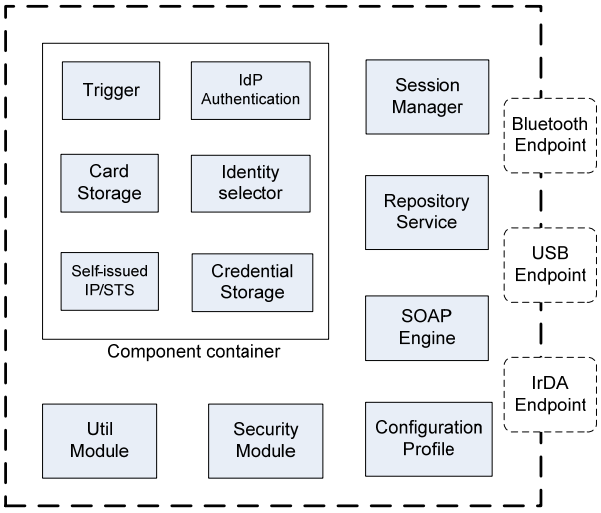


Figure 11: High-level system architecture

SOAP Engine: This module encapsulates requests from local identity components into SOAP requests [16] and parses SOAP responses sent from remote identity components.

Configuration Profile: This module manages the service profile of the local system. It is useful when there is a need to store/retrieve persistent attributes of the local system (e.g. system identifier).

Utility Module: The utility module provides some common functions for other modules (e.g. converter, encoder/decoder or file manipulation) as well as basic cryptography functions.

One user session goes through the following four phases:

Phase 1 - Connection setup: Before exchanging messages, the two systems must be able to locate each other and discover their offered services. This phase depends on the discovery mode of user's trusted device. In *active mode*, the device broadcasts its profile and users can do pairing using the terminal. In *listen mode*, the terminal broadcast its service and users do pairing using the trusted device. We assume there are discovery mechanisms that can be used by our system (for example, Devices Profile for Web Services [20]).

Phase 2 - Session initiation: User confirms the connection setup (either on the trusted device or on the terminal, depending on the working mode), and optionally enters additional security code or PINs if required. After that, the session manager modules and the repository service modules on both systems are updated so that identity components can locate the correct target components later. The terminal and the trusted device are now ready to exchange messages.

Phase 3 - Message exchange: Before sending out a request message, one identity component queries the *Repository Service* module for the target components and forms the request to be sent to the local *Session Manager*. The structure of the SOAP request is composed as follows:

```
<message>
    <request>
        <source>componentX</source>
        <target>componentY</target>
        <operation>OperationABC</operation>
        <params>
            ParamsXYZ
        </params>
    </request>
</message>
```

The *Session Manager* determines how to reach the target component(s): if the target component belongs to local *Component Container*, the message is forwarded to that container which hosts the target component. Otherwise, the *Session Manager* passes the request to its *Endpoint* to be transferred to the remote system. The *Session Manager* on the remote machine receives the request from its *Endpoint* and delivers it to the *Component Container* hosting the target component. The container finally passes the request to the target identity component for processing. The response message follows the same path where the source and target have switched their places. The structure of the SOAP response is composed as follows:

```
<message>
    <response>
        <source>componentX</source>
        <target>componentY</target>
        <operation>OperationABC</operation>
        <status>error_code</status>
        <params>
            ParamsXYZ
        </params>
    </response>
</message>
```

Phase 4 – Session termination: In this final phase, user's metadata information and any temporary data including session information, cache, and history data on the terminal are cleaned up. After the session termination the terminal should turn into its default configuration mode (i.e., configuration 0).

4.2 Implementation Environment

We have implemented a prototype of our proposed extensions. In this section we describe the application flow of configuration 2 (mobile identity provider). On the terminal side, we use the identity components from Novell's Bandit project (implemented in C++ language) and wrap them in our middleware. On trusted device side, we decided to use J2ME [21] as a platform. This is both an advantage and a challenge since J2ME is widely supported on mobile phones but has a limited set of features. We have implemented the identity components for the J2ME environment from scratch, used open-source libraries such as kXML [22] for XML document processing and Bouncy Castle [23] for cryptographic operations.

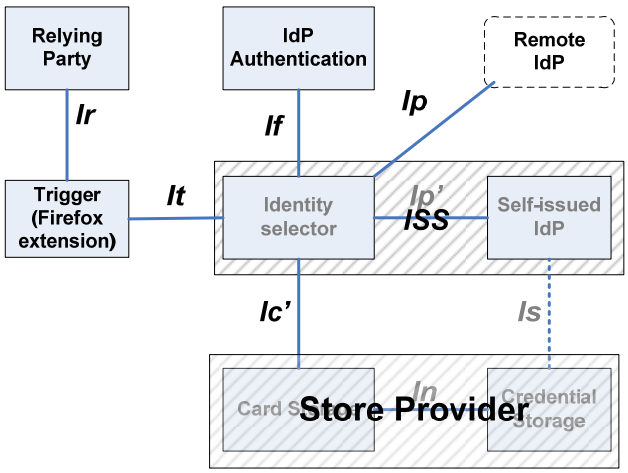


Figure 12: Bandit implementation

Let us start with the current implementation of Bandit project. Figure 12 shows Bandit's high-level design. The main module *ISS* loads complete set of available i–card documents [24] from the *Store Provider* through ***Ic'***. To get a self-issued token, the selector passes an i-card object together with other parameters to the self-issued IdP module. This design causes security problems when we move to distributed environment because the identity selector has access to all secret information and keeps the loaded i-card object in memory. Besides, we have observed that the card enumeration is only needed in 3 cases. The first one is when populating and displaying user's information cards, which only needs card's metadata such as name, id, and picture. The second case is to import/export cards and this should be done directly from the storage component instead of selector component. The last case is to get the claim values and secret data in order to generate a self-issued security token. In this case, the self-issued IdP should have a direct and secure channel the credential store to access user's private data (***Is*** in our design).

Therefore, we have made changes to Bandit's components and wrapped them in our architecture as described in 4.1. Table A2 compares the changes between the Bandit implementation and our current implementation.

In configuration 2, the user's information cards (self-issued or managed) are stored as i-card documents on the trusted device side. ***Ic***-operations which involve card enumeration now only work with the metadata. We have defined some *filtering rules* to be used by Card Storage component to expose only metadata part of information card content (denoted by i-card* in Table A2). When the trusted device connects to a public terminal, only the public portion of i-card structure is transferred to the terminal so that end user can use their information cards on the terminal identity selector. The main idea is to give only non-sensitive metadata section of the i-card document to the terminal and let the terminal provide the user interface functions. All "metacards" and other temporary data such as history, cache, session on the terminal will be deleted when the trusted device disconnects.

For the *addCard*, *editCard* and *removeCard* operations, in the current implementation we assume that user can only add or edit cards from a trusted terminal like a home PC (or on the trusted device itself). These operations require that the terminal has been explicitly declared trusted by the end user in configurations where ***Ic*** is a cross-system interface (configuration 2 for example).

When a relying party asks for a security token, Bandit's DigitalMe user interface on the terminal collects RP's certificate, policies, and requested claims. If the security token specified on RP's policy is expected to be from a managed identity provider, the authentication process is done normally as described in [3]. If the token is expected to be self-issued, the identity selector component constructs a request and sends it to the self-issued IdP component on the trusted device using our middleware. Once the self-issued IdP component receives the request, it generates a SAML token [25] with relevant attribute values extracted from the card store on the trusted device through ***Is***. The security token is signed with a RP specific private key and encrypted by the public key of the RP before sending it back to the terminal and subsequently passed to the RP.

As described above, although the role of the trusted device in the authentication process is important, its functionality is quite simple. Most of complex processing including collecting certificate, user interface and service execution are done in the public terminal. The heaviest processing is to generate a specific RP-card key pair. Ideally, it should be done in the IdP in the trusted device. However, if the trusted device has limited computational power, RP-specific key pairs may be generated on a trusted terminal. Each RP specific key pair needs to be generated only once and transferred to the trusted device. One way to store the generated key pairs for later use is to put them in a new XML extension of the i-card format to store private data.

5. CONCLUSION

The notion of using a trusted device to secure transactions on a less trusted terminal is well-known [27][28][29] . However, these methods propose their own protocols/frameworks, making it difficult to interoperate with other identity systems. The distinguishing feature of our work is that we have shown how an existing identity metasystem can be extended to support the use of a trusted terminal. Our contribution is two fold: (1) specifying the architecture for identity metasystem implementation that makes it easy to distribute identity components; and (2) using (1) to implement a two-device configuration which enables digital identity roaming across security domains. To summarize, we are able to move to the next level of current dimension of identity metasystem [30]. In our current work, we are planning to extend our implementation to support configuration 3 (mobile identity selector) and configuration 4 (all in mobile device) on various types of devices. We plan to support not only J2ME platform but also the Symbian operating system [26] because Symbian provides a strong platform for security processing plus a very good native user interface. In addition, we are developing mechanisms for securing inter-component communication which can be used instead of or in addition to any transport-level secure communication mechanisms. We also intend to study the usability and system performance aspects with the intention of improving the overall user experience. The final target is to give the end users better security and richer user experience.

6. REFERENCES

[1] Microsoft organization. Microsoft's Vision for an Identity Metasystem. Microsoft Whitepaper, May 2005. http://msdn2.microsoft.com/en-us/library/ms996422.aspxhttp://zoo.cs.yale.edu/classes/cs457/tsui_digital_identity_management.doc

[2] David Chappell. Introducing Windows CardSpace. Windows Vista Technical Articles, April 2006 http://msdn2.microsoft.com/en-us/library/aa480189.aspx

[3] Arun Nanda, Microsoft Corporation. Identity Selector Interoperability Profile V1.0 April, 2007 http://www.microsoft.com/downloads/details.aspx?FamilyID=B94817FC-3991-4DD0-8E85-B73E626F6764&displaylang=en

[4] Michael B. Jones. The Identity Metasystem: A User-Centric, Inclusive Web Authentication Solution. W3C Workshop on Transparency and Usability of Web Authentication. New York City, March 2006

[5] CodeIdol.com. InfoCard Architecture and Security http://codeidol.com/csharp/indigo/InfoCard/InfoCard-Architecture-and-Security/

[6] Novell corp. Bandit project http://www.bandit-project.org/index.php/Welcome_to_Bandit

[7] Kim Cameron. The Laws of Identity. Microsoft Whitepaper, May 2005. http://www.identityblog.com/stories/2004/12/09/thelaws.html

[8] Microsoft organization. Windows Data Protection. Microsoft MSDN, 2001. http://msdn2.microsoft.com/en-us/library/ms995355.aspx

[9] Higgins Project. Higgins home page http://www.eclipse.org/higgins/

[10] The Official Bluetooth® Technology Info Site http://www.bluetooth.com/bluetooth/

[11] Web Services MetadataExchange. August 2006. http://schemas.xmlsoap.org/ws/2004/09/mex/

[12] Web Services Policy Framework. March 2006. http://schemas.xmlsoap.org/ws/2004/09/policy/

[13] Massachusetts Institute of Technology. Kerberos: The Network Authentication Protocol http://web.mit.edu/Kerberos/

[14] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280. April 2002. http://tools.ietf.org/html/rfc3280

[15] Sun Microsystem. Remote Procedure Call. Request for comments 1831. August 1995. http://tools.ietf.org/html/rfc1831

[16] W3C Working Group. Simple Object Access Protocol (SOAP). http://www.w3.org/TR/soap/

[17] W3C Working Group. Web Services Architecture. February 2004 http://www.w3.org/TR/ws-arch/

[18] OASIS. WS-Trust Version 1.3. March 2007 http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html

[19] OpenID community. OpenID Authentication 2.0 - Draft 11. 2007. http://www.openid.net/specs/openid-authentication-2_0-11.html

[20] Microsoft organization. Devices Profile for Web Services February 2006 http://schemas.xmlsoap.org/ws/2006/02/devprof/

[21] Sun Microsystem. Java 2 Platform, Micro Edition (J2ME). Java ME homepage http://java.sun.com/javame/index.jsp

[22] kXML A small XML pull parser http://kxml.sourceforge.net/kxml2/

[23] The Legion of the Bouncy Castle. Bouncy Castle Java cryptography API http://www.bouncycastle.org/java.html

[24] Information card format http://en.wikipedia.org/wiki/I-Card

[25] OASIS. SAML Token Profile Version 1.1. February 2006 http://docs.oasis-open.org/wss/oasis-wss-SAMLTokenProfile-1.1

[26] Symbian Ltd. Symbian OS 2003 http://www.symbian.com

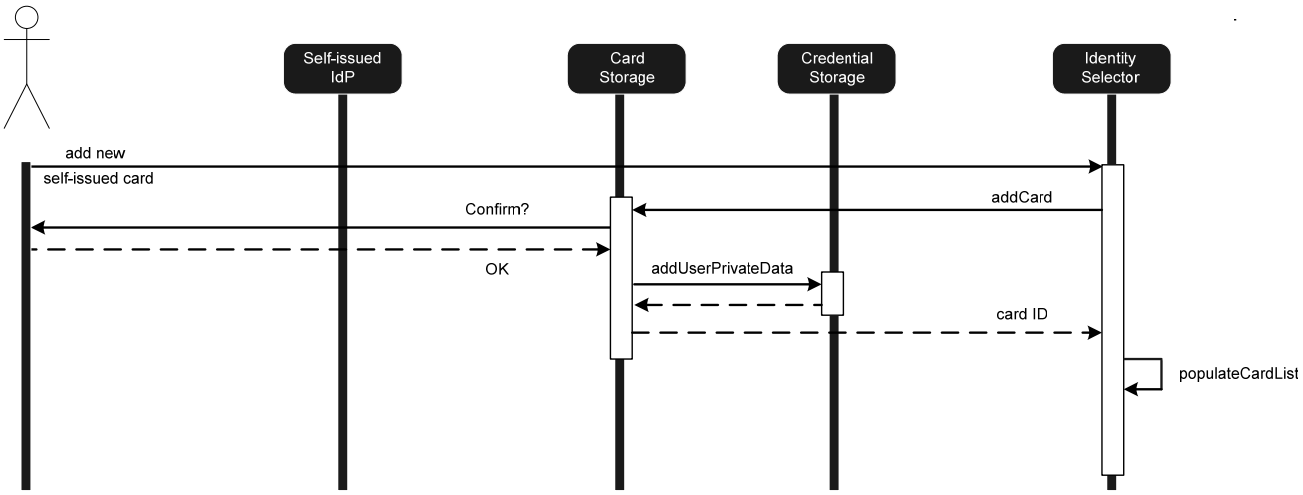
[27] B. Parno, C. Kuo, and A. Perrig., Phoolproof phishing. In Proceedings of Financial Cryptography and Data Security 2006, Lecture Notes in Computer Science 4107, Springer.

[28] M. Mannan, P.C. van Oorschot, Using a Personal Device to Strengthen Password Authentication from an Untrusted Computer. In Proceedings of Financial Cryptography and Data Security 2007.

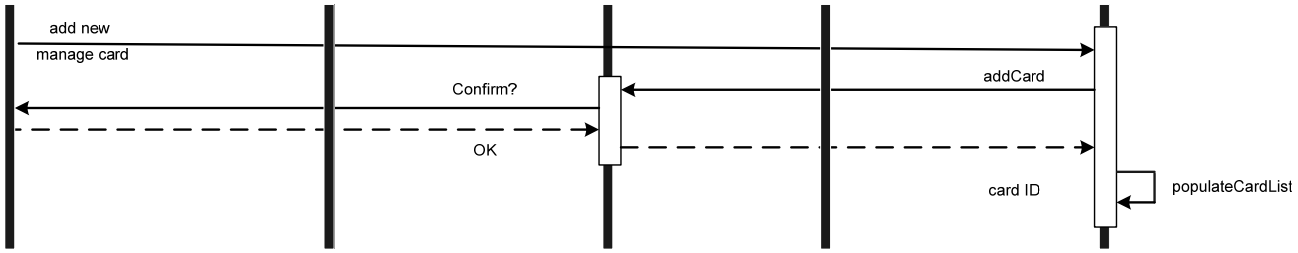
[29] D. Balfanz, E. Felten, Hand-held computers can be better smart cards. In Proceedings,8th conference on USENIX Security Symposium - Volume 8, 1999.

[30] Axel Nennker. The CardSpace dimensions. Published on Ignisvulpis's blog June 2007. http://ignisvulpis.blogspot.com/

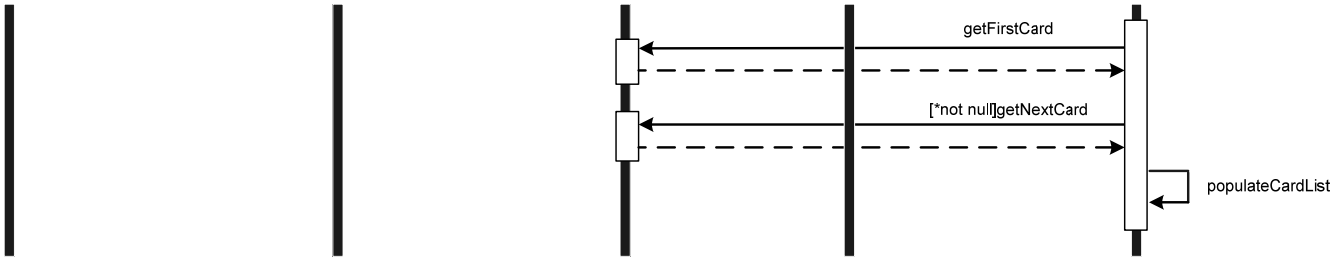
APPENDIX



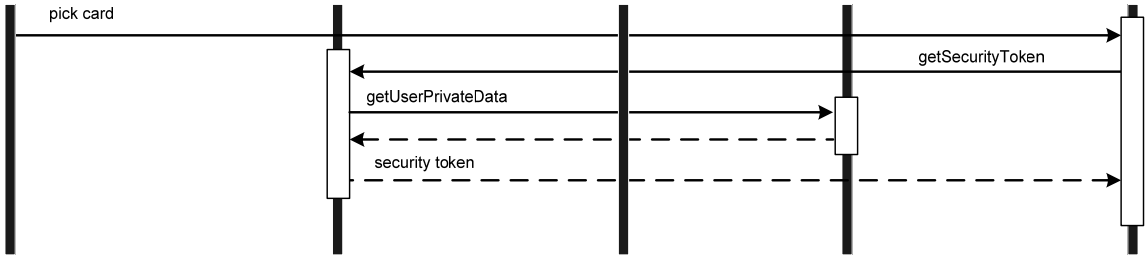
a) User adds a self-issued card



b) User adds a managed card



c) User interface of identify selector populates list of information cards



d) User picks a card

Figure A1: Interaction between identity components

Table A1: Interface operations

Interface	Operation	Input	Output	Description
Ip	*getSecurityToken*	token type , relying party, credential[1] required claims optional claims[2]	display token, security token	Get security token from identity provider
Ic	*getFirstCard*	N/A	i-card*[3]	Get first card from store
	getNextCard	N/A	i-card*[3]	Get next card from store
	getCard	N/A	i-card*[3]	Get specific card from store
	addCard	i-card[4]	card ID	Add one card to store
	editCard	card ID, i-card[4]	N/A	Modify one card
	removeCard	card ID	N/A	Delete one card from store
Ir	*activateSelector*	x-informationCard	security token	Browser extension HTML parsing
It	*getToken*	token type , relying party, required claims optional claims	security token	Get security token from selector then pass it to relying party.
	manageCards	N/A	N/A	Open selector's user interface for card management
If	*getCredential*	N/A	credential	Obtain credential from user to authenticate to IdP
Is	*getUserPrivateData*	card ID	private data	Manipulate claim values and other info such as key-pair, master key, PIN.
	addUserPrivateData	card ID, private data	N/A	
	editUserPrivateData	card ID, private data	N/A	
	removeUserPrivateData	card ID, private data type	N/A	
In	*getCardContent*	card ID	i-card[4]	called when exporting cards
	addUserPrivateData	card ID, private data	N/A	called when importing cards, adding personal cards
	editUserPrivateData	card ID, private data	N/A	called when edit personal cards
	removeUserPrivateData	card ID, private data type	N/A	

[1] Credential being used to authenticate to identity provider

[2] Optional claims which are selected to be send by end user. These claims are subset of optional claims stated in RP's policy

[3] i-card* denotes filtered i-card document with only public metadata section

[4] i-card denotes full i-card document

Table A2: Comparison of Bandit implementation and our implementation

Interface	Operation	Bandit implementation		Our implementation	
		Input	Output	Input	Output
Ip	*getSecurityToken*	token type , i-card, relying party, credential required claims optional claims	display token, security token	token type , ***i-card*,*** relying party, credential required claims optional claims	display token, security token
Ic	*getFirstCard*	N/A	i-card	N/A	***i-card****
	getNextCard	N/A	i-card	N/A	***i-card****
	getCard	card ID	i-card	card ID	***i-card****
	addCard	i-card	card ID	i-card	card ID
	editCard	card ID, i-card	N/A	card ID, i-card	N/A
	removeCard	card ID	N/A	card ID	N/A
It	*getSecurityToken*	token type , relying party, required claims optional claims	security token	token type , relying party, required claims optional claims	security token
	manageCards	N/A	N/A	N/A	N/A
If	*getCredential*	N/A	credential	N/A	credential
Is	*getUserPrivateData*				
	addUserPrivateData				
	editUserPrivateData				
	removeUserPrivateData				
In	*getCardContent*				
	addUserPrivateData				
	editUserPrivateData				
	removeUserPrivateData				

Require access control for cross device boundaries.

Haven't been implemented yet. In current implementation, ***Is*** and ***In***-operations are replaced by *getCard*, *addCard* with full i-card document

Secure Roaming with Identity Metasystems

Long Nguyen Hoang, Helsinki University of Technology
***Pekka Laitinen, Nokia Research Center**
N. Asokan, Nokia Research Center

Outline

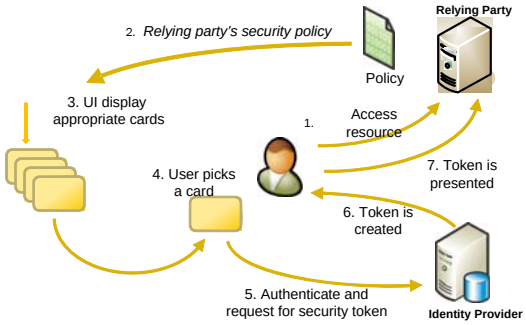
- *What is identity metasystem?*
- *Why we need roaming?*
- *Roaming configurations*
- *Implementation*
- *Conclusion*

Outline

- **What is identity metasystem?**
- Why we need roaming?
- Roaming configurations
- Implementation
- Conclusion

What is identity metasystem?

- provides abstract identity management layer
  - allows inter-operability between different identity systems
- provides consistent user experience
  - introduces information cards
- existing implementations
  - CardSpace by Microsoft
  - DigitalMe by Bandit project

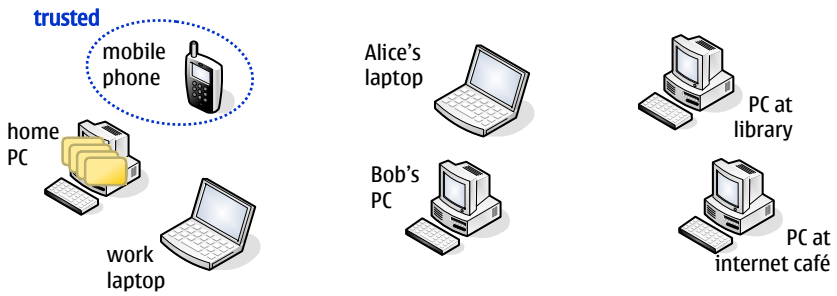


Outline

- *What is identity metasystem?*
- ***Why we need roaming?***
- *Roaming configurations*
- *Implementation*
- *Conclusion*

Why do we need roaming?

*How may **different devices** do you use to **access** Internet services?*



*You want to **use** your information cards in **all of them** !*

Solutions for roaming, part 1

Existing solution:
export/import information cards

- *bad security: requires trusting all classes of devices at same level*
- *bad usability: export-import-remove cards*

Solutions for roaming, part 2

Potential solution:
smart card or any removable hardware token

- *good security: can manage different trust levels between device classes*
- *good usability: insert smart card to PC and information cards become available*
- *bad security: no user interface on smart card*
- *just bad: need extra hardware*

Solutions for roaming, part 3

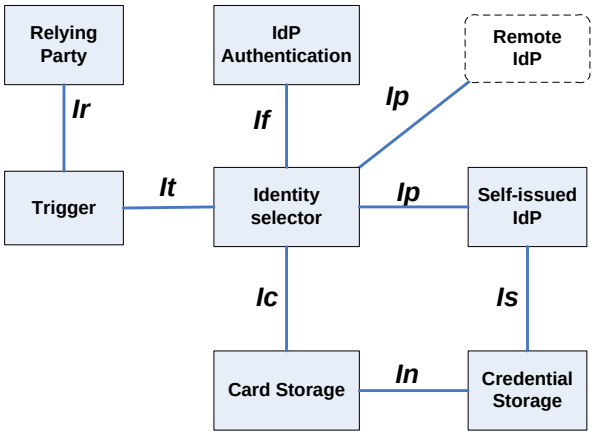
"The right solution":
mobile phone as personal trusted device

- *good security: can manage different trust levels between device classes*
- *good usability: connect phone to PC and information cards become available*
- *good security: has user interface*
- *just good: everybody has mobile phone*

Outline

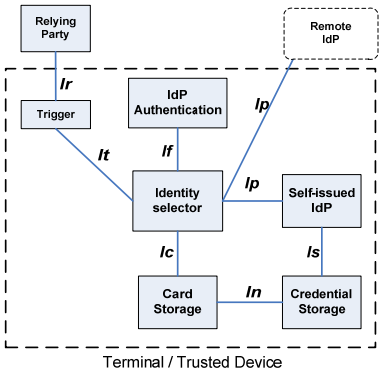
- *What is identity metasystem?*
- *Why we need roaming?*
- ***Roaming configurations***
- *Implementation*
- *Conclusion*

General architecture of identity metasystems

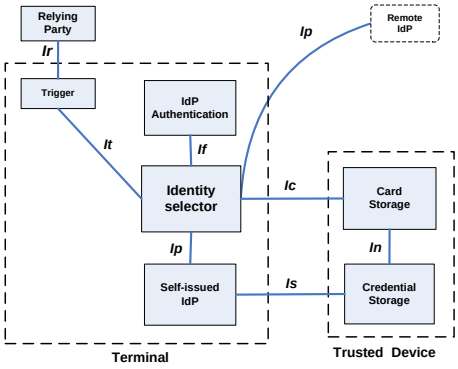


NOKIA Connecting People

Distribution of components



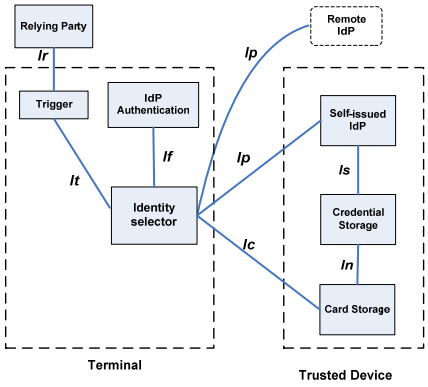
Configurations 0:
All in one device



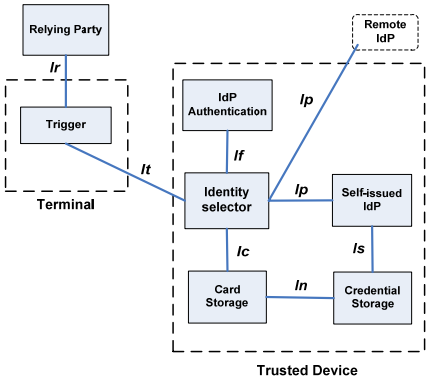
Configuration 1:
External storage

NOKIA Connecting People

Distribution of components, continued



Configuration 2:
Mobile identity provider

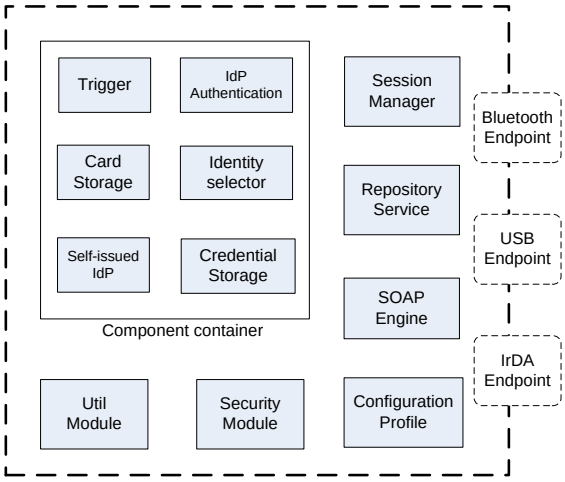


Configuration 3:
Mobile identity selector

Outline

- *What is identity metasystem?*
- *Why we need roaming?*
- *Roaming configurations*
- ***Implementation***
- *Conclusion*

Distributable system architecture



NOKIA Connecting People

Implementation

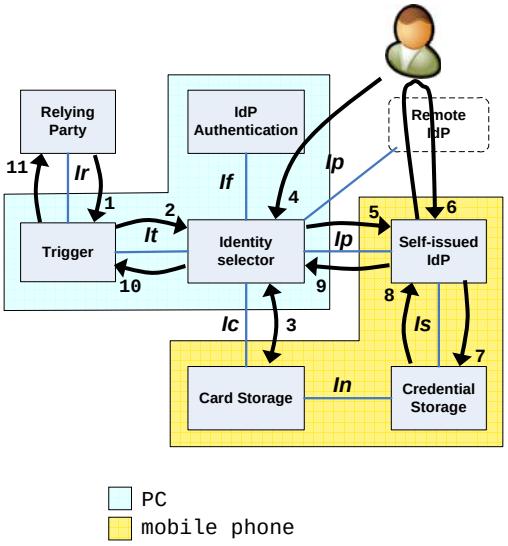
- ***Configuration 2*** *has been implemented*
- ***Middleware***
  - *allows distribution of components*
  - *components are loosely coupled*
  - *SOAP messaging between components*
- ***DigitalMe*** *(modified from version 0.4.1309) on Linux*
  - *BlueZ Bluetooth driver*
- ***Java ME*** *(MIDP 2.0) on mobile phone*
  - *Bouncy Castle's crypto library*
  - *kXML for XML processing*
- ***Connection*** *endpoints*
  - *Bluetooth*
  - *IP*

NOKIA Connecting People

Example sequence flow

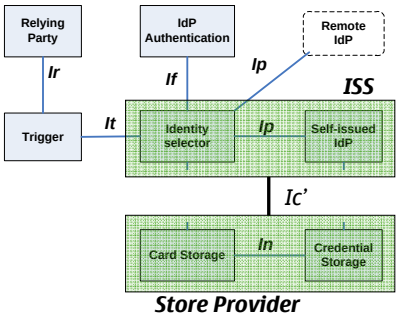
1. *RP sends its policy.*
2. *Trigger forwards RP's policy to Selector.*
3. *Selector fetches card metadata from Card Storage and displays them in Selector's UI.*
4. *User selects a card.*
5. *Security token is requested from Self-issued IdP.*
6. *User is asked to approve security token request.*
7. *IdP requests credentials from Credential Storage.*
8. *Credentials are returned; IdP generates RP specific key pair and creates security token.*

9-11. *Security token is forwarded to RP.*



NOKIA Connecting People

Implementation issues



Issues with DigitalMe:

- *Components are tightly coupled (ISS and store provider)*
- *ISS handles master key and RP specific key pair generation (master key sent over Ic' interface)*
  - *→We adapted DigitalMe to our distributable architecture*

Issues with mobile phone:

- *RSA key generation takes time*
  - *→ RP specific key pair should be stored to avoid unnecessary regeneration*

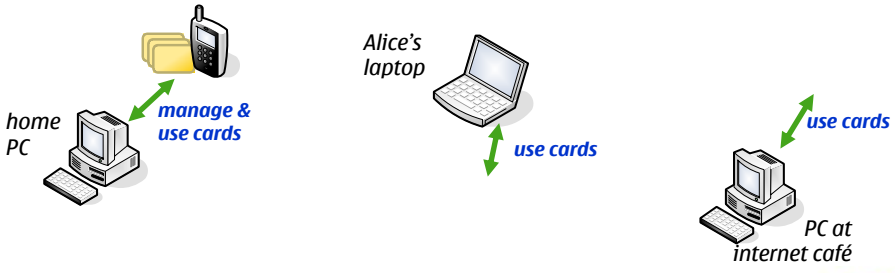
NOKIA Connecting People

Outline

- *What is identity metasystem?*
- *Why we need roaming?*
- *Roaming configurations*
- *Implementation*
- ***Conclusion***

Conclusion

designed and implemented ***architecture*** *that*
enables ***distribution*** *of*
identity metasystem ***components***
to ***two*** *(or more) terminals*



Thank you!

Questions?

Secure communication for ad-hoc, federated groups

Andreas Sjöholm
Swedish Institute of Computer Science, Box 1264, 164 29 Kista, Sweden
Axiomatics AB, Electrum 223, 164 40 Kista, Sweden
andreas@axiomatics.com

Ludwig Seitz
Swedish Institute of Computer Science
Box 1263, SE-16429 Kista, Sweden
ludwig@sics.se

Babak Sadighi
Swedish Institute of Computer Science, Box 1264, 164 29 Kista, Sweden
Axiomatics AB, Electrum 223, 164 40 Kista, Sweden
babak@axiomatics.com

ABSTRACT

Ad-hoc federated groups are getting increasingly popular as means of addressing collaborative tasks that require information sharing. However, in some application scenarios, the security of the shared information is vital. Managing the communication security of such groups in an efficient way is a difficult task.

This paper presents an architecture that enables secure communication for ad-hoc, cross-organisational groups. Our architecture covers group admission control, group key management and secure group communication. The groups in question are expected to be ad-hoc groups where the potential participants have no prior knowledge of each other and thus federation mechanisms need to be used to establish group admission rights. In order to handle group admission we use the SAML and XACML standards, for group key management we use the TGDH protocol. Our approach thus supports decentralised management of the most important tasks in secure group communication using an integrated approach based on established security standards. We have also produced a demo implementation to show the feasibility of our architecture.

This research was pursued as part of the TrustDis project funded by the Swedish Governmental Agency for Innovation Systems (Vinnova).

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and protection

Keywords

Access Control, Tree-based Group Diffie-Hellman, Secure Group Communication

1. INTRODUCTION

The core concept of the Internet has always been to share information distributed among different locations. As Virtual Organisations became increasingly popular, a new aspect of information sharing got into the focus of attention: cross-organisational, ad-hoc collaborations. Today clients for such applications include the scientific community, business and the military. Even other public organisations (e.g., health-care, firefighters, police) are increasingly interested in tools facilitating ad-hoc collaboration for achieving common goals.

As an application scenario, imagine the occurrence of a major traffic incident on a major public road. Several vehicles are involved in a pile-up, including trucks with dangerous chemicals. The harsh weather is complicating the rescuers work. A police car is quickly on site followed by additional rescue workers such as ambulances and firefighters.

It soon becomes apparent that many other public and governmental authorities and agencies such as the road administration authority, the national meteorological institute, a foreign embassy and the hazardous materials unit must be involved as their competence are necessary or they need to be informed. These actors need to share information, coordinating their efforts and avoiding fragmentation.

Such ad-hoc collaborations are expected to work without extensive infrastructure and formation should require minimal set-up time by being parallel to informing a group of actors. This type of collaborative group can be expected to accommodate for several hundreds of users spread amongst different organisations with little or no knowledge of each other except for a goal or interest in common.

Frequently, information sharing in such heterogeneous groups is subject to security related questions such as confidentiality and integrity. While these questions have clear solutions in client-server and even decentralised architectures, implementing the same solutions for ad-hoc groups without loosing the advantages of ad-hoc group collaboration is far from obvious.

The lack of a controlled infrastructure and a central authority in an ad-hoc collaborative environment makes it difficult to maintain both security and manageability. Who can join and be part of a collaborative network? How can sensitive information be exchanged in a group of mostly unknown members? How can a user, away from his home network, be authenticated? Who enforces the rules of the network? How can administration be kept manageable under such circumstances?

A solution which addresses the questions above, applicable to large ad-hoc groups, is needed. This paper presents a novel architecture that deals with the problems of authori-

IDtrust '08, March 4-6, 2008 Gaithersburg, MD

sation, authentication, confidentiality as well as availability in such groups.

The rest of this paper is structured as follows: In section 2 we define the problem at hand. Section 3 presents related work. In section 4 we present our architecture designed to solve the problems. Our demo implementation of the architecture is presented in section 5. We then discuss the advantages and drawbacks of the architecture in section 6. Finally we summarise, give a conclusion and point to future work in section 7.

2. PROBLEM STATEMENT

The two main problems we face with respect to secure group communication are[1]:

- Who may join the group?
- How can the group members share cryptographic keys in order to ensure confidentiality and message integrity?

A sensible solution to the first problem is a combination of federated identity management and a policy based access control system. This requires a Policy Decision Point (PDP) which issues access control decisions, Policy Enforcement Points (PEP) that enforce the PDP's decisions, and Policy Information Points (PIP) that provide and the policies used by the PDP [14]. The federated identity management system is needed in order to provide certified information about the user (e.g., roles, affiliations, qualifications) to the PDP.

A common solution to the second problem are group key agreement schemes that allow all participants of the group to compute a common group key. Such a scheme needs to provide new group keys to the members in reaction to the following events [9]:

- Single member join: A new member joins the group.
- Single member leave: A current member leaves the group.
- Group merge: Two groups merge into one.
- Group partition: A group is split into new groups.

In order to be cryptographically secure, the key agreement scheme needs to fulfil the following cryptographic requirement [9]:

Definition 1. Key Independence:
Assuming that the key agreement scheme has produced the sequence of keys $K = \{k_1, k_2, k_3, \ldots, k_m\}$ in reaction to a sequence of group events.

A passive adversary who knows a proper subset of group keys $K' \subset K$ cannot discover any other group key $k_i \in (K \setminus K')$.

This requirement also ensures that a new group member can not decrypt any of the messages sent before the join event, and an ex-member can not decrypt any messages sent after the leave event (*Forward and Backward Secrecy*).

Secondary problems that need to be addressed are:

- Scalability: The group communication architecture must scale well with the number of users.
- Full representation of the group: All members shall have a complete, consistent representation of the group and its members.
- Distributed functionality: It must be possible to distribute functionality (such as PDPs), thus preventing single points of failures.
- Implementation of leader role with possibility of delegation: A group leader shall exist with the role of creating the policies that regulate group admission control. It shall be possible to issue constrained delegations of this role.
- Fault tolerance and self healing: Link disruption must not jeopardise the security and operability of the group more than that the functionality can be restored when the link is restored. During disruption when members are lost or cut off from each other, the group must be functional as partitioned and autonomous groups.

Summarising one can say, that the problem is to find an architecture that is able to solve all of these problems in an integrated, efficient, and decentralised way.

3. RELATED WORK

We present related work in the areas of group key agreement schemes and of group admission control. Further we present related work that combines both to create secure group communication solutions. jj A large number of group key management systems use trusted third parties (TTPs) to distribute the keys to the group members (e.g., [10, 15]). The use of a TTP greatly simplifies key management and update issues, due to the centralisation. However, the centralisation also disqualifies such schemes for our application scenario (we further discuss this in subsection 4.1).

In the area of contributory group key agreement schemes, the Tree-based Group Diffie-Hellman (TGDH) protocol by Kim, Perrig, and Tsudik [9] is an approach that fulfils our requirements. Therefore we choose it as component of our architecture.

Furthermore the Dynamic SubTree (DST) group key agreement scheme by Mao *et al.* [11] could be considered, since its average time cost is less than TGDH's. However, DST uses the unrealistic assumption that group member departure times is known in advance.

Another promising approach is the PFMH tree based contributory group key agreement protocol suite (PACK) by Yu, Sun, and Liu [16]. PACK seems to have better theoretical performance than TGDH for single user join, while the author's simulation results suggest worse performance for single user leave.

Kim, Mazzocchi, and Tsudik propose a number of approaches for admission control in peer groups [8]. The paper's main focus is on admission by voting whereas the use of access control mechanisms is limited to Access Control Lists.

The commercial product *MindAlign* from *Parlano* advertises secure communication for federated groups. No detailed information on the technical details could be obtained[1].

[1]Actually since Parlano has been acquired by Microsoft, there is no information at all available from the Parlano website, our main source of information was http://en.wikipedia.org/wiki/MindAlign.

Judge and Ammar propose a Group Access Control Architecture for Secure Multicast and Anycast [7]. This architecture provides group policy management, member authorisation and group key management. However, their key agreement scheme is not *key independent*. Moreover, the access control engine used in this architecture is not further specified.

Agarwal *et al.* suggest an integrated solution for secure group communication [2]. This approach uses GDH, the predecessor of TGDH for group key agreement. Furthermore the non-standard access control system Akenti is used to determine group admission. Sine work both on this approach and on Akenti appears to have ceased (last update of Akenti was 2005), this does not seem like a promising candidate for solving our problem.

We can therefore summarise, that a number of promising approaches for contributory group key agreement schemes exist. Their main difference lies in the performance optimisation of specific group events. Thus alternative schemes can be investigated, should future experiments indicate performance problems with our TGDH based approach.

As for comprehensive architectures including group admission control, related work seems to either rudimentary or based on outdated, non-standard access control technology.

4. SECURITY ARCHITECTURE

In this section we present our architecture for secure group communication *TgdhXacml* and explain some of the choices we made.

4.1 Contributory key schemes versus TTPs

As we pointed out in section 3 our main problem with TTPs is their centralisation. Setting up a TTP and reconfiguring it when the group structure changes drastically, is bound to introduce delays in availability.

TTPs availability can also become problematic when the number of users increases dramatically, furthermore the TTP can be a single point of failure and therefore a prime target for attacks. Finally the use of a TTP raises the question of who should provide and maintain this service, which can increase the level of distrust between group members and thus limit the information they are willing to provide. All these points speak against using a TTP in our ad-hoc, federated group scenario. Therefore we have not considered TTP based key management systems for our approach.

4.2 XACML and SAML

When choosing systems for authorisation and for supporting federated identity, our main objective was to use widely accepted standards. This choice stems from the facts that a range of different actors have to interoperate and to agree on a common authorisation model as well as both a direct (corporate's policy) and a indirect (through SOX[2]) requirement to exclusivly use open standards.

XACML [6] is a widely accepted access control standard with good support both from the industry and the scientific community [3]. Furthermore a range of useful support tools are available both specifically for XACML[3] and for processing XML.

Another determining factor in our choice of XACML is the fact that the upcoming version 3.0 of XACML [13] has support for delegation, a feature that is very important for our decentralised group management approach. The delegation mechanism allows authority holders to delegate a precisely constrained part of their authority to others, thus facilitating decentralised administration, without the risk of giving away too much authority.

Using these delegation features, the broad access control policies for group communication can be specified by a coordination authority, while power to create fine-grain authorisations (e.g., for specific subjects) can be delegated to the organisations participating in the group.

Using SAML [4] as assertion language for supporting federated identity is an obvious choice when combined with XACML. In order to evaluate a specific request, an XACML PDP needs to establish an evaluation context. This includes collecting attributes for the subject of the request. Providing such attributes in a secure way is a typical application for SAML.

Having chosen SAML as syntax and protocol suite for attribute assertions, an Attribute Authority (AA) is needed to generate the SAML assertions and to manage the federation of attributes. The Assertion Server[4] is a system closely integrated with XACML 3.0 and SAML. It's main function is to provide attribute assertions and to manage attributes. Besides this the Assertion Server handles attribute hierarchies, delegation of attribute authority and parallel administration of multiple sources of attribute authority. It can be queried according to the protocol defined in the SAML V2.0 [4] standard.

Internally it uses XACML policies to represent attribute value assignments (including attribute hierarchies) and attribute authority (including authority delegation), however a user never needs to see these internal policies.

Using the Assertion Server, a PDP can gather all necessary attributes in order to evaluate a request. Multiple Assertion Servers can serve one PDP and one Assertion Server can serve multiple PDPs.

Figure 1 shows an example set up with a policies and some attributes provided externally by an Assertion Server. All XACML policies have been greatly simplified to make them less verbose. In our application scenario from the introduction, a crisis coordinator has the authority over a group resources allocated to the incident, which has been named *Incident23*. He issues a root policy delegating authority over these resources to the *Police board*, however the authority constrains the possible subjects to members of the police force (*Delegated Subject* must have *organisation = Police*).

Alice who is a member of the police board, coordinates the police operation. In the second policy, she allows Bob who is the policeman on site to access said resources. In order to be valid, this policy needs to be supported by the root policy. This is the case, since the Assertion Server confirms both that Alice has the *role = Police board* attribute and Bob has the *organisation = Police* attribute.

4.3 Architecture

We now proceed to tie together the different components of our approach into an integrated architecture. The in-

[2]Sarbanes-Oxley Act

[3]see: http://www.oasis-open.org/committees/xacml under the heading *Additional Information*

[4]Available from: http://www.sics.se/spot/assertion_server.html

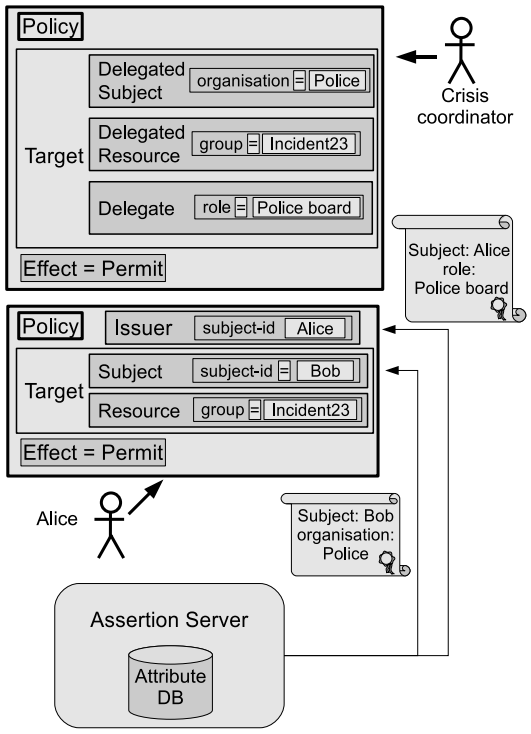


Figure 1: Example for a delegation and related attribute assertions provided by an Assertion Server.

tegrating component deployed with all group members is called the *context handler*. The purpose of the context handler is to coordinate information and integrate the services needed for TGDH and XACML in a ad-hoc environment. This means that a group member's context handler must implement the following functionality:

- Locate information for XACML contexts
  The most important functionality of the context handler. TgdhXacml stores a list of addresses where certain information and services can be found in the distributed group (such as PDPs and Assertion Servers). The TgdhXacml context handler uses these addresses to provide the access control component with additional information (e.g., policy databases, assertion server locations).

- Respond to queries about the TgdhXacml environment
  All members in the group should have the same information about services and newly joined members need to be updated on this knowledge. Therefore the context handler must be ready to provide the TGDH sponsor with information about the environment, such as lists of PDPs and Assertion Servers.

- Start up (additional) PDPs and Assertion Servers
  Upon creation of the group the group leader's context handler starts up one PDP, one Assertion Server and one PIP and publishes this information in the list of available services. Other members can later start additional modules if the workload is too high for a single member or just to distribute services in the group. This could be made voluntarily by asking a member to take responsibility for lowering the workload on a stressed member or to start up dummy members that do not participate in the group communication and just provide additional services.

- Synchronisation of policy and assertion databases
  Since it should not make a difference which PDP or Assertion Server a member is using, databases containing attribute and policy information must be synchronised. An approach would be to use a replicating database server. If the underlying distributed network layer supports data replication this feature can be used.

- Communication security
  The context handler gets provided with a group master key from the TGDH layer which allow it to encrypt and decrypt messages. The group master key is used as seed go generate an AES-256 content encryption key (CEK). Thus even if a CEK is compromised, the current group master key remains secure.

  Our TGDH layer also provides DSA keys for all group members, which are used for digital signatures. DSA has been chosen for message authenticity instead of a MAC algorithm because DSA is asymmetric and thus the DSA public key can be sent in clear text together with the join group request.

  The raw data communication protocol for the group is managed by the network layer.

- Implement API
  As the TgdhXacml provides no user functionality, applications are to be plugged in on top of the TgdhXacml. This can be any kind of service that provides or retrieves information.

Figure 2 shows an example layout of the TgdhXacml architecture for a group with four members. Every member M_i has a context handler, coordinating the other modules. All members are also running the TGDH protocol and a PEP. Additionally some members are running applications, PIPs and Assertion Servers.

In the example the members $M1$ and $M3$ provide an Assertion Server. The Assertion Server's attribute database needs to be synchronised between these two members. Members $M2$ and $M4$, both provide a PIP, therefore the policy database must also be synchronised. The top level of the architecture is the application layer.

Observe that member $M3$ is not running any application, which implies that $M3$ is a virtual member that was only set up to lower workload of the other members and increase availability of the attribute database.

4.4 Authentication

Since all Diffie-Hellman offspring lack authentication and are susceptible to active attacks (such as man-in-the-middle attacks), TgdhXacml needs a mechanism to authenticate group members. This authentication is to provide a unique identifier that can be used to bind attribute assertions to specific group members.

In order to prevent man-in-the-middle attacks, this authentication infrastructure needs to be set up over secure channels prior to joining the group. We use a X.509 v3

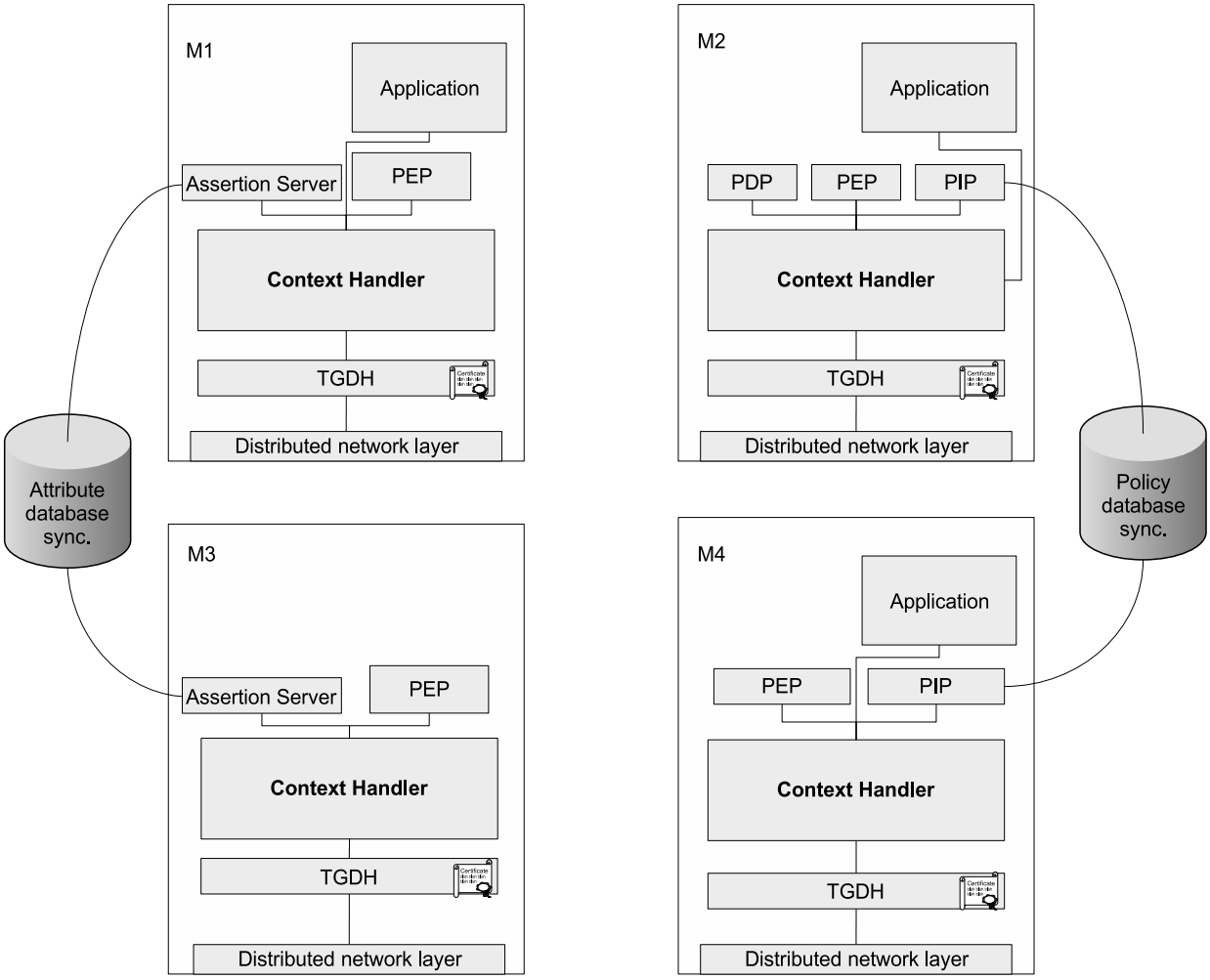


Figure 2: The architecture of our group communication system.

based public key infrastructure for our approach, because it is the most widespread standard and is supported in several open source libraries.

The group creator either generates a self-signed certificate or requests a certificate from a Certificate Authority (CA). Other members' certificates must be signed by a CA according to a certificate infrastructure policy. We then use the Distinguished Names (DN) from these certificates as subject identifiers in the XACML policies and SAML assertions.

In our application scenario it is reasonable to assume pre-existing authentication structures within the organisations participating in the group. Nevertheless chances are that incompatibilities between different authentication systems would require some further federation mechanisms in order to become interoperable. This however is out of the scope of this article.

4.5 The modified join protocol

This section describes the protocol used when a new member wants to join the group. It is slightly modified from the classical TGDH protocol in order to accommodate for the access control and the transfer of DSA public keys we added.

A join request must contain the TgdhXacml group name one wishes to join, the blinded key for the TGDH join-protocol, the potential member's DSA pubic key and authentication certificate.

The join request is broadcasted to all members of the group. One existing TgdhXacml context handler declares itself as the sponsor and further on only that context handler will react to the join message. The sponsor now checks the validity of the authentication certificate and requests an admission decision from a group PDP, in order to determine if the new member should be allowed to join. Thus the sponsor acts as PEP, creating a valid XACML request and then enforcing the PDP's decision.

If the join request is denied, the sponsor notifies the potential member and takes no further action. If the join request is permitted the sponsor sends a message containing vital tree information such as tree structure, blinded keys, sequence numbers, PDP and Assertion Server lists to the new member. The sponsor also blocks the whole TGDH tree and broadcasts an update message containing the new blinded keys in the tree.

Figure 3 shows a sequence diagram of a join.

4.6 Management

One important goal of this architecture is to support security management of ad-hoc collaborative groups. This mainly boils down to supporting the following tasks:

1. Controlling who may join a group based on attributes
2. Delegating administrative power
3. Controlling access to information at the application level

The first task is accomplished by creating XACML access control policies that define the conditions of group admission. After these have been defined, the job of deciding on group admission is automatically performed by the PDP. Updates of the group admission policies may be necessary, but they will certainly occur less frequently than actual group admission decisions.

Tools that support the creation of XACML policies exist[5], however analysing these and integrating them into our group management architecture is out of the scope of this paper.

The second task is supported by the XACML v3.0 Administrative Policy standard [13]. An example for the basic principle of delegation using administrative policies was shown in figure 1. A full XACML delegation policy is shown in appendix A.

The third task is supported by restricting access to the group key, used to encrypt all communication between group members. Thus only group members will have access to the applications available within the group. Furthermore these applications can be programmed to enforce additional access control policies towards members of the group trying to access them. Such application access control could also make use of the access control infrastructure already available within the group (PDPs, Assertion Servers, PIPs).

5. DEMO IMPLEMENTATION

In order to demonstrate the feasibility of our architecture, we have produced a demo implementation for the main elements of our architecture. This includes the context handler, the TGDH layer, XACML PDP and PEPs, SAML assertions and Assertion Servers.

The code was written in the Java programming language and includes a basic graphical user interface shown in figures 4 and 5.

The interface allows users to

- join and leave the group,
- to display a list of PDPs available in the group,
- send and receive encrypted messages,
- display the TGDH tree (Show Tree),
- show the current master group key, the number of members and the position of this member in the tree (Show info),
- and to start a new PDP for the group.

The distributed network layer has been implemented by using IP Multicast.

The TGDH layer is based on the TGDH library from Lijun Liao at Ruhr-University Bochum, Germany[6].

The Assertion Server and the XACML PDP are open source projects previously published by SICS[7]. We currently use non-synchronised flat files as attribute and policy databases and a simple PIP has been localised with each PDP, performing non-optimised policy retrieval.

The core components that where build from scratch according to our architecture are the Context handler and the PEP. The Context handler is configured by a simple file, specifying the group name the multicast address and port and the DSA key parameters for this member.

As a simple application layer we implemented a group chat sending and receiving strings.

[5]For example: http://xacml.dif.um.es/

[6]http://www.nds.ruhr-uni-bochum.de/research/top/gc/tgdh/ used with kind permission

[7]See: http://www.sics.se/spot for more information

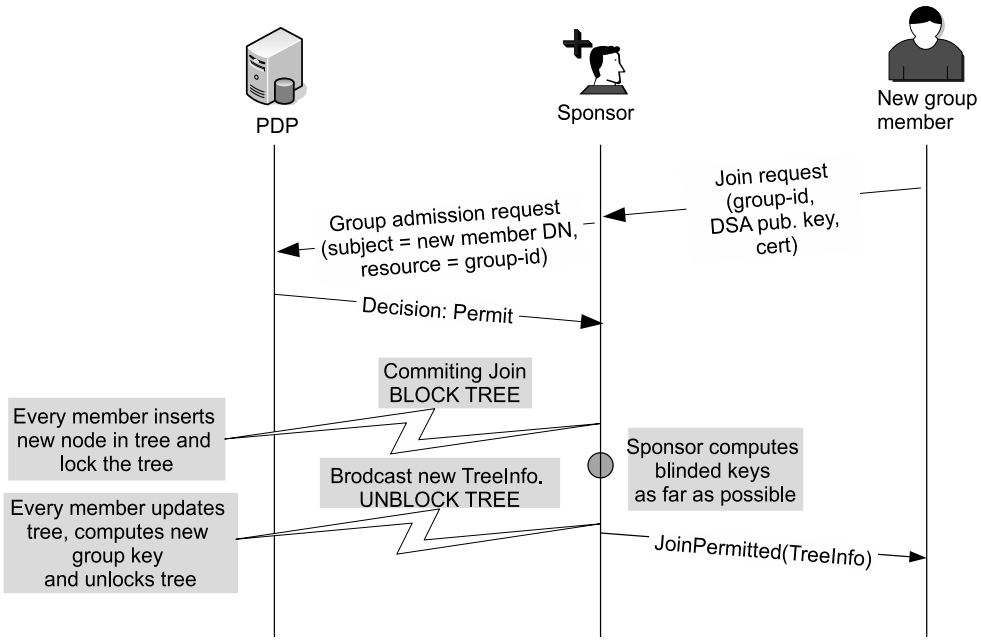


Figure 3: The sequence of diagram of the join protocol.

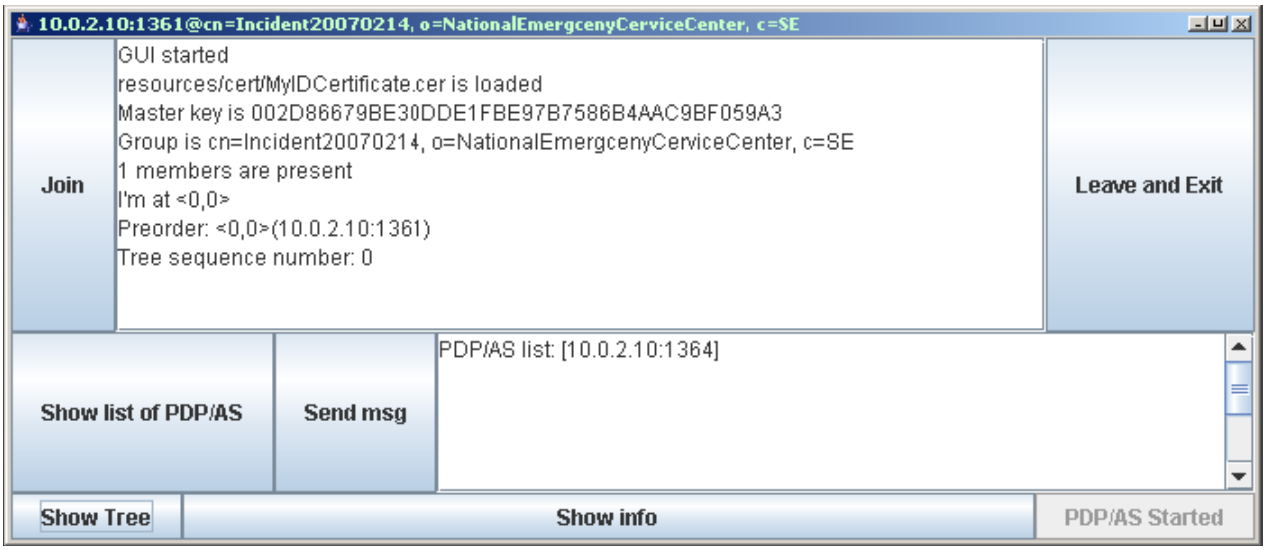


Figure 4: GUI at start.

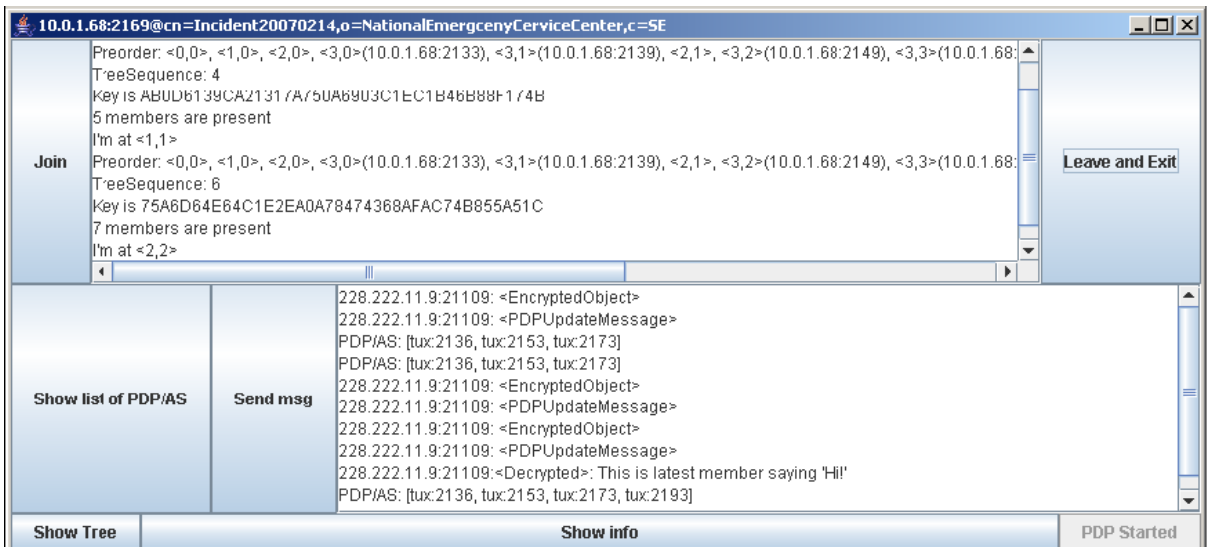


Figure 5: GUI after a few joins.

Figure 4 shows the user interface at start. The certificate has already been loaded and a master key for the group was generated. The DN of the group certificate is displayed together with some other TGDH tree information.

In figure 5 a few users have joined the group, thus expanding the TGDH tree and some encrypted messages have been sent (in this version the chat shows a lot of debug information too).

This demo exemplifies how we intend our architecture components to interact and shows some necessary steps for an implementation of the full architecture (customising a PEP, adapting a TGDH library etc). Since it was produced as part of Andreas Sjöholm's master thesis, the time frame did not allow for the extensive testing that would have been required to get meaningful performance data. Such testing will be performed on a more complete version of the architecture, including distributed attribute and policy databases and a more advanced application layer.

6. DISCUSSION

In this section we first discuss issues related to the contributory group key agreement component of our architecture. We then proceed to discuss issues related to the authorisation components.

A criticism voiced about contributory group key agreement schemes [5] claims that:

- An established communication layer is required which must support broadcast operations.
- If there are frequent changes in the group membership, due to member mobility and/or link outage, the topology of the ad-hoc group may change before the key agreement scheme has been completed.

We believe that our application area does not face these problems to a degree that would make contributory key agreement schemes problematic. In a crisis management scenario as presented in the introduction either mobile internet access is going to be available or dedicated communication infrastructures will be provided (e.g., the Rakel[8] digital radio communication system in Sweden).

As for the second problem, we assume that the main bottleneck will be communication cost and not computation (since the computational power of even small mobile platforms is quite high, especially if they include dedicated crypto-hardware). Considering the low communication cost of the TGDH join and leave events, it would require a constant stream of such events to seriously disrupt the function of our architecture. This is quite unlikely for our application scenario, since new members will more probably join as a batch and then want to stay group members for a period of time.

Concerning the use of XACML and SAML in our authorisation architecture, common criticisms are:

- These standards are too complicated because they provide too much functionality (thus confusing the users).
- The use of XML as base syntax introduces unnecessary verboseness.

It is true that the full range of XACML and SAML functionality is not needed for addressing our problem. However, building custom made proprietary systems is not a reasonable approach either, since we then would loose interoperability through standard compliance, and make future extensions more complicated.

Therefore we claim that a reasonable approach is to create profiles for the use of XACML and SAML in specific applications, that define subsets of the full functionality. Such profiles can address the access control problems in the specific vocabulary of the application, thus making them understandable for users. Since the profile only restricts the use of policy and assertion syntax, future updates of the functionality can simply be achieved by updating the profile

[8]see http://www.krisberedskapsmyndigheten.se /default_ _ _ _176.aspx for more information

(thereby allowing the use of new features of the standard). With custom made approaches, this would often require a redesign of the whole authorisation infrastructure.

There are two possible problems with the verboseness of XML encodings: Firstly it can be a question of bandwidth use, secondly understanding and reading the documents (policies or assertions in our case) can be very difficult for humans. The former problem can be addressed by various XML compression techniques, e.g., WBXML [12]. The latter problem can be addressed by appropriate user-interfaces that support policy editing and attribute management without requiring direct interaction with the raw XML format.

For our architecture as a whole, some points should be observed: Group members are trusted, no protection against insider attacks is inherently provided by this architecture. This is due to the peer-to-peer nature of our group, as opposed to a client-server group where there would be a group leader taking certain security responsibilities. At the TgdhXacml middleware layer, the damage an insider attack could do is the same as an active attacker can do. Such insider attacks must instead be prevented at application layer.

We are convinced that the benefits of having a flat group hierarchy (no single point of failure, distribution of workload between the group members) outweigh the drawbacks in our application scenario.

Our architecture relies on no single point of control or command since group key establishment (TGDH) is self healing and resources access information (policies) are distributed to reach an acceptable degree of redundancy. What will happen in case of network disruption is inability to propage policy updates (database synchronization, see Figure 2) and reovcation information, but group functionality will not be affected.

Another important property that one needs to be aware of when planning to use our architecture is that it requires authentication frameworks already to be in place with all potential group participants. Such frameworks often exist in organisations we aim at in our applications scenarios, however some identity federation efforts could be required to have them work together in a heterogeneous group. The presence of such authentication structures within organizations makes distributed authentication possible and makes TTPs such as a Kerberos server unneccessary (which would have been hard to implement across multiple organizations). This by letting only one authoritative representative per organization mutually authenticate with the crisis coordinator and thereafter take advantage of the Assertion Server and delegations.

7. CONCLUSION AND FUTURE WORK

In this paper we have presented an architecture supporting secure communication for ad-hoc, federated groups. Our main contribution lies in combining advanced access control mechanisms, federated identity management and contributory key agreement into an integrated architecture.

This architecture allows decentralised management of group admission control and therefore makes it possible to spread the administrative workload to the most competent parties, while not exposing the full set of rights to all delegates.

In the future we plan to investigate how to provide a better administration interface for policy and attribute management. We also consider to replace the chat application in the demo with a more realistic service. A suitable application could be Helping Hand[9], a graphical utility for cross actor support in emergency response.

APPENDIX

A. EXAMPLE POLICIES AND ASSERTIONS

In this section we present the full policies from figure 1 and the related SAML attribute assertions. In order to reduce the verboseness, we have defined the following abbreviations:
`nsp = "urn:oasis:names:tc:xacml"` and
`string = "http://www.w3.org/2001/XMLSchema#string"`
Please note that both policies are contained in a single PolicySet and follow the XACML 3.0 syntax, which is quite different from the XACML 2.0 syntax. The first policy (*RootPolicy*) delegates to the Police board the authority to create policies for accessing resources from the group *Incident23*, provided the subjects of these policies are members of the police force.

The second policy (*Alice'sPolicy*) is issued by Alice, and gives Bob access to the resources from the group *Incident23*.

```
<PolicySet xmlns="nsp:3.0:schema:os"
    PolicySetId="Incident23Policies"
    PolicyCombiningAlgId="nsp:1.0:policy-combining-
                        algorithm:permit-overrides">
  <Target/>
  <Policy PolicyId="RootPolicy"
      RuleCombiningAlgId="nsp:1.0:rule-combining-
                      algorithm:permit-overrides">
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="nsp:1.0:function:
                                  string-equal">
            <AttributeValue DataType="string"
              >Police</AttributeValue>
            <AttributeDesignator
              Category="nsp:3.0:attribute-
                     category:delegated:Subject"
              AttributeId="organisation"
              DataType="string"/>
          </Match>
        </AllOf>
      </AnyOf>
      <AnyOf>
        <AllOf>
          <Match MatchId="nsp:1.0:function:
                                  string-equal">
            <AttributeValue DataType="string"
              >Incident23</AttributeValue>
            <AttributeDesignator
              Category="nsp:3.0:attribute-category:
                              delegated:Resource"
              AttributeId="group"
              DataType="string"/>
          </Match>
        </AllOf>
      </AnyOf>
      <AnyOf>
        <AllOf>
```

[9]developed at the Viktoria Institute, see http://www.viktoria.se/∼landgren/crossactorsupport

```
        <Match
          MatchId="nsp:1.0:function:
                          string-equal">
          <AttributeValue DataType="string"
            >Police board</AttributeValue>
          <AttributeDesignator
            Category="nsp:3.0:attribute-category:
                            delegate"
            AttributeId="role"
            DataType="string"/>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule RuleId="Rule1" Effect="Permit">
    <Target/>
  </Rule>
</Policy>

<Policy PolicyId="Alice'sPolicy"
    RuleCombiningAlgId="nsp:1.0:rule-combining-
                  algorithm:permit-overrides">
  <PolicyIssuer>
    <Attribute AttributeId="nsp:1.0:subject:
                                    subject-id">
      <AttributeValue DataType="string"
        >Alice</AttributeValue>
    </Attribute>
  </PolicyIssuer>
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="nsp:1.0:function:
                              string-equal">
          <AttributeValue DataType="string"
            >Bob</AttributeValue>
          <AttributeDesignator Category="Subject"
            AttributeId="nsp:1.0:subject:
                              subject-id"
            DataType="string"/>
        </Match>
      </AllOf>
    </AnyOf>
    <AnyOf>
      <AllOf>
        <Match MatchId="nsp:1.0:function:
                              string-equal">
          <AttributeValue DataType="string"
            >Incident23</AttributeValue>
          <AttributeDesignator
            Category="Resource"
            AttributeId="group"
            DataType="string"/>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule RuleId="Rule1" Effect="Permit">
    <Target/>
  </Rule>
</Policy>
</PolicySet>
```

The SAML assertions would look like this (slightly abbreviated):

```
<saml:Assertion ID="58e1c517"
                IssueInstant="2007-11-16T13:27:15Z"
                Version="2.0">
  <saml:Issuer Format="string">asServer</saml:Issuer>
  <ds:Signature>
    ...
  </ds:Signature>
  <saml:Subject>
    <saml:NameID Format="string"
          NameQualifier="nsp:1.0:subject
                    :subject-id">Alice</saml:NameID>
  </saml:Subject>
  <saml:AttributeStatement>
    <saml:Attribute Name="role"
                    DataType="string">
      <saml:AttributeValue
            >Police board</saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

This one asserts that Alice has indeed the role *Police board*.

```
<saml:Assertion ID="7df9c595"
                IssueInstant="2007-11-16T13:33:28Z"
                Version="2.0">
  <saml:Issuer Format="string">asServer</saml:Issuer>
  <ds:Signature>
    ...
  </ds:Signature>
  <saml:Subject>
    <saml:NameID Format="string"
          NameQualifier="nsp:1.0:subject
                    :subject-id">Bob</saml:NameID>
  </saml:Subject>
  <saml:AttributeStatement>
    <saml:Attribute Name="organisation"
                    DataType="string">
      <saml:AttributeValue
            >Police</saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

This would assert that Bob is indeed in the organisation *Police*. The combination of these two assertions validate the second policy with respect to the first one, during evaluation by a PDP. Without these assertions, the second policy would be discarded as invalid.

B. REFERENCES

[1] A. Sjöholm. Secure Group Management in Dynamic Networks. Master Thesis at Department of Computer and System Sciences, Royal Institute of Technology, Stockholm, Sweden, 2008.

[2] D. Agarwal, O. Chevassut, M. Thompson, and G. Tsudik. An Integrated Solution for Secure Group Communication in Wide-Area Networks. In *Proceedings of the Sixth IEEE Symposium on Computers and Communications (ISCC'01)*, Hammamet, Tunisia, July 2001. IEEE Computer Society.

[3] A. Anderson. Xacml References and Products, Version 1.83, January 2007. http://docs.oasis-open.org/xacml/xacmlRefs.html.

[4] S. Cantor, J. Kemp, R. Philpott, and E. Maler Eds. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) v2.0. Standard, Organization for the Advancement of Structured Information Standards (OASIS), March 2005. http://www.oasis-open.org.

[5] A. Chan and E. Rogers. Distributed Symmetric Key Management for Mobile Ad hoc Networks. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 4, pages 2414–2424, Hong Kong, China, March 2004. IEEE Computer Society.

[6] S. Godik and T. Moses Eds. eXtensible Access Control Markup Language (XACML). Standard, Organization for the Advancement of Structured Information Standards (OASIS), February 2003. http://www.oasis-open.org/committees/xacml.

[7] P. Judge and M. Ammar. GOTHIC: A Group Access Control Architecture for Secure Multicast and Anycast. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications INFOCOM*, volume 3, pages 1547–1556, New York, USA, June 2002. IEEE Computer Society.

[8] Y. Kim, D. Mazzocchi, and G. Tsudik. Admission Control in Peer Groups. In *Proceedings of the Second IEEE International Symposium on Network Computing and Applications*, pages 131–139, Cambridge, MA, USA, April 2003. IEEE Computer Society.

[9] Y. Kim, A. Perrig, and G. Tsudik. Tree-based Group Key Agreement. *ACM Trans. Inf. Syst. Secur.*, 7(1):60–96, 2004.

[10] J. Kohl and C. Neuman. The Kerberos Network Authentication Service (V5). Technical report, The Internet Engineering Task Force IETF, 1993. http://www.ietf.org/rfc/rfc1510.txt.

[11] Y. Mao, Y. Sun, M. Wu, and R. Liu. Dynamic Join-Exit Amortization and Scheduling for Time-Efficient Group Key Agreement. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 4, pages 2617–2627, Hong Kong, China, March 2004. IEEE Computer Society.

[12] B. Martin and B. Jano Eds. Wap binary xml content format. W3c recommendation, World Wide Web Consortium, June 1999. http://www.w3.org/TR/wbxml/.

[13] E. Rissanen, H. Lockhart, and T. Moses Eds. XACML v3.0 administrative policy. Standard, Organization for the Advancement of Structured Information Standards (OASIS), June 2006. http://www.oasis-open.org/committees/xacml.

[14] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. AAA Authorization Framework. Request For Comments (RFC) 2904, Internet Engineering Task Force (IETF), August 2000. http://www.ietf.org/rfc/rfc2904.txt.

[15] W. Wang and B. Bhargava. Key Distribution and Update for Secure Inter-group Multicast Communication. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks (SASN)*, pages 43–52, Alexandria, VA, USA, 2005. ACM.

[16] W. Yu, Y. Sun, and R. Liu. Minimization of Rekeying Cost for Contributory Group Communications. In *Proceedings of Global Telecommunications Conference GLOBECOM*, volume 3, pages 1716–1720, St. Louis, MO, USA, November 2005. IEEE Computer Society.

SECURE COMMUNICATION FOR AD-HOC, FEDERATED GROUPS

Andreas Sjöholm
andreas@axiomatics.com
Axiomatics AB
Swedish Institute of Computer Science

Babak Sadighi
Axiomatics AB
SICS

Ludwig Seitz
SICS

AXIOMATICS REDUCING SECURITY OVERHEAD

SICS

SCENARIO

Severe accident

- Group collaboration goal
- Can involve many organizations (nodes)

AXIOMATICS REDUCING SECURITY OVERHEAD

SICS

THE EMERGING EVERNET

A federated collaborative group

- An in common goal
- No intra-knowledge between nodes
- Dynamic infrastructure and group composition
- Uses the Evernet as medium

AXIOMATICS REDUCING SECURITY OVERHEAD

SICS

PROBLEM

- How can confidentiality and integrity be ensured?
- Who may join a group?

The ad-hoc and federal structure of the group worsen the situation.

AXIOMATICS REDUCING SECURITY OVERHEAD

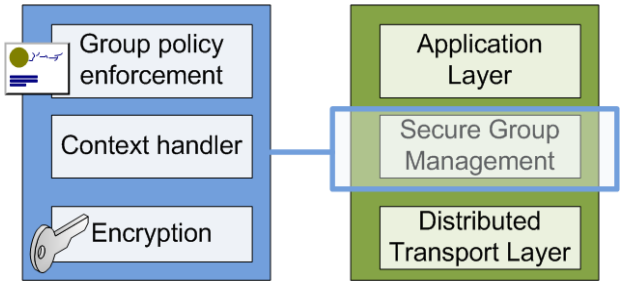
SICS

A MIDDLEWARE SOLUTION

Group policy enforcement

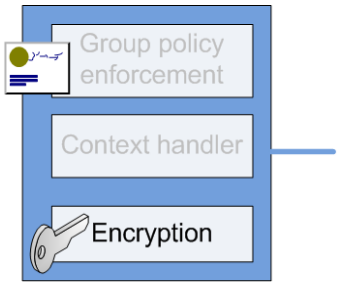
Context handler

Encryption/ Integrity



GROUP KEY DISTRIBUTION

- Contributory / Key dealer / TTP ?
- Symmetric / Asymmetric ?

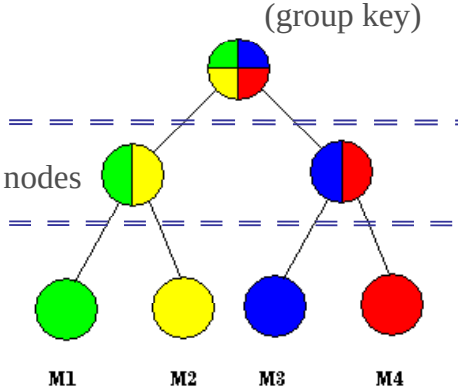


- Computational and decisional group key secrecy
- Implicit key authentication

TREE-BASED GROUP DIFFIE-HELLMAN

Proposed in 2004 at UC Irvine and UC Berkeley

Apply two part DH recursively to a group via binary tree

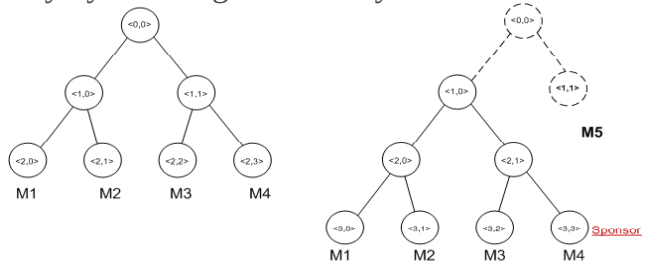


TGDH

DH :

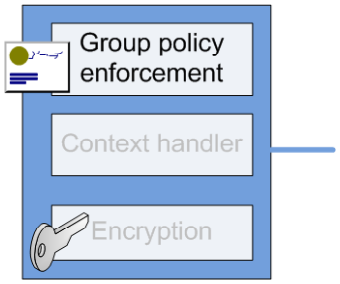
Two nodes (siblings) can compute an in common (parent) secret key by knowing its own key and the other's blinded key.

TGDH:



A member (leaf) can compute the master key, k<0,0>, by knowing all ancestors' keys and knowing the blinded keys of all ancestors direct siblings.

ACCESS CONTROL AND TRUST

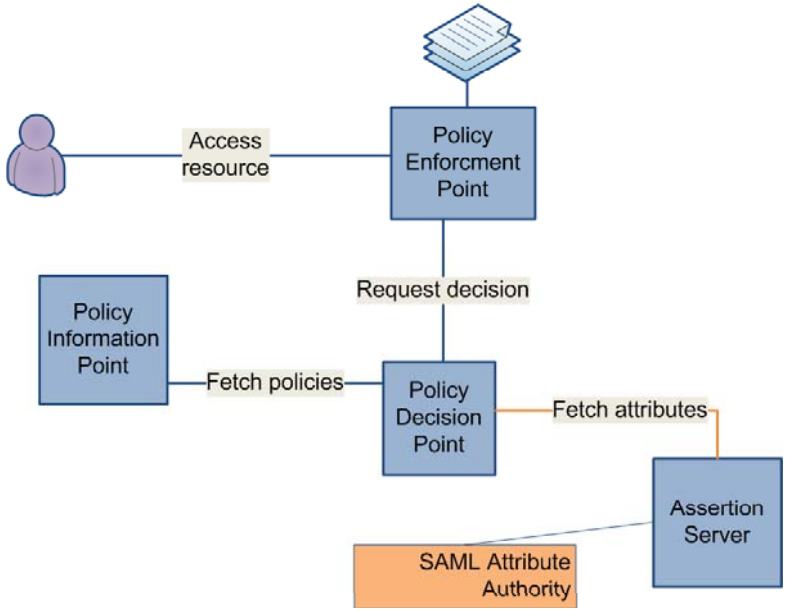


XACML – eXtensible Access Control Markup Language

SAML – Secure Assertion Markup Language

Assertion Server

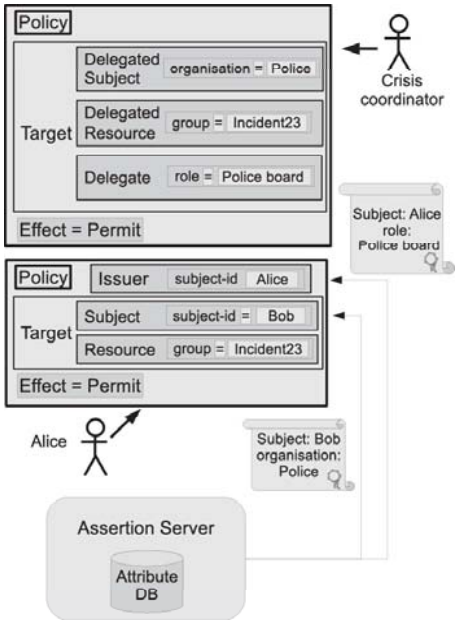
PKI



AXIOMATICS REDUCING SECURITY OVERHEAD
SICS

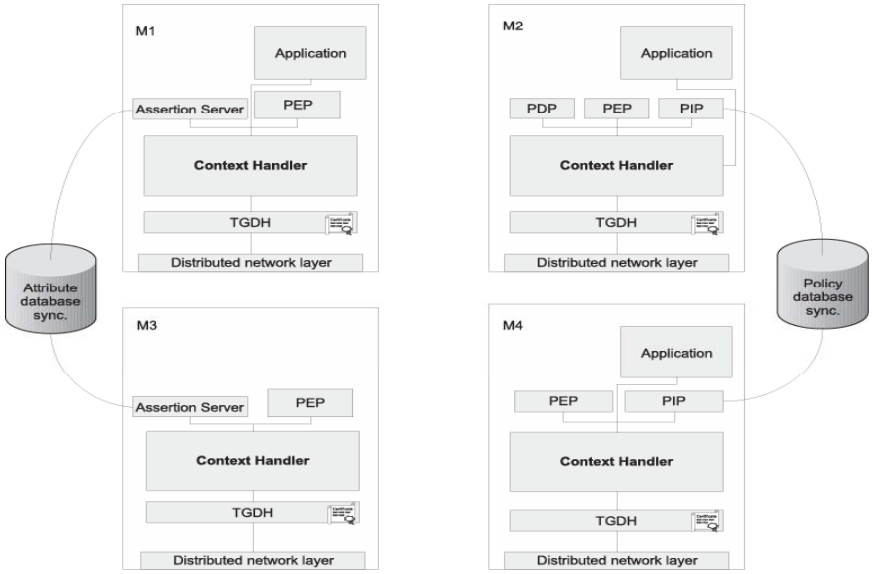
ATTRIBUTE ASSERTION & DELEGATION

1. CC delegates resource *group: incident123*
2. Alice has attribute *role: Police Board*
3. Alice creates administrative policy targeting Bob
4. Bob can access group



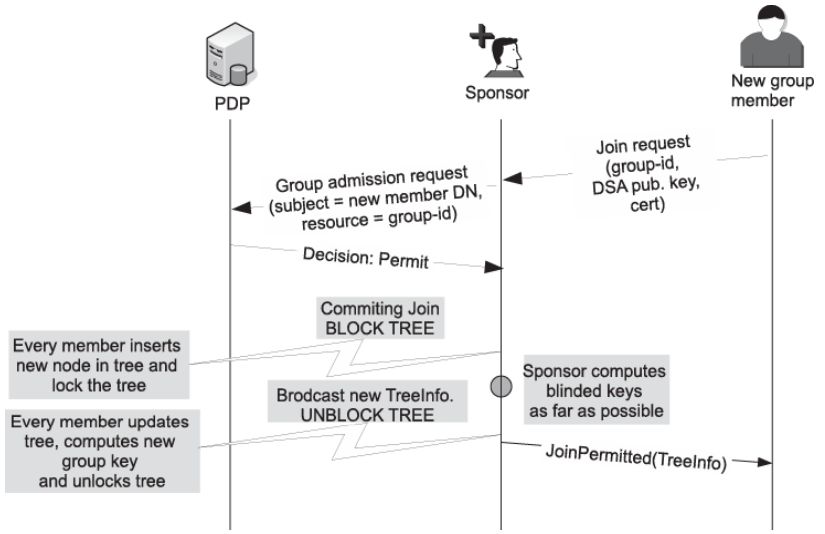
AXIOMATICS REDUCING SECURITY OVERHEAD
SICS

MIDDLEWARE - TGDHXACML



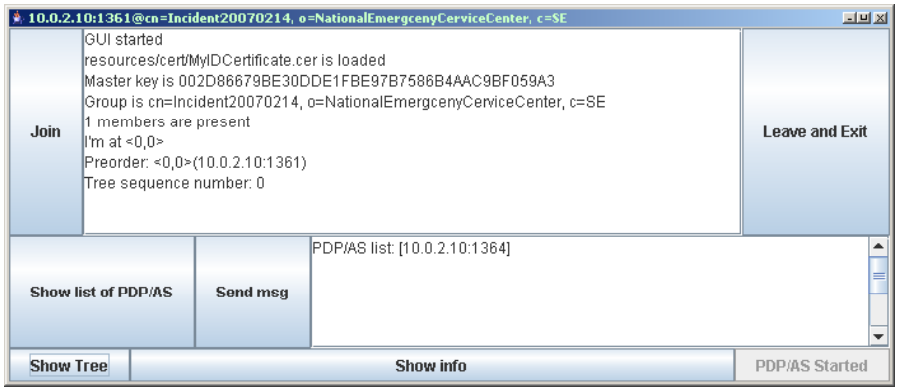
AXIOMATICS REDUCING SECURITY OVERHEAD
SICS

JOIN EVENT



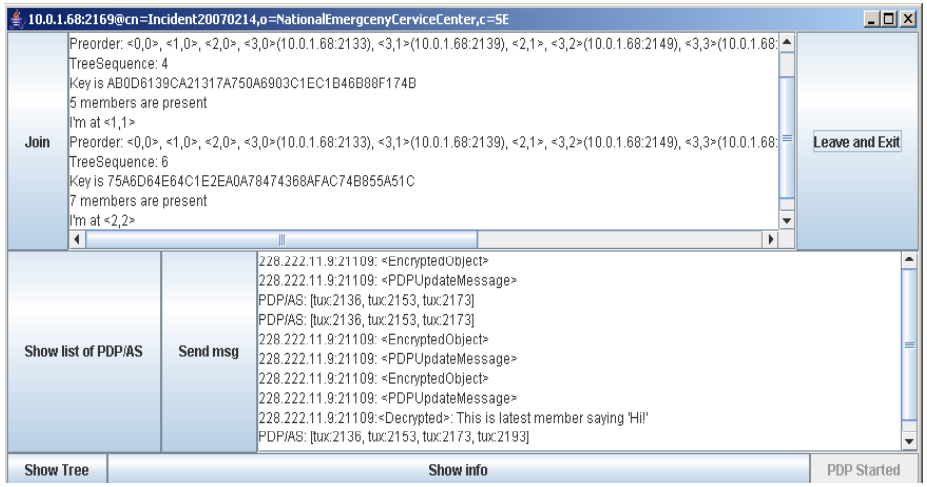
AXIOMATICS REDUCING SECURITY OVERHEAD

SICS



AXIOMATICS REDUCING SECURITY OVERHEAD

SICS



AXIOMATICS REDUCING SECURITY OVERHEAD

SICS

DISCUSSION

- XACML/SAML too excessive
- Block-free TGDH
- Sync of data (policies, assertions)

- Questions?

AXIOMATICS REDUCING SECURITY OVERHEAD

SICS

OASIS IDtrust

IDtrust Member Section

John Sabo, CA, Inc.
john.t.sabo@ca.com

PKI IDtrust Steering Committee

- Dr. Abbie Barbir, Nortel
- June Leung, FundSERV
- Arshad Noor, StrongAuth
- John Sabo, CA, Inc.

Many thanks to Ann Terwilliger, Visa who recently left the committee!

OASIS IDtrust

OASIS provides a neutral setting where government agencies, companies, research institutes, and individuals work together to advance the use of trusted infrastructures.

The old OASIS PKI Member Section has restructured as the OASIS Identity and Trusted Infrastructure (IDtrust) Member Section

The IDtrust MS has expanded its scope to encompass additional standards-based identity and trusted infrastructure technologies, policies, and practices.

OASIS IDtrust

Four Strategic Focus Areas:

- **Identity and Trusted Infrastructure Components**
  - studies and projects addressing technology-based Identity and Trust models and standards, relevant protocols and standards, trust infrastructures, and costs, benefits and risk management issues
- **Identity and Trust Policies and Enforcement**
  - policies and policy issues; policy mapping and standardization; assurance; technical validation mechanisms; and trust path building and validation

OASIS IDtrust

Four Strategic Focus Areas:

- **Education and Outreach**
  - documenting trust use cases and business case scenarios, best practices and adoption reports and papers; organizing conferences and workshops; and establishing Web-based resources
- **Barriers and Emerging Issues Associated with Identity and Trusted Infrastructures**
  - data privacy; interoperability; cross border/organizational trust; outsourcing; cryptographic issues; application integration; and international issues

IDtrust Member Section TCs

- **OASIS Digital Signature Services eXtended (DSS-X) Technical Committee**
  Advancing new profiles for the DSS OASIS Standard
- **OASIS Enterprise Key Management Infrastructure (EKMI) Technical Committee**
  Defining symmetric key management protocols
- **OASIS Public Key Infrastructure (PKI) Adoption Committee**
  Advancing the use of digital certificates as a foundation for managing access to network resources and conducting electronic transactions
- **OASIS Extensible Resource Identifier (XRI) Technical Committee**
  Defining a royalty-free URI-compatible scheme and resolution protocol for abstract structured identifiers used to identify and share resources across domains and applications
- **Open Reputation Management Systems (ORMS) Technical Committee**

  Providing the ability to use common data formats for representing reputation data, and standard definitions of reputation scores

IDtrust Member Section Study on the Use of PKI in OASIS Standards

- Survey initiated in 2006, addressing
  - Use and applicability of PKI in OASIS standards
  - Instances where the standards explicitly or implicitly define expectations regarding using PKI for authentication, encryption, or digital signature
  - Assumptions made within the standards regarding the methods and systems used to support the PKI
  - Whether explicit PKI-specific standards are referenced or expected (such as ISO, IETF etc.)
  - Perceived barriers to the deployment of PKI
- Soon to be posted on ***idtrust.xml.org***

- **Learn** through the IDtrust Knowledgebase of educational materials and background on the standards
- **Share** news, events, presentations, white papers, product listings, opinions, questions, and recommendations through postings, blogs, forums, and directories.
- **Collaborate** with others online through a wiki interface

http://idtrust.xml.org

IDtrust xml.org

Online community for identity and trusted infrastructure standards

Welcome to IDtrust XML.org.

This is the official community gathering place and information resource for identity and trusted infrastructure standards.

The site is hosted by the OASIS IDtrust Member Section, a group that encourages new participation from developers and users. This is an open, vendor-neutral community-driven site, and the public is encouraged to contribute content. See more about this site.

OASIS issues Call for Participation in new ORMS Committee

News: Submitted by carolgeyer on Mon, 03/03/2008 - 15:31.

All interested parties are invited to participate in the new OASIS Open Reputation Management Systems (ORMS) Technical Committee. The group plans to develop a system that provides the ability to use common data formats for representing reputation data and standard definitions of reputation scores. ORMS will not define algorithms for computing the scores; however, it will provide the means for understanding the relevancy of a score within a given transaction. The Committee's output will enable the deployment of distributed

Read more Login or register to post comments report spam

News

Identity theft: Six clicks from a cyber crook

News: Submitted by dschur on Sun, 03/02/2008 - 19:22. Last updated on Mon, 03/03/2008 - 15:23.

Posting innocuous personal details on social websites could expose millions to fraud, says Heather McLean. Our love affair with social networking, it appears, may be coming to an end. After almost 18 months of exponential growth, Facebook has suffered its first UK dip in user

Search

Browse
- Categories
- Site map
- Recent posts
- Recent changes

Contribute
- Edit or add Wiki pages
- Post news
- Add an event
- List your product
- List your service
- Recommend a resource
- Participate in a forum
- Create a blog
- See more options

Syndicate

Recent Content
- First Meeting of the OASIS ORMS Technical Committee *4 min ago*
- OASIS issues Call for Participation in new ORMS Committee *24 min ago*
- Identity theft: Six clicks from a cyber crook

Hosted by: OASIS

Sponsored by: IDtrust

IDtrust Wiki Knowledge base

Wiki page: Submitted by carolgeyer on Tue, 10/30/2007 - 18:24. Last updated on Wed, 01/23/2008 - 23:10.

Browse, edit, and add pages to this Wiki Knowledgebase on IDtrust. *Blue links point to existing wiki pages. Red links represent suggestions for pages where you can be the first to add content.*

Identity and Trust : Strategic Issues and Policy

- Fundamentals of Identity & Authentication
- Policy Frameworks for Trust & Identity
  - Authentication frameworks
- Regulatory Approaches to Trust & Identity
- Information Privacy
  - PKI and Privacy
- Interoperability
- Cross-border trust
  - Cross recognition arrangements
- Outsourcing
- Cryptographic challenges
- Return on Investment
- Application integration

PKI Technologies

- Introductions to PKI
- PKI Derivatives
  - Smartcards

IDTrust Member Section

- Tremendous growth and energy
- Many opportunities to get involved
- ***!*** European Security Workshop, London, Sept 28 – 30 2008 ***!***
- Consider joining OASIS and participate in the MS and/or TCs

- Contacts:
  - Dee Schur
    - Dee.schur@oasis-open.org
  - **http://www.oasis-open.org/join**
  - **Idtrust.xml.org**

User-centric PKI

Radia Perlman
Sun Microsystems
16 Network Circle
Menlo Park, CA 94025
1-425-651-4094
radia.perlman@sun.com

Charlie Kaufman
Microsoft
1 Microsoft Way
Redmond, WA 98052
1-425-707-3335
charliek@microsoft.com

ABSTRACT
The goal of supporting Single Sign-On to the Web has proven elusive. A number of solutions have been proposed – and some have even been deployed – but the capability remains unavailable to most users and the solutions deployed raise concerns for both convenience and security. In this paper, we enumerate desirable attributes in a scheme for authenticating from an Internet browser to a web site and the authorization that follows. We categorize the currently deployed or advocated approaches, describing their benefits and issues, and we suggest incremental improvements to such schemes. We then outline a design for public-key based authentication particularly suited to what we believe to be the common case: users, acting on their own behalf (as opposed to as an employee of an organization), performing actions on the web such as making a purchase or maintaining an account at a service provider. We contrast the usability/privacy/security properties of our design with other identity management/authentication schemes deployed or being proposed today. Our design is truly user-centric, in the sense that the user acts as his own CA, and as a decision point for authorizing release of user information to web sites, rather than having an Identity Provider be the center of trust.

Categories and Subject Descriptors
K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *Authentication, Unauthorized access (e.g., hacking, phreaking).*

General Terms
Design, Security, Human Factors.

Keywords
Web services, PKI, authentication, single sign-on

1. INTRODUCTION
This paper starts with a description of various problems that it would be desirable to address when using the Internet, particularly in the context of a user doing business with a variety of services on the web. Then we describe a variety of existing approaches, and review their strengths and weaknesses compared to the set of problems as well as to each other. We do not focus on particular syntax (such as SAML or X.509), or the details of particular deployed systems, but rather the properties of families of schemes that may or may not use the same standardized syntax. For example, we use the term "certificate" to refer to any digitally signed statement, which would include X.509 identity certificates[12], "attribute certificates" [7], or SAML assertions [3].

After describing the properties of currently deployed or advocated solutions, we describe an approach which is truly "user-centric", in the sense that not only does the user decide which attributes to divulge to which sites, but mutual authentication to web sites with which the user has an existing relationship is directly performed between the user (the user's machine) and a server, rather than, for instance, having the user authenticate to a third party (an "Identity Provider"), and have that identity provider vouch for the user. Also, the user's private attributes, although perhaps stored (encrypted) at some service on the web to facilitate user roaming, are not available to anyone except the user, the user's local machine, and on servers to which the user chooses to reveal them.

IDtrust '08, March 4–6, 2008, Gaithersburg, MD

2. DESIGN GOALS
Today, users typically have independently configured and maintained user names and passwords at lots of different web sites, and must enter the appropriate ones when visiting particular sites. This is both inconvenient and insecure. Ideally, users would like to be able to authenticate to a machine once when they log in (possibly using the evolving "best practice" technologies like smart cards and fingerprint readers) and then have the machine authenticate to all visited web sites securely, efficiently, and transparently. We first list some desirable attributes (as a means for comparing various approaches). There are inevitable trade-offs; no solution meets all of the goals.

2.1 Single Sign-on
As with the slogan "the network is the computer", the user experience should be of a single authentication step, for instance, activating a smart card and inserting it into a machine, or typing a username and password. After that, the user should be able to access any service on the net with which the user has an ongoing relationship, and be logged into the user's account without the user consciously doing another authentication.

2.2 Protection from Phishing
Safety: If a user is tricked into authenticating to an evil site (e.g., phishing), the site should gain no credentials with which to impersonate the user, either at the site the user intended to visit, or at other sites to which the user has access.

2.3 Minimal Dependence on Third Parties

Efficiency: A user should be able to communicate directly with a web site without having to first communicate with other machines.

Robustness: A user should be able to communicate with a web site even if other sites are not reachable at the time.

Off-line Trusted Entities: It is desirable to have the system operate in a way that the most trusted components are not connected to the network. The reason for this is that a system that requires a trusted server to be on-line in order for authentication between other systems to work will be less secure because:

- An on-line server is an attractive target for network-based attacks. It is likely to have security flaws, and thus likely to be compromised. Even if not compromised, it is subject to DDOS attacks.
- Such a server must be replicated for availability and performance, and each replica must be physically secured.

2.4 Flexible use of accounts

Multiple accounts with the same service provider: A user may have multiple accounts with the same service provider. For instance,

- one account for purchasing work-related items, and another for personal items
- entering chat rooms with different pseudonyms
- multiple email accounts with a single email provider

It is desirable for authentication to be automatic, while still allowing the user to choose which account to access.

Multiple users with the same account: A family might share an account at a site such as a DVD-by-mail service. A user would like to be able to access all his private accounts through a single sign-on, but also be automatically logged into accounts on sites in which he is one of several authorized users.

Easy and Secure Account Creation: It should be easy to create a new account and have it automatically use, for future authentication, the credentials with which the user is currently logged in. If the user has several potential credentials (e.g., password, set of client certificates), the user should be able to select which should be used for this new account.

2.5 Convenient Roaming

Convenient Roaming: A user would like to be able to access accounts from a variety of different platforms (home desktop machine, work machine, laptop, cellphone, hotel lobby shared desktop machine).

Least Privilege: Machines (even a user's own machine) are not necessarily secure. Logging in with "full credentials" to perform a single operation such as printing a boarding pass, or checking to see what conference room the next meeting is in, might be taking an unwarranted risk.

Credential Agility: Although a smart card might be the most secure choice for user authentication, the desktop machine in the hotel lobby from which the user would like to print his boarding pass might not have a smart card reader. Or the user might have left his smart card at home, and still want to do work. It is desirable for the user to be able to access the web through whatever authentication methods are practical for the situation.

Authorization based on authentication method: If there is the ability to log into different machines using different credentials as described above, some logins will be more secure than others. Therefore, it might be desirable for authorization decisions to depend on where the user is authenticating from, as well as how he authenticated. For instance, the user might decide that making trades in his brokerage account only be possible when he has authenticated with his smart card, or be working on his home desktop machine, whereas checking his available balance might be authorized from any location and using any authentication method.

Or a server holding a company's confidential files might allow access to those files only to users that have authenticated from an onsite machine using a smart card.

2.6 Additional user attributes

Convenient access to user information: There is some information that the user always knows, for instance, home address. It might be convenient for this information to be automatically filled in when required, but it is less necessary than other information that the user might need and not easily remember, such as passport number and credit card numbers.

Authenticated Attributes: There are some attributes that a user cannot simply claim by filling information into a field, such as whether they are over 18 (e.g., for purchasing alcohol over the web), or whether they are not a citizen of one of the countries currently disapproved of by some government (e.g., for export control decisions), or whether they are employed by a company that has corporate membership in some organization (e.g., a service that allows all employees of member corporations to have access to on-line search).

2.7 Credential loss or theft

One-step revocation: The user might want to change the credential with which he performs his single sign-on. This might be because, as is good security practice, he changes his password periodically, or because his smart card has been stolen. When this happens, accounts at all servers must (within a reasonable amount of time) know not to accept the old credential.

Protection against loss of credentials: In many schemes (including the one we advocate) a lot of important information is stored, encrypted with the user's private key or password. If the user forgets this password or loses the smart card with the private key, it should be possible to salvage all the information.

One-step rollover to a new credential: Once the old credential is revoked, it should be convenient to start using a new credential, without the requiring the user, for instance, to individually reinstalling his account at each service provider.

2.8 Minimal platform changes

Deployability of an authentication scheme depends critically on how many different components must be updated before it can be used. Virtually all of the schemes being proposed or deployed fall into one of two categories: schemes that can be deployed on client devices with supporting infrastructure without modifying the existing web sites; and schemes that can be deployed on web

servers with supporting infrastructure without modifying the deployed base of clients. Schemes that require changes to both clients and servers face a chicken and egg problem, since there is substantial cost and no benefit to deploying the first half of the solution.

Note that some client changes are easier than others. All of the major browsers are highly extensible (though incompatibly), allowing add-ons to be installed that do additional processing. There are a wide range of such add-ons aimed at specialized processing of web site logons. Users should be leery of installing add-ons from unfamiliar web sites, though most are not. Two problems with such add-ons are that they may not be available to users who are logging on to kiosk or otherwise unfamiliar machines, and they may not be available for the browsers on specialized devices like cell phones.

2.9 Access Privacy

A user should be able to access sites without any server being able to know which other services a user is using. Where possible, it is desirable for a user to have a degree of anonymity where the server knows that the user is authorized to access something without knowing the identity of the user. If the user is accessing two different sites, the user may wish that the two sites not be able to determine that the requests to them originate with the same carbon based life form.

3. EXISTING SCHEMES

In this section we review existing schemes. By "existing", we mean either deployed or advocated. We are intentionally grouping families of schemes that have similar properties for the purposes of our discussion, and hope their developers won't be offended by our glossing over design aspects they might consider to be crucial. This is also a rapidly growing field, and we are not attempting to make our list complete.

3.1 Classic Username/Password

While within an enterprise or a university environment, more sophisticated authentication schemes based on centrally registered "accounts" are used for file, database, and sometimes http access, the dominant authentication means on the World Wide Web remains username and password. On a good day, those credentials are protected from eavesdropping by SSL. They are used for everything from accessing banks and brokerage accounts to email and social networking sites, and by and large users have to choose separate usernames and passwords for each site they use. Payment schemes are predominantly based on users typing in credit card numbers and expiration dates to merchant sites. The inconvenience of these mechanisms is well known and the security weaknesses are well publicized. Users cannot remember high quality secrets (even in small quantities, much less in large ones), smart cards are not widely deployed (especially outside of an enterprise), and there is no coordination among Internet services that would allow any centrally managed form of authentication to be accepted everywhere.

Passwords present many security issues: some obvious and some more subtle. Users often choose passwords that are easy to guess, users write them down, the passwords are exposed to keyboard loggers and shoulder surfers, and users tend to use the same password on multiple systems, allowing servers to impersonate the user to other servers.

Furthermore, prompts for usernames and passwords take place on ordinary web pages that can be easily forged, tricking users into giving away that information to impostor sites. While the protocols for authenticating web servers to browsers are cryptographically strong, the visual cues by which browsers identify web sites to users have been shown to be ineffective[27]. Users should not be expected to understand the arcana of URLs, and know, e.g., that http://84.212.151.26/www.creditunion.com is not actually the web site "creditunion.com".

Although it might be desirable for a user to use a different password at every site, there is no way for a user to remember all the different usernames and passwords at all the sites at which the user has an account. Neither can the user pick a strong password and use it everywhere because sites have incompatible restrictions on what kinds of characters must and must not be contained in passwords.

The user can't even use the same username at all places for reasons such as:

- The username the user has been using is already in use at some new site the user would like to register with.
- The site might have its own syntax rules about usernames that the user's customary username does not fit (such as minimum or maximum length, or legal characters in the name).
- The user might not want anyone to be able to correlate his accounts at different sites.
- Some sites use the user's email address as their username. Email addresses tend to change, since they are often provider-based or employer-based.
- The user might have several accounts at a site, and wish to select which one to use for the session.

To prevent passwords used at multiple sites from being exposed if one of the sites is compromised, sites should store a hash of the user's password for verification rather than the actual password. Unfortunately, many sites store the cleartext password. This is obviously done at sites which email the actual password to the user when the user has forgotten the password.

When forgotten password recovery is to be based on the (dubious) security of email, there are two possibilities:

- A site keeps the user's actual password, and emails the password to the email address the user has registered with, when the user claims to have forgotten her password.
- A site keeps a hash of the user's password, and resets the password to some single-use password. When the user logs in (or clicks on the link in the email), the user is asked to set the password to something else.

The first system has the advantage that someone cannot maliciously claim to have forgotten someone else's password, and cause that person to go through the inconvenience of resetting

their password. This would likely be only a minor inconvenience unless the user does not have access to email at the time.

The second system has the advantage that someone who steals the password database at one merchant cannot easily impersonate users at other systems. With the first system (cleartext passwords stored), even if a user were to customize passwords at different sites, a thief who notices the user's password is "amazonPwD89351" might guess that the user's password at ebay might be "ebayPwD89351".

Evaluating this "classic" username/password authentication against our design goals, it fails at Single Sign-on (the user must separately authenticate to each site) but provides some protection against phishing: it is easy for the attacker to trick the user into revealing the password to a target site, but at least the password is (hopefully) only useful in impersonating the user at the target site. It would be much worse if phishers could obtain a password that would allow access to all of the user's resources. It actually fares fairly well against the other criteria. Roaming is easy – passwords can be typed into any machine. Credential loss recovery is different for every site and there is no coordinated recovery if the user loses many at once (e.g., they were all stored on a sticky note or in a file on a stolen laptop). It has no dependence on third parties, allows maximum flexibility to users with multiple accounts, and by definition requires no platform changes.

Another merchant practice that causes inconvenience for users, as well as making them more vulnerable to security and privacy problems, is for merchants to maintain customer information beyond anything that adds value to the user. An occasional purchase of merchandise from a web site does not require a long term relationship with the user -- it is largely for the merchant's benefit that the merchant maintains personal information about the user after the transaction has completed.

Worse yet, in the case of the occasional purchase from an on-line merchant, it is often more onerous to be a returning customer than a new customer. If it has been several years since the user last interacted with the merchant, it is very likely the user has forgotten his username and password on the site. A new user need only type in information that he knows (such as his home address) and credit card (which he copies from a credit card he selects.

A common experience for a user that has purchased something from the site several years ago is that once she types her email address (needed to obtain a receipt), she is informed by the site that she is indeed a returning user, and must now log in, furnishing her username and password. Given that she has almost certainly forgotten both, she now has to go through a "username recovery procedure" that may or may not succeed (because she doesn't remember which telephone number she gave them when she bought from them 4 years ago), or what she made up for answers to the few amazingly inappropriate security questions the web site makes her choose from. Personally, I have no pet, no favorite sports team, I don't remember my 2nd grade teacher's name, and my father does not have a middle name.

How many things are wrong with this currently widely deployed strategy?

- *It should be no less convenient for someone who has previously purchased something at the merchant to make a purchase, than someone who is making a first time purchase.*
- *If an account is to be maintained, and the user must find and authenticate with a username/password at the site, security questions for retrieving this forgotten information must be chosen by the user.*

Furthermore, in the case of an occasional purchase at a site, not only is the experience typically more onerous for a returning user than a first time user, but the ongoing relationship (the merchant keeping information about that user) is of extreme *negative* value to the customer. Not only is it less convenient for the user to make a future purchase (because of having to first go through the procedure for recovering the old username and password, which even if succeeds is likely to take several minutes), but the user's personal information is stored in a database that is likely to be broken into at some point. Most information the web site needs, such as the user's address, is reliably maintained in the user's head, thank you, and not a burden to type in. Other information, such as the credit card that the user used last time the user purchased something, is not only dangerous to store, but is likely to be out of date when a customer only purchases something from that merchant every few years,. Also, users tend to have many credit cards, and have reasons (perhaps based on the current balances on each of their cards, or current promotional rebates) for selecting a particular card for a particular purchase. Storing credit card information is thus not very useful to a customer and very dangerous.

There should be a law:

> *After a merchant has been paid, a customer must be allowed to request that any subset of the information about that user stored at the merchant (including all of it) must be deleted by the merchant. And of course, the merchant should be required to comply with this request.*

But there are some web sites that do need to maintain an ongoing relationship with a user -- DVD subscription services in which a user maintains a movie queue, brokerage services where a user trades online, airline accounts where a user purchases tickets and accesses their frequent flier points, tax packages where users prepare their taxes and where their records are maintained year after year, to name a few.

3.2 Enhanced Username/Pwd Schemes

Browsers usually have a feature where they will remember, for various sites, what the user's username and password are, and fill them in for the user. The problem with this is that then it becomes even more likely the user will not remember his username and password when he winds up needing to access the site from a different machine, or discovers all that state irretrievably gone when his computer dies and he needs to replace it. *The less frequently a user needs to type his username and password to a site, the more likely it will be that the user will forget the username and/or password when the user actually needs to remember and type it.*

The Local Password store approach requires no changes to servers. Browsers typically have this feature "built-in" for storing this information when *HTTP authentication* [8] is used, but there

are a growing collection of add-ons that support it for the more common case where passwords are entered on a web page.

Examples include PwdHash[19], TrustBar[10], PasswordVault[20], Pvault[5], Sxipper[23], and Passpet[29]. The idea is that plug-in software installed on client machines notices password prompts from servers and automatically satisfies them (usually invisibly to the user except for a small delay). PwdHash is a little different from the others. It does not keep a database of passwords, and does not change the user experience at all (users have to type in as many usernames and passwords as they ever did). PwdHash improves security in the common case where a user uses the same password at lots of sites. It hashes the password typed by the user with the site name to get a unique password for each site. This means that breaking into one site will not enable impersonation of the user at other sites and breaks phishing for passwords.

These systems are relatively straightforward conceptually, but in practice involve a lot of fragile trickery dealing with the plug-in architectures of the various browsers.

There is no standard password prompt that web pages display, though they do have a lot in common. The password field usually is configured to echo some special character rather than what the user types in the field, and it is often called “password”. Figuring out exactly how to recognize a password prompt page and respond correctly for the great variety of existing web sites is a continuing challenge for people who maintain this sort of software.

This approach often has as a design goal that the user never sees the actual password, and it is long and randomly chosen to make it difficult to guess. That means that the software must also recognize the account creation page and know what various site’s rules are for acceptable passwords so it can fill in appropriate ones. For sites that require periodic password changes, the software can automatically change the password while the user is doing something else.

A challenge with this approach is keeping the various password databases synchronized if the user accesses the web from multiple machines. Another is that the software (and database) must be installed on any machine from which the user wants to access the web, which may not be allowed in the case of kiosk or borrowed machines.

Most of the add-on schemes either have some sort of roaming built in or have some sort of export to a file capability to allow such roaming to take place manually. They also generally allow the roaming of other personal data like addresses, phone numbers, and favorite web sites.

Measuring these add-on schemes against our design goals, they do a good job at providing single sign-on. They typically depend on third parties only to support roaming. Their support of roaming, carrying additional user information, access privacy, and failure recovery is potentially very good, though the details vary by scheme. Dealing with shared accounts and users with multiple accounts is a user interface challenge, since it conflicts with seamless single sign-on, but these schemes have the potential to do as well as is possible. The most interesting issue regards how these schemes deal with phishing. They generally cite protection from phishing as a major feature, since users don’t know the site passwords and therefore can’t be tricked into revealing them. While users don’t check the identity of a web page before entering a password, the automation software will. The flip side is that the credentials for the one single sign-on are much more valuable than those of any particular web site, so if some phishing attack (or getting the user to run hostile software on his machine) can acquire that secret the damage is greater.

3.3 Identity Provider Federated Solutions

A variety of web-based schemes have been proposed (Liberty[25], web services[26], Shibboleth[15][21]) in which a user authenticates to an "identity provider", which provides the user with a credential that identifies the user to a service provider. In such systems, the service providers must be affiliated with the identity provider in a trust relationship (in a previously arranged "federation" arranged between the service provider and the identity provider), and the user must "link" the identity at the service provider with the user's identity at the identity provider.

As originally conceived, browsers were designed so that web sites could not interact with one another except in specific limited ways. A web site can store state associated with a user session – or even a long term memory associated with the site – in the form of a cookie on the user’s machine. The short term memory enables users to log into a web site once rather than for each page viewed, but it is forgotten when the user logs out (of either the site or the browser), and not shared with other sites. The long term memory survives reboots and allows web sites to know what user last accessed the web site from a particular machine, but it’s generally considered identification rather than authentication because machines are frequently shared. And like the short term memory, this is designed not to be shared with other web sites.

The cookie scheme works well for sessions within the pages of a single web site, but an SSO scheme can’t create a cookie visible to web sites whose DNS names (contained in the URL) are not sufficiently similar to those of the sign-on site. The common workaround for this involves HTTP/URL redirection. If I enter the URL of a site into my browser, my browser will go to the site whose DNS name is embedded in the URL, and try to retrieve the data associated with the URL. A site can – if it chooses – provide a different URL for my browser to fetch instead. The URL to which I am “redirected” need not have any similarity to the original URL, so by generating a URL that contains data, this provides a mechanism for independent sites to interact. The second site can – if it chooses – give my browser a third URL to retrieve, with the chain extending indefinitely.

Most web SSO schemes work with some variant on the following:

1) When a web site that is part of a federation receives an http request, it looks for authentication information included as a session cookie or part of the URL. If there is none (and there won’t be the first time through), it redirects the browser to the IDP (identity provider) site with a URL that begins with the name of the IDP site but includes (as data) the URL in the request.
2) The IDP site looks for a cookie indicating that the user is already “logged in”. If so, it redirects the user’s browser again, pointing it back to the original URL, this time adding the user’s authentication information.
3) Now back at the original web site, there is still no cookie, but there is additional information in the URL authenticating the user, so the web site creates a session

cookie and returns it along with the page associated with the original request. In normal use, the two redirects take place so quickly that the user doesn't notice that anything extra is going on. The authentication takes place seamlessly.

If when the user reached step 2, he had not yet logged into the identity provider (i.e. no cookie), then instead of redirecting the user back to the original web site the identity provider authenticates the user using any of a number of techniques that could include username and password, but could also involve smart cards or X.509 certificates. After the user authenticates, the IDP sends the user's browser a web sign-on cookie and then redirects him back to the original web site.

This design requires no changes to the browser. Functionality put in the browser for other purposes is exploited to do the authentication. Web sites that want to join a federation and the identity providers must carefully coordinate what they are doing, but there could be many identity providers operating independently with collections of web sites behind each of them. They would not be aware of one another.

Several problems with this sort of approach are:

- There are limits on the maximum size of a URL and of cookies, and if applications are already running close to those limits adding authentication information can push them over. Remember though, that you can (with some cost in performance) always substitute a URL, an encryption key, and a cryptographic hash for an arbitrary amount of information and then store the real information at the URL encrypted with the key.
- A web site might want to affiliate itself with multiple identity providers and allow the users logged into any one of them to authenticate seamlessly. The problem is that the site doesn't know to which identity provider it should redirect a user. If the identity providers cooperate, there are ways of smoothing over this, but without browser changes or configuration restrictions there are no perfect solutions.
- Similarly, a user might have many identity providers, and might want to have accounts at many service providers, where not all the service providers are affiliated with the same identity providers. Currently advocated schemes are largely silent on the issue of dealing with multiple identity providers.
- Although in theory user authentication to the identity provider can be done with any sort of technology, it is usually deployed with a username and password. If it were done with a stronger mechanism, say smart cards, it would lose the benefit of being deployable on any browser from any machine. Given the fragility of server authentication to the user, if a user can be tricked into typing the user's identity provider username/password to a site impersonating the identity provider to the user, that evil site can then impersonate the user, not just at the identity provider, but at any affiliated site.
- If the identity provider is currently not reachable, there is no way for the user to attach to any of the service providers.

Identity Provider solutions do help the issue of service provider impersonation (phishing), because the Identity Provider site is not likely to have a relationship with the evil site, nor would the user have given permission for any of the user's information to get shared with the evil site. If the user has authenticated to the real identity provider, the user will not be tricked by unaffiliated service providers.

However, especially with user authentication based on username and password, the Identity Provider approach is very vulnerable to a user being spoofed by a site impersonating the Identity Provider, especially because the user will often not type in the URL for the Identity Provider, but instead be redirected to the page in which the user is prompted for his username and password. And the username/password at the Identity Provider is particularly disastrous to divulge to an impostor site, because compromise of that information compromises the user at all sites the user has linked with that identity provider account.

Looking at how well this sort of solution meets the other design goals, identity providers are unlikely to actually provide single sign-on, but rather a reduced number of sign-ons. That's because it is unlikely that a single identity provider will sign up all of the sites a user wants to access. Banks, brokers, and other particularly sensitive sites will likely affiliate with no identity provider or only with very specialized ones. These schemes are totally dependent on third parties, introducing both security and availability issues. They are likely not to work very well with shared accounts or users with multiple accounts because the http redirection technology is hard to make both flexible and transparent. They handle roaming well. They handle carrying specific kinds of user configuration information along like credit card numbers and other authentication-related information, but they are unlikely to deal with ad-hoc information like lists of favorite sites. They are well qualified to carry information that requires attestation, like age, memberships, and corporate affiliations, and can enhance privacy by proving affiliations without identifying the individual requestor to the web site. In principle, they should be able to do a good job of credential changes because credentials need only be revoked in one place (at the IDP). While client-only enhancements are largely transparent to web sites, Identity Providers are often transparent to the client software. Web sites can deploy them without requiring that users install any new software on their machines (which is particularly important when dealing with kiosks, specialized devices (like cell phones), and a user base running a variety of types of browsers.

3.4 CardSpace

CardSpace [4] is a system that allows a user to select one of several "cards" (signed assertions about various attributes of a user) to present to a service provider. These cards might be signed by an identity provider, in which case CardSpace becomes similar functionally to the systems we described in section 3.3. It improves on those schemes by having local storage for remembering things like what credentials are associated with which sites and by having the user be prompted with a difficult to spoof UI asking what information the user would like to pass to the web site. The cost of those security improvements is that it requires additional software be installed on all clients as well as all servers and it asks users lots of questions (breaking the transparency of other single sign-on solutions).

CardSpace provides another mode of operation with self-issued cards, which is like having a private identity provider on the client. This eliminates the need for any initial user registration with an identity provider, and the need to have another machine in the authentication loop. With self-issued cards, CardSpace is similar to one aspect of the scheme we propose in Section 4 (registering the user's public key directly with the service provider as a method of authenticating the user).

3.5 Controlling Access to User Information

Section 3.3 only discusses how the Identity Provider solution addresses single sign-on. An enhancement to the Identity Provider Federation solutions is to also store user information (such as address, credit card, phone number, etc.) at some location. It could be at the Identity Provider, or at another machine, or some of the user's information might be at one server, and some at another. For simplicity, we will assume all the user information is stored at the Identity Provider.

The vision is that the user can set policy on each item of personal information, as to which information should be made available to which web sites. Information that has been approved to be sent to a site is sent automatically to that site when needed.

This might be a mechanism for keeping user personal information more secure, because then the service provider would not need to keep the user's information in a database, and there would be fewer databases on the web with personal information for the user. Since each database has some probability of being broken into, the fewer places that user information is stored, the more likely it is that it will remain secure.

However, it is likely that service providers will keep the information. They might use the identity provider to retrieve the information when the user is first setting up an account, but it is in the service provider's interest (less bandwidth, latency, and work), to store the user information on the service provider site so that in subsequent interactions it would not need to be retrieved from the identity provider.

3.6 Authenticated Attributes

Some attributes are intuitively ones that need more proof than just having the user fill in fields on a form. Examples are:

- I am a member of an organization that is authorized to use the services of this site, for free.
- I am over 18.
- I am not a citizen of one of the countries that the company I am communicating with is barred from allowing downloads to.
- My credit card number is X.

Other attributes seem like they would not need proof, such as:

- Address
- Telephone number
- Email address

First, even attributes that seem like they would not need proof might, in certain situations. It would seem as though the "rent out my house for a week" service, or the "demolish my house" service would need some sort of proof of address. And the "register my telephone number to be called every time this stock changes value" service ought to verify a phone number.

Email address is routinely verified by web sites, and they do this by sending a token to the web site, which you have to return by clicking on the link in the received email. This low tech solution is secure enough in practice.

For authenticating the attributes listed above as needing to be authenticated, there are only ad hoc and not very secure mechanisms deployed. For downloading export-controlled code, typically a check of IP address is made, with perhaps a form that you have to click agreeing that you are not a citizen of one of the frowned-upon countries. This does not prevent someone from using a proxy site to disguise his IP address, or downloading it at a public machine while visiting an approved country.

The identity provider solutions have a solution, in that identity providers can sign the attributes. However, this implies that there are sites out there that are trusted to assert all the needed attributes for a user, that the user has some mechanism for authenticating to those sites and requesting the signed attributes, and – perhaps hardest of all – there is some infrastructure in place that enables users to find them.

3.7 Strong Password Techniques

A strong password protocol solves the problem of doing mutual authentication using a weak secret (such as a password) in such a way that neither side in the exchange (client or server), or an eavesdropper, can do a dictionary attack. This form of mutual authentication might be a more reliable way of ensuring that the user is not fooled by an imposter web site than TLS, or TLS alone, because of the problems we mention in section 3.9. Because this form of authentication is mutual, TLS authentication is redundant except when establishing the shared secret initially.

The straightforward mechanism of sending a password does not do mutual authentication. There are other protocols such as CHAP[23] in which the password is used as a secret in a challenge/response protocol. Variants of such protocols can be used for either one-way authentication or mutual authentication. However, these sorts of exchanges are subject to dictionary attack, by at least one side in the exchange, and by an eavesdropper.

There are a number of cryptographic techniques for authenticating with a weak secret (a password) in a cryptographically strong way. The first of these was EKE[1], though there are others SPEKE[13], SRP[28], PDM[18].

Roughly, these protocols weave a password into a Diffie-Hellman exchange in such a way that someone impersonating either side can only verify one password guess during an authentication, and an eavesdropper cannot even verify a single password guess. A strong password protocol provides mutual authentication. Both the server and the user must know the password in order for authentication to complete.

For usability, it would be desirable for the user to use a single password at all the service providers. The obvious vulnerability there is that all the service providers could then impersonate the user at the other service providers (at which the user used the same password). To solve this problem it has been suggested (e.g., in PDM[18]) that the password at a site named "X" be a hash of the user's single password, and the string "X".

Problems with strong password solutions as currently advocated:

- The user must first install a password in a secure way at each service provider.
- If the user changes his password periodically, the user will wind up with unsynchronized passwords, and be confused about which passwords to use at which sites.

Strong password protocols might be useful as a mechanism of authenticating to the Identity Provider in a way that is more secure than sending the user's password directly, or somewhat more secure than doing a protocol such as CHAP in both directions. However, it does require changing software at both client and server machines.

3.8 Kerberos

Kerberos [16] is a secret key based system commonly used within an enterprise to authenticate non-web based clients and servers. There are a number of ways that it could be used for web authentication with different side effects.

In Kerberos-based systems, a KDC stores a secret (or with the PKINIT[30] enhancement, a public key) for each user, and a secret for each server. Each time a user wants to connect to a server, the user first contacts the KDC and obtains a (short-lived) ticket to that server.

The KDC can impersonate the user at any server in the realm (as can an Identity Provider), and although it is possible to do cross-realm authentication, Kerberos is mainly practical within an enterprise.

The initial authentication is usually done with a password, though PKINIT is used in organizations in which all employees are provided with smart cards.

A KDC is similar in some ways to an identity provider. However, with Kerberos, the ticket provides a shared key between the user's machine and the server, and a cryptographic mutual authentication exchange is done between the user's machine and the server. In contrast, with the http-based Identity Provider solutions such as Shibboleth and Liberty, the client machine is given secret information in a redirected URL, and the client machine sends that information to the server. Given that it is possible despite TLS, (for instance, by presenting a self-signed certificate claiming to be the server name and having the user click "OK") to impersonate a server to the user's machine, the Kerberos approach is somewhat more secure than the Identity Provider approaches, since the actual secret is not sent to a server.

One way to integrate Kerberos with web authentication is for an Identity Provider to use Kerberos tickets as the secrets it sends to servers. In this configuration, the browser client is unaware of the use of Kerberos. The Identity Provider goes to the KDC to get Kerberos tickets for the authenticated client to forward to the service. In this configuration, the security properties are essentially the same as with other Identity Provider schemes.

Another way to do the integration is for the client to authenticate to the server using Kerberos as an HTTP authentication mechanism within the TLS encrypted tunnel. This is most useful in scenarios where the clients and servers are already provisioned with Kerberos in support of other protocols. The security properties resemble those of other intranet protocols.

The challenge in both cases is scalability. It is logistically difficult to securely provision all of the components in the system with secret keys, and Kerberos is designed for configurations where all components are willing to fully trust a centrally managed service.

3.9 SSL Client Certificates

The SSL and TLS protocols provide for the ability to do mutual authentication (user to server as well as vice versa), but as deployed, authentication is almost always just server to user.

Public key based authentication of clients to servers eliminates the straightforward man-in-the-middle (phishing) attacks because the server learns nothing that would allow it to impersonate the user to another server. But the UI problem remains that users are unlikely to notice that they are connected to impostor web sites, so if they enter confidential information on a web page, that information might be given to an unauthorized server. Users can be tricked by web sites for several reasons:

- Confusion around server names, where a misspelled name might not be noticed, or worse yet, internationalized names can look identical to the expected name
- Confusion around URLs, where understandably users might not know which part of the URL is the real DNS name
- Faulty trust anchors, in which an evil-doer manages to trick any of the trust anchors into issuing a faulty certificate
- Also, because there are so many cases in which the web site the user wants to visit chooses to use a self-signed certificate or an expired certificate rather than go through the hassle and/or expense of obtaining a certificate from one of the commonly configured trust anchors, users have been trained to ignore security warnings around certificates and accept anything. Therefore, it would be relatively easy for a web site to present itself as any server name, using a self-signed certificate, and many users will merely accept the certificate signed by the "unknown trust anchor".

Client side PKI is not usually deployed for use between unaffiliated clients and servers over the Internet. It is used within organizations (where the user is an employee/student/etc. of the organization operating the server). Nevertheless, use of certificates between unaffiliated clients and servers is rare. There is work on Bridge CAs that might enable PKI use across organization boundaries, but it hasn't penetrated the broad population yet. There have been some CAs deployed to give users certificates. A typical method of doing this is:

- the user contacts the CA and tells the CA the user's email address and public key
- The CA sends a token to the email address
- The user, after receiving the email, sends the token to the CA (proving that the user has access to the email account)
- The CA sends the client the certificate, which is stored in the browser.

Why are client certificates not widely used in practice?

- It is confusing for users to obtain a certificate.
- Some CAs, wanting to be able to provide more assurance to their certificates, would require onerous procedures such as mailing them a notarized letter. They may also charge for certificates both initially and for renewals.
- Historically it has been hard to move certificates from browser to browser (vendors believed they were making certificate based authentication more secure by introducing this inconvenience) making roaming difficult.
- Users are likely to have different usernames with different service providers, and today browsers don't remember which certificate to send to a particular web site (the user must select it each time).
- Certificates are generally issued to some name for the user, for example, his email address, which might not be the same as the account name at the server, and might not even be stored as a property of the account, so it is not necessarily straightforward for the server to know which account is being accessed, based on the certificate.

Another issue with currently deployed PKI-based solutions is that TLS has no ability for the client to request a particular certificate from the server. If there were many CAs in the world, and a particular user chooses to trust only some subset of them, then there is no way for the server to be able to guess which certificate to send a user.

A fairly small change to TLS would enable this sort of negotiation. Such a change is proposed in [11], and a usable subset is specified in RFC4366 [2], but it has not been widely deployed.

One big advantage of PKI over the other solutions is that the CA need not be on-line, which is important for the reasons we gave in section 2.3.

3.10 Obtaining the user's private key

If user authentication is to be done using public keys, the first challenge is obtaining the user's private key(s). Unlike a password, it cannot be carried in the user's head (because user memorable passwords are nearly always weak keys, and weak private keys offer little value). Ideally a user would have a smart card, but there are problems with smart cards that have prevented them from being widely deployed:

- Users lose them, and need a backup plan for when the smart card is either not with the user, or permanently lost.
- Not all machines have a method for attaching a smart card.

If the user does not have a smart card, another approach is to store **{priv}pwd** (the user's private key encrypted with the user's password), in some location on the net (for instance a directory). There are various means of downloading the private key to the machine the user is using, using only a password.

- The simplest scheme is to have **{priv}pwd** be world-readable. If the user's password were sufficiently strong this would even be reasonably secure, but most users do not choose passwords strong enough to withstand an offline dictionary attack. Kerberos version 4 [14] was deployed in this mode (anyone could request a credential for the user which would enable an offline dictionary attack).
- The next enhancement is to have a simple challenge/response protocol (perhaps in a TLS-protected exchange based on server-side authentication) in which the user proves knowledge of her password before the server sends **{priv}pwd**. This is similar to the "pre-authentication" that was introduced in Kerberos version 5[16]. Someone impersonating the server would be able to do a dictionary attack, but others would only be able to do an on-line attack. Without TLS protection, an eavesdropper would also be able to launch a dictionary attack. However, in practice, it is more difficult to arrange to be eavesdropping on the private key download exchange than merely to send a message to the server requesting it to send **{priv}pwd.** When used with TLS server-side authentication, this approach would be quite deployable and quite secure today. Various schemes for doing this have been proposed (for example, Pvault [5] and SACRED [6]).
- Another approach would be to use a strong password-based credentials download protocol such as in [17].

Ultimately, of course, it would be desirable for users to have smart cards. USB connections are nearly ubiquitous today, and there are USB-based smart cards. Or phones can communicate with a work station using bluetooth technology.

With strong password techniques, someone impersonating the server or client can only do an on-line dictionary attack, and someone eavesdropping gets no opportunity for a dictionary attack. However, if someone were to steal the server database, then one would be able to do an off-line attack, since the server must store the user's private key encrypted with a password.

4. TOWARDS A USER-CENTRIC PKI

In this section we describe a number of ways of addressing, not only the single sign-on problem, but also the issue of providing user information in a way that is convenient, safe, and under control of the user. One aspect of our approach, doing mutual authentication without using 3rd party CAs, is philosophically similar to that used for email in [9], as well as to self-issued cards in CardSpace. We also address other issues, though, including one-stop revocation and switchover to a new user credential, that have not been addressed before.

4.1 User Information/User "Wallet"

Once a user has obtained a private key (either with a smart card or having downloaded the private key from the net and decrypted it with a password), we propose that information about a user be available on the net. We will call this information the user's "wallet". It can contain information such as the user's credit card number, address, and passport number. We will advocate that it also carry information that can enable any machine to recreate the environment of the user's home machine, such as browser bookmarks, preferences, and long-lived cookies.

This requires the installation of support software (perhaps a browser plug-in) on the host machine, which should be easy for the user's own machine and could be constructed so as to be safely installable on a kiosk machine.

The Liberty approach has user information stored at a server and retrieved by other servers. The server that stores the user's information has access to all the user's information at all times, so if that server is broken into, all user information for all users can be stolen.

Another approach is to store this information in the user's machine. The problem with this is that this information would not be available if the user were working at a different machine.

We advocate that the user information be stored at a server in the web, but that this information be encrypted with the user's public key. This information would therefore not be accessible to the server on which it was stored. This greatly increases security for the user.

We also claim that it will be easier for the user to reliably and conveniently decide policy for which web sites should be allowed to access which information than it would be if the user had to set up the policy at a remote server.

The user's machine should download and decrypt the user's wallet. When a web site requests information, the user's machine can let the user know what information is flowing, and the user can make decisions such as "let anyone see this information", "don't let anyone other than my machine access this information", "always let this web site see this information". It might be a burden if the user must make a yes/no decision for every piece of information, every time, but it can be made quite convenient, for instance, by having the wallet software fill in all the fields in a form, and then let the user change or erase fields. Also, for fields like "credit card number", the user could select from a list of credit cards.

This approach does require reaching a 3rd party in order to download the wallet (2.3) when logging in from a new client platform. However, this access only needs to take place once per login session (or once ever, from the user's home machine), as opposed to solutions such as those in section 3.3, which require accessing the identity provider for a token every time a service provider is accessed.

4.2 Least Privilege

Suppose a user is using a less trusted machine, and only wants to unlock a subset of his personal information. For instance, he may want his name and address book, but not credit card information.

If all information in the user's wallet is unlocked simultaneously, then there would be no way for the user to get enough information to print a boarding pass at the machine in the hotel lobby without also giving that machine credit card numbers and access to all the confidential files which the user is authorized to access.

This problem can be solved by having some of the information in the wallet be protected with an additional level of encryption, most likely based on a password. It would be burdensome to have every item of information separately locked, but it would be practical to have some of the information directly unlocked by the private key, and other information require an additional key to unlock. Or to have separate wallets stored on the network, one for low-risk information, and one for higher-risk information. The user would separately authenticate and download the various wallets, presumably with different credentials.

For instance, if it is desired to support authorization based on which authentication method was used (as described in section 2.5), there might be a wallet that is unlockable with a password, and more secure information might be only unlockable with the private key in the user's activated smart card.

4.3 Establishing a relationship with a service provider

So far the user has a private key, and no certificates. Certificates are not strictly necessary for public key authentication. We propose instead a scheme where each server either certifies or stores the user's public key.

A user establishes an account with a web site through the following steps:

- The client machine makes an initial contact with the server, using current server-side authentication to be reasonably certain the user has reached the correct server.
- The user creates an account. The account is given a username, through some mechanism such as is done today. The account name on that server is stored in the user's wallet. If the user creates multiple accounts on the same server, the wallet keeps track of all the usernames, and software allows the user to select from these when the user visits that service provider.
- The user's machine sends the server machine the user's public key. The server either stores this information with the user's account, or certifies the public key and sends the client machine the resulting certificate, which the user stores in the user's wallet.
- The server sends its public key to the client machine. The client either stores the merchant's public key in the user's wallet, or certifies the public key and sends the server the resulting certificate.

4.4 Connecting to a service provider

Now assume that a user already has an account with a service provider. The user's machine has first downloaded the user's wallet. The user asks to connect to a service provider. The wallet software finds the account names for that user at that service provider, and if there is more than one, asks the user to select one. The user's machine and the server perform mutual authentication based on the server's and user's public keys.

Mutual authentication will be done with public keys. Either each side will have stored the other side's public key, and there is no need for certificates, or one or both sides can instead store a certificate signed by the other side, and present that certificate during subsequent visits.

When would it be the case that the service provider would sign a certificate for the user's key (and have the user store this certificate and present it during future authentications), rather than simply storing the user's public key with the account? A good use case for this is when the service provider might not store any information associated with the account, but instead sign a certificate authorizing use of the account to a particular public

key. For instance, a user might pay a site that stores information a monthly subscription fee, allowing unlimited free downloads during that month. The user might pay the site with anonymous cash, and create a public key (which the user keeps in her wallet) for use at that site, along with the certificate from the site which authorizes that public key to use the service for a specified time.

When might it make sense for the service provider to send the client machine a certificate (signed by the user), rather than having the user store the server's public key in the user's wallet? A possible reason for this is to keep the user's wallet smaller.

4.5 Unlinkable Accounts

A user might want it to be difficult (or preferably impossible) for two web sites to compare accounts and discover which accounts are for the same identity. If the user uses the same public key at different sites, it is evident that the accounts belong to the same user.

For this reason, a user might want to use different public keys to authenticate to different sites. This can be accomplished by using the "main" private key (the one stored in the user's smart card), for unlocking the user's wallet, and storing other public key pairs, and the sites on which each public key is to be used, in the user's wallet.

CardSpace automatically creates a different public key for signing self-issued cards, for each service provider.

4.6 Revocation of the user credentials

Suppose the user's smart card were stolen, and he wishes to change his private key. With our scheme, there is direct authentication between the user and a server. The user is acting as his own CA, and therefore there needs to be some sort of revocation mechanism. In CardSpace, in the mode where the user is his own identity provider, revocation would need to be done manually, with the user remembering every site with which he'd registered with the stolen credential, and having some method of authenticating, (without the stolen credential) to assert the authority to revoke the old credential.

This model is impractical, both because the user will not remember all the sites, and because the process of individually revoking the old credential at each of those sites would be onerous.

We advocate a one-stop revocation mechanism by use of a "revocation service". Such a service is trusted to inform servers about revocation status of a user's public key.

If a user needs to revoke his public key, he contacts the revocation service, authenticating with some sort of method other than using his private key (because he likely does not have access to his private key when he needs to revoke it). This might be with a phone call in which they call his home phone back, or through an email interaction similar to what is done today for password resets. This need not be super secure because the threat is a denial of service threat, not an impersonation threat, and the user has to go through the same sort of procedure as he would have needed to do in order to establish the account with revocation service in the first place.

In our scheme, when the user registers with a web site (see section 4.3), in addition to telling the web site his public key, he also tells the site the URL for the user's revocation service.

There are several possible ways in which a site can find out about the revocation status of a user's public key:

- Each web site can check the revocation status of each user account, communicating with that user's revocation server periodically, or each time the user logs in.
- The web site can register with the revocation service when the user creates the account, so that the revocation service will notify each of the user's web sites if the user's key has been revoked.

There is a potential privacy issue. The user might not want the revocation service to know all the web sites that the user is registered with. However, if he uses different keys for different sites, and possibly even different revocation servers for the different public keys, this will help. All revocation schemes face painful performance vs. timeliness trade-offs.

4.7 Key Rollover

Although section 4.6 addresses the problem of informing all the relevant service providers of the invalidity of the user's old public key, the user will want to be able to resume use of his account with a new public key.

One method of doing this is as follows:

- Before the user's current public key becomes unusable, the user creates a "next" public key pair.
- He certifies the next public key with his current private key. Call this certificate the "next key" certificate.
- He escrows the "next" private key by breaking it into n shares, with a quorum of k [22], encrypts each share with the public key of an escrow agent, and stores all the encrypted pieces in his wallet, or at some trusted storage location on the net, along with the certificate signed by his current private key that delegates to the next public key.
- If he needs to revoke his key, he obtains the escrowed "next" private key (through some out of band method of authenticating to the site that is storing the escrowed pieces, or authenticating to the escrow agents), and stores the "next key" certificate at the revocation service URL.

A web site that is checking for the user's key validity will either retrieve from the revocation service URL in the user's account a status of "public key is P", or one or more "next key" certificates.

If a web site has not checked the validity of the user's public key recently, it might encounter a string of "next key" certificates. The web site might be storing P for the user, and the revocation service URL might contain "P", "next key is P1" (signed by P), "next key is P2" (signed by P1), and so forth.

While the reassembly of the "next key" and revocation of the previous one can be automated, the process by which a user chooses escrow agents and later authenticates to them cannot. It's intended that the genuine user can complete this procedure while the person who has stolen the user's credentials cannot; that makes it inherently messy and fragile. Making the user experience straightforward, secure, and robust will be a real challenge.

4.8 Preventing Loss of the Wallet Information

If the user loses his smart card, and the user's wallet is only stored encrypted with the key in the smart card, the user will lose all this important state.

For this reason, it would be valuable to escrow the user's private key, or escrow some other key with which the user's wallet information is encrypted, if the user would not like to share knowledge of his private key with the escrow agent.

Escrow of a key can be done quite securely. The key can be broken into shares using a quorum scheme [22] such that k out of n shares can recover the secret. Each of the shares is then encrypted with the public key of each of n escrow agents, and stored someplace safe. The encrypted pieces need not be given to the escrow agents until the key needs to be recovered.

4.9 Authenticated Attributes

Some of the systems we covered in section 3, (e.g., Shibboleth) have some capability for authenticated attributes, such as proving membership in an organization whose members are allowed to use the service for free. In theory a user could acquire signed credentials, by someone authorized to assert such attributes, and present them to a service provider when required to access a service. These credentials could be acquired at time of access to the service (as in Shibboleth), or collected and stored in the user's wallet as certificates (asserting the attribute to a public key owned by the user). If acceptable to the service, the user's privacy could be protected by only proving the required attribute and not revealing the user's identity. There are issues, as we discussed in section 3.6, such as how a user finds a site that is trusted by the service provider to assert the attribute. In theory there might be a single site that is completely trusted by a service provider to make decisions about all user attributes. This attribute assertion site could make complex decisions about all the attributes, in whatever way is appropriate for each, and then simply make assertions to the service provider.

A fully general solution to authenticated attributes has not been proposed or deployed, and is largely orthogonal to what we propose here. If solved, it could be added to the authentication, revocation, and credential rollover solutions we propose here.

5. CONCLUSIONS

In our solution there is no need for federations of identity providers and service providers. Also, no traditional PKI is needed, except for initial contact by a user with a service provider as provided with TLS today (and which would be a necessary step in an Identity Provider solution as well).

The basic ideas are:

- The user carries his private key around on a smart card, or downloads it from a server.
- Once the user obtains his private key, he downloads the rest of his "wallet" information from a server, which stores it encrypted with the user's public key.
- When a user wishes to enroll with a web site, the user's machine exchanges public keys with the server, and either certifies the server's key with the user's private key, or stores the server's public key in the user's wallet. Likewise, the server either stores the user's public key, or certifies the user's key and gives the user's machine the certificate to store and present on future visits.
- Authentication to a server with which the user has an ongoing relationship is done using public keys.
- When the user creates an account at a service provider, in addition to telling the service provider the user's public key, the client also tells the service provider an associated revocation service.
- The service provider registers with the revocation service if it wishes to be notified when the user revokes his public key. (or else, the service provider checks periodically).
- If the service provider registers with the revocation service, the revocation service informs the service provider if the user's key has been revoked.

Properties of this approach:

- The user has the perception of single sign-on, which is done when the user activates and inserts his smart card, or downloads her private key from a server (goal 2.1).
- Authentication to web sites is strong, so no credentials can be compromised by phishing (goal 2.2).
- Contact between a user and a server can be done directly. There is no need for redirection to and from identity providers. This is more efficient (goal 2.3)
- Contact between a user and a server works even if no other machines other than that server are currently reachable (goal 2.3).
- User information is still available from anywhere on the web, but encrypted, so that the user's information is safe and directly under the user's control (goals 2.3, 2.5, and 2.6).
- There is no on-line trusted service whose compromise can enable user impersonation (goal 2.3).
- There is no on-line trusted service whose compromise can leak private information for large numbers of users (goal 2.3).
- Credential backup through a quorum of backup agents safely protects against credential loss (goal 2.7)

These enhancements do require changes to both clients and servers (failing to meet goal 2.8), but they can be deployed incrementally with benefits accruing with each step.

6. ACKNOWLEDGMENTS

We wish to thank Eve Maler, Ray Perlner, and the anonymous reviewers for helping with background information and helpful comments.

7. REFERENCES

[1] Bellovin, S. and Merritt, M. 1992. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, May 1992. pp. 72-84

[2] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and Wright, T., 2006. Transport Layer Security (TLS) Extensions. Internet RFC 4366 April, 2006. DOI=http://www.ietf.org/rfc/rfc4366.txt?number=4366.

[3] Cantor, S., Kemp, J., Philpott, R, and Maler, E. 2005. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard March 2005. DOI=http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[4] CardSpace. DOI=http://msdn2.microsoft.com/en-us/library/aa480189.aspx.

[5] Chandra, R., Mehrotra, S., and Venkasubramanian, N. 2005. Pvault: A Client Server System Providing Mobile Access to Personal Data. Proceedings of the 2005 ACM workshop on Storage security and survivability (StorageSS), 2005.

[6] Farrell, S. 2004. Securely Available Credentials Protocol. Internet RFC 3767 June, 2004. DOI=http://www.ietf.org/rfc/rfc3767.txt?number=3767.

[7] Farrell, S. and Housley, R. 2002. An Internet Attribute Certificate Profile for Authorization. Internet RFC 3281 April, 2002. DOI=http://www.ietf.org/rfc/rfc3281.txt?number=3281.

[8] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and Stewart, L. 1999. HTTP Authentication: Basic and Digest Authentication. Internet RFC 2617. June, 1999. DOI=http://www.ietf.org/rfc/rfc2617.txt?number=2617.

[9] Garfinkel, S., Miller, R., "Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express", Symposium on Usable Privacy and Security, 2005.

[10] Herzberg, A. and Gbara, A. 2004. Security and Identification Indicators for Browsers against Spoofing and Phishing Attacks. Cryptology ePrint Archive, Report 2004/155. DOI= http://eprint.iacr.org/2004/155.pdf.

[11] Hess, A., Jacobson, J., Mills, H., Wamsley, K., Seamons, K.E., and Smith, B. 2002. Advanced Client/Server Authentication in TLS. Network and Distributed System Security Symposium, San Diego, CA. February 2002.

[12] Housley, R., Polk, W., Ford, W., and Solo, D. 2002. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Internet RFC 3280 April, 2002. DOI=http://www.ietf.org/rfc/rfc3280.txt?number=3280.

[13] Jablon, D. 1996. Strong Password-Only Authenticated Key Exchange. Computer Communication Review, Vol. 26, no. 5, ACM SIGCOMM. October 1996. pp. 5-26

[14] Kaufman, C., Perlman, R., and Speciner, M. 2002. Network Security: Private Communication in a Public World. Prentice Hall PTR. pp 307-336.

[15] Morgan, R., Cantor, S., Carmody, S., Hoehn, W., and Klingenstein, K. 2004. Federated Security: The Shibboleth Approach. Educause Quarterly. pp. 12-17

[16] Neuman, C., Yu, T., Hartman, S., and Raeburn, K. 2005. The Kerberos Network Authentication Service (V5). Internet RFC 4120. July, 2005. DOI=http://www.ietf.org/rfc/rfc4120.txt?number=4120.

[17] Perlman, R. and Kaufman, C. 1999. Secure Password-Based Protocol for Downloading a Private Key. Proceedings of the 1999 Network and Distributed Systems Security. February 1999.

[18] Perlman, R. and Kaufman, C. 2001. PDM: A new Strong Password-based Protocol. 10th USENIX Security Symposium, August 2001.

[19] Ross, B., Jackson, C., Miyake, N., Boneh, D., Mitchell, J., "Stronger Password Authentication Using Browser Extensions", 14th Usenix Security Symposium, 2005.

[20] Rubenking, N. 2003. PC-Mac PasswordVault 2.0. PC Magazine, 2/13/03.

[21] Scavo, T. 2005. Shibboleth Architecture: Technical Overview. Working Draft 01, January 9, 2005. DOI=http://shibboleth.internet2.edu/docs/draft-scavo-shibtechoverview-01.pdf.

[22] Shamir, A. 1979. How to Share a Secret. CACM vol 22. no. 11, pp 612-613.

[23] Simpson, W. 1996. PPP Challenge Handshake Authentication Protocol (CHAP). Internet RFC 1994. August, 1996. DOI=http://www.ietf.org/rfc/rfc1994.txt?number=1994.

[24] Sxipper, http://www.sxipper.com.

[25] Tourzan, J. and Koga, Y. 2004. Liberty ID-WSF Architecture Overview. Version 1.0. Liberty Alliance Project.

[26] Web Services Architecture. 2004. W3C Working Group Note. DOI=http://www.w3.org/TR/ws-arch/.

[27] Wu, M., Miller, R., and Garfinkel, S. 2006. Do security toolbars actually prevent phishing attacks? Proceedings of the 2006 SIGCHI conference on Human Factors in computing systems. 601-610. DOI= http://portal.acm.org/citation.cfm?id=1124863.

[28] Wu, T. 1998. The Secure Remote Password Protocol. Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium. March 1998. pp. 97-111

[29] Yee, K., Sitaker, K., "Passpet: Convenient password management and phishing protection", Proceedings of the second symposium on Usable privacy and security (SOUPS), 2006.

[30] Zhu, L. and Tung, B. 2006. Public Key Cryptography for Initial Authentication in Kerberos (PKINIT). Internet RFC 4556. June 2006. DOI=http://www.ietf.org/rfc/rfc4556.txt?number=4556

User-centric PKI

Radia Perlman
Radia.Perlman@sun.com

Charlie Kaufman
charliek@microsoft.com

1

Motivation

- Why can't things be simple?

2

Motivation

- Why can't things be simple?
- I can't cope with username/pwds
  - I'm not alone…
- The federated identity things seem really complicated to me

3

I don't care about formats

- "Certificate" is any signed thing

4

My view of federated things

- Microsoft created the "Passport" vision, with Microsoft the center of the world
- Others said, "Hey, let's not anoint one organization to be an eternal monopoly
- So, the notion of lots of IDPs, and a federation is the set of SPs that trust that IDP

5

If there is just one IDP

- User authenticates to that IDP
- That IDP vouches for the user at all the affiliated sites

6

But what if there are hundreds?

- And what if the SPs the user wants to use affiliate with different subsets of them?

7

And what value does the IDP give, anyway?

8

Downside of IDP (vs. peer-to-peer mutual authentication)

- Security (on-line IDP can impersonate all users)
- Availability (if IDP is down, nothing works)
- Performance (bouncing around between boxes)
- Privacy (IDP knows everyone you talk to)

9

Upside of IDP

- ?

10

Quick rant on today's web

Perils of Perlman

Buying something

- Scenario: Buy something from a merchant you haven't bought from recently
- All prepared with your info, credit card, etc.
- It asks you for your email address…

13

You're a returning user!

- Type your username and password!

14

You're a returning user!

- Type your username and password
- Of course you can't remember it, so…

15

You're a returning user!

- Type your username and password
- Of course you can't remember it, so…
  - you manage to find "recover username"

16

You're a returning user!

- Type your username and password
- Of course you can't remember it, so…
  - you manage to find "recover username"
  - suddenly you are in a Monty Python movie
    - Answer the following questions three:
      - Telephone number
      - Address
      - Mother's maiden name

17

New Rule

- It should be no more onerous to be a returning user than a new user

18

Security questions for password/username recovery

- Favorite sports team
- 2nd grade teacher's name
- Pet's name
- Father's middle name
- My middle name

19

New Rule

- Security questions must be specifiable by the user
- I'd say "or selectable from a very large list", but I'm sure they can come up with an arbitrarily long list of questions I can't answer

20

Security question in comedy routine

(Q&A Chosen by user, to be asked by the bank for phone verification of customer)

21

Security question in comedy routine

- Question: “Are you wearing underwear”?

22

Security question in comedy routine

- Question: “Are you wearing underwear”?
- Answer: “I don’t think that’s an appropriate question”

23

Keeping customer information

- I do not want to do “single click ordering”
- I do not mind typing in my address
- I do not mind typing in my credit card number
- Merchants insist on keeping all of this information
- And eventually this information gets stolen

24

New Rule

- After a merchant is paid, any subset of information about a customer (including all of the information) must be expunged by the merchant at the customer's request

25

Simple intra-organizational PKI

26

Within an organization

- Should be trivial, single CA
- To create an account
  - Sysadmin told username and initial pwd
  - Types that into "create new account" tool
  - Tool generates key pair, certifies public key, encrypts private key with pwd, stores cert in dir
- User logs in
  - Types name and pwd, retrieves private key
  - Accesses resource: authenticates with public key

27

Better with smart cards

- Badges have smart cards. What's the problem?
- Doesn't do mutual authentication, but could, by having everything (including client machine) know CA's public key

28

And, IDP and Kerberos also work

- Within an organization, also easy to have everything trust the same KDC, or IDP
- But better without having an online trusted box

29

Online vs offline trusted box

- Performance, availability
- Security
  - Can impersonate all users
  - More likely to be compromised vs offline box
  - Knows who is talking to who
  - May have database that if stolen, can compromise users

30

But our talk is really about individuals

31

How about individuals?

- Think of this as just doing what we do with username/pwd, but more securely, and without torturing the user
- Assume first the user has a smart card with a secret (private key, or secret key)

32

"Wallet"

- A bunch of data cryptographically protected with the user's smart card secret
- Downloadable from one or more places
- Contains, for instance, public keys of various merchants, perhaps private keys to use with that merchant, information such as passport number and credit card numbers

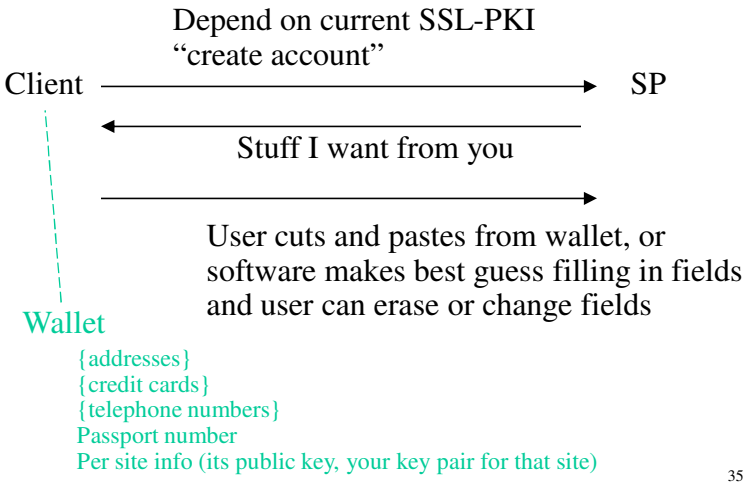
33

Enrolling at a site

- Just like today, except username/pwd is replaced by "public key"
- The wallet information (such as address) can be filled into the form, to save the user typing, or the user could drag info she wants into the form
- The SP sends the user its public key

34

Enrolling



35

Note

- Instead of enrolling with a username and password, your account name is your public key, and you authenticate with your public key
- And by saving the SP's public key (a la SSH), you can do mutual authentication, knowing you are again reaching the same site as before

36

Revisiting the site

- Mutual authentication using public keys (e.g., SSL with client certs)

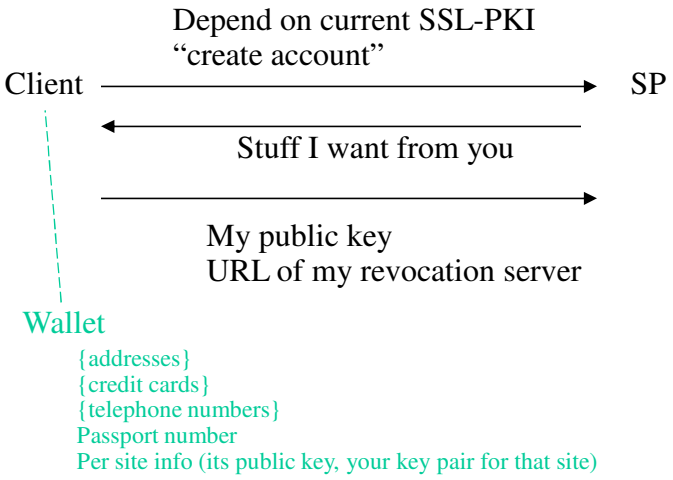
37

One-step revocation

- Suppose you are using your public key at lots of sites
  - (not sure how useful different keys for each site is)
- And someone steals it
- Use “revocation service”

38

Enrolling



39

Revocation service

- SP learns user's revocation server along with the user's public key
- SP can "enroll" with that revocation service, to be notified in case of revocation
- Or SP can check periodically
- User has to have some sort of out-of-band mechanism to authenticate and revoke the key
- User can store {next keys} signed by current key, and escrow the future private keys

40

Authenticated attributes

- User can have, in wallet, certs signed by whoever is trusted to assert the attribute, that a public key associated with the user is over 18, a citizen, whatever
- Can send such certs to SP when needed, along with proof of knowledge of the private key

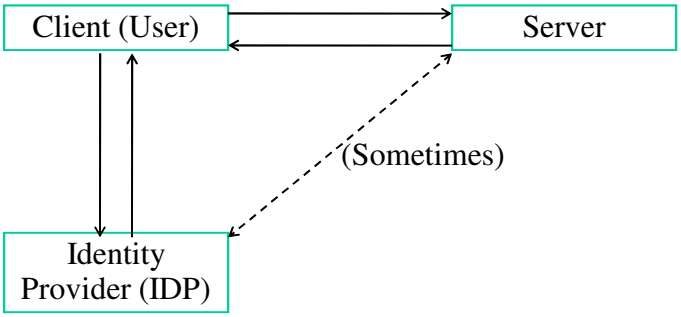
41

Yes, things can go wrong

- Establish trust, then after increasingly large purchases, skip town
- Credit cards today somehow work "well enough" – certainly could be improved, but banks seem to think it's not worth the bother

42

Federated Identity: What is it?



Many variations on this theme: IDP holds secret information needed by client to authenticate to server.

43

Federated Identity: What problems are we trying to solve?

- Something more convenient for the user than username/password?
- Something more secure than username/password?
- Reducing aggregate administrative costs of maintenance?

44

Federated Identity: What problems are we trying to solve?

- Something more convenient for the user than username/password?
- Something more secure than username/password?
- Reducing aggregate administrative costs of maintenance?
- Something new we can implement, publish papers about, and use to excite customers?

45

What do users want?

- Don't want to remember so many usernames and passwords
- Don't want to type usernames and passwords as often
- Don't have to type address/phone number/email address as often
- Less hassle when I forget a password or someone steals it

46

What do users not want to give up?

- Sign-on from anywhere
- Not needing to carry smart cards or other hardware
- Ability to have lots of accounts at a single vendor, share some accounts (without sharing all of them)
- Having more than one credit card
- Having more than one email address, phone number, etc. and choosing which to give to a site

47

Users' Security Goals?

- For most, vaguely understood worries about 'identity theft' or 'stolen credit card numbers'
- For a few of us geeks:
  - Servers can't correlate identities
  - Some degree of anonymity
  - No authority can know all the things I do
  - Not putting 'all eggs in one basket' in case of kiosk or hijacked machine

48

Reducing Administrative Costs

- Users forget usernames & passwords
  - Centralize password reset
  - Less often if fewer to remember
- Recovery when passwords are stolen
  - Revocation in bulk rather than site by site
  - Correlate suspicious activity
- A wallet full of smart cards is impractical
  - A single hardware token should work with many services

49

Service Provider Security Goals

- Authentication that's harder to compromise
  - Get users to choose hard to guess passwords and not use the same ones on multiple sites
  - Make phishing attacks and other network based attacks harder to mount
- Discourage users from sharing subscriptions
- Make it easier to track down users who misbehave

50

Special Cases

- My credit card number should not be secret
  - Purchases should be authorized through some strong protocol
  - Issuing banks or alternate intermediaries like PayPal should deploy this – should not be linked to other aspects of federation
- If I get a AAA or AARP or ACM discount, proof of membership should be partly automated

51

The Allure of Federated Identity

- Authenticated attributes (sometimes with anonymity)
  - Classic examples:
    - Age > 21
    - Employee of Microsoft
    - Member of IEEE
    - Paid attendee of IDTrust 2008
- What is the end-to-end scenario in which this is useful?

52

Bottom Line

- Current State of the Art is awful!
- Something simple based on roaming credentials and SSL client certificates could solve the most important problems
- Federated identities could potentially solve those same problems, but they are harder to deploy and no better

53

Bottom Line

- Current State of the Art is awful!
- Something simple based on roaming credentials and SSL client certificates could solve the most important problems
- Federated identities could potentially solve those same problems, but they are harder to deploy and no better
- Why is the world so excited about them?

54

Federations in R&E:
Current Soup and Future Bread

INTERNET2®

Topics

- The Research and Education Sector
- Shibboleth
- InCommon and international R&E federations
- Next Steps
  - Inter-federation soup
  - Collaboration management platforms and the attribute ecosystem

The Research and Education Sector

- Intense need to collaborate
  - For both research and education
  - Inter-institutional and international
- Historic roles
  - Innovator in networking
  - Technology transfer
  - Shaping a new generation of consumers

R&E Engagements

- TCP/IP, first as a technology and then as a market-maker
- SAML/Shibboleth, first as a technology and then as a market-maker
- Collaboration tools and collaboration management platforms

Shibboleth

- Deployments > 10,000; countries > 20
- Shib 2.0 release ~ Mar 4, 2008
- Landmark release; strong platform for years to come, on which considerable enhancements will be broadly provided
- “More interoperable with commercial SAML systems than they are with each other”
- OpenSAML 2.0 already heavily used by Verisign, Tata, etc.
- It is suggested that federations migrate to 2.0 sooner than later.

Shib and OpenId

- Shib 2.0+ containing an OpenId provider under discussion
- Shib 2.0++ may contain more clever and useful integration of federated and ad hoc identity management
- The OpenId platform within Shib will have a warning reminding applications to use caution in their consumption of external identities.
- UK leading development, with broad conversation, but OpenId is generally a US concern.

Missing pieces

- End-user attribute release management
  - InfoCard?
  - Attribute release editors within Shib
- Dynamic metadata (not dynamic trust)
- N-tier tokens

InCommon

- Approximately 80 members and growing steadily
- More than two million “users”
- New types of members
  - {St. Mary’s of the Plains}
  - National Institute of Health
  - Student service providers
  - Energy Labs
  - MS, Apple
- On third generation of Steering Committee
- Has less value right now than it will…

The DreamSpark Nightmare

- Microsoft delivery of developer kits, source code, etc to students
- Shows how far down you can bury federation in the user experience (and way under a LiveId and custom WAYF)
- A learning experience for us all
- Showed the value in implementing the full features of Shib (such as the ePTId generator)
- http://www.pcworld.com/article/id,142597-c,software/article.html
- https://downloads.channel8.msdn.com/
- Asserting “studentness” is highly useful

InCommon Next Steps

- New members
  - MS, Apple, Michigan, University of Mary Washington, William and Mary, Lafayette, Emory, Richmond
  - Pending – MIT, Google, student service companies, medical consortia, 1200 institutions from the National Student Clearinghouse
- InCommon Silver
  - LOA-2
  - Not hard but lots of thought upfront – profiles, audit guides, etc.
  - Rich new set of applications from NIH
  - Intended to be ultimately revenue neutral
  - Remarkable barn-raising experience so far…

Federations

- Almost everywhere now
  - Internationally – UK (2-3 new members a day), Spain, France, Sweden, Finland, Switzerland, Netherlands, Germany, Denmark, Norway, Australia, Brazil, Japan, Canada, etc.
  - State university systems
  - Community college libraries
  - Medical associations
  - DoJ and DoD
- All do SAML; most are Shib
- Limited interfederation interactions – Kalmar Federation, UK-Australia, MS, Elsevier

International federation highlights

- Several countries at 100% coverage, including Norway, Switzerland, Finland
- Community served varies somewhat by country, but all are multi-application and include HE
- UK intends a single federation for HE and Further Education ~ tens of millions of users
- Real use cases involving international team science now driving interfederation peering urgency

International Activities

- http://www.terena.org/activities/refeds/
  - A summary of discussions among R&E networks, including a survey of national efforts
- http://www.jisclegal.ac.uk/access/
  - Excellent policy analytics, especially around international issues of privacy, peering, and attributes
- http://ec.europa.eu/idabc/
  - TransEuropean activities in IdM for use among citizens, governments, and businesses

IDABC

- IDABC stands for **I**nteroperable **D**elivery of European eGovernment Services to public **Administrations**, **Businesses** and **Citizens**.
- http://ec.europa.eu/idabc/en/document/6484/5644
- eID Interoperability for PEGS -Report on interoperable eIDM technical solutions, December 2007 (http://ec.europa.eu/idabc/servlets/Doc?id=29619). Offers technical assessment of several technologies
- Final recommendations due soon.
- Federated approaches are likely; open source standards may be identified

Interfederation

- We used to know more…
  - We thought there was primarily peering and we could do that
  - Things changed…
- A rich mix of emerging relationships – nested, leveraged, peered, orthogonal, etc.

Some of those relationships

- Nested – UC Trust and InCommon
- eduRoam – single application cross-federation
- Texas
- Multi-homed SP –Microsoft, Elsevier, student service industry, etc.

Peering

- Efforts between InCommon and EAuth collapsed a while ago
  - We got close, but EAuth priorities changed
- International Peering
  - UK Feasibility analysis
  - Attribute Alignment
  - Privacy due out in May
  - Peering drafts to follow

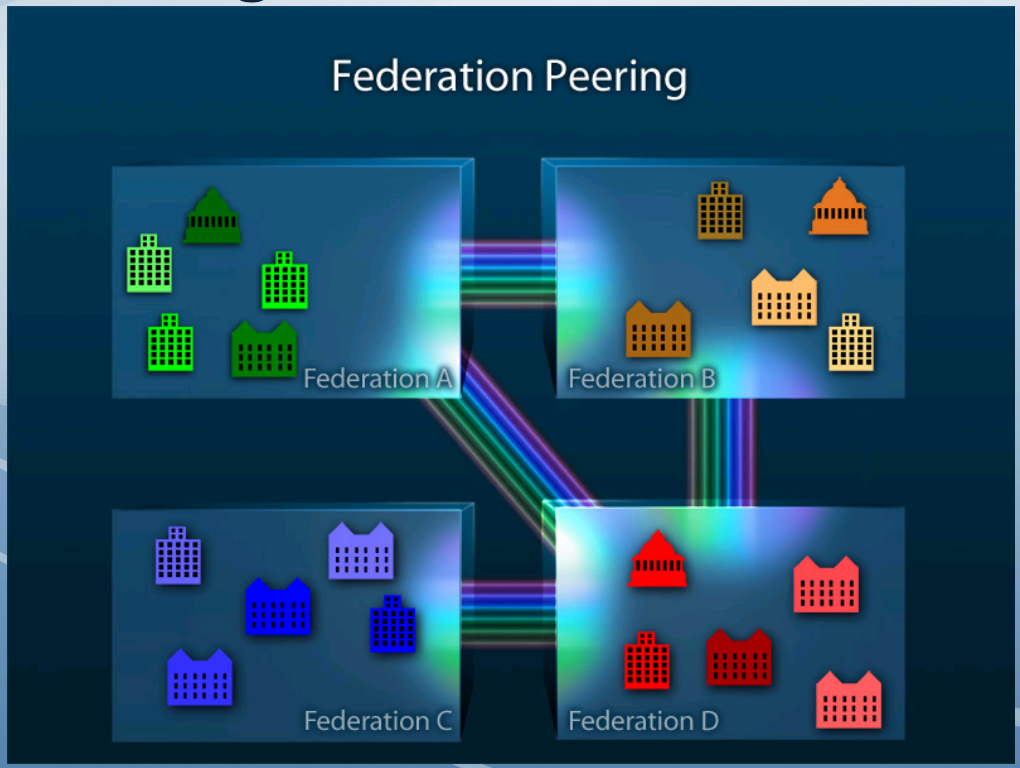
Some of the bases to touch in peering

- Typical issues -
  - Problem resolution and adjudication, liability and indemnification, financial considerations, impact on member agreements, etc.
- New issues -
  - Metadata exchange
  - Attribute mapping
  - Transitive trust

UK Bilateral Interfederation Template

- Purpose, scope and limits of agreement
- Entity assurance
- Member-operator behavior
- Problem resolution
- Member-member behavior
- Interfederation infrastructure

Peering Parameters



Parameters:

- LOA
- Attribute mapping
- Legal structures
  - Liability
  - Adjudication
- Metadata
  - VO Support
- Economics
- Privacy

Federation Soup

- Workshop to held early June, with NSF support likely
- Bringing together all manners of federation to figure out federation relationships
  - InCommon, JISC, state federations, library federations, university system federations, grid federations, etc.
  - Topics include alignment of policies, technologies, attributes, metadata, etc.
- Approaches include peering, nested, leveraged, and a whole lot of ad hoc
- Outputs may include best practices, multi-homing, etc.

Emerging key themes

- Privacy and consent
- Privacy and the aggregation of attributes
  - The glory of ePTID
- Support of collaboration and virtual organizations
- Complexity
  - Lack of federal law coupled with state laws

Future issues

- Filling in some of the missing policy pieces
- Dynamic metadata, maybe dynamic trust
- End-user experience – wayf and privacy manager
- Attribute mapping
- Federations as trusted metadata sources
- Inter-federation
- Collaboration management platforms and the attribute ecosystem

Collaboration and Federated Identity

- Two powerful forces being leveraged
  - the rise of federated identity
  - the bloom in collaboration tools, most particularly in the Web 2.0 space but including file shares, email list procs, etc
- Collaboration management platforms provide identity services to “well-behaved collaboration applications”
- Results in user and collaboration centric identity, not tool-based identity

Collaboration Management Platforms

- Management of collaboration a real impediment to collaboration, particularly with the growing variety of tools
- Goal is to develop a “platform” for handling the identity management aspects of many different collaboration tools
  - Platform includes a framework and model, specific running code that implements the model, and applications that take advantage of the model
  - This space presents possibilities of improving the overall unified UI as well as UI for specific applications and components.

Comanage

- A collaboration management platform, supported in part by a NSF OCI grant, being developed by the Internet2 community, with Stanford as a lead institution
- Open source, open protocol
- Uses Shibboleth, Grouper, and Signet
- Parallels activities in the UK and Australia

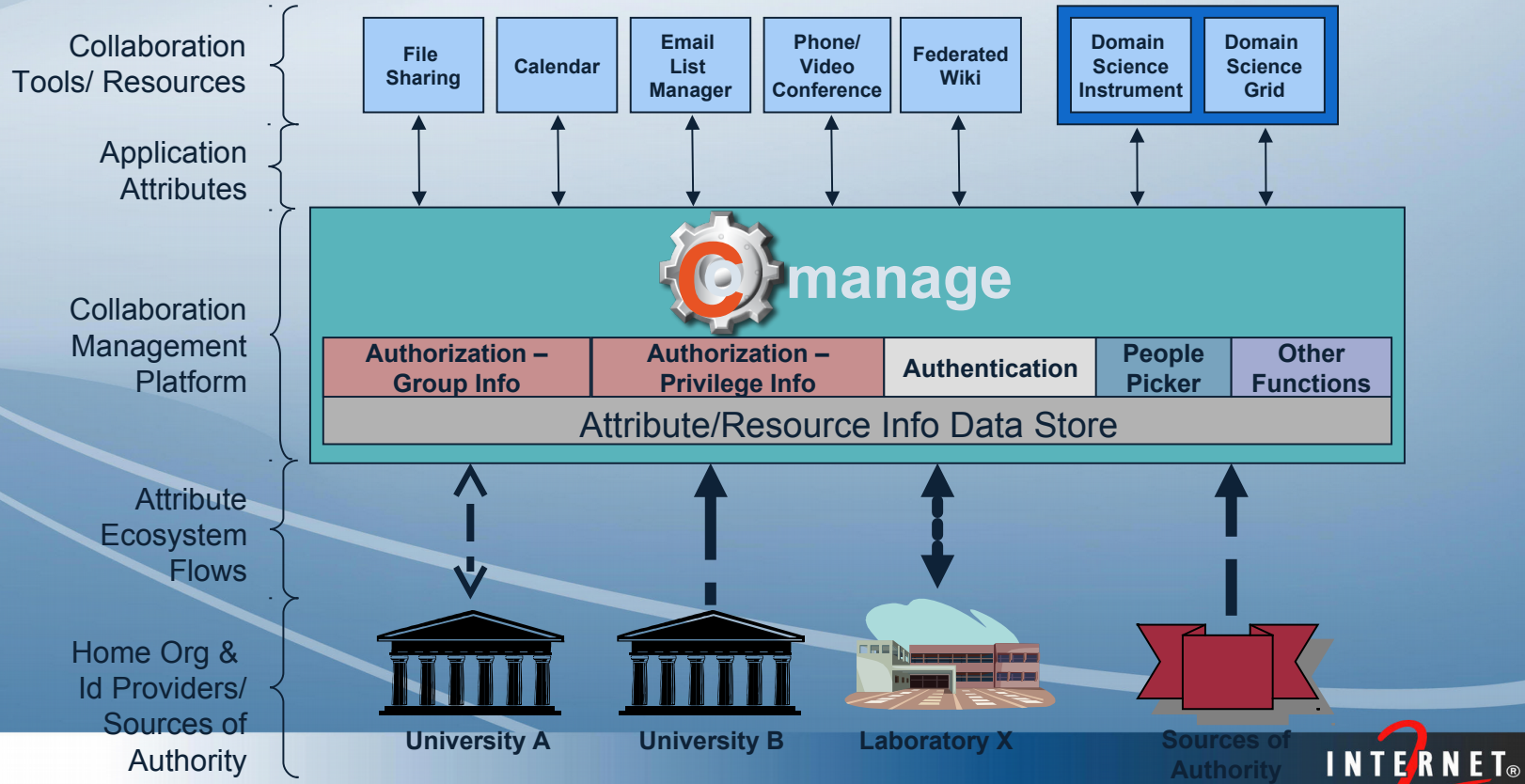
Comanageable applications

- Already done
  - Sympa, Federated wikis, Asterisk (open-source IP audioconferencing), Dim-Dim (open-source web meeting), Bedeworks (federated open-source calendar)
- Immediate targets
  - Rich access controlled wikis
  - Web-based file shares, IM, Google Apps for Education
- Domain science resources
  - Instruments
  - Grids

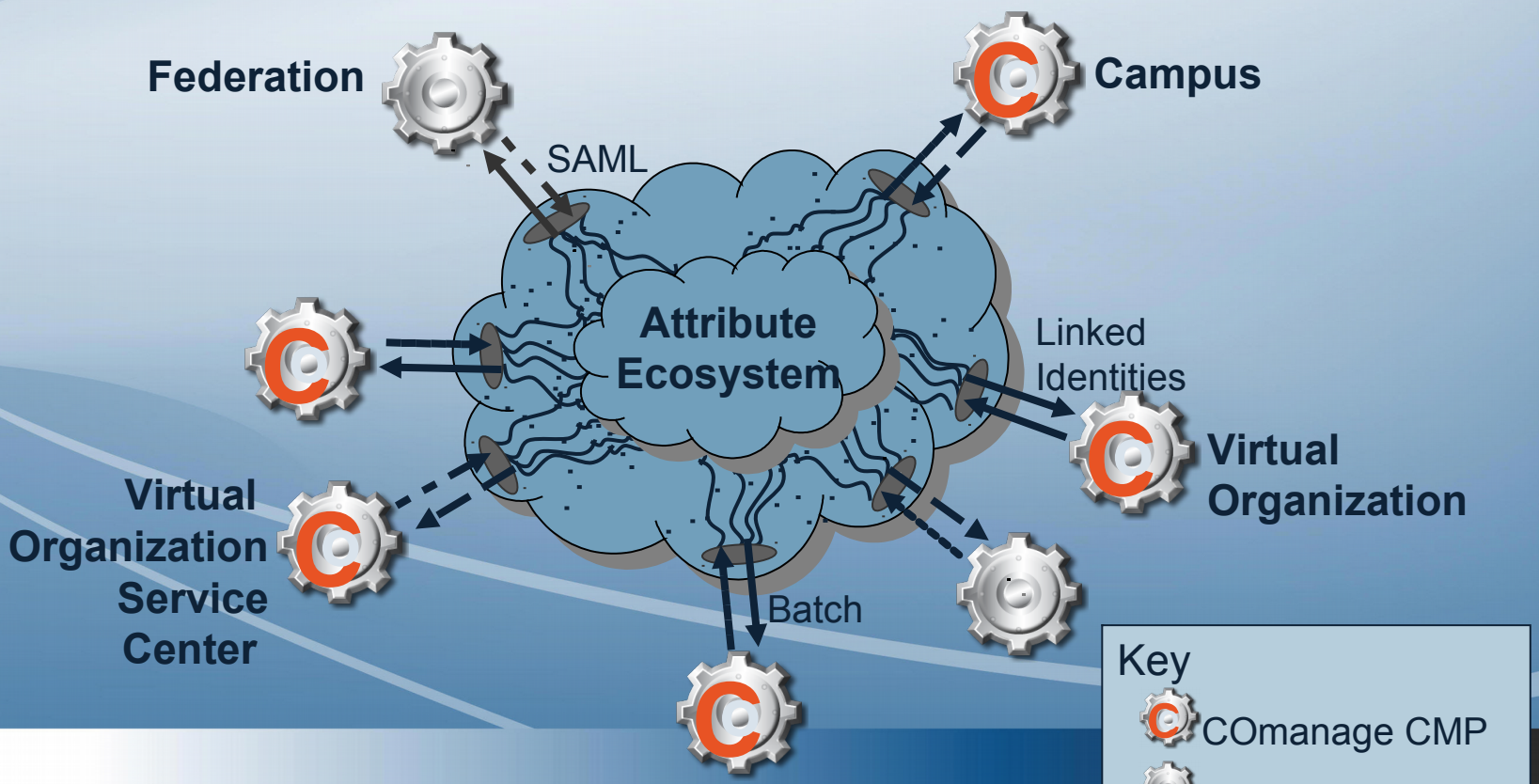
Some general COmanage comments

- A limited number of consoles present the basic identity services; can move directly between services as a standard workflow
- Early in the development; the GUI is particularly primitive
- Underlying store is an LDAP directory; alternatives include MySQL db, RTF store, etc.
- COmanage can be deployed by a campus, a department, a VO, a VO service center; COmanage instances communicate with each other by the “attribute ecosystem” voodoo

Collaboration Management Platform (CMP) and the Attribute Ecosystem



How Collaboration Management Platforms (CMP) Communicate



Discussion Topics

- Likeliness of other vertical sector federations
- Peering between a service sector vertical and R&E federations
- Federations and OpenId
- Government interactions and federations
- Sanctioned sources of authorities for attributes
- Other roles for federations
  - As trusted distributors of other metadata

Federation: Today and Tomorrow

March 5, 2008

PingIdentity™

Patrick Harding

CTO

Ping Identity

- Market Leader for **Secure Internet Single Sign-On**
- Founded in 2002
- Based in Denver, Colorado USA
- Privately held

PingIdentity™

Yesterday

- The Register – “OASIS Ratifies SAML” - 11/2002

> “SAML is an XML-based framework for web services, that allows the exchange of authentication and authorization information among business partners. It enables web-based security interoperability functions, such as single sign-on, across sites hosted by multiple companies”

“PKI has been dogged by issues of complexity, integration difficulties and user apathy”

http://www.theregister.co.uk/2002/11/07/oasis_ratifies_saml/

These Enterprises Are All Federating

So Are These Service Providers

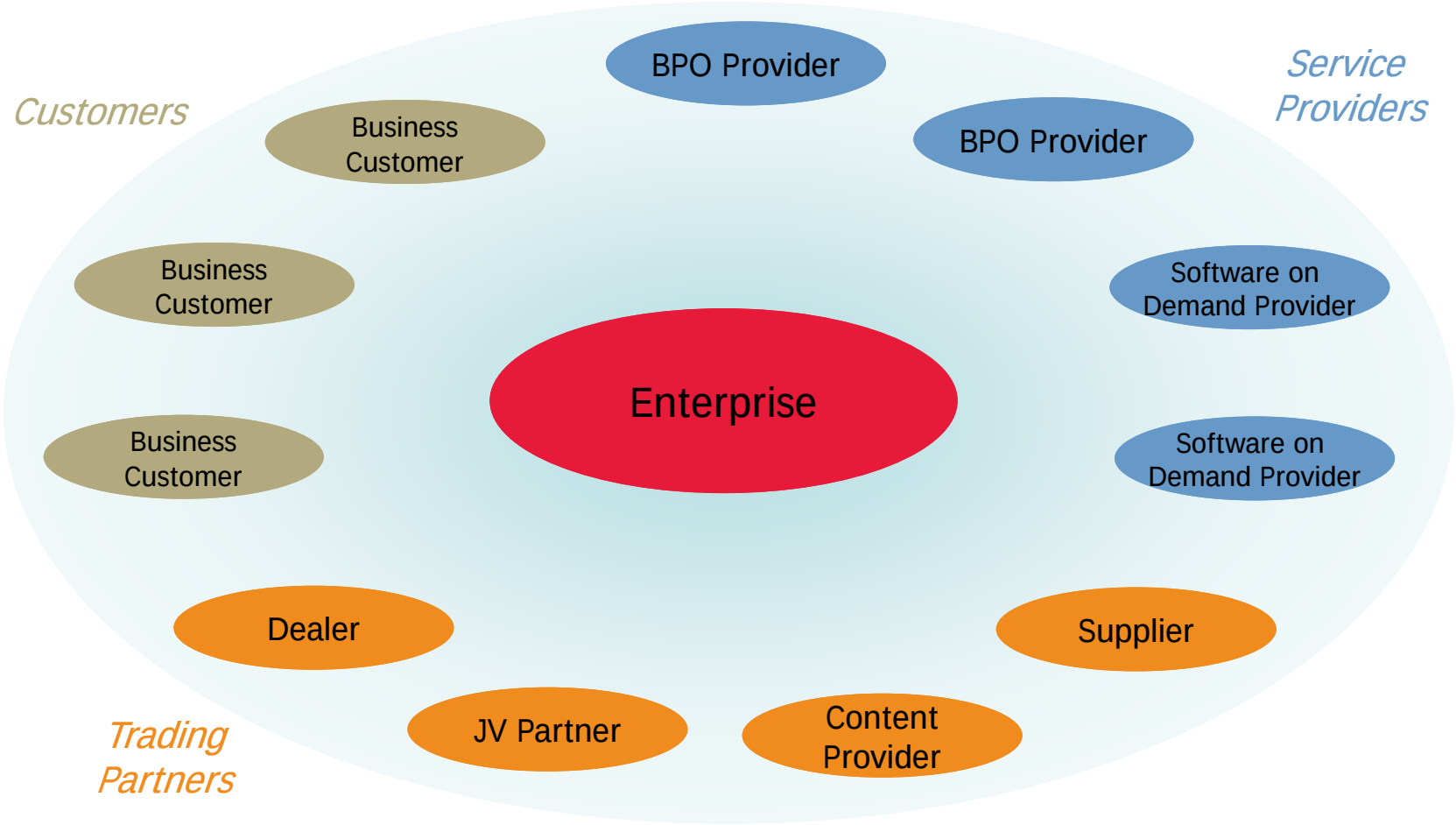
B2B Federation Today

- Protocol Debate is Over
- Organizations Have Enabled 5 – 10 Federation Connections
  - The Value of Federation Has Been Justified
- Common Business Scenarios Have Become Apparent

Common Use Cases

- Outbound SSO
  - for users to access software-as-a-service (SaaS) applications, business process outsourcing (BPO) services, and trading partners
- Inbound SSO
  - for relationships such as BPOs and managed services where external users access the enterprise's resources over the Internet
- Internal SSO
  - for the enterprise and its acquisitions, affiliates, subsidiaries and joint ventures
- SSO to third-party hosted industry hubs
  - for information sharing by users and application access among industry organizations

Today Federation is Enterprise-Centric



"High Leverage" Partner Drives Federation

PingIdentity™

The Federation Challenge

Identity Federation Can Takes 6–9 Months to Implement with Some Vendors

50 partners **x** 60 days per connection **=** Over 12 years

Does not scale!

"Federation has been dogged by issues of complexity, integration difficulties and user apathy"

Not Always a Business Issue

- Perception that lawyers must get involved
- Companies rely on existing business contracts to address:
  - Operational service level agreement disputes
  - Liability associated with protecting sensitive information
- Most Service Providers are actually happy to out source authentication to their customers and partners

Technical Friction

- Partners must negotiate which of many, many, many SAML options to use
  - Multiple profiles & bindings
  - Multiple identifier options
  - Flexible attributes
  - Authorization overloaded onto authentication
- Service Providers NOT leveraging SP-Initiated SSO
  - IdP Selection/Discovery is poorly defined
- Products require manual configuration of partner information
- Certificate Management is problematic
  - Trust established through manual exchange of certificates

Speed to Connect is Crucial

- Simplifying Federation Connectivity
- Publish Conventions and Best Practices
  - Industry convergence on POST/Redirect bindings
  - Optionally use email address/domain for IdP Discovery
- Automated meta-data exchange
- Dynamic domain-based trust
- Standard Attribute Schemas for B2B

What's Coming?

- Business Themes
  - Collaboration
  - Cloud Computing
  - Master Data Management
  - Increasing Internet Crime
  - SaaS & OnDemand Apps
- Federated Web Services
  - REST vs. SOAP
  - User Present (or Not)
  - oAuth vs. WS-* vs. ID-WSF
- User-Centric Use Cases?
  - OpenID
  - CardSpace
- High Assurance/PKI Environments
  - Identity Assurance Framework
  - Holder-of-Key SAML POST

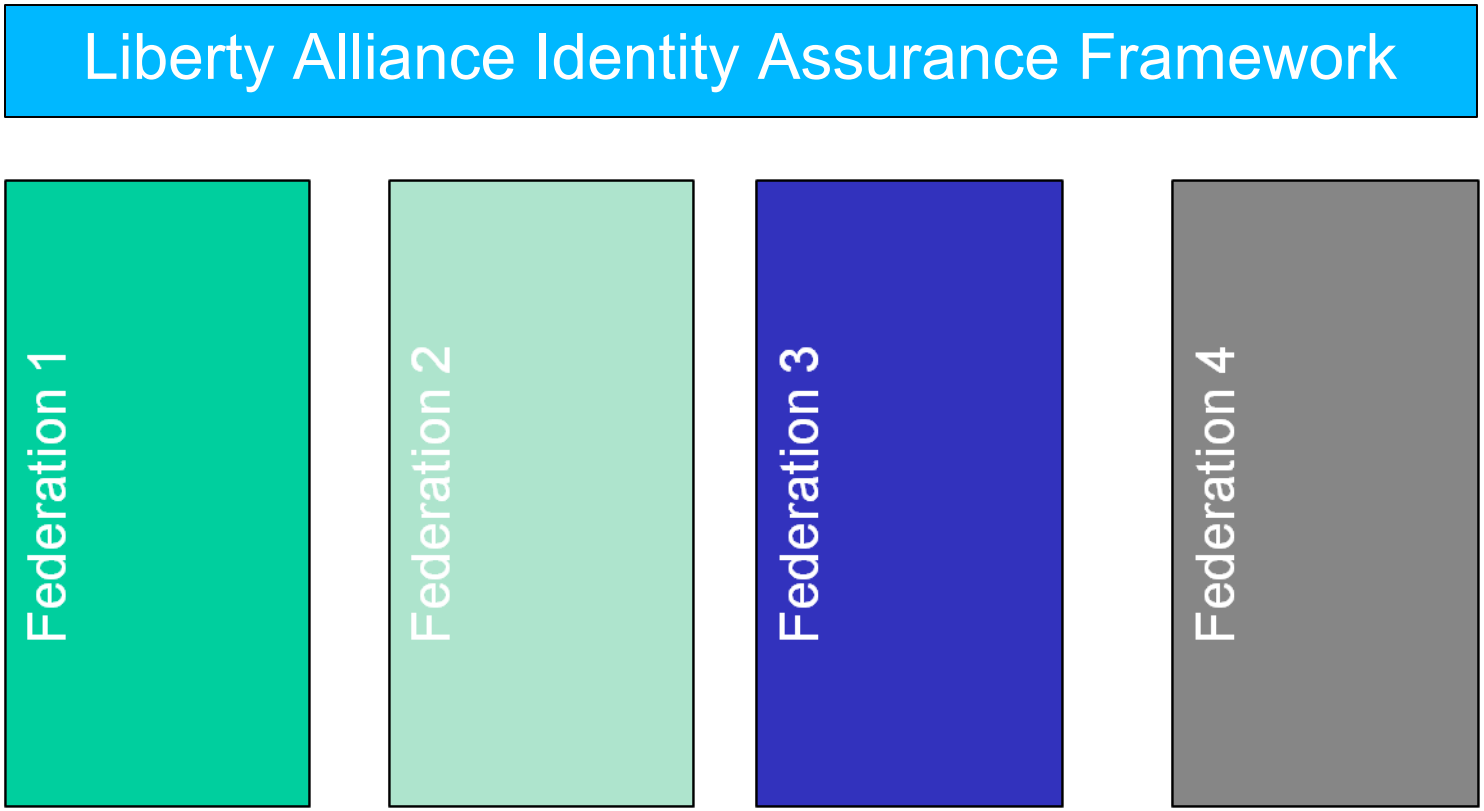
Identity Assurance Expert Group (IAEG)

- Goal: to foster adoption of identity assurance services, and uniformity and interoperability amongst identity service providers by promulgating overarching values
  - Public SIG exists alongside to solicit and encourage broad public participation

IAEG Key Activity: Identity Assurance Framework

- **GUIDANCE** (vs. policy) on what Federation members should consider and have policies for
  - 1.0 focused on IdPs/CSPs; Future phases to consider RPs & Federation Operators
  - Enable Federations to map their policies, practices and technologies to each other and ensure comparable trust levels
- **HARMONIZED, BEST-OF-BREED** industry identity assurance approaches
  - Re-use contributions from other sources (EAP, US E-Auth Federation, T-Scheme, TSCP, others)
  - Build upon accepted industry standards
- **NEUTRAL, COMPREHENSIVE & GLOBAL**
  - Cover all aspects of identity assurance: risk classification and assurance levels, service assessment criteria for organizations, credential management & identity proofing; business rules and certification models

Liberty IAF: The Picture



Upcoming Activities

- IAF in draft review
  - (http://www.projectliberty.org/liberty/content/download/3736/24651/file/liberty-identity-assurance-framework-v1.0.pdf)
  - Public webcasts:
    - **March 5, 8 am PT,** *Credential Management Service Assessment Criteria*
    - **March 12, 8 am PT,** *Identity Proofing Service Assessment Criteria*
    - **March 26, 8 am PT,** *Certification/Accreditation Business Rules*
  - Public input welcome: https://maa.projectliberty.org/id/idf-feedback.html

- Panel Session, RSA Conference, SF, April 9, 9:10 am
- Pre-Conference workshop, TowerGroup Conference, Boston, May 28
- Panel Session, Gartner Identity and Access Summit, London, June 23-25
- Panel Session, Burton Catalyst Conference, San Diego, June 23-25
- Others, TBD

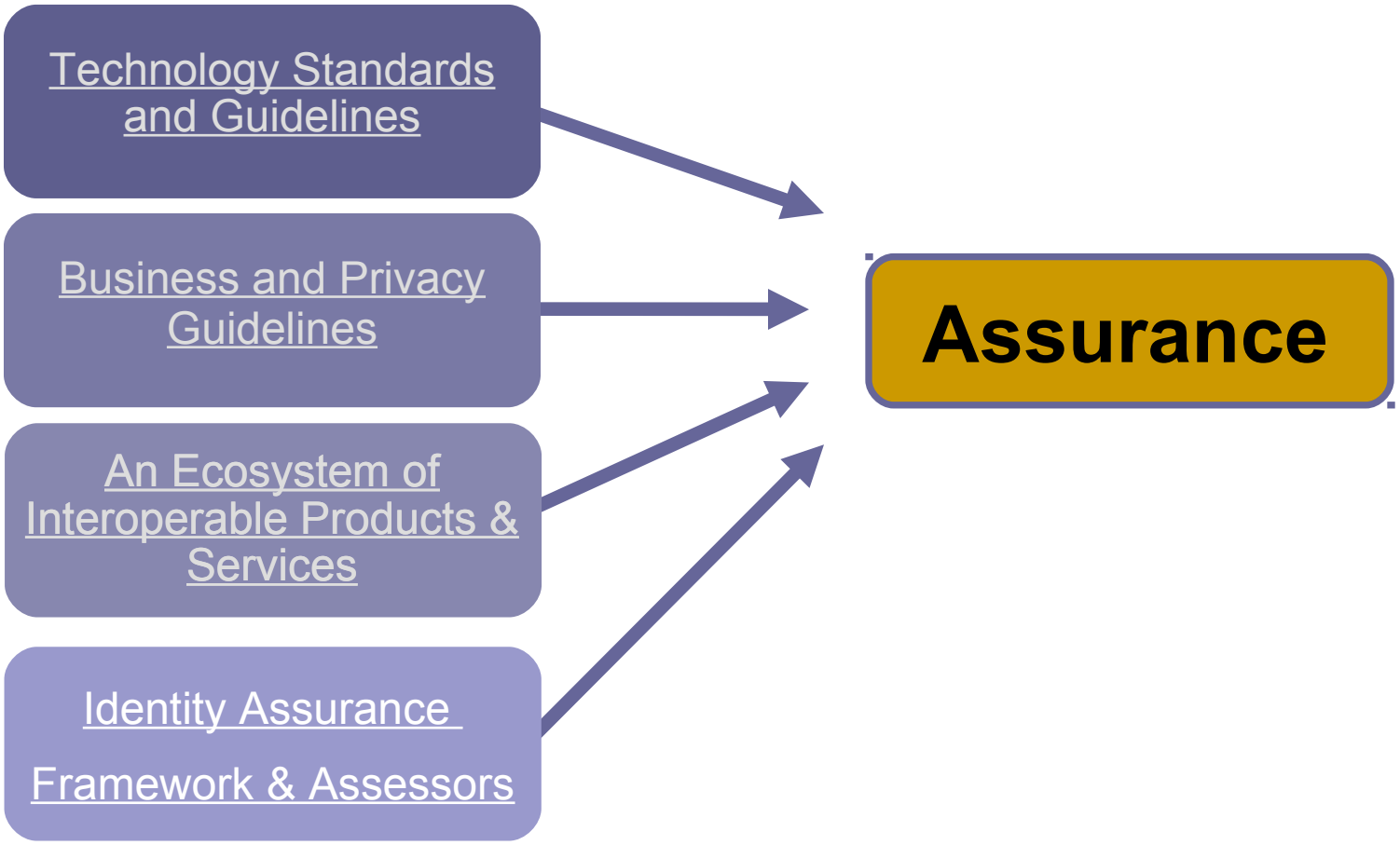
LIBERTY ALLIANCE PROJECT

Backup

Why was Liberty Alliance Formed?

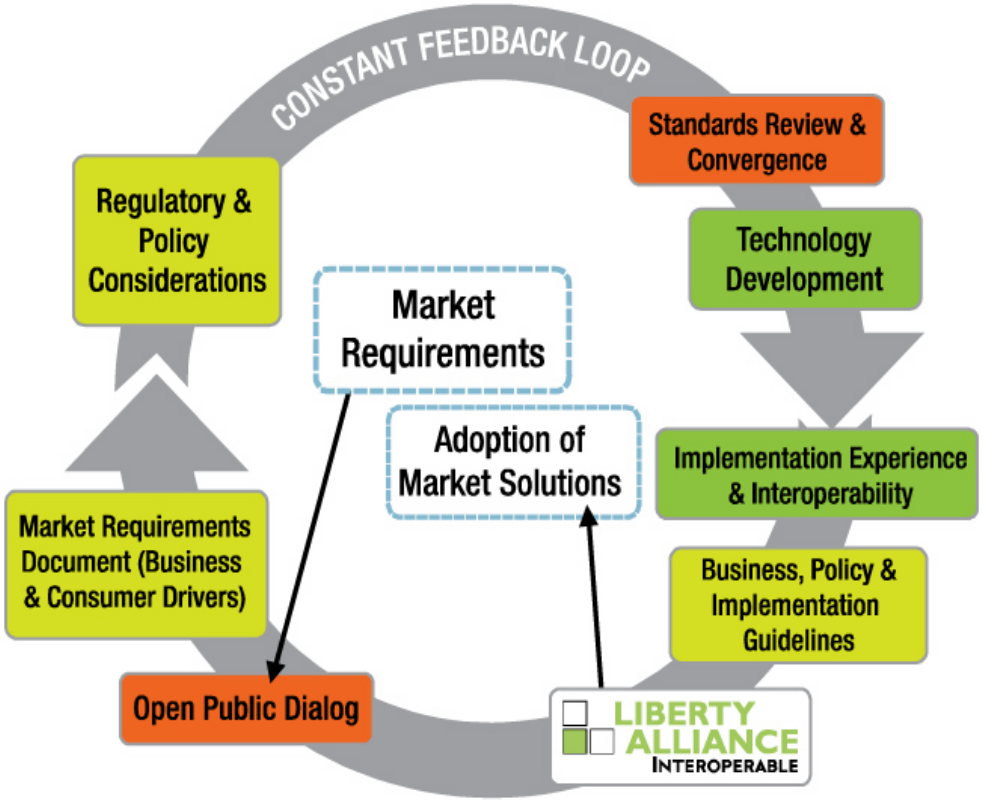
- Foster the ubiquitous, **interoperable**, privacy-respecting, identity layer (holistic identity management):
  - Liberty represents all constituencies toward this objective
    - (vendors, enterprise, government, consumers, universities, SME's, etc.)
  - Must be an open, collaborative system vs. single vendor strategy
  - Identity is important & complex. We must come together OR:
    - industry will become more fractured
    - governments will intervene
- Develop privacy-compliant practices to exchange identity information
- Develop standards-based model to …
  - Interoperate in heterogeneous environments
  - Avoid proprietary vendor lock-in
  - Provide flexible foundation for future growth
  - Scale to the WWW
- Deliver consumer & enterprise ***confidence that security, privacy and data integrity will be maintained***

More than Technology



The Liberty Process - Market Centric

- By …
  - Listening to the Market
  - Collaborating with other relevant groups
  - Documenting the requirements
  - Developing specifications and frameworks to meet the needs
  - Certify the products & assessors
  - Continuous evolution and improvement



Identity Assurance Roadmap

- Phase One of Certification Program for CSPs/IDPs, ratified in Identity Assurance Framework v1.0 FINAL (Q2 2008)
- Launch Accreditation Program to enable the Certification model and spur the market (Q2 2008)
- Scope and define Phases 2 and 3 for Federation Operators and Relying Parties (begins Q3 2008)

Getting Involved with Identity Assurance

- **Liberty Alliance Identity Assurance Expert Group (formal membership in Liberty Alliance is required)**

http://www.projectliberty.org/liberty/membership/become_a_member

- **Identity Assurance Special Interest Group (formal membership in Liberty Alliance is not required)**

http://wiki.projectliberty.org/index.php/IASIG

- **Identity Assurance Framework for Review and Comment**

http://www.projectliberty.org/liberty/content/download/3736/24651/file/liberty-identity-assurance-framework-v1.0.pdf

ID-Trust: Liberty Alliance Panel

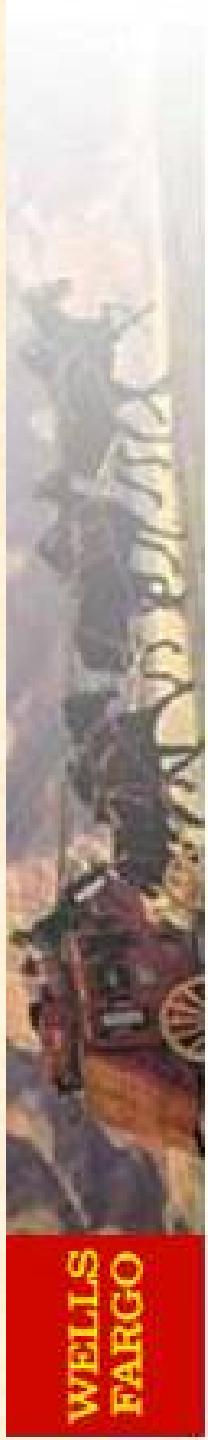
Advancing Common Levels of Trust

An Operational Perspective

- *Policy rules, agreements*
- *Vendor product support*
- *PKI and trust agreements*
- *Managing the trust agreements over time – an operations view*

Policy rules, agreements

- *Get aligned*
  - *Policy harmonization*
    - *Agree on definitions: NIST assurance levels, ANSI standards, etc. – extend / connect with enterprise policies*
  - *Extra-organizational policy*
    - *If possible, leverage high level terms and positions on implementation issues – subscribe, translate, and qualify*
- *Stay aligned*
  - *Rate of change*
    - *Understand the requirements of change*
    - *Plan for the change process*
  - *Backward compatibility*
    - *Recognize that all agents won't keep pace*
    - *Attempt to keep all agents subscribed*
      - *Risk level changes, etc.*
  - *Recourse and penalty*
    - *There will be non-compliant applications. Plan for it.*



Vendor product support

- *A better dialog required*
- *Requests for standards implementation*
- *Application certification (interoperation)*
- *Ask for instrumentation*
- *Assume the need for application meta-information and output to analytic products*

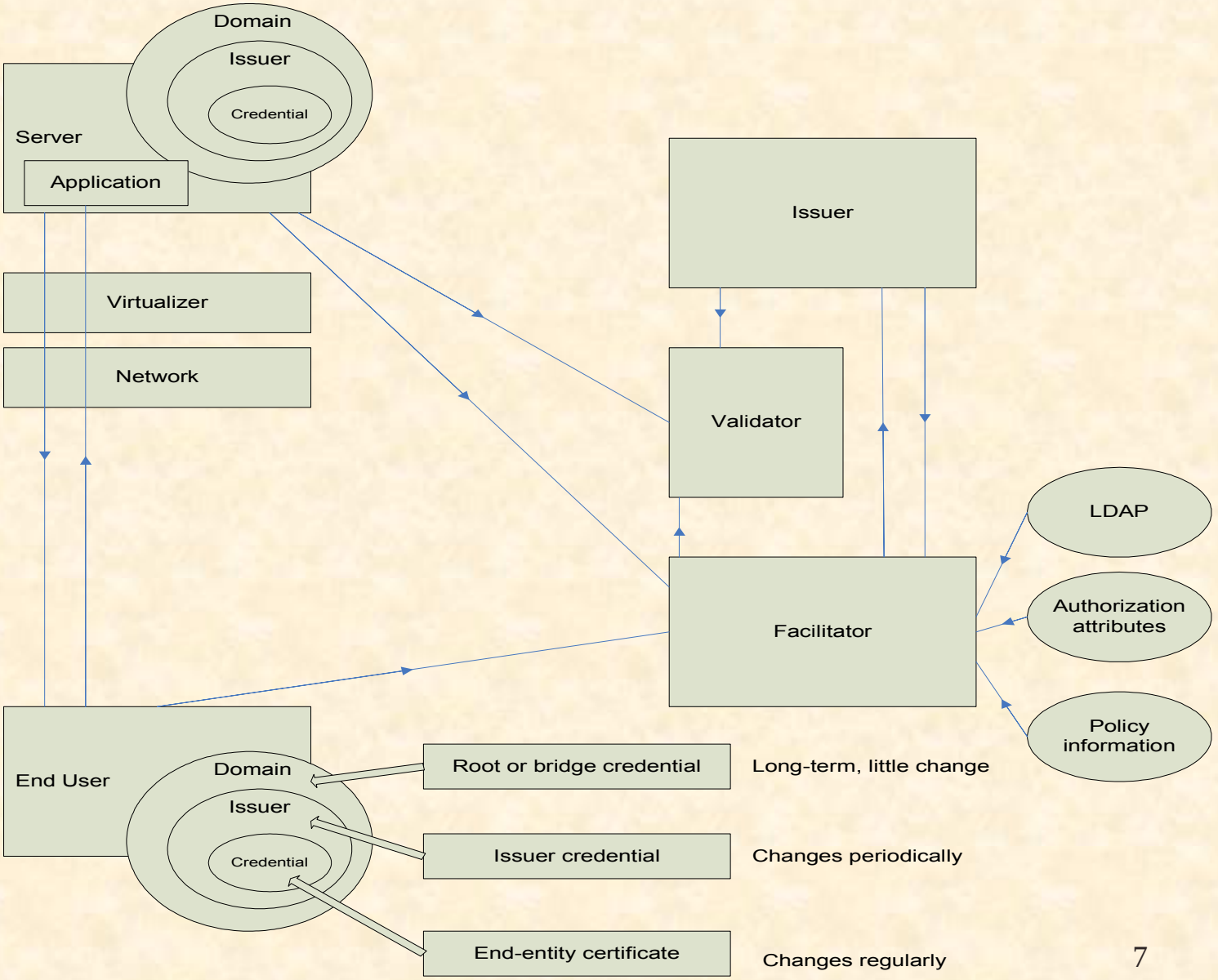
PKI and trust agreements

- *Ad-hoc, non-SAML, PKI "federations"*
  - *Extra-enterprise trust domains?*
    - *Defined by the "relying-application"*
      - *Either active or passive*
  - *Application level domains?*
    - *Defined as enterprise or extra-prise*
- *Industry [pki] Trust-Authorities*
  - *Default [pki] trust-stores (appropriate?)*
  - *Industry Trust structures: benefits?*

An Industry Trust Authority – Who does it help? How?

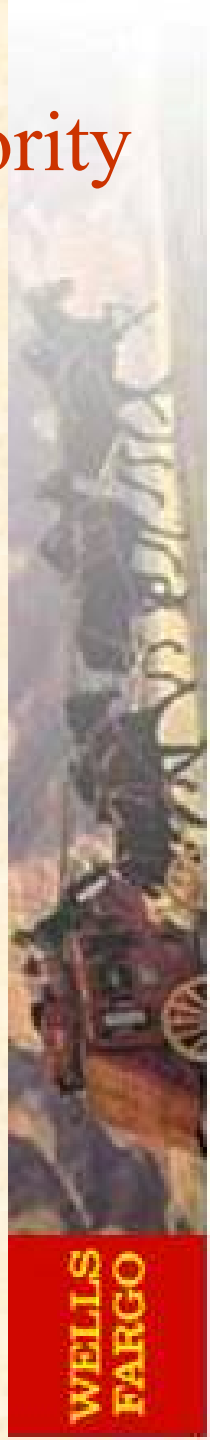
- Technology, Business, or both?
- Business Process Enhancement?
- Information Security Improvement?
- FI Compliance Policy Improvement?
- Customer Experience Improvement?

Managing the trust agreements over time – an operations view



A Financial Industry Bridge Certification Authority

- Support growing digital certificate use in Financial Institutions
- **(Technology)**
  - PKI has moved from a "star" technology to a role as second-clarinet in the technology orchestra
  - Digital certificates in heavier use as internet utilities become e-transportation network
- Enable businesses to snap together new connections: less cost, new products faster
- **(Business Processes)**
  - Using best / favorite certified digital certificate provider instead of negotiating and building connections for each project / service
- Secure connections with strong, standard security components and protocols **(Information Security enhancing)**
  - Using FI policy approved digital certificate issuers, FIs reduce risk of trusting the un-trustworthy
- Maintain connections with a strong policy framework across institutions **(Policies and Compliance)**
  - The non-technical… a policy / trust authority board to decide which issuer certificate authorities conform to FI standards and requirements
- Interact with customers using strong, financial-Institution issued credentials **(Customer experience)**
  - Defining a trusted set of issuers with common issuance policies, regularly certified for use outside of a single institution.



Conclusions

1. *Identify good policy – subscribe and maintain*
2. *Work with vendors to deliver robust standards based products*
3. *Observe to formation of informal, un-defined "federations" [not necessarily SAML]*
4. *Bridge PKI and federation are not exactly the same thing*
5. *A federation could provide a bridge service as part of a federation*
6. *Trust agreements are not one-time static events*

Multi-Domain Identity Management

Deployment Interoperability, Validation and Certification

7th Symposium on Identity and Trust on the Internet (IDTrust 2008)

Lena Kannappan,
Founder & CEO, FuGen Solutions, Inc.

Connecting Identities Everywhere.™

March 4-6, 2008

FuGen Solutions

FuGen Solutions

FuGen Solutions enables successful Multi-domain Identity Management Deployments

FuGens' MISP™ Services include:

- Managed Identity Interoperability and Compliance Verification
- Federation Partner on-boarding
- Federation Partner Certification program implementations
- On-going real-time monitoring and reporting

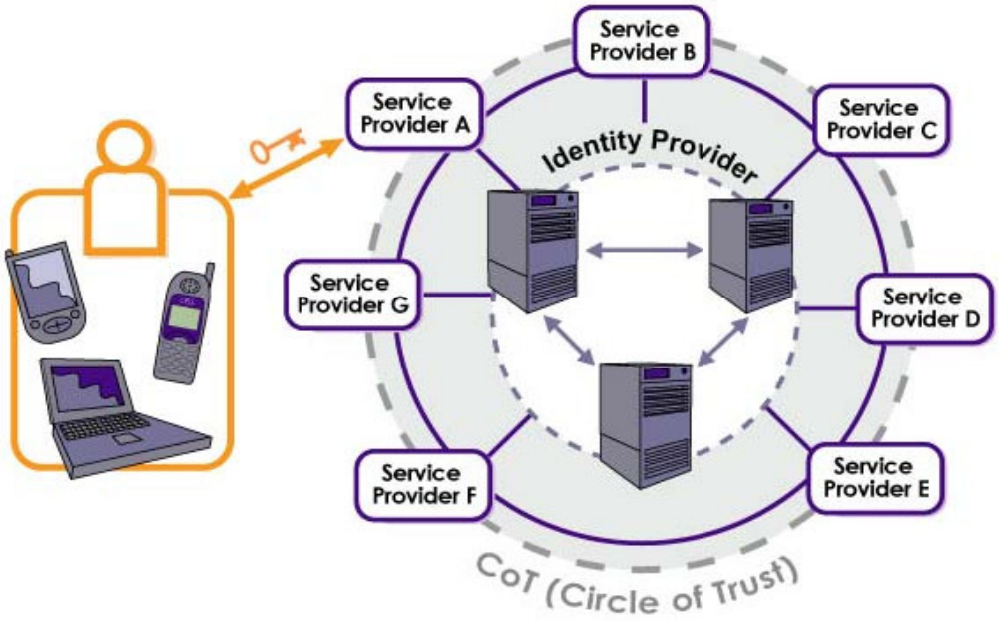
- Founded in February 2006, privately held, HQ'd in Sunnyvale, CA
- Customers include Financial Institutions and Government Agencies

"The path to simplicity in Identity Management is not so easily bought. Outside the lab environment, enterprises have a staggering and complex array of products intersecting with ever-changing demands."
The Burton Group

Connecting Identities Everywhere.™

FuGen Solutions

Federated World



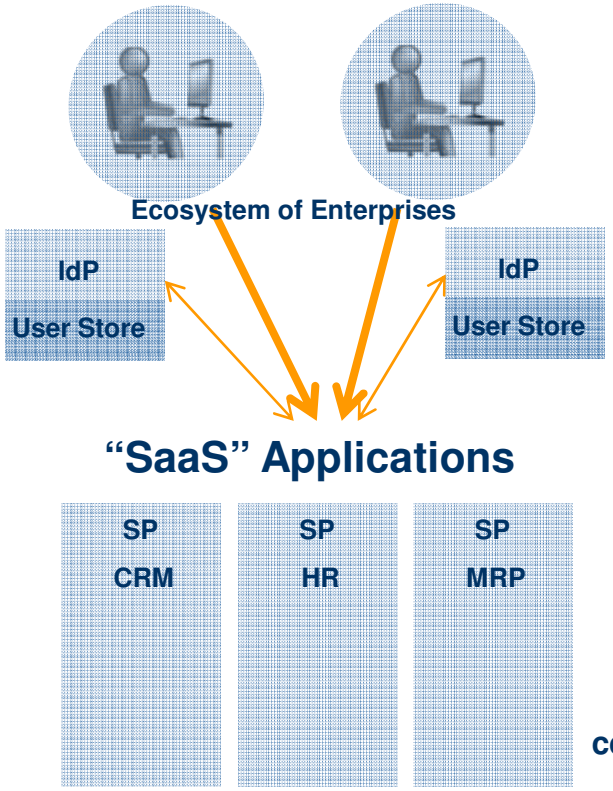
- **Federation Models & Communities**
  1. Corporate Federation
     1. Internal/External
  2. Identity Federation (COT)
  3. Internal/Intranet Federation
  4. IDP to IDP Roaming/Proxying
  5. Federated Communities
  6. Identity Brokering models
- **Standards**
  - SAML 1.1, SAML 2.0
  - Shibboleth
  - Liberty Alliance
  - WS-Federation
  - WS-Trust, WS-Policy (WS-*)

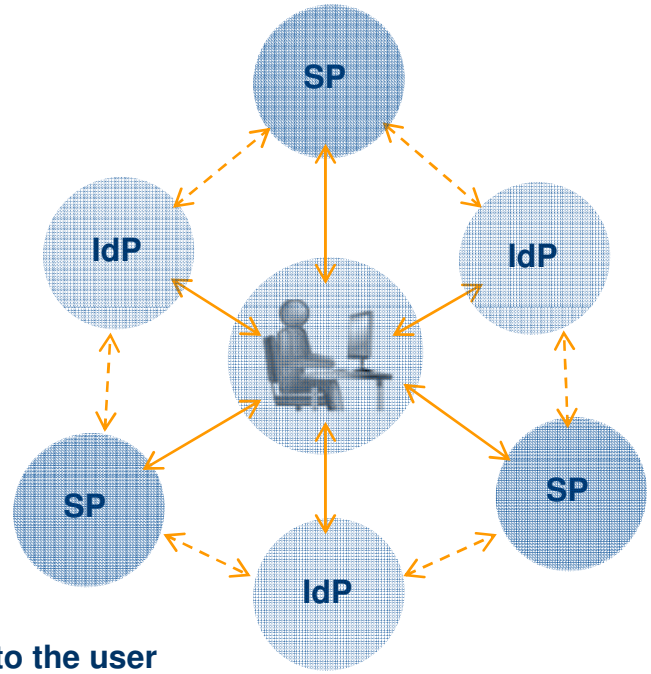
- IDP or IP Centric Customers
- SP Centric models or Relying Parties
- Pairwise IPs and RPs
- Distributed Identity issuers

Connecting Identities Everywhere.™

FuGen Solutions

Identity Management: Model of the Future





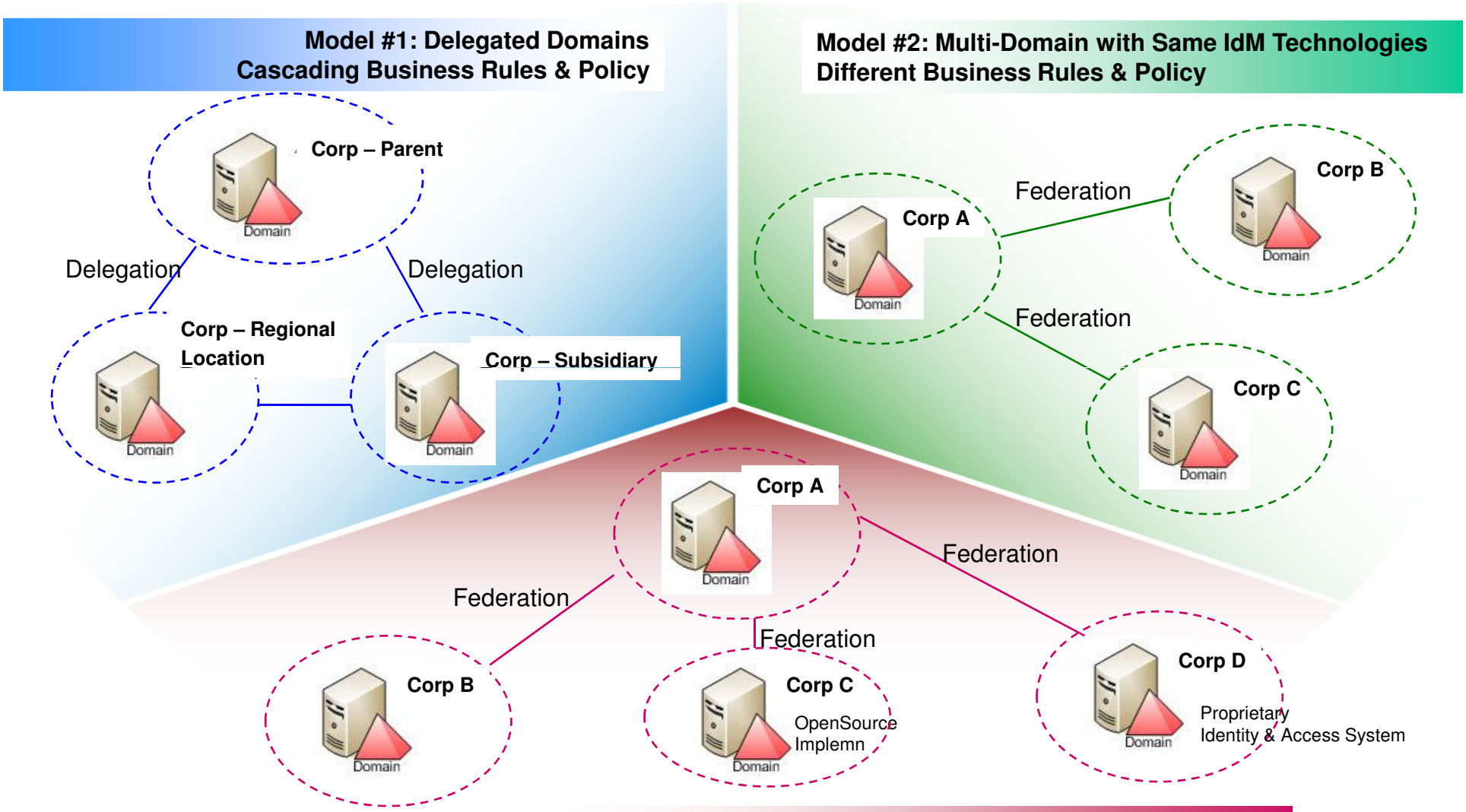
Enterprises will need to adapt to the user centric model as enterprise users will require access to consumer services

Who to trust and how?
How to manage the identity "network"?

Connecting Identities Everywhere.™

FuGen Solutions

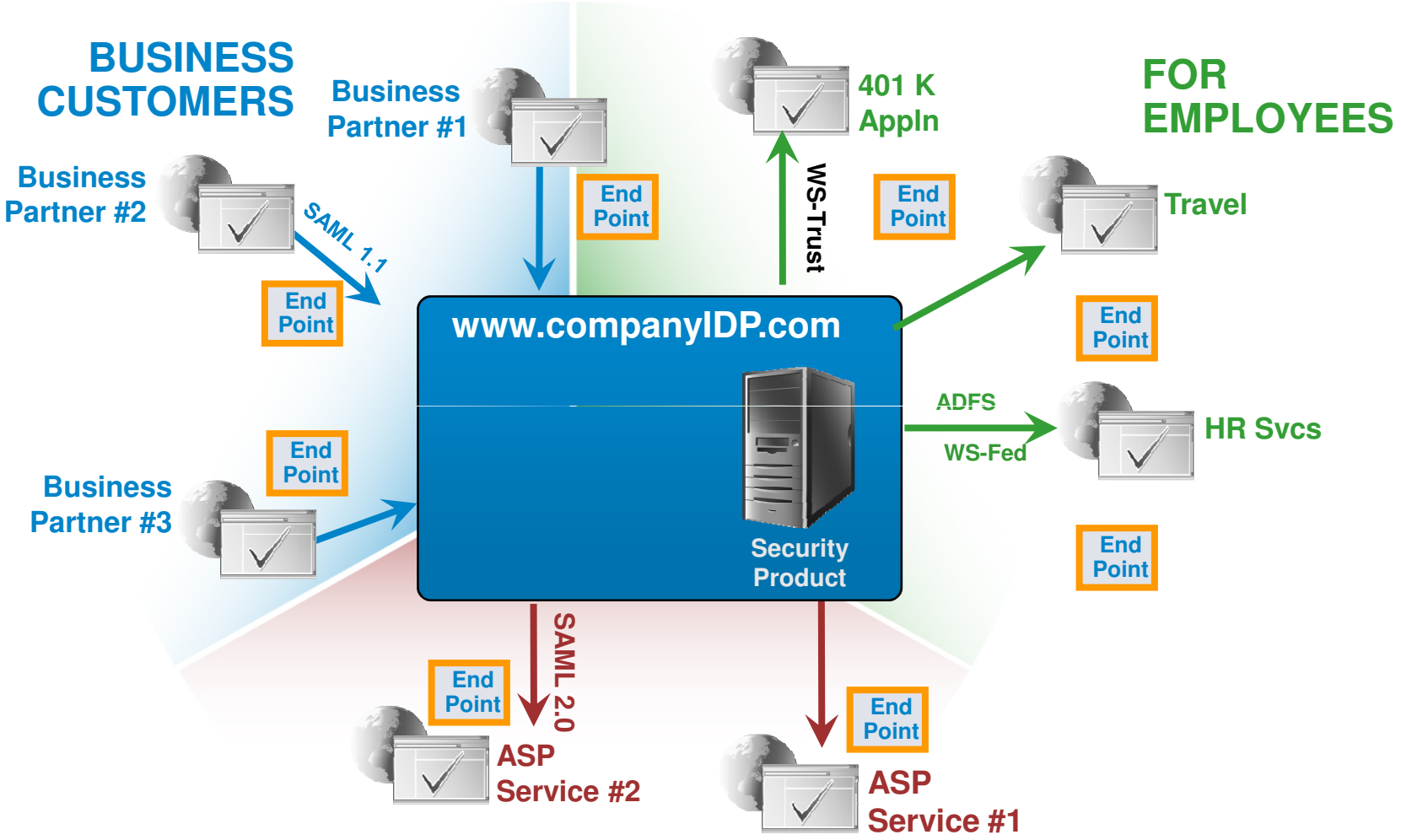
Enterprise Identity Management Deployment Models



Connecting Identities Everywhere.™

FuGen Solutions

Corporate Federation COT
- External Service Providers

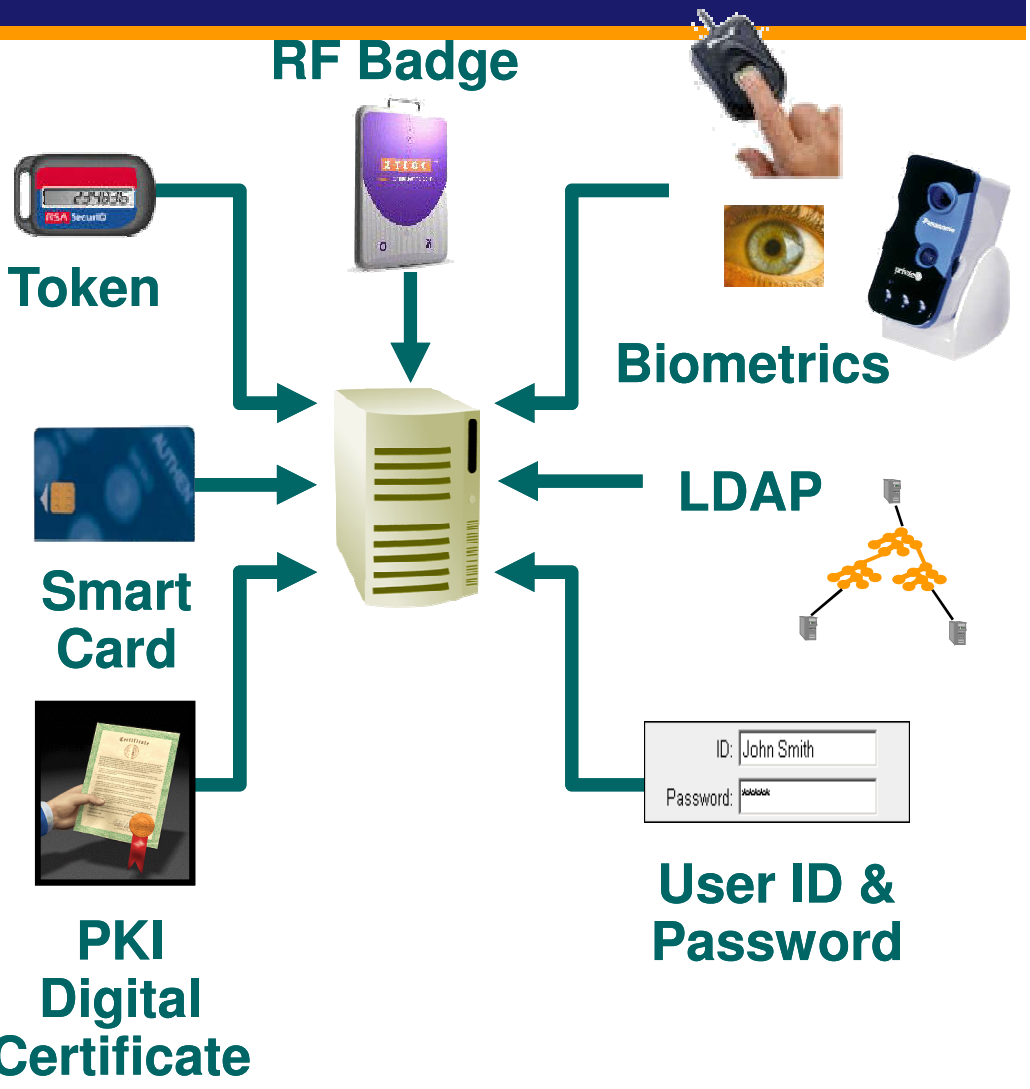


Connecting Identities Everywhere.™

FuGen Solutions

Primary Authentication Methods

- **Automated Login**
  - Active Directory, NT
- **User Id/Password**
  - Native SSO
  - LDAP
- **Token**
  - RSA SecurID
- **PKI Digital Certificates**
  - Smart Cards, USB Keys
- **Kerberos**
- **Strong Authentication**
  - 2 Factor Auth
  - Biometrics



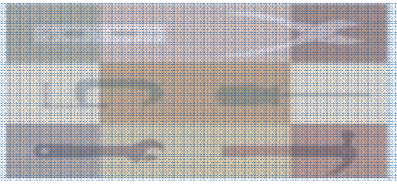
Connecting Identities Everywhere.™

FuGen Solutions

Multi-Domain IDM Deployment Interoperability

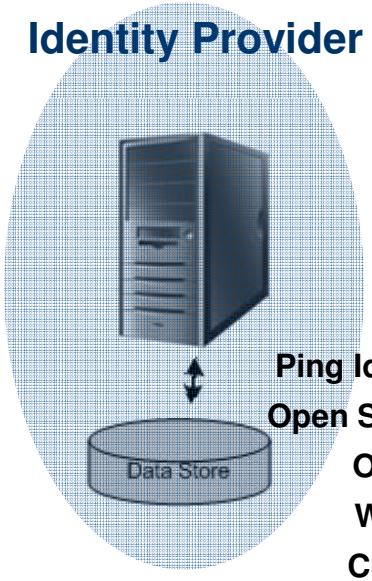
Federated Identity Management

Business Criteria
Standards
Technology requirements
Security and Trust needs



Regulatory and Compliance
Governance and Policy aspects
Privacy implications
Legal requirements

Identity Provider

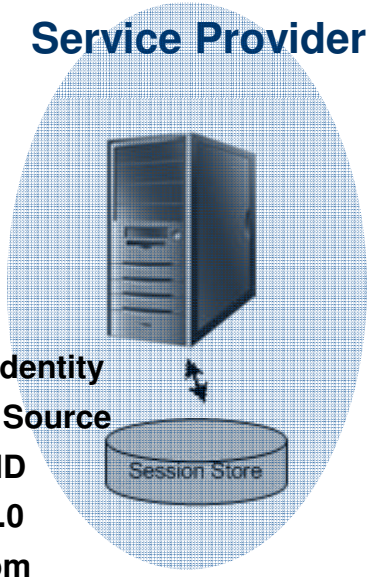


CA
IBM
MSFT
Sun
RSA
Ping Identity
Open Source
OpenID
Web2.0
Custom
Others

OASIS SAML 1.1 / 2.0
WS-* (Security, Trust, Federation, …)
Liberty Alliance ID-WSF
OpenID, Higgins, OAuth, XRI/XDI

Service Provider

CA
IBM
MSFT
Sun
RSA
Ping Identity
Open Source
OpenID
Web2.0
Custom
Others



Client Identity Technology:
CardSpace, OSIS, Higgins

Connecting Identities Everywhere.™

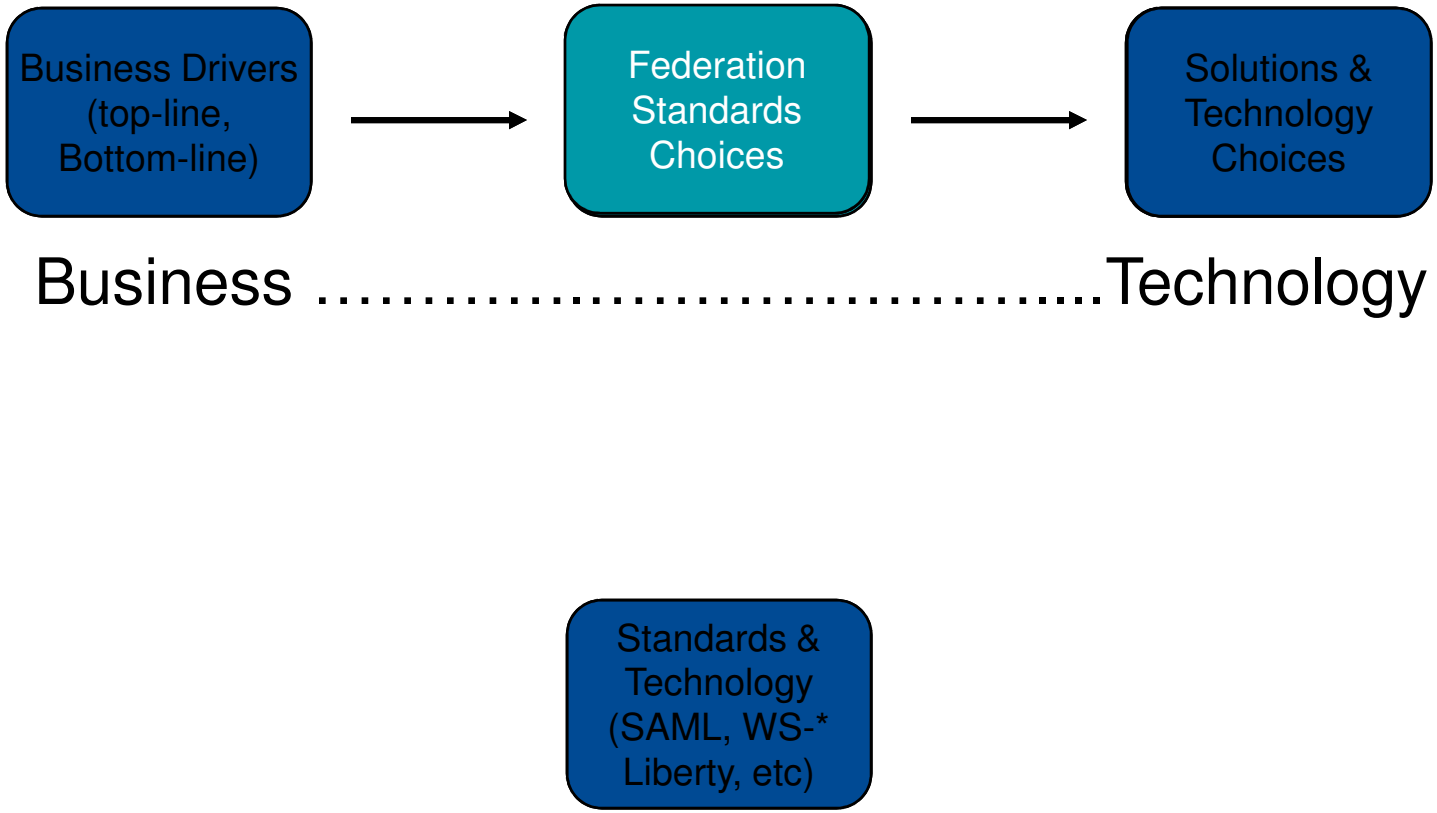
FuGen Solutions

Current Federation Architecture Deployment Issues

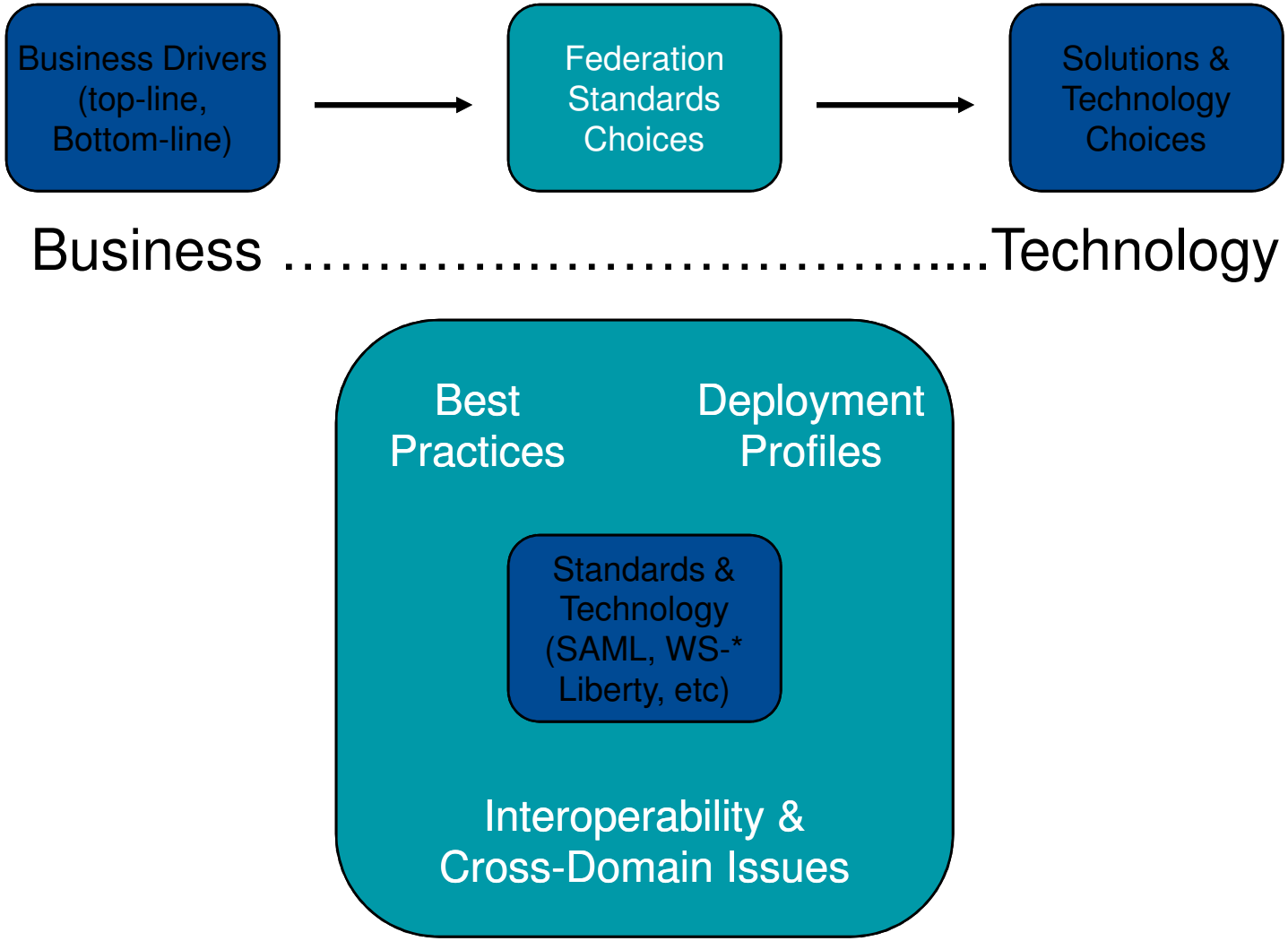
- Fast emerging technology/need; Companies moving towards Federation
- Need of federation not understood due to lack of knowledge as well as lack of resources to deploy; Leads to poor or failed implementation
- Deployment Interoperability and Verification; Standards compliance
- Need to minimize risks/guesswork
- Deployment profiles
- Multi-protocols, Several Features / Sub-Features
- Partner Disparate infrastructure
- Trust models
- Policy and Legal aspects
- Various Authentication mechanisms, Encryption methods
- Types of Assertions
- Data integration issues

Identity Mapping, Attribute Mapping, Meta Data Exchange and Policy mapping issues

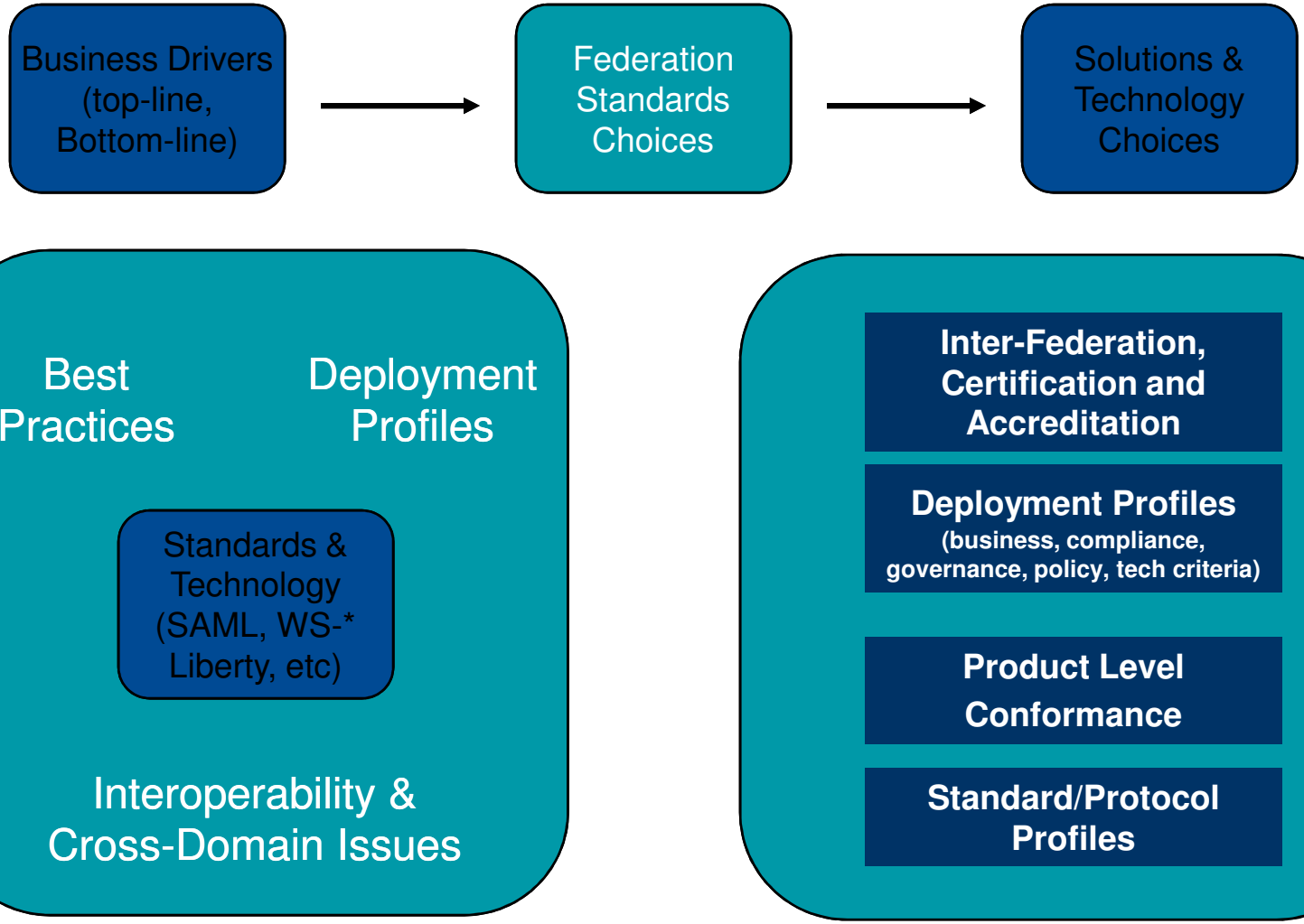
Deployment Interoperability challenges and Need beyond Product level certifications



Deployment Interoperability challenges and Need beyond Product level certifications



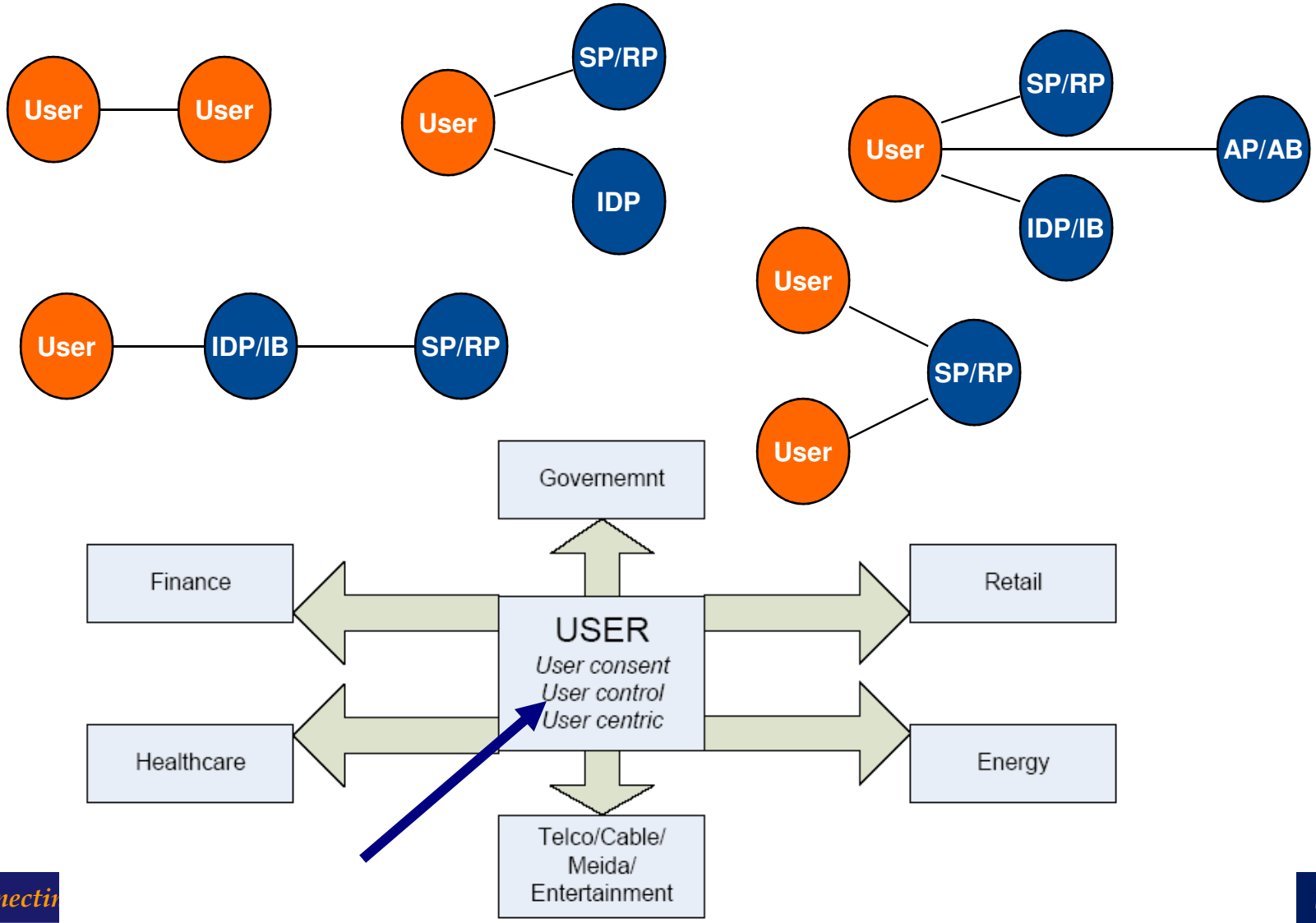
Deployment Interoperability challenges and Need beyond Product level certifications



Connecting Identities Everywhere.™

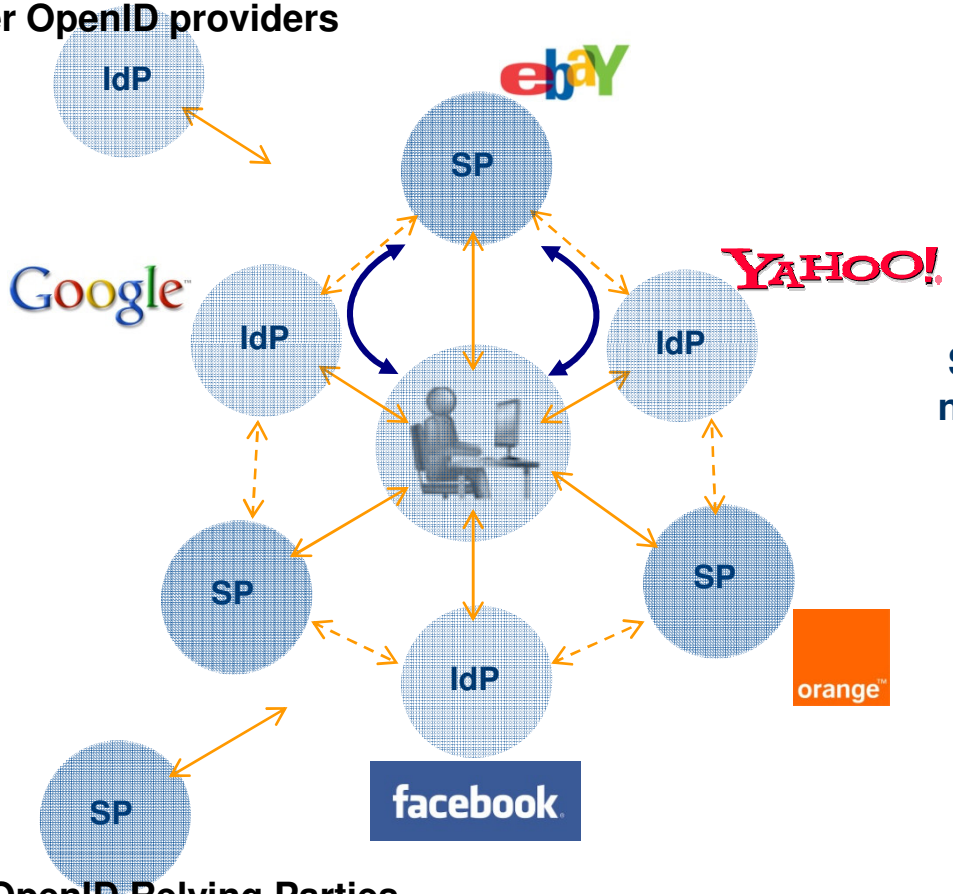
FuGen Solutions

User Centrism models in Federation



Identity Management Challenges for Consumer Service Providers

Other OpenID providers



Web 2.0 users demand ease of use and choice of identity provider

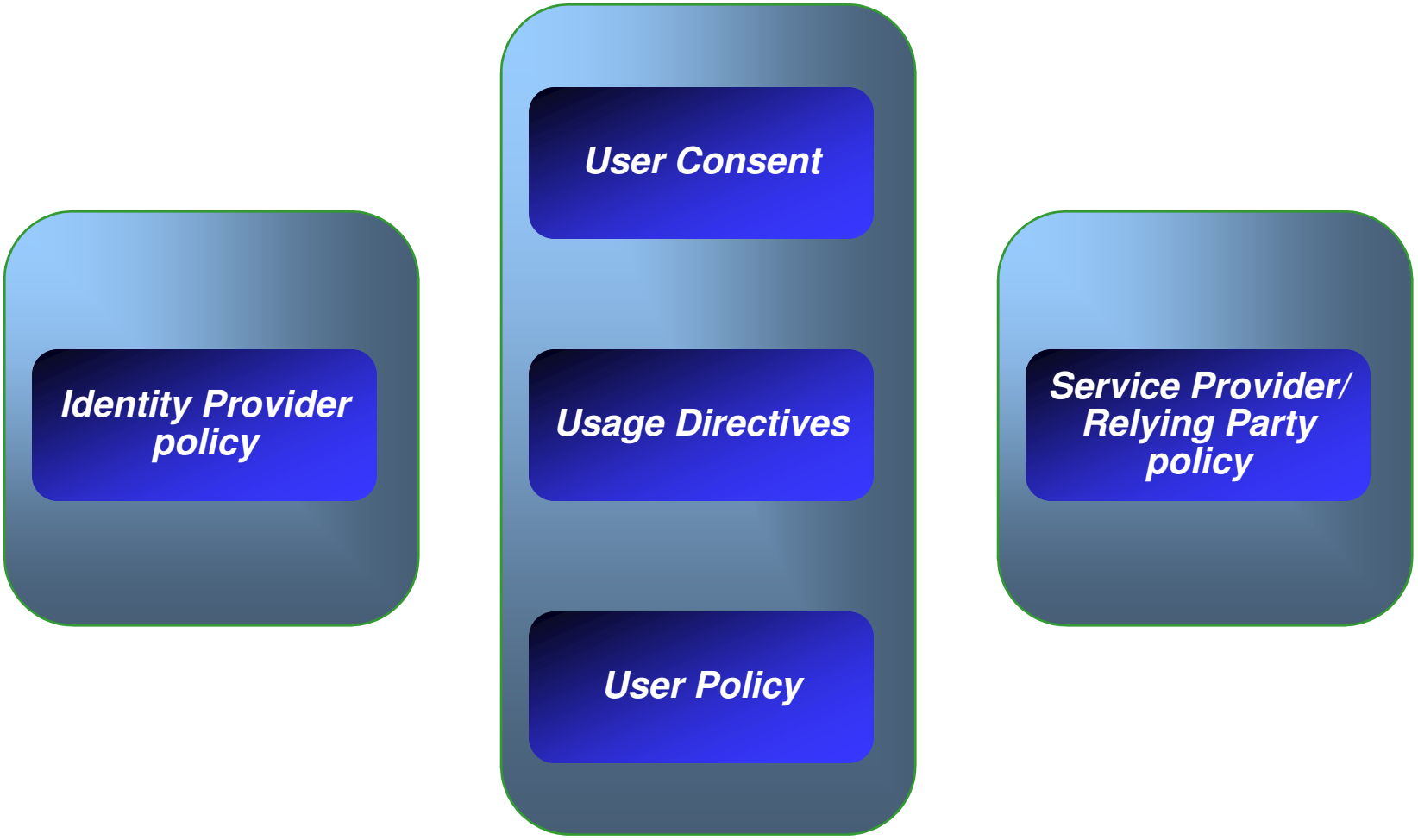
Service Providers need to develop an ecosystem of trusted IdPs

Other OpenID Relying Parties

Connecting Identities Everywhere.™

FuGen Solutions

User Privacy model for Providers



Connecting Identities Everywhere.™

FuGen Solutions

Next Generation thinking to User profiles and Privacy framework

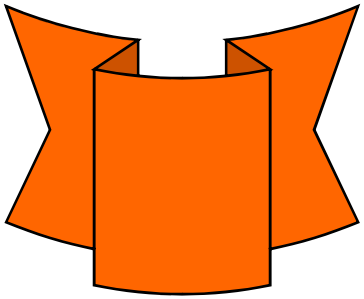
Government profile

Converged Services Profile

Google, Yahoo .. profile

Healthcare profile

User Centrism



Federation

Employer/ Enterprise profile

Blogs, Wikis profile

- Layered user profiles
  - Multiple personas

Connecting Identities Everywhere.™

FuGen Solutions

Closing Thoughts

- User Centrism and Federation and NOT User Centrism Vs Federation
- Strong Authentication will play a significant role in Federation
- Standards and industry efforts still need to mature – cohesive approach is important
- SAML, Web Services and Light weight technologies will co-exist
- Layered profiles for the converged, ubiquitous and federated world; New generation privacy framework will emerge while embracing current successful models
- Identity Assurance Services – Inter Federation Requirements – Assessment and Accreditation of Federations
- Need to tackle Identity Architecture Deployment Interoperability challenges to increase adoption !

Connecting Identities Everywhere.™

FuGen Solutions

Questions

Connecting Identities Everywhere.™

FuGen Solutions

Liberty Alliance Identity Assurance Framework

from a practical point of view

... in a Danish context

Jan Riis
jri @ lakeside.dk

A little History

- Danish Healthcare has been working 3 years with Identity Based Web Services
- 2005 MedCom and Danish Regions
  - ”Competed” for the first standard/profile
- No governance towards standardization:
  - No Authentication levels defined
  - No high level architecture for WS communication
  - No criteria for assuring trust of key WSP’s

Consequences

- Parties started out with 6 levels of ”authenticity”
  - Some based on PKI
  - Some based on username/pwd
  - Some levels for ”delegated trust” (systems vouching for user authenticity)
  - Some levels target cross-cutting security properties (non-repudiation of messages etc.)

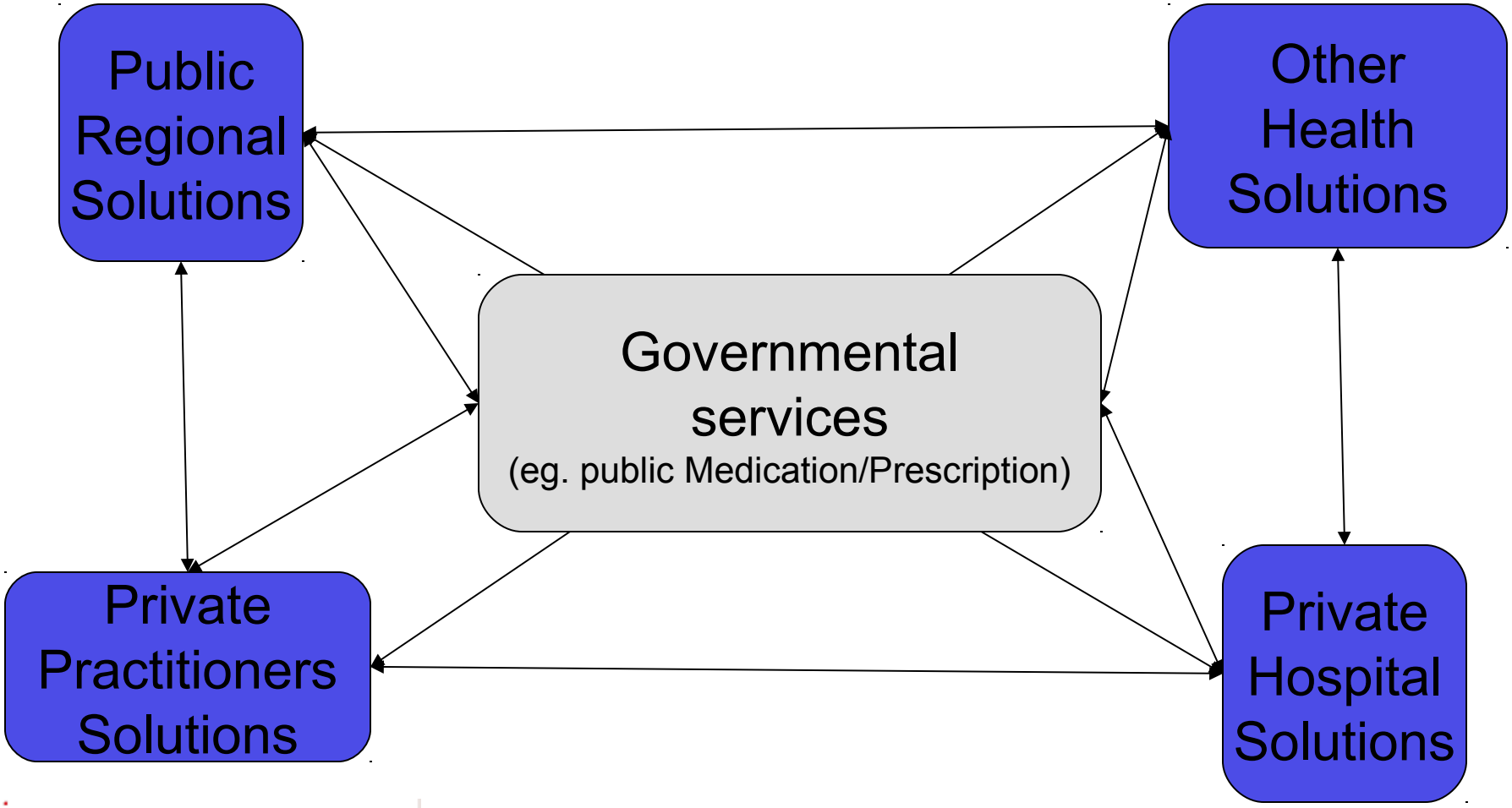
There is a need for IAF!

- ITST standardized authentication levels in 2006 for all public systems
- Directly referred to NIST work
- 2007 Health sector standards were aligned with national guidelines
- Without the national/international standards, this would not have happened!

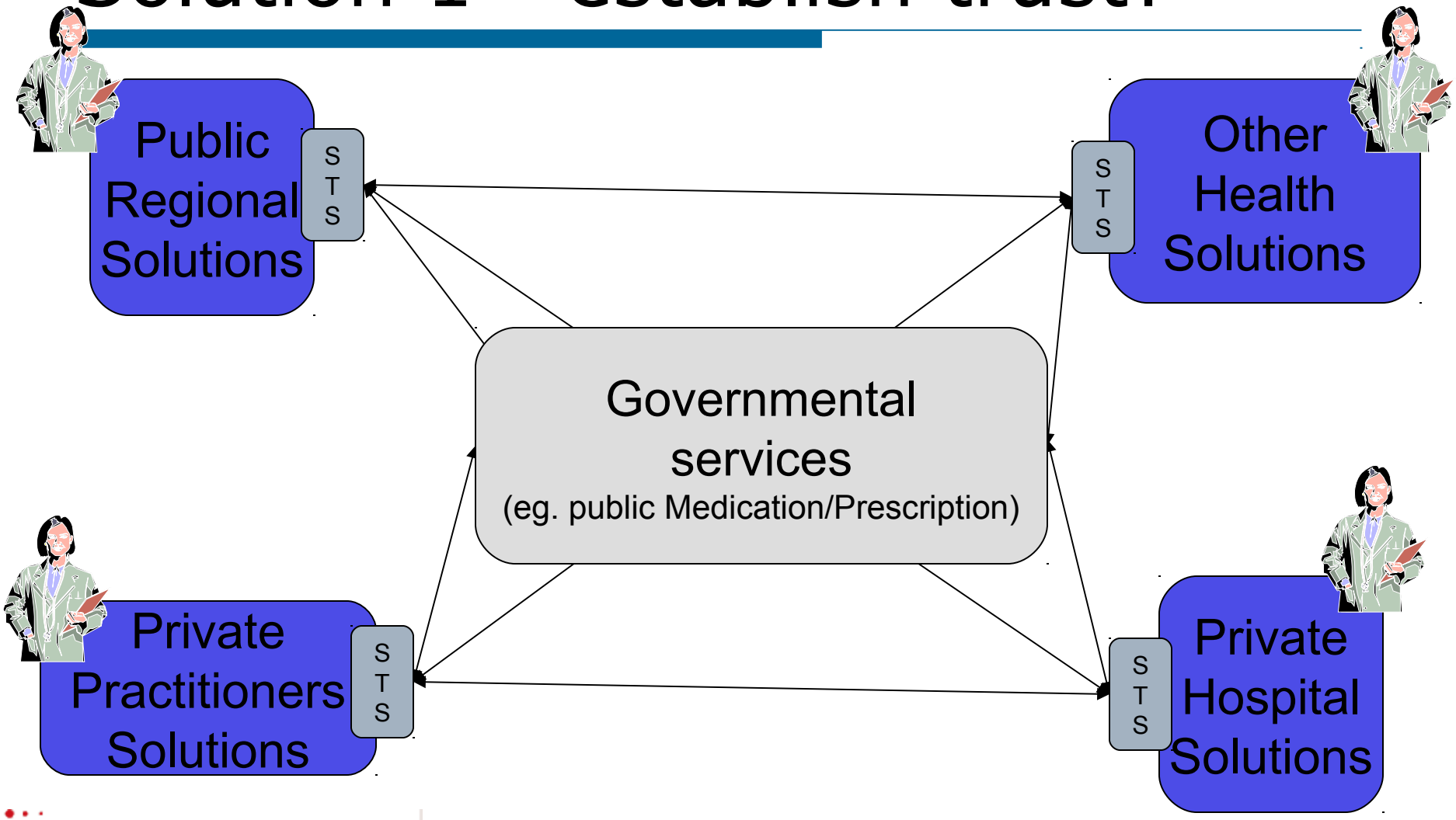
Trust relationships?

- NIST Authentication levels does not relate directly to “trust”
- So how will the concept of “trust” be used in Danish Health Care?
- Enter: “Digital Health Denmark”
  - Aims at increasing treatment quality by “enabling” access to all relevant information

A few years from now?



Solution 1 - establish trust?



Solution 2 - National ESB+PKI?

Public Regional Solutions

Other Health Solutions

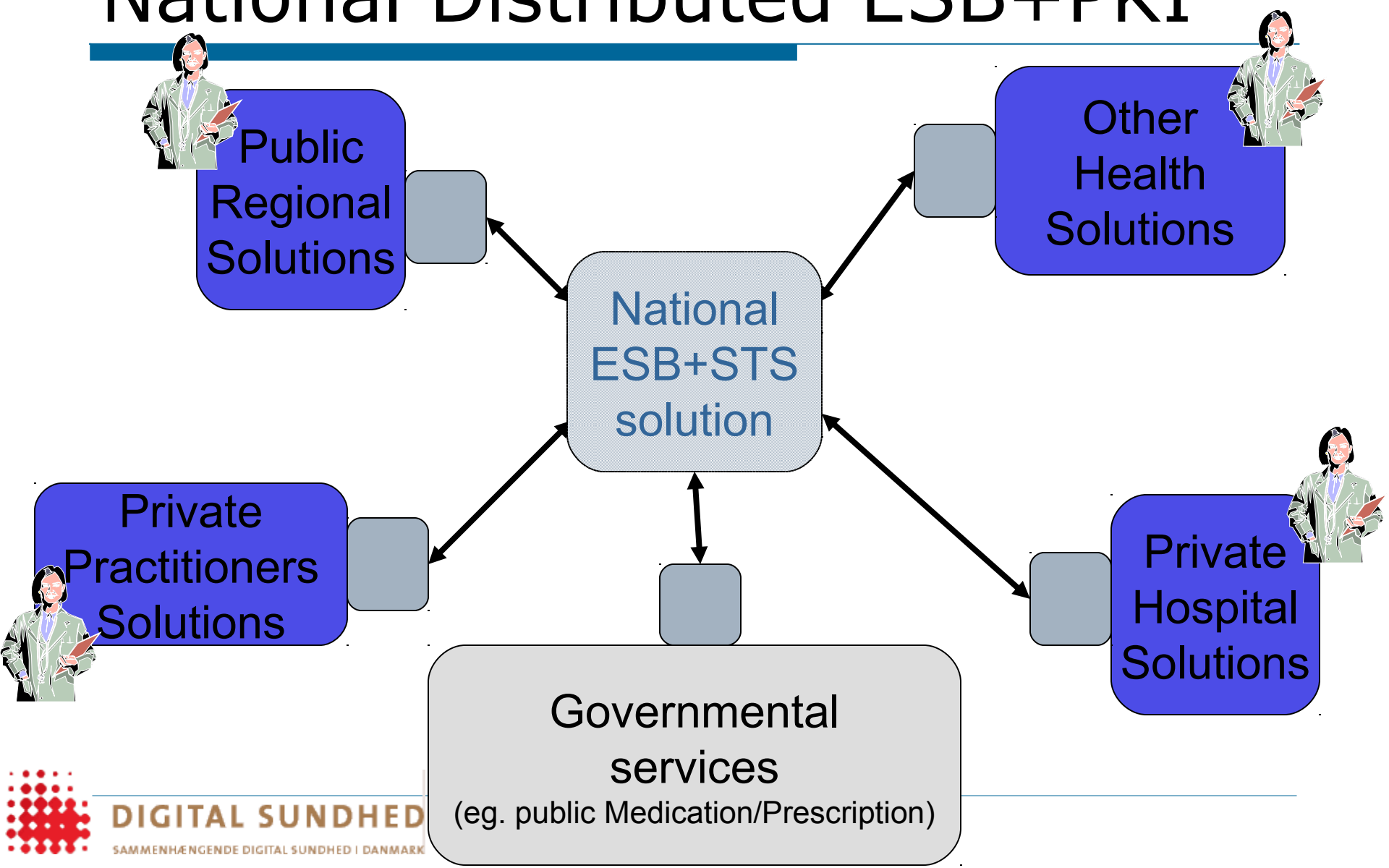
National ESB+STS solution

Private Practitioners Solutions

Private Hospital Solutions

Governmental services
(eg. public Medication/Prescription)

National Distributed ESB+PKI



Preconditions for implementation

- Based on a "Federated ESB" pattern
- Other parties are now exposing services on the "National ESB"
- Digital Health is responsible for QoS etc.
- Preconditions:

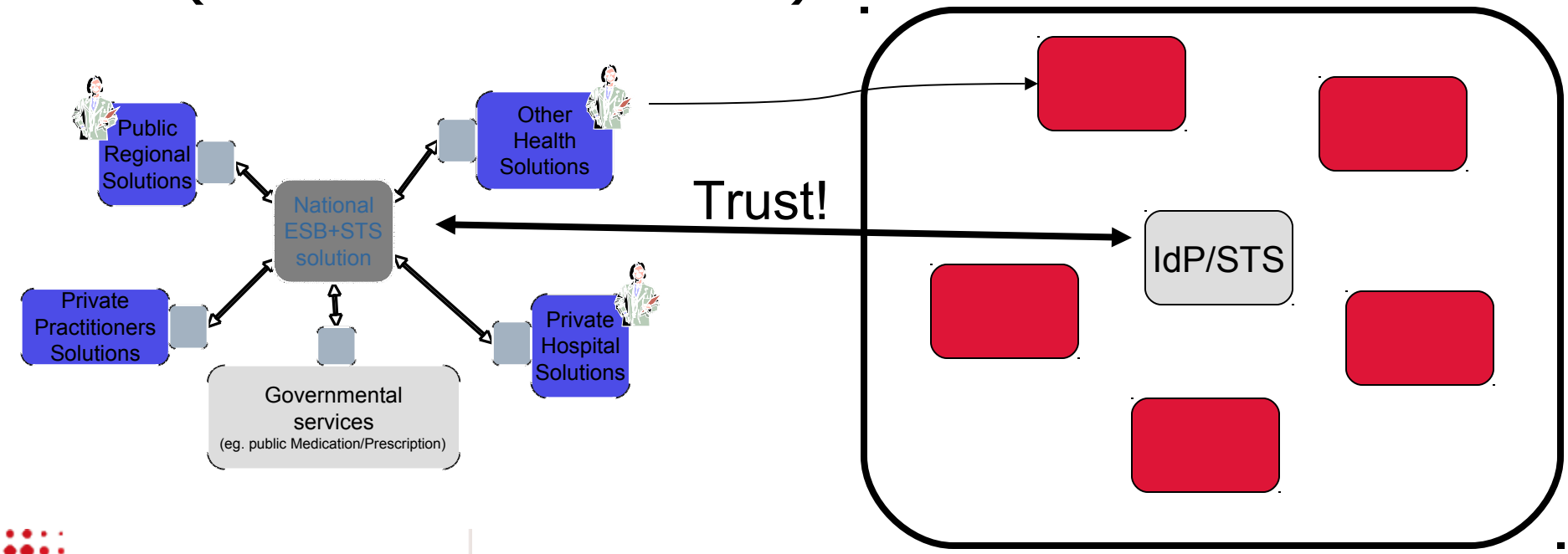
= Many parts of IAF

Taking IAF further?

- IdP's/STS' are also WSP's
- My wish:
  Separate the WSP assessment criteria from and create "SPAF"
- Make IAF an IdP specialization of "SPAF"

Another example of IAF usage

- Health Professionals will once and again need access to other domains (other federations)



Thank You!

Questions?

Public Key Superstructure
"It's PKI Jim, But Not As We Know It!"

Stephen Wilson
Lockstep Consulting Pty Limited
11 Minnesota Ave
Five Dock NSW 2046 AUSTRALIA
+61 414 488 851
swilson@lockstep.com.au

ABSTRACT
While PKI has had its difficulties (like most new technologies) the unique value of public key authentication in paperless transactions is now widely acknowledged. The naïve early vision of a single all-purpose identity system has given way to a more sophisticated landscape of multiple PKIs, used not for managing identity *per se*, but rather more subtle memberships, credentials and so on. It is well known that PKI's successes have mostly been in closed schemes. Until now, this fact was often regarded as a compromise; many held out hope that a bigger general purpose PKI would still eventuate. But I argue that the dominance of closed PKI over open is better understood as reflecting the reality of *identity plurality*, which independently is becoming the norm through the Laws of Identity and related frameworks.

This paper introduces the term "Public Key Superstructure" to describe a new approach to knitting together existing mature PKI components to improve the utility and strategic appeal of digital certificates. The "superstructure" draws on useful precedents in the security printing industry for manufacturing specialized security goods without complicated or un-natural liabilities, and international accreditation arrangements for achieving cross-border recognition of certificates. The model rests on a crucial re-imagining of certificates as standing for *relationships* rather than identities. This elegant re-interpretation of otherwise standard elements could truly be a paradigm shift for PKI, for it normalizes digital certificates, grounding them in familiar, even mundane management processes. It will bring profound yet easily realized benefits for liability, cost, interoperability, scalability, accreditation, and governance.

General Terms
Authentication, Security, Standardization, Legal Aspects.

Keywords
Public key infrastructure, PKI, digital certificates.

IDtrust '08, March 4-6, 2008 Gaithersburg, MD

1. INTRODUCTION: HOW DID PKI GET SO HARD?
PKI has been a notoriously disappointing technology. Much of the difficulty experienced bringing it to market can be traced back to the earliest PKI simply coming too soon. In the absence of well specified applications, an intuitive but ultimately distracting metaphor was allowed to dominate the agendas and rule the thinking of developers, product managers, policy makers, lawyers and standards setters. Ironically, while the metaphor was deceptively simple, it bred almost unlimited complexity.

Technically of course, an X.509 certificate does little more than bind a name to a public key value. This sort of arcane service hardly makes for compelling advertising, so naturally the early CAs and web browser vendors needed a simpler bit of imagery. What, they asked, was an X.509 certificate *like*? Whoever first suggested it was *like a passport* deserves a special place in the PKI hall of infamy.

1.1 The digital passport red herring
The notion of a digital passport had extra traction in the mid 1990s as it was already widely appreciated that creating "trust" in and on the Internet was going to be a crucial challenge.[1] So the world was much enamored with the idea that proving one's identity with a universally recognized passport is literally the key to doing business online. Perhaps the charm of the passport metaphor distracts people from the reality that most business is actually done in a local context. Furthermore, personal identity is not usually paramount, at least not in business; as a general rule in all walks of life, the less identification needed the better.

The implied objective of a one-size-fits-all digital certificate was perhaps the single biggest complicating factor in all of PKI. By trying to make one certificate type meet the needs of all possible transactions, the legal arrangements became almost entirely unmanageable. A good question is why the futility of the universal PKI project wasn't spotted sooner.

The first Certification Authorities set up shop years before any meaningful e-commerce was on offer. Imagine trying to draft a

[1] Peter Steiner's cartoon "On the Internet, nobody knows you're a dog", the exemplar of the 'trust problem', appeared in New Yorker magazine in July 1993, thus predating almost all e-commerce as we know it today.

subscriber agreement when you have no idea what a certificate is going to be used for. Any reasonable Threat & Risk Assessment has to explicitly relate to the application and its context. In the absence of any actual details, the only possible risk mitigation ploy is to enforce strenuous proof-of-identity checks on certificate subscribers so that in the event that something goes wrong, there is the prospect of sheeting home some blame.

If any trustworthiness at all could be vested in this type of certificate, then it is premised entirely on the rigor of the CA's certification practices. And so in turn the quite artificial situation arose where CAs, all of them brand new "trust" businesses, competed on the quality of their arcane Certification Practice Statements, as if customers could really be expected to read and care about these tomes.

So the digital passport idea, divorced as it was from any actual application, led immediately to legal complexities. The metaphor all on its own is likely to also be responsible for several operational quagmires, as follows.

1.1.1 Cost to the end user

Retail digital certificates are famously expensive and inconvenient to obtain. In many jurisdictions, the *de facto* proof-of-identity test was precisely (and arbitrarily[2]) the same as that of a passport. Such a level of identity vetting is highly unusual in everyday business. In Australia, an opt-in national PKI for healthcare professionals was met with strong opposition on this basis; administrators long complained of PKI being a "slow and unwieldy process" because of the personal identity vetting, and have cited "resistance from doctors and staff to fill out [registration] forms" as a major reason for the slow uptake of certificates [7].

1.1.2 The failure of Post Office CAs

The national postal authorities of several countries, including Australia, Belgium, Hong Kong, Malaysia, the UK and the US, rather quickly started up CA businesses on the strength of their existing privileged positions as passport registrars. Most post office CAs failed to generate any sustainable free market customer base for their certificates.

1.1.3 Cross Certification

The most unfortunate (albeit subtle) side effect of the passport metaphor in my view was the way it helped to inspire Cross Certification and certificate "policy mapping" as the dominant frame for creating PKI interoperability. Cross Certification is the orthodox way for certificates issued in different domains to be assessed for 'compatibility'. What's really going on here is a determination as to whether or not one CA's detailed processes – especially their registration policies – are *equivalent* to another's.

[2] In Australia, the identity vetting protocol for passports and the related know-your-customer rules for opening a bank account were codified in legislation in 1988. The same identification standard was uncritically adopted by default eight years later when Standards Australia made its first efforts to standardize PKI [23]. Yet there is no logical connection in fraud mitigation measures between face-to-face retail banking and online transactions, and no obvious reason for the same identity vetting standards to have been carried over.

Leaving aside the practical matter that it shouldn't even be necessary for both counterparties to carry a certificate and belong to a CA, the deep limitation of Cross Certification is its inability to recognize different certificates as being fit for different purposes. Consider whether *it even makes sense* to ask if the certificate of for instance a Taiwanese doctor is "equivalent" to the certificate of an American immigration official. Cross Certification together with its offspring, the Bridge CAs, are premised on the assumption that one identity is all we need. As we shall see later, that notion has been repudiated several times over in the decade or more since PKI got its false start.

1.2 E-mail not a killer application for PKI

A total lack of real applications would explain why e-mail became by default the most talked about PKI application. Many PKI vendors to this day continue to illustrate their services and train their users with imaginary scenarios where our heroes Alice and Bob breathlessly exchange signed e-mails. Like the passport metaphor, e-mail seems easily understood, but it manifestly has not turned out to be a 'killer application', and worse still, has contributed to a host of misunderstandings.

The story usually goes go that Alice has received a secure e-mail from stranger Bob and wishes to work out if he is trustworthy. She double clicks on his digital signature and certificate in order to identify his CA. And now the fun begins. If Alice is not immediately trusting of the CA (presumably by reputation) then she is expected to download the CP and CPS, read them, and satisfy herself that the registration processes and security standards are adequate for her needs.

Does this sort of rigmarole have any parallel in the real world? A simple e-mail with no other context is closely equivalent to a letter or fax sent on plain white paper. Under what circumstances should we take seriously a message sent on plain paper from a stranger, even if we could track down their name?

In truth, the vast majority of serious communications occurs not between strangers but in a rich existing context, where the receiver has already been qualified in some way by the sender as likely being the right party to contact. In e-business, routine transactions are not usually conducted by e-mail but instead use special purpose software or dedicated websites with purpose built content. Thus we see most of the digital signature action in cases such as e-prescriptions, customs broking, trade documentation, company returns, patent filing and electronic conveyancing.

Several important simplifying assumptions flow from the fact that most e-business has a rich context, and these should be heeded when planning PKI:

1.2.1 Emphasize straight-through processing

In spite of the common worked example of Alice and Bob exchanging e-mails, the receiver of most routine transactions – such as payment instructions, tax returns, medical records, import/export declarations, or votes – is not a human but instead is a machine. The notion that a person will examine digital certificates and chase down the CA and its practices is simply false in the vast majority of cases. One of PKI's great strengths is the way it aids straight-through processing, so it has been a great pity that vendors, through their training and marketing materials, have stressed manual over automatic processing.

1.2.2 *Play down Relying Party Agreements*

The sender and receiver of digitally signed transactions are hardly ever un-related. This is in stark contrast to orthodox legal analyses of PKI which foundered on the supposed lack of contractual privity between Relying Party and CA. For example the Australian Government's extensive investigation into legal liability in digital certificates after 111 pages still could not reach a firm conclusion about whether a "CA may owe a duty of care to a [Relying Party] who is not known to the CA" [22]. The fact is, this sort of scenario is entirely academic and should never have been given the level of attention that it was. The idea of a "Relying Party Agreement" to join in contract the RP and the CA is moot in all "closed" e-business settings where PKI in thriving. It is this lesson that needs to be generalized by PKI regulators, not the hypothetical model of "open" PKI where all parties are strangers.

1.2.3 *Play down certificate path discovery*

The fact that in real life, parties are transacting in the context of some explicit scheme, means that the receiver's software can predict the type of certificate that will most often be used by senders. For instance, when doctors are using e-prescribing software, there is not going to be a wide choice of certificate options; indeed, the appropriate keys and certificates for authenticating a doctor issuing a prescription will likely be installed at both the sending and receiving ends, at the same time that the software is (see also a worked example at subsection 4.4). When a doctor writes a prescription, their private key can be programmatically selected and invoked to create a digital signature, according to business rules enshrined in the software design. And when such a transaction is received, the software of the pharmacist (or insurance company, government agency etc.) will similarly 'know' by design which certificates are expected to verify the digital signature. All this logic in most transaction systems can be settled at design time, which can greatly simplify the task of certificate path discovery, or eliminate it altogether. In most systems it is straightforward for the sender's software to attach the whole certificate chain to the digital signature, safe in the knowledge that the receiver's software will be configured with the necessary trust anchors (i.e. Root CA certificates) with which to parse the chain.

2. BIG PKI: ONLY EVER A STRAWMAN

"Big PKI" should have always been seen as a strawman, one that was construed with no real compelling need. Instead, in the vain attempt to allow stranger-to-stranger e-business, PKI inevitably grew ever more bloated and vulnerable to criticism.

Just consider the conventional sort of definition of PKI. NIST defines PKI as "personnel, policy, procedures, components and facilities to bind user names to electronic keys so that applications can provide the desired security services".[3] Microsoft considers it to be "the combination of software, encryption technologies, processes, and services that enable an organization to secure its communications and business transactions"[4] (the definite article at the start of the definition rather extravagantly seems to admit no other way for an organization to transact other than PKI). From the outset, this language sets PKI apart from any other authentication system. Traditional PKI requires an enterprise to commit itself to establishing novel and incredibly complex policies and procedures, in addition to deploying public key components. Allowing any new technology to so impact a business is plainly asking for trouble.

From the late 1990s a succession of critics sought to demolish PKI, usually on the basis of the mirage of a universal digital passport. The best known popular critique was probably that of Ellison and Schneier in 2000 [13] which detailed ten risks that we were supposedly "not being told about". On closer examination however, most of their concerns apply to the quality of security policies and the safekeeping of cryptographic keys in any setting, not just PKI. And when Ellison and Schneier do focus on PKI, it is actually the special case of a global infrastructure that they have in mind. For example, their argument that PKI doesn't resolve "which John Robinson is he" is unimportant in closed PKIs where communities of interest already have – indeed, *must* have – reliable mechanisms for guaranteeing unique handles in their local namespace. No PKI implementation should ever change the way users are known by the parties they deal with.

Another much cited assault on PKI came from academic law professor Jane Winn in her catchy 2001 exposé of "the shocking truth" [28] about digital certificates. Winn lampooned the prospects of forming new contracts over the Internet purely on the strength of strangers' certificates. Yet far from producing the definitive critique of PKI in general, she herself wrote that "what is now becoming apparent is that a more important [application] for digital signatures than 'open' Internet commerce among strangers may be 'closed' Internet commerce systems among parties *already in contractual privity with each other or to a system administrator*" (emphasis added).

It is this point that helps explain why, in the face of such widespread disillusionment and cynicism, PKI through the early to mid noughties continued to grow steadily and thrive in pockets. Well known examples include the Johnson & Johnson corporate PKI,[5] the Pan Asia Alliance trade documentation system,[6] Swedish financial sector's *BankID*,[7] the US Patent & Trademark Office online patent filing system,[8] the pharmaceutical industry's SAFE *Biopharma* scheme,[9] and Skype.[10]

It's possible that the florid ambitions of early PKI were amplified by dot-com mania. One analysis of the underwhelming demand for third party CA audit services suggested:

> *"[During] the Internet boom there was a belief that e-business was going to release a massive pent-up demand to conduct stranger-to-stranger commerce. But truly un-vetted business introduction is rare"* [15].

---

[3] See http://csrc.nist.gov/nissc/1999/program/isso/tsld005.htm.

[4] See http://tinyurl.com/3ahtaw.

[5] See http://www3.ietf.org/proceedings/04nov/slides/easycert-1/easycert2.ppt.

[6] See http://www.paa.net.

[7] See http://www.bankid.com.

[8] See http://idtrust.xml.org/entrust-us-patent-office-success-story.

[9] See http://www.safe-biopharma.org.

[10] See http://share.skype.com/sites/security/2006/02/zui_and_the_skype_pki.html.

In parallel with the dawning realization that PKI works best when parties are not strangers, several other shifts in the identity management landscape at large have informed contemporary thinking, as follows.

2.1.1 The need for more than one certificate

The chair of the IETF PKIX Working Group Dr Stephen Kent has criticized the rigidity and unreality of orthodox "big CAs". In 2005 he told a conference of the Asia PKI Forum:

> *"For many big CAs, there is an assumption that a single certificate is all a user should need. This assumes that one identity is sufficient for all applications, which contradicts experience. For personal privacy and security, multiple independent certificates per user are preferable"* [18].

2.1.2 Supply chain perspective of certificates

In 2005 the OASIS PKI Technical Committee developed a new *digital certificate supply chain* to help better describe various cost components that impact on return on investment in PKI [27]. The supply chain recognizes separable components of a PKI:

- the toolkits, libraries, services and so on used to PKI-enable software applications
- end user support
- digital certificates themselves
- Registration Authority services, costs and overheads
- Certification Authority operations
- key media (e.g. smartcards, SIM cards), if applicable.

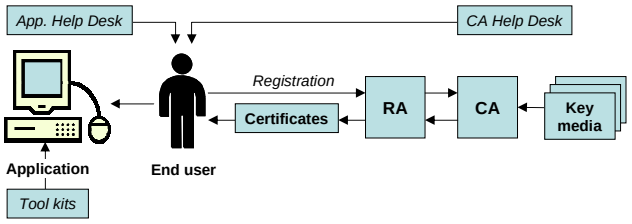


Figure 1: Digital Certificate Supply Chain

One important upshot of the supply chain perspective is that it more vividly underscores the separation of CA and RA, which most often are legally treated as the one entity. For instance, the most detailed legal analysis yet to be carried out on PKI in Australia, by law firm Clayton Utz in 2000, assumed that the CA carries out the functions of RA [22]. Decoupling the CA from the RA can be usefully extended further to create a *wholesale* approach to certificate production, as we shall see later.

2.1.3 Relationship Certificates

Greater separation of CA and RA helps the fresh formulation of "Relationship Certificates", originally developed by me for the Australian Government's *Gatekeeper* PKI program [5].

Orthodox digital certificates representing the personal identity of their Subjects are issued after an RA performs identity proofing on the applicant. They therefore represent an affirmation by the RA that the Subject has passed certain documented threshold tests relating to evidence of identity. A *Relationship* Certificate simply represents a different type of affirmation, namely that the Subject has a particular type of relationship with the RA. By extension, a Relationship Certificate can thereby stand for the Subject's rights or entitlements to participate in certain transactions sanctioned by the relationship. In a great many cases, significant and powerful credentials derive directly from membership of chartered professional associations, or simply from being employed by a company, and so can be instantiated by the relationship the user has with the organization's administration. Under these circumstances, a Relationship Certificate issued by the administrator means *nothing more and nothing less than the fact that the Subject is a member of the organization*; in particular, this type of certificate makes no formal representations about the subject's identity outside the organization. Relationship Certificates should lose their meaning outside the context of the relationship.[11]

Relationship Certificates have philosophical parallels with the idea of "authorization PKI" which has been floated sporadically as an alternative to "authentication PKI". For example, a recent IETF draft for secure Internet routing suggests that "[if] issuers need not verify the right of an entity to use a subject name in a certificate, they avoid the costs and liabilities of such verification" [6]. I believe that Relationship Certificates represent a fundamental shift in the way we think about PKI mainly because they break the nexus between authentication and authorization. A Relationship Certificate can evince its Subject's authorization to act in a given role on a given transaction domain *without needing to separately establish the person's "identity"*. In this regard, Relationship Certificates differ from two superficially similar constructions:

- ***Attribute Certificates***. Classically, Attribute Certificates do not bind public keys to users but rather only bind authorizations to names. They therefore cannot be used on their own to validate digital signatures, but instead are generally used in conjunction with some other general purpose public key "identity" certificate (a degree of complexity that appears to have inhibited the take-up of Attribute Certificates in commercial PC applications). Relationship Certificates on the other hand are just regular public key certificates. They stand alone to assert the Subject's role or responsibilities, and can be processed by conventional software. That is, Relationship Certificates can substitute for conventional X.509 certificates in standard applications without any software modifications; only the "business rules" for interpreting what a certificate means need updating.
- ***SPKI ("Simple PKI")***. SPKI [12] was formulated in the late 1990s in response to some of the challenges summarized above, as a way of mapping an authorization directly to a key, thereby skipping the cumbersome mappings of names to keys (using regular Identity Certificates), and authorizations to names (using Attribute Certificates). In this regard Relationship Certificates closely resemble SPKI Authorization Certificates. Unfortunately, SPKI has not penetrated the market as far as hoped, perhaps because it positions itself

[11] In the real world, all credentials have context, and the appropriate credential depends on the transaction at hand. For example, if a doctor were pulled over by a traffic cop and asked to show her drivers licence, she should get nowhere trying to present her medical qualifications.

implicitly as an adjunct to the name-key mapping. SPKI is often associated with a needlessly complicated triangle formed from Identity, Authorization and Attribute certificates; see for example [20].

Further operational details of how Relationship Certificates could be implemented are provided in Section 4 and a worked example provided in subsection 4.2.3.

2.1.4 Smart key media

The historical complexity faced by users in managing keys and certificates is being almost entirely put to bed by an increasingly rich array of smart key media. With key pair generation integrated into their chips, and certificate lifecycle management being absorbed into card management systems, PKI enabled smartcards in particular are set to transform PKI. They are exactly as easy to use as any conventional magnetic stripe card.

2.1.5 New thinking about identity

Finally, we can look to the new wave of thinking about identity in general as an indication of better ways to utilize PKI. Much of the focus of "identity 2.0" (as promoted by organisations such as sxip[12]) is on the multiplicity of things that we say about ourselves, and the things that others say about us. That is, identity 2.0 begins with a realization that the usefulness of online identity depends on context, and it must be responsive to the natures of the diverse relationships we have with those we transact with.[13]

It seems to be increasingly accepted that people live with multiple identities. The preeminent exposition of modern identity theory is probably Kim Cameron's Laws of Identity [10]. The laws include a new definition of *digital identity* as "a set of claims made by one digital subject about itself or another digital subject". Cameron knows that this sort of relativist definition might not sit comfortably with everyone:

> *"We recognize [that our definition] does not jive with some widely held beliefs – for example that within a given context, identities have to be unique. Many early systems were built with this assumption, and it is a critically useful assumption in many contexts. The only error is in thinking it is mandatory for all contexts."* [10]

But Cameron is certainly not alone, not anymore. Other researchers have reached the view that there may be a many-to-one mapping of identities onto entities; see e.g. Jøsang and Pope:

> *"An identity is a representation of an entity in a specific application domain. For example, the registered personal data of a bank customer, and possibly also the customer's physical characteristics as observed by the bank staff, constitute the identity of that customer within the domain of that bank. … A person may of course have different identities in different domains. For example, a person may have one identity associated with being customer in a bank and another identity associated with being an employee in a company."* [17]

The notion of what I would call *identity plurality* is not merely a semantic or philosophical point. A simple example demonstrates that in business we clearly conduct ourselves according to multiple identities, and that we seamlessly switch between them without trouble. Furthermore, when we exercise a context-dependent identity, we beneficially mask our biological one. Imagine that a company Acme Inc. has a corporate bank account with A Banking Corporation (ABC). The Acme company secretary, Alice, would be a signatory to the Acme bank account and would have custody of an ABC key card for the purpose. Alice might also hold a personal account with ABC. Now, when she banks on behalf of Acme, Alice exercises a different identity compared with when she banks on her own behalf, even if she happens to access both accounts during the same visit to the branch or ATM. The distinction is both emotional – Alice probably won't feel any real attachment to the millions of dollars she routinely handles for the company – and legal. Corporate law says clearly that the Acme account holder is not Alice but the company.

One deep implication for PKI of identity plurality is that it inverts the expectation that closed PKI is a compromise while open PKI is the proper long term goal. On the contrary, we should now appreciate that open PKI would be a special and highly theoretical instance. It is the closed PKIs – each with its own arrangements and business rules – that represent the general case.

3. WHAT IS PKI REALLY GOOD FOR?

I contend that clearly the best use of PKI is to help automate electronic transactions in a particular context between parties that already have a formal relationship.

The orthodox textbook amount of the benefits of PKI invariably list authentication, integrity and something called "non-repudiation". These high level properties may actually be delivered by all manner of technologies, a fact that made early PKI's over-inflated marketing claims seem frankly silly, even at the time.[14]

3.1 A clearer benefit description for PKI

We need a more sophisticated shared understanding of what makes PKI unique. I suggest its unique benefits would be better told as follows:

- Digital signatures create long-lived, tamper-resistant evidence of 'who did what to whom', which is so critical to electronic transactions carrying high legal risks or compliance requirements.
- PKI, when deployed with hardware key media like smartcards, is recognized as "the only practical solution [to eavesdropping and account hijacking] today" [9]; digital signatures

[12] See http://www.sxip.com.

[13] However, many in the identity 2.0 movement go from this background to a position of desiring all diverse relationships to be federated into a multi-context transcendent identity. In my opinion this is one step too far. Dick Hardt's famous identity 2.0 conference presentation is frankly utopian in the way it advocates linking all our reputations together, for it overlooks the privacy problems arising when linked records are exposed by accident, when wrong doers exploit the linkages, or when a user seeks to sever one of their relationships for some reason.

[14] PKI has no monopoly on "non-repudiation" despite the term only being coined in connection with it. PKI marketing too often suggests that only PKI delivers non-repudiation. If this were true, credit card holders who use their cards online could try to mischievously repudiate any one of their payments on the basis that it was *not* digitally signed and therefore did *not* have "non-repudiation"!

originated by the end user protect against Man-in-the-Middle attack, while smart key media offer a sufficiently compact logic engine to be certifiably resistant to malware.

- Digital certificates can convey *authority information* – like credentials, licences, affiliations and so on – and digital signatures bind that authority information directly to messages, to decentralize and greatly simplify transaction processing.

PKI digital signatures are persistent over both time and 'distance', meaning the separation of sender and receiver. At essentially any time in the future[15] a digitally signed transaction can be easily re-validated to prove where it originated: all that is needed is a trusted copy of the root public key, the certificate chain, and the relevant CRLs (all of which are routinely available from any decent CA). In addition, authority information about the sender can be sealed into their certificate at the time of issue, and this authority information also has great longevity, thanks to the digital signature of the Certificate Authority on the certificate.

The integrity of digitally signed data is not reduced by being copied or forwarded across systems or across borders. In contrast, other authentication technologies rely heavily upon audit logs to prove 'who did what to whom'; forwarding non-PKI transactions from one system to another complicates and dilutes the strength of the audit trail. So PKI is uniquely suited to complex transaction environments, like healthcare, pension fund management and trade documentation, where there are multiple relying parties, formal authorizations, and/or long lifetimes.

3.1.1 The challenge of persistent credentials

Verifying transactions originating from professionals is a case in point. Consider a lawyer who signs conveyancing documents relating to a land sale. When the contract is settled, all parties (the buyer, the seller, their respective banks and so on) will be acutely interested to know that the lawyer's credentials are valid. It is straightforward to check credentials online, at the time, by looking up a database of qualified practitioners. But in electronic conveyancing, what becomes important is the ability to check the credentials of a lawyer who signed documents in the past. Unless special measures are taken to archive practitioners' databases, it is difficult to obtain definitive machine readable information about the state of someone's credentials at a given time in the past. With digital signatures and digital certificates on the other hand, the matter becomes trivial: if Relying Party software knows the relevant Policy OID and has a trusted copy of the root public key, then it can verify the credentials of the lawyer at the same time as it verifies the digital signature, no matter how old it is (with reason, as qualified by the long term risks mentioned previously).

[15] Several very long term risks ultimately threaten the validity of old digital signatures. Brute force attack by future computers on asymmetric cryptographic algorithms, exploitation of likely weaknesses in MD5 and SHA-1, and eventually the possibility of quantum computing, all mean that any digital signatures intended to remain valid for a few *decades* should have their keys and certificates comprehensively archived. Note that the stability and usability of archive media over the decades is another quite separate challenge.

3.2 PKI in plain English

The steadily improving automation of digital signature and certificate management operations means that the way we describe PKI to lay people can now side-step the technical details of asymmetric cryptography, hash functions and so on, and focus instead on what it actually *does*. A fresh, plain English description might run as follows (assuming smartcards are the key media).

> *A smartcard plus application software combine to produce* digital signature *codes for electronic transactions. Unlike any other electronic signature method, digital signature codes are unique to the owner and also to each transaction. Digital signatures operate as if a personalized electronic stamping machine was inside each smartcard, creating a specific tamper resistant 'mark' on each message or file created by the card holder. Digital signatures remain valid indefinitely; at any time in future, the 'mark' can be easily verified to prove its origins.*
>
> Digital Certificates *are electronic notices that bind identities to such devices as smartcards.[16] Certificates can thereby bind individuals to transactions signed using their smartcards. A digital certificate can identify the card holder and can also hold any other information that the issuer is qualified to declare. If the issuer is authoritative over information such as professional credentials, then that information can be sealed within its digital certificates and thus bound to each card holder plus the transactions they sign.*
>
> *To process digitally signed transactions, the receiver's software requires a copy of the sender's certificate, plus a special "master code" – known as a root certificate – which is used to mathematically validate all certificates in a given PKI scheme. Different master codes define different PKI schemes, be they sector-specific, national or general purpose such as SSL website authentication. Application software can ship with all necessary master codes, or can have them installed later.*
>
> *Digital certificates can be electronically revoked at any time. Revocation may be requested by the holder in the event that they lose their smartcard. Alternatively, revocation of a professional's certificate may follow automatically from their membership lapsing, their qualifications being cancelled, or their employment changing.*

3.3 Modern PKI success stories

Many of the more recent PKI success stories resonate with the concepts of identity plurality and digital certificates having more to do with multiple relationships than a single identity. Examples follow.

- A large public hospital in Australia developed a new "Known Customer" certificate to be issued on smartcards to several thousand of its staff [8]. The intended digital signature applications include electronic medical notes created by nurses, electronic hospital discharge notes, and online employee self-service access to pension fund administration, leave forms and so on. The hospital's human resources

[16] This simplified account deliberately but without loss of generality suppresses the intermediate detail that the certificate actually binds the identity to a key pair, which is separately bound to the smartcard by way of hardware key management.

department will operate a delegated Registration Authority workstation. A commercial back-end CA will independently manufacture customized certificates on request from the RA, and inject them onto smartcards. The same CA will be able to produce similar but distinct Relationship Certificates for other communities of interest in the health sector. Overlap between healthcare communities is commonplace, with geographically related area health services often sharing information management resources and infrastructure. "Interoperability" of the hospital's staff certificates with other local applications will be easily fostered simply by promulgating knowledge of the certificate Policy OIDs.

- The Australian government has been exploring how digital certificates can act as electronic credentials for a number of different types of professionals. A state association for legal professionals has researched how digital "practicing certificates" can be issued to attorneys.[17] The most compelling application for digital signatures in the practice of law is electronic conveyancing. E-conveyancing is forecast to provide direct savings of AU$70 per transaction for vendors and purchasers, and an overall saving to industry of AU$33 million p.a. by 2010, assuming 66% of transactions are by then done online [24].

- Most e-health projects around the world anticipate the use of digital certificates. The use of digital signatures in the pharmaceutical industry has been fostered by the Food & Drug Administration's Title 21 Code of Federal Regulations (21 CFR Part 11) in respect of Electronic Records and Electronic Signatures. Health smartcards in France[18] and Germany[19] are currently being upgraded with PKI-capable chips so as to support a new wave of applications that require patient signatures, such as e-prescribing. The Australian federal Department of Health and Ageing in 2006 commissioned independent security analysis that strongly endorsed digital certificates for e-prescribing [2].

4. PUBLIC KEY SUPERSTRUCTURE

Having painted a newly optimistic picture for the future of PKI, one that resonates with broader identity management trends, I will now describe a number of fresh ways to better knit together extant mature building blocks – X.509 and similar certificates, RAs, CAs and PKI audit services – to deliver better, more flexible transaction authentication.

4.1 Relationship Certificates in practice

Relationship Certificates, as described in subsection 2.1.3, are best managed within an arrangement where a defined "community of interest" deploys digital certificates that represent membership of the community. Operationally, Relationship Certificates are issued with the administrator of the community acting as a delegated RA.

Such arrangements have been studied extensively and piloted by both the legal and medical professions in Australia, as mentioned in subsection 3.3. In response to market demand for PKI-based digital credentials that convey richer information about professional qualifications without being burdened with artificial registration requirements, the Australian Government PKI program recently introduced a special category for Relationship Certificates [5].

4.1.1 The Relationship Certificate profile

To be most effective, Relationship Certificates would have information in their X.509 (or similar) profile to specify the precise nature of the relationship between RA and Subject, allowing straight-through processing by any Relying Party software application configured to recognize the validity of the relationship. The best way to codify the meaning of a Relationship Certificate is probably in the Policy OID, which can be specified at design time.

Ideally, technical controls should be implemented as well to make it difficult to misuse a Relationship Certificate outside its intended context. One way to implement technical restrictions on misuse would be to include a *Critical* extension in the profile. Recall that the X.509 standard requires any software processing a certificate which has an extension marked as *Critical* to reject that certificate unless it expressly recognizes the extension. Since special purpose software (as opposed to general purpose web and e-mail clients) is usually used in PKI-enabled transaction systems, within communities of interest, programming in awareness of *Critical* extensions is easy. And by the same token, it is safe to assume that if a given software program does not recognize the *Critical* extension, then it is proper behaviour to reject the certificate, on the grounds that such certificates are not supposed to be used outside special purpose applications. *Critical* extensions proved unpopular in the past because they were thought to harm interoperability. But if a special purpose Relationship Certificate is only intended to work with certain applications, then "interoperability" is more or less moot, since no other applications should be expected to accept it.

4.1.2 Practical benefits of Relationship Certificates

Relationship Certificates would bring major simplifications over third party identity certificates in several areas:

- Overheads associated with registering for certificates are greatly reduced; customers already known to the administrator in a community of interest will be able to receive certificates almost automatically without having to present in person at an unfamiliar RA.

- Certificate Subjects will require no legal relationship with the backend CA; any important new obligations introduced by PKI – such as responsibility to safeguard one's smartcard and promptly report its loss – can be folded into the administrator's formal contractual relationship with its members, rather than expressed in the traditional CA's "Subscriber Agreement".

- Users will no longer be required to pay up-front for a certificate from a third party CA in order to use PKI-enabled applications.

---

[17] See news about this project at http://www.galexia.com/public/projects/projects-Law.html (accessed 31 Jan 2008).

[18] http://www.sesam-vitale.fr/programme/programme_eng.asp.

[19] http://www.die-gesundheitskarte.de (in German).

- Furthermore, the price of certificates should fall towards "wholesale" levels, because the cost of identity proofing associated with traditional identity certificates will be eliminated.
- Support overheads and complexities may be lessened by having just one help desk for all business, application and certificate-related matters.

4.2 Wholesale certificate production

Historically, CAs have been tied legally into the whole of the certificate management process, no matter how they might operationally involve RAs in the registration and certificate lifecycle management processes. CAs tend to be joined in liability arrangements and contracts to potentially any wrongdoing or misadventure associated with certificates. Certificate policies, practice statements and user agreements have been correspondingly difficult to construct. To date, the separation of roles of RA and CA has done little to quarantine the two functions from one another, nor to simplify liability arrangements. Accreditation remains complex and sensitive to the slightest changes at either the RA or CA.

There might be a new way however of looking at backend CAs, likening them to conventional *security printers*, and dramatically simplifying the way that legal liability is apportioned when something goes wrong with a digital certificate.

4.2.1 The business of security printing

For decades it has been well known that in order to combat fraud, special care must be taken in printing certain documents: blank checks and prescription pads in particular, as well as business forms, concert tickets, gift vouchers, barcodes and so on. A whole industry has been built around special printing technologies, including watermarks, holograms, reactive inks that detect photocopying, and micro-printing. Moreover, a coherent business model has been built around security printing bureau services. In many sectors, standards have been introduced to cover the necessary security of premises and processes, and formal accreditation schemes govern compliance with these standards. For example, since 1 January 2005, written prescriptions for controlled substances in California must be on tamper resistant security prescription forms produced by a printer approved by the State Board of Pharmacy.[20] And the United Kingdom's payment clearing regulators APACS introduced a formal Check Printer Accreditation Scheme (CPAS) in 1995.[21]

4.2.2 The governance of security printing

Many of the standards governing security printing closely resemble those for traditional CAs. For example, check printer accreditation typically covers assurance of the following aspects of the operation:

- *'Equipment & Materials*
- *'Premises Security (external security for prevention of unauthorized access, and internal security with appropriate restrictions on access to different areas*
- *'Process Security (end-to-end process controls from raw materials, through to end product, full audit trail in place for each print job, destruction of unused/damage stock, protection of confidential information, employee screening and confidentiality clauses)*
- *'Order Processing*
- *'Quality Assurance*
- *'Dispatch & Delivery (secure & auditable dispatch system, sign-off for delivery, secure transport arrangements, secure packaging, appropriate labeling not to identify as checks, processes for lost/stolen consignments)'* (adapted from [14]).

Accredited printers under the British CPAS variously emphasize their personnel screening, internal segregation of access-controlled security cages, and perimeter fences and monitoring systems. Clearly a similar degree of effort is involved physical, procedural and personnel security for security printing operations as for well run CAs such as those typified by accreditation under *Identrust*, *WebTrust for CAs*, Australia's *Gatekeeper* PKI scheme, or the UK's *tScheme*.

4.2.3 Worked example: barcodes and certificates

A practical worked example of how digital certificates could replace a conventional paper security mechanism helps to further develop the comparison between backend CAs and security printers.

Consider a stock exchange that arranges for statutory announcements made by listed companies to be communicated by fax and secured by means of barcodes. Each listed company is provided with a roll of self adhesive barcode labels. The barcodes uniquely identify the company and are individually serial numbered. When a statutory announcement needs to be made in accordance with the stock exchange's Listing Rules, the announcement is printed, signed by a duly authorized company officer, and has a barcode label affixed to it, before being faxed to the announcement processing centre. When received, optical character recognition software scans the fax, extracts the announcement, and verifies the barcode, before broadcasting the news across stockbrokers' screens. See Figure 2 below.

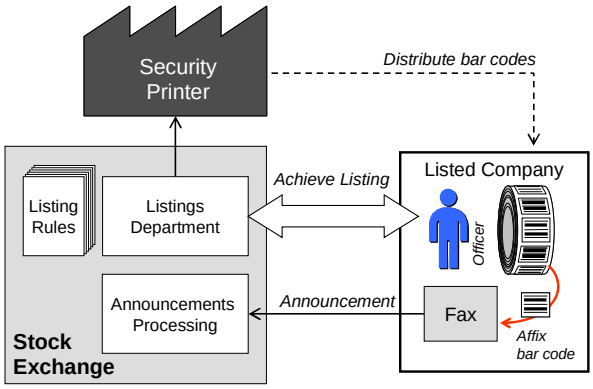


Figure 2: Authenticating faxed company announcements by means of secure barcode

The barcode label is an authentication token. Inclusion of a barcode on a fax is taken as reasonable evidence that the sender is a listed company, operating under the stock exchange's rules. Clearly such barcode labels are precious items. They need to be

[20] See http://www.ag.ca.gov/bne/security_printer_list.php.

[21] See http://www.apacs.org.uk/payment_options/cheques_accreditation_scheme.html.

produced by a reputable security printer, with the ordering and distribution processes being subject to strict controls.

Now let us consider how the announcement processing system could be reengineered to use PKI and electronic messaging in place of fax machines. Figure 3 shows a nearly identical system, where the listings unit operates an RA (not shown), and instead of ordering barcode labels from a security printer, it orders digital certificates from a backend CA. In order now to make an official announcement, the company officer would use the certificate to digitally sign an electronic message.

Note that the certificates issued in this particular scheme are an instance of *Relationship Certificates*. They are not intended in any way to stand for the "identity" of company officers. Rather, they represent nothing more and nothing less than the fact that each Subject is an officer of a company listed on the stock exchange.

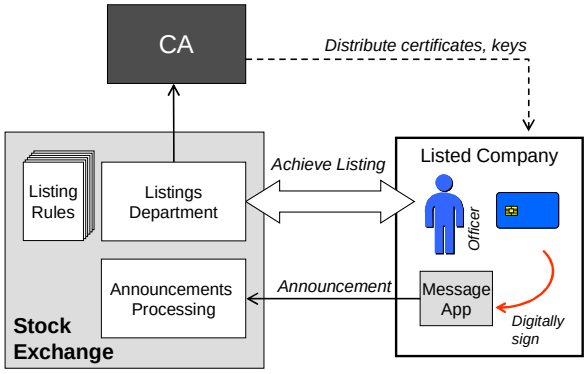


Figure 3: Authenticating electronic company announcements by means of digital certificate

Let's compare the security requirements of announcement methods that alternately use barcodes or digital certificates. Regardless the authentication method, the announcement system requires a common set of security controls:

1. The company listing process (which is where the relationship between a company and the stock exchange is established) must be robust and difficult to subvert.
2. It must be difficult to fraudulently order barcodes [or digital certificates].
3. Barcodes [or private keys and digital certificates] must be difficult to counterfeit.
4. The security printer [or backend CA] must be difficult to subvert.
5. Barcodes [or private keys] must be distributed and stored carefully.

If the conventions of orthodox PKI were to be applied to this operation, then a number of additional complexities would be imposed from outside on how the stock exchange runs its business. In particular, most PKI regulators today would expect standardized identity proofing for all certificate recipients at a level equivalent to passport application, irrespective of how the existing listing rules operate.[22]

Furthermore, the company officer, as certificate Subject, would generally have execute a user agreement *with the CA*. In contrast, requiring them to sign up with the security printer responsible for the barcodes would be unthinkable! Finally, changing backend CA typically triggers major re-accreditation of any regulated end-to-end PKI solution, because peak documents like the CP and CPS tend to intertwine all of the operational aspects. In the "real world", if the stock exchange gets a better deal from a competing printer, the changeover in backend operational matters would be completely invisible to the listed companies.

Comparing the digital certificate approach to the barcode system is suggestive of a more streamlined approach to PKI operations and accreditation. First note, referring to the list of security requirements on the previous page, that security controls can be clearly separated according to whether they relate to (1) the risk of *impersonation*, which tends to be managed by process or (2) the risks of *counterfeiting* or *theft*, which tend to be managed by technology:

- ***Impersonation related risks***
  - the company listing process must be robust and difficult to subvert.
  - it must be difficult to fraudulently order barcodes or certificates.
- ***Counterfeiting and theft related risks***
  - barcodes, private keys and certificates must be difficult to counterfeit.
  - the security printer or backend CA must be difficult to subvert.
  - barcodes or private keys must be distributed and stored carefully.

At the front-end of this authentication scheme, where the stock exchange deals with its companies, there is no logical difference between using barcodes or digital certificates, so we should expect the security of existing stock exchange registration processes to carry over to the PKI implementation without change. And at the backend, there is no need for either a security printer nor a CA to be concerned with the details nor even the integrity of the company listing and customer service processes, so long as there are controls in place to mitigate against wrongful ordering.

4.2.4 Implications of security printing for CAs

So, why couldn't we treat backend CAs in the same way as we treat regulated security printers? If a CA was set up as a service bureau, responsive to a particular set of RAs with which the CA has a specific arrangement, producing certificates on instruction more or less automatically via standard certificate request protocols, then a number of major simplifications to PKI management and governance could follow:

- The CA need have no interest at all in the semantic contents of the certificates it produces on instruction from a contracted RA. So long as there are safeguards in place to mitigate against false certificate requests being injected between the

[22] In the current climate of concern for homeland security, anti-money laundering, improved corporate governance and so on, it happens that many organizations are looking to strengthen their identity vetting processes, but nevertheless, that is an exercise that is not logically connected with PKI *per se* and the two should not be confused.

RA and the CA, the CA need not know anything at all about the RA's business process, nor the intended application of the certificates. The CA's business model and detailed processes could be held entirely constant over a wide range of different PKI applications. Protecting against injection of false certificate requests is a standard feature of most CA-RA products and is an express part of most if not all PKI product certification.

- There is no need for a contract or other legal arrangement between end users of certificates and the backend CA (just as there is no need for end users of checks and barcodes to have any relationship with the respective security printer).
- The CA's liabilities are straightforward to analyse and codify. For example (and in stark contrast to orthodox RA/CA arrangements) it seems clear that a CA would not normally be joined in legal action resulting from an RA being negligent in registering an impostor for a certificate. On the other hand, acts of omission or commission by a CA in producing poor quality certificates which led to harm on the part of message recipients, could be identified and prosecuted as such, and isolated from the RA.
- The meaning of the root key – which in orthodox PKI has led to so much confusion – can be likened to a unique watermark featured in all products from a given security printer. The chaining of a certificate back to the CA root would represent the simple fact that the certificate has come from an accredited facility (see subsection 4.4.1 for more details). The Root CA signature means only that it is extremely unlikely that a certificate has been forged, and does not impart any approval or endorsement of the contents of the certificate.[23]
- It should be much easier to novate backend CA service arrangements from one supplier to another.

Note that the security printing model would essentially preserve the physical, procedural, personnel and technological security controls of most current CA accreditation schemes, in order to protect against counterfeiting and subversion of the backend process. In particular, the benchmark of Common Criteria EAL4 rated CA and RA products would probably be retained, to help prevent fraudulent ordering of certificates.

4.3 Revisiting certificate interoperability

As discussed above, the historical focus on cross certification appears to have been a well-intended but misguided attempt to determine the equivalence of certificates issued in different domains. If we take time to revisit the business need for accreditation of PKIs, we can formulate a more powerful and yet lower cost approach to interoperability.

4.3.1 How should certificates "interoperate"?

Is there a topic in PKI more important and yet more confused than interoperability? A senior finance sector executive captured the uncertainty perfectly:

"[PKI] interoperability is something of a will-o'-the-wisp. You think you understand what people mean by it, and then quickly realize that you don't. In my experience, it's possible when discussing interoperability to be at cross-purposes for all of the time. Interoperability between members of the same PKI is axiomatic. Certificates issued by one bank should be recognizable by another. Interoperability becomes an issue when it is between different PKIs … But this still leaves the basic question of interoperable in respect of what?" [21].

The best place to start thinking about interoperability is to unpack with a functional focus how digital certificates can help with authentication. A fine definition of authentication comes from the APEC eSecurity Task Group: *"The means by which the recipient of a transaction or message can make an assessment as to whether to accept or reject that transaction"* [2]. In the case of digital certificates, from the perspective of the receiver or Relying Party, the central question is really very simple: What information is available, in the certificate chain and elsewhere, to help the receiver decide whether to accept or reject the certificate and hence a digitally signed message?

There are three main things the receiver needs to know about a certificate in order to tell if it is fit for purpose:

1. ***What representations does the certificate make about its Subject?*** Or equivalently, was the certificate intended to be used in the transaction concerned? With Relationship Certificates standing for specific credentials or memberships conferring particular authorizations, each will bear a unique Policy OID indicating its intended applicability and context.
2. ***Is the certificate valid (i.e. not revoked)?*** Note that while revocation status is usually thought of as a question posed in real time, sometimes it will be backdated; that is, we may need to know if the certificate Subject was valid at the time they launched the transaction (see e-conveyancing discussion in subsection 3.1.1).
3. ***Was the certificate issuer acting in compliance with applicable standards and regulations?*** Relevant standards will vary from one domain (or PKI scheme) to another; examples include the Australian government's *Gatekeeper* program, the finance sector's *Identrus* and the more general purpose *WebTrust for CAs*.

All of the information that an application needs in order to accept or reject a certificate could be found in the certificate chain, under the right circumstances. Compared with orthodox PKI which referred vaguely to "chains of trust", we need to be more precise about what certificates issued to CAs represent. If they represent each CA's compliance with standards (like *Webtrust for CAs* or *Identrust*) then when an end user certificate chains back to the root we can be sure that all intermediate CAs are doing what they're supposed to do. And if the end user certificate's Policy OID matches our expected value, then the certificate can be relied upon. For more details, see subsection 4.4.1.

4.3.2 Cross recognition versus cross certification

When transactions cross between jurisdictions or communities of interest, users must be able to determine whether or not to accept a transaction signed using an certificate issued elsewhere. This then is the fundamental issue in electronic authentication, rather

[23] When couched in this way, the certificates issued by a Root CA can be seen recursively as special instances of Relationship Certificates.

than the quite arbitrary question of whether counter parties' certificates happen to be equivalent, as discussed in subsection 1.1.3.

In contrast to cross certification, *cross recognition* is defined as *"an interoperability arrangement in which a relying party in one PKI domain can use authority information in another PKI domain to authenticate a subject in the other PKI domain"* [1].

Users in a community of interest require information and guidance from their community leaders about the fitness for purpose of whichever external certificates can be expected to be received with incoming transactions. With a range of CAs issuing certificates for different uses, it is essential that a Relying Party can tell if an incoming certificate is acceptable for the transaction concerned; ideally their software application should be able to decide on-line and automatically whether to accept or reject a given certificate.

If a CA has been accredited under an external PKI scheme, then the issue boils down to whether or not that accreditation is acceptable to the local community of interest for the intended use of the certificates [26]. This is perhaps the simplest statement of the problem of cross recognition of PKIs.

4.4 How to convey "fitness for purpose"

Where a CA is audited or accredited under a particular scheme, its standing under that scheme should be made available to Relying Parties online. *Webtrust for CAs* does this to some extent by way of a web seal on the CA's site, but this requires out-of-band examination by the Relying Party, at least on occasion. That is, the fact of accreditation is not machine readable. It would be far better for a Relying Party application to be able to recognize programmatically the fact of accreditation.

4.4.1 Rendering CA audits machine readable

A more powerful and interoperable way to represent accreditation is to use a conventional X.509 certificate issued by (or on behalf of) the auditor. In 1999, I proposed an "accreditation based" way to construct PKI, in which "the X.509 certificate issued by an intermediate CA to a user CA is interpreted explicitly as a compliance certificate, directly analogous to the paper certificate issued by a [quality standard] certifier to a compliant organisation" [25]. The basic thrust of this proposition was adopted by the Australian Government PKI program in its *Gatekeeper Accreditation Certificate* CA initiative, which, while not yet operational, is envisaged will:

> *"… issue a digital certificate – the Gatekeeper Accreditation Certificate (GAC) – to each Gatekeeper Accredited CA. Issuance of the GAC would confirm that the CA has satisfied the Australian Government's requirements for Gatekeeper Accreditation. In issuing a GAC, Finance will* ***not*** *be acting as a Root Certification Authority as it does not impose a policy regime on digital certificates issued by subordinate CAs"* (emphasis in original) [4].

(Note the evident reluctance to act as a Root CA, an inhibition that will be considered in more detail in subsection 4.4.2.)

A key advantage of the accreditation based PKI model is that the audit certificate can be parsed and interpreted by entirely conventional X.509 software. The existence of a valid and current certificate chain extending from an end user back to a recognized auditor can be interpreted to mean that the user certificate is fit for purpose as circumscribed by the scope of the audit, and that the certificate was issued by a CA that was, at the time of the last inspection, found to be in compliance with its own policies and procedures as well as any other prescribed standards.

For an illustration, see Figure 4 which depicts a digital signature of a qualified doctor chaining through a Relationship Certificate to an issuing CA, the certificate of which is signed by a Root CA representing a health sector scheme.

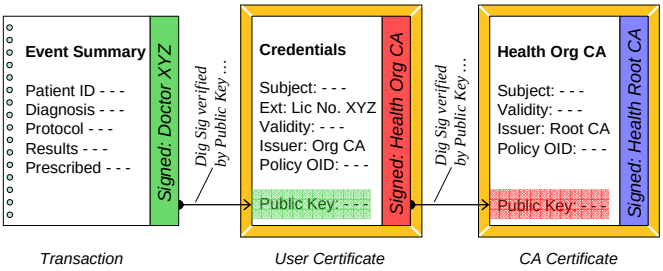


Figure 4: Imputing fitness for purpose from a certificate chain

If we adopt a somewhat fresh interpretation of what it means for various CAs to sign certificates, then the chain in Figure 4 allows fitness for purpose to be imputed as follows.

In this example, the user certificate issued by (or on behalf of) a reputable health organisation represents nothing more and nothing less than the fact that the Subject has a meaningful medical qualification. The Policy OID in that certificate directly represents a transaction domain on which the digital qualification is deemed to be valid. For example, if the health organisation were a medical practitioner registration board, then it could be that its certificates confer the authority to sign e-prescriptions and other transactions under certain legislation (and the Certificate Policy would call out that legislation explicitly and incorporate, probably by reference, all associated rights, responsibilities, terms and conditions). On the other hand, if the health organisation were a hospital, then its certificates might have a more restricted scope of meaning, such as the authority to admit patients for procedures according to a contract between the hospital and a doctor. Similarly, a certificate issued by a Health Maintenance Organisation (HMO) or private health insurer could confer authority to order particular tests electronically. In these cases the Certificate Policy would call out the applicable contract from which the certificate Subject's authority obtains.

So, if the digital signature on a health transaction chains correctly to a user certificate issued by the authoritative Health Organization CA, then the receiver can be assured of the veracity of the provider's credentials. It is straightforward for receiving software that deals with a whole class of healthcare transactions to be configured with the Policy OIDs of all issuers of certificates deemed to be authoritative for those transactions (obviating the need for complex certificate path discovery, as discussed in subsection 1.2.3).

Turning to the Health Organisation CA itself, it has been issued a certificate by a Health Sector Root CA. We interpret the signature of the Root CA as conferring membership of a health sector scheme. For a CA to hold a valid certificate signed by the Root means that the CA has been deemed to have met the scheme rules, and has passed whatever audits are specified by those rules. If the conditions of membership of the scheme are ever breached then the CA's certificate can, as an ultimate sanction, be revoked.

4.4.2 Root CA as "conformity assessment" anchor

Now we can consider re-inventing the role of Root CA. Orthodox formulations of the role and responsibility of Root CAs have historically been confusing (if not confused). It has been difficult to avoid unspecified legal liabilities growing as we move up the "chain of trust" from CA to Root.

But what exactly is it that a Root CA does? Or what should it do? As we saw in the previous subsection, there is a presumption that Root CAs "impose a policy regime on digital certificates issued by subordinate CAs" [4]. At least one PKI scheme – Australia's *Gatekeeper* – is reluctant to have Certificate Policy imposed by the Root CA. Instead, it prefers to allow member CAs to remain entirely autonomous in the way they construct their OID trees.

Operationally of course, a PKI needs a top-most CA that spawns other operational CAs, and provides a "trust anchor" to which certificates can chain. Relying Party software needs a dependable copy of the Root CA Public Key, and when a chain of certificates is established that terminates at a self signed Root CA certificate, it is said that they can be "trusted". In terms of certificate parsing and processing, this much is conventional wisdom and yet the *point* of chaining CAs together has not been obvious, and confusion has reigned over the types of bodies thought most apt to have custody of Root CAs.

The plain English synonym for Root CA certificate, "trust anchor", happens to be suggestive of a precise and powerful new type of role for Root CAs, which derives from existing audit and control structures. Obviously, most PKIs already embody various forms of audit, against standards that vary in rigor from one industry to another. But regardless of the details of a particular audit methodology, it is possible to gauge whether or not the audit has been conducted in a manner that is suitable for its environment. In the field of technical inspection or "conformity assessment", the pivotal question of 'who audits the auditors?' has long been addressed by a nested system of international inspection and accreditation standards. Across a very wide range of technical domains – from traditional materials testing to independent software validation – the standard ISO 17025:1999 *General Requirements for the Competence of Calibration and Testing Laboratories* [16] has been applied in the accreditation of inspection bodies.[24] The outcome of such accreditation is an assertion that a given inspection body is independent and competent to carry out audits in its field of expertise, no matter what that field may be, and regardless of the peculiarities of the standards that apply to that field. This outcome of auditor accreditation coupled with the fact that national ISO 17025 accreditation bodies have established multilateral *Mutual Recognition Arrangements* (MRAs) enables a new way of achieving "interoperability" between PKIs, as we shall soon see.

Multilateral MRAs are nowadays administered by overarching "co-operations" to which numerous national accreditation bodies are signatories. Examples include the Asia Pacific Laboratory Accreditation Cooperation (APLAC) and the International Laboratory Accreditation Cooperation (ILAC).[25] Just as there are standards like ISO 17025 for 'auditing the auditors', recently another level has been added to govern accreditation bodies. For example, ISO 17011:2004 *General requirements for accreditation bodies accrediting conformity assessment bodies* is used by APLAC as the basis for its MRA. Accreditation bodies can join the MRA under a number of headings according to the broad focus of their activities, such as *inspection*, *calibration*, *testing*, or *reference materials production*.

Note that the accreditation of inspection bodies can be fine tuned to meet particular needs. When a certain field demands special skills and qualifications, ISO 17025 needs to be interpreted in respect of the type of inspection at hand and any special techniques involved. When a specialist field has its own conformity requirements, as might be set by local standards, accreditation bodies can produce *supplementary requirements* for accreditation of particular types of inspection bodies, to augment the general requirements of ISO 17025. Thus for example, Australia's National Association of Testing Authorities publishes detailed Accreditation Requirements, all based on ISO 17025, but fine tuned to a wide range of domains, including information technology.[26]

In relation to PKI governance, it is especially noteworthy that the information security community has already successfully applied ISO 17025 accreditation to overseeing the ISO 15408 Common Criteria evaluation scheme. The Common Criteria arrangement [11] requires security evaluators to be accredited in accordance with ISO 17025.

4.4.3 Scaleable global PKI from an ISO 17025 MRA

One reason that the Common Criteria scheme as been uniformly adopted with relative ease across 25 countries[27] is the existence of MRAs. The practical result of an MRA is that assessments done by accredited inspection bodies in one country can be readily accepted by interested parties in other jurisdictions. International trade in particular benefits from MRAs because goods that are subject to safety and type testing, such as electrical equipment, can be evaluated once in the country where they're made prior to export, and subsequently accepted by a large number of importers

[24] In the first expression of the accreditation-based PKI model [25], I suggested that quality management systems were a suitable model for what CA auditors do, and that the peak standard ISO/IEC Guide 62 *General requirements for bodies operating assessment and certification/registration of quality systems* could be applied. More recent research indicates that ISO 17025 (formerly known as ISO/IEC Guide 65) is a better fit for PKI auditors because it tests the competence of auditors and is generally more apt for technically demanding fields. It should be noted that a number of superficially similar accreditation standards exist including ISO/IEC 17020:1998 *General criteria for the operation of various types of bodies performing inspection*. To build a working PKI using these standards, a technical choice of high order standard would be made by expert standards bodies.

[25] ILAC members include American Association for Laboratory Accreditation (A2LA; http://www.a2la.org), ACLASS Accreditation Services (http://www.aclasscorp.com), International Accreditation Service (IAS; http://www.iasonline.org), the United Kingdom Accreditation Service (UKAS; http://www.ukas.com) and Australia's National Association of Testing Authorities (NATA; http://www.nata.asn.au).

[26] See http://tinyurl.com/2yevj8.

[27] See List of Common Criteria Recognition Arrangement members at http://www.commoncriteriaportal.org/public/consumer/index.php?menu=4 (accessed 8 Jan 2008).

around the world without the need for repeat testing. APLAC describes this as the "free-trade goal of 'tested/inspected once, accepted everywhere'".[28]

The ability to accept and rely upon a specialized technical audit done in another country is surely the key to international PKI, if it is accepted that different digital certificates can mean different things, as argued throughout the earlier parts of this paper. ISO-17025 MRAs represent a hugely important asset in this regard because they can accommodate a flexible range of audit matters. Member accreditation bodies can be empowered under an MRA to implement new supplementary accreditation guidelines within a broadly defined scope such as "inspection",[29] and have the outcomes of their accreditations recognized in other countries, without them having to review in detail the substance of those guidelines. In turn, the outputs of organisations that have passed inspection under those new guidelines can be accepted across borders in other participating jurisdictions. Therefore, cross-border PKI could be constructed as follows.

The new PKI model treats digital certificates as the products of a special class of manufacturers, namely, RAs and CAs working in concert. The model also rests on the fact that from one jurisdiction (or industry) to another, reasonable decisions have already been made and broadly accepted regarding the appropriate standards that should govern RAs and CAs (including for example X.509, the PKIX series, RFC 3647, or the detailed *Identrus* technical requirements). Some jurisdictions and industries have gone one step further to select or design for themselves an appropriate PKI conformity assessment program. Examples include *Webtrust for CAs* (which was adopted by Microsoft as a pre-condition for being included in Internet Explorer's list of *Trusted Certification Root Certification Authorities*), *Gatekeeper*, *Identrus* accreditation (which has come to be recognized by *Gatekeeper*[30]), or various countries' local regulations implemented under the European *community framework for electronic signatures*[31].

Referring to the three preconditions for being able to accept a certificate cross-border, as defined in subsection 4.3.1, the proposed PKI model will deliver two[32] of them:

1. the conformity of the certificate issuer with agreed standards would be assessed by an approved PKI auditor, and the results of that assessment conveyed to the receiver, and
2. the intended purpose of the certificate would be precisely specified by its Policy OID (the uniqueness of which on the relevant domain would be enforced by the audit).

As described in subsection 4.4, to convey the results of the conformity assessment in this proposed PKI, special digital certificates will be issued by (or on behalf of) PKI auditors to each approved CA. The meaning of each these certificates is simply (but *precisely*) that the Subject has passed an audit according to detailed procedures and standards that would be uniquely indicated by a Policy OID. Different audit regimes and different auditors would map onto different OIDs.

Now, it is just as important that the status of the PKI auditors also be conveyed to receivers, and for that purpose, a special digital certificate will similarly be issued by (or on behalf of) an accreditation body to each accredited PKI auditor. In accordance with the regular provisions of ISO 17025, accreditation bodies would be chiefly concerned with the independence and the competence of PKI auditors. It may be the case that additional considerations, specific to PKI, are needed to be applied to make this determination, but as discussed, ISO 17025 accommodates this nicely. We should expect supplementary guidelines to be developed during the course of establishing the PKI model.

The chain of digital certificates corresponding to the different levels of conformity assessment could terminate at the accreditation body, with a self signed certificate. And yet the existence of international accreditation co-operations and MRAs presents the tantalizing prospect of cross border PKI resulting almost as a by-product of existing arrangements, with a conceptually simple switch from paper-based audit and accreditation certificates to digital certificates representing the same thing.

To operationalize the PKI, national accreditation bodies would act as jurisdictional Root CAs. It would not be necessary for these bodies to actually build and operate the CAs themselves; rather, they could outsource certificate production and concern themselves only with an RA. When looking at the potential legal liability of an accreditation body taking on the role of digital certificate issuer, we should be reminded that their proposed role in PKI is the identical to their role in traditional conformity assessment schemes; that is, they 'audit the auditors'. As such, their potential liability is well understood in industry, and tends to be well contained. If the digital certificates issued by accreditation bodies in this proposal are understood to mean nothing more and nothing less than the fact that the certificate Subject is an accredited PKI auditor, then the fact that the accreditation body is acting as a "Root CA" shouldn't introduce any new liabilities.

Figure 5 illustrates the proposed ISO 17025 MRA based PKI. The grey boxes represent the chief participants in the PKI, from CA through to international cooperation association (note that every one of these players already exists). The nested boxes show the scope or community of each participant: the smallest boxes are for the members served by each CA, intermediate for the CAs inspected by each auditor, and largest for the auditors inspected by each accreditation body (note how the communities are nested at the levels of country, auditor and CA). The block arrows indicate the frame-work that governs the work of each participant. At the top level, a working assumption is that of the existing types of MRA, an appropriate heading for PKI would be "inspection" (as opposed to *calibration* or *testing*).

---

[28] See http://www.aplac.org/aplac_mra.html.

[29] See the various scopes of recognition for each of the APLAC MRA signatories at http://www.aplac.org/aplac_mra.html (accessed 8 Jan 2008).

[30] See http://www.dbcde.gov.au/Article/0,,0_4-2_4008-4_116523,00.html (accessed 8 Jan 2008).

[31] See http://europa.eu/scadplus/leg/en/lvb/l24118.htm (accessed 8 Jan 2008).

[32] The third precondition had to do with the certificate not being revoked; the ability to perform a CRL or OCSP check is taken for granted here.

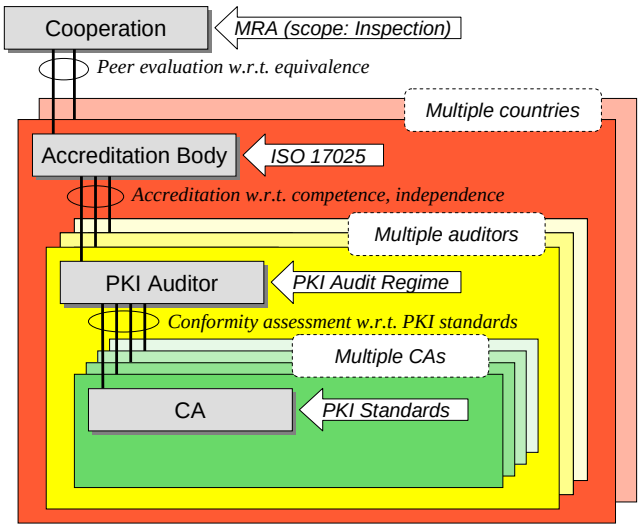


Figure 5: A PKI based on ISO 17025 Mutual Recognition

Note that the proposed PKI can be grown from the bottom up. It is not necessary for an international cooperation to come on board right away; in the interim it would be practical to have local self-signed trust anchors for each of the jurisdictional accreditation bodies. Whenever a new body joins an MRA and has its processes approved, it follows that all CAs within its jurisdiction automatically enter the fold of the international scheme. Thanks to their maturity and long established authority, having accreditation bodies in the PKI solves the hoary problem of infinite regress; that is, how far back do you go before you find a CA you can "trust"? The answer is you stop at a national body, or ultimately at an international cooperation like APLAC or ILAC. Most important of all, because there are existing protocols and agreements by which national accreditation bodies recognize and work with one another, this approach to PKI provides a natural and robust means for cross-border recognition of digital certificates.

In closing this account of an international PKI, let us remember what it is that a certificate chain can represent. If the receiver of a digital certificate knows what end user Policy OID is appropriate to the transaction at hand, and if the receiver's software has a trusted copy of the root key, then any certificate featuring that OID which chains to that root key can be taken to be fit for purpose, no matter which CA issued it. Certificate chains in this international PKI scheme would each embody an unambiguous cascade of dependable assertions:

- The end user was vouched for with reference to a certain CP by an RA authoritative in a given community, and was issued a certificate with a corresponding unique Policy OID, produced by a named CA.
- The CA was approved with reference to agreed standards, CPS etc. by a named PKI auditor, which issued (or had issued on its behalf) a digital certificate to the CA.
- The auditor was approved with reference to ISO 17025 by a named accreditation body, which issued (or had issued on its behalf) a digital certificate to the auditor.
- The accreditation body was approved with reference to a Mutual Recognition Agreement by a named international cooperation, which issued (or had issued on its behalf) a digital certificate to the accreditation body.

The certificate chain conveys the membership of all participants in the scheme as anchored by the root key controlled by the top level cooperation. And yet certificates that chain through different auditors and accreditation bodies are entirely autonomous. Neither the root nor the intermediate accreditation bodies would impose any arbitrary policies on the conduct of end user CAs and communities of interest. The uniqueness of the end user certificate Policy OIDs plus the separation of powers of auditors and accreditation bodies means that the one PKI could embrace any number of diverse communities, and could accommodate existing closed PKI programs like *Identrus* or *WebTrust for CAs* so long as their methods are transparent and compatible with ISO 17025.

5. CONCLUSIONS

There is no intrinsic reason that PKI should be as complex as it has. Plenty of complicated physical principles have been successfully engineered and deployed as commercial technologies, such as magnetic stripe cards. PKI historically has been unwittingly burdened by well intended metaphors, such as that of the passport, that associated it with a vague ideal – universal identification of strangers online – which turned out to be hugely complicated and not even necessary. Meanwhile, smaller scale, closed PKIs have prospered in support of special purpose applications. This fact can now be appreciated as the natural state of affairs, resonant with the modern view of identity plurality.

We should build on the deeper lessons of successful closed PKIs, to regard certificates as evincing not absolute identity but rather any number of relationships, with special meaning in the contexts in which the certificates were issued and intended to be used. This simple change of aspect could herald a true paradigm shift, rendering digital certificates and their production much more mundane. Radical improvements would result on several fronts. Firstly, the practical application of PKI would be greatly simplified by breaking the nexus between authentication and authorization, for it allows X.509 formatted Relationship Certificates to stand alone in most transactions. Secondly, by localizing RA functions and more effectively decoupling certificate production, we could operate back-end CAs along the same lines as security printers, with vastly simpler legal arrangements than seen in orthodox PKI. And finally, existing nested frameworks for conformity assessment and accreditation provide the ready means for cross-border recognition of certificates, knitting together today's heterogeneous PKI applications, policies and audits into the one international Public Key Superstructure.

6. ACKNOWLEDGMENTS

The concept of "Relationship Certificates" was originally researched and developed by Lockstep Consulting under contract to the former Australian Department of Finance and Administration, represented by the Australian Government Information Management Office (AGIMO). The author gratefully acknowledges the permission of AGIMO to reproduce aspects of that work.

7. REFERENCES

[1] APEC Telecommunications Working Group – E-Authentication Task Group, Achieving PKI Interoperability (1999).

[2] APEC Telecommunications Working Group, Electronic Authentication: Issues Relating to its Selection and Use ISBN 981-04-7690-6 (2002).

[3] Australian Department of Health and Ageing, Electronic Signatures for Prescribing and Dispensing, eHealth Branch (2006) http://www.msia.com.au/esig_prescript_document.pdf (accessed 31 Jan 2008).

[4] Australian Government Information Management Office (AGIMO), Gatekeeper PKI Framework Cross Recognition Policy, Department of Finance and Deregulation (2008) http://www.gatekeeper.gov.au/__data/assets/file/0004/52276/Cross_Recognition_Policy.rtf (accessed 8 Jan 2008).

[5] Australian Government Information Management Office (AGIMO), Relationship Certificate Guidebook, Department of Finance and Administration (2006) http://www.agimo.gov.au/__data/assets/pdf_file/0016/52252/Relationship_Guidebook.pdf (accessed 19 Nov 2007).

[6] Barnes, R. & Kent, S. An Infrastructure to Support Secure Internet Routing, IETF Secure Inter-Domain Routing Working Group (2007) http://tools.ietf.org/id/draft-ietf-sidr-arch-00.txt (accessed 31 Jan 2008).

[7] Barnett, S. A pilot project: Sending encrypted specialist letters to GPs, *Health Openware Foundation Argus Forum*, (Canberra, Australia, 2004).

[8] Brewer, J. & Wilson, S., Smartcards and PKI at Medicare Australia, Australian Electrical & Electronic Manufacturers Association ICT Forums (2006) http://www.aeema.asn.au/ArticleDocuments/41/Smartcards%20and%20PKI%20at%20Medicare%20Australia%20-%2014Feb06.pdf (accessed 31 Jan 2008).

[9] Burr, W. Electronic Authentication in the U.S. Federal Government, *Asia PKI Forum*, Tokyo (2005) http://www.asia-pkiforum.org/feb_tokyo/NIST_Burr.pdf (accessed 23 Nov 2007).

[10] Cameron, C. The Laws of Identity, Microsoft Corporation (2005) http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf (accessed 31 Jan 2008).

[11] Common Criteria, Arrangement on the Recognition of Common Criteria Certificates in the field of Information Technology Security (2000) http://www.commoncriteriaportal.org/public/files/cc-recarrange.pdf

[12] Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., & Ylnen, T. SPKI Certificate Theory RFC 2693, IETF SPKI Working Group (1999) ftp://ftp.isi.edu/in-notes/rfc2693.txt (accessed 31 Jan 2008).

[13] Ellison, C. & Schneier, B. Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure *Computer Security Journal 16, 1* (2000).

[14] Forey, M., Cheque Printer Accreditation Scheme, *Xplor Document Management Conference* (Anaheim, California, USA, 2002).

[15] Freeman, R., Trust Services – A Market Appraisal, Mack Interact (2002) http://www.tscheme.org/library/tSi0156_01%20TSP%20market%20status%20report.pdf (accessed 19 Nov 2007).

[16] International Organization for Standardization, General requirements for the competence of testing and calibration laboratories, ISO/IEC 17025:1999.

[17] Jøsang, A. & Pope, S., User Centric Identity Management, *AusCERT Security Conference 2005* (Gold Coast, Australia) (2005).

[18] Kent, K. Global PKI: Status, Trends and the Future *Taipei International PKI Conference* (Taipei, September 2005) http://www.pki.org.tw/pkiforum2005/d_file/01_Stephen%20Kent.pdf (accessed 23 Nov 2007).

[19] National Office for the Information Economy, Liability and Other Legal Issues in the use of PKI Digital Certificates, Australian Department of Communications, Information Technology and the Arts (2000).

[20] Nazareth, S. & Smith, S. W. Using SPKI/SDSI for Distributed Maintenance of Attribute Release Policies in Shibboleth, Computer Science Technical Report TR2004-485, Dartmouth University (2004) http://www.ists.dartmouth.edu/library/spk1004.pdf (accessed 31 Jan 2008).

[21] Smith, P., Trust and Digital Certificates, *16th Payment Systems International Conference* (Belgium, 2000).

[22] Sneddon, M. Legal Liability and e-transactions, Australian Department of Communications, Information Technology and the Arts (2000). http://www.egov.vic.gov.au/pdfs/publication_utz1508.pdf (accessed 23 Nov 2007).

[23] Standards Australia, Strategies for the implementation of a Public Key Authentication Framework (PKAF) in Australia, Miscellaneous Publication MP 75 (1996).

[24] Victorian Office of the Chief Information Officer, Land Exchange (LX) Case Study, Government of Victoria (2004) http://www.egov.vic.gov.au/pdfs/Land%20Exchange-shh-30April-v1.0-CIO.pdf (accessed 21 Nov 2007).

[25] Wilson, S. New models for the management of public key infrastructure and root certification authorities. In *Proceedings of Information Security Management & Small Systems Security (IFIP WG 11. 1/2)* (Amsterdam, Sept 30 - Oct 1, 1999). Kluwer, Deventer, The Netherlands, 1999, 221-230.

[26] Wilson, S. Leveraging external accreditation to achieve PKI cross-recognition, *Australian Attorney General's Privacy and Security in the Information Age Conference* (Melbourne, Australia, 16-17 Aug 2001) http://www.ag.gov.au/www/agd/agd.nsf/Page/Privacy_PrivacyandSecurityintheInformationAgeConferencePapers (accessed 31 Jan 2008).

[27] Wilson, S. Guidelines on how to determine Return on Investment in PKI, OASIS PKI Education Sub-committee, V 1.4 (2005) http://idtrust.xml.org/guidelines-how-determine-return-investment-pki (accessed 14 Jan 2008).

[28] Winn, J. K. The Emperor's New Clothes: The Shocking Truth About Digital Signatures and Internet Commerce, *37 Idaho L. Rev. 353* (2001)

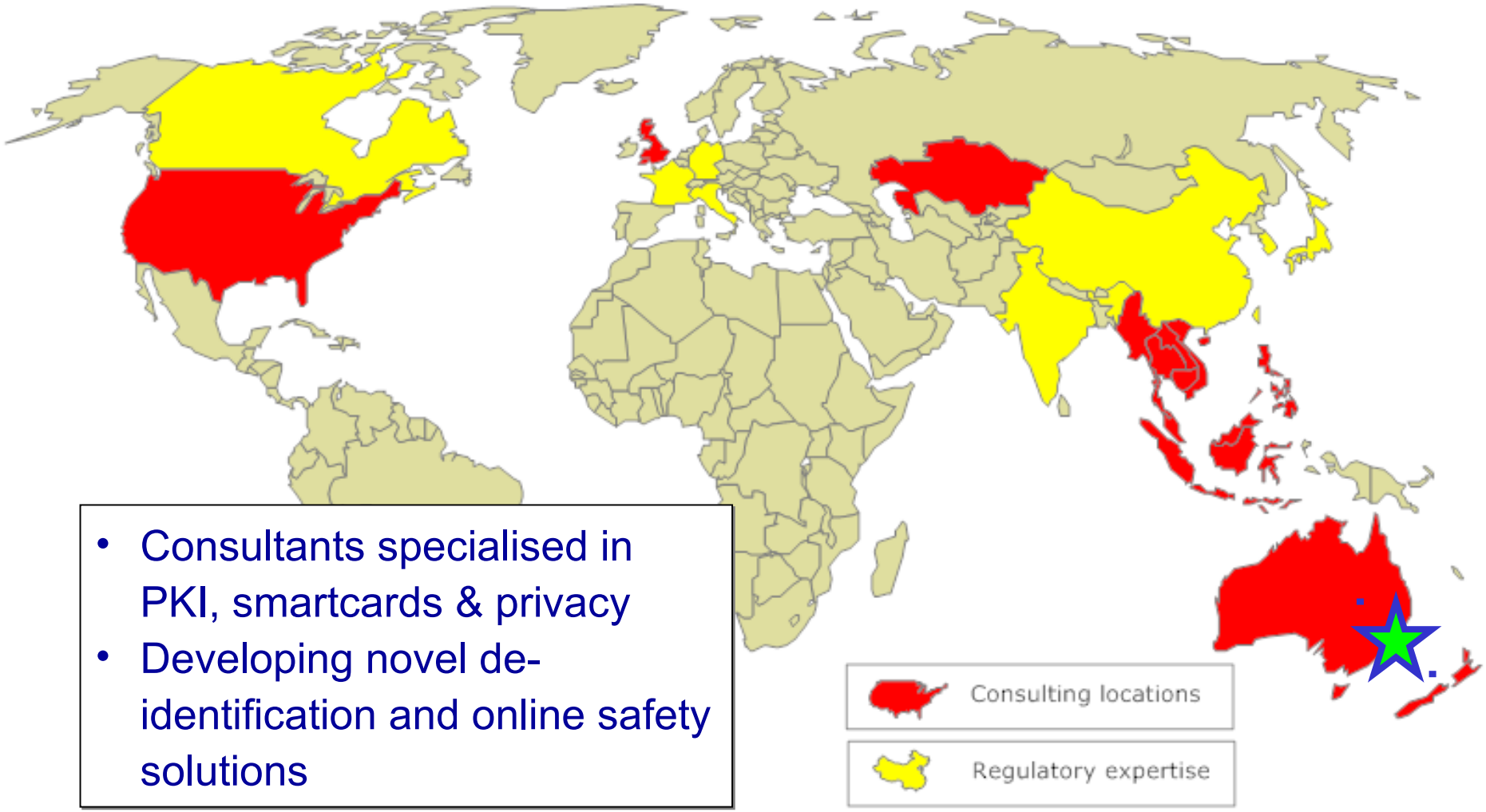
Public Key Superstructure

It's PKI Jim, but not as we know it!

7th Annual "IDtrust" Symposium
5 March 2008, Gaithersburg MD, USA

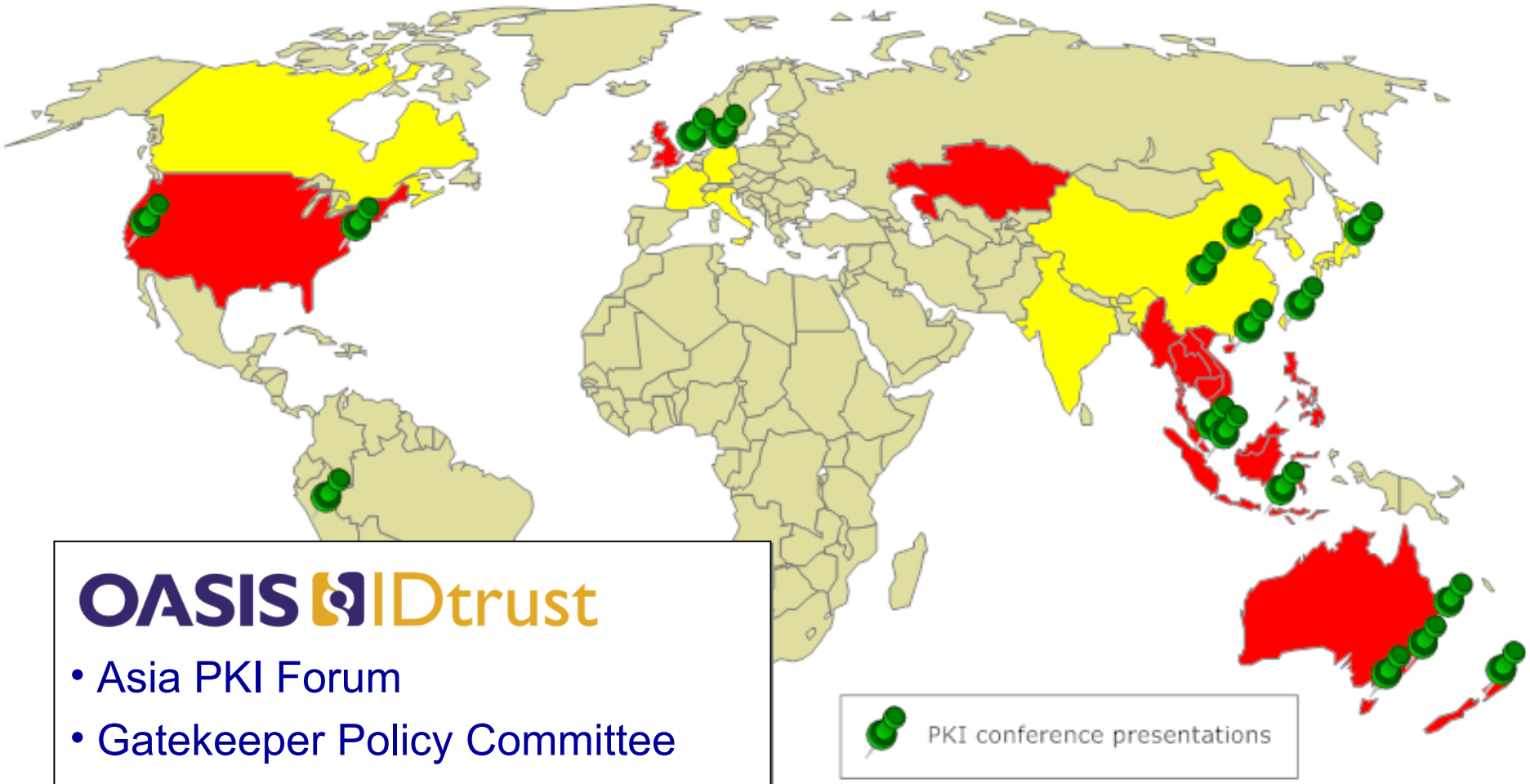
Stephen Wilson
Lockstep Consulting Pty Ltd

About Lockstep



- Consultants specialised in PKI, smartcards & privacy
- Developing novel de-identification and online safety solutions

About Lockstep



OASIS IDtrust

- Asia PKI Forum
- Gatekeeper Policy Committee
- Aust. Law Reform Commission

Historical PKI experience

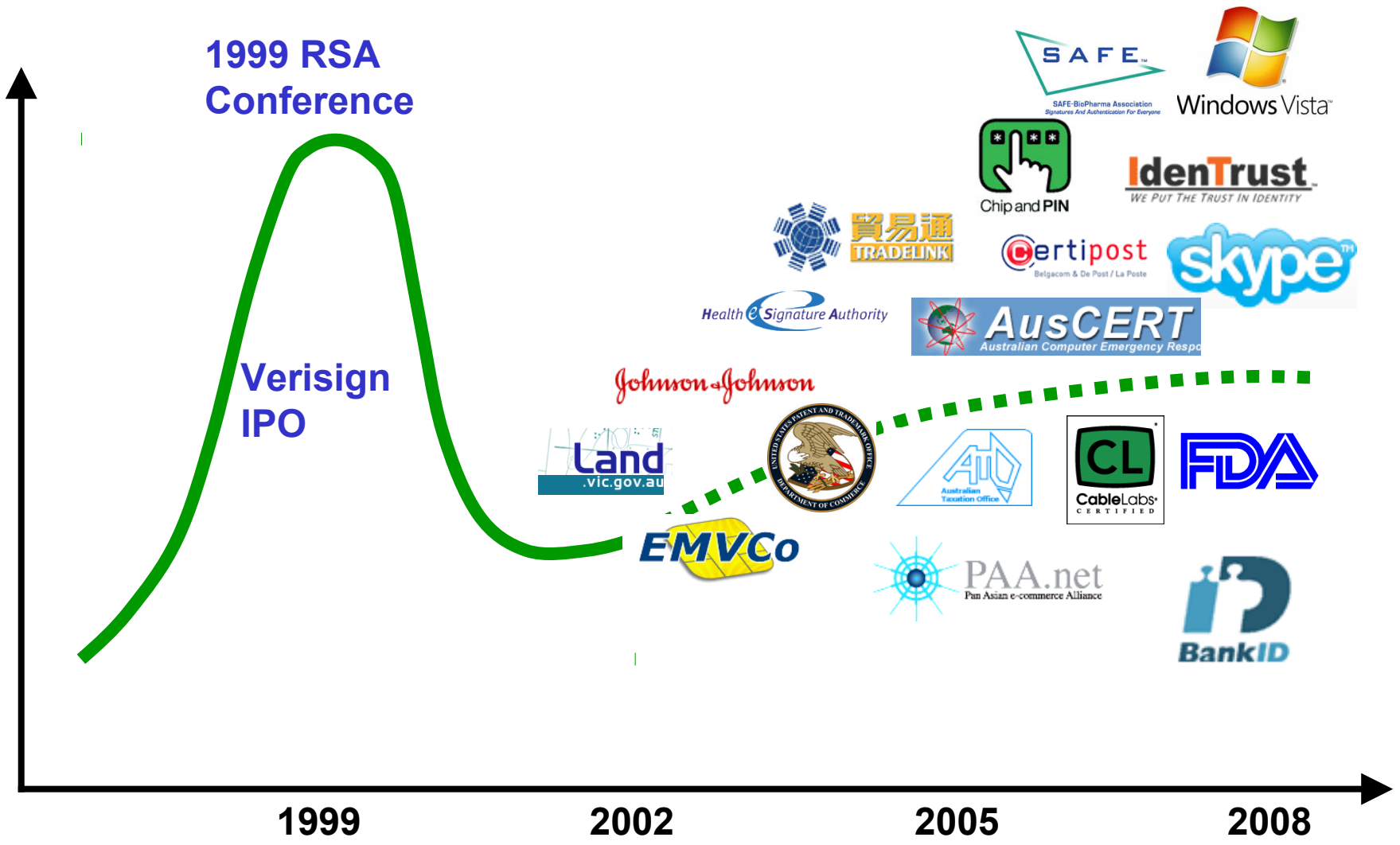
The passport metaphor

- Non-descript applications
  - impossible for CAs to manage risk
- Stranger-to-stranger e-business
  - “It’s good to trust but it’s better not to”
- Novel TTP business models
  - Imposed incredible CPSs upon users
- Notion of a single identity
  - “Interoperability” = cross certification

“Cross-certification and policy mapping has been a rat hole that has sucked up vast amounts of energy better spent elsewhere”

Anonymous, Feb 2008

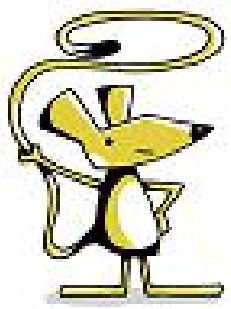
PKI thickets

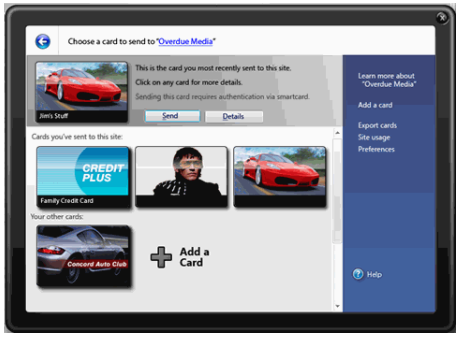


PKI in practice

- Works best in closed communities
  - Automates transactions *in context*
  - This is a Good Thing
- Embedded keys & certificates
- Fits with *identity plurality*

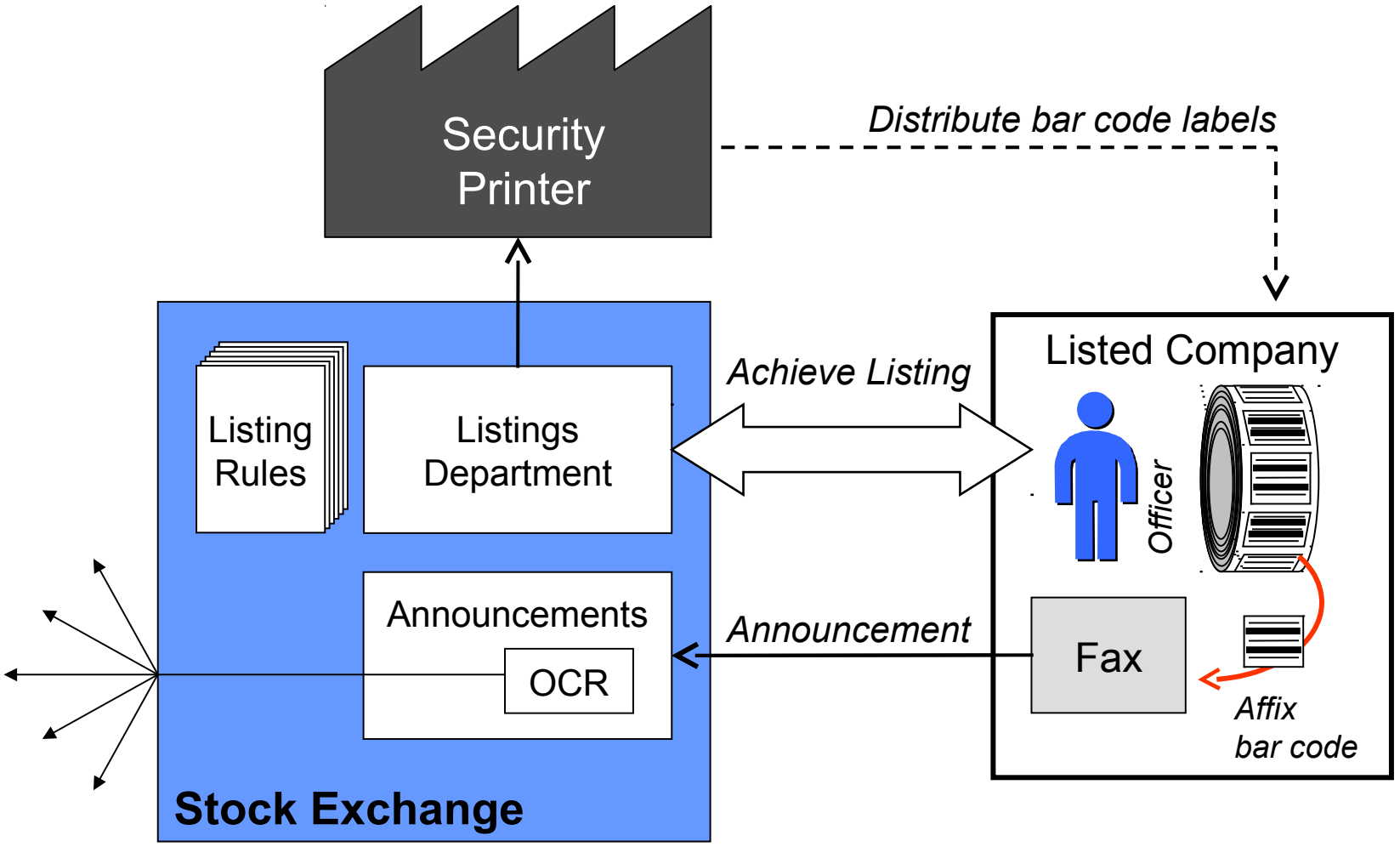
OpenID



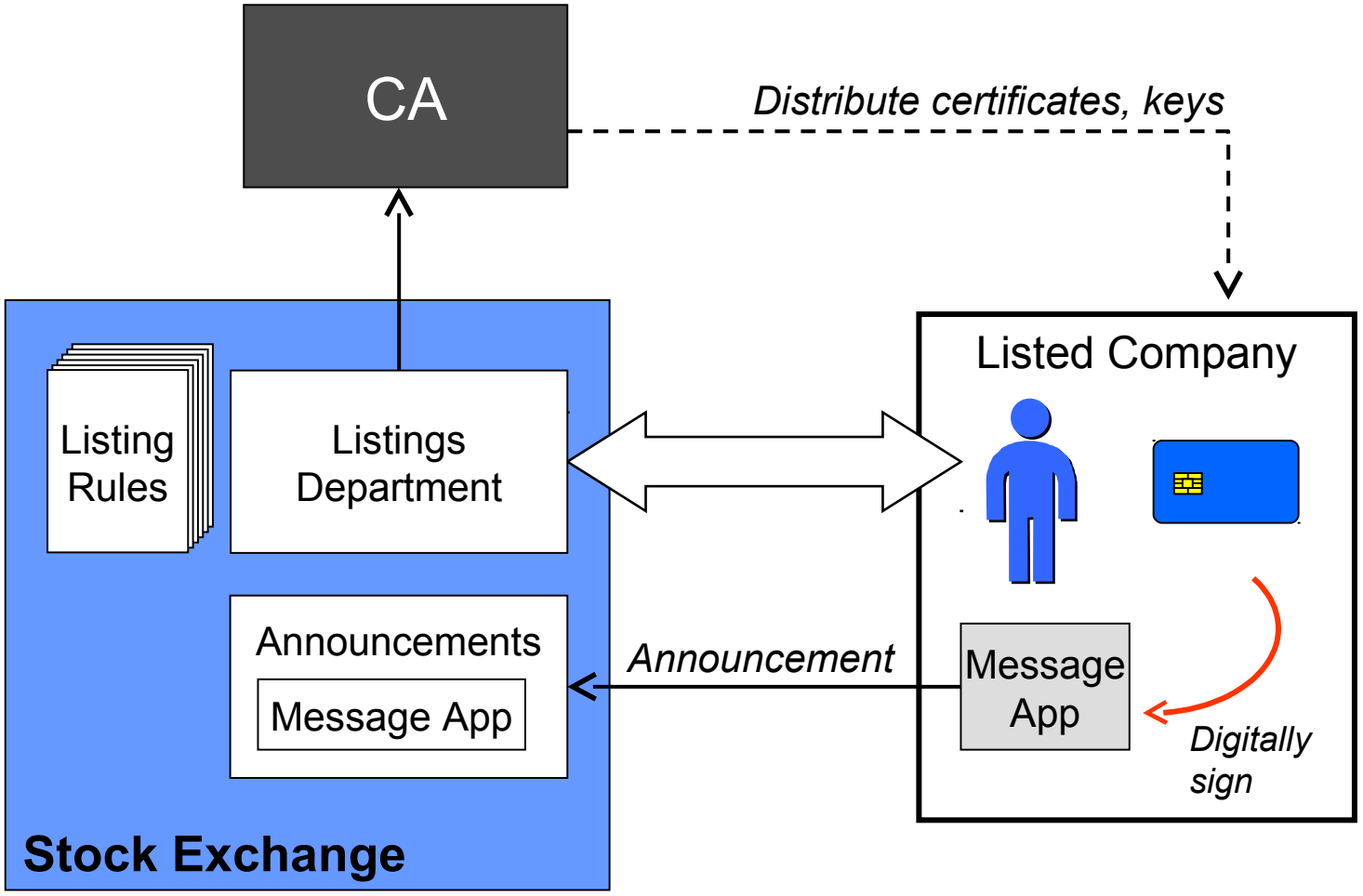


PK Superstructure

CA as *Security Printer*



CA as *Security Printer*

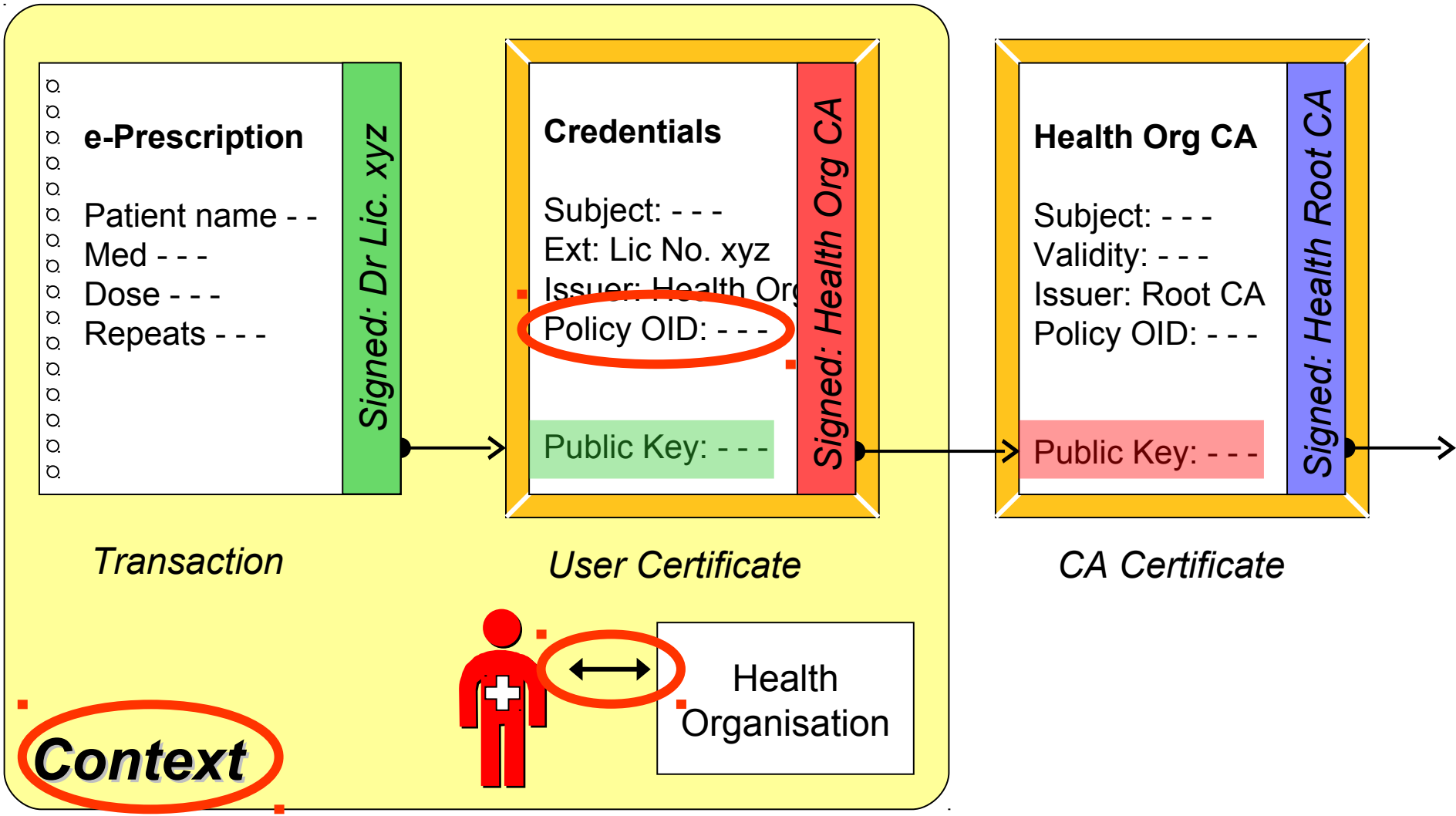


Security printer implications

- Decouples registration from production
- Manages risks associated with registration & production separately

- No contract between Subscriber & CA
- No exposure of CPS to Subscriber
- Easier to novate CA service providers
- Accreditation not affected by new Policies

“Relationship Certificates”



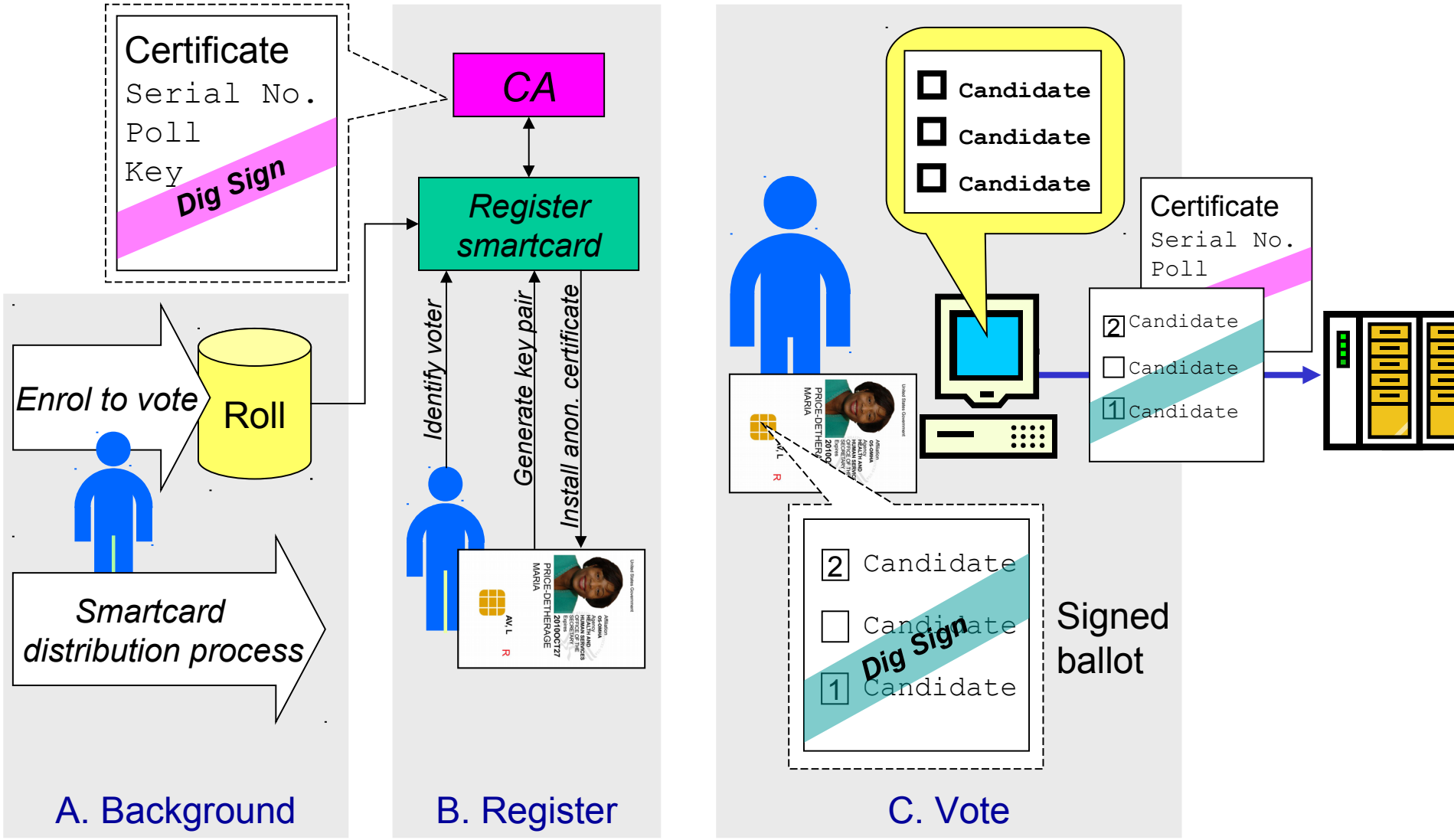
“Relationship Certificates”

- Form of “Authorization PKI”
- Kill the holy cow of authentication being primary over authorization

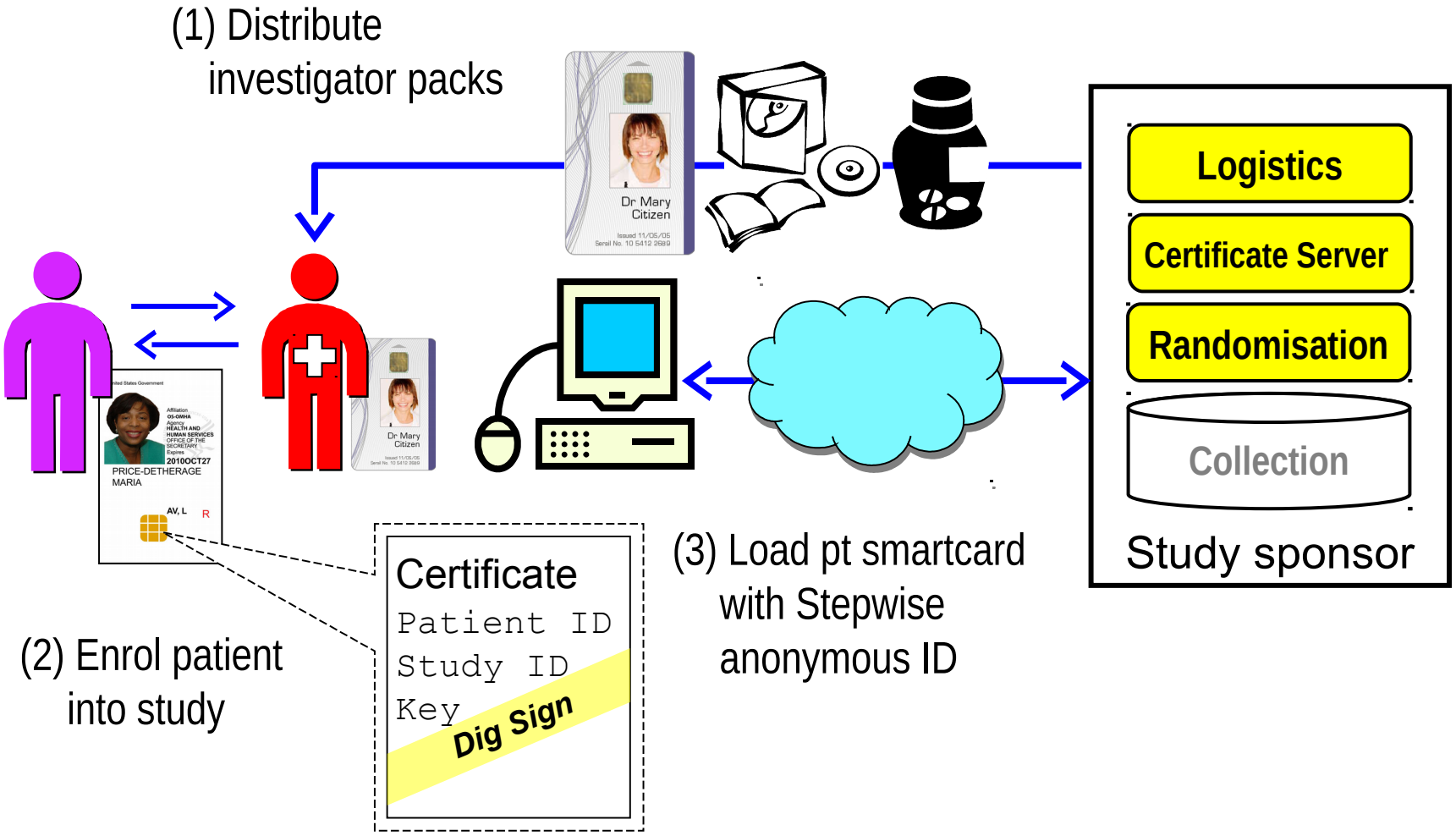
- Preserves X.509 formats, software

- *Not SPKI*: no ‘primary’ ID certificate
- *Not Attribute Certs*: we can *sign* with cert

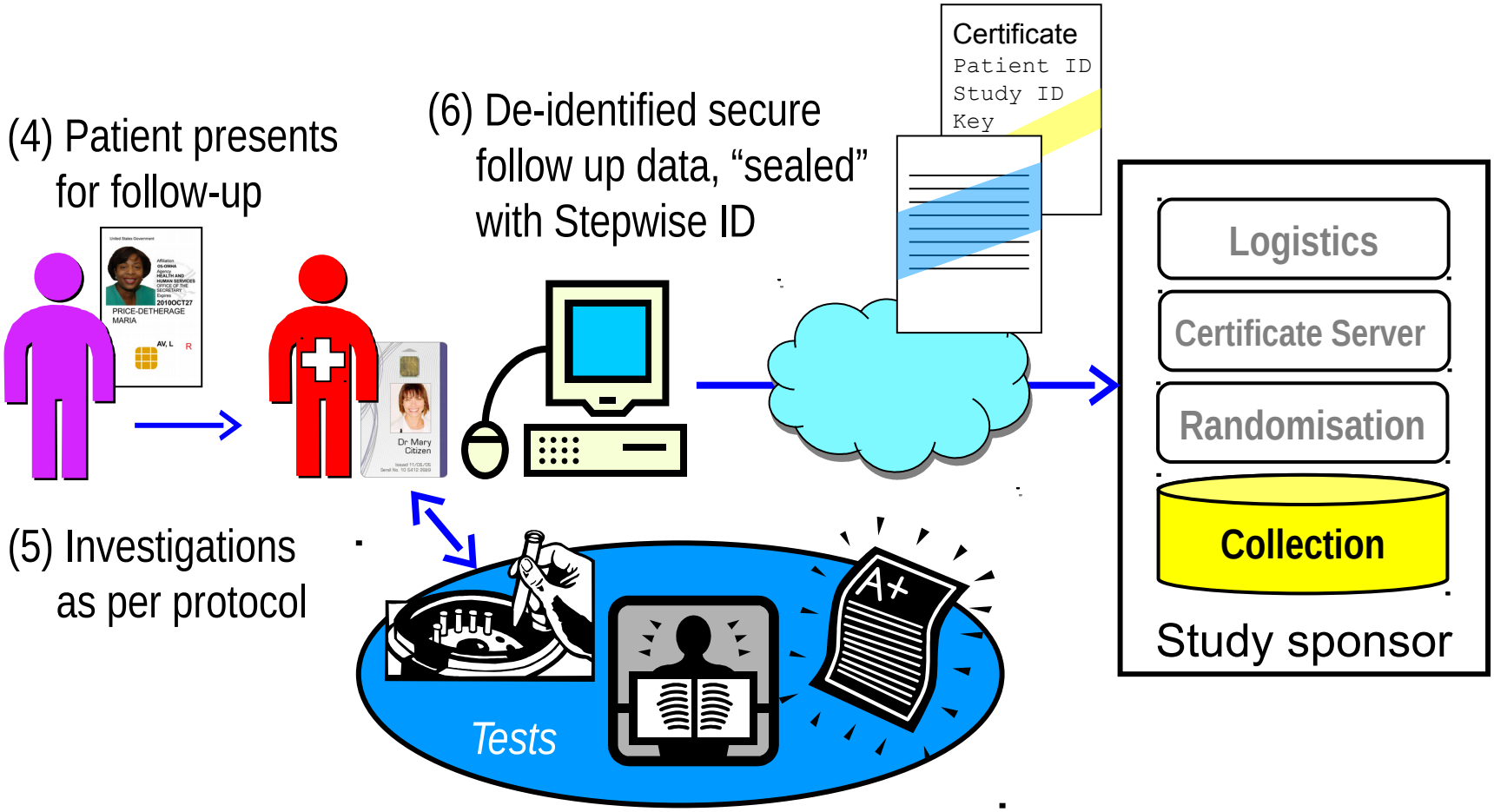
Lockstep anonymous e-voting



Lockstep clinical study privacy



Lockstep clinical study privacy



Discussion

See also www.lockstep.com.au/technologies

OASIS IDtrust

swilson@lockstep.com.au

Audit and backup procedures for Hardware Security Modules

Túlio Cicero Salvaro de Souza
LabSEC – UFSC*
Florianópolis – SC – Brasil
salvaro@inf.ufsc.br

Jean Everson Martina†
Computer Laboratory - University of Cambridge
Cambridge – United Kingdom
Jean.Martina@cl.cam.ac.uk

Ricardo Felipe Custódio
LabSEC – UFSC*
Florianópolis – SC – Brasil
custodio@inf.ufsc.br

ABSTRACT

Hardware Security Modules (HSMs) are an useful tool to deploy public key infrastructure (PKI) and its applications. This paper presents necessary procedures and protocols to perform backup and audit in such devices when deployed in PKIs. These protocols were evaluated in an implementation of a real HSM, enabling it to perform secure backups and to provide an audit trail, two important considerations for a safe PKI operation. It also introduces a ceremony procedure to support the operation of such HSMs in a PKI environment.

Categories and Subject Descriptors

C.2 [**Computer communication network**]: Miscellaneous; D.4.6 [**Security and protection**]: Cryptographic controls; E.3 [**Data Encryption**]: Public key cryptosystems

Keywords

Key Management, Public Key Infrastructure, Embedded Cryptographic Hardware, Hardware Security Module, Key Life-cycle, PKI Ceremony

1. INTRODUCTION

Securely managing keys is one of the most important and resource consuming tasks required to guarantee the security on a public key cryptosystem. This is due to a close relationship between security and the proper management of private keys. A public key cryptosystem can be considered secure as long as the private keys are secured. Taking this as a premise, it should be guaranteed that a (private) key is strictly secure during all events in its life cycle. This goal can be achieved by designing systems to securely create, manage and destroy (private) keys, maintaining an audit trail of every operation which was done during their existence. Such systems are known as Hardware Security Modules (HSMs).

HSMs are specialised tamper-proof devices in which cryptographic functions and embedded software have been built to properly manage keys and control their life cycles. They are designed in such a way that if an unauthorised attempt to access them is made, this is considered an attempt to tamper and all critical internal parameters and keys are destroyed.

Although very common in the banking industry, HSMs are also desirable in PKI, but not always implemented. As shown in Table 1, their common usage in the banking industry leads to specialisation of the HSMs to perform tasks such as PIN calculations or payment protocols, that are suitable in such industry.

In a PKI environment, the required characteristics are different from those for banks. In a PKI HSM, it is necessary to establish who can configure, who can access the keys and who can audit the system. For a PKI HSM, it is desirable to have at least three different roles for users.

It is also recommended not to establish these roles for single users since they are easily corruptible, but also to use triggered group mechanisms, such as defined by Shamir[17] and Blakley [2].

Table 1: Comparison between Bank and PKI HSMs

Bank HSMs	PKI HSMs
PIN Calculation	Strong authentication
Role based authentication	Identity based authentication
Dual key entry	Strict key life-cycle control
Payment protocols	Fully auditable operation
Cryptographic speed	Triggered group mechanisms

Normally, it is established groups of administrators (also known as security officers), operators (also known as users), and auditors, who can track all operations done by the previous groups. The administrators are responsible for creating auditors, operators and the keys managed by the latter. The operators are responsible for releasing the keys for use in an application. The auditors are responsible for analysing the hardware's logs, which register everything that happens inside the HSM - for instance, when a key is used. Responsi-

*Computer Security Laboratory at Federal University of Santa Catarina.

†Supported by CAPES Foundation/Brazil on grant #4226-05-4

IDtrust '08 March 4-6, 2008, Gaithersburg, MD

bility and trust in the keys generated, used and destroyed in HSMs are anchored in these three groups. Moreover, every person who is designated as a member of a group should be carefully selected and all tasks executed by him be registered by automated devices in the PKI context.

Much has been written about HSMs, especially concerning administrators and operators. However, there is no detailed information about audit schemes or backup procedures in HSMs. All developments in this field were done by companies[9, 14, 10, 5, 11], which never published all their internal mechanisms due to industrial secrets concerns and users, and people outside these companies, have no detailed information about their internal protocols.

In this context, "backup" is a way of making a copy of the internal state of the HSM. With a backup, one can prepare new hardware and put it to work as a fully functional copy of the previous HSM. Additionally, the HSMs are products produced by specialised companies and are given some evaluation tests, as described by FIPS 140-2[8].

The HSMs are very expensive devices to use in places such as universities and research centres. There are many of these institutions that would like to deploy a PKI for internal use. Due to the high cost of the HSMs, most of these institutions decide not to use them. Keys are kept in the memory of the host machine which runs the certificate management system. This is a security concern.

Recently, the National Education and Research Network (RNP), a Brazilian social organisation, has created a working group called GT ICPEDU, the main aim of which is to study and develop hardware and software to deploy PKIs for its members. One of its projects was to design an HSM. The hardware architecture of this HSM was conceived by the GT ICPEDU and built by a local company. The software implementing the full keys' life cycle protocol designed for use in PKI environments, called OpenHSM, was the product of an undergraduate final work [18] and a master's thesis [12]. The keys' life cycle protocol passed through several stages of development and improvement. Part of its final version was described by Martina [13]. Two important parts of the protocol, which have not been described yet, are presented by this paper: backup procedures and an audit control mechanism.

This paper considers in section 2, why auditors and backup are important issues in the use of HSMs in PKI deployments and presents related works about these services. Section 3 describes the premises used to state the audit and backup schemes and two basic algorithms used to create and authenticate groups. Section 4 describes the auditing scheme while section 5 describes in detail the backup schemes. Section 6 presents the certification practice statement, in which is described the steps to be followed when a PKI is deployed and briefly introduces the backup-file creation ceremony. The statement includes policies and tasks, and explains how the witnesses can guarantee the backup procedure. Section 7 gives an overview of the main contributions to the field this paper presents and the next step to be undertaken by the working group. Appendix states the primitive functions used in the proposed algorithms. These are session key generation, secret sharing schemes, asymmetric key generation, load and store information, encryption and decryption, and cryptographic token tasks.

2. RELATED WORK

As already exposed in section 1, this work is directly derived from the work by Martina[13] in the effort of establishing an open protocol to run embedded in HSMs. The protocols proposed by Martina in his work are used to create a liable environment to store our private keys for PKI usage. The main characteristics of these protocols are the existence of an internal PKI, where all administrative keys are subject of, the existence of groups to share the control over the roles being used, and the existence of two different roles: the administrators (security-officers) and the operators (users). Another important characteristic of this work is the establishment of policies when enabling keys to use, giving in this way a very strict control over their usage.

Although we see a growing concern about key escrow systems to give availability to (private) keys [16, 3] in a secure way, there is very few specific work relating this with the confinement idea of Hardware Security Modules, where we have a cryptographic perimeter.

Some companies have been working in addressing the problems of make backup copies of operational HSM. Ncipher Corporation, address the problem in their nShield Series[14], by designing what they call a Security World. A security world consists in a cryptographically wrapped environment that is saved in the host machine with all the content of the HSM. This security world will only be unwrapped in an authorised HSM, that belongs to the security world in question. This approach gives very little auditing data, and trusts blindly in the administrators' group.

Other approach includes token replication [5], which involves just copying the private keys to other tokens in a controlled way, leaving administrative and auditing data behind and without backup.

It can look simple, but it is not easy to make a backup of an HSM. First of all, rigid control is needed in the number of keys and their usage. This requirement must be taken into account when the backup procedure is carried out. A basic protocol might be for administrators to copy the internal data of the HSM and transfer it to new hardware. The problem with this and other protocols is that it is hard to control the backup and keep it safe. To improve simple protocols, the backup may remain encrypted by the administrators' public key. If this is done, only the administrators can install the backup onto new hardware. However, doing this means trusting the administrators blindly.

Blind trust issues are always not recommended by best security practices in PKI [4]. It has shown that it is advisable to have an independent group to audit the procedures performed by administrators. To achieve this, we should introduce the auditors role. A protocol should be designed to take into account two different kinds of user profiles — administrators who make the backups and auditors who trace the number of backups created, and the number of them made operational.

Each time the HSM is used, the auditors should analyse the logs produced. To carry this out properly, it is necessary to have previously established a PKI practice statement. This statement is a document which describes different ceremonies, including important ones such as when a backup is created, and when a backup HSM is made operational. We will discuss this issue in Section 6.

Other important and parallel work to be overviewed is the key management in group protocols and communications [15, 1]. Although not directly related, studies in this field

show that it is often difficult to assure the behaviour of any involved party in group protocols. So, in this way, this is a reinforcement for the point of applying an auditing strategy and a ceremony design to trace these known problems.

3. BASIC ALGORITHMS

This paper deals with backup and audit procedures of an HSM. Some premises were taken into account to facilitate the description of these procedures. Moreover, there are two basic protocols that are used to create and to authenticate groups. This section lists these premises and protocols.

The premises are:

- HSM when initialised generates a key pair (kr_h, ku_h) and a self signed certificate (c_h). This certificate is used to uniquely identify the HSM and is trusted to issue certificates to the groups' members;
- each type of group has different data storage (DS). N.b. administrators' data storage (ADS) for administrators, auditors' data storage $(AudDS)$ for auditors and operators' data storage (ODS) for operators groups. Although they look like different pieces in the algorithm descriptions, when analysing and formalising the algorithms we consider them part of the owning principals;
- each group has an identifier (id) which uniquely refers to one group of the same type in an HSM. This is not true for administrator groups because an HSM has just one valid group at the same time;
- each group uses the secret sharing scheme[17, 2], owning a symmetric key (ks), that is split in n shares where at least m must be joined to recover the group's key. The thresholds must follow the rule: $1 \leq m \leq n$. A set of shares is represented by Ks;
- each member of each group receives a key pair (kr_i, ku_i) and a certificate (c_i), issued by HSM. Each member owns a cryptographic token (ct_{s_i}) to store this information for authentication;
- A share is assigned to each group's member. This share is encrypted by using the public key of the member's certificate. The shares are used for group authentication;
- every internal certificate is verified before its use;

The first common algorithm to be stated, after the premises presented, is the **createGroup()**. It works to create administrators, operators and auditors groups.

The **createGroup()** algorithm starts creating a session key ks for the group. In step 2, ks is split in n parts where at least m are required to recover it, resulting in Ks, a set of shares. The steps between 3 and 10 are executed n times, a time for each member of the group. Starting the loop, in step 4, a key pair is generated for the current member. Following, in step 5, the HSM loads from the host machine the name of the member id_i. This name will be used in step 6 as the member's distinguished name on the certificate c_i issued by the HSM. After that, the private key (kr_i), the certificate (c_i) and the HSM's certificate (c_h) are stored in his cryptographic token ct_{s_i}. In order to perform future authentication, the member's share is encrypted and, after, stored into the group's data storage (DS) with identification

[1]All descriptions for principals and primitive functions are available in the Appendix under section A

Algorithm createGroup(DS**,** id**,** m**,** n**,** c_h**,** kr_h**)** [1]

Creates a group based on secret sharing scheme, generating a key pair (kr_i and ku_i) and a certificate (c_i) for each member (s_i). The members' certificates are issued by HSM using its private key (kr_h) and information entered using the host machine (hm). The issued certificate and corresponding private key are stored into the cryptographic token (ct) belonging to each group's member. The group's session key (ks), which is split in n shares, is also returned as a result of the algorithm run.

1: $ks \leftarrow$ genKeySession()
2: $Ks \leftarrow$ splitSecret(ks, m, n)
3: for ks_i in Ks do
4: $(kr_i, ku_i) \leftarrow$ genKeyPair()
5: $id_i \leftarrow$ load(hm, s_i)
6: $c_i \leftarrow$ genCert(id_i, ku_i, c_h, kr_h)
7: store(ct_{s_i}, kr_i, c_i, c_h)
8: $eks_i \leftarrow$ encrypt(ks_i, c_i)
9: store(DS, id, c_i, eks_i)
10: end for
11: store(DS, id, m, n)
12: return ks

id and his certificate c_i. Finally, in steps 11 and 12, the values of m and n are stored and the group's session key is returned.

Authenticating a group and recovering the group's session key (ks) is another common procedure used in the HSM. To authenticate, the algorithm **authenticateGroup** uses a nonce u in order to avoid Dolev-Yao's replay attack[6] as explained by Martina[13].

Algorithm authenticateGroup(DS[, id] **)** [1]

Authenticates a group identified by id in the data storage DS. The member's certificates are read from the cryptographic token ct. The process is, basically, to decrypt at least m pieces of shared secret, using members' private key, joining after to recover the group's session key ks.

1: $m \leftarrow$ load(DS[, id])
2: for i = 1 to m do
3: $c_i \leftarrow$ load(ct_{s_i})
4: $eks_i \leftarrow$ load(DS[, id], c_i)
5: $u \leftarrow$ genKeySession()
6: $eu \leftarrow$ encrypt(u, c_i)
7: $eks_u \leftarrow$ ctDecrypt(ct_{s_i}, eks_i, eu)
8: $ks_i \leftarrow$ decrypt(eks_u, u)
9: end for
10: $ks \leftarrow$ joinSecret(Ks)
11: return ks

A group's authentication starts loading from DS the information of threshold m identified by id. This information is used for the iteration which will use enough shares to recover the group's session key. In step 3, the current member's certificate (c_i) is loaded from his cryptographic token. In step 4, the encrypted share belonging to the member is loaded from DS. In steps 5 and 6, the nonce u is generated and then encrypted by using the member's public key (which can be found in c_i). In step 7, the encrypted share and the nonce are sent the cryptographic token to be decrypted using the member's private key. The result of this operation will be the shared secret encrypted using u as a key session.

In step 8, a piece of the group's secret is decrypted. After at least m shares be decrypted, ks can be recovered as stated in step 10 and, finally, the algorithm returns ks in step 11.

4. AUDITING SCHEME

To meet all the security requirements of the process, audit trails must be created, and a specific group of people, who will act as the HSM auditors, should be defined. Their main purpose is to collect auditing data from the HSM, enabling them to analyse the administrators group behaviour and to report any incident to the PKI's management team. This team should take the necessary procedures in order to guarantee the security of the whole PKI.

The auditors groups in an HSM should be considered as an inspection body. They always act with the purpose of assuring that other groups are acting and behaving as expected by their controlling authority (like PKI policy team).

The auditors group is also an important piece in the whole OpenHSM protocol, because it checks the auditing trail during the entire life of an HSM, and assures that no tampering, physical or logical, happens to the HSM while it is being used or kept in storage.

Our proposal states that the HSM should have at least one auditors group, but this does not limit the possibility of having more than one. Even the auditors group should not be trusted by other groups, and should be inspected by others. No destruction or changes to the auditors groups are planned, because a new group can easily be created if an existing one has been compromised, since they do not perform any vital cryptographic operations in the OpenHSM protocol.

The creation of an auditors group must be made just after the initialisation of the HSM, and the creation of the administrator groups, so it will be possible to audit everything that happens in the HSM life cycle. The **createAudGroup** algorithm has been proposed to handle the creation of auditors group.

Algorithm createAudGroup(id**,** m**,** n**)** [1]

Creates an auditors group id following threshold n and m, authenticating the administrators group before. A key pair is created and assigned to it

1: $ks_{ad} \leftarrow$ authenticateGroup(ADS)
2: c_h, $ekr_h \leftarrow$ load(ADS)
3: $kr_h \leftarrow$ decrypt(ekr_h, ks_{ad})
4: $ks_{au} \leftarrow$ createGroup($AudDS$, m, n, c_h, kr_h)
5: kr_{au}, $ku_{au} \leftarrow$ genKeyPair()
6: $c_{au} \leftarrow$ genCert(id, ku_{au}, c_h, kr_h)
7: $ekr_{au} \leftarrow$ encrypt(kr_{au}, ks_{au})
8: store($AudDS$, id, c_{au}, ekr_{au})
9: return c_{au}

The **createAudGroup** algorithm starts when the administrators group enters the identifier, id, and the thresholds for the new group, m and n. In step 1, the administrators group is authenticated using **authenticateGroup()** algorithm, recovering the group's session key (ks_{ad}). From ADS, the certificate c_h and encrypted private key ekr_h of the HSM is loaded. The private key is decrypted in step 3 using ks_{ad}. In step 4, the auditors group is created and the group's key session ks_{au} is the result of it. In step 5, the key pair of the group is created. HSM, in step 6, issues the group's certificate. In step 7, the group's private key is encrypted by using its key session. After that, all required data describing the auditors group are stored into $AudDS$, such as the group's identifier (id), certificate (c_{au}) and encrypted private key ekr_{au}. Finally, the group's certificate is exported as a result of the algorithm.

In order to achieve the objectives of auditors groups the next algorithm is presented. It implements the protocol to export the HSM's logs. It enables the auditors to trace all operations that have occurred in the HSM. The algorithm, basically, authenticates the auditors group which is requesting the log and, using its private key, signs the log package. Afterwards, the signed log package is exported.

Algorithm exportLog(id **[,** $rangeDate$**]** **)** [1]

Allows an auditors groups, identified by id, to export the signed HSM's logs. It's possible to specify a date range $rangeDate$

1: $ks \leftarrow$ authGroup($AudDS$, id)
2: $ekr \leftarrow$ load($AudDS$, id)
3: $kr \leftarrow$ decrypt(ekr, ks)
4: $L \leftarrow$ load(LDS, $rangeDate$)
5: $sL \leftarrow$ sign(L, kr)
6: return(sL)

The **exportLog** algorithm starts when an auditors group informs its identifier id and, optionally, a date range, specifying a period for the logs. In step 1, the auditors group is authenticated and the group's session key ks is returned as result of it. The group's encrypted private key is loaded in step 2, and then, decrypted. In step 4, the specific block of log is selected and, therefore, signed in step 5 using kr. Finally, the signed log sL is exported.

5. BACKUP SCHEME

The following sections present diagrams and descriptions of the algorithms relating to the backup procedures, showing how to create a new operational HSM by installing the most recent copy of the data used by the old HSM into new hardware.

5.1 Preparing an HSM to be a backup unit

The first step, in having a secure backup system, is obtaining a spare hardware which is specifically configured to replace the existing HSMs when required. This hardware should be on standby, ready to become functional by receiving data previously copied from an operational HSM.

The preparation requires no initial information from any operating HSM, only the hardware in its factory state. The procedure is even possible when there are no operational HSMs. As soon as an HSM is prepared to be a recipient, it becomes possible to make a backup of the operational HSM. A backup of an operational HSM can only be made by using the certificate of the backup HSM. The administrator group must install this certificate into the operational HSM before it can create any backup file.

Once the backup HSM is a backup unit, it cannot be used for other activities. The backup itself is an encrypted copy of the data from an operational HSM. This encrypted copy is made by the administrators of the HSM from which should make new copies from time to time, as established in the PKI practice statement. The physical security of the HSM is responsible for guaranteeing the backup private key, kr_{bkp}, enclosed.

The algorithm proposed to handle the preparation of an HSM to be a backup unit is as follows:

Algorithm prepareBkpHsm(id_{bkp} **)** [1]

Prepares an HSM to be a backup unit, generating a key pair (kr_{bkp} and ku_{bkp}) and, then, issuing a self-signed certificate c_{bkp} using it. The certificate is exported as result of it

1: kr_{bkp}, ku_{bkp} ← genKeyPair()
2: c_{bkp} ← genSelfSignedCert(kr_{bkp}, ku_{bkp}, id_{bkp})
3: store(BDS, kr_{bkp}, c_{bkp})
4: return c_{bkp}

To run the **prepareBkpHsm** algorithm, the administrators group sends the information id_{bkp} to the new HSM, which is used for issuing the self-signed backup certificate c_{bkp}. Starting in step 1, the HSM generates a key pair kr_{bkp} and ku_{bkp}. This pair and the id_{bkp} information are the required data to issue the self signed certificate c_{bkp} in step 2. Therefore, kr_{bkp} and c_{bkp} are stored in the backup data storage (BDS), keeping kr_{bkp} inside the cryptographic perimeter of the HSM until it is necessary to recover a secure backup.

To enable the administrators to follow the next algorithm, the result of this algorithm is a certificate c_{bkp}, which contains the information sent by the administrators during the initial phases of this protocol run.

5.2 Importing a Backup Public Key Certificate

Figure 1 illustrates how a backup certificate is exported

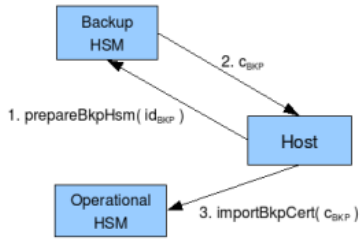


Figure 1: Copying the backup certificate.

from a backup HSM and imported into an operational one. It is possible to use the same or different host machines to do so. By running the remote management system in the **Host**, the new hardware is commanded to work as a backup HSM. Upon receiving the command, the new hardware generates its own key pair and self signed certificate. Next, **Host** downloads the certificate and saves it as a file. By using the remote management system again, the certificate is uploaded into the operational HSM. After this, the operational HSM can make backups of its internal data. With this design, there is no limit to the amount of backup certificates that can be imported into an operational HSM. The backup, made in this way, can be installed in any one of the prepared backup HSM. Thus, the PKI management team could decide to have additional backup HSMs prepared, which can be used if one of them fails.

Purely cryptographic means cannot assure the origin of a backup certificate because it is a normal certificate and could have been generated anywhere. So, here, it is essential to have a record and an auditing group to inspect it. Certificates that have been imported must be verified by the auditor group: importing a flawed backup certificate could lead to leakage of sensitive material.

To import a backup certificate into a operational HSM, it must be initialised and must have an administrators group beforehand. The algorithm to import a backup certificate into an operational HSM is as follows:

Algorithm importBkpCert(c_{bkp} **)** [1]

Imports a backup Certificate (c_{bkp}) which has been exported from a backup HSM. The authentication of administrators group is required.

1: ks ← authenticateGroup(ADS)
2: ekr_h ← load(ADS)
3: kr_h ← decrypt(ekr_h, ks)
4: sc_{bkp} ← sign(c_{bkp}, kr_h)
5: store(BDS, sc_{bkp})

To start the **importBkpCert** algorithm, the administrators group must give the backup certificate c_{bkp} to the HSM. Following, the HSM authenticates the administrators group, in step 1, and release the HSM's private key kr_h over the steps 2 and 3 (this process is covered in section 4). Finally, c_{bkp} is resigned using kr_h and stored into backup data storage (BDS). The backup certificate is resigned to avoid any unauthorised inclusion in the BDS of unrelated backup HSM certificates.

5.3 Creating Backups

One of the most important operations in an HSM is to create secure and targeted backups. First, it must be stated that the backup creation process is a very delicate procedure, because it goes directly against normal goals, i.e., to keep sensitive cryptographic material inside the HSM protected perimeter. It is also something that must be done to ensure the continuity of the key life cycle, even in the case of a hardware fault, or because of tampering.

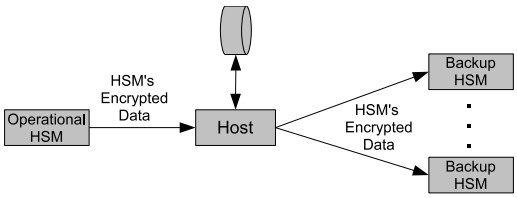


Figure 2: Creating backups.

Despite this, periodic copies of the HSM content must be made in order to enable the recovery of the whole environment just in case of a failure. Figure 2 illustrates this idea.

Data can only leave the internal perimeter of the HSM in an encrypted form; the design of the proposed backup process can therefore be described as "secure" as stated in FIPS 140-2[8]. And it can also be labeled as "targeted" because the Backup Package File BPF can only be opened in previously designated $HSMs$.

The administrators group must exist as at least one backup certificate must be imported to perform the backup of an HSM. Additionally, for security reasons, it is necessary to have one or more auditors groups. The last requirement

comes to not blindly trust to a single group (administrators group) the backup process.

The following algorithm is proposed to handle the generation of a security copy:

Algorithm bkpHsm() [1]

Creates a secure copy of an operational HSM just after the administrators group authentication. This copy can be recovered at any backup HSM whose backup certificate has already been imported into the operational HSM. This procedure copies the whole internal environment, excluding non-exportable (NXD's) data.

1: $ks \leftarrow$ authenticateGroup(ADS)
2: ekr_h, $c_h \leftarrow$ load(ADS)
3: $kr_h \leftarrow$ decrypt(ekr_h, ks)
4: store(BPF, load(CTL))
5: store(BPF, load(ADS))
6: store(BPF, load($AudDS$))
7: store(BPF, load(ODS))
8: store(BPF, load(KDS))
9: store(BPF, load(BDS))
10: $C_{bkp} \leftarrow$ load(BDS))
11: $ks_{bkp} \leftarrow$ genKeySession()
12: $eBPF \leftarrow$ encrypt(BPF, ks_{bkp})
13: $seBPK \leftarrow$ sign($eBPK$, kr_h)
14: for c_{bkp_i} in C_{bkp} do
15: $\quad eks_{bkp} \leftarrow$ encrypt(ks_{bkp}, c_{bkp_i})
16: end for
17: return $seBKP$, EKs_{bkp}

Once the backup process is triggered, the administrators group is authenticated using **authenticateGroup** algorithm, stated by step 1. In step 2, the HSM's certificate (c_h) and encrypted private key (ekr_h) are loaded from ADS, and then, the HSM's private key is released in step 3. The steps between 4 and 9 copy the current content of all data stored into backup package file (BPF), excluding non-exportable data storage (NXD) as explained by Martina[13]. Briefly, the NXD stores a part of the secret required for the administrators to perform any action over operators groups, then, once the backup is recover in a new HSM, each operators group must explicitly authorise the adminitrators to act again on its behalf. This gives assurance to the operators group in question that no backup of their keys can be recovered and become usable without their consent.

In step 10, the HSM loads the set of backup certificates (C_{bkp}) previously imported to define the target HSMs. The backup session key is generated in step 11 and used in step 12 for encrypting the BPF. A signature is made in step 13 from the encrypted backup package file for external checks. Therefore, in steps 14 to 16, the loop performs an encryption of the backup session key ks_{bkp} using each public key of backup HSM's certificates, creating as many encrypted session keys as backup certificates there exist. Finally, the list of encrypted backup session keys and the signed encrypted backup package file ($seBKP$) are exported as a result of the execution.

5.4 Recovering Backups

Recovery of HSM backups is only allowed in HSMs already prepared as backup HSM, i.e., where the algorithm from subsection 5.1 were run, and the exported backup certificate has been imported previously before the backup was made. The backup receiver unit has its private key kr_{bkp}, which it keeps protected against disclosure inside the HSM perimeter and will be used for decrypting the backup package file BPK, recovering the content of the HSM.

Recovering backup procedure authenticates administrators and an auditors group. Both authentications are only required whereas these procedures are considered administrative and necessary to keep the auditors aware of a new operational HSM.

The following algorithm is proposed to handle the security copy recovering:

Algorithm recoverBkp($seBPF$, $EK_{S_{bkp}}$, id_{audit}) [1]

Recovers a backup package file BPF, extracted from a signed encrypted package $seBPF$, in a previously prepared HSM, authenticating administrators group and an auditors group identified by id_{audit}. Both authentication are not really required to happen by cryptographic means

1: $kr_{bkp} \leftarrow$ load(BDS)
2: $ks_{bkp} \leftarrow$ decrypt(eks_{bkp}, kr_{bkp})
3: $BPF \leftarrow$ decrypt($seBPF$, ks_{bkp})
4: store(CTL, load(BPF))
5: store(ADS, load(BPF))
6: $ks_{adm} \leftarrow$ authenticateGroup(ADS)
7: store($AudDS$, load(BPF))
8: $ks_{audit} \leftarrow$ authenticateGroup($AudDS$, id_{audit})
9: store(ODS, load(BPF))
10: store(KDS, load(BPF))
11: store(BDS, load(BPF))

The private key of the backup HSM (kr_{bkp}) is loaded from backup data storage BDS in step 1. In step 2, the session key ks_{bkp} used to encrypt the backup package file (BPF) is decrypted using kr_{bkp}. The BPF is decrypted in step 3 using ks_{bkp}, released previously. As explained before, this algorithm must authenticate the administrators group and a auditors group, so, in steps 4 and 5, the HSM reads CTL and ADS from BPF and, in step 6, authenticates the administrators group. Once again, in step 7, $AudDS$ is read from the BPF, enabling the HSM to authenticate the auditors group in step 8. Finally, in steps 9 to 11, ODS, KDS and BDS are read from BPF and a new instance of the HSM is made fully operational.

6. ADDITIONAL PROCEDURES

Security hardware by itself, in real situations, is not sufficient to guarantee the security of critical systems. Additional procedures are necessary, such as the testimony of witnesses. These procedures and testimonies should be previewed in a PKI practice statement as well as the description of all ceremonies. Important ceremonies include the creation of backups and the moment at which a backup HSM is made operational.

Despite very pertinent to PKI operation, this topic is oftenly forgotten. Recently, it was revisited by Ellison [7], who points its real importance to cover all outbound operations, normally done by human nodes, in any secure operation which involves mechanised nodes. Normally we do have to consider for security reasons, all operations in any security protocol or algorithm, even those needed to create the requisites to their operation.

In our case,as a prerequisite, every operation must be logged because these logs allow the auditors groups to create

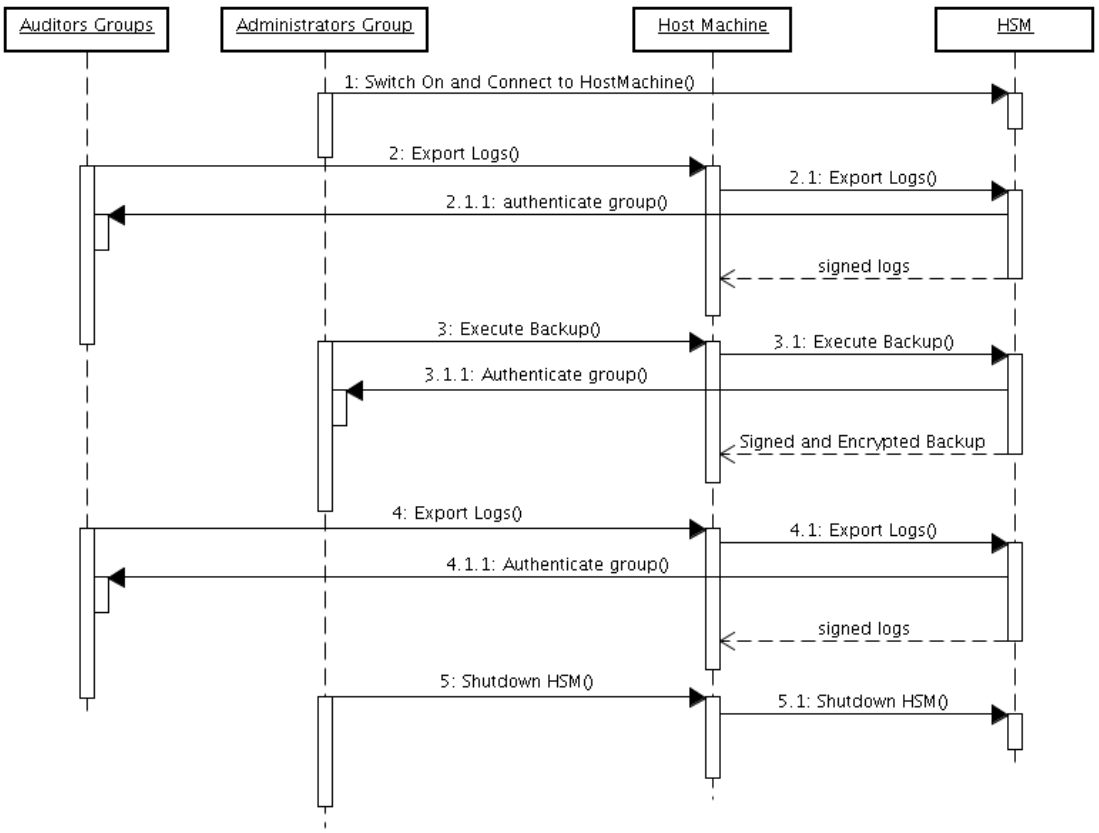


Figure 3: Creating backup file ceremony

activity trails. If any problem occurs with the main HSM, its backup can be recovered. Its internal state will be just as at the backup procedure, without logs and procedures done subsequently. So, just a backup does not solve their problems. The auditors groups have an important role in this context: tracing operations, controlling the administrators, validating the operators groups' activities and others.

As shown in the last section, the creation of a backup involves many steps and, consequently, some critical points are met. A useful way of avoiding possible mistakes and maintaining the systems is to perform ceremonies. Figure 3 shows the ceremony to create backup files in the OpenHSM.

The full ceremony description for the OpenHSM usage is not the focus of this present paper, but this small fragment already shows us two important concerns:

- A log extraction in the beginning of the ceremony, to enable us to trust the HSM before starting the procedure, thus avoiding operation if any tamper tentative was tried since last operation;

- A log extration in the end of the ceremony, enabling us to assure that the operation succeeded and that nothing else happened during this operation.

Ceremonies should become essential resources for all main operational activities. Ceremony steps should be planned and should extract logs from the beginning and the end of the executed procedures, with registers of all the ceremony steps, involving as many people as possible and, ideally, recording the ceremony.

After a ceremony, a final report must be generated using all available data, which should also include ceremony steps' registers and logs. Finally, attendees present sign the report guaranteeing the veracity of the operation. Tests on the ceremony steps should be performed beforehand, using a parallel environment to try to avoid problems. These ceremonies will also allow the auditors groups to follow the system trail using logs.

Additionally, to avoid any possible compromise of the solution, some additional measures should be taken. If, for instance, an HSM becomes inoperative, the HSM should not be repaired. In this case, the HSM should be burnt, for instance, in a incinerator. This eliminates any possibility of keys being recovered.

Finally, the operational HSM and its backup HSM should be kept in different sites, avoiding the loss of both at the same incident.

7. FINAL CONSIDERATIONS

Open Hardware Security Module (OpenHSM) is a project carried out by the National Education and Research Network (RNP). RNP provides a Brazilian infrastructure of advanced networks for collaboration and communication in the fields of teaching and research. OpenHSM is a specialised HSM directed to PKI applications such as Certification and Registration Authorities.

This paper described the deployment of the OpenHSM in a Public Key Infrastructure. Its main contribution was to improvements in the auditing system and backup operations for the OpenHSM project. It covered descriptions of algorithms used by the OpenHSM to enable it to perform secure backups and provide an audit trail. It also introduced a ceremony procedure to support the operation of such HSMs in a PKI deployment.

The next step in this project will be synchronising the internal clock to a trusted external time source. Then, the OpenHSM can also be used as a security time stamping server. Other future work can include the formal analysis of all protocols relating to the OpenHSM and the creation of all related ceremonies.

8. ACKNOWLEDGEMENTS

We would like to acknowledge the Brazilian Research Network (RNP) for the financial support in the GT ICPEDU workgroup, Brazilian Chamber of e-Commerce (Câmara e.net) for scholarship support in LabSEC, and also to Chris Sowton, Kristian Glass and Mark Reynolds for proof reading the paper.

9. REFERENCES

[1] Y. Amir, Y. Kim, C. Nita-Rotaru, and G. Tsudik. On the performance of group key agreement protocols. *ACM Trans. Inf. Syst. Secur.*, 7(3):457–488, 2004.

[2] G. R. Blakley. Safeguarding cryptographic keys. In *AFIPS 1979 National Computer Conference*, pages 313–317. AFIPS, 1979.

[3] J. Brown, J. M. G. Nieto, and C. Boyd. Efficient and secure self-escrowed public-key infrastructures. In *ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 284–294, New York, NY, USA, 2007. ACM.

[4] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu. Internet X.509 public key infrastructure certificate policy and certification practices framework. RFC 3647, 2003.

[5] Chrysalis-ITS. Luna pki hsm planning and integration guide. http://www.chrysalis-its.com, 2002.

[6] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[7] C. Ellison. Ceremony design and analysis. Cryptology ePrint Archive, Report 2007/399, 2007. http://eprint.iacr.org/.

[8] FIPS. Security requirements for cryptographic modules, FIPS PUB 140-2, 2002.

[9] IBM Coorporation. Ibm 4758 model 2 and 23 pci cryptographic cooprocessor manual. 2000.

[10] iVEA (Rainbow Technologies). Cryptoswift hsm user's guide. http://www.rainbow.com/, 2001.

[11] iVEA (Rainbow Technologies). Cryptoswift hsm user's guide. http://www.rainbow.com/, 2001.

[12] J. E. Martina. Project of a hardware security module focused on public key infrastructures and its applications. Master's thesis, Federal University of Santa Catarina, March 2005.

[13] J. E. Martina, T. C. S. Souza, and R. F. Custódio. OpenHSM: An open key life cycle protocol for public key infrastructure's hardware security modules. In *Fourth European PKI Workshop: Theory and Practice, EuroPKI 2007*, volume 4582 of *LNCS*, pages 220–235. Springer-Verlag Berlin Heidelberg, 2007.

[14] nCipher Corporation Ltd. Nshield user guide linux version 4.17.38. http://www.ncipher.com, 2004.

[15] S. Rafaeli and D. Hutchison. A survey of key management for secure group communication. *ACM Comput. Surv.*, 35(3):309–329, 2003.

[16] P. Samarati, M. K. Reiter, and S. Jajodia. An authorization model for a public key management service. *ACM Trans. Inf. Syst. Secur.*, 4(4):453–482, 2001.

[17] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[18] T. C. S. Souza. Aplicações embarcadas para gerenciamento de chaves criptográficas. Technical report, Federal University of Santa Catarina, 2005.

APPENDIX

A. CONVENTIONS

The algorithms in this paper are subject to conventions from Table 2. This table presents the objects which store data from the HSM. This information is used to, with the protocols, manage the HSM. For instance, in the auditors data storage *AudDS* has details of auditors groups such as groups' name, groups' size and others.

Table 2: Principals of the Protocols

Principal	Description
ADS	Administrators Data Storage
$AudDS$	Auditors Data Storage
BDS	Backup Data Storage
BPF	Backup package file
CTL	Certificate Trust List
KDS	Keys Data Storage
LDS	Log Data Storage
NXD	Non-exportable data
ODS	Operators Data Storage

Additionally, these are the description of all primitive functions used, along with the algorithms, which have not been introduced yet. Basic functions are considered primitive when only a couple of operations are done and/or its name is self-explanatory:

ctDecrypt(ct**,** $edata$**,** eu**)** Uses the private key stored in the cryptographic token ct to decrypt data $edata$ and nonce eu. Before returning the result, $data$ is encrypted using nonce u.

decrypt($edata$**,** k **)** Decrypts encrypted data $edata$ using key k (symmetric or asymmetric).

encrypt($data$**,** k **)** Encrypts $data$ using key k (symmetric and asymmetric). If k is a certificate, its public key is used.

genCert(id**,** ku**,** c_{ca}**,** kr_{ca} **)** Certification authority ca generates a certificate with identification id and public key ku. c_{ca} is the certification authority certificate and kr_{ca} is its private key.

joinSecret(Ks **)** Joins the shares Ks in order to reconstruct session key ks. Remember that the set Ks must have the minimum number of shares, as when split.

load (DS**[,** a **)]** Reads information from data storage DS (DS might represent the host machine). Optionally, it specifies one or more pieces of information to restrict the result, such as identifiers.

genKeySession() Generates a random session key.

genKeyPair() Generates an asymmetric key pair

genSelfSignedCert(kr**,** ku**,** id **)** Generates a self signed certificate with identification id and public key ku. kr is the private key.

sign($data$**,** kr **)** Signs $data$ using the private key kr.

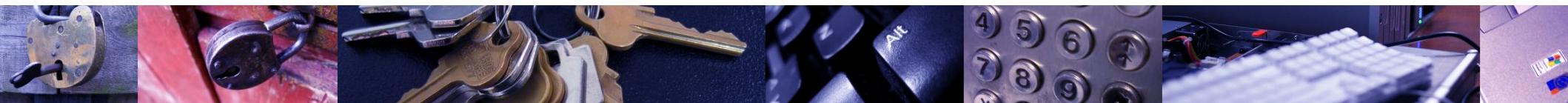
splitSecret(ks**,** m**,** n **)** Splits, using secret sharing scheme, session key ks in n shares where at least m of them are required to reconstruct it.

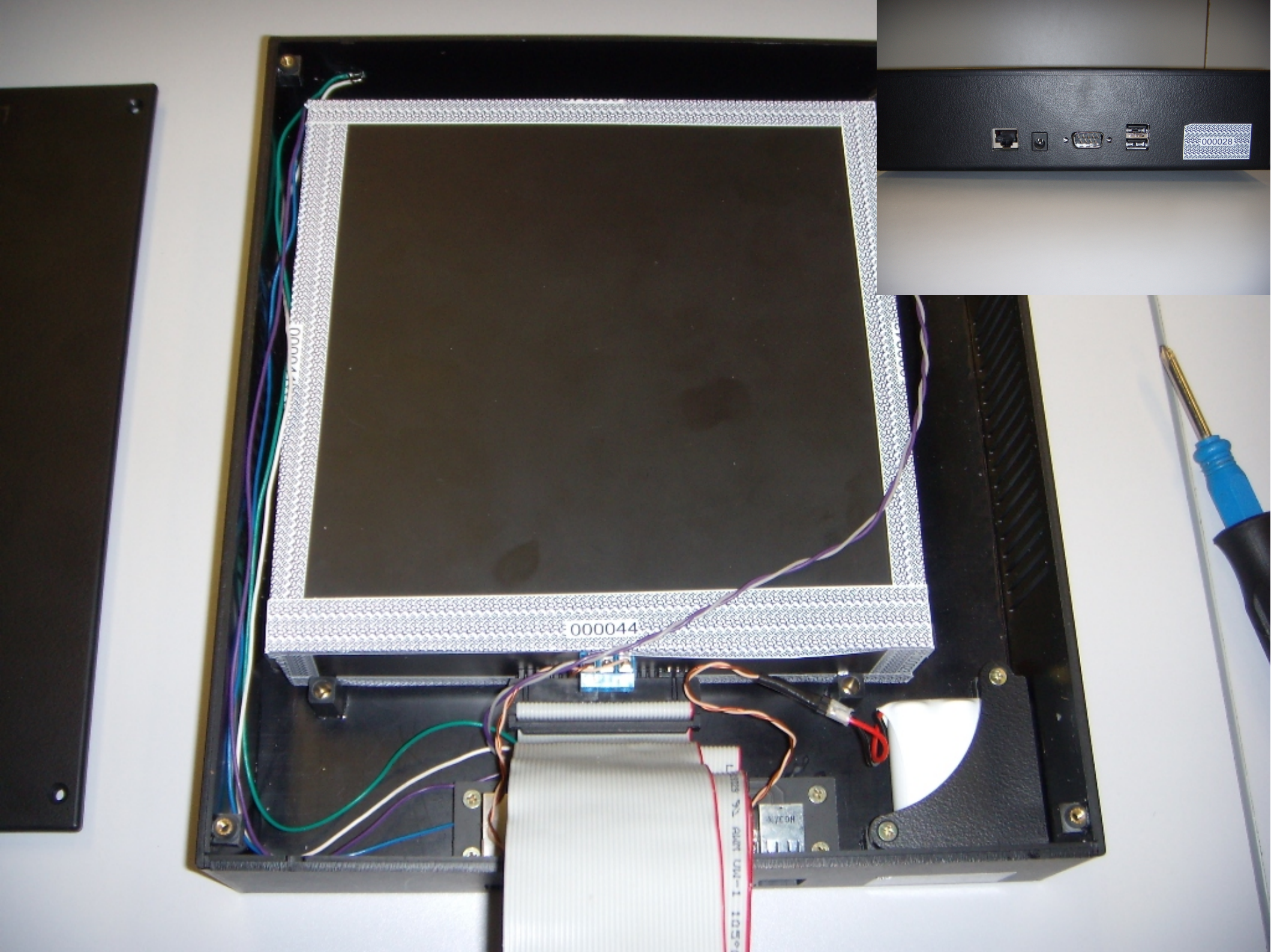
store (DS**, ...)** Stores into data storage DS all other remaining parameters.

Audit and backup procedures for Hardware Security Modules

Jean Martina
Jean.Martina@cl.cam.ac.uk

Joint work with:
Túlio de Souza, UFSC, Brazil
Ricardo Custódio, UFSC, Brazil

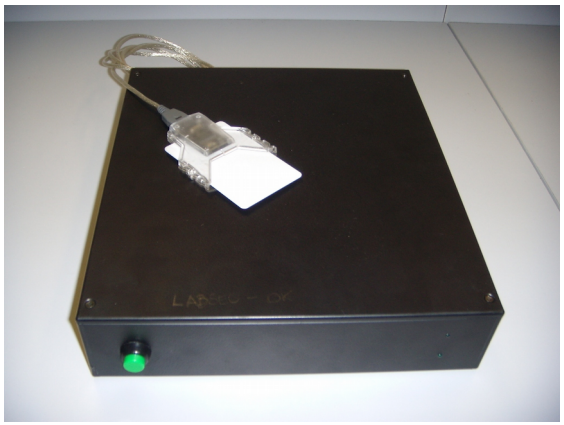



000028
000044

If we thought.....

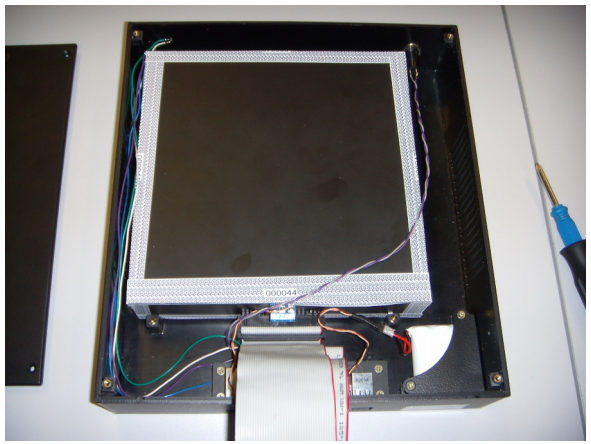
PIN Calculation
Role based authentication
Dual key entry
Payment protocols
Cryptographic speed

If we thought.....



Strong authentication
Identity based authentication
Strict key life-cycle control
Fully auditable operation
Triggered group mechanisms

What we (the authors) think:



HSM designed for PKI
Inside the box
Auditing Scheme
Backup Scheme
Additional Procedures

HSM Designed for PKI(platform)

What we had:

- FIPS 140-2 Level 3 Hardware (Equivalent)
- Upgradeable firmware
- Open source based
- Smart-card support
- Ethernet connection
- Java management interface
- Crypt Interfaces (OpenSSL, PKCS#11)

Inside the Box (before)

- Built-in PKI to control the HSM
- Administrators (security officers)
- Operators (users)

Threshold Cryptography

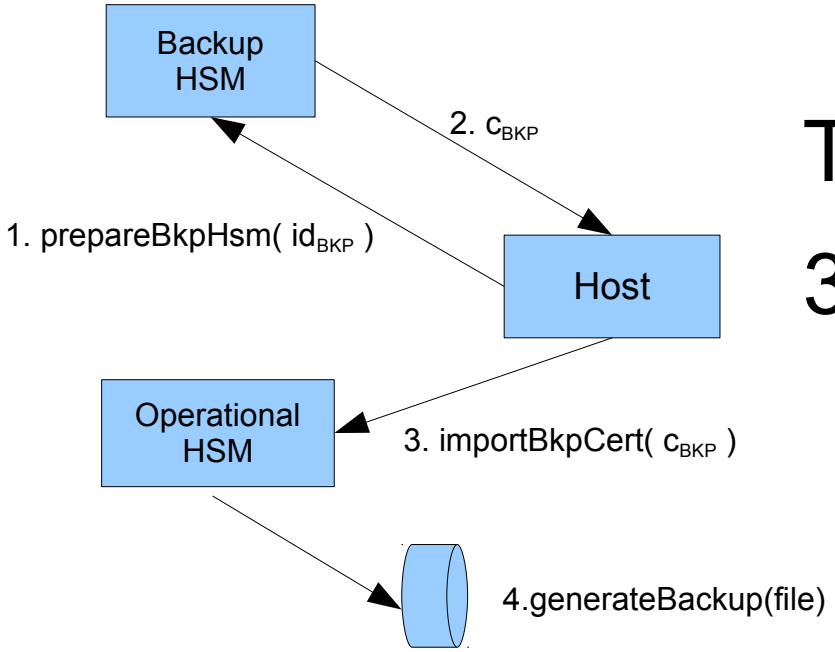
- Managed keys
  - Extended operational policies
- Additional operations
  - Changes administrators
  - Change key's ownership

Auditing Scheme

- Everything must be logged
- Auditors group

  Threshold Cryptography
- Trace administrative operations
  - Initialisation
  - Typical operations
  - Backup control
- Unlimited Auditors groups

Backup Scheme (Initialise)

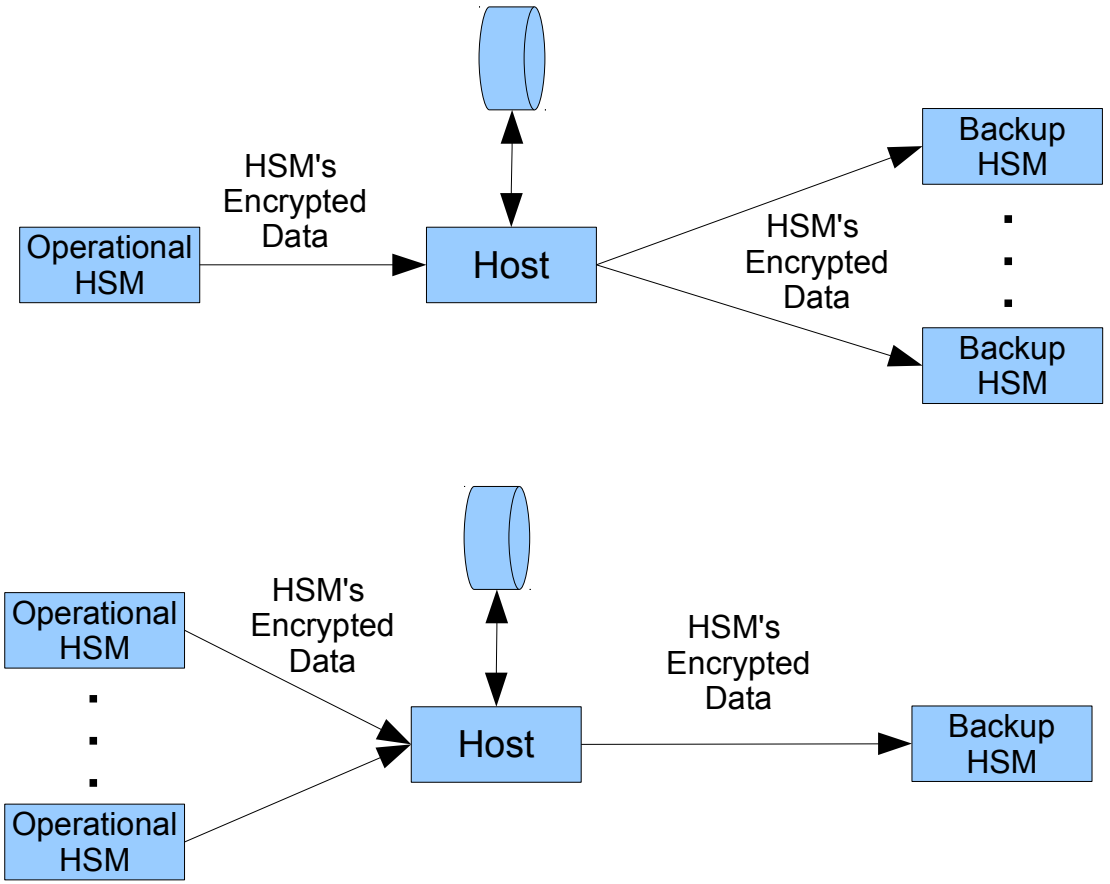


- Only one key operational at a time
- Targeted
- 3 Steps:
  - Prepare backup unit
  - Import backup certificate
  - Generate backup file

Questions:
- Do we need a spare unit?
- How to trust a self-signed certificate?
- Is it secure?

Answers:
- Yes, but the relation is n/n
- Auditors will check this in an additional procedure (ceremony)
- Well, if nothing bad happens....

Backup Scheme (Operate)



Additional Procedures

Ceremonies?

Why?

- Cryptography only can not guarantee “human factor”
- Out bound threats must be addressed

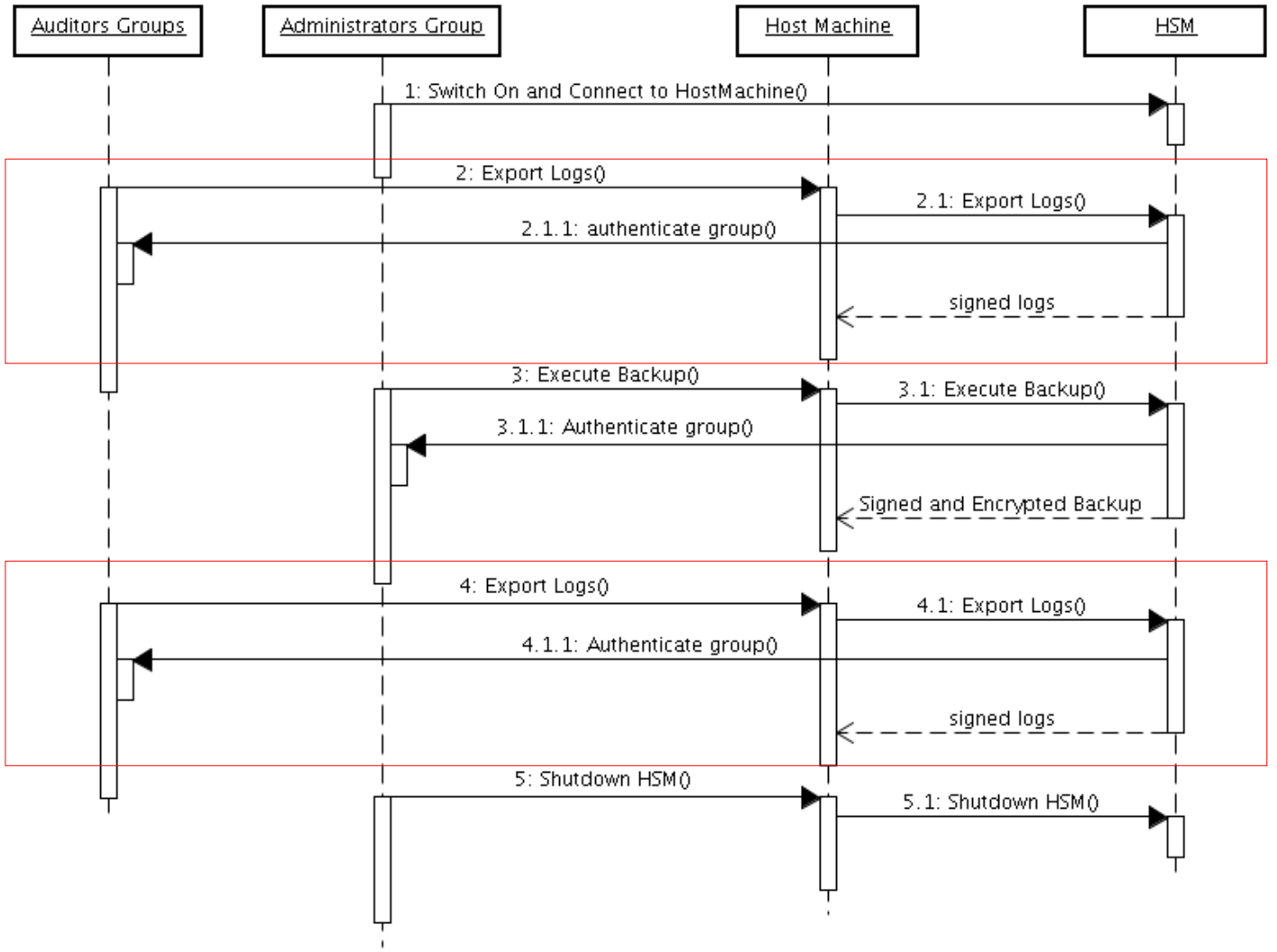
Log strategies

- Log extraction before (check initial state)
- Log extraction in the end (what happened)

Witnesses

Written Acts

One Ceremony



Final Considerations

HSMs in general are designed for Banks

Auditing is a key issue for a PKI HSM

HSM's backups are a contradictory issue

- Keep the key protected, however spread encrypted copies for safety.

Ceremonies are an important tool

- Very few work in general

Already operational in ICP-EDU project in Brazil

ICP-EDU

Brazilian National Education and Research Network

11 HSMs

10 CAs

300 SSL Certs and 3000 End-user Certs

2008

- Increase to 35 HSMs, more CAs, and 100.000 end users Certs

Future Work

Formal verification

Hardware improvement

Better CMS integration

Full ceremony set

Questions

Securing the Core with an Enterprise Key Management Infrastructure (EKMI)

Arshad Noor
StrongAuth, Inc.
550 Lakeside Drive, Suite 10
Sunnyvale CA 94085
arshad.noor@strongauth.com

ABSTRACT

The last twenty-five years has witnessed an emphasis on protecting the network and computing host as a proxy for protecting data from unauthorized access. While this was a reasonable strategy at the dawn of network-based computing, given the state of the internet today with its security issues, this strategy is proving to be hopeless.

This paper advances the notion that the time has finally come to begin what we should have done initially – protect the core of our computing infrastructure: the data – in addition to protecting the network and computing host.

The paper describes an architecture - and a specific implementation of that architecture - to enable the encryption of data across the enterprise in a platform and application-independent manner. The architecture describes the use of a Public Key Infrastructure (PKI) and a Symmetric Key Management System (SKMS) within an Enterprise Key Management Infrastructure (EKMI), to securely - and centrally - manage the life-cycle of the symmetric encryption keys used for data encryption.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and protection – *cryptographic controls.*
E.3 [**Data Encryption**]: *Public key cryptosystems, Standards.*

General Terms

Management, Design, Security, Standardization.

IDtrust '08, March 4-6, 2008 Gaithersburg, MD

Keywords

Enterprise Key Management Infrastructure (EKMI)
Key-management (KM)
Public Key Infrastructure (PKI)
Symmetric Key Client Library (SKCL)
Symmetric Key Management System (SKMS)
Symmetric Key Services (SKS)
Symmetric Key Services Markup Language (SKSML)
XML Encryption (XENC)
XML Signature (DSIG)

1. INTRODUCTION

Most security professionals are familiar with symmetric-key based cryptography when presented with terms such as **Data Encryption Standard (DES)**, **Triple DES (3DES)** and the **Advanced Encryption Standard (AES)**. Some are also familiar with **Public Key Infrastructure (PKI)** as an enterprise-level solution for managing the life-cycle of digital certificates used with asymmetric-key cryptography.

However, the term **Symmetric Key Management System (SKMS)** - the discipline of securely generating, escrowing, managing, providing access to and destroying symmetric encryption keys - almost always draws blank stares. Given the number of applications that needed to encrypt data in the past, this is not surprising; symmetric encryption key management has traditionally been buried within the business applications performing encryption. These applications primarily focused on business functions, but managed encryption keys as an ancillary function. Consequently, there was no reason to emphasize key-management. This paper advances the notion that the time has come to address SKMS as an application-independent, enterprise-level defense mechanism. Additionally, this article advocates the use of a PKI for securing an SKMS within an **Enterprise Key Management Infrastructure (EKMI)**.

While encryption has been in use for centuries[1], computer-based cryptography entered the general-computing field with the advent of the DES algorithm. The primary business uses for this technology was within the military and later banking. Given the nature of what encryption tech-

nology was protecting, implementers were willing to live with custom key-management solutions, however contrived they may have been. With the explosion of the world-wide web, businesses have been racing to implement business processes on the Internet, bringing sensitive information significantly closer to attacks. Although businesses have invested billions in firewalls, intrusion detectors, intrusion prevention systems and other network and host-based defense mechanisms, the US has witnessed more than 400 breach disclosures[2] since the passage of California's Breach Disclosure law[3]. Of some note are the disclosures by the University of California over the years, with the most recent one in Los Angeles (UCLA)[4]. Given that this is the *seventh* breach disclosure by the University of California across all their schools, it's reflective of a situation spiraling out of control.

All data breaches pale in comparison to the one at TJX[5], which is currently ranked as the largest breach ever with nearly 95M credit card numbers exposed at this Massachusetts-based retailer[6]. A quarterly financial statement[7] filed by this company in August 2007 shows that, so far, it has taken a charge of US $216M against this single breach. At least three lawsuits are pending against TJX with the full extent of damage yet unknown.

Breaches at retailers such as TJX, Ralph Lauren, BJ's, DSW and credit-card processing companies such as CardSystems have prompted credit-card giants Visa, Mastercard, American Express and Discover to standardize on security requirements for merchants and card-acquirers through the **Payment Card Industry's Data Security Standard (PCI-DSS)**[8]. One critical element required within PCI-DSS is the encryption of credit-card numbers and a robust key-management system to accompany it. This effort is aimed at strengthening controls protecting sensitive data on systems that have potential for causing financial damage to customers and the credit-card brands.

While the Retail sector has been particularly battered by data-breaches, the need to encrypt sensitive data in companies is also driven by the following regulation across the world:

- The **Health Insurance Portability and Accountability Act of 1996 (HIPAA)**[9], which specifies rules for how medical information must be secured within the US when used within computerized systems;
- The **Financial Modernization Act of 1999,** aka the **"Gramm-Leach Bliley Act" (GLBA)**[10], which specifies rules for how financial information must be secured within the US when used within computerized systems;
- The **Personal Information Protection and Electronic Documents Act (PIPEDA)**[11], which specifies rules for how personal information must be secured in Canada, when used within computerized systems;
- **Directive 95/46/EC of the European Parliament**, aka the European Union Directive or EU Directive[12], which specifies rules for how personal information must be secured in the EU, when used within computerized systems;
- Thirty-eight US "computer breach disclosure" laws requiring companies that have breached sensitive information of US residents, to disclose those breaches to affected individuals. Most of these laws provide a "safe-harbor" to the company by not requiring a disclosure, if the affected data affected was encrypted;

With the publication of each new computer breach, governments across the world are reacting with increasing legislation that regulates the protection of sensitive data. Companies are also becoming sensitive to adverse publicity and private lawsuits when stolen data or lost laptops and computer tapes, etc. are publicized in the media. As a result, security professionals now accept that sensitive data needs to be encrypted across the enterprise – on desktops, laptops, Personal Digital Assistants (PDAs), mobile telephones, etc. - and not just on servers and on-line or off-line storage within the Data Center.

1.1 Organization of Paper

This paper begins by describing some of the problems with current key-management systems and architectures in Section 2. In Section 3 an architecture for an SKMS is presented with an explanation for why this particular architecture makes sense and the rationale for the design decisions that were made. Section 4 describes the SKSML protocol and how applications are expected to use it. In Section 5, the paper goes on to describe a specific implementation of this architecture in an open-source product and the experience gained from such an implementation. Section 6 discusses the security required for such an architecture. Section 7 provides a high-level plan for how an SKMS can be built for production use in an IT environment and what are the issues that implementers must take into consideration. The paper finally concludes in Section 8.

2. PROBLEMS WITH CURRENT SYSTEMS

Why is symmetric key-management a problem? After all, applications seem to have addressed the problem within the applications for decades, and appear to be continuing to do so. The problem becomes obvious from the perspective of a manager in IT Operations. As an illustration, if the IT Operations Manager was responsible for the following in a retail enterprise that accepted credit-cards for payment:

- A Point-of-Sale (POS) application used in hundreds of stores across the country;

- An e-commerce application that required credit-card numbers for payment;
- A payment-processing application in the back-office that communicated with the credit-card network for settling transactions;
- A back-office database that consolidated transactions for accounting; and
- A business analytics application for determining retail fraud;

the IT Operations manager would have five applications that required encryption and thus, key-management. With the proliferation of laptop and PDA losses or thefts, companies are now mandating encryption on these devices, thus adding two more key-management schemes to the infrastructure. Finally, add database and operating system-specific encryption to the mix, and you round out the picture with 8-10 key-management infrastructures.

Since applications are typically purchased from multiple vendors, each vendor, focusing primarily on their own business application, implements encryption and perform key-management functions using its own design. As a result, the IT Operations staff are forced to manage at least 8-10 distinct symmetric key-management infrastructures, each with its own technology, training, documentation, procedures and audits (PCI-DSS regulated entities are required to perform annual audits of any system that stores credit-cards).

Not only does this border on the ridiculous as it raises Total Cost of Ownership (TCO) for the company, but more importantly, one could argue that there is the potential for a compromise in such a security strategy because "with so many pots cooking on the stove simultaneously, something is bound to get burned".

Presented with the problem in this perspective, the logical solution springs to clarity: the key-management capability needs to be abstracted from applications that need the capability, and managed independently in its own infrastructure. Applications need only have access to a key-management service, thus enabling encryption and decryption, without having to be aware of implementation details. Such a solution is not unlike the architecture of the Domain Name System (DNS) for hostname-IP-address resolution, the Dynamic Host Configuration Protocol (DHCP) for dynamic IP-address allocation or a Relational Database Management System (RDBMS) for data management.

2.1 File, Database and Disk-based Encryption

There are many commercial technologies on the market today that are capable of transparently encrypting data at the file, database or storage-media layer. These technologies, which include encrypting file systems, full-drive encryption, database encryption, etc., have their own built-in key-management.

In the presence these technologies what advantage would an abstract key-management system offer implementers?

Aside from the issue that current implementations of such technologies offer incompatible key-management designs , the single biggest issue with such techniques is that it does not address the problem completely.

An application, typically, consists of many layered technologies in a stack, through which data must pass before it is stored on storage media. The layers vary depending on the complexity of the application, its architecture, operating system and physical implementation. File, database and storage-media encryption occur at some of the lowest layers of such a technology stack, oblivious to the applications that create the data. Implementers of such encryption schemes offer this feature as a benefit, because it does not require applications to be modified to take advantage of the encryption capabilities; all applications can immediately start using it.

However, the lower in the technology stack the encryption occurs, it leaves open the possibility for attackers to compromise a layer higher up in the stack and avail themselves to plaintext data. Even in a perfectly functioning encryption system (consisting of file, database or media-based encryption),, data could be compromised in one of many layers above the encrypting layer.

On the contrary, encrypting at the application layer allows implementers to encrypt data at the highest possible layer in the stack, leaving little "wiggle-room" for the attacker to compromise the data. While there is no guarantee that an attacker cannot compromise the application layer and still get to the data before it is encrypted (or after it is decrypted), it, nonetheless, reduces the attack surface to the smallest possible target within the stack. It allows implementers to focus their resources on protecting just one layer - the application layer - towards making the most effective use of encryption.

Encrypting at the application-layer also has the added benefit, that once data is encrypted at this highest stack-layer, implementers need have little concern for the safety of data in lower layers of the stack: the data is protected no matter how many technology layers it must pass through on the host or the network.

3. ARCHITECTURE

An Enterprise Key Management Infrastructure (EKMI) is defined as a collection of technology, policies and procedures for managing all cryptographic keys within an enter-

prise – both symmetric and asymmetric. Therefore, an EKMI consists of a PKI and an SKMS.

Note: In the short-to-medium term, PKI and SKMS will be managed as distinct entities within enterprises. However, the two infrastructures will merge towards a cohesive infrastructure in the not-too-distant future. This paper dispenses with detailed discussions of PKI except where it integrates with an SKMS, and focuses more on the discussion of the SKMS itself.

The PKI part of an EKMI is a standard public key infrastructure issuing and managing X.509 and PKIX-compliant digital certificates. It uses industry-standard protocols for communicating with client requests: CMS, PKCS#10, PKCS#7, etc., and the digital certificates issued out of the PKI are managed per the PKI's Certificate Policy (CP) and Certification Practices Statement (CPS). Nothing unusual here, so we won't dwell on this too much. We will point out where digital certificates from a PKI are needed to enable the capability described in this paper.

The SKMS, on the other hand, has an architecture based on the the following business, technical and operational requirements:

- Centralized policy-definition and key-management;
- Platform, application and language independent;
- Highly-available; yet KM client applications were required to continue functioning - i.e., encrypt and decrypt data - even in the absence of the KM service;
- Highly-scalable;
- Very secure;
- Leverage existing standards and security certifications of cryptographic components;

Given these requirements, the following design decisions were made for the SKMS architecture:

3.1 Client-Server

While key-management can be easily abstracted from applications even while running locally on the same machine as the application, the requirement that KM policies be defined centrally and symmetric keys be managed centrally led the design towards a client-server architecture for key-management.

The client is implemented as a library, named the **Symmetric Key Client Library (SKCL),** and is much like the name-service library in DNS or the database connectivity libraries - ODBC, JDBC, etc. - for RDBMS. Client applications use the SKCL to request and receive KM services from the server.

The **Symmetric Key Services (SKS)** server functions as a centralized service-provider on the network, listening for and responding to KM requests. When requested, it generates all keys centrally based on predefined policies, escrows them, and then sends the symmetric key to the authorized client. The client and server communicate with each other using a secure protocol: **Symmetric Key Services Markup Language (SKSML)** (discussed later).

By using the client-server architecture, not only are the centralized policy definition requirements addressed, but also the centralized key-management requirements.
An alternative architecture is to define policies centrally and push them down to the clients, and similarly have the clients generate keys locally and push them up to the server. However, the SKMS architecture avoided this design for one reason: to avoid the possibility of catastrophic data-loss.

If a client were to generate a symmetric key locally, encrypt the plaintext, delete the plaintext (to eliminate the vulnerability), but cannot persist or send the generated symmetric key to the server for any reason, the plaintext might be lost forever.

While it is possible to design around such conditions, the complexity of the SKCL increases significantly because it is difficult to predict potential catastrophic conditions on a client machine - especially mobile devices. With centralized policy-definition and key-generation, this loss is avoided altogether by escrowing the symmetric key first, and then sending it to the client for use.

Given that a client-server architecture makes it possible to have multiple servers servicing numerous clients using hundreds of application, the following schema was chosen to uniquely identify every symmetric key in the system. This is how applications will map the symmetric key they've used, to the ciphertext:

- Every enterprise is assigned a unique **Domain-ID** - a monotonically increasing sequential number - to distinguish its SKMS from others'. The design chose to reuse the Private Enterprise Numbers assigned by IANA[13] for this purpose. While using DNS-style domain-names was an option, the design opted for sequential numbers as it was similar to other identifiers in the SKMS; besides, since humans would rarely need to know these identifiers, it was more efficient to maintain them as numerals;
- Every server within an SKMS is also assigned a unique **Server-ID** - once again, a monotonically increasing sequential number - to distinguish a server from others within an SKMS domain;
- Every symmetric key generated by a server is assigned a unique **Key-ID** - a monotonically increasing sequen-

tial number - to distinguish it from every other key generated by that SKS server;

Concatenating the Domain ID, the Server ID and the Key ID, with simple hyphens to separate them, allows the SKMS to create a unique **Global Key-ID (GKID)** that can be referenced by a client, anywhere on the internet. An example of a GKID would be **10514-3-34348**, indicating that this is key ID 34348, generated on SKS server number 3 within an SKMS for the organization whose domain is represented by the unique private enterprise number, 10514, as issued by IANA.

It is necessary for applications to be modified to maintain the GKID with the ciphertext so that applications know which symmetric key to use to decrypt the ciphertext. For applications that use a structured database of any kind, this is relatively easy by modifying the database schema as a one-time enhancement activity. For standalone ciphertext files, it is recommended to use the World Wide Web Consortium's XML Encryption standard whose schema allows for identifying the unique identifier of the encryption key, the URL for locating the key and many other encryption parameters along with the ciphertext. The implementation experience described in Section 5 shows how this design was used successfully for relational databases as well as standalone ciphertext files.

3.2 XML-based protocol

To support platform, application and programming-language independence, the SKCL and the SKS server communicate using an eXtensible Markup Language (XML)-based protocol, named the **Symmetric Key Services Markup Language (SKSML)**. Details of this protocol are defined in the next section.

SKSML is designed to be a very thin layer, leveraging the Simple Object Access Protocol (SOAP) for sending and receiving messages. SOAP is well-understood, accepted and supported industry-standard for sending and receiving complex messages in client-server communications. SOAP libraries are available for every major programming language and platform, allowing almost every modern operating system to support applications using SOAP and thus, SKSML.

The question arises - why was Abstract Syntax Notation (ASN) not used for the protocol?

The PKI community has been using ASN, successfully, across all major platforms for a number of years. ASN is well-understood, accepted and supported by the PKI community and has the advantage of being compact, thus allowing for significant efficiencies in communication, as well as by small devices where space is at a premium.

Despite these advantages of ASN, the design chose to use XML for the protocol for the following reasons:

i. XML is equally well-understood, accepted and supported by the *general computing community* - not just the PKI or protocol-development community;
ii. It is extremely easy to understand and explain XML - even to non-computer professionals;
iii. There are significantly larger number of developers who know and use XML than ASN, thereby increasing the probability of adoption and making a larger pool of candidates available for its development;
iv. XML is the lingua-franca of Service Oriented Architecture (SOA) - the architecture for a new class of applications that has the support of every major software vendor in the world;
v. XML does not require special tools; its messages can be assembled and debugged using simple text editors;
vi. While ASN definitely is more compact than XML, with the exception of a small number of use-cases, the verbosity of XML is not a handicap in an environment whose bandwidth only keeps increasing every few years. With gigabit networking now showing up in new computers for LANs, 108Mb/s for WiFi-enabled devices expected to become the standard by the end of this decade, and broadband connectivity to be near the 1Mb/s range for small mobile devices, bandwidth is not a constraint for the vast majority of computing devices.

Primarily driven by the ease-of-use feature and industry trends, it was decided to use XML for the representation of the key-management protocol.

3.3 Scalability and Availability

There are well-known and proven design architectures for creating highly-scalable and highly-available software systems for servers. However, most of the work in creating such software requires addressing basic infrastructure requirements such as authentication, authorization, logging, performance management, scheduling, persistence, etc.

This architecture chose to leverage the capability built into the Java 2 Enterprise Edition (J2EE) for these services rather than develop them from scratch.

The J2EE architecture was created to address precisely such infrastructure issues, while scaling to address the requirements of extremely large and demanding infrastructures. It has evolved over the last decade and continues to improve with the input of millions of Java developers. Java Enterprise Edition 5 (JEE5), the latest incarnation of this architecture, has once again improved on this design through the community-development process.

By choosing J2EE, and the newer JEE5, the architecture addressed every infrastructure requirement for a scalable and highly-available service, allowing the designers to concentrate on the core functionality required within an SKMS.

On the client side, scalability of the SKCL isn't expected to be a major issue (even for an e-commerce transaction server, which would still be an SKMS client) because the client library would execute within a thread of the client application. It is expected that the designers of the client application will have addressed performance issues for their application in general, and the SKCL will benefit from those design improvements.

The SKMS architecture abstracted the cryptographic processing of the SKMS service to execute outside the service. This allows implementations to use third-party cryptographic accelerators for enhancing the performance of SKMS clients and servers, if desired. Using well-known interfaces such as Public Key Cryptography Standard #11, Cryptography API (CAPI) and Java Cryptography Extension (JCE) allows the SKMS architecture to provide flexibility to implementers for dealing with scalability.

The architecture defines the notion of a **"global" SKS (GSKS)** server, used for defining policy for the SKMS domain. All other SKS servers, named "local" SKS servers, are expected to replicate symmetric keys generated by them to the GSKS. Since every symmetric key has a unique GKID within an SKMS domain, a single GSKS, appropriately sized, is capable of storing every local SKS servers' symmetric keys for redundancy. The GSKS itself never generates any symmetric keys itself; it only serves as a centralized repository within an SKMS infrastructure. In the event of an outage of a local SKS server, the client just contacts the GSKS and requests the symmetric key, much as the nameservice library contacts a different DNS server when the first is unavailable. SKMS implementations must ensure that they have implemented sufficiently redundant GSKS servers to accommodate their business requirements.

Finally, to ensure clients can continue processing – encrypt and decrypt – even in the face of network failures, the SKMS architecture includes "secure key-caching" on the client side. How and when a client may cache keys is based on "key-caching policies" defined on the SKS server, centrally. All SKMS clients can be directed by centralized "key-caching policies" to either cache or not cache symmetric keys on the client-side, securely. While implementers have the flexibility to choose the strategy that works best for them, this design uses the "message-level" security built into the SKMS architecture to enable secure key-caching.

3.4 Application Integration

Since the SKCL is merely a library, it is necessary for an application to integrate the SKCL to take advantage of the benefits offered by an SKMS. However, unlike the DNS model, an application cannot just use the SKCL, acquire the symmetric key, use it and carry on without maintaining some state of the operation it performed.

In order for an application to be able to decrypt its ciphertext, it must maintain knowledge of the GKID of the symmetric key it used in the cryptographic operation. Without the GKID, the application will never know which symmetric key to request from the SKMS.

This architecture recommends the modification of the database schema of the application, to include the storage of the GKID in the same database or file where the ciphertext is stored. If the symmetric key was requested from a non-default domain – one in which the client normally does not belong – the client application may also need to maintain the URL of the SKS service where the key can be located.

For example, where a database schema before SKMS integration might look like the following:

Employee table

Name	Type	Comment
ID	Long	Unique identifier
Name	Char	
DOB	Date	
SSN	Char	
.....		

After including the GKID, the schema might resemble the following:

Modified Employee table

Name	Type	Comment
ID	Long	Unique identifier
GKID	Char	Unique key identifier
NAME	Char	
DOB_CIPHERTEXT	Date	
SSN_CIPHERTEXT	Char	
SSN_SHA256	Char	SHA-256 hash of SSN
.....		

By associating a GKID with every ciphertext in the database, an application can now retrieve the required sym-

metric key from an SKMS to decrypt data when it needs it. However, when ciphertext is transmitted from one application to another, it must carry the GKID with it. It is recommended that the W3C XML Encryption standard be used for the transmittal of such information, since the XML Encryption schema allows for specifying details such as the GKID, SKS server, etc., along with the ciphertext.

An alternative choice to this design was to use a complex data structure in the ciphertext – such as the Cryptographic Message Syntax (CMS) – to embed many cryptographic elements into a binary "blob" that can be stored within a single data element in the database. This has the advantage of carrying many required pieces of information for cryptographic processing, together.

However, this design avoids this for the following reasons:

i. Most applications expected to use the SKMS will be the traditional client-server, enterprise applications that use an RDBMS for data-storage. Developers of such applications expect to see the database schema laid out in its entirety rather than to have data structures collapsed into blobs within a single element;
ii. Collapsing data-structures into a single element hides information that might help improve performance of the client application. For instance, if the application determined that the same GKID was in use for the next 100 records, it might be able to cryptographically process the next 100 records after fetching the symmetric key once, as opposed to fetching it once per record;
iii. It allows the application developers to use standards such as XML Encryption by "exporting" the data elements into XML elements more easily and efficiently than if additional processing had to be performed to "explode" the data-structure for conversion to XML. Once again, this design assumes that XML significantly eases the adoption of complex technologies by neophytes, so decisions were made to favor XML.

Applications link in the SKCL as any other library in the standard software development process. Once linked in, the client application need two additional pieces of information before they can make requests of an SKS server:

1. A list of the SKS/GSKS servers that the client may contact when it needs SKMS services. This information is provided in a text file (much like the /etc/resolv.conf file for DNS clients) and simply lists the URL's of the SKS/GSKS servers that the client may contact.
2. A digital certificate with a corresponding key-pair to be used for signing requests and for decrypting server responses. The key-pair and digital certificate is provisioned to the client in an out-of-band process using any of the traditional PKI mechanisms. It is assumed that the organization implementing the SKMS has the use of a PKI for this purpose.

With this, the client application is now ready to make requests for symmetric key services from the SKMS.

3.5 Security

Given that a centralized key-management system would be the ultimate treasure trove to attackers, it is incumbent that the architecture consider known threats and address them with appropriate counter-measures. The design of the SKMS addresses a number of threats by using "message-level" security in the messages that were in transit or during rest.

Message-level security addresses the following vulnerabilities in the following manner:

- **Authenticity of requests and responses**: Every message arriving at the client or server is assumed to be from an untrusted source. As such, the architecture requires that every message be digitally signed and verified against a trusted certificate hierarchy known to clients and servers within the SKMS. This requires every SKMS client and server be provisioned with an X.509/PKIX-compliant digital certificate for signing requests and responses. This ensures that clients and servers are dealing with only properly validated messages;
- **Integrity of requests and responses**: To ensure that the messages sent and received by SKMS clients and servers do not lose their integrity either accidentally or through a deliberate attack, all SKSML messages use digital signatures to verify the integrity of the messages. These are the same signatures used for authenticity checking;
- **Confidentiality**: Since an SKMS deals with the most sensitive data within an IT infrastructure, it is incumbent on the architecture to protect the payload to the fullest extent possible. **For transmission** of the symmetric key to clients, the design uses public-key encryption, using the digital certificate of the recipient. Since the client is expected to be the only one having access to the private key corresponding to the targeted digital certificate, the payload can only be accessible to authorized recipients. **For storage and recovery** of the symmetric key within the databases, the SKMS architecture uses the public-key in the digital certificate of the SKS server to protect the payload. The architecture also permits the creation of "global" SKS servers, and Security Officers, with their own digital certificates, so that the symmetric key is encrypted with their public-keys too. This allows the symmetric key to be recovered, through an appropriate process, by entities other than the SKS server that generated the symmetric key.

- **Database access**: The SKMS architecture uses an RDBMS to store its data. Traditionally, Database Administrators (DBA) have privileged access to such databases over application administrators and security officers. To ensure that the SKMS database contents are protected from unauthorized manipulation, the SKMS design protects database objects with digital signatures. For example, when SKMS objects are stored in the database, a new digital signature is computed and stored with the object by the SKS server. When reading the object back from the database, the SKS server verifies its digital signature to ensure that the integrity of the message has not been compromised.
- **Key protection**: The astute reader will have recognized that the SKMS architecture relies on the cryptographic key-pair of the clients and servers to preserve the integrity of the infrastructure. This is correct. To ensure these keys are protected, the SKMS uses well-established interfaces to external cryptographic tokens – such as PKCS#11, CAPI and JCE – to take advantage of FIPS 140-2 validated devices, as necessary. By controlling access to the private keys through mechanisms provided in these tokens, an SKMS implementation can extend the security of its infrastructure beyond the boundary of the SKMS client or server itself.

4. SKSML PROTOCOL

The heart of the SKMS is in its protocol – the Symmetric Key Services Markup Language (SKSML). SKSML consists of the following types of messages:

- A request for a single new symmetric key;
- A request for multiple new symmetric keys;
- A request for a single existing symmetric key;
- A request for multiple existing symmetric keys;
- A request for a key-caching policy object;
- A response with a single symmetric key;
- A response with multiple symmetric keys;
- A response with a key-caching policy object;
- A response with a fault message;

Each of these SKMS messages is wrapped in a Web Services Security (WSS) header, which provides the digital signature and encryption capabilities at the message layer using SOAP for object-encapsulation. The use of the WSS standard allowed the SKMS to focus on application functionality rather than on the mechanical details of security messages.

All SKSML messages are in XML, and use the XML Schema Definition (XSD) language to provide the semantic rules for how messages must be constructed.

(*Note: This protocol is currently working its way through the standards process at the Organization for the Advancement of Structured Information Standards (OASIS), and has not been finalized yet. It is, however, targeting for anticipated standards status by the summer of 2008*).

4.1 Request for a single new symmetric key

This request forms the most basic request – for a single new symmetric key without specifying a key-class. (*Note: Key-classes are designations that allow clients to request keys conforming to specific business requirements*.) The abbreviated request (without the WSS header) is as follows:

```
<ekmi:SymkeyRequest
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01>
  <ekmi:GKID>0-0-0</ekmi:GKID>
</ekmi:SymkeyRequest>
```

The highlighted line - **<ekmi:GKID>0-0-0</ekmi:GKID>** - essentially specifies a GKID where the DomainID, ServerID and KeyID are all zeros. This special value is an indicator to the server that the request is for a new symmetric key (since all existing keys would have non-zero GKIDs).

The fact that the request does not specify a key-class does not mean that the returned symmetric key does not belong to a specific key-class; the SKS server will simply generate and return a symmetric key of the "Default" key-class for the specific requester.

A request for a single symmetric key of a specific key-class (without the WSS header), is as follows:

```
<ekmi:SymkeyRequest
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01>
  <ekmi:GKID>0-0-0</ekmi:GKID>
  <ekmi:KeyClasses>
      <ekmi:KeyClass>HR-Class</ekmi:KeyClass>
  </ekmi:KeyClasses>
</ekmi:SymkeyRequest>
```

In this case, the client has requested a single symmetric key of the "HR-Class" key-class. The meaning of "HR-Class" is application-defined. It is assumed that the creators of security policies within the SKMS have defined "HR-Class" to mean the issuance of a specific type of symmetric key with specific permissions (to be discussed later).

4.2 Request for a multiple new symmetric keys

While the request for a single symmetric key could dispense with specifying the key-class in the request, a request for more than a single symmetric key must specify the key-classes of the requested keys in the symmetric key request, as follows:

```
<ekmi:SymkeyRequest
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01>
  <ekmi:GKID>0-0-0</ekmi:GKID>
  <ekmi:KeyClasses>
     <ekmi:KeyClass>EHR-CDC</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-CRO</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-DEF</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-EMT</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-HOS</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-INS</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-NUR</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-PAT</ekmi:KeyClass>
     <ekmi:KeyClass>EHR-PHY</ekmi:KeyClass>
  </ekmi:KeyClasses>
</ekmi:SymkeyRequest>
```

In this request, the client application (assumed to be some Electronic Health Record application) is requesting nine (9) new symmetric keys, each corresponding to the named key-class. It is assumed that the EHR application is choosing to encrypt different parts of this new health record with different symmetric keys, so that target users of this EHR will need to request only a subset of the nine symmetric keys to view data meaningful to their business process.

For example, when this patient's health record is stored, encrypted with nine symmetric keys, applications used by nurses will be authorized to request only keys that conform to the EHR-DEF (Default) and the EHR-NUR (Nurse) key-classes. This allows them to perform their work without compromising the security of other parts of this EHR.

It is possible for an application to request multiple new symmetric keys of the same class with the following request – for three new keys of the "ATM-Class" key-class:

```
<ekmi:SymkeyRequest
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01>
  <ekmi:GKID>0-0-0</ekmi:GKID>
  <ekmi:KeyClasses>
     <ekmi:KeyClass>ATM-Class</ekmi:KeyClass>
     <ekmi:KeyClass>ATM-Class</ekmi:KeyClass>
     <ekmi:KeyClass>ATM-Class</ekmi:KeyClass>
  </ekmi:KeyClasses>
</ekmi:SymkeyRequest>
```

4.3 Request for a single existing symmetric key

The following shows a request for a single symmetric key, that was previously generated and escrowed, and is now being requested by the application to decrypt ciphertext. It is assumed that the requester has the authorization to receive this key, otherwise it would be pointless to make the request. In the following example, the client application is requesting a symmetric key with a GKID of 10514-1-23 (the DomainID is 10514, ServerID is 1 and the KeyID is 23):

```
<ekmi:SymkeyRequest
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01>
  <ekmi:GKID>10514-1-23</ekmi:GKID>
</ekmi:SymkeyRequest>
```

4.4 Request for a multiple existing symmetric keys

These requests resemble the following:

```
<ekmi:SymkeyRequest
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01>
  <ekmi:GKID>10514-1-2783</ekmi:GKID>
  <ekmi:GKID>10514-3-532</ekmi:GKID>
  <ekmi:GKID>10514-2-1423</ekmi:GKID>
  <ekmi:GKID>10514-6-243</ekmi:GKID>
</ekmi:SymkeyRequest>
```

Not only is each GKID individually named, but there is no need to specify a key-class, since the returned key is going to belong to whatever class it belonged to, at the time of key-generation.

4.5 Request for a key-caching policy object

The only request type that does not request a symmetric key, is when a client needs to know its key-caching policy (KCP). This is normal in three situations:

i. When the client has been connected to the SKMS for the first time and is making its first request for a symmetric key and must know its caching policy before it can request a key;
ii. The current key-caching policy on the client machine has expired and the client must refresh it to get new policy information;
iii. The key-caching policy object on the client is either corrupted, or the integrity of the object cannot be verified;

In all cases, the client machine sends the following KCP request (shown without the WSS header):

```
<ekmi:KCPRequest
  xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01"/>
```

As one can notice, the KCP request is empty.

However, the WSS header that carries the digital signature of the requesting client is all that the SKS server needs, to determine the authenticity of the request as well as the identity of the requester. Based on this deduced & verified information, the SKS server is able to determine the precise key-caching policy that applies to this requester and respond accordingly.

4.6 Response with a single symmetric key

In response to a request for a single symmetric key, whether new or existing, the SKS server responds with the following SKSML (shown without the WSS header):

```
<ekmi:SymkeyResponse
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01'
  xmlns:xenc='http://www.w3.org/2001/04/xmlenc#'
  xsi:schemaLocation=
      'http://docs.oasis-open.org/ekmi/2008/01
  symkeyResponse.xsd'>

  <ekmi:Symkey>
    <ekmi:GKID>10514-1-235</ekmi:GKID>
    <ekmi:KeyUsePolicy>
      <ekmi:KUPID>10514-4</ekmi:KUPID>
      <ekmi:PolicyName>DES-EDE Policy</ekmi:PolicyName>
      <ekmi:KeyClass>HR-Class</ekmi:KeyClass>
      <ekmi:KeyAlgorithm>
         http://www.w3.org/2001/04/xmlenc#tripledes-cbc
      </ekmi:KeyAlgorithm>
      <ekmi:KeySize>192</ekmi:KeySize>
      <ekmi:Status>Active</ekmi:Status>
      <ekmi:Permissions>
        <ekmi:PermittedApplications>
          <ekmi:PermittedApplication>
            <ekmi:ID>10514-23</ekmi:ID>
            <ekmi:ApplicationName>
                 Payroll Application
            </ekmi:ApplicationName>
            <ekmi:Version>1.0</ekmi:Version>
            <ekmi:DigestAlgorithm>
                 http://www.w3.org/2000/09/xmldsig#sha1
            </ekmi:DigestAlgorithm>
            <ekmi:DigestValue>
                 NIG4bKkt4cziEqFFuOoBTM81efU=
            </ekmi:DigestValue>
          </ekmi:PermittedApplication>
        </ekmi:PermittedApplications>
        <ekmi:PermittedDates>
          <ekmi:PermittedDate>
            <ekmi:StartDate>2007-01-01</ekmi:StartDate>
            <ekmi:EndDate>2007-12-31</ekmi:EndDate>
          </ekmi:PermittedDate>
        </ekmi:PermittedDates>
        <ekmi:PermittedTimes>
          <ekmi:PermittedTime>
            <ekmi:StartTime>07:00:00</ekmi:StartTime>
            <ekmi:EndTime>19:00:00</ekmi:EndTime>
          </ekmi:PermittedTime>
        </ekmi:PermittedTimes>
      </ekmi:Permissions>
    </ekmi:KeyUsePolicy>
    <ekmi:EncryptionMethod Algorithm=
         "http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
    <xenc:CipherData>
      <xenc:CipherValue>
         E9zWB/y93hVSzeTLiDcQoDxmlNxTuxSffMNwCJmt1dIqzQH
         BnpdQ81g6DKdkCFjJMhQhywCx9sfYjv9h5FDqUiQXGOca8E
         U871zBoXBjDxjfg1pU8tGFbpWZcd/ATpJD/UJow/qimxi8+
         huUYJMtaGHtXuL1Wtx27STRcRpIsY=
      </xenc:CipherValue>
    </xenc:CipherData>
  </ekmi:Symkey>
</ekmi:SymkeyResponse>
```

A successful symmetric-key response (SymkeyResponse) contains one or more Symkey elements, each of which contains the encrypted symmetric key and an associated **KeyUsePolicy (KUP)**. The symmetric key is encrypted with the recipients' public key so that only the targeted recipient of the response message may decrypt it.

The KUP object provides detailed information on how the SKCL may use the associated symmetric key. Other than some meta-data, the heart of the KUP is in the **Permissions** element. SKSML allows SKMS implementations to restrict the use of symmetric keys by specifying a complex permissions-model that permits the use of the symmetric key when the conditions in the permissions-model are satisfied.

The Permissions element in SKSML allows SKMS implementations to restrict the use of the symmetric key based on one or more of the following categories:

- Permitted Applications;
- Permitted Dates;
- Permitted Durations – i.e., between any two date-times;
- Permitted Levels – for multi-level security (MLS) aware systems;
- Permitted Locations – that can be based on GPS coordinates;
- Permitted Times;
- Permitted Transactions – i.e., the actual number of encrypted transactions permitted with a key; and
- Permitted Uses;

If a permission appears for a specific category in the KUP, the SKCL enforces the use of the key according to that permission. If a permission does not appear for a specific category, the key *can be used* within that category without restrictions. For example, if two (2) applications are explicitly listed within *PermittedApplications*, then only the listed applications are authorized to use the symmetric key. However, if the *PermittedApplications* category is missing from the KUP, then *all* applications are allowed the use of the symmetric key within that client.

While this is an atypical method of granting use, it permits the KUP to be minimal when granting broad access to keys. Otherwise, KUPs might tend to become overly verbose and complex.

4.7 Response with multiple symmetric keys

In response to a request for multiple symmetric keys - new or existing - the SKS server responds with the following SKSML (shown without the WSS header):

```
<ekmi:SymkeyResponse
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01'
  xmlns:xenc='http://www.w3.org/2001/04/xmlenc#'
  xsi:schemaLocation=
      'http://docs.oasis-open.org/ekmi/2008/01
  symkeyResponse.xsd'>

  <ekmi:Symkey>
     <ekmi:GKID>10514-4-1235</ekmi:GKID>
     (Content removed for conciseness).
  </ekmi:Symkey>
  <ekmi:Symkey>
     <ekmi:GKID>10514-1-2385</ekmi:GKID>
     (Content removed for conciseness).
  </ekmi:Symkey>
  <ekmi:Symkey>
     <ekmi:GKID>10514-3-1237</ekmi:GKID>
     (Content removed for conciseness).
```

```
  </ekmi:Symkey>
  <ekmi:Symkey>
     <ekmi:GKID>10514-4-1238</ekmi:GKID>
     (Content removed for conciseness).
  </ekmi:Symkey>
</ekmi:SymkeyResponse>
```

The content inside each Symkey element is similar to the response presented in the earlier section (Response with a single symmetric key). The SKCL parses through each Symkey and processes it as if it were a single-key response.

4.8 Response with a key-caching object

In response to a request for a **KeyCachingPolicy (KCP)** the SKS server responds with the following SKSML (shown without the WSS header):

```
<ekmi:KeyCachePolicy
   xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01'
   xsi:schemaLocation=
         'http://docs.oasis-open.org/ekmi/2008/01
   EKMICoreLibrary.xsd'>

   <ekmi:KCPID>10514-17</ekmi:KCPID>
   <ekmi:PolicyName>
        Corporate Laptop Symmetric Key Caching Policy
   </ekmi:PolicyName>
   <ekmi:Description>
      This policy defines how company-issued laptops
      will manage symmetric keys used for file/disk
      encryption in their local cache.  This policy must
      be used by all laptops that use the company EKMI.
   </ekmi:Description>
   <ekmi:StartDate>2008-01-01T00:00:01</ekmi:StartDate>
   <ekmi:EndDate>2008-12-31T24:00:00</ekmi:EndDate>
   <ekmi:PolicyCheckInterval>
      86400
   </ekmi:PolicyCheckInterval>
   <ekmi:Status>Active</ekmi:Status>
   <ekmi:NewKeysCacheDetail>
     <ekmi:MaximumKeys>3</ekmi:MaximumKeys>
     <ekmi:MaximumDuration>86400</ekmi:MaximumDuration>
   </ekmi:NewKeysCacheDetail>
   <ekmi:UsedKeysCacheDetail>
     <ekmi:MaximumKeys>3</ekmi:MaximumKeys>
     <ekmi:MaximumDuration>86400</ekmi:MaximumDuration>
   </ekmi:UsedKeysCacheDetail>
</ekmi:KeyCachePolicy>
```

The KCP essentially tells the client machine how many new and used symmetric keys (used for at least one encryption transaction) it may cache and for how long. This policy is defined centrally for individual, group and/or all clients requesting key-management services. The *PolicyCheckInterval* tells the client how frequently it must check back with the SKS server for updates to the KCP. This is to ensure that client machines' caching policies can be changed centrally with a small notice period.

4.9 Response with a fault message

In the event the SKS server cannot respond to a request for one or more symmetric keys successfully, it sends back the following SKSML (shown without the WSS header):

```
<ekmi:SymkeyResponse
   xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01'
   xsi:schemaLocation=
      'http://docs.oasis-open.org/ekmi/2008/01
   symkeyResponse.xsd'>

   <ekmi:SymkeyError>
      <ekmi:RequestedGKID>0-0-0</ekmi:RequestedGKID>
      <ekmi:RequestedKeyClass>
         HR-Class
      </ekmi:RequestedKeyClass>
      <ekmi:ErrorCode>9025</ekmi:ErrorCode>
      <ekmi:ErrorMessage>
         A KeyUsePolicy to issue a symmetric key with
         the requested key-class does not exist for
         this request.  Please contact your Security
         Officer if you have any questions.  Provide
         them the following information if asked:
         SRID: 10514-2-8643
      </ekmi:ErrorMessage>
   </ekmi:SymkeyError>
</ekmi:SymkeyResponse>
```

The response provides a reference to the requested GKID and key-class for the client application to correlate the error message with its request. It is up to the application to determine how to process the *ErrorCode* and *ErrorMessage* elements.

A response for multiple symmetric keys that results in partial success will return the following SKSML that includes both symmetric keys and SymkeyError's:

```
<ekmi:SymkeyResponse
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01'
  xmlns:xenc='http://www.w3.org/2001/04/xmlenc#'
  xsi:schemaLocation=
      'http://docs.oasis-open.org/ekmi/2008/01
  symkeyResponse.xsd'>

  <ekmi:Symkey></ekmi:Symkey>
  <ekmi:Symkey></ekmi:Symkey>
  <ekmi:Symkey></ekmi:Symkey>
  <ekmi:SymkeyError></ekmi:SymkeyError>
  <ekmi:SymkeyError></ekmi:SymkeyError>

</ekmi:SymkeyResponse>
```

Applications are expected to keep track of which requests received successful responses, which ones did not, and how to deal with the mixed result.

5. IMPLEMENTATION EXPERIENCE

Based on the design and protocol described in earlier sections, an open-source software implementation of this architecture[14] was released on the Internet in 2006.

The SKMS consisted of two centralized SKS servers – a primary and a disaster recovery server – and any number of clients using the **Symmetric Key Client Library (SKCL)** to request services from the SKS servers. (While they are referred to as clients, the client software may themselves be database servers, web-servers, application-servers and/or any business application).

The SKSML protocol implemented in this software is based on a DRAFT 1.0 version of the protocol (which is a little different from the DRAFT 3.0 protocol described earlier in the paper). This protocol is currently going through a standardization process at OASIS[15].

Each implemented SKS server consisted of:

- a server-class computer running an operating system – typically Linux, UNIX or Windows - that had a compliant Java Virtual Machine (JVM) available for it;
- a relational database to serve as the storehouse for the symmetric encryption keys;
- a J2EE-compliant application server to host the application that would respond to requests over the network, serving as the workhorse of the SKMS;
- a JCE-compliant cryptographic provider to perform the cryptographic operations of key-generation, key-protection, digital signing, verification, etc.;
- an optional, **but strongly recommended,** Hardware Security Module (HSM) or Trusted Platform Module (TPM) for securely storing the cryptographic keys that protect the database's contents;
- the SKS server software itself, consisting of an Enterprise Archive (EAR) and a Web Archive (WAR) file for the administration console, along with ancillary utilities;

Each SKCL client platform consisted of:

- a client machine running an operating system – once again, typically, Linux, UNIX or Windows, but included the OS/400 - that had a compliant Java Virtual Machine (JVM) available for it;
- a JCE-compliant cryptographic provider to perform the cryptographic operations of encryption, decryption, digital signing, verification, etc.;
- an optional, but highly recommended, Trusted Platform Module (TPM), smartcard or other USB-based cryptographic token for securely storing the cryptographic keys that protect the clients' authentication credentials;
- the SKCL software itself, consisting of an API callable by Java applications for communicating with the SKS server and performing cryptographic functions (non-Java applications used a Java Native Interface (JNI) library to call the SKCL);

To exercise the protocol a client utility called “xenc” was created that would encrypt files, directories and data in relational database tables. Using xenc and the implemented architecture, we were successfully able to demonstrate the request of symmetric encryption keys from dissimilar client platforms and receive symmetric keys based on predefined policies at the server. The encrypted data was stored in the W3C XML Encryption standard for compatibility and transferred to other platform machines, where another successful call by xenc retrieved the required symmetric key from the SKS server and decrypted the data successfully.

Key-caching was tested by first getting the key-caching policy from the SKS server. This led to the SKCL requesting and receiving symmetric keys from the SKS server to conform to the KCP. Once the keys were cached on the client, the client was disconnected from the network and xenc was used to successfully encrypt and decrypt files on the local client.

6. SECURITY

Given the sensitivity of the information managed within the SKMS implementation, the infrastructure was predicated on an extraordinary level of security. (*As with any security architecture, the controls and procedures in place at the implementation determined the degree of vulnerability the SKMS had against attacks; we still continued to configure the firewall and other operating system controls to secure the machine.*).

The implemented SKMS incorporated the following security features:

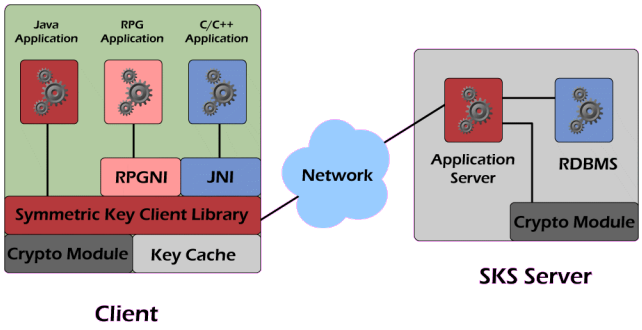
- all symmetric keys were generated using multiple compliant cryptographic providers – some hardware and others in software;
- all symmetric encryption keys were themselves, encrypted using multiple RSA asymmetric keys – one belonging to the SKS server, one to the GSKS and one ot the Security Officer;
- all database records on the SKS server were digitally signed before storage, and verified upon retrieval to ensure their integrity hadn't been compromised;
- all administrative operations through the console were digitally signed and maintained in a history log for audit purposes, and verified upon retrieval;
- all administrative operations through the console required SSL/TLS-based client-authentication;
- only digitally signed client-requests were accepted by the SKS server from SKCL clients;
- only digitally signed responses from the SKS server were accepted by SKCL clients;
- all symmetric keys were transported, encrypted for the specific client making the request;
- all cached-keys on the client were digitally signed and encrypted on storage, decrypted and verified upon retrieval to ensure their integrity;

To have this level of security enabled within the SKMS, and to ensure that this security could scale to internet levels, the architects of the open-source SKMS software predicated the use of a PKI to secure the SKMS.

The PKI allowed the implementers to manage large numbers of digital certificates much more easily than managing raw asymmetric cryptographic keys. With the use of a PKI, every SKMS client and server was issued a digital certificate. Not only was the security level maintained, but once the digital certificates were issued, the provisioning of symmetric key-management services was completely automated thus providing the internet-level scalability required for enterprise operations.

6.1 Operation

The following diagram explains how the implementation works.



When a client – be it a laptop, a DB application or an e-commerce web-server - needs a symmetric key to encrypt some information, it makes a request for a new symmetric key to the linked in SKCL.

The SKCL checks its key-cache to determine if it has any cached symmetric keys that are valid for use. If so, it retrieves the key, decrypts it, verifies its integrity, checks its key-use-policy (every symmetric key object has an encryption policy embedded in it, previously defined by the site Security Officer) and then hands the requesting application the symmetric key for use.

If any of the local checks result in no valid symmetric key being available for use, the SKCL creates a new symmetric-key request, digitally signs it with its authentication credentials, and sends the request to one of its pre-configured SKS servers as an OASIS Web Services Security (WSS)-compliant SOAP request. (*Note: It is noteworthy to mention here, that since all requests and responses between the SKCL and the SKS servers were secured (digitally signed and encrypted) at the message-level, transport-level security (SSL/TLS or IPSec) was not required for the operations of the SKMS; plain old HTTP was sufficient. Administration console communications, however, did rely on mutually-authenticated SSL/TLS).*

The SKS server, upon receiving such a request, verifies the authenticity and integrity of the request, determines the authorization and the symmetric-key policy in force for the requester (or the default policy), generates a new symmetric key based on this policy, assigns it a **Global Key-ID (GKID)**, escrows the key (which includes encrypting it with multiple RSA keys), encrypts the key with the requester's transport digital certificate, logs the transaction details (which includes digitally signing the transaction) and responds to the client with a WSS-compliant SOAP response.

The SKCL client, upon receiving the response, verifies the authenticity and integrity of the request, caches the secured object if so configured, decrypts the symmetric key and the embedded key-use-policy and returns it to the calling application. The calling application at this time may choose to have the SKCL perform the actual encryption or perform it, itself.

A similar process is repeated when a client application needs to decrypt a previously-encrypted object such as a file, directory of files, database record, etc. The application determines the GKID of the symmetric key it needs (which was previously stored with the encrypted ciphertext in the XML Encryption format for files, and in a corresponding column for an RDBMS) and makes a request for this key to the SKCL. The SKCL checks to see if the requested key is in the key-cache. If it is, it goes through the standard security-checks and returns the symmetric key to the application; if not, it makes a request to the SKS server for this symmetric key. Upon receiving the request and after the standard security-checks, the SKS server responds with the symmetric key to the client. If the key does not exist for any reason, or the client is not authorized to receive the key, or for other error conditions, the SKS server returns a SOAP Fault to the requesting client.

It is noteworthy to mention, that given this operational infrastructure, it was feasible to use a unique symmetric key to encrypt every record in a database. With such an encryption policy, the breach of any key reduces the exposure of the database down to just a single record. This is in stark contrast to existing designs, where a single key typically encrypts an entire database or dataset, thus magnifying the loss associated with the loss of that single key.

7. BUILDING AN SKMS

The construction of an SKMS began with the creation of a PKI – or procurement of PKI services - to manage the issuance of digital certificates to every client. *The architecture deliberately eschewed the use of User-ID/Password for authentication because of their inability to prevent attacks against single-factor credentials.* The clients and servers in an SKMS use digital certificates for authentication, and secure storage & transport of symmetric keys within the infrastructure. (Notwithstanding the use of digi-

tal certificates, the administration console allows an Operations/Security officer to “deactivate” any client or server on the network without revoking the digital certificate of the affected entity).

Simultaneously, the application that will use the SKCL was created/modified to integrate the SKCL's API, accommodate the encrypted data (ciphertext) and the GKID in its database.
Multiple SKS servers were deployed and encryption policies configured on the servers while digital certificates were issued to clients that communicate with the servers. The applications were now ready to start requesting key-management services from the SKS servers. The SKMS transitioned to Production status at this point, and traditional operational activities took over (backup, configuration management, DR, etc.).

8. CONCLUSION

While symmetric encryption has been in use for decades within general computing, we have reached a confluence of inflection points in technology, the Internet and in regulatory affairs, that require IT organizations to implement Symmetric Key Management Systems (SKMS) as independent infrastructures. Using the soon-to-come Symmetric Key Services Markup Language (SKSML) standard from OASIS and the architecture defined in this paper, , IT organizations have another – and perhaps, one of the most effective – defense weapon in their arsenal against an increasingly hostile Internet.

REFERENCES

[1] History of cryptography - http://en.wikipedia.org/wiki/History_of_cryptography

[2] A chronology of data breaches - http://www.privacyrights.org/ar/ChronDataBreaches.htm

[3] California's Senate Bill 1386 - http://www.strongauth.com/regulations/sb1386/sb1386Index.html

[4] Breach at UCLA exposes data on 800,000 - http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9005925

[5] Retailer TJX reports massive data breach - http://www.infoworld.com/article/07/01/17/HNtjxbreach_1.html

[6] TJX Breach was twice as big as admitted, banks say - http://www.theregister.co.uk/2007/10/24/tjx_breach_estimate_grows

[7] TJX Form 10Q - http://www.theregister.co.uk/2007/10/24/tjx_breach_estimate_grows

[8] PCI Security Standards Council - https://www.pcisecuritystandards.org/index.htm

[9] Health Insurance Portability and Accountability Act of 1996 (HIPAA) - http://aspe.hhs.gov/admnsimp/pl104191.htm

[10] The Financial Modernization Act of 1999, aka “Gramm-Leach-Bliley Act (GLBA) - http://www.ftc.gov/privacy/privacyinitiatives/glbact.html

[11] Personal Information Protection and Electronic Documents Act (PIPEDA) - http://www.privcom.gc.ca/legislation/02_06_01_01_e.asp

[12] Directive 95/46/EC of the European Parliament aka EU Directive - http://www.cdt.org/privacy/eudirective/EU_Directive_.html

[13] IANA Private Enterprise Numbers (PEN) as used by RFC2578 - http://www.iana.org/assignments/enterprise-numbers

[14] StrongKey - http://www.strongkey.org

[15] OASIS Enterprise Key Management Infrastructure Technical Committee - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ekmi

Securing the Core with an Enterprise Key Management Infrastructure (EKMI)

Arshad Noor
StrongAuth, Inc.
arshad.noor@strongauth.com

Today's Goals

- What is an EKMI?
- What are its components?
- How do you build one?
- How do you secure one?
- What is the SKSMS Protocol?

What is an EKMI?

An **Enterprise Key Management Infrastructure** is:

"A collection of technology, policies and procedures for managing the life-cycle of ***all*** cryptographic keys in the enterprise."

What is an EKMI?

- PKI

  "A collection of technology, policies and procedures for managing the life-cycle of ***asymmetric*** cryptographic keys in the enterprise."

- SKMS

  "A collection of technology, policies and procedures for managing the life-cycle of ***symmetric*** cryptographic keys in the enterprise."

EKMI Components

- Public Key Infrastructure (PKI)
- Symmetric Key Management System (SKMS)

The problem

- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

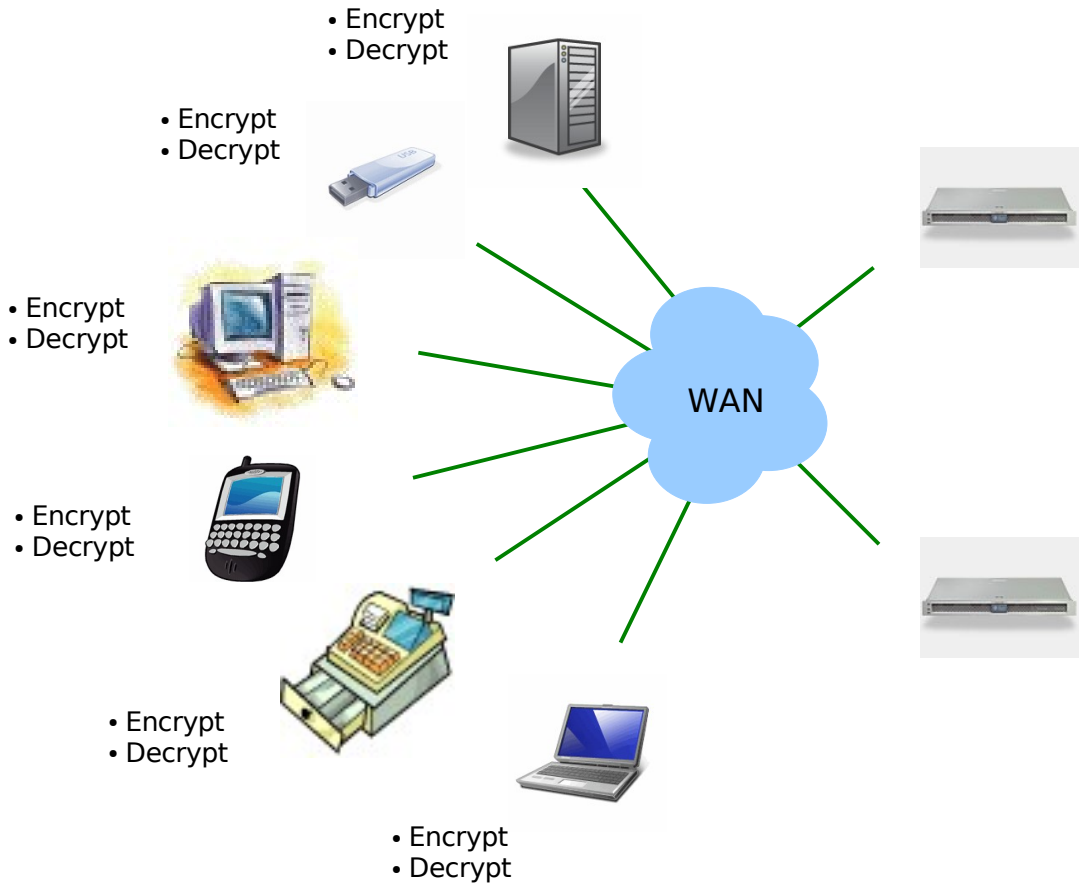
- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

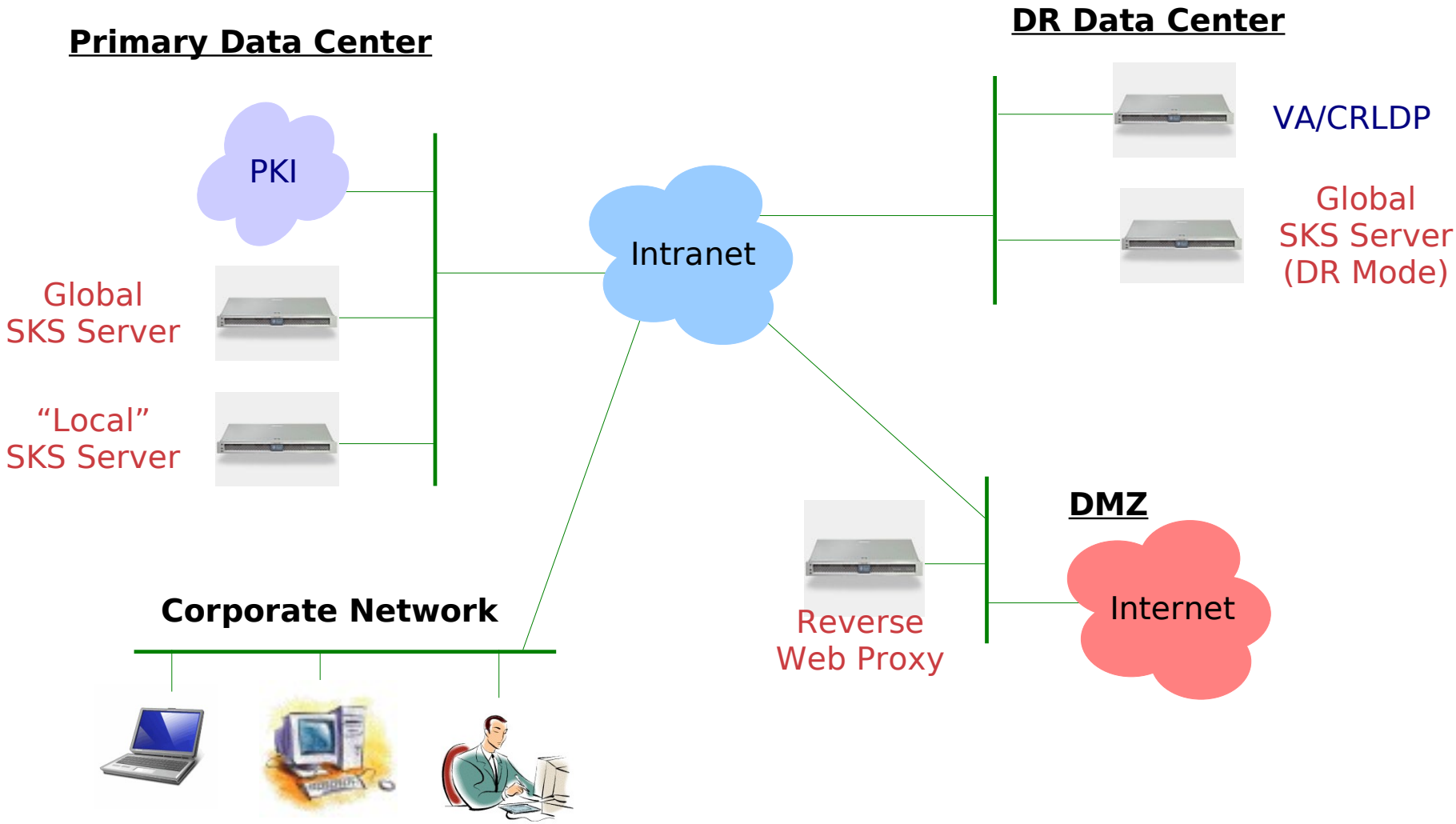
- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

………and on and on

Ideally...


• Encrypt
• Decrypt
• Encrypt
• Decrypt
• Encrypt
• Decrypt
• Encrypt
• Decrypt
• Encrypt
• Decrypt
• Encrypt
• Decrypt
WAN
• Define Policy
• Generate
• Protect
• Escrow
• Authorize
• Recover
• Destroy
• Audit

Typical SKMS



Global SKS Server

- One per enterprise
- Define all SKMS objects here:
  - Clients, Servers, Client Groups, Key Groups, Key Use Policies, Key Cache Policies, Grants
- DR Mode GSKS server is identical but Read-Only*
- HSM critical to security of server

SKS Server

- Any number per enterprise, as needed
  - One per continent recommended for global enterprises
- Configured to replicate to GSKS*
- HSM critical to security of server

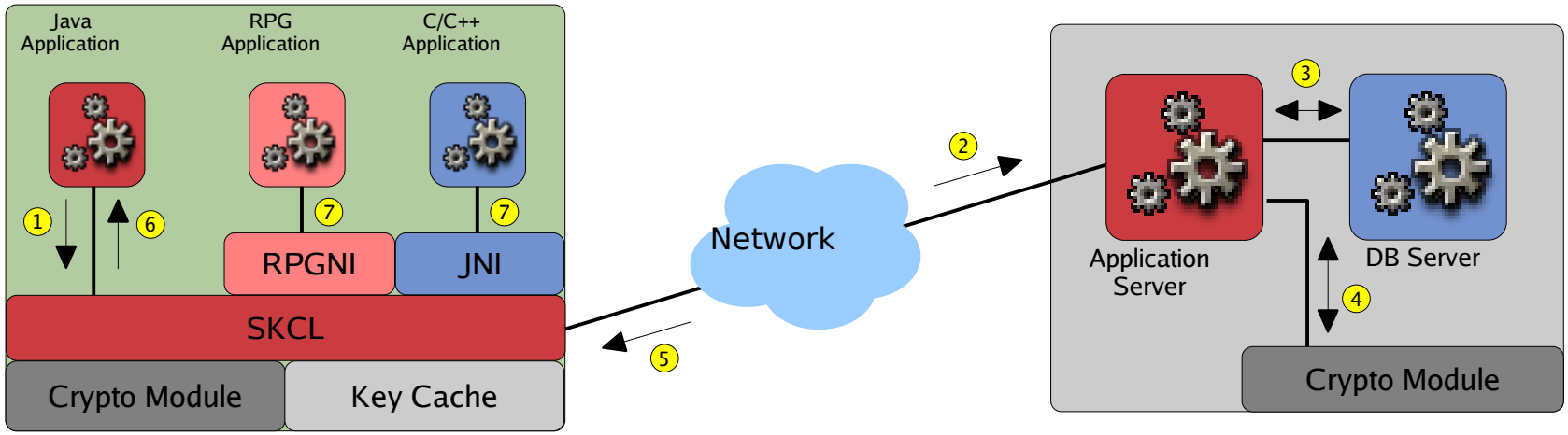
SKMS Client

- Any number per enterprise
- Maintains a list of SKS servers to get KM services from:
  1) Nearest SKS server on network
  2) GSKS Server
  3) GSKS DR-Mode Server
- HSM, smartcard token or TPM chip **strongly recommended** for security

Client Examples

- Database servers
- Web Application servers
- Network File servers
- Desktops/Laptops
- Automated Teller Machines (ATM)
- Point-of-Sale (POS) Registers
- Personal Digital Assistant (PDA)
- Smart mobile devices: Banking, Healthcare

The Big Picture



1. Client Application makes a request for a symmetric key
2. SKCL makes a digitally signed request to the SKS
3. SKS verifies SKCL request, generates, encrypts, digitally signs & escrows key in DB
4. Crypto HSM provides security for RSA Signing & Encryption keys of SKS
5. SKS responds to SKCL with signed and encrypted symmetric key
6. SKCL verifies response, decrypts key and hands it to the Client Application
7. Native (non-Java) applications make requests through Java Native Interface

SKMS Security

- Every request/response is digitally signed
- Every response is encrypted
- Every object in database is digitally signed
- All symmetric keys in cache are digitally signed and encrypted
- All crypto code is abstracted
  - FIPS 140-2 devices are easily integrated
- Administration console does not use UserID and Passwords; only SSL Client Auth.

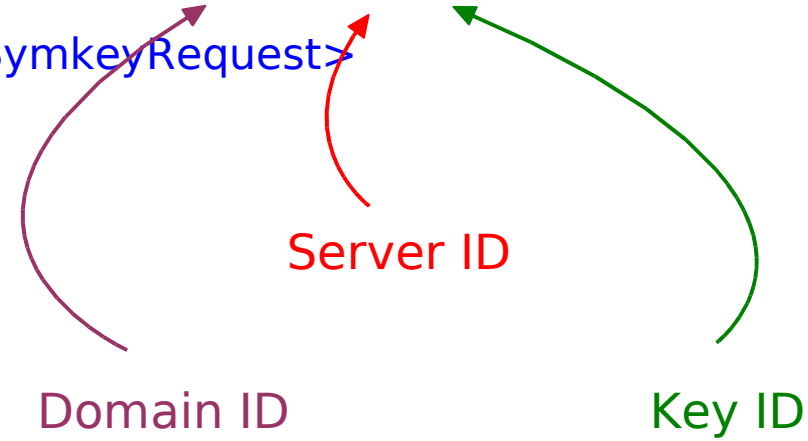
SKSML Protocol

- Symmetric Key Services Markup Language
- Donated to OASIS on royalty-free basis by StrongAuth, Inc.
- Currently a TC DRAFT; anticipated standard in Summer 2008
- Two (2) Request types: Key and CachePolicy
- Three (3) Response types: Key, CachePolicy and Fault

SKSML Protocol

Request for a new Symmetric Key

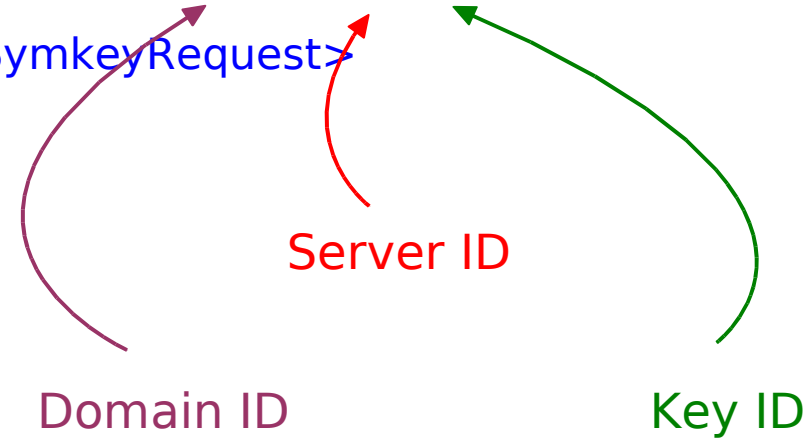
```
ekmi:SymkeyRequest xmlns:ekmi=”http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:GKID>10514-0-0</ekmi:GKID>
</ekmi:SymkeyRequest>
```



SKSML Protocol

Request for an existing Symmetric Key

```
ekmi:SymkeyRequest xmlns:ekmi=”http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:GKID>10514-4-312</ekmi:GKID>
</ekmi:SymkeyRequest>
```



SKSML Protocol

Request for a new Symmetric Key of particular KeyClass

```
ekmi:SymkeyRequest xmlns:ekmi=”http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:GKID>10514-0-0</ekmi:GKID>
    <ekmi:KeyClasses>
        <ekmi:KeyClass>HR-Class</ekmi:KeyClass>
    </ekmi:KeyClasses>
</ekmi:SymkeyRequest>
```

SKSML Protocol

Request for many new Symmetric Keys of specific KeyClasses

```
ekmi:SymkeyRequest xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:GKID>10514-0-0</ekmi:GKID>
    <ekmi:KeyClasses>
        <ekmi:KeyClass>EHR-CDC</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-CRO</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-DEF</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-EMT</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-HOS</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-INS</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-NUR</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-PAT</ekmi:KeyClass>
        <ekmi:KeyClass>EHR-PHY</ekmi:KeyClass>
    </ekmi:KeyClasses>
</ekmi:SymkeyRequest>
```

SKSML Protocol

Request for many existing Symmetric Keys

```
ekmi:SymkeyRequest xmlns:ekmi=”http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:GKID>10514-4-312</ekmi:GKID>
    <ekmi:GKID>10514-4-313</ekmi:GKID>
    <ekmi:GKID>10514-4-314</ekmi:GKID>
    <ekmi:GKID>10514-4-315</ekmi:GKID>
    <ekmi:GKID>10514-4-316</ekmi:GKID>
</ekmi:SymkeyRequest>
```

SKSML Protocol

Successful response with one key

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:Symkey>      .....      </ekmi:Symkey>
</ekmi:SymkeyResponse>
```

SKSML Protocol

Successful response with multiple keys

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:Symkey>      .....      </ekmi:Symkey>
    <ekmi:Symkey>      .....      </ekmi:Symkey>
    <ekmi:Symkey>      .....      </ekmi:Symkey>
</ekmi:SymkeyResponse>
```

SKSML Protocol

Failed response for one key

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:SymkeyError>      .....      </ekmi:SymkeyError>
</ekmi:SymkeyResponse>
```

SKSML Protocol

Failed response for multiple keys

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:SymkeyError>      .....      </ekmi:SymkeyError>
    <ekmi:SymkeyError>      .....      </ekmi:SymkeyError>
</ekmi:SymkeyResponse>
```

SKSML Protocol

Mixed response for multiple keys

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:Symkey>        .....        </ekmi:Symkey>
    <ekmi:Symkey>        .....        </ekmi:Symkey>
    <ekmi:SymkeyError>        .....        </ekmi:SymkeyError>
    <ekmi:SymkeyError>        .....        </ekmi:SymkeyError>
</ekmi:SymkeyResponse>
```

SKSML Protocol

Symkey element

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">

    <ekmi:Symkey>
        <ekmi:GKID>10514-1-287</ekmi:GKID>
        <ekmi:KeyUsePolicy>      .....      </ekmi:KeyUsePolicy>
        <ekmi:EncryptionMethod Algorithm=
            "http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
        <xenc:CipherData>
            <xenc:CipherValue>
                huUYJMtaGHtXuLlWtx27STRcRpIsY=
            </xenc:CipherValue>
        </xenc:CipherData>
    </ekmi:Symkey>

</ekmi:SymkeyResponse>
```

SKSML Protocol

KeyUsePolicy element

```
<ekmi:KeyUsePolicy>
    <ekmi:KUPID>10514-4</ekmi:KUPID>
    <ekmi:PolicyName>DES-EDE KeyUsePolicy</ekmi:PolicyName>
    <ekmi:KeyClass>HR-Class</ekmi:KeyClass>
    <ekmi:KeyAlgorithm>http://www.w3.org/2001/04/xmlenc#tripledes-cbc</ekmi:KeyAlgorithm>
    <ekmi:KeySize>192</ekmi:KeySize>
    <ekmi:Status>Active</ekmi:Status>
    <ekmi:Permissions>      .....      </ekmi:Permissions>
</ekmi:KeyUsePolicy>
```

SKSML Protocol

Permissions element

```
<ekmi:Permissions>
    <ekmi:PermittedApplications> .....  </ekmi:PermittedApplications>
    <ekmi:PermittedDates>    .....      </ekmi:PermittedDates>
    <ekmi:PermittedDuration>     .....  </ekmi:PermittedDuration>
    <ekmi:PermittedLevels>   .....      </ekmi:PermittedLevels>
    <ekmi:PermittedLocations>    .....  </ekmi:PermittedLocations>
    <ekmi:PermittedTimes>    .....      </ekmi:PermittedTimes>
    <ekmi:PermittedTransactions> ..... </ekmi:PermittedTransactions>
    <ekmi:PermittedUses>     .....      </ekmi:PermittedUses>
    <ekmi:Other>     .....              </ekmi:Other>
</ekmi:Permissions>
```

KeyUsePolicy element

```
<ekmi:KeyUsePolicy>
        <ekmi:KUPID>10514-4</ekmi:KUPID>
        <ekmi:PolicyName>DES-EDE KeyUsePolicy</ekmi:PolicyName>
        <ekmi:KeyClass>HR-Class</ekmi:KeyClass>
        <ekmi:KeyAlgorithm>http://www.w3.org/2001/04/xmlenc#tripledes-cbc</ekmi:KeyAlgorithm>
        <ekmi:KeySize>192</ekmi:KeySize>
        <ekmi:Status>Active</ekmi:Status>
        <ekmi:Permissions>
                <ekmi:PermittedApplications>
                        <ekmi:PermittedApplication>
                                <ekmi:ID>10514-23</ekmi:ID>
                                <ekmi:ApplicationName>Payroll Application</ekmi:ApplicationName>
                                <ekmi:Version>1.0</ekmi:Version>
                                <ekmi:DigestAlgorithm>http://www.w3.org/2000/09/xmldsig#sha1</ekmi:DigestAlgorithm>
                                <ekmi:DigestValue>NIG4bKkt4cziEqFFuOoBTM81efU=</ekmi:DigestValue>
                        </ekmi:PermittedApplication>
                </ekmi:PermittedApplications>
                <ekmi:PermittedDates>
                        <ekmi:PermittedDate>
                                <ekmi:StartDate>2008-01-01</ekmi:StartDate>
                                <ekmi:EndDate>2008-12-31</ekmi:EndDate>
                        </ekmi:PermittedDate>
                </ekmi:PermittedDates>
                <ekmi:PermittedTimes>
                        <ekmi:PermittedTime>
                                <ekmi:StartTime>07:00:00</ekmi:StartTime>
                                <ekmi:EndTime>19:00:00</ekmi:EndTime>
                        </ekmi:PermittedTime>
                </ekmi:PermittedTimes>
        </ekmi:Permissions>
</ekmi:KeyUsePolicy>
```

Symmetric Key Response

```
<ekmi:SymkeyResponse xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
    <ekmi:Symkey>
        <ekmi:GKID>10514-1-235</ekmi:GKID>
        <ekmi:KeyUsePolicy>
            <ekmi:KUPID>10514-4</ekmi:KUPID>
            <ekmi:PolicyName>DES-EDE KeyUsePolicy</ekmi:PolicyName>
            <ekmi:KeyClass>HR-Class</ekmi:KeyClass>
            <ekmi:KeyAlgorithm>http://www.w3.org/2001/04/xmlenc#tripledes-cbc</ekmi:KeyAlgorithm>
            <ekmi:KeySize>192</ekmi:KeySize>
            <ekmi:Status>Active</ekmi:Status>
            <ekmi:Permissions>
                <ekmi:PermittedApplications>
                    <ekmi:PermittedApplication>
                        <ekmi:ID>10514-23</ekmi:ID>
                        <ekmi:ApplicationName>Payroll Application</ekmi:ApplicationName>
                        <ekmi:Version>1.0</ekmi:Version>
                        <ekmi:DigestAlgorithm>http://www.w3.org/2000/09/xmldsig#sha1</ekmi:DigestAlgorithm>
                        <ekmi:DigestValue>NIG4bKkt4cziEqFFuOoBTM81efU=</ekmi:DigestValue>
                    </ekmi:PermittedApplication>
                </ekmi:PermittedApplications>
                <ekmi:PermittedDates>
                    <ekmi:PermittedDate>
                        <ekmi:StartDate>2008-01-01</ekmi:StartDate>
                        <ekmi:EndDate>2008-12-31</ekmi:EndDate>
                    </ekmi:PermittedDate>
                </ekmi:PermittedDates>
                <ekmi:PermittedTimes>
                    <ekmi:PermittedTime>
                        <ekmi:StartTime>07:00:00</ekmi:StartTime>
                        <ekmi:EndTime>19:00:00</ekmi:EndTime>
                    </ekmi:PermittedTime>
                </ekmi:PermittedTimes>
            </ekmi:Permissions>
        </ekmi:KeyUsePolicy>
        <ekmi:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
        <xenc:CipherData>
            <xenc:CipherValue>Yjv9h5FDqUiQXG0ca8EU871zBoXBjDXmlNxTux+mt1tXuLlWtx27STRcRpIsY=</xenc:CipherValue>
        </xenc:CipherData>
    </ekmi:Symkey>
</ekmi:SymkeyResponse>
```

SKSML Protocol

SymkeyError element

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">

    <ekmi:SymkeyError>
        <ekmi:RequestedGKID>10514-0-0</ekmi:RequestedGKID>
        <ekmi:RequestedKeyClass>Payroll</ekmi:RequestedKeyClass>
        <ekmi:ErrorCode>SKS-100010</ekmi:ErrorCode>
        <ekmi:ErrorMessage>
            Unauthorized to request this key-class
        </ekmi:ErrorMessage>
    </ekmi:SymkeyError>

</ekmi:SymkeyResponse>
```

SKSML Protocol

SymkeyError element

```
<ekmi:SymkeyResponse
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">

    <ekmi:SymkeyError>
        <ekmi:RequestedGKID>10514-2-2254</ekmi:RequestedGKID>
        <ekmi:ErrorCode>SKS-100004</ekmi:ErrorCode>
        <ekmi:ErrorMessage>Unauthorized request</ekmi:ErrorMessage>
    </ekmi:SymkeyError>
    <ekmi:SymkeyError>
        <ekmi:RequestedGKID>10514-0-2254</ekmi:RequestedGKID>
        <ekmi:ErrorCode>SKS-100001</ekmi:ErrorCode>
        <ekmi:ErrorMessage>Invalid GKID</ekmi:ErrorMessage>
    </ekmi:SymkeyError>

</ekmi:SymkeyResponse>
```

SKSML Protocol

Request for a Key Caching Policy

```
<ekmi:KCPRequest   xmlns:ekmi="http://doc.oasis-open.org/ekmi/2008/01"/>
```

Full SOAP request for a Key Caching Policy

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
      <wsse:BinarySecurityToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="XWSSGID-1172790286164-631000224">MIIDfDCCAmSgAwIBAgIIAe/AvliGc3AwDQYJKoZIhvcNAQELBQAwZzEmMCQGA1UEAxMdU3Ryb25n
        S2V5IERFTU8gU3Vib3JkaW5hdGUgQ0ExJDAiBgNVBAsTG0ZvciBTdHJvbmdLZXkgREVNTyBVc2Ug
        T25seTEXMBUGA1UEChMOU3Ryb25nQXV0aCBJbmMwHhcNMDYwNzI1MTcxMDMwWhcNMDcwNzI1MTcy
        MDMwWjBtMREwDwYKCZImiZPyLGQBARMBMjEZMBcGA1UEAxMQUE9TIFJlZ2lzdGVyIDIyMjEkMCIG
        A1UECxMbRm9yIFN0cm9uZ0tleSBERU1PIFVzZSBPbmx5MRcwFQYDVQQKEw5TdHJvbmdBdXRoIElua
        YzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAyAmxMZhYA8wHJ4UE4b61s51JVWe4Fygj4MCf
        U7LA3JhpUS4TlX0XFWqrcmltLOiVG7YBFarJFluBFJW2X6q8FuvUprv4V9nJrgiwAPtkiRyIx96n
        qKXIxkUlQ4idlEg1AZI9dEdf4Y5cqBBCygPYnBoTudgIM7R47AjR4nr4ks8CAwEAAaOBqTCBpjAO\
        MTAvoC2gK4YpaHR0cDovL2RlbW8uc3Ryb25na2V5Lm9yZy9kZW1vLXN1Yi1jYS5jcmwwDQYJKoZI
        hvcNAQELBQADggEBACK05PtvZD4WPglOe+EHUiApzFyCdRzf0pFZtxRwG9lR1PZUWUjmwTNfGFsL
        S6kyoHgUfVa5fpT1EU1mXUB/Lmo3hFGyprZjfmD7DwuBcYgmZHv7yHrmGOMIOXjFTACvHpM0vOce
        hVx2e4VE0yhBLu/ldH9awGGDp6Bk2XzxqQcs8y6ZzOXZAnPgKQZdjbFKERSsy/d1D8pk5baBk4bd
        Zh568OcaUrbm9ZReRVTVaY5qiQpkOU+tDrBSj/HIL6GAqegYllkz6KYCy6RVOy6iVVSjHocDqdJr
        EVOR+ds6xn8mmojdlERrILmuxiLpibPp609SfnDIxNlzLwe5g7ep3lc=</wsse:BinarySecurityToken>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <InclusiveNamespaces xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="wsse SOAP-ENV"/>
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <ds:Reference URI="#XWSSGID-1172790287316421676595">|
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>xgITEmXg4yMMg+RkKDTyVUw87VM=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#XWSSGID-11727902873271435349604">
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>tlWtno1OMA7Os8GCZrUhmHXmmhM=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>Uv1zAePPoQmFey6UyslAYc2IzArOPwfXB48eTehoYaESrgzYVXSJetis2+2Jd8AgAsh1LF6Ayh67LhIAdigZPHR/9SxUYh4=</ds:SignatureValue>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity- tility-1.0.xsd" wsu:Id="XWSSGID-1172790287191-1130524177">
            <wsse:Reference URI="#XWSSGID-1172790286164-631000224" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"wsu:Id="XWSSGID-11727902873271435349604">
        <wsu:Created>2007-03-01T23:04:47Z</wsu:Created>
        <wsu:Expires>2007-03-01T23:04:52Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1172790287316421676595">
    <ekmi:KCPRequest xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Key Cache Policy Response

```
<ekmi:KeyCachePolicy xmlns:ekmi='http://docs.oasis-open.org/ekmi/2008/01'>
    <ekmi:KCPID>10514-17</ekmi:KCPID>
    <ekmi:PolicyName>Corporate Laptop Symmetric Key Caching Policy</ekmi:PolicyName>
    <ekmi:Description>
        This policy defines how company-issued laptops will manage symmetric keys
        used for file/disk encryption in their local cache.
    </ekmi:Description>
    <ekmi:StartDate>2008-01-01T00:00:01.0</ekmi:StartDate>
    <ekmi:EndDate>2008-12-31T24:00:00.0</ekmi:EndDate>
    <ekmi:PolicyCheckInterval>86400</ekmi:PolicyCheckInterval>
    <ekmi:Status>Active</ekmi:Status>
    <ekmi:NewKeysCacheDetail>
        <ekmi:MaximumKeys>3</ekmi:MaximumKeys>
        <ekmi:MaximumDuration>7776000</ekmi:MaximumDuration>
    </ekmi:NewKeysCacheDetail>
    <ekmi:UsedKeysCacheDetail>
        <ekmi:MaximumKeys>3</ekmi:MaximumKeys>
        <ekmi:MaximumDuration>7776000</ekmi:MaximumDuration>
    </ekmi:UsedKeysCacheDetail>
</ekmi:KeyCachePolicy>
```

SKSML Protocol

SOAP Fault

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      ERROR: Other error reported; please review logs for details.  Server error message is: No authorization
      to request this key:10514-2-2; if you believe this response is an error, please contact your Security Officer
   </SOAP-ENV:Header>
   <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
         wsu:Id="XWSSGID-1154644495295194261602 4">
      <SOAP-ENV:Fault>
         <faultcode xmlns:skf="http://www.strongauth.com/2006/01/symkey#SymkeyFault">
            skf:SymkeyFault
         </faultcode>
         <faultstring>symkey.sks.msg.severe.0085</faultstring>
         <detail>
            <EndEntity>
               <EEID>10514-2</EEID>
               <DN>O=StrongAuth Inc,CN=POS Register 222,UID=2</DN>
               <Status>Active</Status>
            </EndEntity>
            <Request>
               <RID>10514-3</RID>
               <GKID>10514-2-2</GKID>
               <Timestamp>2006-08-03 15:34:55.0</Timestamp>
               <Disposition>Failed</Disposition>
            </Request>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

OASIS EKMI TC

- Technical Committee with 4 goals:
  1. Standardize Symmetric Key Services Markup Language (SKSML)
  2. Create Implementation & Operations Guidelines
  3. Create Audit Guidelines
  4. Create interoperability test-suite for SKSML

EKMI TC Members

- FundServ* (Canada)
- MISMO
- NuParadigm Government Systems
- PA Consulting (UK)
- PrimeKey (Sweden)
- Red Hat
- StrongAuth*
- US Dept. of Defense
- Visa*
- Wave Systems
- Wells Fargo
- OS Software company
- Database SW company
- Storage/Security SW company
- Storage/Security SW company
- Govt. Agency (New Zealand)
- Individuals representing Audit and Security backgrounds*

* Founder Members

Conclusion

- Questions?
- Contact Information
  - www.strongauth.com
  - www.strongkey.org
  - info@strongauth.com
  - (408) 331-2000

A Federation of Web Services for Danish Health Care

Esben Dalsgaard
Chair, SOSI steering committee
Digital Health Denmark (SDSD)
Rugaardsvej 15
DK-5000 Odense C, Denmark
ead@sdsd.dk

Kåre Kjelstrøm
Solution Architect
Silverbullet A/S
Skovsgaardsvaenget 21
DK-8362 Hoerning, Denmark
(+45) 2092 8244
kkj@silverbullet.dk

Jan Riis
Solution Architect / Project Manager
Lakeside A/S
Aabogade 15
DK-8200 Aarhus N, Denmark
(+45) 2160 7252
jri@lakeside.dk

ABSTRACT

Having relevant, up-to-date information about a patient's health care history is often crucial for providing the appropriate treatment. In Denmark, IT systems have been built to support different work flows in the health sector, but the systems are rarely connected and have become islands of data.

To remedy this situation, a service-oriented architecture based on web services for online exchange of health care data between the vast array of heterogeneous IT systems in the sector is being built.

The architecture forms a federation of web services and enables secure and reliable authentication of end-users and systems in the Danish health sector. The architecture is based on national and international standards and specifications. Yet it defines its own profile for secure interchange of data due to a lack of available international profiles that could handle the special needs of the health sector at the time of project inception.

The architecture has evolved through a pilot project from mid 2005 to the end of 2007, and is being tested in a small scale 1st quarter 2008. This paper aims to convey experiences from the project, so rich in benefits that the architecture has been accepted and standardized as the foundation for the future of system integration in the health sector in Denmark.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications

D.2.11 [**Software Architectures**]

D.2.12 [**Interoperability**]: Distributed Objects

D.2.13 [**Reusable Software**]: Reusable Libraries

General Terms

Performance, Design, Reliability, Experimentation, Security, Human Factors, Standardization, Legal Aspects, Verification.

Keywords

Federated Identity Management, Web Services, SOA, SAML, WS-Trust. Single sign on, X509 Certificates, Digital Signatures, SOAP, Security Token Service, Health Care, Electronic Patient Records.

IDTrust '08, March 4–6, 2008, Gaithersburg, MD.

1. INTRODUCTION

The IT system landscape in the Danish health care sector contains a plethora of different systems targeting various needs: patient administration, general practitioning, specialized care, electronic health recording, citizen access through web based health portals, etc.

The systems fall more or less uniformly into three classes:

1) Off-the-shelf systems typically obtained by privately held companies (e.g. health centers)

2) Tender based regional systems (e.g. for hospitals) and

3) National systems, typically tender based systems hosted by health care related departments.

Some of these systems are integrated today, but typically integration has been done locally, with the aim to reduce information redundancy. The real benefit in terms of quality of patient treatment and care, however, lies in a deeper integration of health care systems across organizational boundaries, such that *all relevant* information for treatment and care is made directly available in the systems that the health care professionals use in their daily work.

Founded in the strategic vision to strive for better quality in patient treatment, better systems for health care professionals, and the optimization of resources, the health care sector in Denmark has started the work on a national health care integration architecture that supports this vision.

The quest for universal availability of relevant and up-to-date information has been *the* most important force in shaping the architecture. There are, however, many other premises that have governed this work, for instance the fact that in this domain, the "business" is never closed even if some or all of its IT systems become unavailable: People will still need treatment and care.

In 2005, the Danish health care sector launched an initiative with the purpose of analyzing, profiling and testing a combination of national and international standards in order to standardize service based integration mechanisms including measures for strong authentication of principals based on PKI.

The initiative was coined the SOSI project for "Service Oriented System Integration". It was initiated by the Capital Region of Denmark, The Region of South Denmark, and the Danish Medicines Agency. Present in the steering committee was also the Danish Ministry of Science, Technology and Innovation. The project was funded by Danish Regions and is now governed by the Danish National eHealth initiative: Digital Health Denmark [4].

1.1 Life in the Health Sector

The Danish health sector is financed by the Danish government via taxes and treatment is otherwise free. As opposed to other countries such as e.g. the USA this means that there is only one major health care provider, and that many facilities including hospitals are owned by the public sector.

The health sector employs a wide array of different professionals, most notably hospital doctors and nurses, general practitioners (GP), and caregivers to the elderly and disabled.

In contrast to most of the other organizations in the health sector, GPs are private and often times small businesses employing only one or a few doctors as well as a secretary. The use of electronic health record systems (EHR) is widespread in this part of the sector: Today doctors receive patients in their consulting room from behind a computer screen, running an EHR system.

From time to time, the secretary will act on behalf of the GP in taking blood samples, screening patients over the phone while checking health records on his own computer, etc.

GPs will prescribe medications for the elderly or disabled, submit patients to hospitals, receive patients from hospitals for outpatient treatment, and more.

For hospital doctors and nurses, life is somewhat different from that of their GP colleagues.

Sometimes a doctor will work with patients in a ward all day together with a nurse in a situation akin to that of the GP. Wards are often equipped with a single computer that must be shared between the two professionals. At other times, a doctor must take ward rounds and share computers with local nurses and other doctors.

Doctors typically use IT systems in a read-only fashion while employing a dictation machine to take notes and make adjustments in patient treatment. Information is then entered into the system by a secretary who acts on behalf of the doctor.

Where GPs typically use a single system for all purposes, hospital doctors and nurses will use a set of systems during the day.

In another part of the sector, caregivers follow the directions of GPs in administering medicine for the elderly and disabled. Sometimes those receiving care will live in a nursing home, sometimes in private homes.

Because of the geographical distribution of clients, caregivers have a need for mobile access to medical information. In Denmark, this is usually realized through portable computers in the shape of PDAs. In nursing homes, a number of computers are shared by different caregivers in a situation akin to the one in hospitals.

Caregivers cannot prescribe and have only limited access to health care information, but can report the administration of medicines.

Danish law protects the rights of patients through "The Danish Act on Processing of Personal Data" [20] and "The Danish Health Law" [22]. These laws govern who can access what kinds of information and why, and points out that citizens have a right to know to whom information has been disclosed.

In 2004, a web based national health portal, Sundhed.dk, [23] was launched with the purpose of providing a single point of access to health information for citizens. One vision of the portal is to make it possible for a citizen to learn what health care information is registered about her, and who has viewed it in compliance with the laws.

The health portal hence has a need to pull information from GPs, hospitals, laboratories and more, and to act as a medium between GPs and citizens e.g. for electronic consultation.

In summary, health care professionals across the sector need to exchange information about patients that pass from one practitioner to the next in order to provide the best possible care. At the same time, citizens have the right to gain insight into their own health care records.

These needs call for an IT-infrastructure that provides up to date health care information about patients across the sector and across the country in a timely fashion.

1.2 Privacy vs. Safety

Health care records often contain sensitive data, which could potentially harm a person's reputation or private life, should it be exposed to unauthorized people. More seriously, though, these records are the basis on which a patient receives care, and errors caused by negligence, malicious intent, or the like can potentially cause physical harm.

For these reasons, health care records are surrounded by security measures.

Ensuring the confidentiality of information while in transit from one practitioner to the next, and while being stored, is imperative to avoid eavesdropping by unauthorized individuals.

Organizations that handle sensitive data are bound by Danish law to ensure that only authorized staff gains access. In order to comply with the privacy acts then, a practitioner should only be able to access information that she is authorized to, and which is of relevance with respect to her current treatment of a patient. Identifying health care personnel with a high degree of certainty, and performing authorization checks are hence prerequisites for exposing information.

Yet even with all the locks and latches of the world, information will eventually be spilled to unauthorized people, typically through authorized personnel. When such a breach is detected, it is imperative to be able to trace the identity of the malefactor for forensic purposes.

The need for privacy is complicated by the fact that access to information is sometimes a matter of life and death. While citizens have the right to privacy, safety has a higher priority in Danish health care. In other words, the life of a patient takes precedence over the unconfirmed exposure of sensitive information to authorized health care staff.

In the event of unconfirmed exposure, tracking the identity and following up on such an event is a necessity in ensuring privacy.

Hospital doctors and nurses typically use not just one, but several IT systems during the day. In the worst case scenario, a user has different identities and uses different credentials with each system.

Logging in and out of systems can therefore be a time consuming task if care is not taken.

The first step towards providing a faster and simpler authentication mechanism is hence to create a single identity with single credentials for access to all systems. The second step is to provide a single step of authentication to all systems, so called single sign on (SSO).

1.3 Availability

The existence of health care IT systems is justified only by promises of improvements in the overall quality and efficiency of patient treatment.

When work routines based on IT systems replace manual, paper based ones, health care professionals will begin to plan their daily schedule accordingly. This means that once a certain quality of service (QOS) has been established for day-to-day operations, this QOS has to be maintained.

In a complex of systems that exchange clinical information, any participant *must* hence minimize the impact caused when other systems fail for instance by switching to emergency states where only locally cached information is available.

The longer such external information systems are unavailable the higher the risk of inefficiency or errors in patient treatment, and hence the higher the risk of physical harm, adverse effects or permanent maladies. As a safeguard, applications, power sources, communication lines, etc. must therefore be highly redundant, and built with robustness in mind to ensure continued operations even when external systems are down.

2. THE SOSI SOLUTION

The Danish National board of Health is responsible for an overall IT-strategy with an ambitious goal: to provide a connected health care sector in which health professionals have access to all relevant EHR data regardless of where citizens seek treatment and no matter where or when this information was registered [5]. It is nonetheless important to stress that the motivation for the SOSI solution is primarily rooted in user requirements.

Most health care professionals wish to provide the best possible quality possible in their work, and hence base their decisions on all accessible and relevant information. There are therefore examples of health professionals starting up at least five applications every morning: some local, some regional, and some national. Single sign on as well as infrequent sign on are hence examples of real world requirements.

The SOSI project aimed at evaluating standards and technologies that could provide value to users by standardizing authentication mechanisms, standardizing the way service providers should expose services and by providing tools that could lower the threshold for both service providers and service consumers to be part of this new game.

Given the environment of disparate IT systems scattered across the country with a need for common information, it was decided to realize this goal through a national Service-Oriented Architecture (SOA) via SOAP based web services over HTTP.

The architecture had to address the availability and security issues identified earlier and build on existing infrastructure to reduce costs, while adhering to national and international standards in order to ensure maintainability.

There exists within the context of SOAP based web services a profusion of specifications aimed at solving various well-known issues from the world of computing: security, reliability, messaging, addressing, transactions, etc.

Each specification adds levels of complexity and typically provides not just one, but multiple ways of achieving the same overall goal. Add to this the fact that often times, specifications from different bodies compete to become the de-facto standard, each attacking the problem at hand in slightly different ways: There's the recipe for non-interoperability.

The solution to this problem comes in the shape of profiles that cut through the stack of specifications, paving a narrow path of design choices for specific usage scenarios.

In the world of federated single sign on over the Internet, a number of profiles and specifications exist. OASIS defines the SAML specification [14], which is implemented by the Internet2 initiative Shibboleth [6]. A large group of non-Microsoft companies drive The Liberty Alliance Project [21], whose specifications extend SAML. IBM and Microsoft push the WS-Federation [7] specification and implement support in a range of products.

The Ministry of Science, Technology and Innovation (MVTU) drives much of the standardization effort in the Danish public sector. It does so in part by evaluating international profiles and specifications and classifying them in an interoperability framework [25]. For federated identity management, SAML 2.0 is classified as the preferred framework of choice. Any Danish SSO architecture should hence build on SAML, which was therefore chosen for SOSI as well.

2.1 Security Architecture

An IT system that participates in a service-oriented architecture must weigh the risks associated with revealing information to unauthorized people against available security measures.

In the health sector, risks generally include unauthorized disclosure of sensitive information and in some cases physical harm. While some types of information e.g. classifications of diseases may be harmless if disclosed, others such as patient records are often not.

Because in Denmark it is legally the responsibility of the data owning organization to prevent unauthorized disclosure, every web service provider must perform a risk analysis and potentially strengthen security measures before exposing information via the federation.

That said, it still makes sense to define a set of basic security properties that are always in place, and to lay out a simple set of security choices that can be implemented depending on identified risks.

A cornerstone in the security architecture of SOSI, the CIA triad addresses the aspects of Confidentiality, Integrity and Availability [19]:

When in transit, data will pass over a number of networks and confidentiality is ensured through the use of encryption techniques to avoid eavesdropping, no matter the content.

Communicating parties also need a guarantee that data has not been altered during transit. The integrity property is also guaranteed via cryptographic techniques.

Finally, services must be available to be of any value to users, guaranteed by redundancy of critical components, communication lines, and enforced through policies and agreements.

Single sign on is achieved through the use of a trusted third party, who will verify credentials on behalf of all parties in the federation. This is a delegation of trust model, which reduces the burden of federation participants: instead of knowing all possible users, it is enough to be able to verify claims from the trusted third party.

SAML assertions are useful for propagating claims about digital identities that can be used in e.g. authorization checks. To be trusted by a third party, though, credentials must additionally be supplied for external verification.

Credentials come in many shapes and sizes with different security properties. While passwords may be simpler to manage, they are susceptible to eavesdropping attacks and may be easy to crack if not chosen carefully and changed often. X509 certificates offer stronger properties, non-repudiation and confidentiality, but require equally careful handling of private keys to be trusted.

The SOSI federation employs strong credentials based on X509 certificates, which provide a high level of certainty in the identification of the sender.

2.2 Architectural Building Blocks

When designing how exactly confidentiality and integrity should be realized through cryptography, it came down to reusing an existing national VPN based health care network or to employ "end-to-end" message encryption/signing. Although end-to-end encryption/signing may seem captivating because messages can then pass freely over potentially any network, the benefits were deemed smaller than the burden of encrypting and signing streams of very large messages.

A large part of the health sector organizations in Denmark are already connected to the above mentioned VPN network known as "SDN" [8]. The network was originally planned for teleconferencing, exchanging large amounts of data e.g. x-ray images, and accessing web based applications in a secure manner.

Any organization that wants access to services on SDN is evaluated for relevancy and must sign a mutual agreement per system-to-system connection. Although cumbersome, this procedure provides a certain degree of certainty that the network is primarily made up of organizations with legal business in the health sector.

By supplying an integrity and confidentiality protected transport mechanism, which is immune to known security attacks, and which has many of the relevant organizations connected already, SDN is useful for web services as well.

Part of the Danish it-strategy is the mandated use of digital signatures for secure identification of health care personnel. An important precondition in the design of a solution for SOSI would therefore be to leverage the Danish national certificate initiative, OCES.

OCES provides all the features of a nationally implemented X509 based PKI. What makes OCES even more interesting is the Open Source components and commercial products that surround the initiative which can also be used in WS integration. Specifically a Signature Server [1] that enables secure centralization of private keys in the client environment and the OpenOCES [18] components that enable Java access to the Windows crypto API have been used and evaluated in the SOSI project.

Last but not least the current OCES operator has several web services that support the OCES initiative, e.g. services for converting certificate subject serial number to the national Danish person identification number.

In mid 2005, when the SOSI project was initiated, none of the existing Single sign on projects gave good solutions to the particular needs of the project. Although there was a SOAP binding for SAML, no profile existed that laid out a complete protocol stack for exchanging SOAP messages with SAML assertions, while achieving Single sign on to web services.

There was and still is a heavy bias towards providing SSO for browser-based clients, with specifications relying on facilities such as HTTP redirect and cookies.

However, most health care applications in Denmark are non-browser based. In most cases users need specialized and highly supportive systems, something which until very recently was not feasible to build with web browser technology.

Many of the existing SSO projects include services or components that increase system dependencies instead of reducing them, thereby introducing potential single points of failure. One of the best examples is that most SSO profiles mandate service provider initiated user authentication, typically done by communicating with an authentication service.

In browser-based applications, where connectivity is a precondition, service provider initiated authentication is natural and probably the only viable solution, in part because client systems are very thin and mostly session based applications. If the central authentication service is unavailable, service providers cannot be called either. .

However, in a pure web service integration architecture, where clients more often than not are servers themselves, it is possible to build more fault tolerant architectures.

Client-initiated user authentication can for instance be mandated, and issued security tokens be cached in client system for later use. In a model where security tokens are additionally off-line verifiable by service providers, only users that are not already authenticated will be affected if the authentication service is unavailable.

So, although many of the existing profiles had elements that could be reused, the use-cases and interaction schemes were not. A basic SAML and WS-Trust based profile [9] was therefore created based on the following principles:

1. A user should be able to authenticate with the federation once and then be able to use any service for which she has authorization for as long as she can present a valid federated security token. The design should, in other words, help reduce the number of sign-ons to the federation.
2. Basic information security should be provided by the existing security infrastructure.
3. Using a client initiated authentication scheme, a WS client (WSC) system should be responsible for logging the user into the federation before starting to interact with any WS provider (WSP).

4. Inspired by current work on short-lived PKI certificates [16][17] the security token must have a limited lifetime and hence eliminate the need for revocation checks by WSPs.
5. Security tokens must be off-line verifiable by WSCs and WSPs, i.e. without having to communicate with any third party.
6. Security tokens should be able to carry basic end-user and client-system attributes that most WSPs use for authorization and/or logging. The design should support trust re-use, such that when the credentials within the security token have been verified, the embedded attributes can also be trusted. In effect this reduces the effort that WSPs must put into implementing web services. It also stabilizes the entire architecture by reducing system dependencies to a minimum.
7. Security tokens must not be subject to theft, i.e. measures must be put in place to hamper hostile token takeover.

The proposed technical solution consists of:

- A trusted federation Security Token Service (STS) with a maximum validity of 24 hours.
- Security tokens as digitally signed SAML Assertions
- Client initiated authentication that results in STS signed SAML assertions
- Core attributes embedded in the SAML security token
- Message integrity through digital signing of SAML assertions combined with web service body data.
- Confidentiality of transport, but not of message data in a trusted circle of participants.

Figure 1 shows a simple interaction between a WSC, an STS, and a number of WSPs. Please note the WSC initiated authentication scheme and that the WSPs are not depending on access to the STS to verify security tokens and basic attributes:

Step 1. The user authenticates with the federation either just-in-time before calling a service or as part of the local log-on to the WSC system. The WSC builds a SAML assertion with core attributes and user credentials, in this case a digital signature.

Step 2. The STS checks that

a. the digital signature of the WSC system is valid

b. the WSP system certificate is valid and not revoked

c. the WSP is on the white-list of systems that are allowed to enter the federation

d. the user's digital signature is valid

e. the user's certificate is valid and not revoked

Step 3. The STS now seeks to verify that the client-specified core attributes are valid by using backend attribute services. Some of these verified attributes are cached for a short period for optimization purposes.

Step 4. If everything is OK, the security token is digitally signed by the STS and returned to the WSC.

Step 5. The security token can now be used in interactions with different WSPs until it expires.

Step 6. Upon receipt, the WSPs validate the security token by verifying the STS digital signature and leverage the embedded attributes for logging and authorization.

Step 7. Finally a result, i.e. business information or an error is returned.

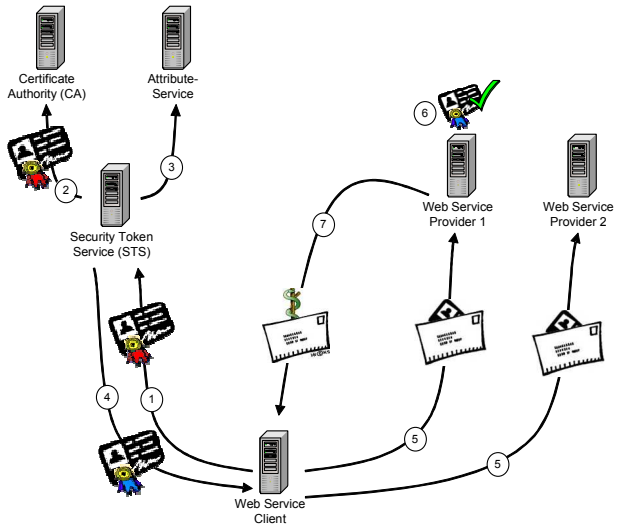


Figure 1: a simple WSC/WSP interaction

It is important to note the temporal flexibility between steps 1-4 and steps 5-7: The authentication request for the STS could be executed as part of the user's log-on to the WSC system. They could even be performed asynchronously and would only become blocking if the user entered a step in a workflow where entrance to the federation was needed, for instance in order to gather information from outside the system.

What is actually happening in the STS is that the user's long lived credentials are converted to short lived credentials combined with core attributes. In the SOSI project these short-lived security tokens were coined "virtual health professional identity cards" (or ID cards for short). The STS issues digitally signed ID cards that by PKI properties are verifiable by service providers without on-line access to the STS or attribute services.

In effect the federation is established by the sole STS certificate, which also means that the STS certificate must be protected viciously. In the SOSI project it was discussed whether a security breach on the STS should be handled by policies and emergency procedures (e.g. by phoning all service providers and having them shut down their services) or having all service providers check the STS certificate for revocation from time to time.

The latter seems to be the only manageable solution and was decided upon. The solution hence yields a system dependency from WSPs to the CA revocation service, but these calls are done infrequently and independently of the high volume business calls. The risk of compromising the federation certificate is comparable to the risk of compromising the CA root certificate, which is considered to be very low.

The maximum validity of ID cards is 24 hours in the SOSI architecture. However, the amount of trust a WSP can put into the security token depends on how "fresh" the token is. In other words the level of trust degenerates over time.

If the token is 5 minutes old when received by a WSP, the WSP can be pretty confident that the same user is still operating the console. The SOSI proposal opens up for the possibility that a WSP can choose to reject security tokens that are "not fresh enough" at its own discretion. In effect this means that users are, within 24 hours, *only* challenged to authenticate themselves when they use a service that needs authentication proof, which is more "fresh" than the security token the user currently holds.

It is worth noting that this mechanism is in direct opposition to the "single-sign-on" requirement: If all WSPs reject ID cards that are more than 5 minutes old, the user will be forced to re-login to the federation every 5 minutes, effectively disabling SSO.

This kind of time-out requirement should, however, only happen for service-operations, which provide or receive very sensitive information and hence demand very rapid security token time-outs, which means they are more likely to require a real digital signature and hence entail the end-user to provide credentials for activating the private key.

The decision, of which credential strength, security token time-out level and which verifiable authorization attributes the specific service provider should require, must be based on a thorough risk analysis.

The time-out level is a true measure of "trust", and hence a scheme where WSPs' have different time-out requirements for different (types of) client systems can be considered. In a national service infrastructure this can also be used for governing client systems towards federation compliance.

For the STS and WSPs' to be able to distinguish which client system an ID card pertains to, the request needs to carry information about the "system-principal" that is performing the request on behalf of the user.

The current profile mandates that any request will carry two digital signatures: One from the user on the SAML assertion and one from the requesting system on the entire message, which provides a combination of system identity and message authentication.

The STS will check both signatures as well as both certificates and will acknowledge successful verification by embedding references to the original certificates, and sign the ID card itself. By including secure certificate hashes into the issued ID card, WSPs will be able to verify message authentication signatures without checking the client-systems' certificate for validity or revocation since the revocation checks have been performed "recently" by the STS.

2.3 A New Profile

SOSI defines a web service profile where every request and response message will carry an assertion that identifies the sender, and every assertion will contain credentials that allow the receiver to verify the identity of the sender.

Credentials come in the shape of digital signatures over the XML elements that make up the SAML assertion stored in compliance with the XML-Signature specification [26]. A receiver can use the unique certificate identifier to lookup the person or company via trusted web services.

Routing information is embedded in the SOAP messages to enable other transport mechanisms than HTTP. WS-Addressing [26], the de-facto standard for such information in web services, is leveraged and profiled. WS-Addressing contains a unique message-id, which is required on all messages in the shape of a Universally Unique Identifier (UUID), used to prevent replay attacks.

All messages are integrity protected through an additional digital signature on the XML that makes up the SOAP body, the WS-Addressing headers, and the SAML token. This ties the sender's identity to the supplied information and prevents identity theft: Without this signature, an eavesdropper would be able to create a new message and embed a stolen SAML assertion, effectively impersonating someone else.

Messages are confidentiality protected only when in transit via the underlying transport layer. The federation is made up only of well-known parties who have signed an agreement before being granted physical access. Further each party is bound by laws not to disclose sensitive information, and it was therefore decided that it was not necessary to employ message encryption.

The profile defines an SSO mechanism, which uses WS-Trust [15] messages to perform authentication via a trusted third party, the Security Token Server (STS). WS-Trust was chosen over its SAML equivalent because it seemed to have the most momentum with respect to actual implementations in products at the time.

The profile uses a request-response model and leverages the SAML specification's SOAP binding to embed the assertions in SOAP headers. In this respect, the profile is completely in compliance with the SAML specification.

3. IMPLEMENTING SOSI

At the time of writing, the pilot project is being tested on a smaller scale, yet many relevant observations have been made not least in the process of realizing the architecture.

3.1 Participating Systems

From the outset of the SOSI project, two existing hospital systems were planned to implement the solution: One in the capital region of Denmark and one in the region of South Denmark.

More specifically it was planned to improve the quality of available patient related medicines information by connecting the medication modules of these systems with the national medication and prescription services hosted by the Danish Medicines Agency. The latter, then had to be enhanced as well to be able to participate in the federated solution. The fourth party in the system setup was the authentication service (STS).

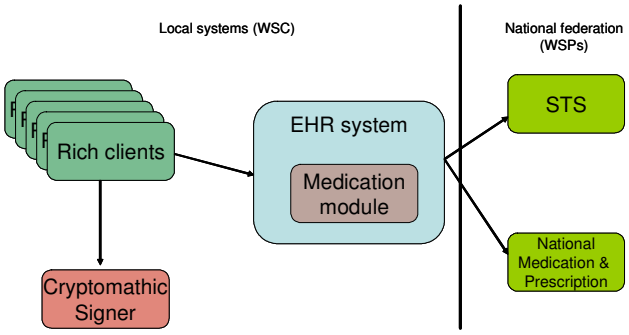


Figure 2: Participating systems overview

Since health care personnel moves around and uses the rich client part of the medication applications at different terminals, PKI private key handling is complicated somewhat. Fortunately this problem had already been identified and solved by the regions when the clinical workplaces were integrated with the Danish

eHealth Portal [23]. In both cases a regional "signing server" [1] was introduced that centrally handles and protects private keys for all end users (see figure 2).

When the EHR system needs to authenticate the user with the national federation, it challenges the rich client with parts of the STS request. Once the signed challenge returns to the EHR system, it can produce the final STS request and store the returned ID card locally for 24 hours. As the user then moves to another terminal, the same ID card can still be used in the national web service federation.

The requirements of the STS service were specified and quotes were requested from relevant vendors. The STS was required to handle the following number of requests for ID cards with the following response times for the pilot test period:

- Verification and signing of 12.000 ID cards in 24 hours
- A maximum continuous throughput (MCT) of 1500 ID cards per hour, with a peak of 10 simultaneous ID card requests.
- Mean response times < 2 seconds while upholding MCT
- 95% percentile < 5 seconds while upholding MCT
- 99% percentile < 10 seconds while upholding MCT

Of the two quotes received, one based was on an off-the-shelf solution, and one was custom development. The off-the-shelf solution was considerably more expensive than the custom solution and had some annoying limitations in the usage of SAML/WS-Trust. In addition, the modules that had to be developed to encompass the special health sector needs were proprietary, which made it difficult to move to another product without considerable expenses. The STS was optionally requested to be developed under a reciprocal Open Source license, which only the custom solution adhered to.

The custom solution was chosen for the pilot project and has now been developed and committed to the Public Danish Open Source initiative [12]. This fulfills the needs for the SOSI pilot project and a few soon to come smaller projects, but as the national federation develops, and the commercial STS market is maturing, the custom made STS will most probably be replaced with an off-the-shelf product.

At time of writing 3 of 4 systems have been developed/enhanced and a very slow scale-up has begun. Until now only a few doctors have been authorized to use the new facilities in their clinical systems and only a few dozens of ID cards have been retrieved from the STS. Nonetheless users have welcomed the solution and can see the benefits and perspectives right away.

The overall work flow has performance issues, which are currently being analyzed by the vendors. On the STS the biggest bottleneck is the OCES web service that resolves the Danish person ID from the PKI certificate. This STS back-end verification more often than not accounts for a 2 second "delay" which of course is unacceptable.

One of the problems in the current state of the project is that the systems are actually misleadingly slow because they were built for higher transaction volumes. In the current sparse user volume, caches and prepared database statements and connections are getting evicted between invocations, which results in unnecessary startup penalties almost every time the user is accessing the system.

When the (real) performance problems have been solved, more doctors will be allowed access to the system, and the false startup penalties should disappear. Since the new facilities can be activated on a per user basis, the systems can be scaled at a very fine grained level, enabling the monitoring of STS and WSP systems very carefully as more users get access.

3.2 Lowering the Threshold

The outlined architecture puts quite a burden on service providers and clients. In order to join, a party must be able to handle SAML, speak SOAP over HTTP, create and verify digital signatures, communicate with a trusted third party, check revocation lists, and more.

Because the architecture is based on standards it might be possible to find commercial off-the-shelf (COTS) products that would be able to handle these tasks. Each product would require some configuration, though, and might have a steep price tag on it. In some parts of the health sector, in particular at the GP's, steep prices are not an option.

Further, the federation is built on existing software, which was not initially meant to communicate with other systems. Hence, any implementation of federation infrastructure will include some amounts of custom code.

It is crucial to the success of the health federation that all parties have the means to join in. As a remedy to the situation, it was decided early on to establish an open source organization, which would provide software that implemented federation infrastructure and that could support the vendors that were to use the Open Source products.

A set of tools would make it viable for even small companies to join in, lowering the threshold and enabling the federation.

Faced with a lack of product support due to a lack of profiles for SAML based web service interactions, it was clear early on that support for the SOSI profile would have to be implemented into every IT system in the federation in a custom manner.

While the SAML AttributeStatement although somewhat verbose in its syntax is not hard to implement, creating XML digital signatures is an entirely different story.

A programmer whose development platform does not support the XMLDSig [28] standard out-of-the-box will have to piece signing and verification functionality together e.g. from a crypto API. This includes creating secure hashes of data, implementing canonicalization algorithms, encrypting and decrypting, base 64 encoding and decoding, manipulating XML structures, and more.

As a remedy to this ailment it was decided early on by the open source organization to build a Java based library, "Seal.Java" [2] that would provide an abstraction, which would allow a developer to work with high-level primitives and not worry about envelope formats, digital signatures, or the darker secrets of the base-64 algorithm.

The EHR systems that entered into the SOSI project from the hospital side were mainly Java based, and while Seal.Java was relevant here, it could not be used with the EHR systems from the GP side that are mostly rich Win32 or .NET applications. This fact spawned Seal.NET [10], with the exact same purpose as its Java sibling.

Both projects have been constructed on an Open Source license and are available for general scrutiny via the web.

Third party software often suffers from the "not invented here syndrome", a problem which the library projects sought to address by going to great lengths in testing, tuning, and publishing quality reports. When response times are high, multi-threading issues fixed, code coverage of the test suite well above 95%, and long term endurance testing of all API methods does not show any leaks; when the entire library is built from scratch, and all tests exercised on a nightly basis with fresh results published online in the morning [3], chances are that others will accept it as stable and useful as well.

This aggressive strategy for quality reporting has proven to be highly effective. Adoption of both libraries is high with most peers using them. This makes it very easy and low-cost to implement minor adjustments and optimizations to the SAML profile, because most vendors just need a new version of the library to be able to produce a new request or response XML.

Parallel to the developed Open Source components, a support organization was established. Vendors can contact a support mailing list for free and very rapidly get answers to general work flow or detailed coding questions, or solve problems or advice to workarounds with the libraries within hours. The response to this sort of support has been very positive and has had a very positive influence on the vendor / customer relationship.

3.3 The XML Schema Challenge

During development of the libraries, the idea surfaced that it would be useful to implement XML Schema validation for the XML, SAML, SOAP, WS-Trust, etc. that was passed around. Validation would improve overall quality and general faith that standards were followed.

Unfortunately that proved to be very difficult.

A profile that cuts across specifications is in effect limiting the number of possible choices a developer can make. Wouldn't it be great if it were possible to express the new set of limited choices in supporting schemas as well? It isn't! For instance, how do you express that it is a requirement to have an enveloped signature inside a SAML Assertion if the user authenticated using PKI?

Expressing such complex conditions is beyond and above what you can do with XML Schema. Even if it were possible, the problem of how to version a set of XML Schemas in concert arises: There is no great way today in which existing schemas can be narrowed under the same name space.

For development purposes, it was therefore decided to modify the original schemas, SAML, SOAP, etc. to allow only those elements that were mandated by the profile. While helpful for testing, these schemas would not be used for production because they were overly strict and hence not compatible with COTS products that will often attach extra non-critical SOAP headers, id's, etc.

Recently, a central test center for web services in the Danish health sector [11] has been launched. The test center is capable of emulating clients and servers for various concrete services to a certain point not including too much business logic. It is manned by staff that can monitor requests and responses, and aid in debugging. The center provides value in ensuring that all parties wishing to implement a service will get past syntactical obstacles with the profile as well as with the model of the service in question.

For reasons of maintainability, loose coupling, and reuse, web services should be designed in a contract-first manner, where the service interface, the WSDL, including data models and service end-points, is defined independently of the code that implements it.

Unfortunately not that many off-the-shelf toolkits give good support to such a development paradigm. Now that tooling was already being implemented, it was decided to craft a contract-first WSDL tool that would allow for the easy creation of service interfaces as well.

Tooling is an important mechanism to help bridge the gap between specifications and products. Tools can make the difference as to whether a particular IT system will be able to participate in a certain scenario or not and without them, the SOSI project would not have been possible.

While providing tools and libraries to lower the threshold of integrating existing systems, there is also a risk associated with such a strategy: Source code, no matter how well written, will always have flaws, errors, or lack a feature for a given situation. Without an organization to maintain the code, it will eventually fail to be helpful..

On the other hand, it is actually possible to tune the profile over time or align it with coming standards, when all parties rely on a few infrastructure components. Given the volatility of the current specifications for federations of services, this might prove to be a crucial strength.

3.4 Federation Verification and Control

Because the federation is established by digital signatures from the authentication service (STS), all parties in the federated infrastructure are able to check federation security tokens (ID cards) by validating the STS signature and the STS certificate.

Checking the signature is a matter of working with XML-Signatures, something that enjoys library support in many programming languages. The STS certificate check is a little more involved as the entire chain up to the issuing authority must be validated including revocation checks. Fortunately it can be done rather infrequently due to the very low risk that STS credentials should get compromised. As a hardening measure, the SOSI production STS has been placed in the same production room as the Danish OCES CA services, hence the same policies for access, audit etc. are enforced for this system.

Having one key pair that establishes a federation makes it easy to establish test federations e.g. for pre-production, integration-test and other development stages. This has been employed in the SOSI project, where the OCES CA has issued two certificates – one for production and one for integration/pre-production.

However, the STS production certificate will expire some day, and various mechanisms have been put in place to make a smooth transition from one certificate to another. For instance the Seal.Java library has STS certificate checks incorporated that are resistant to STS certificate renewal.

While developing and testing the solutions, the vendors have had some trouble with the VPN based dedicated health network (SDN) through which all production services are accessed. None of the vendors could be authorized to access SDN directly as they are not public health related companies, and since the production servers are never on two networks at the same time, it was very cumbersome getting access to logs, debugging information, not to mention changing configurations or deploying new software versions.

As mitigation for this, the test federation was established on the public Internet, where developers can access a test STS from their development environments. Unfortunately some services e.g. back-end ones used by the STS for core attribute verification only exist on SDN and the semantics and performance aspects can therefore only be tested in the production environment. This is one of the issues that must be resolved in the coming national infrastructure.

3.5 Standardization

There is no profile without a specification, and the SOSI efforts have therefore been captured in a document known as The Apt Web Service (DGWS) [8].

Owned by the Danish Centre for Health Telematics (MedCom), who is responsible for standardizing the communication between parties in the health sector and operating SDN, the specification has now reached version 1.1.

While there might be a few adjustments to make on DGWS in the aftermath of the first pilot, this version is currently poised to become the de-facto standard for all web service communication in the Danish health sector.

To ensure compliance, MedCom staff mans the aforementioned test center [11] and is providing practical assistance in testing that web service clients and servers implement not only DGWS, but also the business data exchanged in DGWS SOAP envelopes.

With specifications, tools, and a certifying entity in place for free, there should be a viable chance of getting even the smaller vendors on board the federation.

4. LOOKING FORWARD

Federated identity management has evolved over the past few years, and there are now a couple of frameworks that might address the needs in the SOSI architecture. Most notably, the Liberty Alliance recently published version 2.0 of its Liberty ID-WSF, which defines interaction scenarios for web service clients with SAML via SOAP over HTTP. Future work will examine Liberty and alternatives in order to evaluate whether it would be feasible to align the SOSI project without critical impact.

Parallel to the initiatives in the health sector, MVTU is driving other pilot projects that address slightly different needs, but define similar architectures. The OIOSI [24] project for instance is being pushed for secure asynchronous business document exchange via the internet using PKI and web services.

The health sector specific infrastructure must to be aligned with a future national infrastructure for all of the public sector without violation of the identified design criteria.

While digital signatures are currently being touted in Denmark as *the* technology to identify citizens and professionals alike, it is loved more by engineers than by end users. A digital signature is cumbersome to deal with and certificate management is not mature from an end-user's perspective.

On the longer term, biometrics and RFID for near field identification could have a place as the identifying technology, e.g. to release the private key of a certificate instead of a password. The driving force for biometrics or RFID will, however, not be the increased security, but the fact that identification will become easier for end users.

At the time of writing multiple initiatives governed by Digital Health Denmark that extend the SOSI architecture are in the crucible. Most notably a security gateway, SOSI-GW, is being developed that enables trusted domain cross-over. This vastly reduces the effort in implementing SOSI support for web service clients and will enable single-sign-on to the national federation across multiple client systems. The gateway becomes the single point of entry to the national web service federation from the trusted domain, and all sorts of common services can be centralized in this service.

Digital Health Denmark is also launching initiatives to analyze possibilities for limiting the impact when the back-end verification systems of the STS fail. One possibility is to allow the STS to issue partially verified ID cards as long as every attribute verification state is clearly stated. The STS may also cache verification states and skip re-verification if the cached verification has not decayed too much. This reduces the impact of failing verification services or decayed attributes to users who are using federation services that need these specific attributes.

Of interest is also the possibility to use "break the glass" solutions combined with logging and control mechanisms. If for instance a WSP requires a newly verified STS attribute, and the presented ID card contains a non-verified or decayed attribute, the WSP may choose to return a "break the glass" warning that informs the client system and subsequently the user that she will be subject to investigation if proceeding. This could be combined with asynchronous mechanisms that seek to resolve the unverified claims.

The SOSI project has not produced any results in the area of web service governance, but as the solution scales to multiple clients and providers this will become a very important issue. Digital Health Denmark is also launching initiatives to meet these challenges.

5. CONCLUSION

The proposed architecture and profile have been developed and tested in real life, and the results are very promising with respect to both the development process as well as the implementation effort.

At the time of writing end-user feedback has not been systematically gathered yet, but purely from a technical perspective the proposed architecture exhibits a set of nice qualities that support the special requirements for the health sector:

- **Single-Sign-On** to Web Services within the national federation / trust domain.
- **Authentication levels**. Users and systems can be authenticated with different degree of certainty, depending on the credentials that the principal presents. This is in accordance with the guidelines [13] from NIST on which MVTU has based their authentication guidelines.
- **Reduction of impact** of unavailability of services. If, for instance the STS is unavailable, only users without a security token or with an expired security token will be hindered in performing their treatment. All other users can continue to treat patients until their security token expires.
- **Reduction of the effort** that WSCs and WSPs must put into implementing web services. WSPs only have to trust/check *one* certificate (the federation certificate owned by the STS)

- **Maximum performance**. The number of requests/messages is minimized. When trust has been established and the user has logged in to the federation, the WSC and WSP communicate directly with no third party involved.
- **Transparency and flexibility** through the use of Open Source licensed tools and products.
- **Reuse of existing infrastructure**. The design reuses existing infrastructure for establishing secure channels that takes care of confidentiality and stream integrity and prevents known cryptographic attacks.

The positive experiences with the architecture and profile outweigh the downside of not yet having international standards that fit the requirements of the Danish health sector.

SOSI is currently acknowledged as the best solution to the integration challenge, and at the time of writing, multiple projects that implement modules and systems based on the SOSI design, its standards and the associated Open Source tools are in the making.

6. REFERENCES

[1] Cryptomathic, Cryptomathic Signer, http://www.cryptomathic.com/Default.aspx?ID=124

[2] Danish Regions, 2006-2007, SOSI Components, http://www.sosi.dk/twiki/bin/view/ProjectManagement/SOSIProducts

[3] Danish Regions, 2006-2007, SOSI Seal Component, http://www.sosi.dk/sosi/seal/

[4] Digital Health Denmark, 2007, http://www.sdsd.dk/

[5] Digital Health Denmark, 2007, National IT strategy, http://www.sdsd.dk/arch/_img/9080664.pdf

[6] Internet2/MACE, 2007, Shibboleth Project – Internet2 Middleware, http://shibboleth.internet2.edu/

[7] Lockhart et al., 2007, Web Services Federation Language, http://www.ibm.com/developerworks/library/specification/ws-fed/

[8] MedCom, 2003-2008, The Danish Health Network, http://www.medcom.dk/wm110002

[9] MedCom, 2007, The Apt Web Service (DGWS), http://sundcom.health-telematics.dk/svn/DGWS/

[10] MedCom, 2006-2007, Den Gode Webservice Tools, http://www.medcom.dk/wm110344

[11] MedCom, 2007, Testcenter, http://testcenter.medcom.dk/

[12] National IT and Telecom Agency, 2007, Software exchange: Forum for software development in the public sector, http://www.softwareborsen.dk/

[13] NIST, Electronic Authentication Guideline, 2006, http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf

[14] OASIS, 2007, SAML 2.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#samlv20

[15] OASIS, 2007, WS-Trust, http://docs.oasis-open.org/ws-sx/ws-trust/v1.3/ws-trust.pdf

[16] PGP Corporation, 2006, PGP White Paper – Revocation Made Simpler, http://download.pgp.com/pdfs/whitepapers/Revocation-SLCs_060104_F.pdf

[17] Profile for Short Lived Credential Services X.509 Public Key Certification Authorities with secured infrastructure. http://www.tagpma.org/files/IGTF-AP-SLCS-20051115-1-1.pdf

[18] TDC, 2007, OpenOCES, http://www.openoces.org/

[19] The CIA Triad, http://en.wikipedia.org/wiki/Information_security

[20] The Danish Data Protection Agency, The Act on Processing of Personal Data, Datatilsynet, Law no. 429, May 31st, 2000, https://www.retsinformation.dk/Forms/R0710.aspx?id=828

[21] The Liberty Alliance, 2007, Liberty Alliance Project, http://www.projectliberty.org/

[22] The Ministry of Health and Prevention, The Health Law, Law no. 546, June 24th, 2005, https://www.retsinformation.dk/Forms/R0710.aspx?id=10074

[23] The Ministry of Health and Prevention et al., 2007, Sundhed.dk, http://www.sundhed.dk

[24] The Ministry of Science, Technology and Innovation, 2006, OIO Serviceorienteret Infrastruktur, http://www.oio.dk/arkitektur/soa/infrastruktur

[25] The Ministry of Science, Technology and Innovation, 2006. The Interoperability Framework. http://standarder.oio.dk/English/

[26] W3C, 2006, Web Services Addressing 1.0 – Core, http://www.w3.org/TR/ws-addr-core/

[27] W3C, 2002, XML-Signature Syntax and Processing, http://www.w3.org/TR/xmldsig-core/

[28] W3C, 2002, XML-Signature Syntax and Processing, Recommendation. http://www.w3.org/TR/xmldsig-core/

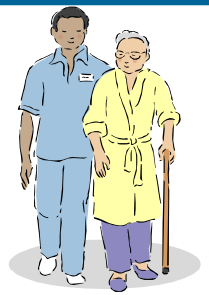
A Federation of Web Services

For Danish Health Care

IDTrust 2008

Kåre Kjelstrøm,
kkj@silverbullet.dk

The Danish Health Sector



Caregiver

Few systems
PDA access

Hospital
Doctor

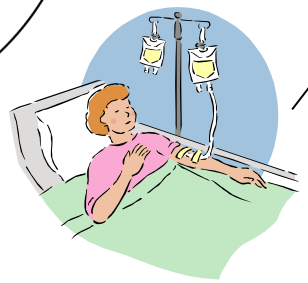
Many systems
Many computers
Doctors
Nurses

Exchange

Information

Single system
Doctor
Secretary

General Practitioner



Patient

Web access to
own data

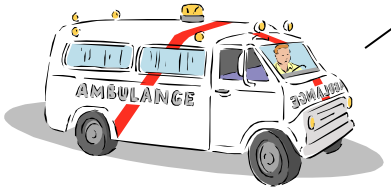
DIGITAL SUNDHED

SAMMENHÆNGENDE DIGITAL SUNDHED I DANMARK

Requirements:Privacy vs. Safety

Authentication &
Authorization

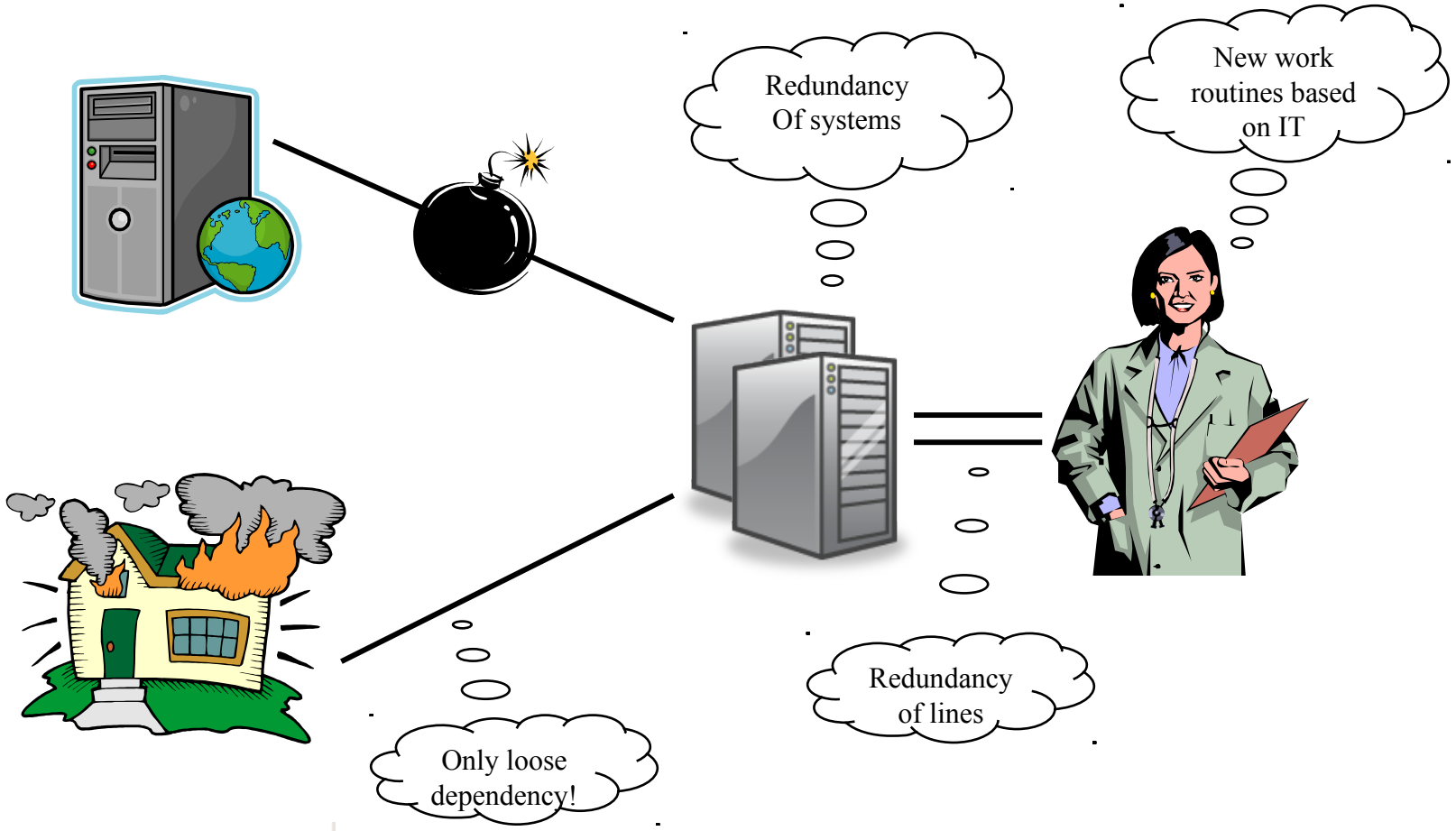
Confidentiality



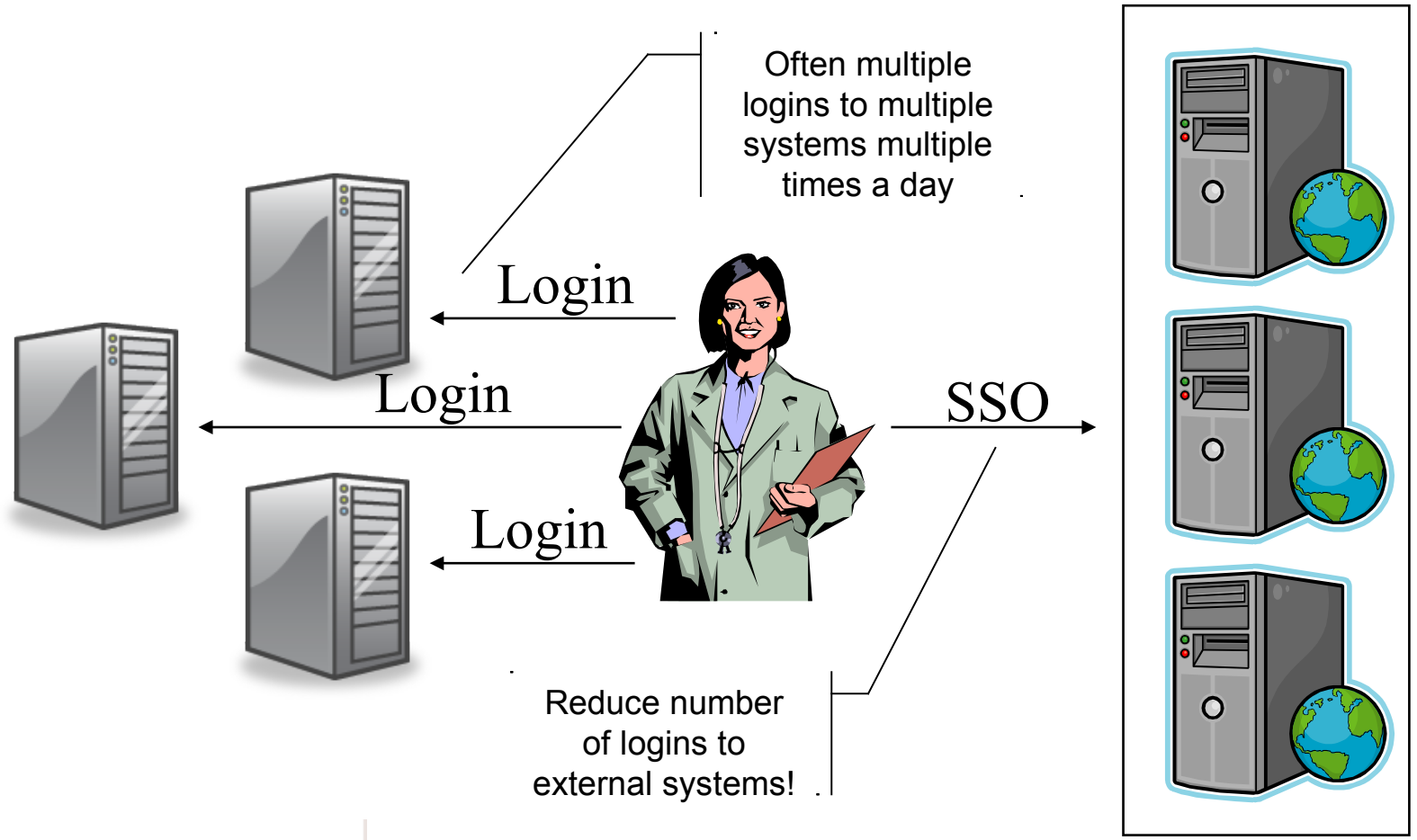
Emergency Access

Record Activity

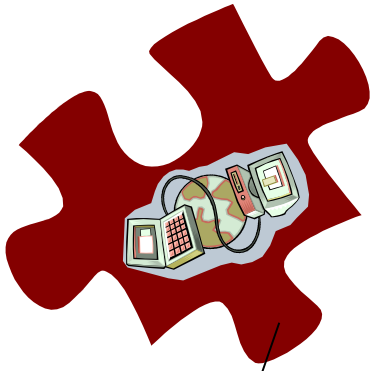
Requirements:Availability



Requirements:Single-Signon

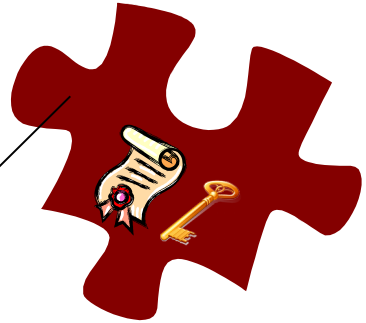


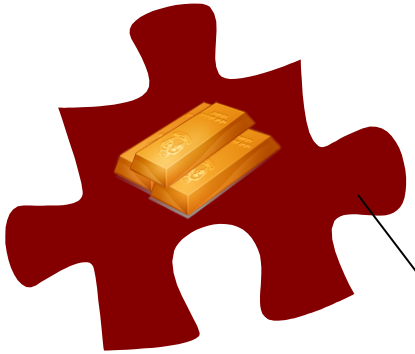
Requirements: Preconditions



Health care network (VPN)

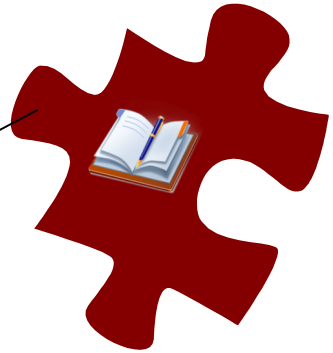
National PKI infrastructure with SSN lookup



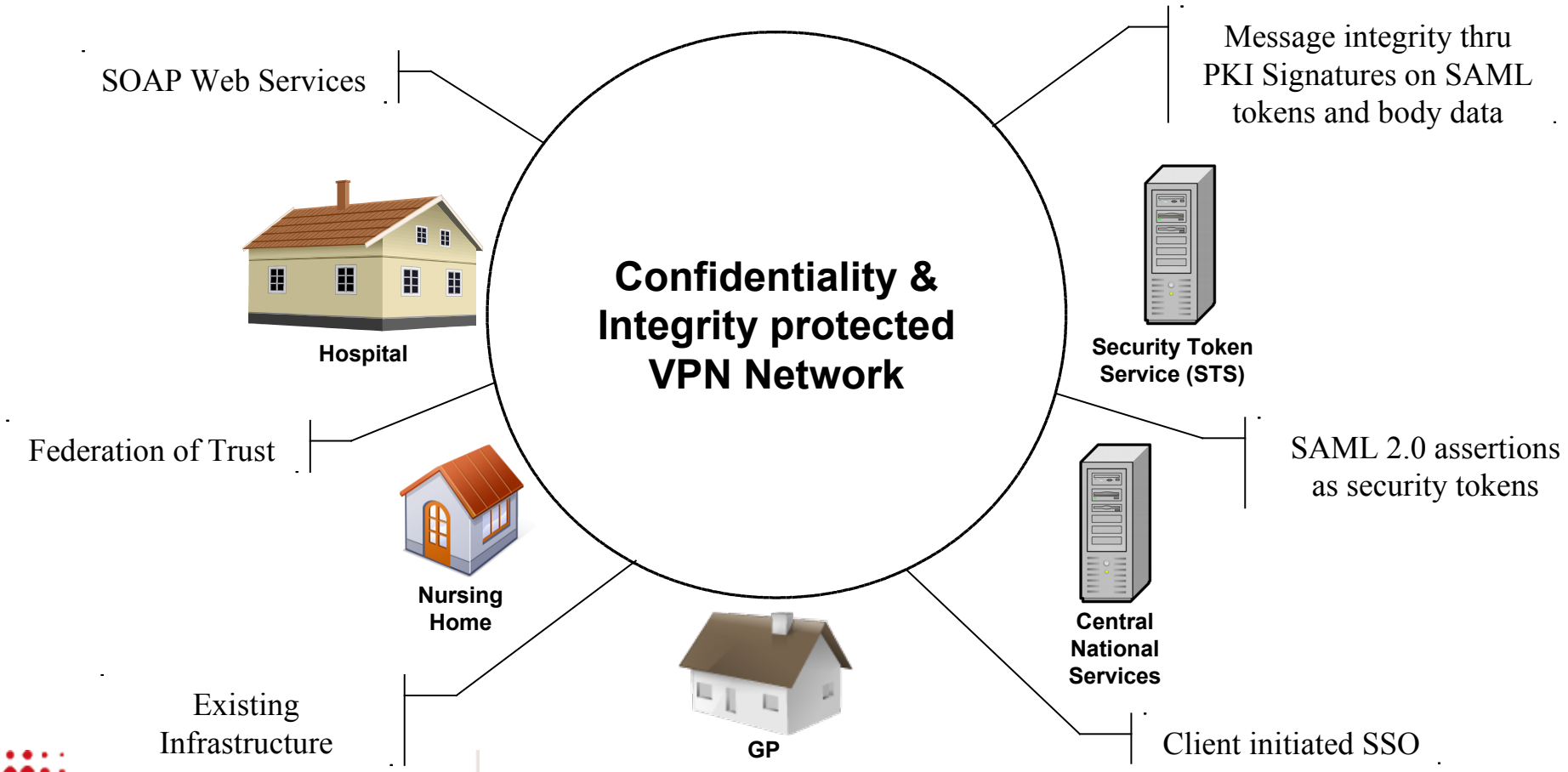


Rich non-browser clients

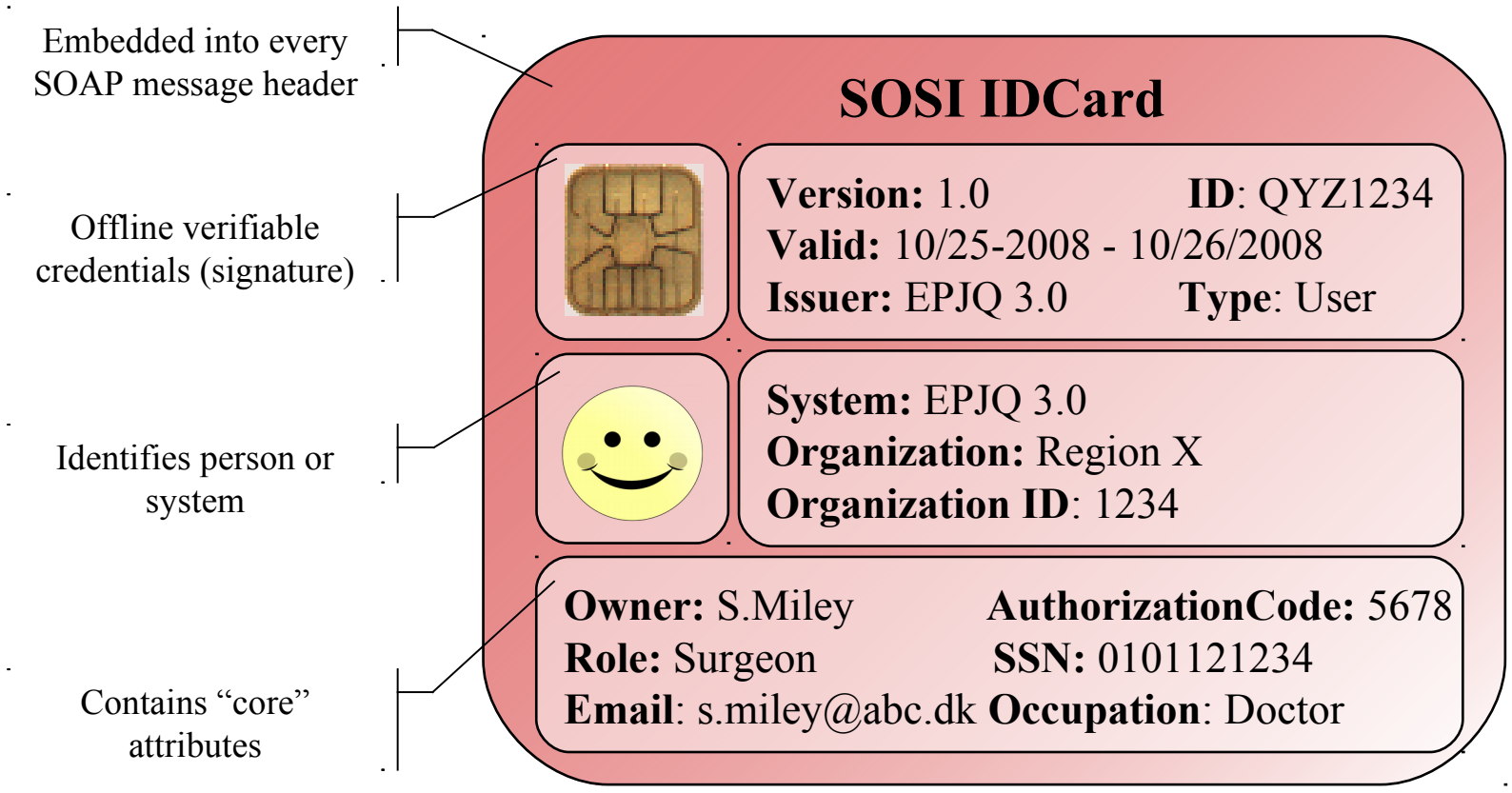
Existing specifications and profiles



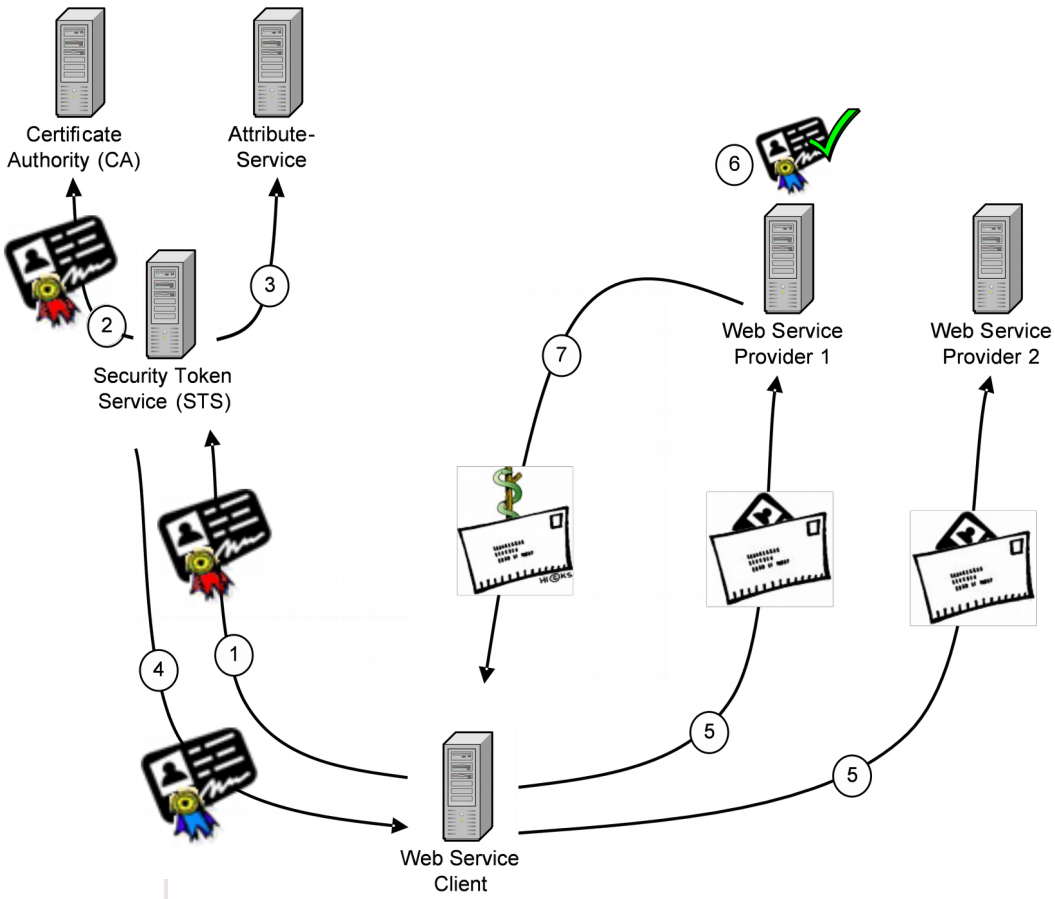
High Level Proposal: SOSI Service-Oriented System Integration



ID card: SAML Security Token

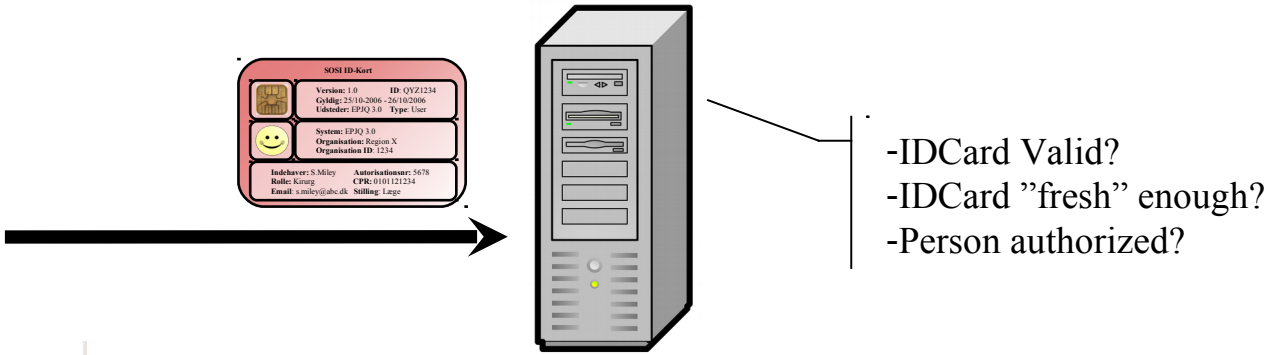


Service Interaction

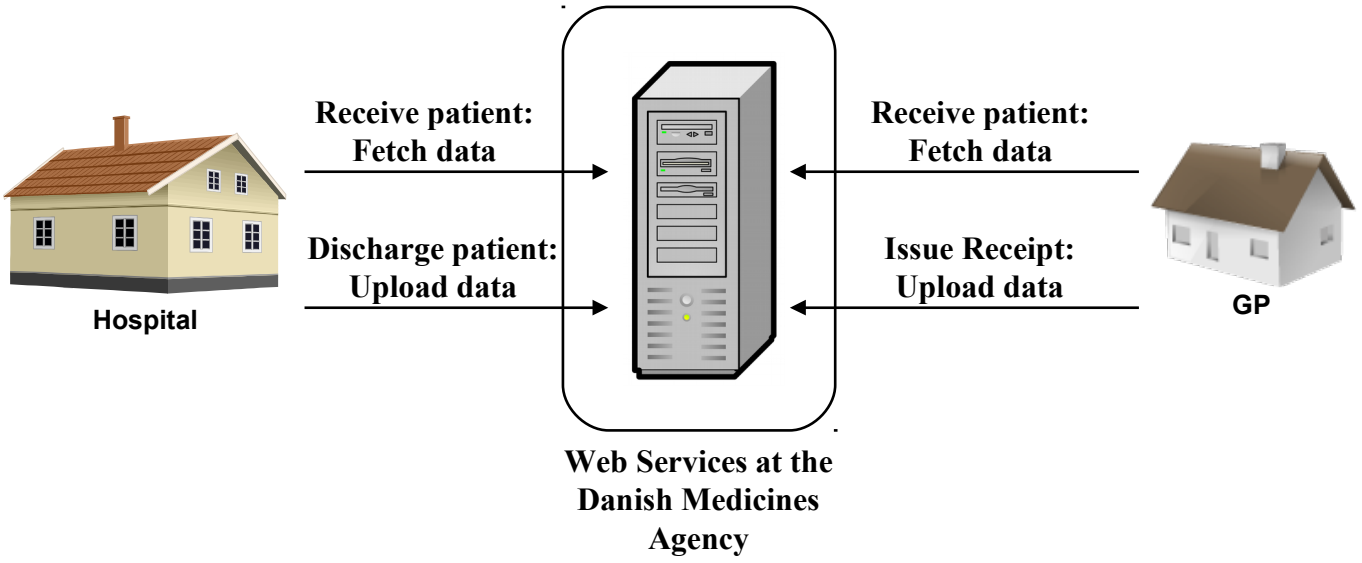


Timeout

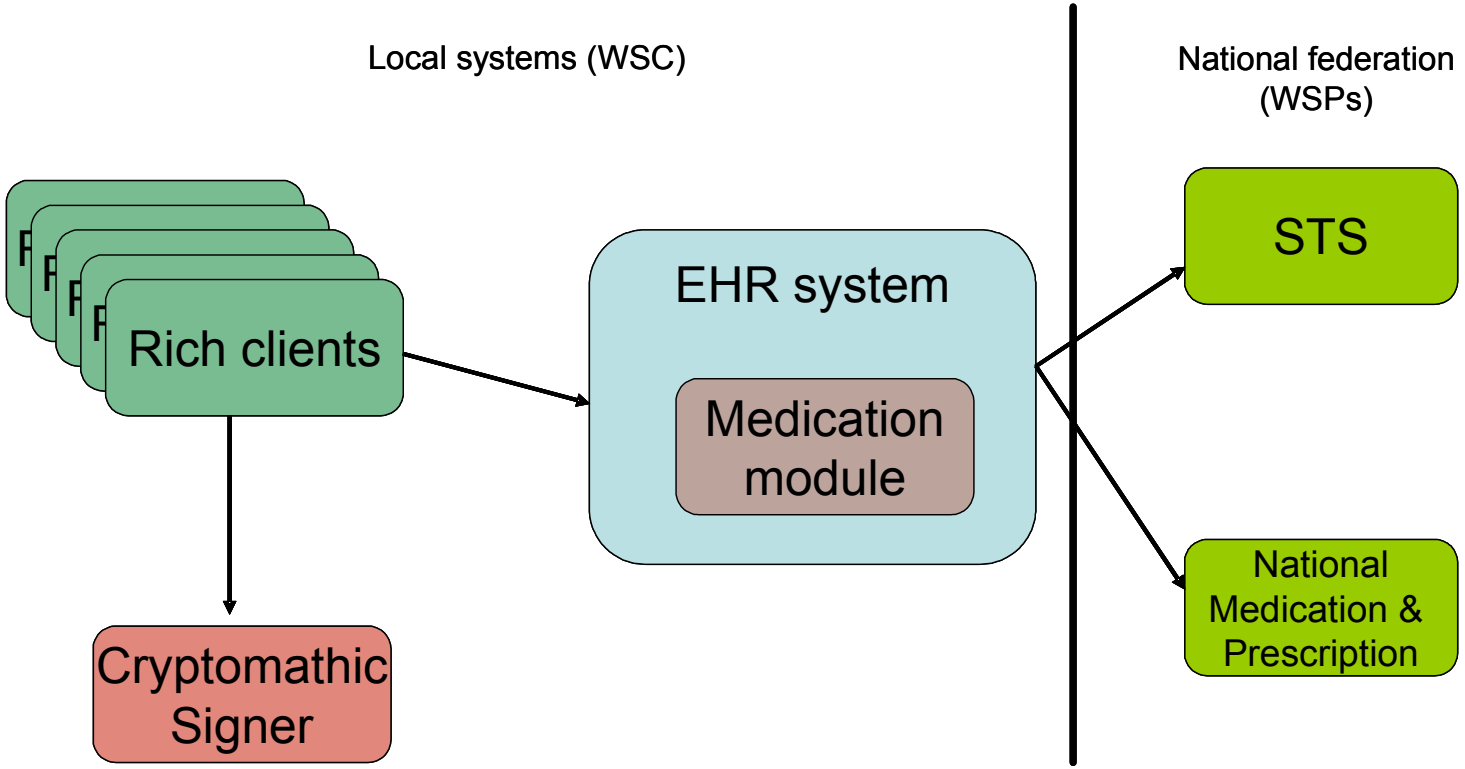
- Maximum ID card lifetime: 24 hours
- Authorization by service provider
- Service provider decides timeout level
- Based on risk analysis



Trail Blazer: Medicines Information



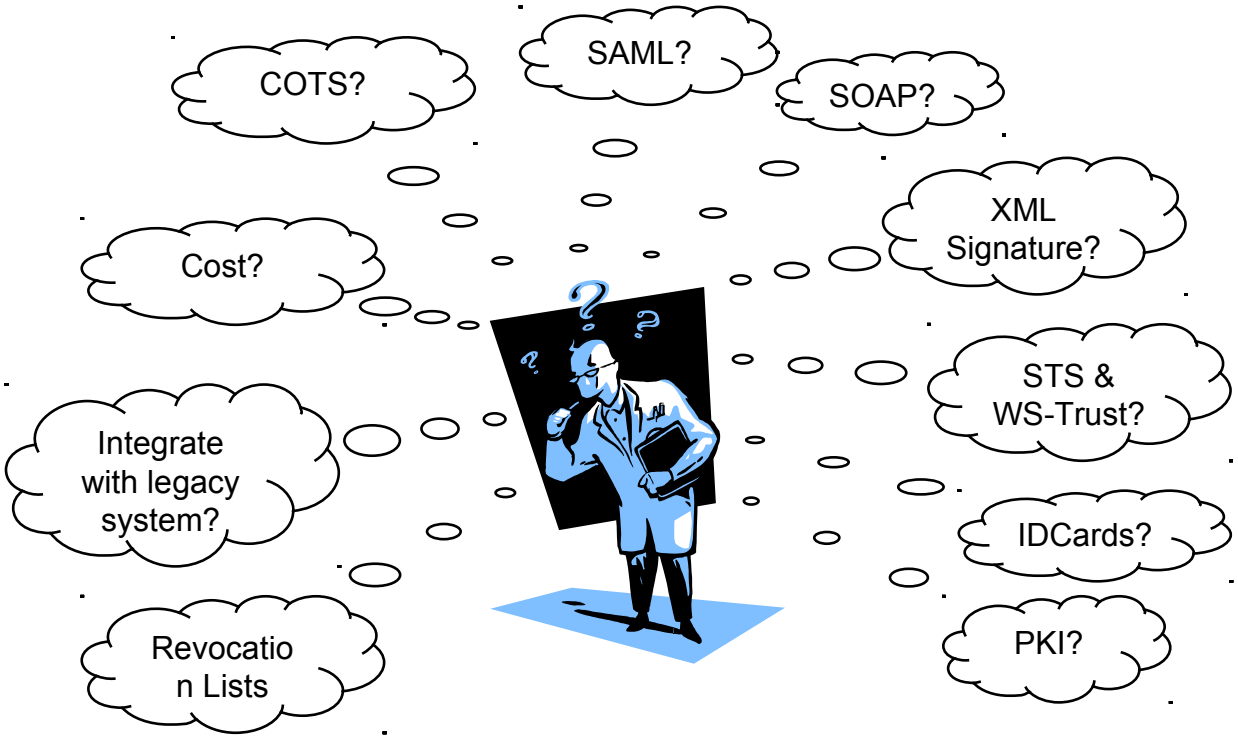
Participating Systems



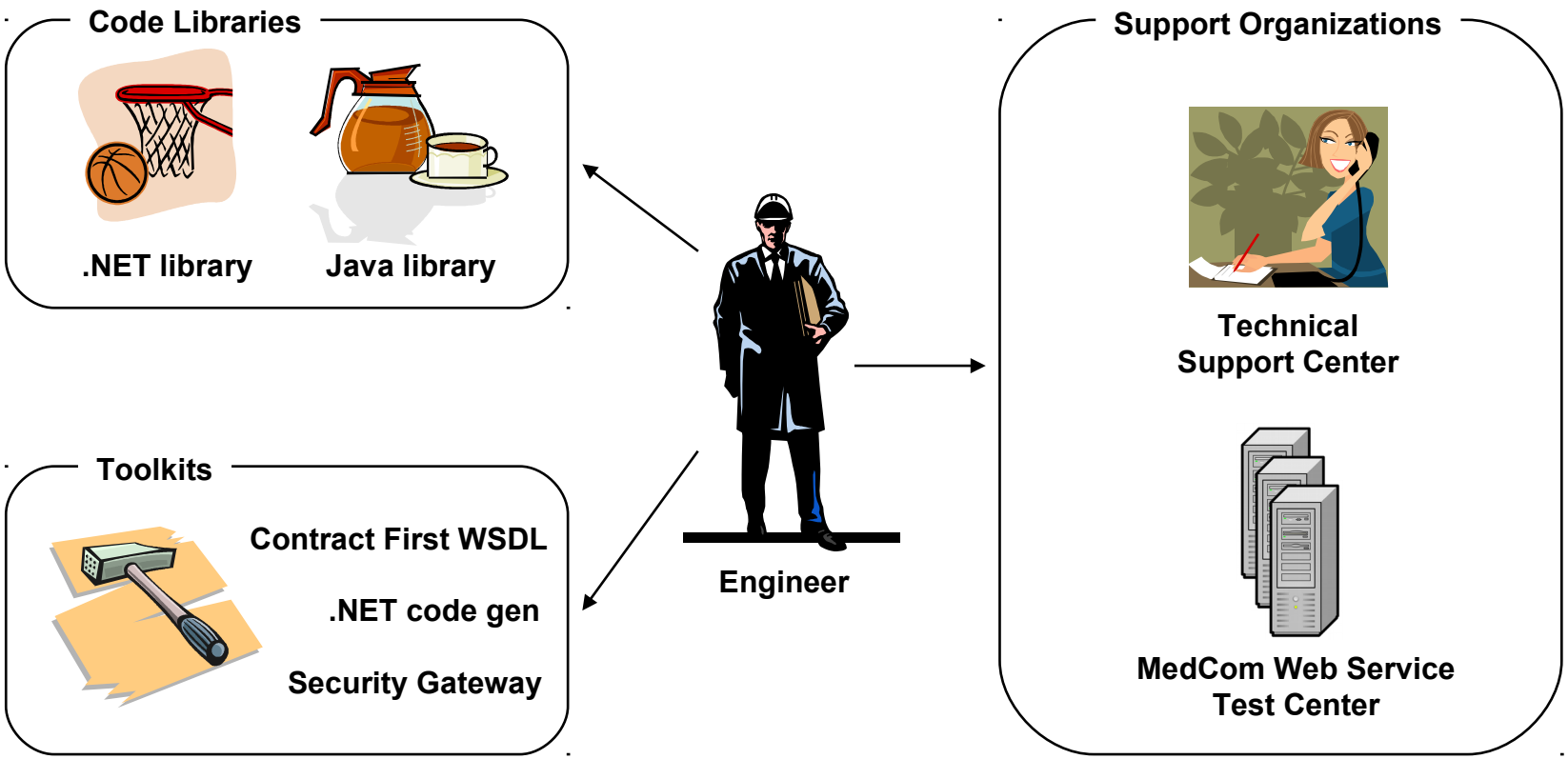
STS Performance

- Verification and signing of 12.000 ID cards in 24 hours
- A maximum continuous throughput (MCT) of 1500 ID cards / hour, with a peak of 10 simultaneous ID card requests.
- Mean response times < 2 seconds at MCT
- 95% response times < 5 seconds at MCT
- 99% response times < 10 seconds at MCT

Implementing SOSI?



Lowering the Threshold



Looking Forward

Partially verified IDCards?

Biometrics, RFID, Near Field Identification?

”Break the glass” solution: Heightened control

Alternatives from National IT- and Telecom Agency?

Service governance

Liberty ID-WSF 2.0 a replacement?

Summary

- **Federation of health care systems** using SOAP web services, SAML and WS-Trust
- **Single-Sign-On** to Web Services within the national federation / trust domain.
- **Reduction of impact** of unavailability of services.
- **Reduction of the effort** that WSCs and WSPs must put into implementing web services.
- **High performance** architecture where the number of requests/messages is minimized.
- **Transparency and flexibility** through the use of Open Source licensed tools and products.
- **Reuse of existing infrastructure**. The design reuses existing infrastructure for establishing secure channels that takes care of confidentiality and stream integrity and prevents known cryptographic attacks

Questions

Security and Privacy System Architecture for an e-Hospital Environment

Kathryn Garson
School of Information Technology and Engineering (SITE)
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5
kgars062@uottawa.ca

Carlisle Adams
School of Information Technology and Engineering (SITE)
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5
cadams@site.uottawa.ca

ABSTRACT
Hospitals are now using electronic medical records and computer applications in order to provide more efficient and thorough care for their patients. The Mobile Emergency Triage system provides doctors with decision support for emergency care by pulling information from a patient's health record and a medical literature database. In order to achieve compliance with privacy legislations PIPEDA and PHIPA, security and privacy measures must be put in place. Encryption and access control are necessary for ensuring proper authorization and confidentiality for patient records. Strong authentication and audit logs are required to ensure access only by those allowed. We discuss differences in security technologies and detail the ones used in our MET system. A new encryption technology called policy-based encryption proves to be quite useful within a health care environment for providing both encryption and access control. We propose an extension to an existing scheme which allows for the use of this cryptography in a hospital setting.

Categories and Terms:
D.4.6 [Operating Systems]: Security and protection
K.6.5 [Management of Computing and Information Systems]: Security and protection
E.3 Data Encryption

1. INTRODUCTION
Many hospitals are moving away from paper-based medical records to use electronic health care records. Specialized software and electronic diagnostic tools are offering a new level of patient care. The move towards electronic based systems provides streamlined automated processes and specific applications that can help doctors with diagnosis and treatment of patients. The introduction of these technologies raises privacy risks with regards to patient information. A malicious person trying to compromise many patient records will be able to collect large amounts of data easily if these records are available electronically.

IDtrust '08, March 4-6, 2008 Gaithersburg, MD

The Mobile Emergency Triage (MET) project provides doctors with decision support for triage, diagnosis, and treatment of patients in an emergency room setting. The project will be implemented first as a trial and then a full production version in the emergency room of a hospital in Ottawa, Canada. Doctors will be using a tablet PC with the MET software to help them in their tasks. The MET software pulls information from a collection of medical literature and electronic health records (EHRs) to provide doctors with evidence-based decision support. The agent-based system is interactive allowing the doctor to input symptoms and view possible diagnosis and treatment options. The doctor can make treatment decisions based on this information or enter additional data and receive different information.

The MET system is set up on a wireless network giving doctors the ability to travel patient to patient while having full access to the software. This system requires security and privacy technologies to prevent malicious users from having access to sensitive patient data. Patients' EHRs are transmitted over a wireless network to these devices and then stored on the PC. Proper access control needs to be put in place to ensure only authorized users can have access to records. We will discuss in this paper the steps we have taken to ensure privacy of patient's medical records. This project has greatly benefited from the input of a number of disciplines while working on the system. Management, computer science, engineering and medical science professionals have been working on this project together. Getting opinions and ideas these diverse backgrounds has been a great opportunity.

The goal of this paper is to present the technologies to be used with the MET project to add security and privacy functionality. We also discuss our motivations for adding security to this project based on privacy laws in Canada. Different alternatives will be discussed for their advantages or disadvantages in this setting. One of the most promising technologies for use in a health care environment is policy-based cryptography for access control. We discuss this concept and propose an extension to an existing scheme to add more flexibility for use in our environment.

Organization of paper
We first talk about the current environment of the hospital setting and what changes will happen with the introduction of electronic medical records. In section 3 we present our goals for securing our system and providing privacy measures. Then we compare access control and encryption methods available to satisfy these goals in section 4. We discuss a policy-based encryption scheme that works naturally in the health care setting. In section 5 we

compare authentication mechanisms for use with our system. Then we give some details of other security issues we faced and how we solved them in section 6. We give a brief overview of the system architecture and how each privacy module will interact within the MET system in section 7. We discuss related work in section 8 and conclude in section 9.

2. HOSPITAL ENVIRONMENT

The hospital environment is unique in both the fact that it contains such highly sensitive data and this data is required in emergency situations. Emergency situations may overshadow the need for privacy procedures. Doctor's main functions are to treat patients, not to follow complicated steps for accessing data. The workflow in the hospital must be taken into account when designing the system. Our goal should be to provide security for the MET software while minimizing tasks required by staff for using it.

Our trial will run in a hospital that currently uses a paper based system for medical records and is migrating towards electronic records. The paper medical record consists of documents and reports pertaining to one patient. It could include patient admission information, medical history, diagnosis reports, and lab reports from the current stay as well as previous hospital visits. Some parts of the medical record, such as lab reports which are available in electronic form, are printed and stored in the medical record file. These paper based records are considered the real authentic record even if the electronic form is still in existence.

The electronic system for accessing the available electronic documents is currently a read-only system since reports are not modified electronically. Employees sign in with their username and password to the software system on shared computers. The shared computers are in areas of high traffic and usually are already logged in to a general account. Each staff member who wants to access software or records will sign in to that software with their own information. Access rights are basic in that everyone who uses the system can read all records. There are groupings of staff and these groupings determine read and write access rights to documents in medical records. When the hospital moves to a full electronic system, these groupings will be defined formally in policies. Currently staff members only have access to the network from inside the building, but are moving towards remote login. This will not be a part of our initial trial but may be something to consider for the long term.

The procedure for obtaining a paper medical record is straightforward. A doctor can place a call to the records department requesting a document. Porters or other employees will bring the requested documents to their department. Records are assigned either to one doctor or one area of the hospital specifically, and are returned by the end of the day. If moved or passed on, medical records should be notified to be able to track the documents. Generally if a medical record stays in one area of the hospital is not necessary to notify them, since many doctors may see one patient during their time in that area.

3. PRIVACY GOALS

In Canada, the Personal Information Protection and Electronic Documents Act (PIPEDA) describes guidelines for the protection of personal electronic information[15]. PIPEDA provides guidelines for handling personal information in electronic form. The Personal Health Information Protection Act (PHIPA) is privacy legislation in the province of Ontario which builds on PIPEDA[14].

Principle 7 of PIPEDA describes the safeguards that should be in place to protect sensitive data. Here we highlight the aspects that pertain to our project:

1. **Sensitive information should be protected with a higher level of security**: Medical records are always considered to contain highly sensitive data. For our project we need to ensure that all patient data is secured by the best available methods. The sensitive information needs to be protected from unauthorized users during both transmission and storage.

2. **Methods of protection should include technological methods such as the use of passwords and encryption**: We should investigate different encryption and authentication methods to find which would be most suitable for a health care environment such as the one for the MET project. The most secure password scheme may be great for privacy but may not be feasible in a health care emergency environment. We need to find technological solutions that are practical.

3. **Limit access to a 'need to know' basis**: Access control methods need to be used. We can limit access to individual documents of a medical record for both read and write permissions. Employee roles within the hospital and permissions associated to those roles can be used in determining a good access control model.

4. Both PHIPA and PIPEDA specify that medical records should be **protected from loss, theft, unauthorized access, disclosure, copying, use, or modification** regardless of what format they are in. Electronic records must be protected by access control and include an integrity mechanism. Furthermore, we must have an audit functionality to ensure that no malicious use of the system goes unnoticed.

These privacy guidelines are not restricted to Canada. In the US, title II of the Health Insurance Portability and Accountability Act (HIPAA 1996) outlines standards for security and privacy of patient health records. They encourage the use of electronic data interchange in the health care system while protecting information. Physical and technical safeguards similar to the PIPEDA safeguards are outlined. Similarly in Europe they have the EU Data Protection Directive (1995).

Our goals for the MET project are to satisfy these privacy guidelines using the appropriate technology. We plan to use encryption to secure all data transmission and storage of patient records. We want to enforce read and write access control not only for a medical record but for individual documents within that record. The security measures put in place should include end-to-end encryption of patient records, strong authentication, authorization, data integrity, and audit logs.

Another privacy goal specified by the hospital staff is to make sure none of these records leave the hospital premises. With the paper based system, paper records are not to be removed from the

hospital. Because in our case the tablet PCs may leave the hospital, we don't want any records that are currently on the PC to be able to leave with it. We will investigate the options we decided on for dealing with this privacy issue.

4. ENCRYPTION AND ACCSS CONTROL

Encryption is necessary in our system for protecting sensitive data such as patient records. We want this data to be stored encrypted and transmitted encrypted so that no one sniffing the network can get access to the data (particularly since a portion of the network is wireless). There are many options for securing data using encryption such as network encryption using SSL and data encryption using public key cryptography. We will discuss the options we considered as well as propose a policy-based encryption solution.

4.1 Network Encryption & Access Control

In complying with PIPEDA we want to ensure all transmissions with sensitive data are encrypted. The Wired Equivalent Privacy (WEP) protocol is intended to provide confidentiality on wireless networks however many weaknesses have been found that could lead to a relatively easy attack[4]. It is not considered to be secure against any more than a casual eavesdropper. For our purposes we need more than this in order to comply with our first privacy goal of protecting sensitive data with a high level of security.

A Virtual Private Network (VPN) is another consideration for securing our wireless network. IPSec and SSL can each be used to implement a VPN. IPSec has been notoriously complex to configure and manage and we will not consider it for this implementation[2]. SSL meanwhile has been used as a much easier alternative and has implementations that have proven to be secure.

SSL/TLS uses a handshake protocol to establish shared keys between a client and server. All communications between client and server use the shared key to encrypt data. This creates an 'encrypted channel' between the client and server. SSL provides confidentiality and data integrity through cryptography. The handshake protocol allows for authentication of the server, so the client is assured of a secure connection with the proper server. OpenVPN uses SSL/TLS technology and has been shown to offer the SSL security properties of authentication, confidentiality, and data integrity[10]. Client software for the VPN would be set up on all tablet PCs and server software on MET servers. Users accessing the network would do so by logging into the VPN software.

Encrypting all transmissions on the network will secure data from anyone who is not logged onto the network. It is still necessary to implement an access control system to manage permissions and access to patient data for employees who access the software. Role-Based Access Control (RBAC) allows for controlling which users have access to which data on a network[6]. Users are assigned one or more roles and must authenticate to the system when requesting access to a resource. Each role has permissions assigned to it, dictating which resources are available to that role. We can express read or write access for documents based on the roles of employees using an RBAC system. Policies would need to be created from the employee groupings and rules about which documents they should have access to. Changes in policies would have to be reflected by changing the permissions for the affected roles.

Role Based Access Control will provide protection from unauthorized access. It would also allow us to limit information disclosure to a strictly 'need to know basis'. Coupled with network encryption this would cover most of our privacy goals. However this may be more work than we need to make our system secure as we will discuss next.

4.2 Encryption Only

It may be unnecessary to employ both encryption and access control as separate technologies in our system. We present here options we considered that combine encryption and access control functionality.

Traditional encryption methods such as public key cryptography (PKC) are cumbersome to apply to access control. In a public key system, each user is assigned a public key and private key. A document is usually encrypted for a single recipient using their specific public key. Only the recipient can decrypt to recover the original document by using their private key. Managing keys in this system has often been a limiting factor in real world applications. Keys need to be created and distributed to all users through the use of digital certificates. Users who leave the system or lose their keys need to have their certificate revoked. Certificates and keys have a finite lifetime and need to be renewed. Old keys however still need to be kept in order to be able to decrypt documents that are encrypted under it for as long as the document's lifetime.

For our system, we need a way to encrypt medical records for access by multiple recipients, who are not necessarily known at encryption time. PKI doesn't offer this flexibility on its own; intended recipients are known and their keys are used in the encryption process. If we wanted to use PKI, we would have to add access control methods to allow for managing multiple recipients. We could then use roles to control access control and public keys to provide encryption when transmitting[19]. However this adds unnecessary complexity to the system. It would be easier to encrypt all transmissions on the network and use traditional access control methods. Thus PKI does not offer a feasible solution for this case.

4.3 Identity/Role-Based Encryption

Identity-based encryption (IBE) potentially offers more flexibility for our environment than PKI. The main idea behind identity-based encryption is that any string can be used as an encryption key[17]. For example, a person's unique email address can be used. A document can be encrypted with the recipient's email address. The recipient must identify themselves to a trusted authority to receive the decryption key to recover the original document. This is often how this scheme is described in order to compare it with PKI in sending an encrypted document to a single known recipient. IBE reduces the need for key management, as a user's public key is a well known unique string such as their email address. The private key is obtained by authenticating to a trusted authority (the Private Key Generator, PKG), and so the user doesn't need to keep keys. Managing user accounts becomes easier. When a user leaves the system, they no longer have access

to decrypt because they can't log in to the system to authenticate (e.g., because their password or account has been disabled).

Advantages of this encryption scheme include users not having to manage their own keys and no need for key certificates. From a usability point of view this is ideal. Doctors using our system won't have to worry about details of encryption. When they need to access a document, they will simply identify themselves by logging into the system. However, we will not be encrypting for one identity as mentioned earlier. For our application we need to encrypt for multiple people.

Using a more general approach, rather than encrypting on a unique string such as an email address, we can encrypt for a general grouping such as a role. This allows for multiple people to get access if they belong to that role. But what if we want to control access for multiple roles? For example a doctor and nurse may have access to a patient's record but not administrative personnel. So we want to encrypt for doctors and nurses. Again further generalizing this approach gives us a more flexible solution. Encrypting based on a policy can allow a document to be encrypted for access by multiple roles.

4.4 Policy Based Encryption

A policy-based encryption scheme offers the greatest flexibility for our security needs. The approach is relatively simple and builds on the idea of encrypting under an arbitrary string. A document is encrypted under a policy which is a combination of rules. Note that in IBE, if the public key can be an arbitrary string, then this string need not be an identity. Rather, the string may be a complete access control policy (or a hash of that policy) for the document that is to be encrypted. A user wishing to have access to a document will authenticate by logging into the system and will obtain a decryption key associated with their role from the PKG. If the user's role satisfies the requirements of the policy, their decryption key will decrypt the document.

The policies can be as simple as combining a few roles or more complex to include other rules such as time constraints. In our implementation for the MET project, a small number of simple policies will be created. Groupings are clearly assigned in the workplace already so policies would be able to be created based on this information. Each document type will have an associated access policy. An example policy could be "doctors or nurses can have read access for lab reports".

Corresponding keys for the decryption process will be created based on a user's role. When a user logs in, a trusted authority (TA) will authenticate them and provide the decryption key associated with the user's role. Then, when the user makes requests for records, if the user's role fits the policy their decryption key will decrypt the document and they will have access.

Keeping usability in mind, it would be favorable to automate the encryption process. Staff should not be asked to enter a policy or choose from a list of policies every time they create a new document. Because of the nature of the hospital setting, we can make policies dependant on the type of document. If a document of a certain type is entered in the system, it will be encrypted based on a corresponding policy. This is possible because of the finite number of document types in a hospital setting and the access rules for these document types. For example all x-ray lab reports are available to be read by doctors. Therefore these documents can be encrypted under a policy specifically for that type. This will also ensure that we have a finite number of policies to manage.

If a policy is changed or updated, then all documents encrypted under that policy will also have to be updated. This can be done automatically by decrypting and then encrypting under the new policy. It will not affect staff having access since they will receive the updated key when they try to get access. If they have a document open already, then the update won't affect them. If they modify and save the document, it will be encrypted under the new available policy. In our system, the database will have access to decryption keys as well as policies to encrypt under. This will allow for indexing of records which are stored encrypted under the policies. Keys are not private to individual users but rather keys are distributed based on user permissions. Thus, allowing the database to use these keys is not compromising individual users' private keys.

In traditional PKI we have revocation of keys which allows for users to leave the system and to manage compromised keys. In our system, we would have two scenarios that may require key revocation. If a staff member leaves or their role is changed, and if a key is compromised. For the first scenario, if the user's role is changed this will be reflected their account and they will receive the corresponding decryption key from the PKG. If the staff member no longer works at the hospital, their account will be disabled and they won't be able to log in, and thus won't be able to have access to the system. Therefore, key revocation is not needed for these types of changes to accounts. However we do need revocation in the case of a key compromise.

Boneh and Franklin propose adding a time period to keys in order to handle this situation[5]. Since the policies and keys used in our system are strings, we can also associate them with a time period. Adding a time period to the policy can be done by simply adding another constraint in conjunction with other rules. Decryption keys generated during that time period will contain the proper time to fit the policy. Updating a policy's time period will disable keys which have been acquired and saved from previous decryptions. Users who are using the system properly will be receiving the proper key each time they make a request to have access to a document and so will not need to keep track of updating keys. In the event of key compromise, updating the policy (and updating records in batches) is sufficient to disable the key.

Molva and Bagga propose a policy-based encryption system that looks promising for use in the MET system[3]. In their encryption scheme a document is encrypted under a policy which is a combination of conditions or rules. Let's consider an easy example of a policy that would be used in our system. An employee with the role of doctor or nurse can read a document. This policy *pol* will specify an action *act* of reading a resource *res* which is a document. In Molva and Bagga's encryption system the document would be encrypted under the following policy

pol = <Doctor, role> OR <Nurse, role>

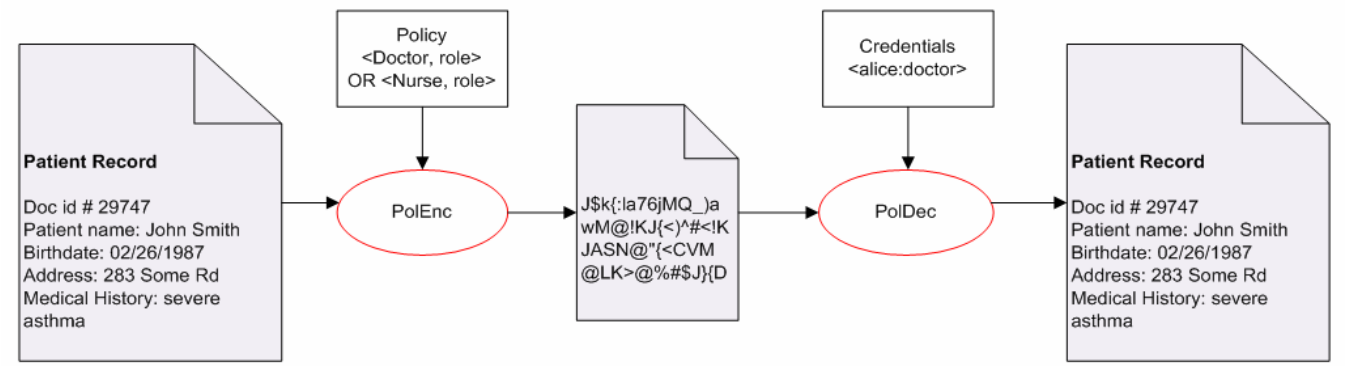


Figure 1. Policy-Based Encryption

The document *res* will be encrypted using policy *pol* as follows

c = PolEnc(res, pol)

A user can decrypt *c* by providing their credentials *cred = (alice:doctor)* that satisfy the policy *pol*

res = PolDec(c, pol, cred)

See Figure 1 for a diagram demonstrating the encryption and decryption process. A brief description of the way their encryption works is that each set of conjunctions is assigned a mask. Each disjunction is assigned a random key value. Then, each key value is encrypted by each mask. A user who satisfies any of the sets of conjunctions will be able to retrieve each key value, and decrypt the entire document.

Their scheme works well for a policy that specifies one action for a specific resource. In our case, we would need for the policy to be applied to more than one action. For example if it is a doctor asking for access to a patient's record then they can read and write, but if it's a nurse then they can only read. Molva and Bagga's scheme allows for policies that are made of combinations of rules, but don't include information about the action allowed on the resource for those who fit the policy.

Therefore, we add expressive functionality to their system by allowing actions to be specified for each conjunction. Since the idea is to be able to encrypt under any arbitrary string, it seems fairly reasonable to say that we can add this information to the policy to encrypt under. A policy could therefore be represented as

pol2 = <Doctor, role> AND <Write, action>

Using default values for actions, all users would be able to perform these actions. However, a user would only be allowed to perform the action if they also satisfy the other rules in the conjunction. In the example of pol2, upon presenting a credential for the role of a doctor, a user could decrypt and have write access. The action rule will signify what permissions the user has for that particular document. This will result in treating keys for each credential in a different way, allowing default key values for the actions.

The efficiency of the Molva and Bagga encryption system depends on the number of conditions that are combined to form the policy. The complexity increases with the number of computations required to encrypt for each rule. For our application, we plan to have few rules combined in a very simple manner. We will also have limited number of credentials for users, as their roles will be the credentials and we have a limited number of actions possible. They also state ways of maximizing the efficiency of encrypting by pre-computing and caching some values. Decryption is more efficient than encryption in their scheme and would cut down on the computation time in the overall system.

By extending their scheme to allow for more expressive policies, we can adapt this system for use in our MET project. Using simple policies that specify actions allowed on the document encrypted under that policy we strive to keep the encryption process efficient. We also have minimal number of credentials for users based on their roles in the hospital. Finally, we can automate the encryption process by specifying policies for each document type, thus allowing staff members to enter documents in the system without worrying about encryption details.

4.5 Network Encryption & Access Control vs. Policy-Based Encryption

4.5.1 Access Control with Network Encryption

Access control and network encryption using a VPN plus a role-based access control mechanism would allow us to store the records in readable format. This has the advantage that for storing over long term, records won't be made obsolete from out of date encryption methods. However, this allows anyone who gets physical access to the servers storing the records access to the records in readable format. If they were encrypted then the attacker would have no advantage by getting the encrypted format records. Encrypting the database separately requires an added step, and each document would require decryption and re-encryption under SSL for transmission on the network. The policy-based encryption would be a simpler solution for providing complete encryption and access control at the same time.

4.5.2 Policy-Based Encryption

The policy-based encryption method has the advantage that encryption and access control are in one package. Not only would it provide for encryption during transmission but also for storage. Records will be transmitted to the devices encrypted, and be decrypted once on the PC. This ensures all sensitive information being transmitted is secure. The encryption scheme also allows us to implement access control as only those who satisfy the policy can decrypt and have access to the record. There is no need for users to manage their own keys nor is there need to distribute

keys. There will be no difference for the user experience between the two choices. In either case, the user will log in and will have access to documents for which they have permission.

A general disadvantage of any encryption scheme is the fact that the records are stored encrypted. The decryption keys must be kept for the lifetime of the record. The decryption algorithm must also be available to recover the original documents. In order to ensure these are both available, it may be necessary to keep a backup of the keys and algorithm. Or it would be possible to store the records in readable format somewhere physically secure and not connected to a network. We would need to store backup copies of the records in plaintext regardless of which system we use, due to possibilities of hardware failure.

Overall, the policy-based encryption method provides the most compliance with PIPEDA regulations with the least amount of complexity. We get a high level of security for the sensitive data. The data is protected by encryption while stored and transmitted protecting it from malicious people who may be eavesdropping. The records are also protected from unauthorized access as only users of the system with the proper account permissions set up are given a decryption key. We have limited the access on a 'need to know' basis by forming policies constraining access to documents based on roles. The users of the system will not have to know about the security technologies used and their only security task will be to log in as they already do on other hospital computer systems.

5. AUTHENTICATION

The security of the encryption system relies on strong authentication. The system is only as secure as its ability to prevent unauthorized access. Currently in the hospital, the standard username and password method is used without any restrictions on password choice. We wanted to explore authentication options that would be both secure and usable so as not to change the user experience too much.

The username and password combination is the most popular authentication method. There are many reasons why this is not a secure method[1]. Users will choose very easy to remember passwords that are also easy to break. If we imposed restrictions on the password choice to force users to have stronger passwords, they would do what they could to make logging in easier, by writing them down for instance. We have already had feedback from doctors who would not be happy having password restrictions being put in place and encouraged us to look at other options. Passwords are subject to social engineering attacks which could be easy to manipulate in an emergency setting. If a malicious person claims there's an emergency and asks for a staff member's password, they may be more likely to divulge information.

A two-factor authentication would be desirable to provide improved security. Our motivation for finding a proper two-factor authentication mechanism lies in working with what we have. One factor will be the username and password system without restrictions on the password. The second factor could be a biometric fingerprint or RFID tag. The tablet PC is equipped with both a fingerprint reader and RFID reader. We discuss both and what we chose for the second factor in our authentication scheme.

5.1 Fingerprint Biometrics

One of the most common biometrics in use for authentication is the fingerprint. Using the fingerprint readers has an advantage that users don't need to 'remember' to bring a token with them to work. However some usability problems may make this less desirable to use. In order to provide your fingerprint for verification, a user must place their finger in a certain position. Factors such as placement, heat, cold, and perspiration can all affect how accurate the system is[11]. We want our users to be able to log in every time because the setting is the emergency room of a real hospital.

Another factor with fingerprint biometrics is that not everyone can give the fingerprint to enroll. Fingerprints damaged by injuries could be a problem. We want to make sure everyone can use the system, including current employees and future employees. In the healthcare setting many employees may be wearing hygienic gloves which would not allow them to use the fingerprint reader without first removing them. This is a real usability problem in our setting. When discussing the fingerprint option with doctors who will be using the system, they seemed reluctant to use fingerprints in the authentication stage and wanted something easier.

5.2 RFID Reader

The RFID reader offers an alternative to the fingerprint reader. Doctors carry employee badges which can be equipped with a barcode. To sign in a doctor swipes their card (or, if it is a proximity reader, simply has their badge somewhere on their person) and provides their login password. Everyone can be given a badge and it will always be accepted. They don't have to have a high entropy password which may be hard to remember. They may be less likely to simply tell someone their password to let them have access, since they would also have to provide their card. However this option is still available if the doctor wishes to delegate his tasks to another employee for a period of time. This provides for a flexible authentication system compared to using a fingerprint which cannot be delegated. We decided to go ahead with using the doctor's badges and the RFID reader as the second factor in our authentication system.

Unlike the fingerprint method, a doctor can forget their employee badge. If they borrow someone else's pass or get a temporary one, this will not work with the reader since it will be a different pass. We could manage to have temporary badges available for the day which can be associated with their account. One way around this is to use a common backup method of asking the user a series of pre-answered questions. If the user answers correctly and provides their usual login information along with it, then the user can log in for the day. Proper tracking of these events is important for being able to notice and document malicious behavior.

6. OTHER PRIVACY CONCERNS

There are other security and privacy issues that we wanted to address for our MET project. The specific security needs arise because of the environment and physical location of the project. Some security measures we will include in our system are audit logs, integrity mechanisms, and automatic purging of sensitive files.

6.1 Audit Logs

An important privacy requirement and a feature our system must ultimately include is an audit log. Audit logs allow for tracking user's activity on the system. If by any chance someone is misusing the system, then a record of that activity must be available. For example if a doctor is accessing large amounts of patient records that clearly aren't all their patients, this would be worth investigating. It could be that a malicious user has gotten access to that account and is stealing patient information. It would be a good addition to our system to add logging functionality of all access and changes to patient records. Specifically we want to track when a user logs on and off, which records they request, and which records they make changes to.

6.2 Integrity Mechanisms

Another privacy aspect is not only protecting the data from unauthorized access but also from unauthorized modification. Ensuring proper access control for read/write permissions is essential. However we also want to include some sort of integrity mechanism to be able to check that no changes have been done maliciously. For example if someone changes a record stored in the database we need to be able to check this.

A Message Authentication Code (MAC) applied to a record would provide us with a way of checking for any changes made[8].We could do this on an individual record basis, creating a MAC of the entire record, storing the MAC value elsewhere on the system, and comparing the stored MAC with a freshly-computed MAC for every record read event.

6.3 Purging Files

Consider the scenario where a doctor has their tablet PC which they bring home everyday. At the end of their shift, they may have seen a number of patients and have their data saved on the PC. The doctor goes home and someone in their household uses the PC for other purposes. It could be connected to the Internet at this point and anyone with access to the PC could have access to the sensitive records.

We need a system of preventing records from leaving the hospital which is facilitated by the use of the tablet PCs. A doctor may not notice or remember that he still has files on his PC when leaving the hospital. It's much more obvious with paper-based records, where a doctor has to actually carry them out or knowingly put them in a bag. If it becomes a habit of taking the tablet PC home, then the doctor may not remember to erase patient data each time. This means we need to know when a tablet PC is being taken out of the hospital so that we can erase those files that haven't been deleted.

One simple thing to do is to purge all files on shutdown. However sometimes PCs are not shut down properly especially in the case of a tablet PC which may lose battery power. Therefore it would be better to implement on system startup. Each time the laptop is booted, the files are purged. Since doctors wouldn't be turning it off and on all day, they shouldn't lose any records until they are done their work. We don't need to implement this on hibernate states, so that if a doctor leaves his PC for a while with no activity then he'll still have the information he needs on it.
If a tablet PC leaves the hospital while still turned on, we still need to purge the files. In this case we can't rely simply on the files being purged when the PC is turned off or rebooted. The connection to the wireless network can be used as an event trigger to purge all files. When the tablet PC is taken out of range and no longer has a connection with the network, the patient data files will be erased. A doctor who is in the wireless network area doing his work will not lose any data. It would only affect tablet PCs that are taken far enough away from the hospital department with the network connection available.

7. SYSTEM ARCHITECTURE

The tablet PCs being used in the trial are the Motion Computing C5 models[13]. The PCs are equipped with a fingerprint reader and RFID scanner. The devices will ensure that doctors are able to travel patient to patient while having access to the information they need on the wireless network.

The MET software and privacy functionality will be implemented in Java using the JADE development environment. The system is agent-based with multiple agents each assigned a specific task in the software. A central temporary storage area, called a blackboard, will be used to store patient records and medical information that agents are using. Adding an encryption agent seemed reasonable but has not become a practical solution. All agents will need access to encryption services for pulling and pushing information to and from the databases, the blackboard, and the tablet PC. Multiple encryption agents would be needed for the servers, the tablet PC, and the databases. Integrating the security functions as a layer, or set of services, available to all agents in the system proved a better solution. The MET architecture is reflected in figure 2 below. The security and encryption services layer will provide encryption, decryption, authentication, and account management services. In addition, a monitoring agent on the tablet PC will be necessary to track when to purge the files in the event of a disconnection from the network or system reboot.

8. RELATED WORK

8.1 Encryption and Access Control

Role-based access control systems using public key cryptography have been proposed. Wilkinson, Hearn, and Wiseman describe an access control system that uses encryption to control access to documents[19]. Documents are encrypted under a group's public key, and members of that group can decrypt with the group public key. They also describe how symmetric keys could be used as the group keys, however they conclude that asymmetric cryptography will provide better protection from key compromise at proxies. While the scheme does provide security measures that we are looking for, it still suffers from the key management issues of public key cryptography schemes. It also doesn't seem straightforward how to encrypt a document for multiple groups or multiple roles.

Kapadia, Tsang and Smith propose an attribute-based encryption system that allows for role based access control[9]. Their system relies on hidden credentials and policies. In our case, the policies will most likely be public since there are a finite set of policies created based on well-known rules. It would be better to have a system that doesn't rely on having secret policies. Other attribute-based encryption schemes aren't as efficient as the Molva and Bagga scheme discussed next[7].

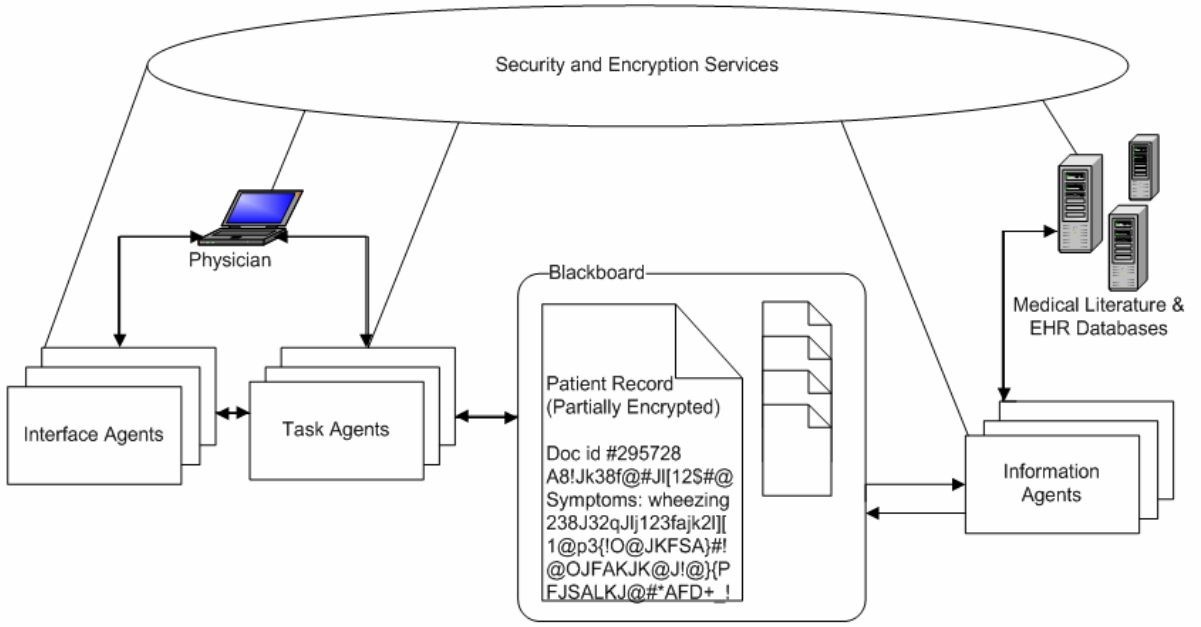


Figure 2. MET Architecture

Most policy-based encryption schemes that have been proposed are based on the Boneh-Franklin ID-based encryption scheme using bilinear pairings over elliptic curves[5]. Smart proposed a scheme extending this for encrypting on multiple identities[18]. Molva and Bagga further extend their work to propose a policy-based cryptography scheme including an encryption scheme and signature scheme[3]. They propose using a policy as a public key to encrypt a document. A user obtains their decryption key based on their credentials and can decrypt if these credentials satisfy the policy rules. As mentioned earlier their scheme is a good basis for us to extend our work on.

8.2 Examples of Privacy Technologies Used in Health Care Environments

Voltage is great example of a company using the technologies discussed here for security and privacy solutions in health care environments. They offer an identity-based encryption for email messaging that is currently being used in hospitals in the United States[20]. Similarly, Secure Computing offers policy-based cryptography products. They implemented a token based authentication system with audit logs to ensure HIPAA compliance for a system in a hospital[16].

Mont, Bramhall, and Harisson from the Hewlett Packard Lab in Bristol, UK have developed a messaging service using identity-based cryptography for a hospital[12]. Their scheme uses the fact that any string can be used to encrypt on including a role. When a user wants to send a message they choose a role to encrypt it under, and recipients of the message can decrypt if they belong to that role. Anyone who doesn't belong to the role cannot see the message. This provides a secure email system for use within the hospital.

9. CONCLUSION & FUTURE WORK

We presented alternative security and privacy technologies considered for use in our system architecture for an e-hospital environment. The motivation for the inclusion of high security technologies comes from the requirements by privacy legislations PIPEDA and PHIPA. Our system therefore includes encryption for data confidentiality, integrity mechanisms, authentication, authorization, and audit logs. An additional security measure put in place for our project also involves automatic deletion of sensitive data when tablet PCs are taken out of the hospital area.

Our main contribution is the use of a policy-based encryption scheme in providing data encryption and access control. We propose extending Molva and Bagga's work to suit a health care environment for access control with a patient's records database. Policy-based cryptography looks promising for use in different settings. The uses of this type of system could span many environments including corporate settings and email systems. The usefulness in creating keys based on roles and the simplicity of key management give policy-based encryption many advantages over current encryption schemes.

10. REFERENCES

[1] A Adams, M Sasse, "Users are not the enemy", In Communications of the ACM, pp 40-46, 1999

[2] Array Networks Inc. SSL VPN vs IPSec VPN, Jan. 2003. white paper.

[3] W Bagga and R Molva, "Policy-Based Cryptography and Applications", In Lecture Notes in Computer Science, pp. 72-87, Springer Berlin / Heidelberg, 2005.

[4] A Bittau, M Handley, J Lackey, "The Final Nail in WEP's Coffin", The 2006 IEEE Symposium on Security and Privacy SP, pp. 386-400, 2006.

[5] D Boneh and M Franklin, “Identity-Based Encryption from the Weil Pairing”, In Proceedings of CRYPTO 2001, pages 213-229, Springer-Verlag, 2001.

[6] D Ferraiolo, J Cugini, R Kuhn, “Role-based access control (RBAC): Features and motivations”, In Proceedings of the 11th Annual Conference on Computer Security Applications, pp 241–248, 1995.

[7] J Holt, R Bradshaw, KE Seamons, and H Orman, “Hidden credentials”, In Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society, ACM Press, 2003.

[8] RR Jueneman, SM Matyas, CH Meyer, “Message Authentication”, IEEE Communications Magazine, pp 29-40, 1985.

[9] A Kapadia, P Tsang, SW Smith. "Attribute-Based Publishing with Hidden Credentials and Hidden Policies." In 14th Annual Network and Distributed System Security Symposium (NDSS '07), pp. 179-192, 2007.

[10] S Khanvilkar, A Khokhar, “Virtual Private Networks: An Overview with Performance Evaluation”, IEEE Communications Magazine, pp 146-154, October 2004.

[11] G Lassmann, “Some results on robustness, security and usability of biometric systems”, In Proceedings of the 2002 IEEE International Conference on Multimedia and Expo ICME ’02, pp 577-579, 2002

[12] MC Mont, P Bramhall, CR Dalton, and K Harrison, “A Flexible Role-based Secure Messaging Service: Exploiting IBE Technology in a Health Care Trial”, Hewlett-Packard Laboratories, technical report HPL-2003-21, 2003.

[13] Motion Computing, “Motion C5”, 2007, http://www.motioncomputing.com/

[14] Personal Health Information Protection Act (PHIPA 2004) http://www.health.gov.on.ca/english/public/legislation/bill_31/personal_info.html

[15] Personal Information Protection and Electronic Documents Act (PIPEDA 2000) http://www.privcom.gc.ca/legislation/02_06_01_01_e.asp

[16] Secure Computing, 2007, http://www.securecomputing.com/

[17] Shamir, “Identity-based cryptosystems and signature schemes”, In Proceedings of CRYPTO 84 on Advances in cryptology, pp. 47-53. Springer-Verlag New York, Inc., 1985.

[18] N Smart. “Access control using pairing based cryptography”, In Proceedings CT-RSA 2003, pp 111–121. Springer-Verlag LNCS 2612, April 2003.

[19] T Wilkinson, D Hearn, and S Wiseman, “Trustworthy access control with untrustworthy web servers”, In Proceedings of the 15th Annual Computer Security Applications Conference, pp 12. IEEE Computer Society, 1999.

[20] Voltage, http://www.voltage.com

Security and Privacy System Architecture for an e-Hospital Environment

Kathryn Garson, Carlisle Adams
University of Ottawa

Agenda

- Introduction
- Hospital Environment
- Privacy Goals
- Encryption and Access Control
- Authentication
- Other Privacy Concerns
- System Architecture
- Conclusion

Introduction

- Mobile Emergency Triage (MET) project
  - Software provides doctors with interactive decision support for triage and treatment of patients
  - To be used as a trial in Ottawa hospital
  - Wireless network allowing physicians to travel patient to patient with tablet PC
  - MET software pulls patient records from database
  - Sensitive data transmitted over wireless network, stored on servers and tablet PC
- Need access control and encryption to protect sensitive information

Hospital Environment

- Highly sensitive data required in emergency situations
  - Cannot have security measures get in the way of physician's access to information
- Patient records currently paper moving towards electronic
  - Each record consists of multiple documents
  - All will be available in electronic form

Hospital Environment

- Current system in emergency department is read-only
  - Employees sign in with username and password (no restrictions)
  - High-traffic areas
  - Access depends on employee role and associated permissions
  - Only access from within hospital, remote log-in will not be a part of our trial

Privacy Goals

- Personal Information Protection and Electronic Documents Act (PIPEDA – Canada)
- Personal Health Information Protection Act (PHIPA – Ontario)
  - Extends PIPEDA
- Principle 7 of PIPEDA specifies safeguards to be used to protect sensitive data

Privacy Goals

- Sensitive information should be protected by high level of security
- Include technological methods such as passwords and encryption
- Limit access on a ‘need to know’ basis
- Medical records should be protected from loss, theft, unauthorized access, disclosure, copying, use, or modification

Privacy Goals

- United States
  - Title II of Health Insurance Portability and Accountability Act (HIPAA) has similar privacy standards
- Europe
  - EU Data Protection Directive
- Our goal is to satisfy these privacy guidelines
  - Provide encryption and access control
  - Use appropriate authentication method
  - Additional privacy need: automatic deletion of records

Encryption and Access Control

- Network Encryption and Access Control
- Encryption Only
- Identity-Based Encryption
- Role-based Encryption
- Policy-Based Encryption

Network Encryption and Access Control

- Wired Equivalency Privacy (WEP) Protocol not considered secure
- Virtual Private Network (VPN) using SSL
  - All communications between client and server are encrypted using a shared key
  - OpenVPN uses SSL technology and provides authentication, confidentiality, and data integrity
  - Physician's device and MET servers will have VPN software

Network Encryption and Access Control

- Still need access control to manage access within network
- Role-Based Access Control (RBAC) systems work well in hospital environment
  - Permissions already based on employee role
  - Authenticate to system when asking for resources
- RBAC and VPN will provide security we need

Encryption Only

- May be unnecessary to separate access control and encryption
- Traditional Public Key Cryptography
  - Each user has a private and public key
  - Keys managed by use of certificates
  - Encryptions are done for a particular user
  - Problems: cumbersome management of keys, cannot encrypt easily for multiple recipients
  - Would need to add access control methods to organize encryptions based on roles

Identity-Based Encryption

- In Identity-based encryption, any string can be used as the public key
  - E.g. email address
  - Recipient authenticates to a trusted authority and receives the corresponding private key
- Users don't need to manage their own keys
  - Public key is well-known and unique to that person, no certificate necessary
  - Private key is generated when user authenticates themselves
- However, we still need to figure out how to encrypt for multiple recipients based on role

Role-Based Encryption

- Using a more general approach, encryption can be done on a role
  - Since any arbitrary string can be used as public key, can use a string such as "doctor"
  - Allows to encrypt a document with access control rules
- However, we need to be able to express more complex rules
  - E.g. "a doctor or nurse can have read access"

Policy-Based Encryption

- Document is encrypted under a policy, which is a combination of rules
  - Can be simple policies such as
    "<Doctor AND Write> OR <Nurse AND Read>
- User wishing to decrypt
  - Authenticates to a Trusted Authority
  - Receives a private key based on their role containing credentials
  - If their credentials satisfy policy rules, their key can decrypt document

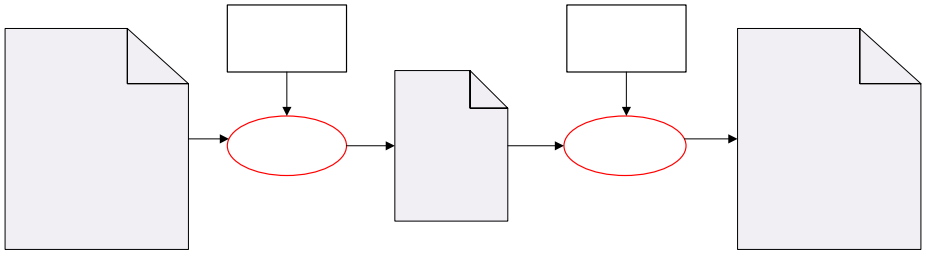
Policy-Based Encryption

- Automated encryption process
  - Physicians and staff will not have to manage private keys
  - Authentication process allows to decrypt
  - Policies can be specified by document type
  - Staff entering/saving documents will not have to select which policy to encrypt under

Policy-Based Encryption

- Encryption system by Molva & Bagga
- Consider a policy stating "an employee with the role of doctor or nurse can read a document"
  pol = <Doctor, role> OR <Nurse, role>
  act = read
  doc = document
- The document is encrypted
  temp = PolEnc(doc, pol)
- A person can decrypt if their credentials satisfy the policy
  cred = (alice:doctor)
  doc = PolDec(temp, pol, cred)

Policy-Based Encryption



Policy-Based Encryption

- Encryption in their system
  - Each conjunction is assigned a mask
  - Each disjunction is assigned a key
  - Each key is encrypted by each mask
  - User who satisfies one conjunction will be able to decrypt
- Our goal is to extend their work
  - Want to include multiple actions
  - Want to add key revocation

Policy-Based Encryption

- Multiple actions
  - What if a doctor is allowed to modify a document, while a nurse can only read
  - Keep track of which conjunction in policy was satisfied
    - E.g. <Doctor, role> AND <Write, action>
  - Trap call to OS, if write action requested, and conjunction with was satisfied, open document in write mode
- Policies can contain action elements
  - Corresponding keys will contain requested action as well

Policy-Based Encryption

- Key revocation
  - What if an ex-employee kept the decryption key, need to change key in current use
  - Add a time period to policy
  - Can only decrypt if decryption key was generated/retrieved during that time period
- Example

  pol = <Doctor, role>AND<01/25/2008-02/25/2008, time>

Policy-Based Encryption

- Updates to policies and keys should not interrupt system usage
  - Can update documents in batches
  - Any document currently opened will be saved encrypted under new policy
  - Any person wishing to decrypt a document under new policy will be issued a corresponding updated key
  - Anyone who saved old keys will not be able to decrypt new documents

Policy-Based Encryption

- Provides both encryption and access control
  - Documents are encrypted for storage and all transmissions
  - Policy determines who can decrypt to have access
- Policy can be based on document type
  - Allows for small number of policies to manage
  - Allows for encryption to be done automatically
- Small number of decryption keys based on roles

Authentication

- Security of encryption system relies on strong authentication
- Username and password most common
  - Users choose weak passwords
  - Imposing restrictions makes it hard to remember
- Two-factor authentication
  - Provides better security
  - Staff won't have to remember hard passwords

Authentication

- Fingerprint Biometrics
  - Tablet PC to be used in trial has fingerprint reader
  - Some problems, including doctors wearing gloves
  - After discussing with doctors, agreed this was not an option
- RFID Reader
  - Employee badges have a barcode
  - Can scan badge barcode
  - Second-factor of ‘something you have’
  - Need to include a backup method for when someone forgets their badge for the day

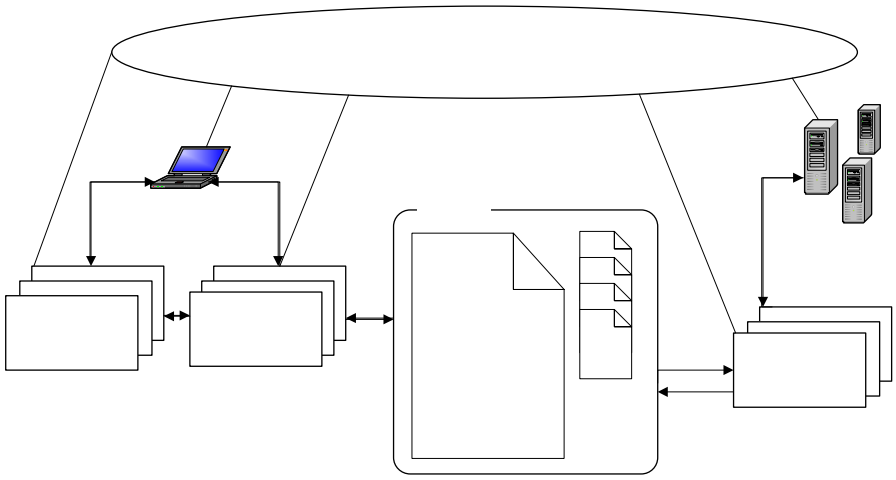
Other Privacy Concerns

- Audit logs
  - Track user activity on system
  - Log on/off, records requested, records modified
- Integrity mechanisms
  - Protect data from unauthorized modification
  - Message Authentication Code (MAC) to be able to verify integrity of records
- Automatic deletion of files
  - Preventing records from leaving hospital
  - Security measure asked for by hospital staff

System Architecture

- MET system is multi-agent
  - Information agents, task agents, interface agents
- Blackboard central temporary storage area
  - Agents pull/push information from/to the blackboard
- Integrating security functionality
  - As encryption agents
    - Problem in organizing agents in this way, encryption and decryption agents needed at blackboard, databases, and tablet PC, will be accessed by all agents
  - Security as a set of services
    - All agents have access to set of encryption services

System Architecture



Conclusion

- Policy-based encryption
  - Extensions to existing system
- Authentication mechanisms
  - Employee badge with a barcode
- Additional security needs
  - Audit logs, integrity mechanisms, deletion of files
- Integrating security into multi-agent system

Questions

Systems Engineering View of Privacy

David Weitzel, M.S., J.D.

MITRE

Statutory Foundations

- Privacy Act of 1974
  - System of Records Notices (SoRN)
- E-Government Act of 2002
  - Privacy Impact Assessments (PIAs)
- Freedom of Information Act (FOIA)
  - FOIA requests
- Homeland Security Act
  - Statutorily Mandated Privacy Officer
- Paperwork Reduction Act
  - OMB Form Control Numbers
- Federal Information Security Management Act
  - FISMA Certification & Accreditation & POAMs
- National Archives, 44 USC 21 et. seq.
  - Records retention & disposal

Fair Information Practices

- Collection limitation
- Notice
- Choice
- Data quality
- Finality / use limitation
- Security
- Accountability

Privacy is a Systems Problem

Intersection of Privacy and Technology

- Assured deletion—especially across the enterprise—of personal information

- RFID and biometrics

Collection

Processing

- Integrity and hygiene of consolidated and aggregated personal information
- De-identification of personal information
- Data flow mapping

Use

Retention

- Detection and prevention of attempts at incompatible secondary use

Destruction

Disclosure

- Security controls to enable appropriate and prevent inappropriate cross-boundary sharing of personal information

- Privacy enterprise architecture
- Privacy system requirements for system development life cycle (SDLC)
- Tools to support analytical aspects of Privacy Impact Assessment
- Privacy risk modeling
- Privacy policy automation and enforcement

Build It In – Create Virtuous Cycles

- Architecture IS policy
  - Lessig – *Code*
- Feedback and control via budget & governance processes
  - OMB 300s
  - FISMA inventory, C&A, POAMs
  - PIAs & SORNs
  - OMB Form Control Numbers
  - NARA Records Retention Schedules
- Unless privacy, information security, and other policy control points are built into the architecture of systems, the chance for appropriate control points to be added later, is, harder, more costly, less effective

A Privacy Approach

Privacy Systems Engineering

- **A repeatable, scalable systems engineering-based approach to uncovering, understanding, and addressing privacy issues**
  - **Explicitly considers multi-dimensional context as well as technology**
  - **Uses risk management to minimize unintended consequences**
  - **Aligns privacy solutions with mission requirements**

Privacy Systems Engineering (1 of 2)

Analyzing Privacy in a System Context

Broadening Context

Privacy Risk Assessment

- Relatively narrow scope, i.e., a specific technology, issue, practice, or policy
- Can be used to evaluate privacy enhancing technologies (PETs)
- Underlying risk model well-defined: potential violations of privacy principles or mandates
- Development and use of technology testbeds when appropriate
- Can be used as formal input to Privacy Impact Assessment

Privacy Impact Assessment

- Moderate scope, i.e., a system or business process
- Design alternatives explicitly considered within the context of the system or business process
- Expanded risk model
  - Risks related to information life cycle
  - Systematic analysis of data flows
- Can be used as formal input to Privacy-Based Systems Analysis

Privacy-Based Systems Analysis

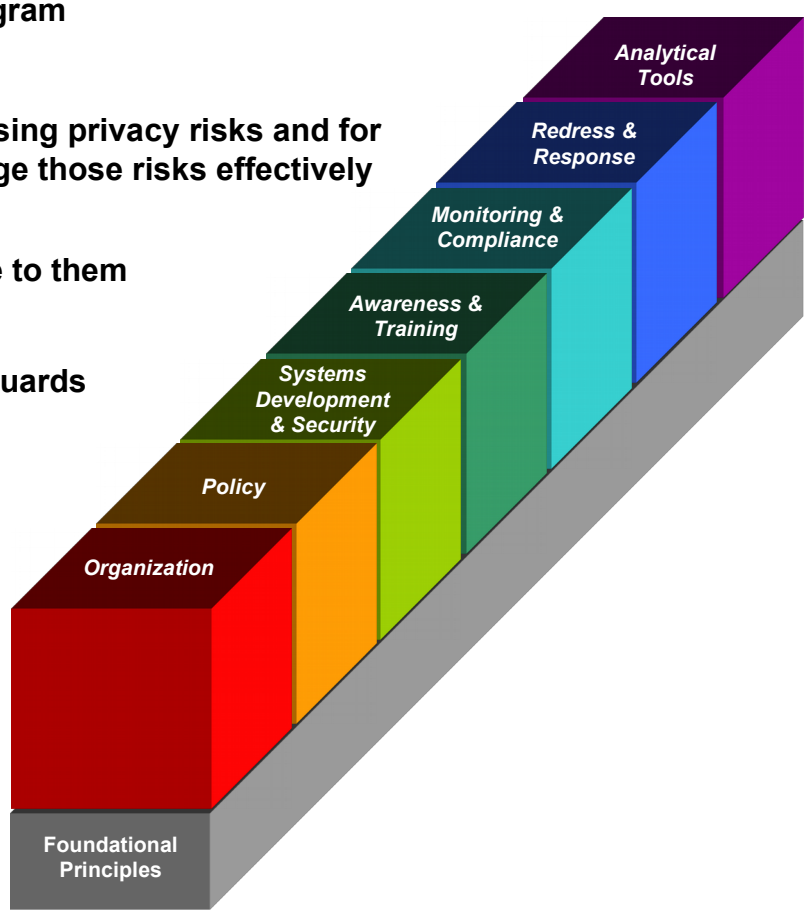
- Broad scope, i.e., mission or program
- Focuses on interaction and integration of technology, processes, and people
- Surfaces unintentional consequences of adopting particular approaches and solutions
- System alternatives explicitly considered and evaluated within the context of the mission or program
  - Includes high-level policy and social dimensions
  - Includes potential application of PETs
- Expansive risk model, less well-defined

Privacy Systems Engineering (2 of 2)

Developing Privacy in a System Context:

Model Privacy Program

Foundational Privacy Principles	Fundamental tenets that guide a privacy program
Organization	People and processes responsible for assessing privacy risks and for developing and implementing plans to manage those risks effectively
Policy	Privacy rules of behavior and ways to adhere to them
Systems Development & Security	Administrative, physical, and technical safeguards that control privacy risks
Awareness & Training	Programs to make the organization, its vendors, and the public aware of the organization's privacy posture and practices
Monitoring & Compliance	Programs to monitor adherence to privacy rules of behavior
Redress & Response	Systems and processes to respond if needed to privacy issues and incidents
Analytical Tools	Tools to support privacy risk management



I3P

Institute for Information Infrastructure Protection

Safeguarding Digital Identity

Bruce J. Bakis, The MITRE Corporation

http://www.thei3p.org

Supported by DHS and NIST

MITRE

Background and Context

- MITRE is leading a 2-year privacy-preserving Identity Management (IdM) research activity: Cornell, Georgia Tech, Purdue, SRI International, University of Illinois
- Supported by the National Institute of Standards and Technology and the Department of Homeland Security
- Managed by Dartmouth College's I3P
  - www.thei3p.org
  - IdM: http://www.thei3p.org/projects/idmgmtoverview.html

Contacts

- The MITRE Corporation
  - Bruce Bakis, 781.271.6915, bbakis@mitre.org
  - www.mitre.org
- The I3P
  - Eric Goetz, 914.954.2466, eric.d.goetz@dartmouth.edu
  - Charles C. Palmer, 914.954.2466, charles.c.palmer@dartmouth.edu
  - www.thei3p.org

What's new with XML Signature

Frederick Hirsch

4 March 2008

NOKIA
Connecting People

XML Signature

- W3C Recommendation February 2002
  - Enables representation of Signature and meta data in XML
  - Designed to enable flexible signing of XML content
  - May also sign binary and non-XML content
  - Flexible
    - Signatures over content in same XML document, or other content
    - Inclusion of signature within XML content or separate
    - Transforms of content before hashing to sign
    - Choice of KeyInfo mechanisms, choice of algorithms
    - Signature properties

Canonicalization 1.1

- XML Canonicalization 1.0
  - W3C Recommendation 15 March 2001
  - Required algorithm for XML Signature inclusive canonicalization
- Exclusive XML Canonicalization
  - W3C Recommendation July 2002
  - Support canonicalization of portions for XML document, excluding inheritance of attributes from ancestors XML elements not in the subset.
- XML Canonicalization 1.1
  - W3C Proposed Recommendation, January 2008
  - Update to enable use of additional attributes in XML namespace with different inheritance properties, including xml:id and xml:base
  - Some additional clarifications

XML Signature, 2nd Edition

1. Require support for Canonicalization 1.1 algorithm, recommend its use for inclusive canonicalization.
2. Incorporate document errata
3. Provide clarifications, but no conformance affecting changes (other than #1)

XML Signature, 2nd Edition Status

- In process to become a W3C recommendation.
- Will also undergo IETF review in order to produce update to RFC 3275.
- Working group has produced a draft intended to become a W3C Proposed Edited Recommendation

XML Security Specifications Maintenance WG

- Chartered in 2007, operating in early 2008.
  - Chair - Frederick Hirsch.
  - W3C Team - Thomas Roessler
- Producing XML Signature, 2nd Edition
- Interop tested C14N11 and XML Signature
- Held public workshop regarding future directions.
  - http://www.w3.org/2007/xmlsec/ws/report.html
- Provided input to W3C team for charter for possible subsequent working group.

Possible Future Work

- Requirements for XML Signature canonicalization, reference and transform processing, algorithms, performance and XML environment.
- Specifications for Canonicalization and XML Signature.
- Algorithms for XML Encryption
- Maintenance of some other XML Security specifications.

Wireless Access using an Identity Provider

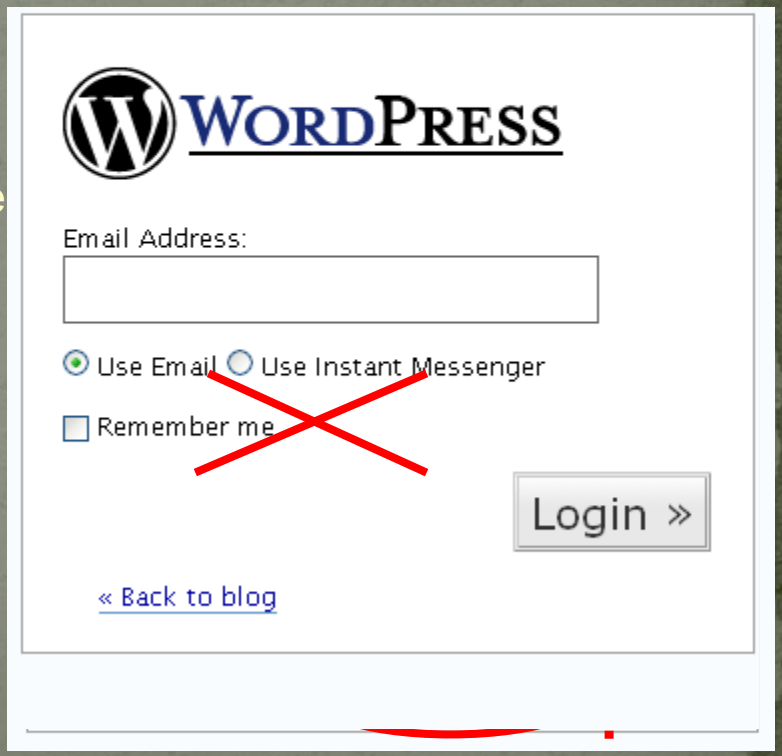
Tim van der Horst and Kent Seamons
Internet Security Research Lab
Brigham Young University

ISRL
Internet Security Research Lab
http://isrl.cs.byu.edu

BYU
BRIGHAM YOUNG UNIVERSITY

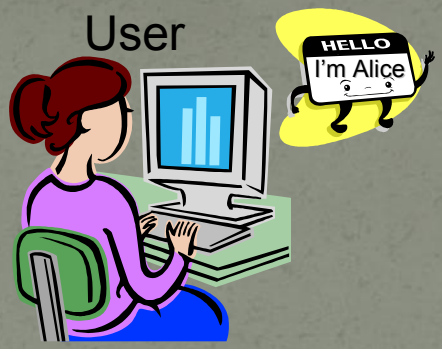
Background

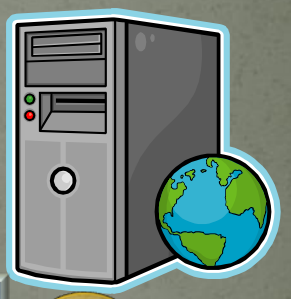
- Problem
  - Users have too many passwords
  - Difficult to share data with outsiders
- Project Goals – Convenience and Security
  - Remove the need for site specific passwords
  - Simple for users to understand and use
    - Existing identifiers (email, IM, text messaging)
  - Easy for administrators to deploy and manage
- Our Approach
  - Leverage forgotten password approach
  - Make it easier and more secure
  - SAW (Simple Authentication for the Web)



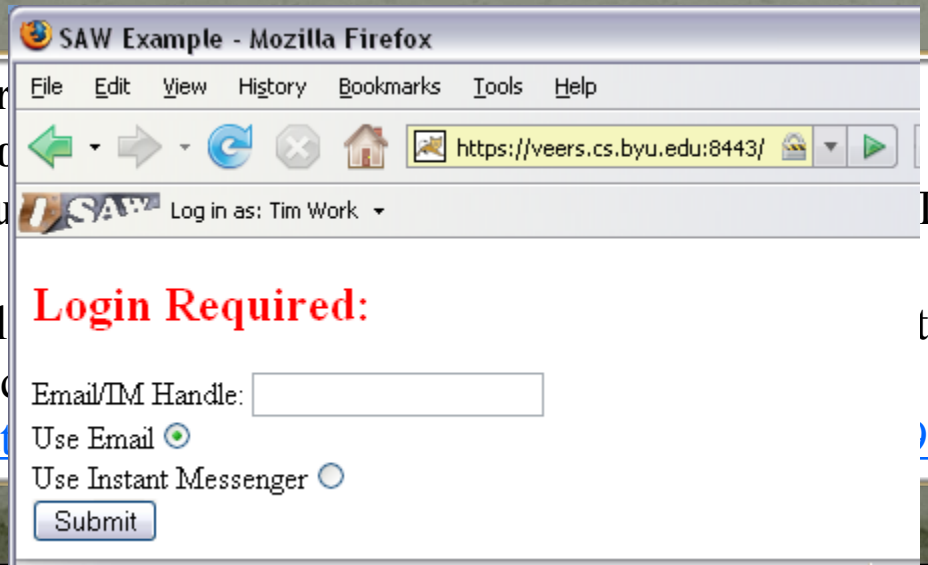
How SAW Works

- Step 1:
  - The user submits her email address
- Step 2:
  - If her address is authorized, a random secret is generated and split into two shares
- Step 3:
  - The u
    - Ma
      By
    - Au
      Us



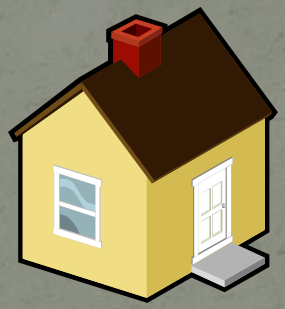


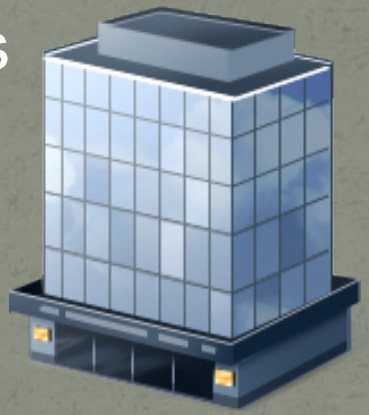
Tokens are:
- Short-lived
-use



Motivation

- It is difficult to provision guest access to access-restricted wireless networks





- Current wireless authentication schemes
  - Global passphrases
  - Username/password
  - User-specific digital certificates

1. Too inflexible
2. Too heavyweight

Wireless Authentication using Remote Passwords (WARP)

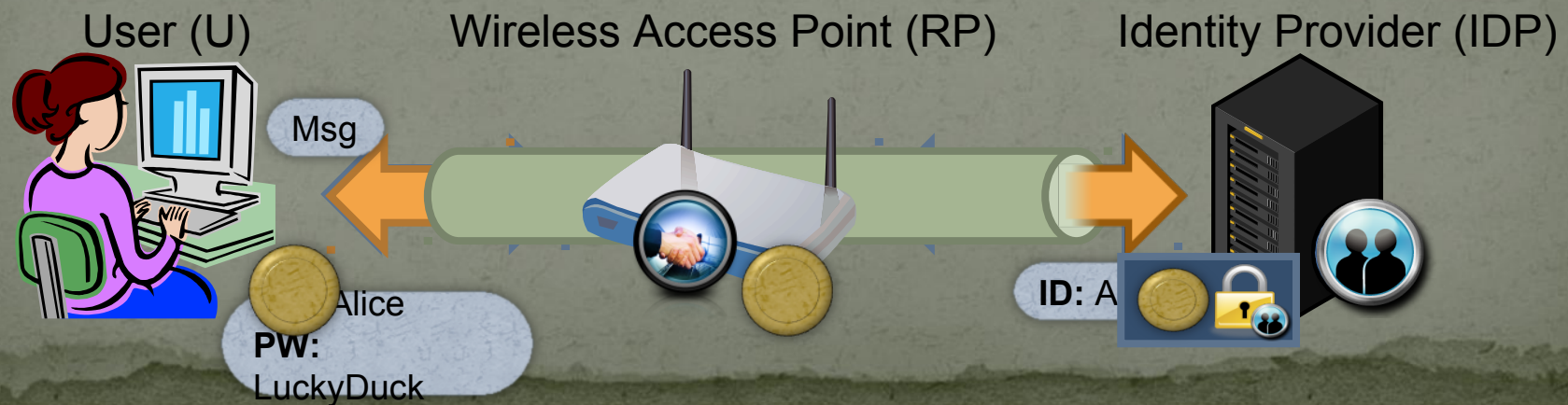
- Goals
  - Bring lightweight decentralized authentication to the wireless realm
    - Users can authenticate to relying parties with whom they have no pre-established relationship
  - Be highly portable
    - Users authenticate via passwords not cryptographic keys
  - Provide strong protections to user login credentials
    - Relying parties or eavesdroppers never learn the user's password

A. Harding, T. W. van der Horst, and K. E. Seamons. Wireless Authentication using Remote Passwords. *1st ACM Conference on Wireless Network Security (WiSec)*, Alexandria, VA, March 2008.

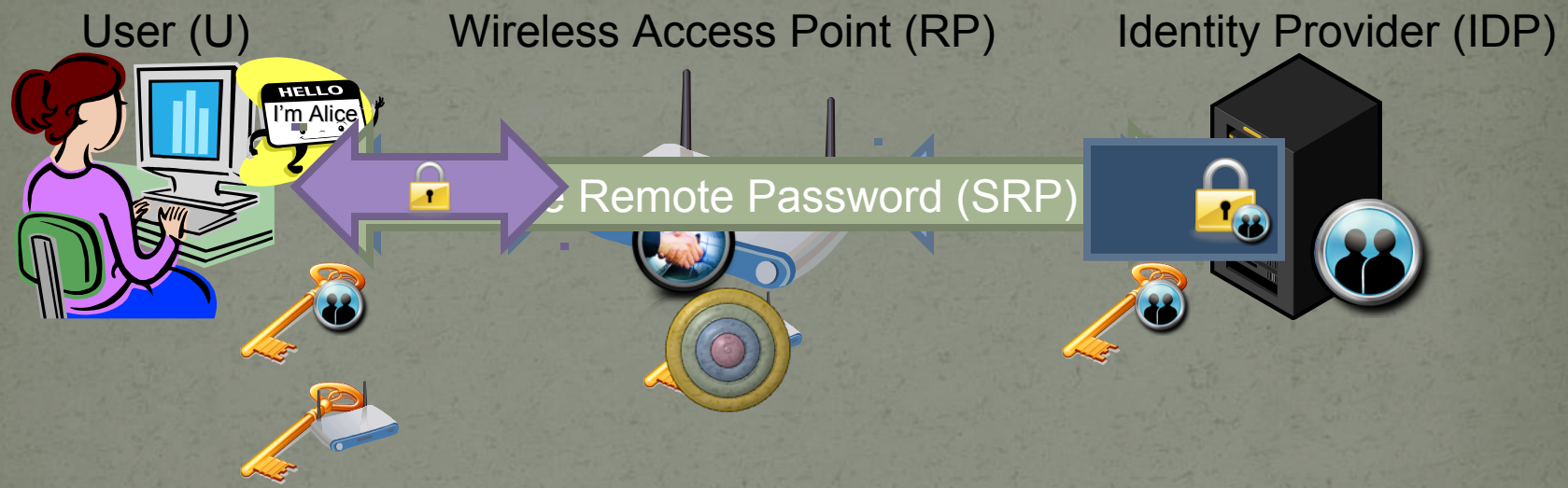
High Level Idea

- How do users authenticate to identity providers when they cannot directly communicate?
  - Giving relying parties the plaintext password is not desirable
  - Allowing an encrypted tunnel invites misuse and requires IP-level connectivity
    - Forwarding several small messages of known composition offers a good compromise
- How do identity providers inform relying parties that users have successfully authenticated?
  - A message sent to that effect by the identity provider could be forged/destroyed and potentially enables user impersonation
  - Token-based approach

Strong Password Protocol



sSRP, Version 1



1. Use SRP to establish a mutually authenticated session key between user and her identity provider
2. Use that key to facilitate a SAW token distribution

Current Status

- Create a generic protocol that supports web site login and wireless access
  - Surrogate SRP (sSRP)
  - Mode 1
    - No PKI!
    - No passive attacks, Limits active attacks
  - Model 2
    - RP and IDP have SSL server certificates
    - No client certificate
- Prototypes built for wireless access and web site login
- Usability studies being planned

XACML – The Standard

Hal Lockhart, BEA Systems

What is XACML?

- XML language for access control
- Coarse or fine-grained
- Extremely powerful evaluation logic
- Ability to use any available information
- Superset of Permissions, ACLs, RBAC, etc
- Scales from PDA to Internet
- Federated policy administration
- OASIS and ITU-T Standard

Trends Driving Fine-Grained Access Control

- De-perimeterization
  - No longer just “them and us”
  - Firewall is no longer sufficient
- Service Oriented Architecture
  - Multiple access contexts for each service
- Software as a Service (looking forward)
  - Complex interactions of internal and external components

OASIS XACML History

- First Meeting – 21 May 2001
- Requirements from: Healthcare, DRM, Registry, Financial, Online Web, XML Docs, Fed Gov, Workflow, Java, Policy Analysis, WebDAV
- XACML 1.0 - OASIS Standard – 6 February 2003
- XACML 1.1 – Committee Specification – 7 August 2003
- XACML 2.0 – OASIS Standard – 1 February 2005
- XACML 2.0 – ITU/T Recommendation X.1142

Powerful Policy Expression

- “Anyone can use web servers with the ‘spare’ property between 12:00 AM and 4:00 AM”
- “Salespeople can create orders, but if the total cost is greater that $1M, a supervisor must approve”
- “Anyone view their own 401K information, but nobody else’s”
- “The print formatting service can access printers and temporary storage on behalf of any user with the print attribute”
- “The primary physician can have any of her patients’ medical records sent to a specialist in the same practice.”

Key XACML Features

- Federated Policy Administration
  - Multiple policies applicable to same situation
  - Combining rules to resolve conflicts
- Decision may include Obligations
  - In addition to Permit or Deny
  - Obligation can specify present or future action
  - Examples: Log request, require human approval, delete data after 30 days
- Protect any resource
  - Web Server, Java or C++ Object, Room in building, Network Access, Web Service, Geographic Data, Health Records, etc.

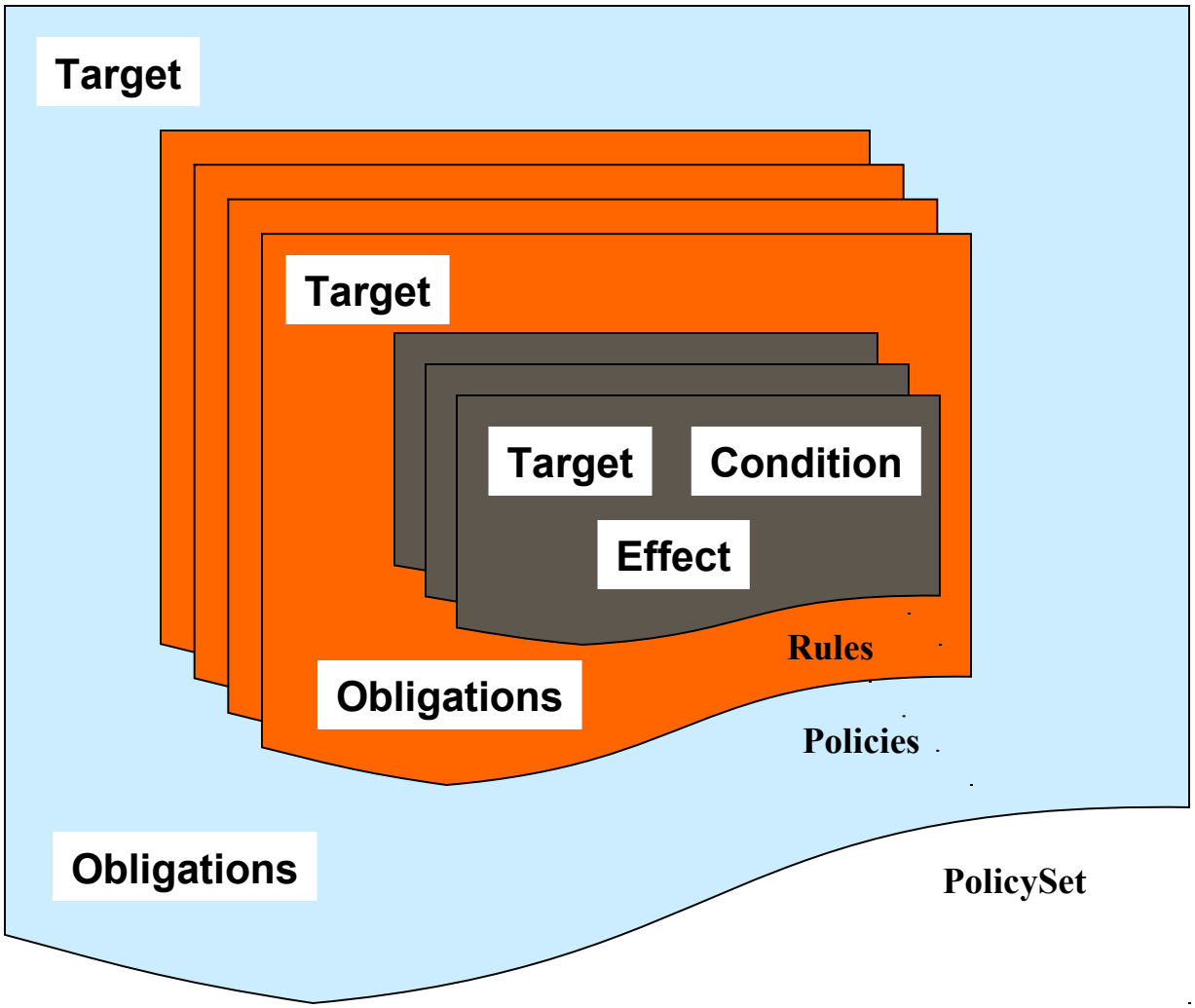
Novel XACML Characteristics

- Large Scale Environment
  - Subjects, Resources, Attributes, etc. not necessarily exist or be known at Policy Creation time
  - Multiple Administrators - potentially conflicting policy results
  - Combining algorithms
- Request centric
  - Use any information available at access request time
  - Zero, one or more Subjects
  - No invented concepts (privilege, role, etc.)
- Dynamically bound to request
  - Not limited to Resource binding
  - Only tell what policies apply in context of Request
  - Two stage evaluation

XACML Concepts

- Request and Response Contexts – Input and Output
- Policy & PolicySet – combining of applicable policies using CombiningAlgorithm
- Target – Rapidly index to find applicable Policies or Rules
- Conditions – Complex boolean expression with many operands, arithmetic & string functions
- Effect – “Permit” or “Deny”
- Obligations – Other required actions
- Bag – unordered list which may contain duplicates

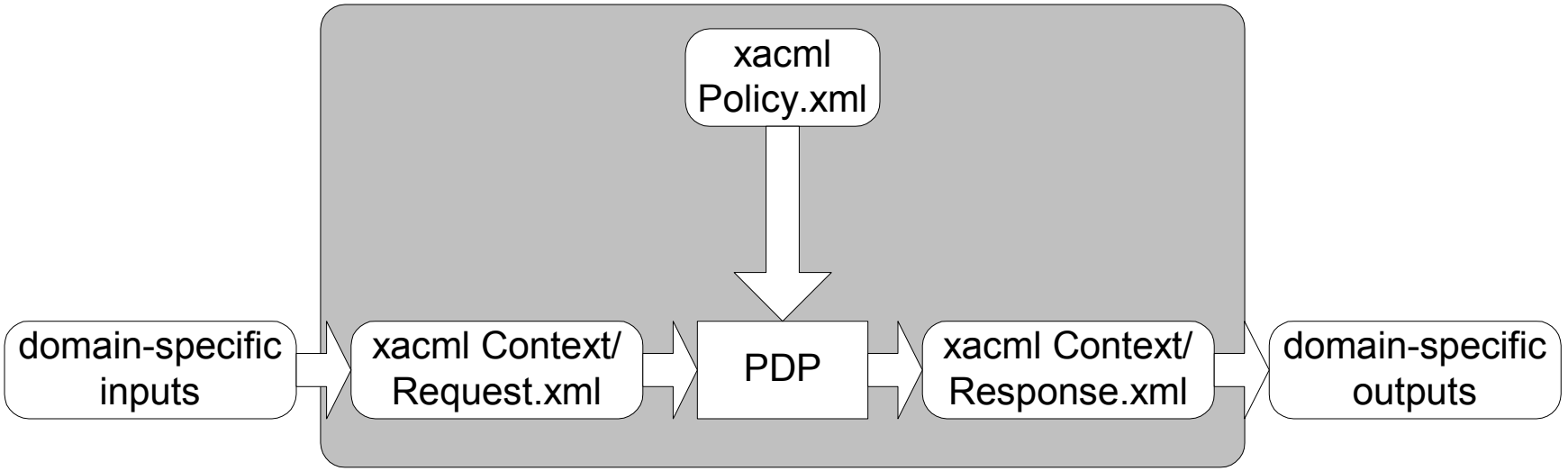
XACML Concepts



Policies and Policy Sets

- Policy
  - Smallest element PDP can evaluate
  - Contains: Description, Defaults, Target, Rules, Obligations, Rule Combining Algorithm
- Policy Set
  - Allows Policies and Policy Sets to be combined
  - Use not required
  - Contains: Description, Defaults, Target, Policies, Policy Sets, Policy References, Policy Set References, Obligations, Policy Combining Algorithm
- Combining Algorithms: Deny-overrides, Permit-overrides, First-applicable, Only-one-applicable

Request and Response Context



XACML 2.0 Profiles

- Digital Signature
  - Integrity protection of Policies
- Hierarchical Resources
  - Using XACML to protect files, directory entries, web pages
- Privacy
  - Determine “purpose” of access
- RBAC
  - Support ANSI RBAC Profile with XACML
- SAML Integration
  - XACML-based decision request
  - Fetch applicable policies
  - Attribute alignment

XACML Benefits

- Standard Policy Language
  - Investment protection
  - Skills reuse
- Leverage XML tools
- Policy not in application code
  - Reduce cost of changes
  - Consistent application
  - Enable audit

XACML Performance

- Some public comments based on ignorance
- Many optimization opportunities
  - Policy encoding
  - Request context
  - Partial evaluation
  - Decision Caching
  - Precomputed admin chaining
- Complex policies cost more to evaluate than simple
  - But is the difference more significant that other factors?

Current Work - XACML 3.0

- Administration/Delegation
- Schema generalization
- WS-XACML
- Obligation combining rules
- Policy provisioning
- Metadata/vocabulary advertisement
- Closely coupled PDP/PEP

Delegation with XACML 2.0

- Use of Intermediary Subject Category
  - Print Format Service can read any file a user wants printed, but not otherwise
  - Access Subject + Intermediary Subject
- Delegation by modifying attributes
  - User can enable family member's access
  - Policy protects subject repository
- Policies protecting each policy repository

Administration/Delegation

- Two primary use cases
  - “HR-Admins can create policies concerning the Payroll servers”
  - “Jack can approve expenses while Mary is on vacation”
- Backward compatible
- Likely to define two compliance levels
- Policies can contain Issuer
- Policies can be Access or Admin
- Admin policies enable policy creation

Administration/Delegation

- Situation – all information values used as policy inputs
- If policy issued by trusted issuer – use
- If not, look for Admin policy for Issuer covering current Situation
- Chain back to Trusted Issuer
- Actual processing is complex, because of interplay with policy combining

Other 3.0 Work

- Schema generalization
  - Improve extensibility
- WS-XACML
  - Builds on WS-Security Policy – more fine grained
  - Good for privacy policies
- Obligation combining rules
  - XACML 2.0 accumulates all Obligations
  - Characterize Obligation types – enable different treatments
- Policy provisioning
  - From repository distribute distinct policy subsets

XACML Interoperability Demo

- Burton Catalyst Conference
  - San Francicso - June 28, 2007
- Participating Organizations
  - BEA, CA, IBM, Jericho Systems, Oracle, Redhat, Securent, Symlabs
- Interop Features
  - Stock Trading Environment
  - Two Usecases
    - Authorization Decision – 18 Scenarios
    - Policy Exchange – 8 Scenarios

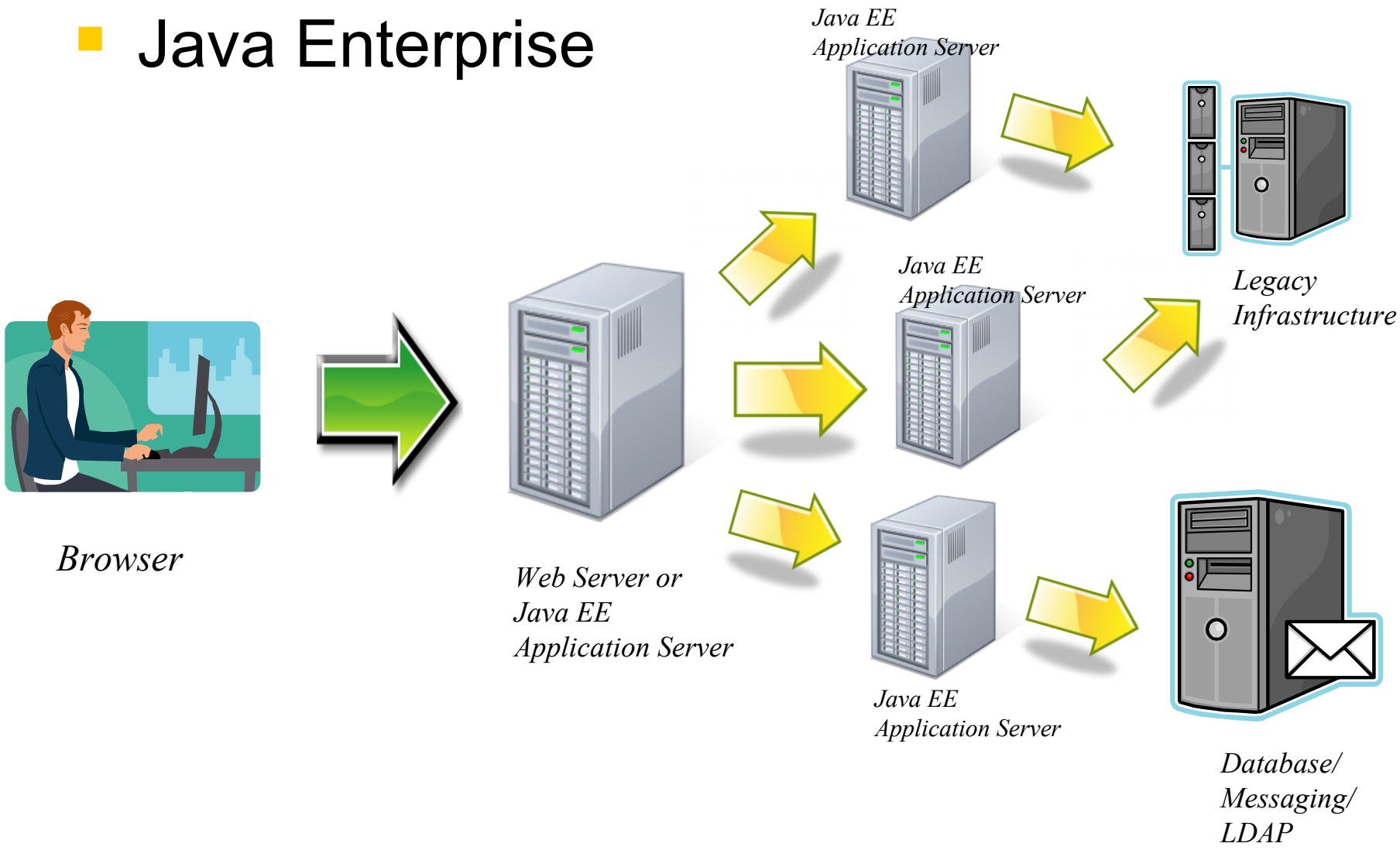
Identity and Access Control Extensions for Java Enterprise Edition (EE)

Anil Saldhana
Red Hat Inc.
Anil.Saldhana@redhat.com
http://anil-identity.blogspot.com

- Anil leads JBoss Security and Identity Management at Red Hat Inc.
- Member of OASIS Consortium
  - Secretary of SAML Technical Committee.
  - Member of XACML, WS-Federation and Enterprise Key Management TCs.
- Member of W3C
  - Co-editor of WSC-XIT Specification (WIP)

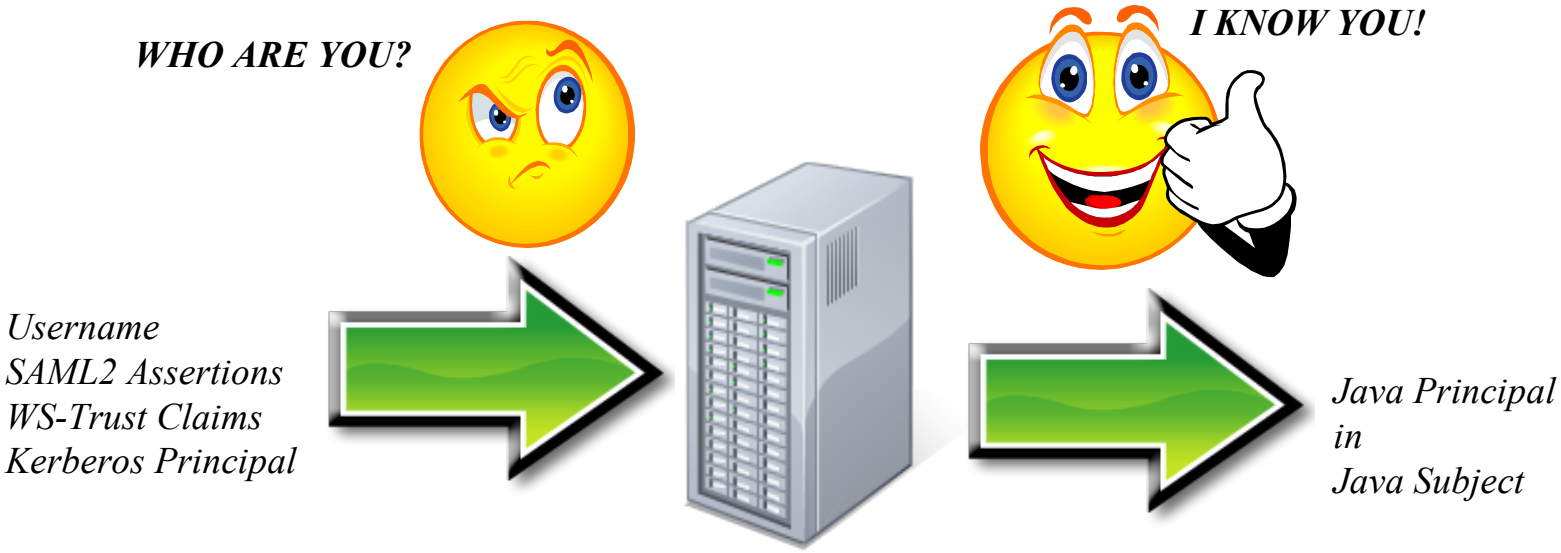
- Java Enterprise Edition (EE) is the premier specification in the Java Enterprise World.
  - Java Community Process (JCP) is the standards body.
  - Currently in version 1.5
  - Containers
    - Web, Enterprise Java Beans (EJBs) etc.
  - Coarse-grained security using RBAC.

Java Enterprise

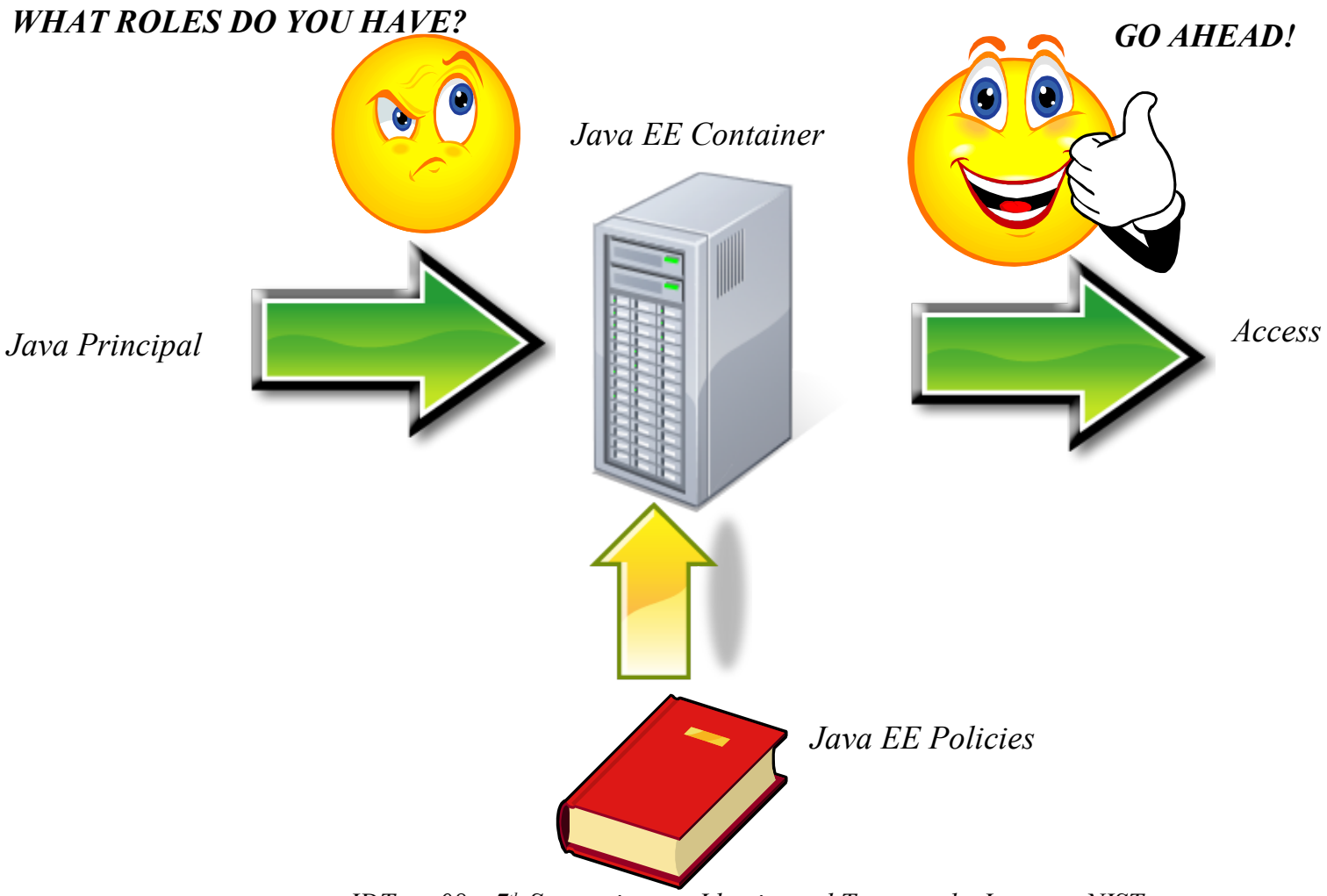


- Java EE Security
  - **Underspecified**.
  - Containers perform 2 sequential steps
    - Establish Principal (Authentication)
    - Determine Roles and undertake enforcement
  - RBAC based coarse-grained access control.
    - Roles shield
      - Web Resources, EJB Methods, Message Destinations.
  - **Security is an aspect external to app**

Java EE Containers Authentication



Java EE Containers Authorization



- Identity Extensions
  - Identity entering authentication phase
    - Certificates (CLIENT-CERT in Web world)
    - Username (JMS Connections)
    - Unspecified
  - Java Principal (in Subject) is the exit artifact.
  - Federated Identity can always be represented as a Java Principal.
    - Automatic extension of the Java EE Spec.

- Authorization Extensions
  - Specification mandated rules are insufficient
    - Web : Roles against web URL for resources
      - Contextual security needs to be provided (XACML)
        - Web resource accessible by employees on business days between 9am and 5pm from a particular subnet only.
  - Allow multiple policy technologies to make one collective decision
    - JACC, XACML, Custom Policies plug-n-play

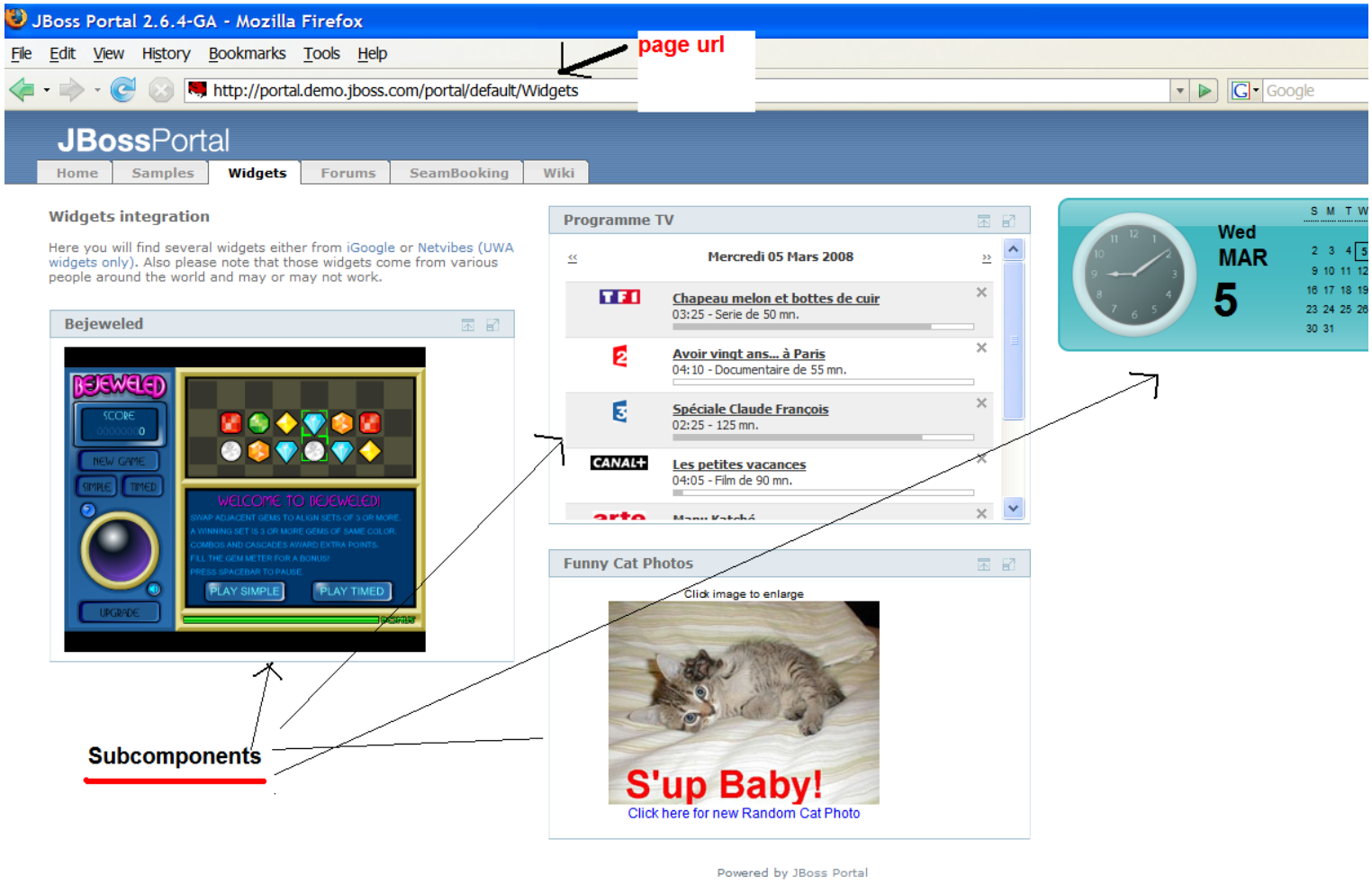
Authorization Extensions

- Example of a policy for resources

```
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                            access_control-xacml-2.0-policy-schema-os.xsd"
        PolicyId="urn:oasis:names:tc:xacml:2.0:jboss-test:X:policy"
       RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
<Description> Policy for Test X. </Description>
<Target/>
<Rule RuleId="urn:oasis:names:tc:xacml:2.0:jboss-test:X:rule" Effect="Permit">
 <Description> Anyone can perform any action on any resource if current-time is 08:23:47-05:00. </Description>
<Condition>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-equal">
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
<EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"
DataType="http://www.w3.org/2001/XMLSchema#time"/> </Apply>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">08:23:47-05:00</AttributeValue>
</Apply>
</Condition>
</Rule>
</Policy>
```

- Use Case – JBoss Portal
  - Portlets are web components running in a Portlet Container (JSR-168)
  - Portal page can contain multiple sub components such as sub pages, sub windows etc.
  - Subcomponents need entitlements.
    - An identity may have access to 5 subcomponents out of 20 on a page.

Use Case – JBoss Portal



- Use Case – JBoss Portal
  - Need for fine-grained authorization is evident
    - XACML is a strong candidate (+)
    - Alternative is a custom ACL implementation (-)
    - JavaEE web.xml access control semantic falls short.
  - Identity can be a federated identity

Use Case – JBoss Portal - Policy

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="…"  xsi:schemaLocation="…." PolicyId="…"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>Policy for Portal Use Case.</Description>
  <Target/>
  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:test:II:rule" Effect="Permit">
    <Description>Portal accessible between 9 am and 5pm</Description>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://host/companyportal/</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
```

Use Case – JBoss Portal - Policy

```
<Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
          <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#time"
AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time" />
         </Apply>
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">09:00:00</AttributeValue>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal">
           <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
             <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#time"
AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time" />
           </Apply>
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">17:00:00</AttributeValue>
         </Apply>
 </Condition>
</Rule>
```

Use Case – JBoss Portal - Policy

```
<Rule RuleId="urn:oasis:names:tc:xacml:2.0:jboss-test:IX:rule“ Effect="Permit">
  <Description>The EighteenYearOld page accessible if you are 18</Description>
  <Target>
     <Resources>
      <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
               <AttributeValue DataType="#anyURI">http://host/companyportal/EighteenYearOld/</AttributeValue>
               <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id“
DataType="#anyURI"/>
            </ResourceMatch>
          </Resource>
       </Resources>
    </Target>
     <Condition>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
         <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:jboss-test:age"
           DataType="http://www.w3.org/2001/XMLSchema#integer"/>
        </Apply>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">18</AttributeValue>
       </Apply>
     </Condition>
   </Rule>
</Policy>
```

Use Case – JBoss Portal – Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os" ...>
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="...#string">
      <AttributeValue>Anil Saldhana</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="...#anyURI">
      <AttributeValue>http://host/someportal/</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="...#string">
      <AttributeValue>read</AttributeValue>
    </Attribute>
  </Action>
  <Environment>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time" DataType="...#time">
      <AttributeValue>09:23:47-05:00</AttributeValue>
    </Attribute>
  </Environment>
</Request>
```

Use Case – JBoss Portal – Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response
    xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
     access_control-xacml-2.0-context-schema-os.xsd">
  <Result>
    <Decision>NotApplicable</Decision>
    <Status>
      <StatusCode
          Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

- Q & A

XACML 2.0 Interop 2008

Anticipated Participants

- Axiomatics
- BEA Systems
- IBM
- Oracle
- Red Hat
- Cisco
- Sun
- U.S dept of Veterans Affairs

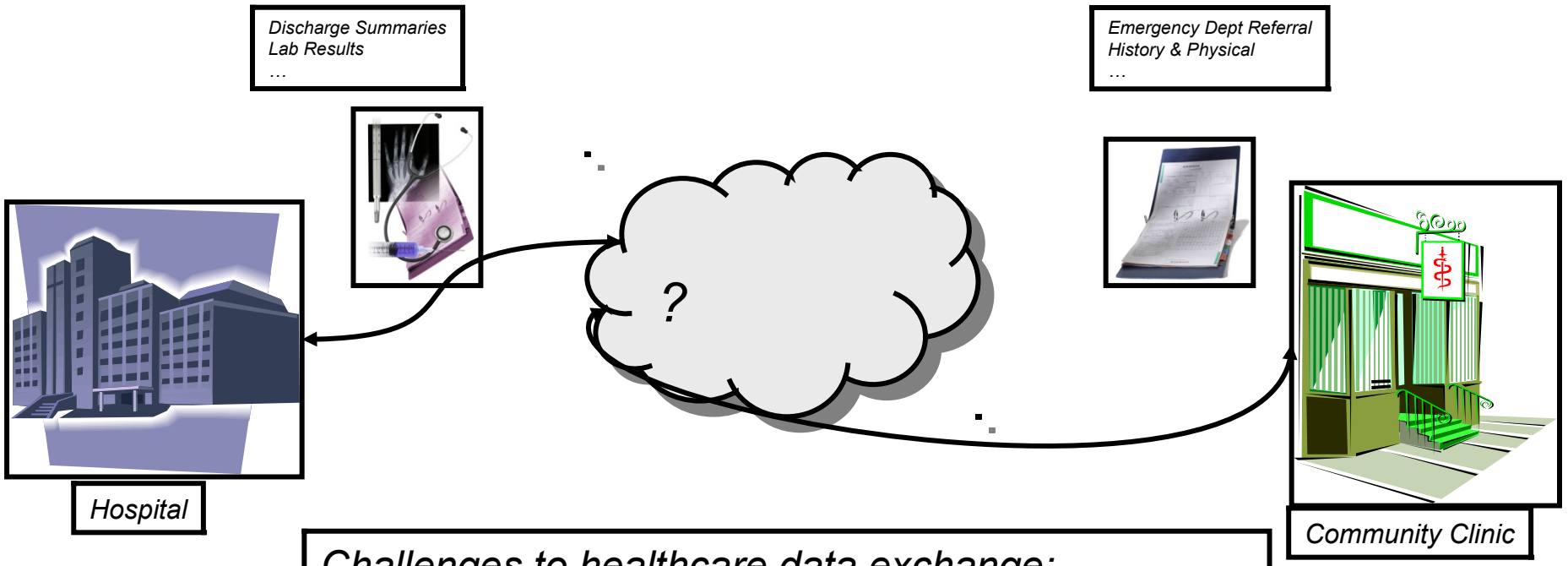
Goals

- The interop will demonstrate the use of XACML V2.0 capabilities in a healthcare scenario involving the election and enforcement INCITS compliant clinical roles and patient privacy directives.

Features/Highlights

- Privacy and security in XACML policy
- HL7 role-based
- Use of HL7 consent code
- Over-ride of patient election by declaring an emergency
- Use of obligations to continue to enforce dynamic patent privacy directives even after release to another system

Scenario



Challenges to healthcare data exchange:

- *Proper Authorization & Privacy*

Scenario

- In the scenario, examples of likely patient privacy directives are stored in a shared policy administration point (central repository).

- Each participant will act as a healthcare enterprise (partner facility) within an identity federation sharing a common repository of patient privacy preferences and consent directives.

- Partner/users (clinicians) request protected patient information from their healthcare facility. Before the release of protected patient medical information to the clinician, each participant will evaluate the clinician roles (in the form of HL7 clinical permissions) and the patient privacy directives stored in the central repository.

Scenario (continued)

- Access to patient information will not be "all or nothing," rather portions of the medical record not releasable to the clinician based on the patient privacy directives will be "filtered" from the provided information view.

- Variations on policy will be used to demonstrate that the patient privacy directives are being honored.

- Changes to the patient privacy directives at the central repository will be reflected in changes to access to the patient information at the partner facility.

XACML Interop at RSA2008

Andreas Sjöholm
Product manager
Axiomatics

XACML Interop at RSA2008

- 2nd XACML Interop
- Demonstrate XACML 2.0 interoperability
- XACML 2.0 capabilities in a healthcare scenario
- Utilizing HL7 etc.
- Axiomatics, BEA Systems, IBM, Oracle, Red Hat, Cisco, Sun and U.S dept of Veterans Affairs

High level objectives

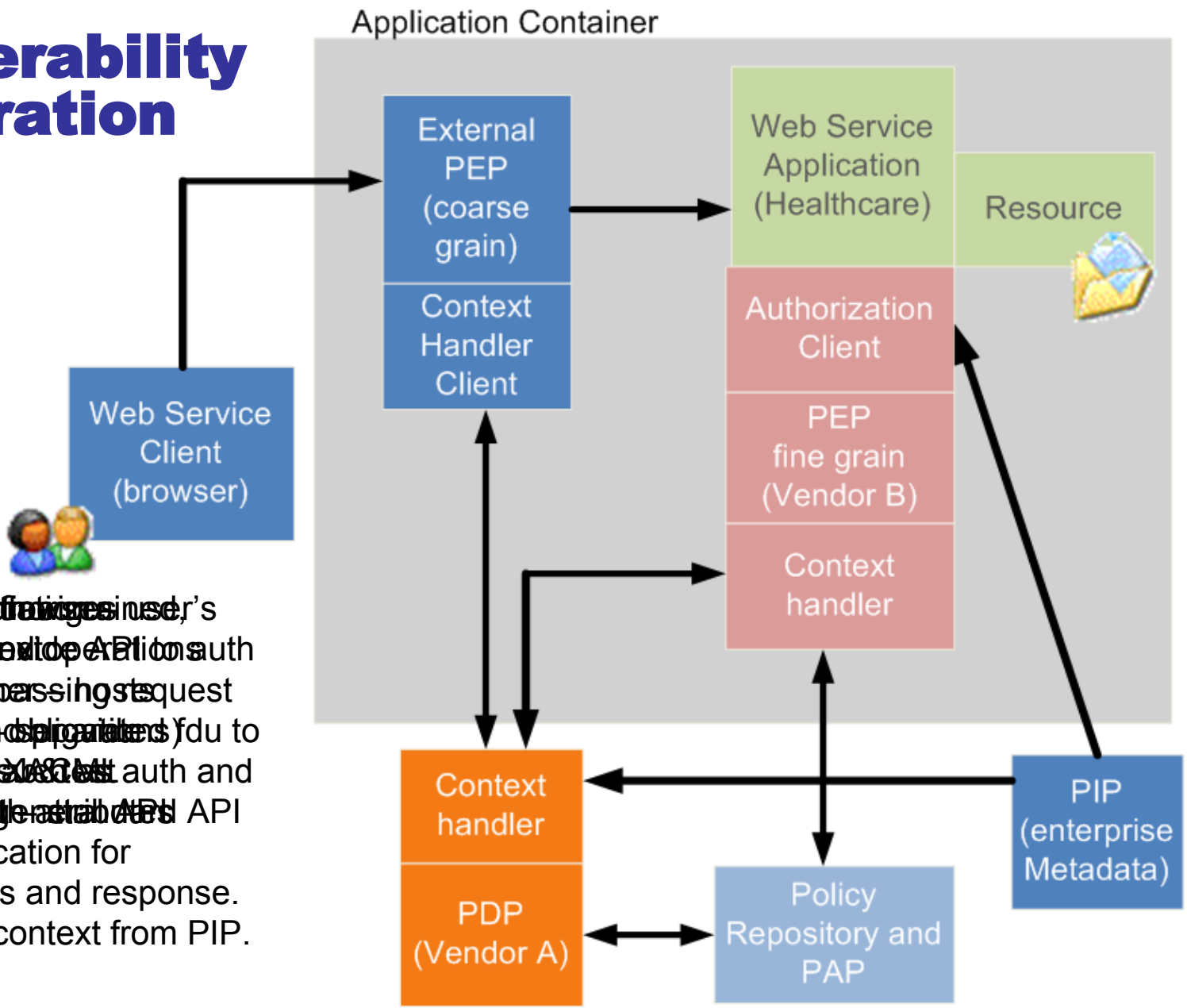
- Control access to *specific* portions of a healthcare record
- Filter sensitive clinical information from being viewed
- Ensure obligations are met
- Provide vehicle to override consent (emergency overrides)

= can-know or must-not-know basis

Use Cases

- Policy exchange
- Authorization Decision Req/Resp
  - Fine grain auth
  - HL7 Permission based access
  - HL7 Patient consent directives
  - Data filtering obligations
  - Emergency override obligation

Interoperability configuration



Web Service Client (browser) - Provides user's credentials and operation to be performed on resource to enterprise application

External PEP (coarse grain) - Intercepts request and response, extracts credentials and operation to be performed on resource, invokes Context handler client API to facilitate passing request and response to enterprise application for submitting requests and response. Gets applications context from PIP.

Context Handler Client - Provides specific extended API to auth and authorization enterprise application for submitting requests and response. Gets applications context from PIP.

Web Service Application (Healthcare) - Enterprise application for submitting requests and response. Gets applications context from PIP.

Resource - Resource to be accessed

Authorization Client - Provides specific extended API to auth and authorization enterprise application for submitting requests and response. Gets applications context from PIP.

PEP fine grain (Vendor B) - Intercepts request and response, extracts credentials and operation to be performed on resource, invokes Context handler client API to facilitate passing request and response to enterprise application for submitting requests and response. Gets applications context from PIP.

Context handler - Provides specific extended API to auth and authorization enterprise application for submitting requests and response. Gets applications context from PIP.

PDP (Vendor A) - Provides specific extended API to auth and authorization enterprise application for submitting requests and response. Gets applications context from PIP.

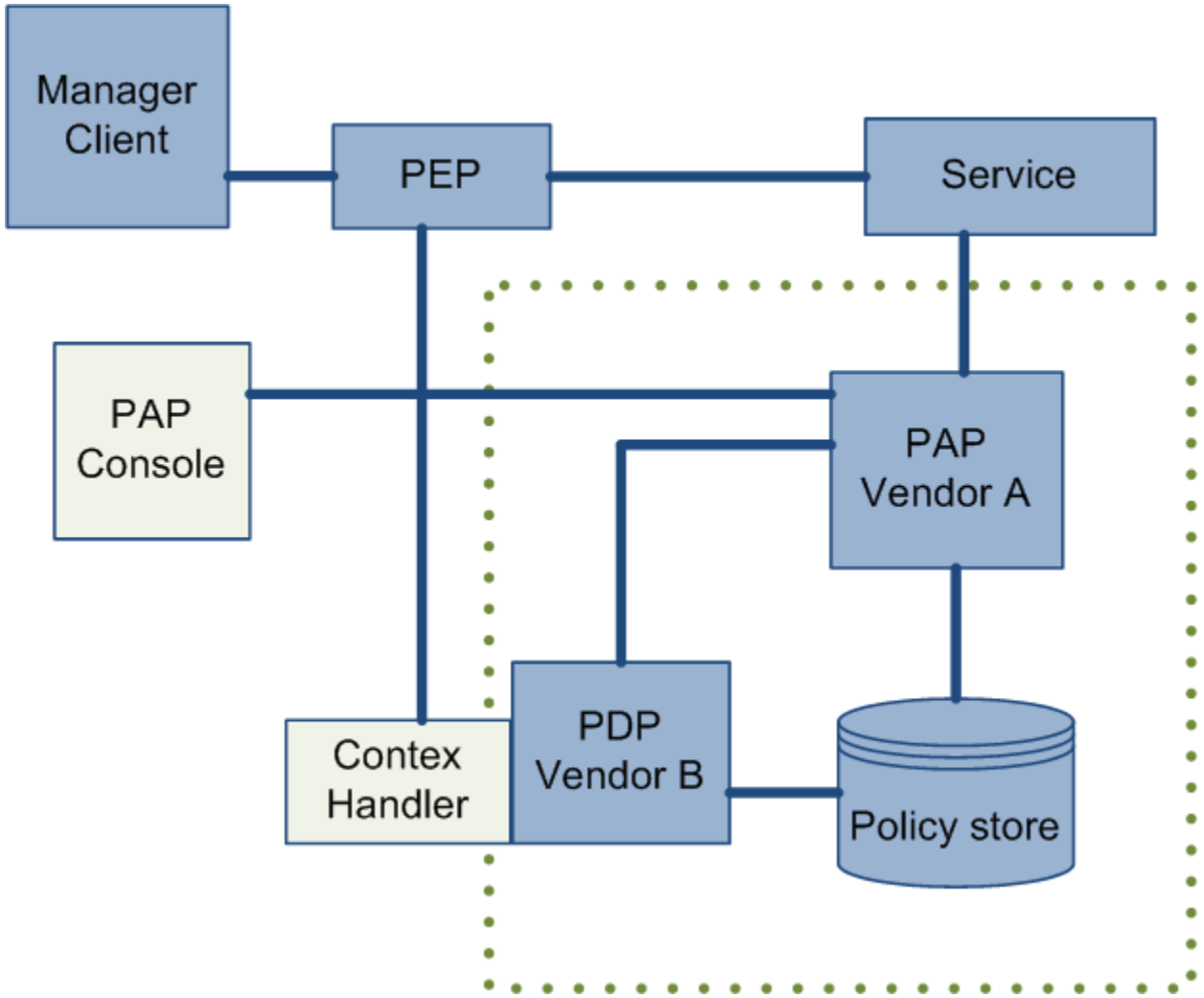
Policy Repository and PAP - Provides specific extended API to auth and authorization enterprise application for submitting requests and response. Gets applications context from PIP.

PIP (enterprise Metadata) - Provides specific extended API to auth and authorization enterprise application for submitting requests and response. Gets applications context from PIP.

Use Case: Policy Exchange

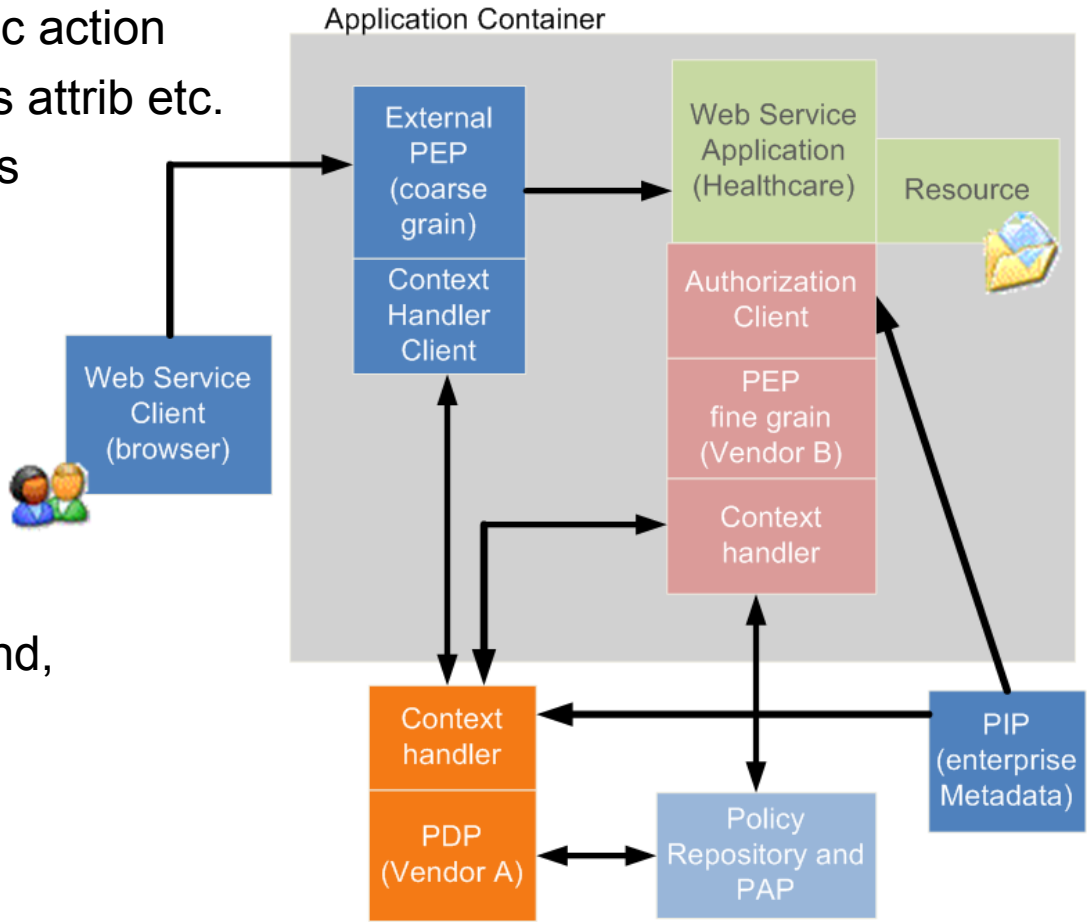
Pri focus (inner):
PAP creates policy
Notification
PDP uses

Next step (outer):
Larger context with
Attribute management
Manager services



Use Case: Fine Grain Auth

- Web browser access Health Care App
- When auth needed for specific action
  Healthcare auth client collects attrib etc.
- Embedded PEP take requests
- Normative XACML resp/req



- Coarse grained auth: front end, establish context

Patient Consent Directives

- Patient authorizes direct providers, but those not assigned to their case should not have access.
- Patient authorizes normal care, except for Dr. Bob Busybody (who is his nosy neighbor)
- Patient authorizes normal care, and further authorizes use of their data by cancer researchers
- Patient authorizes normal care, but requires a confidential S/MIME email sent describing each access.

Patient Consent Directives

HL7 confidentiality codes

N	Normal
CDA	Restricted by consent directive
S	Sensitive
SSA	Shared Secret Access
PSY	Psychiatry related item
MA	Masked access
U	User based access
…	…

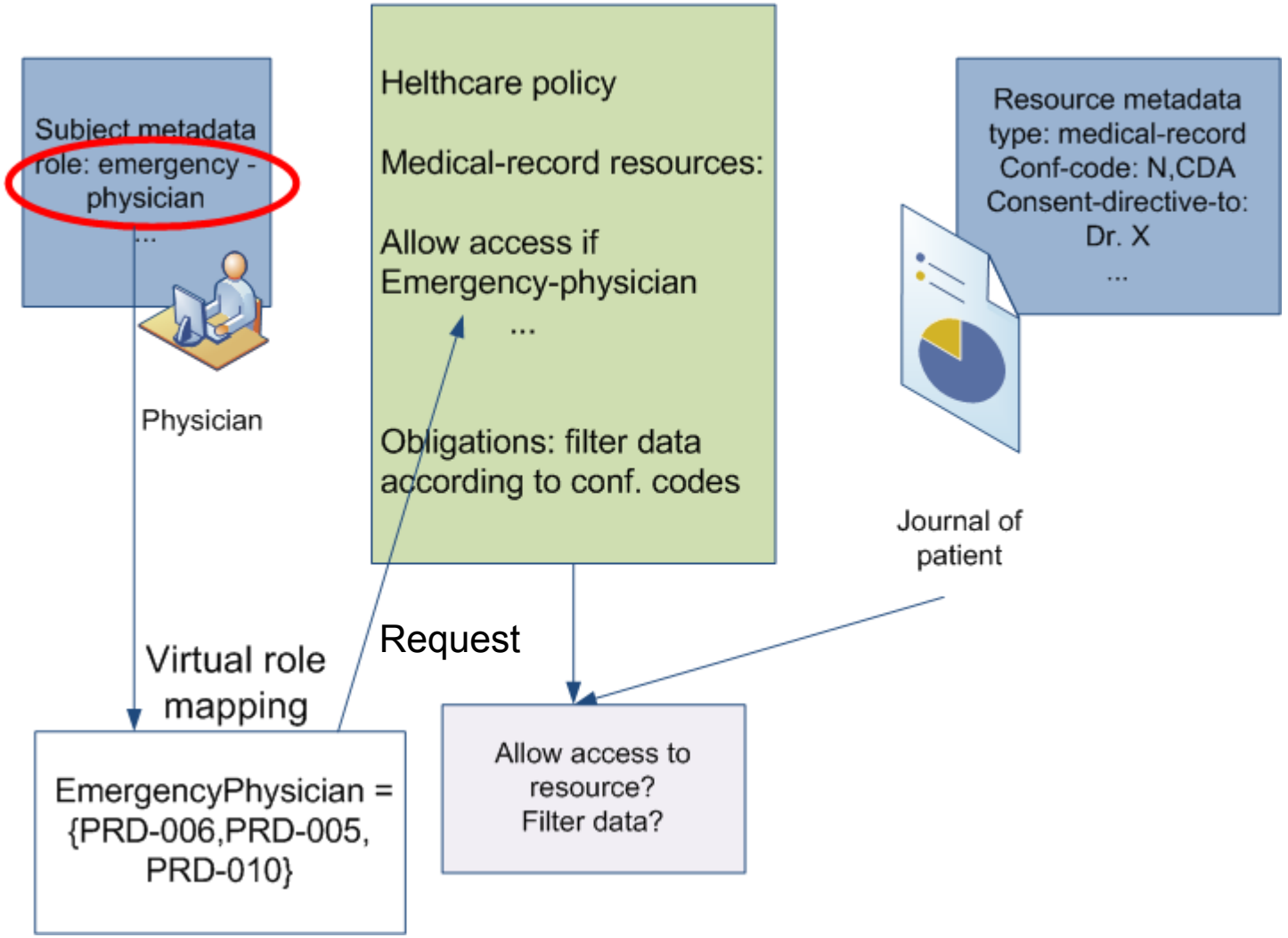
HL7 Permission codes

PRD-006	Patient Identification and Lookup
PRD-017	Review Progress Notes
PRD-012	Review Past Visits
PRD-003	Review Medical History
PRD-005	Review Vital signs/Patient Measurements
PRD-009	Review Current Directory of Provider Information
PRD-010	Review Patient Medications
…	…

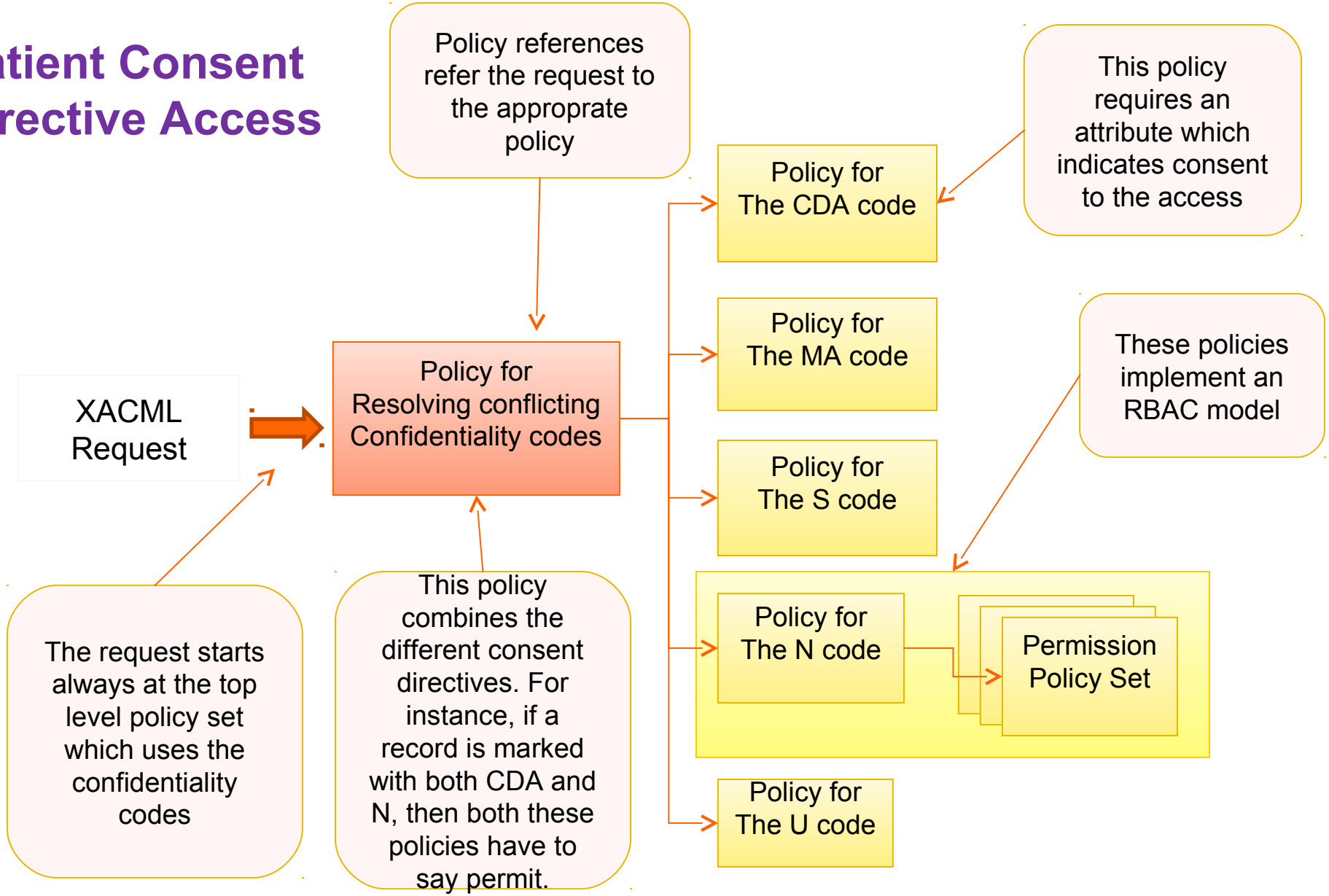
HL7 Permission based access

- XACML 2.0 RBAC Profile
- Demonstrate use of HL7 Identifiers
- Local roles vs. HL7 standard permissions (inter-organizational purposes)
- Requesting user obtains a set of HL7 permissions
- Maps to virtual role

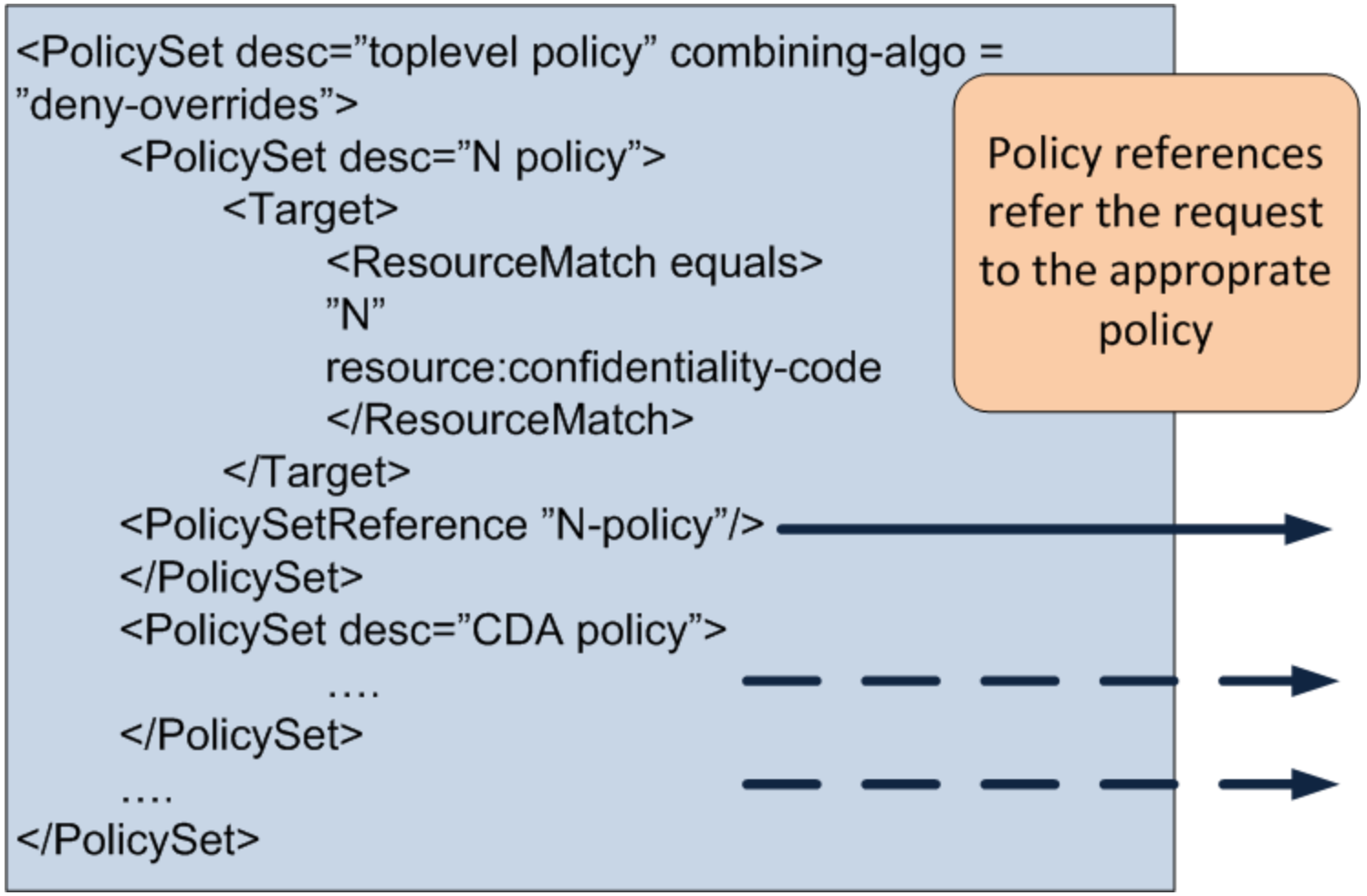
HL7 Permission based access



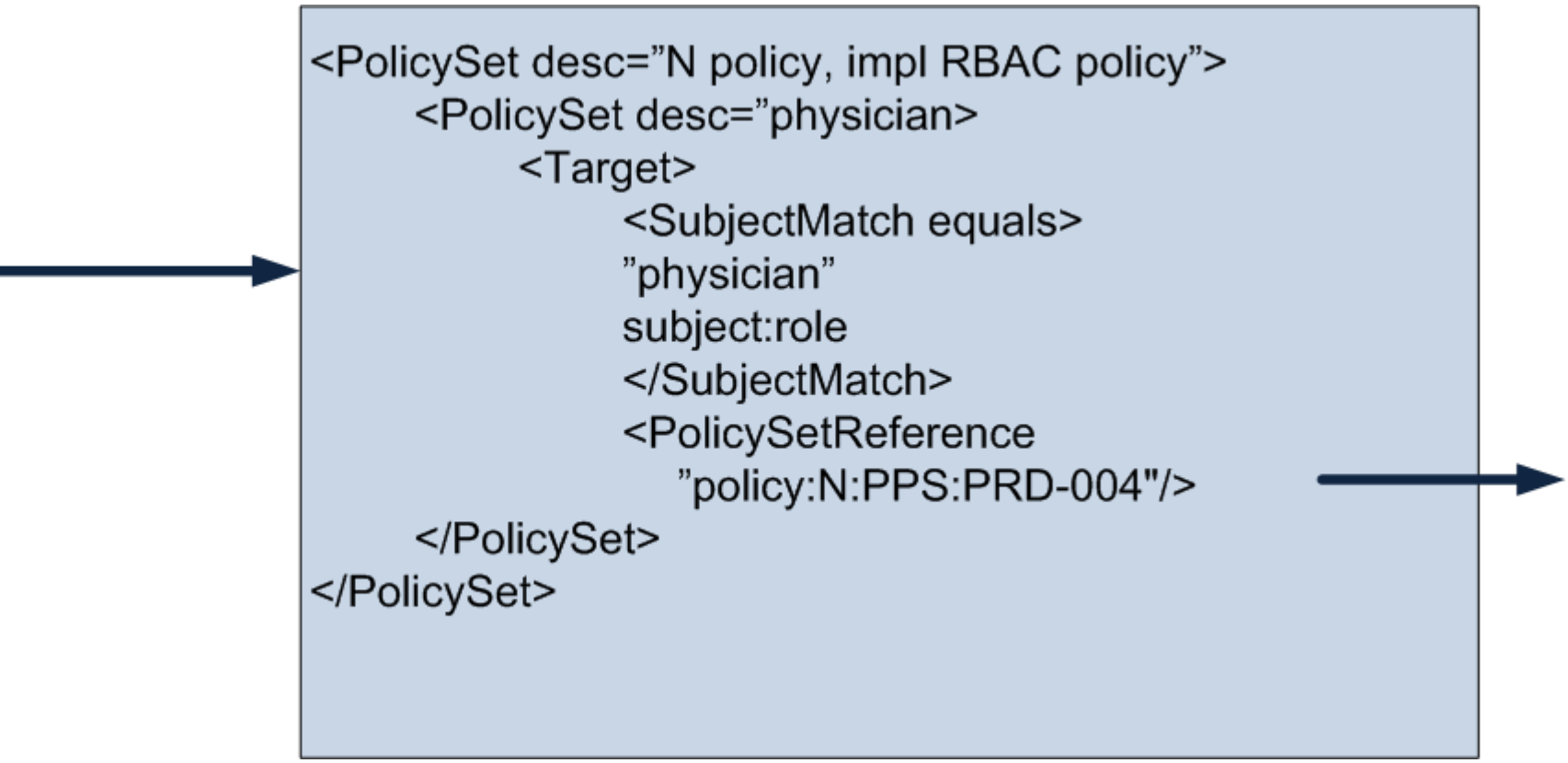
Patient Consent Directive Access



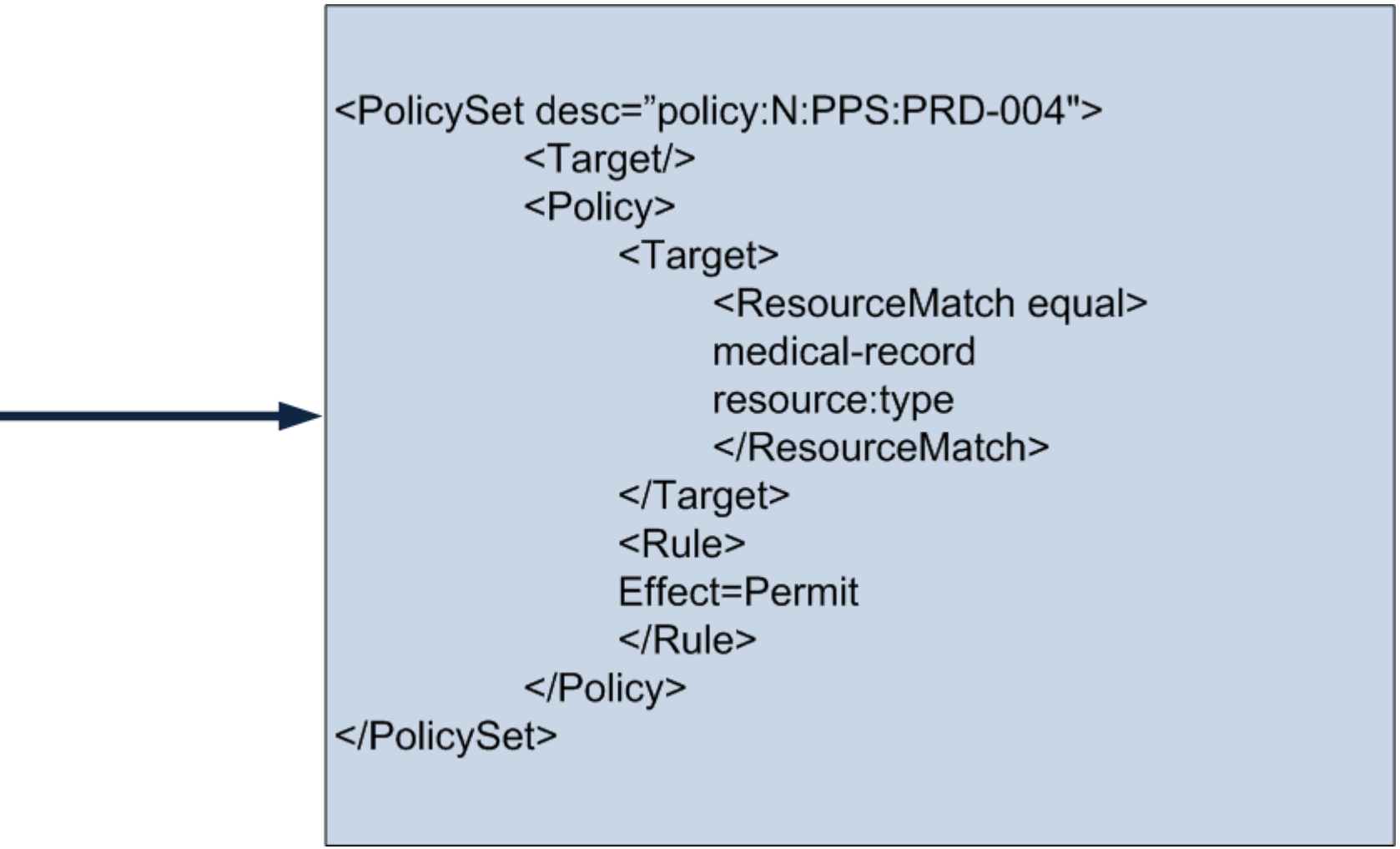
Top-policy for resolving conflicting confidentiality codes



…policy when accessed resource has confidentiality code N (Normal)…



…policy when access subject is role:physician.



Access request

<Request>

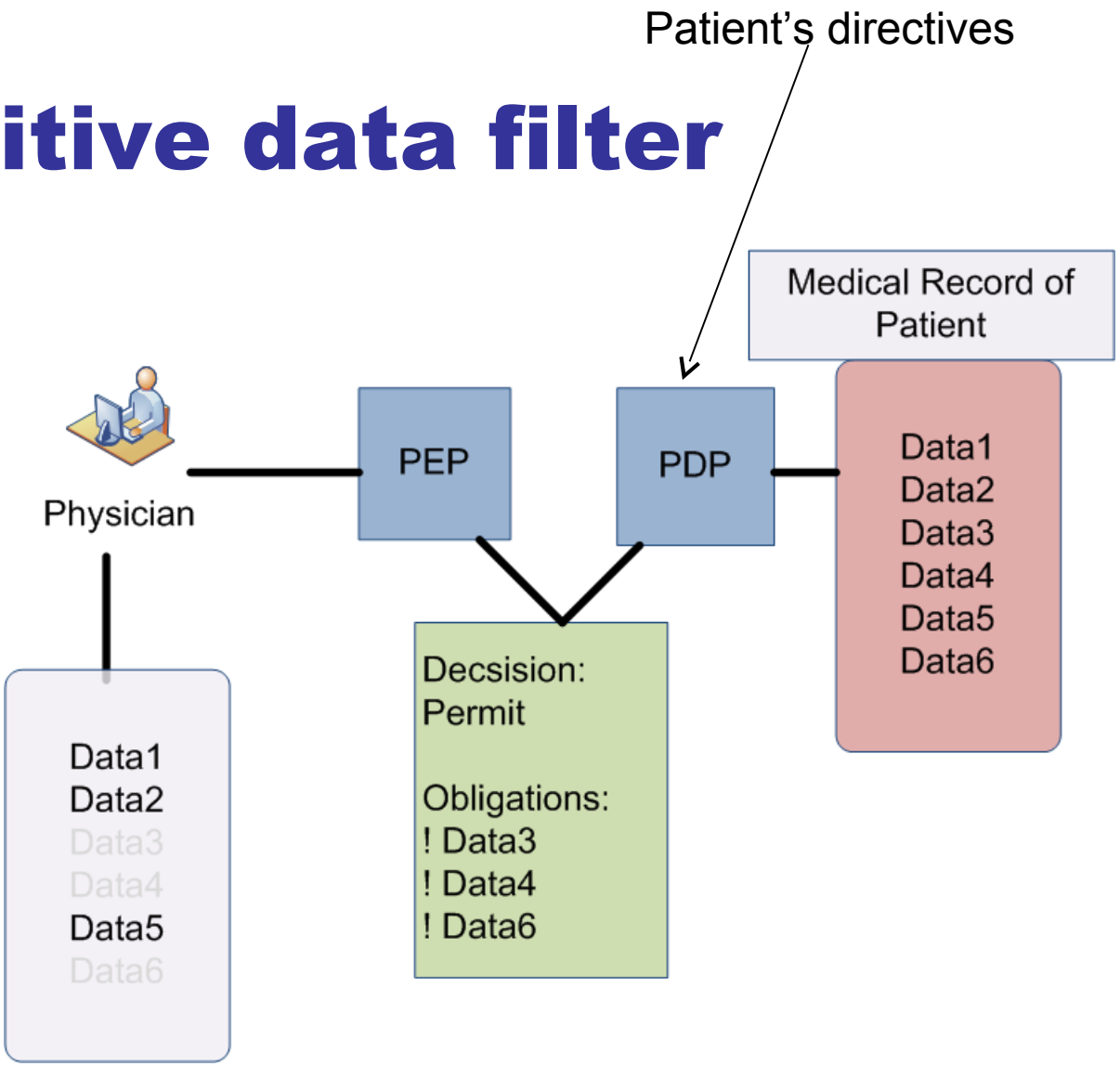
Subject Attribute: subject-id = Julius Hibbert
Subject Attribute: subject-role = physician
Resource Attribute: resource-id = record/patient/
BartSimpson
Resource Attribute: conf-code = CDA
Resource Attribute: conf-code = N
Resource Attribute: consented-subject-id =
Julius Hibbert
Resource Attribute: resource-type = medical record

</Request>

Decision response

- Access permitted
- XACML Obligations - filter out certain sensitive data

Sensitive data filter



XACMLPatientPrivacy

- JAVA EE
  - Java Server Faces (JSF) 1.2
  - Java API for XML Web Services (JAX-WS) 2.1
- Functionality
  - Patient elections
  - Local entity patient search
  - Patient Demographics
  - Patient Chart (problem list, procedures, lab, meds, vitals and radiology)
  - Clinical Notes
  - Patient Directive override for chart items, demographics, and notes.

OASIS

Advancing open standards for the information society

RSA2008 Patient Security and Privacy Testbed

Provider1, RSA2008 Version: 0.2

- Patient Info
- Chart
  - Problem List
  - Procedures
  - Laboratory
  - Radiology
  - Medications
  - Vitals
- Notes
- Patient Search
- Logoff

Interop Tools

- Set Patient Elections
- Provider Permissions

Patient Info

Name GURROLA,ANTHONY **Facility**

Gender Male Female **DoB** 1957-06-16T00:00:00-07:00

Chart Cover Sheet

Active Problems & Diagnosis

ICD9	Desc.	Onset
V70.5 2	Not Available	Feb 3, 2004
V72.1	Not Available	Jan 29, 2004
V65.43	Not Available	Jan 29, 2004
780.99	Not Available	Jan 20, 2004
V72.83	Not Available	Oct 8, 2003
722.0	Not Available	Oct 8, 2003

Active/Recent Medications

Med Name	Dosage	Fill Date	NDC
RANITIDINE (ZANTAC)--PO 150MG TAB	null	Feb 10, 2004	00781-1883-10
FEXOFENADINE (ALLEGRA)--PO 60MG TAB	null	Feb 10, 2004	00088-1107-55
KETOCONAZOLE (NIZORAL)--TOP 2% CREA	null	Feb 5, 2004	50458-0221-30
AMOXICILLIN (AMOXIL)--PO 500MG CAP	null	Nov 17, 2003	00003-0109-60
RANITIDINE (ZANTAC)--PO 150MG TAB	null	Jul 31, 2003	00781-1883-10
FEXOFENADINE (ALLEGRA)--PO 60MG TAB	null	Jul 31, 2003	00088-1107-55

Allergies

Type/Name	Date Noted
No results found	

Vitals

Type	Value	Date
No results found.		

Recent Labs

Lab Name	Completed	Value
GC PROBE	Dec 27, 2007	N
CHLAMYDIA PROBE	Dec 27, 2007	N
CREATININE	Nov 19, 2007	0.8

Recent Procedures

Procedure	CPT	Date
	92559	Jan 29, 2004
	99070	Jan 29, 2004
	99071	Jan 29, 2004
	6123	Feb 27, 2003
	23971	Feb 27, 2003

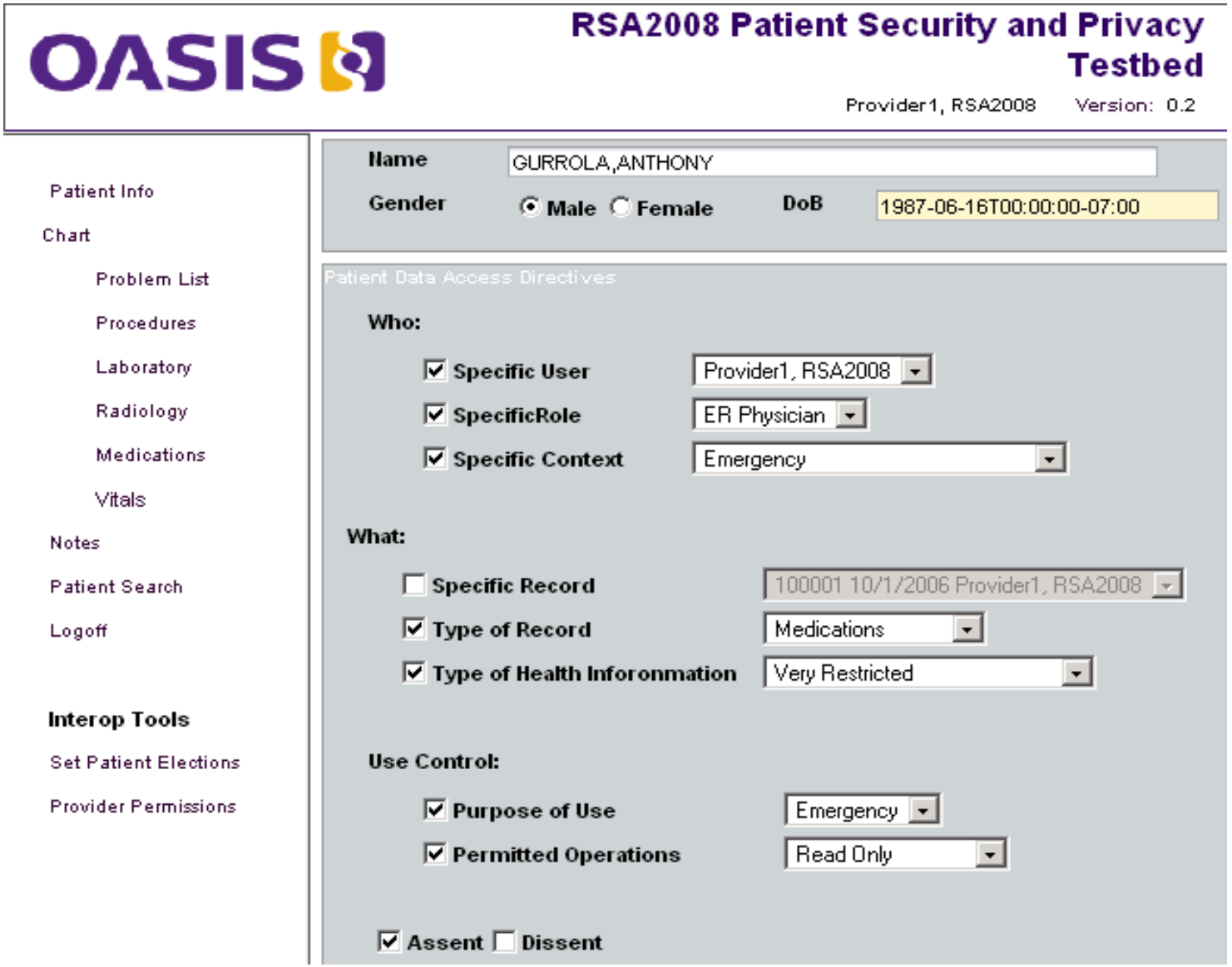
Recent Radiology Procedures

Procedure	ID	Status	Date
CT, CHEST (W/ CONTRAST) (PG)	4128	0	Nov 20, 2007
PORT CHEST	398	0	Nov 20, 2006

Clinical Reminders

Requirement	Due Date
No results found.	

Emergency override



Thank you and
see you at RSA2008!

Sunil Madhu’s presentation is not available, due to encryption in the Microsoft Powersoft source document.

OASIS

OpenID: Status and Challenges

IDTrust Symposium
March 4-6 2008

George Fletcher

OASIS

What is OpenID?

- Lightweight
- De-centralized
- Single-Sign-On
- System

- "OpenID is a free and easy way to use a single digital identity across the Internet."

- http://openid.net

OASIS

What's Happening?

- OpenID Foundation
  - Established
  - IPR, Process, Membership, Bylaws defined
  - Board announced
  - Local chapters

OASIS

What's Happening?

- Adoption… It's growing :)
  - OpenID Providers
    - Yahoo!, Blogger, AOL, Live Journal, MyOpenID.com, LinkSafe, etc
    - 60 listed at openiddirectory.com
  - OpenID Relying Parties
    - Plaxo, Pibb, Magnolia, Blogger comments, Live Journal, Wordpress, wikis, blogs, etc
    - Many, many more listed at openiddirectory.com

What's Happening?

- Specifications
  - OpenID 2.0 Final
  - OpenID Attribute Exchange 1.0 Final
  - OpenID Provider Authentication Policy Extension (Draft)

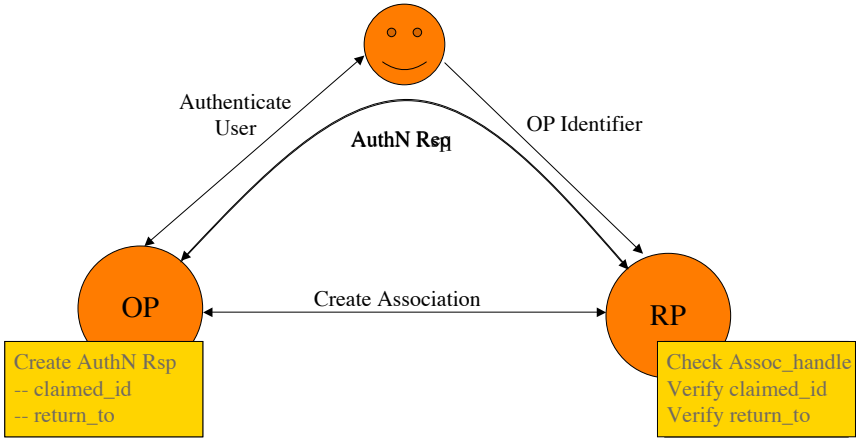
OASIS

What's Happening?

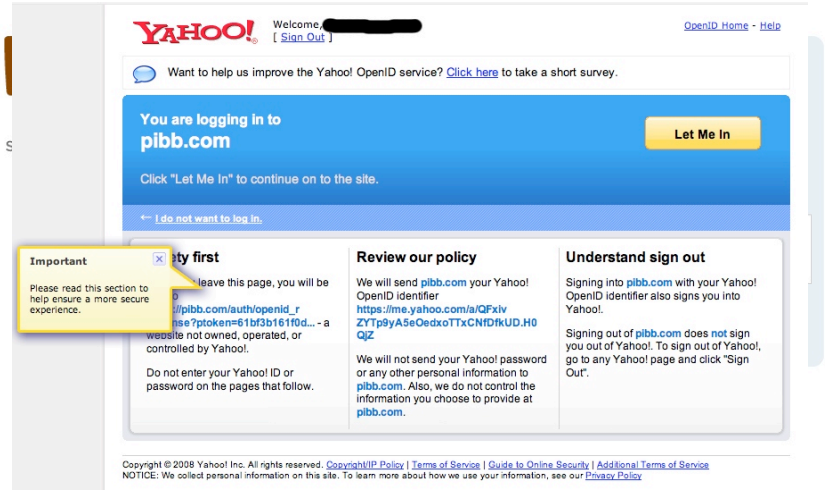
- OpenID 2.0 Specification Highlights
  - Enhanced security
    - RP verification (Section 13)
    - Associations via SSL
  - Better user experience
    - Directed Identity
    - Identifier recycling protection
  - Enhanced privacy
    - Pseudononymous identifiers

OASIS

Relying Party Verification



OASIS



What's Next?

- Specifications
  - Interesting discussions around federation and SSO/SLO
  - Finalization of the Provider Authentication Policy Extension
  - Reputation Extension

Challenges

- Perception
- Relying Party (business)
- User Experience
- Technical

Perception Challenges

- What is it good for? Where does it fit?

- "*But OpenID doesn't have the privacy characteristics that would make it suitable for government applications or casual web surfing.*"

- Kim Cameron's Blog -- Sunday Feb. 24

[Microsoft is a founding board member of OIDF]

Challenges

- Adoption by Relying Parties
  - Risk management
    - Do I trust the user?
    - Do I trust the OpenID Provider?
    - If the user is "bad" what do I lose?
    - What is the liability in a fraudulent transaction?

Challenges

- User Experience
  - The user themselves :)
    - What is an OpenID?
    - Why do I need one?
    - How will it help me?
    - Is it secure?

OASIS

Challenges

- User Experience
  - Bridging the gap… identity agents
    - Be proactive
    - Detect when the user is authenticated
    - When that identity can be used a another site
    - Ask the user if they want to “login” with their current identity?

OASIS

Technical Challenges

- OpenID 2.0 more complex
  - Open source libraries available
- OpenID 2.0 “backward compatibility”
- Difficult to “chain” to back-end web service calls
  - Attribute Exchange is the best option right now
  - May need it’s own extension

OASIS

Lessons Learned

- User education still needed
  - Spreadopenid.com
  - Driving usage is better than text
- Relying Party support needs substantial growth before OpenID can become mainstream
- On-the-fly risk mitigation is non-trivial for those resources that require it

Questions & Answers

- Contact Info:

- George Fletcher
- George.fletcher@corp.aol.com
- http://practicalid.blogspot.com