



FedoraCommons™

Repositories in Web, SOA, WOA, and
Enterprise Architectures

Dan Davis

Advanced Camp

June 19, 2007

Agenda

- Goals
- Dealing with Complexity
- Repository fit with the Web Architecture (as Example)
- Key Trends
- Web and Enterprise Paradigms
- Service Oriented and Web Oriented Architectures
- The Fedora Dilemma
- Fedora Design Goals as Example
- Repository for Integration

Challenges to Address

How can I link all my existing assets, create new business logic, and tie it all together in a way that automates my mission?

... and drive flexibility and efficiency at the same time?

... despite the highly decentralized nature of higher education



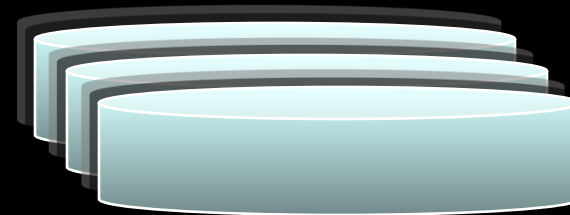
People
Partners
Institutions
Employees



Systems
Partners
Institutions



Existing Applications



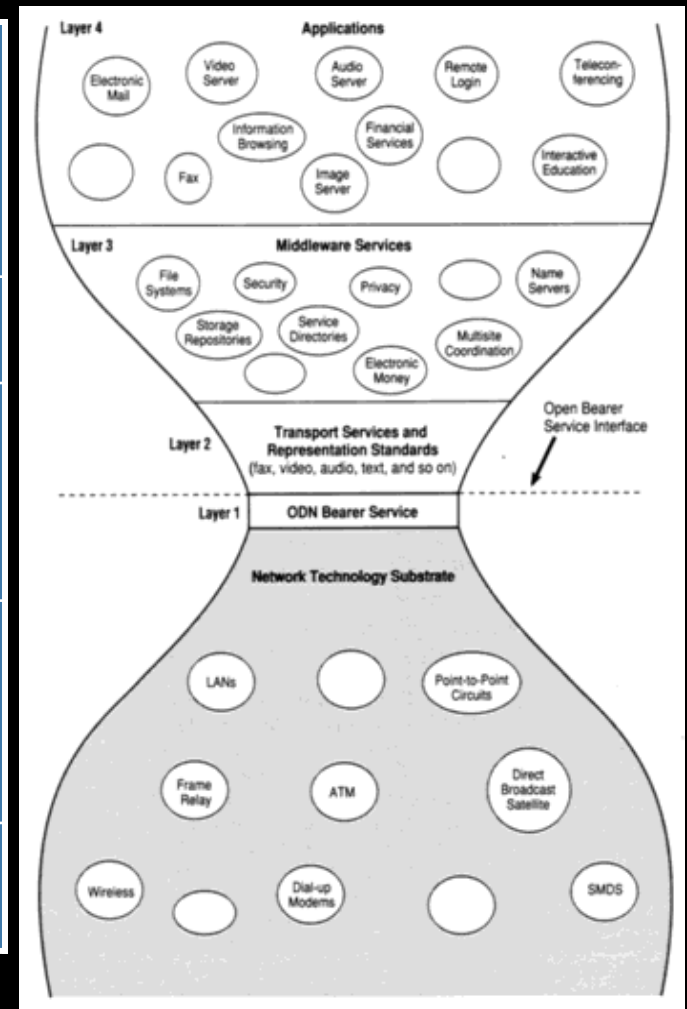
Existing Databases

David Clark's Internet Hourglass

Simple Interface = IFaP Specs

Examples	Identifier Address Reference Name	Format Document Message Container	Protocol Method Operation Process
IP	IP Address	IP Packet	IP Protocol
E-Mail	@ Address	RFC 2822	SMTP (Simple Mail Transfer Protocol)
Web	URI (Uniform Resource Identifier)	HTML (Hypertext Markup Language)	HTTP (Hypertext Transfer Protocol)
WS-*	URI	SOAP Envelope	SOAP Protocol

David Clark's Internet Hourglass

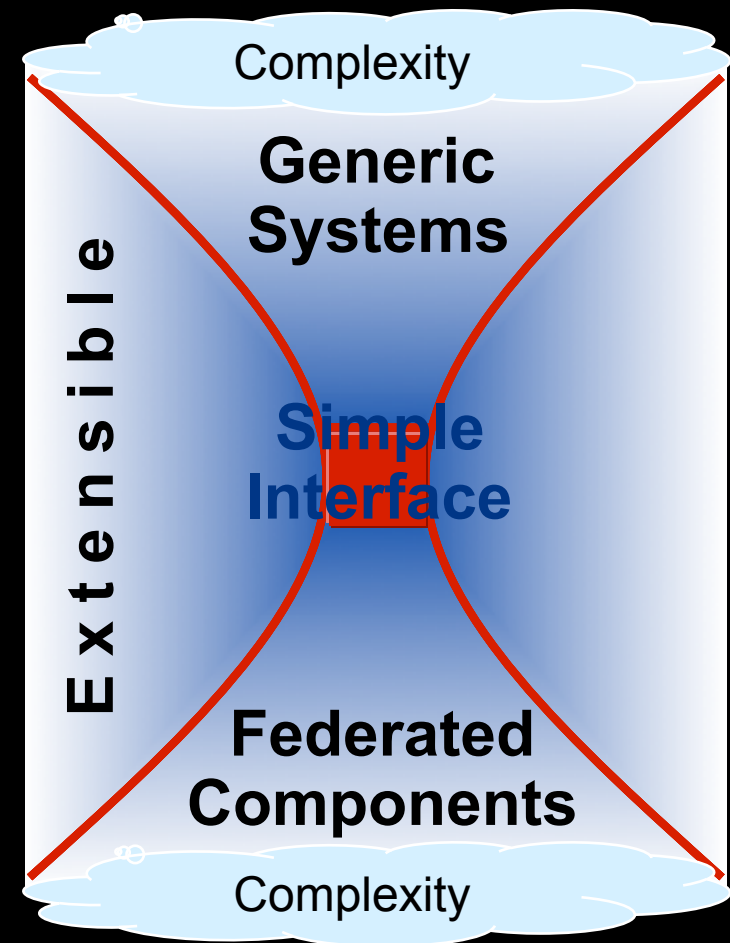


Hourglass Model of Middle-Out Architecture

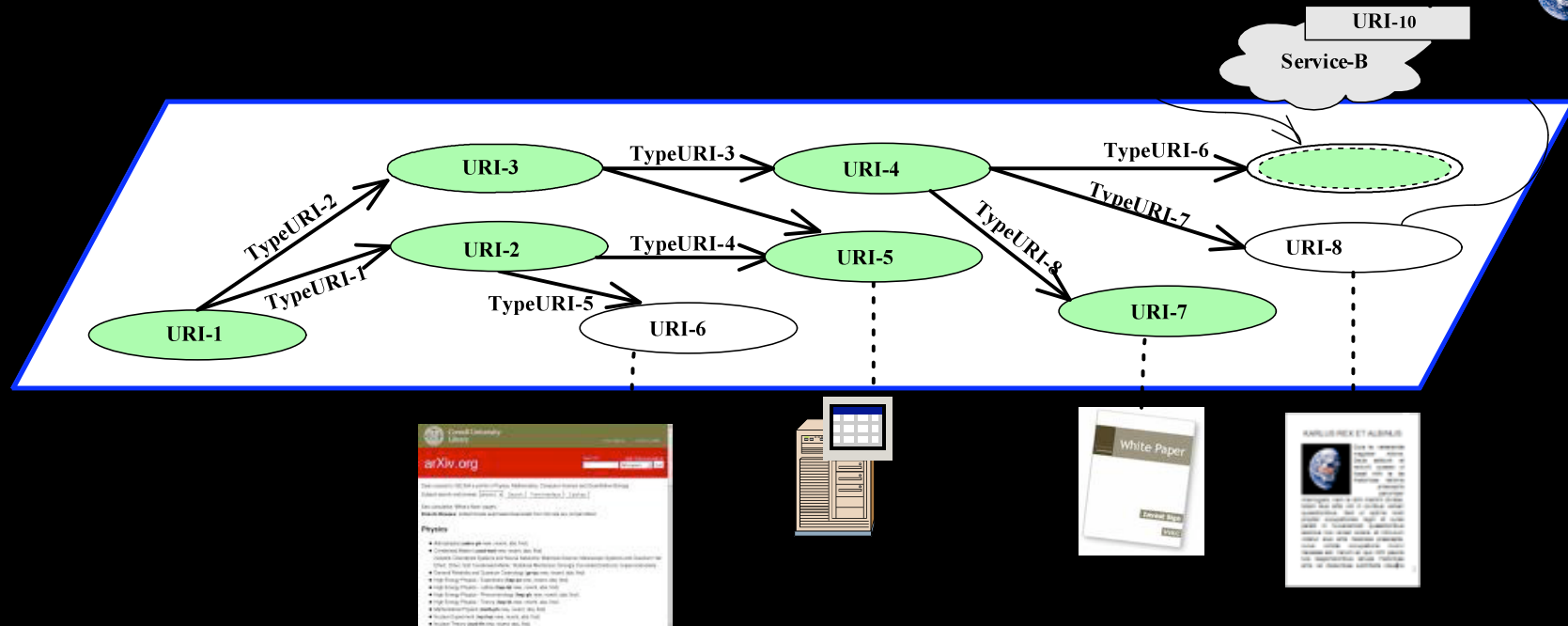
Middle-out is an architectural style that enables decentralized change through the following minimal constraints:

- **Generic Systems:** Underlay a wide range of unexpected uses
- **Simple Interface:** Minimal specification of easily applied identifiers, formats and protocols
- **Federated Components:** Overlay a wide range of unexpected implementations
- **Extensible:** Easy and dynamic forward and backward compatibility

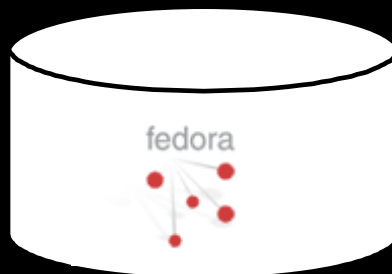
Source: Gartner



Integrated Information Spaces



- Made up of inter-related, content-neutral digital objects
- Separates content, view, and cross-references
- Enables typed relations for all content formats
- Enables third-party annotation and remote resource access
- Persists the content AND the relations (the graph)



Relevant Technology Trends

- Service-oriented architecture

SOA

- Web 2.0

Web 2.0

- Semantic Web (Web 3.0)

RDF

OWL

OWL-S

Implications of Web 2.0 & Web 3.0

- Key themes
 - Services (not packaged apps)
 - Architecture of participation
 - Remix/transform data sources
 - Harness collective intelligence
- Emergent Behavior
 - Upcoming generations of scholars will have a completely different paradigm and expectations regarding technology
 - Collaborative classification (e.g., flickr)
 - Power of collective intelligence (amazon)
 - Alternative trust models (reputation – ebay; open-source)

So how do we handle people using resources outside the institution?

A Repository is Not Just Storage

- A repository is able to deliver value added services over storage (mediation)
- Content can be stored and delivered using multiple interfaces including the Web, intranet services, Web services, ESB, Internet protocols, ...
- Virtually every method of delivering static content is much faster than using a repository but with less agility
- But repositories are able to deliver complex content and integration loosely coupled services (silo, stovepipe killer for 85% of your data)
- Document, disconnected, and event business integration patterns
- And the Web is about HTML period, end of sentence
- The Web operates largely with a low trust model; What is your trust model?
- The Web uses humans (or very adaptable software) to overcome its unreliable architecture
- But the Web is a ubiquitous, infinitely scalable system
- Upon which an enormous number of applications are developed



So how do repositories best fit into the SOA? Web?

Overlap between Web and Enterprise Paradigms

- Both the Web and Enterprise Content Management are content-driven systems with overlapping needs
 - Content Creation and Capture, Collaboration
 - Content Storage differs on time scale optimizations
 - Content managers – Creation and Collaboration
 - Trusted repositories (archives) – Long term storage, integrity, and preservation
 - Both require information lifecycle management capabilities
 - Support for other services and applications
- Both need a well-defined trust and security model

The Repository Dilemma

- Must implement an Enterprise paradigm core because:
 - Trust Model
 - All repositories have a significant trust requirement
 - Low fault tolerance for repository content custodianship
 - Specifiable fault tolerance for mediation capabilities
 - Create (Ingest), Read, Update, and Delete must be transactional
 - Near ACID semantics
 - Architectural Fit
 - Fit as a component in a SOA
 - Ability to create contextualized, durable data persistence
 - Clustering, High-Availability, Transactions, Messaging, Workflow
 - Federation
- *But must also support the Web paradigm*

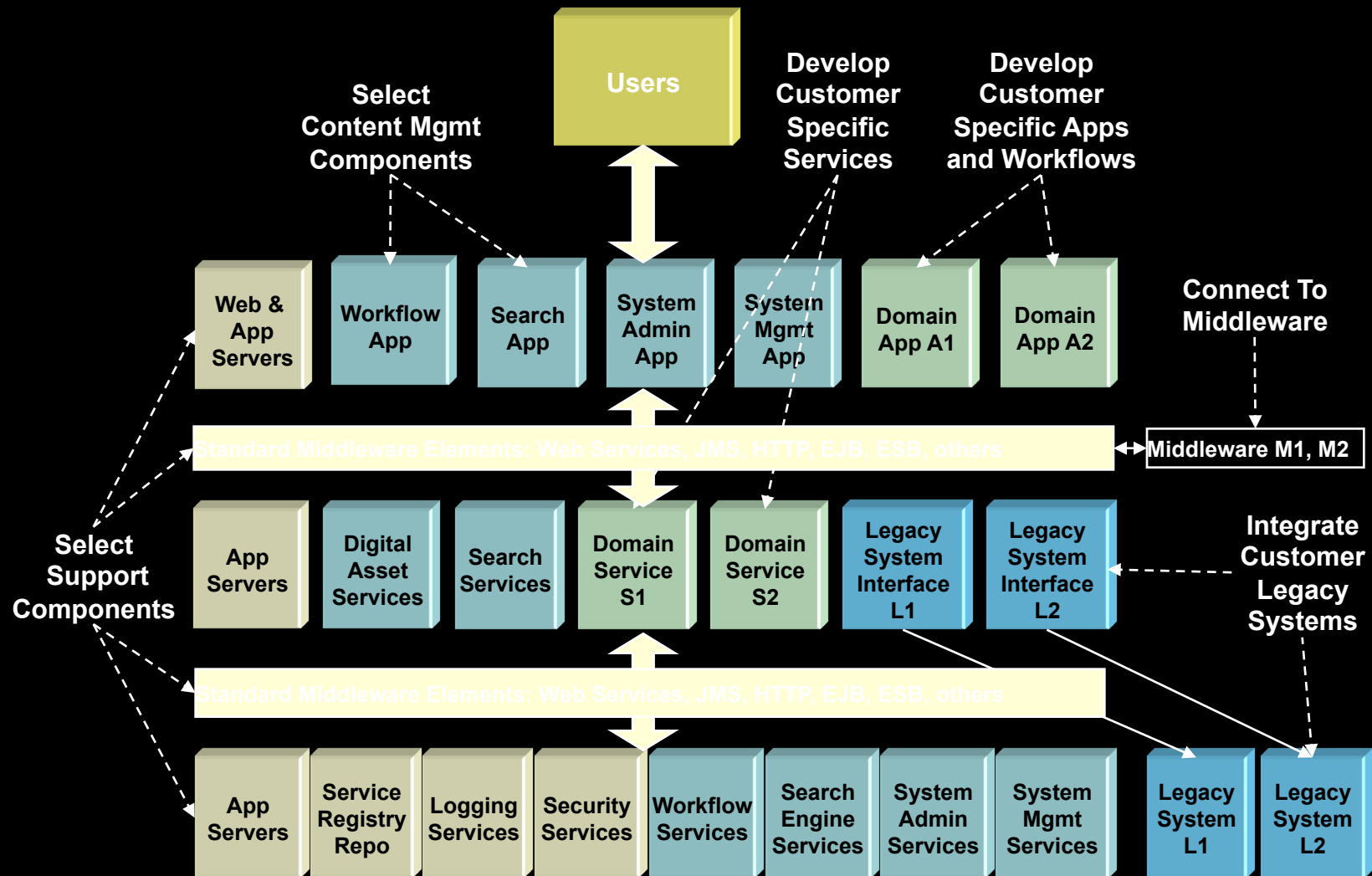
Characteristics of the Enterprise Paradigm



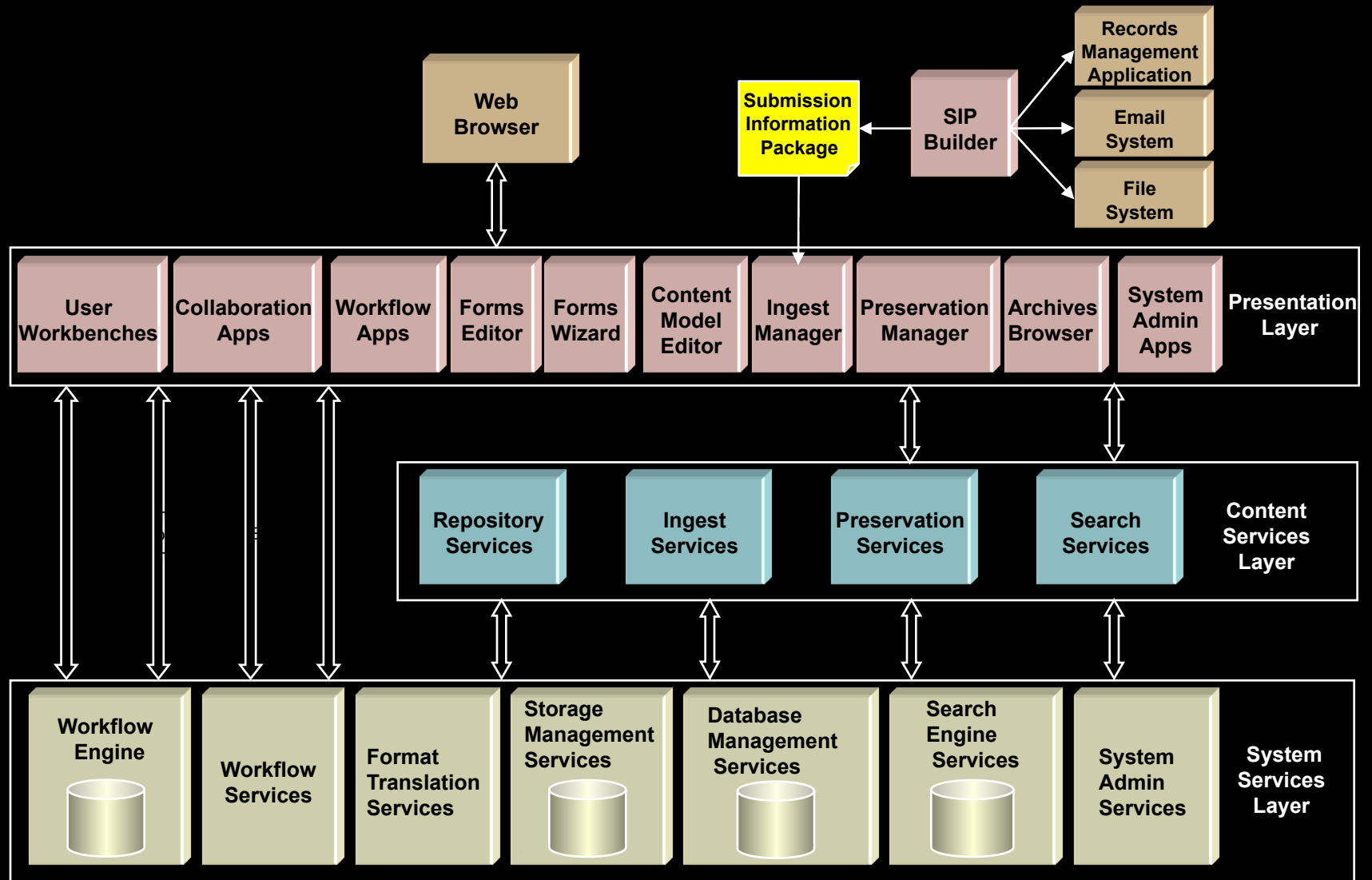
- Enables distributed systems solutions:
 - For mission critical applications and business processes
 - With both technological and organizational complexity
 - Needing support for a high trust and security model
 - Needing support for transactions (2-phase) and messaging (asynchronicity)
- An enterprise system:
 - Requires well-defined integration points & standards to succeed
 - Uses enterprise programming practices having a high learning curve
 - Should be augmented with a well-defined Enterprise Architecture defining business semantics, formats, processes, policies, and standards (a.k.a governance)
- Use of a Service-Oriented Architecture is the preferred approach

So how do we handle people using resources outside the institution?

Notional Software Architecture



Notional Component View



SOA Meets WOA

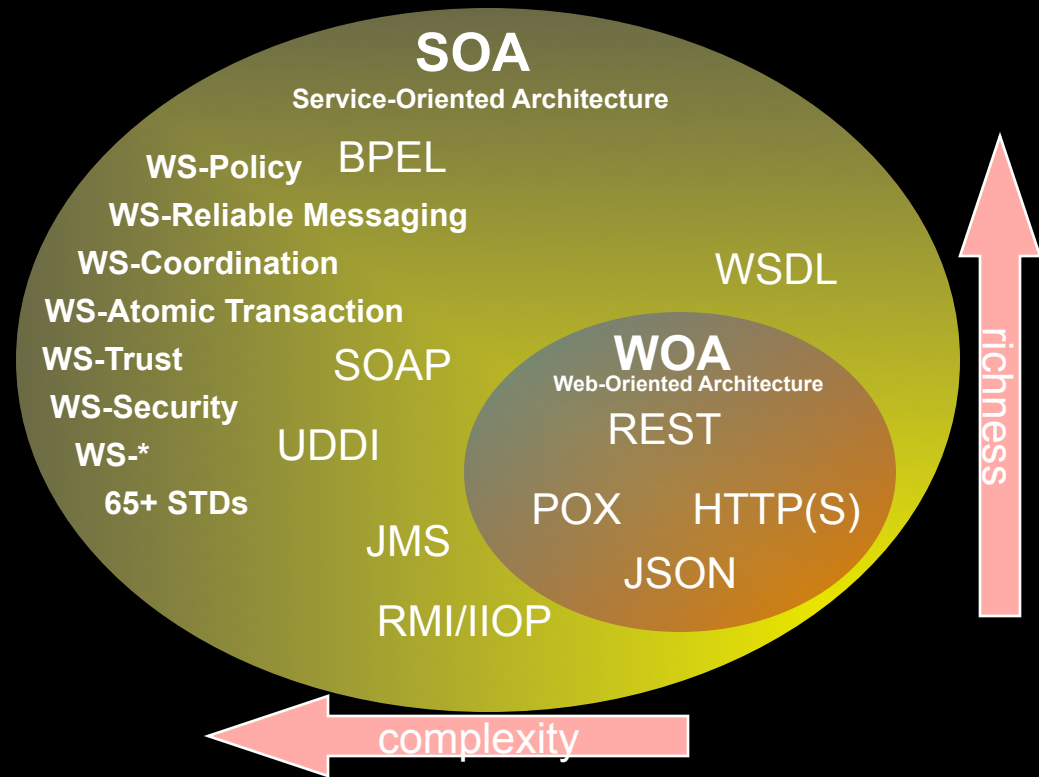
What will the average developer use?

- SOA is:

- + modular
- + distributable
- + shareable
- + loosely coupled

- WOA is:

- + SOA
- + WWW
- + REST

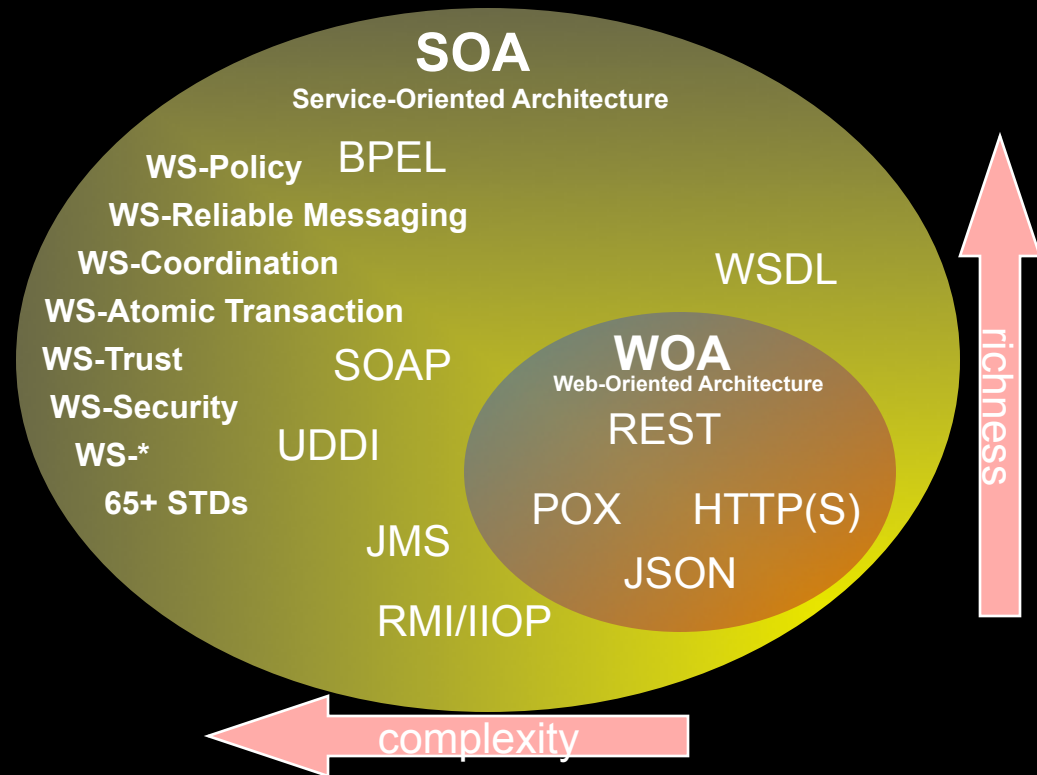


<http://serviceorientation.org>

SOA and WOA Described

<http://blogs.zdnet.com/Hinchcliffe/?p=27>

- WOA is a subset of SOA
- SOA:
 - more trustworthy
 - transactions (2-phase)
 - messaging
 - asynchronicity
 - harder to use
- WOA:
 - fairly trustworthy
 - much easier to use
 - more robust over the Web
 - Lightweight ESB?

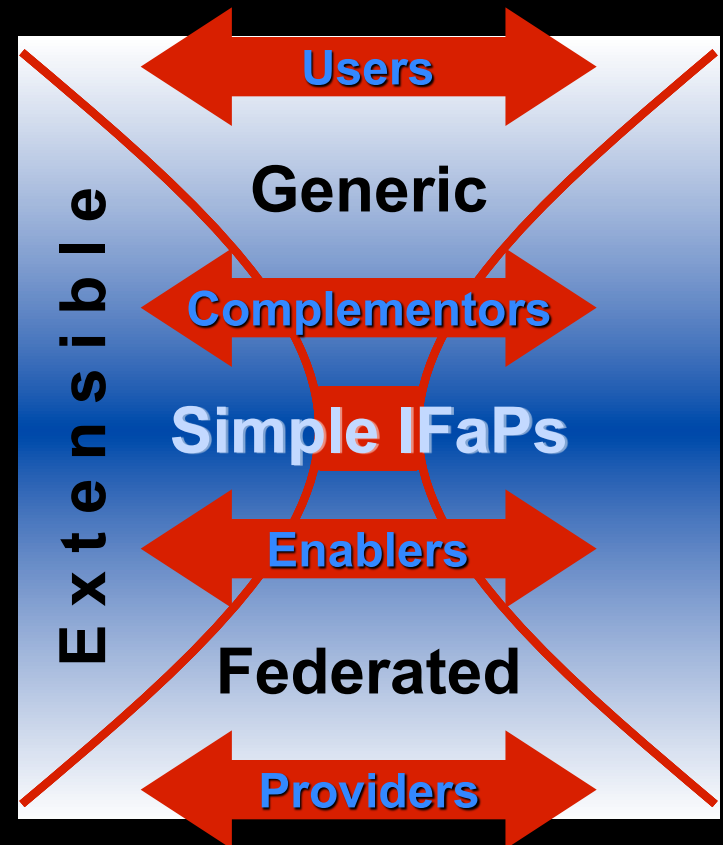


The REST of WOA: Representational State Transfer

- The secret to the Web's dramatic interoperability ("mashability") is its narrow waist: a small number of uniform operations.
- Fundamental Principles:
 - Universal identification of resources: URI
 - Manipulation of resources through representations
 - Self-descriptive messages and uniform intermediary processing model
 - Hypermedia as the engine of application state

Complexity is the killer?

The Key to a Successful
Spanning Layer:
The Hourglass Model



Fedora Design Goals

- Fedora can potentially bridge paradigms
 - Still maintains Web interfaces
 - Adds a reliable WOA interface
 - Can be incorporated into an Enterprise SOA with transactional and asynchronous messaging functionality plus workflow
- Fedora is potentially the model for the new content server
 - Store anything
 - Act as a Web server
 - Act as a Dissemination Server through WOA
 - Act as a service in an enterprise SOA
 - Add new functionality to be conceived
 - Format, Person, Service, Anything registry
- Store everything needed to run the system



Agility with Managed Trust is Required!

Global and Enterprise Class Repositories

Repository for Integration

Next Gen. of Global Class (Web 2.0 & WOA)

- WOA and basic WS
- Consumer Culture
- Outward-Facing
- Very Loosely Coupled
- Security: Assume Everything Is a Threat



Next Gen. of Enterprise Class

- Standards-based (WS-*, JMS, ...)
- Normalized Transactions
- Inward-Facing
- Somewhat Loosely Coupled
- Security: "Lockdown"



Conclusions

- **While Web 2.0 is a major implementation trend among emerging Web-based applications**
- **We can combine the Web, Web 2.0, WOA, and SOA into an integrated system that leverages the best qualities of each**
- **And Web 3.0 uses semantics to help slay complexity (we hope).**
 - Move volatility into data and technology stability into code and content.
 - Empower stakeholders with the freedom to responsibly use, study, copy and change the system.
 - Embrace consumers as an integral part of the application and content development process.
 - Embrace Web mashups as important model to create composite enterprise applications and opportunistic user applications. Most uptake will be in lightweight services.
 - Use SOA to create composite services where high reliability and security is needed.
- **Strong move expected to infrastructure systems due to economy of scale**