SAML-X509-Architecture

A Combined SAML-X.509 Architecture

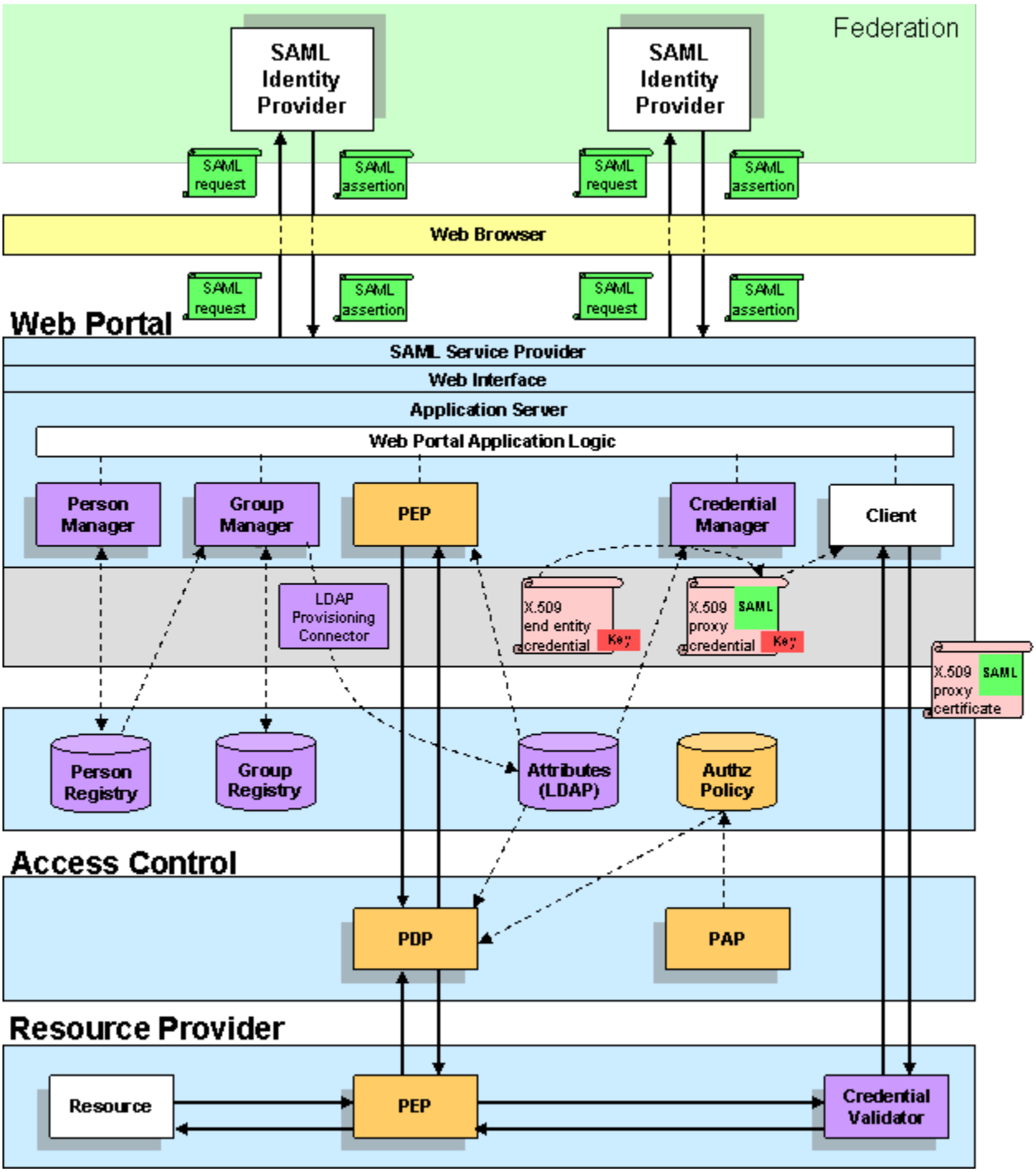
Architectural highlights:

- Lightweight SAML Web Browser SSO on the front channel
- IdP Proxy links accounts and issues local attributes
- X.509-bound SAML authorization on the back channel
- Push technology minimizes heavyweight SOAP-based queries
- Centralized attribute and policy stores

This architecture was inspired by MyVocs, the first VO collaboration environment.

Observations:

- By using a POST binding, SOAP is eliminated on the front channel, which simplifies deployments.
- Federation IdPs do not issue attributes to SPs. Instead, attributes are maintained and resolved locally.
- The SP resolves attributes and issues SAML tokens bound to X.509 certificates. Thus the SP in this architecture is really an IdP Proxy.
- The authentication context from the IdP is bound to the X.509-bound SAML token so relying parties can use it for access control.
- X.509-bound SAML tokens can be used at both the transport level and the message level.
- No new wire protocols are needed. Any implementation that supports WS-Security X.509 Token Profile automatically supports X.509-bound SAML tokens.
- The Credential Manager in this architecture issues X.509 proxy credentials, but short-lived end entity credentials (such as those issued by the Grid Shib CA) are also possible.
- If the PEP and the PDP are co-located at the portal, and the authorization decision is pushed in the certificate, query is totally eliminated on the back channel, which greatly simplifies deployments.



Identity Management

Identity Management components are shown in purple in the diagram.

SAML Web Browser SSO

A browser user issues an authentication request to a SAML Identity Provider (IdP). The IdP authenticates the user and issues a SAML authentication token containing 1) a persistent, non-reassignable identifier, and 2) an authentication context (i.e., a representation of the act of authentication at the IdP). The browser user presents the authentication token to a web portal (or other cyberinfrastructure) protected by a SAML Service Provider (SP). The SP validates the authentication token, consumes the SAML, and populates a local security context with the resulting security information.

Person Manager

The Person Manager (PM) maps the identifier in the authentication token to a local identity. (If the latter does not exist, a local identity is created and the mapping is persisted in the Person Registry.) The mapping is added to the user's security context.

Group Manager

As the user interacts with the portal's cyberinfrastructure (CI), the Group Manager (GM) is called upon to create attributes in the Group Registry. For example, if the user creates a mailing list, the GM creates a membership attribute for the new mailing list and associates this new attribute with the list owner's local identity.

Continuing with the example, the list owner requests that a user be invited to join the mailing list. The CI notifies the user that she has been invited to join the list. If the user accepts the invitation, and is able to present a valid authentication token, the CI associates the list membership attribute with the list member's local identity.

Periodically the GM provisions the group membership information into an LDAP directory that becomes the central attribute store for the CI. This attribute store is made available to other CI components, including the Credential Manager and various subcomponents of the authorization framework.

Credential Manager

The Credential Manager issues an X.509 proxy credential [RFC 3820] signed by its X.509 end entity credential [RFC 3280]. The proxy certificate contains a SAML assertion bound to a non-critical certificate extension. The SAML assertion contains user identity, authentication context and attributes, all used by the relying party for the purposes of access control.

Credential Validator

A client requests a back-end resource and presents the proxy certificate containing the SAML authorization token to a relying party (RP). The RP authenticates the client, consumes the X.509-bound SAML token, and populates a local security context with the resulting security information.

Governance

Governance components are shown in orange in the diagram.

Policy Enforcement Point

A Policy Enforcement Point (PEP) at the resource provider formulates an authorization decision request (SAML or XACML) using the attributes and other information in the security context. The PEP sends this request to a Policy Decision Point.

Alternatively, prior to a client request, an identical PEP at the portal issues an authorization decision request to the PDP. The resulting authorization decision is bound to the proxy certificate and pushed to the RP.

Policy Decision Point

The Policy Decision Point (PDP) combines the information in the request with policy obtained from a central policy store. The PDP renders an access control decision, which is returned to the PEP.

Policy Administration Point

A Policy Administration Point (PAP) maintains authorization policy in a central location. The policy store is made available to the PDP for access control decisions.

Software Mapping

The architecture is realized by integrating off-the-shelf open source software including Shibboleth, Globus Toolkit, and GridShib.

- SAML Identity Provider => Shibboleth Identity Provider
- SAML Service Provider => Shibboleth Service Provider + Apache
- Application Server => Tomcat
- (An implementation of the Person Manager does not exist)
- Group Manager => Grouper
- Credential Manager => GridShib SAML Tools
- Credential Validator => GridShib for Globus Toolkit
- PEP/PDP => any of several open source SAML or XACML implementations, including:
    - Globus Community Authorization Service
    - Globus XACML PDP (beta)