Secure Identity Propagation Using WS-Trust, SAML2, and WS-Security
12 Apr 2011
IBM Impact

Robert C. Broeckelmann Jr., Enterprise Middleware Architect

Ryan Triplett, Middleware Security Architect

Agenda

- Requirements
- Actors
- Related Security concepts
- Specifications
- Technology
    - Websphere 7.0
    - DataPower XI-50
    - TFIM 6.2 Security Token Service
- Flow of identity through the system

Disclaimers

- Here representing ourselves, not our employer.
- What we present here is one of numerous possible ways to use IBM SOA technology.
- Your situation and requirements will probably differ.
- As always, test things in a non-production environment prior to using anything in production.
- We are not responsible for spontaneous combustion of the known universe or any other undesirable outcomes associated with using what is discussed here.

Requirements

Business Requirements-- Hypothetical

- Maintain a high quality customer experience
- Increase business and IT agility and adaptability
- Constantly improve speed to market of new products
- Decrease the growth rate in customer servicing costs
- Provide a stable, scalable platform for payment processing
- Consistent security interface to customers

Technical Requirements-- Hypothetical

- Presentation tier of a customer facing application must be able to access business logic located across different systems within the enterprise.
- Apply industry standards to solve problems whenever possible.
- Will construct wrapper services around legacy systems that are not already SOAP services.
- Standardize on SOAP over HTTP(S)
- All services must satisfy WS-I Basic Interoperability.
- All services are advertised on the ESB; Service Consumer service calls pass through the ESB.

Security Requirements--Hypothetical

- Standards-based, secure end-user identity propagation mechanism
- Mechanism should include a digital signature to ensure message integrity.
- Mutually Authenticated SSL at all hops to ensure message confidentiality (and integrity).
- Token validation and authorization decision made at ESB tier.
- Obtain identity at Service Provider tier(s) via token revalidation.

Actors

Actors

- Service Providers
- Service Consumers
- ESB
- Security Token Service

Service Providers

- Service Provider tiers host WS-I Basic Interoperability-compliant SOAP Web Services.
  - WS-I BP (Basic Profile) 1.1 (see [6])
  - SSBP (Simple SOAP Binding Profile) 1.0 (see [7])
- In our scenario, this is an application running in WebSphere Application Server 7.0.0.7 (or above).
- Application contains POJOs that implement JAX-WS Web Service(s).
  - See [1] for more information about JAX-WS

Service Consumers

- Service Consumer tiers host SOAP clients.
- Again, WS-I Basic Interoperability-compliant (see last slide for description).
- JAX-WS SOAP client.
  - Requires a client stub to be generated by RAD tooling.
- In our scenario, once again, this is a Web Application running in WebSphere Application Server 7.0.0.7(or above).

Security Token Service

- WS-Trust-compliant service(s) that supports issuing, validation, and renewal of security tokens.
- A Security Token contains identification information about a Principal(user of the system).
- We'll talk about Principals shortly.
- More information about these ideas can be found in [2], [3], and [4].

Enterprise Service Bus (ESB)

- SOA pattern
- One possible implementation of a Service Oriented Architecture(SOA).
- A middleware platform. There are many.
- A central access point for reusable, logical components (services) whose use spans multiple spheres of concern.
- For more information, see [5]

Related Security Concepts

Authentication

- Process of a remote entity (user or system) proving its identity to the system.
- Can be achieved in a variety of ways.
- In this story, we will use
  - JEE Security with Form-Based Authentication(WAS 7.0) for end-user authentication.
- Token Validation – confirm a security token is valid and trusted
  - Validating digital signature
  - Checking expiration timestamp
  - Checking user exists in a User Repository
- See [9] for more information.

Principal

- An entity that can be authenticated.
- Could be a system.
  - Batch job.
  - An application.
  - A computer.
- Could be an end user.
  - A Web application user in our case.
- See [8] for more information.

User Repository

- A collection of user information known to the system.
- May include: usernames, passwords, groups, group membership, and other attributes
- Examples
  - LDAP
  - Flat file
  - Database
- Master copy of all user and group information within the system.
- Trust Domain – collection of systems that share a common User Repository
- In our case,
    - LDAP Repository (Tivoli Directory Server, TDS)
    - All systems are in the same Trust Domain
- See [12] for more information.

LDAP

- LDAP—Lightweight Directory Access Protocol.
- A specification.
- Our user repository.
- Contains
  - Inetorgperson objects describing users.
  - Group objects describing groups
  - Users can be members of multiple groups.
- Captures group membership relationships.

Security Token

- A self-contained collection of information that systems can pass around that describes a Principal.
- May contain (we'll assume ours does):
  - User ID.
  - List of Groups.
  - Other attributes(from LDAP).
- May utilize:
  - Encryption
  - Digital signature
  - Timestamp

Authorization

- Process by which the system makes a decision of whether an authenticated principal has permission to access a resource.
- A resource could be:
  - Web Application path (Servlet, JSP, etc)
  - EJB (or EJB method)
  - Web Service
- Will often be based upon:
  - Static information – e.g., LDAP Group membership or a user attribute
  - Dynamic information – e.g., authentication method
- See [10] for more information.

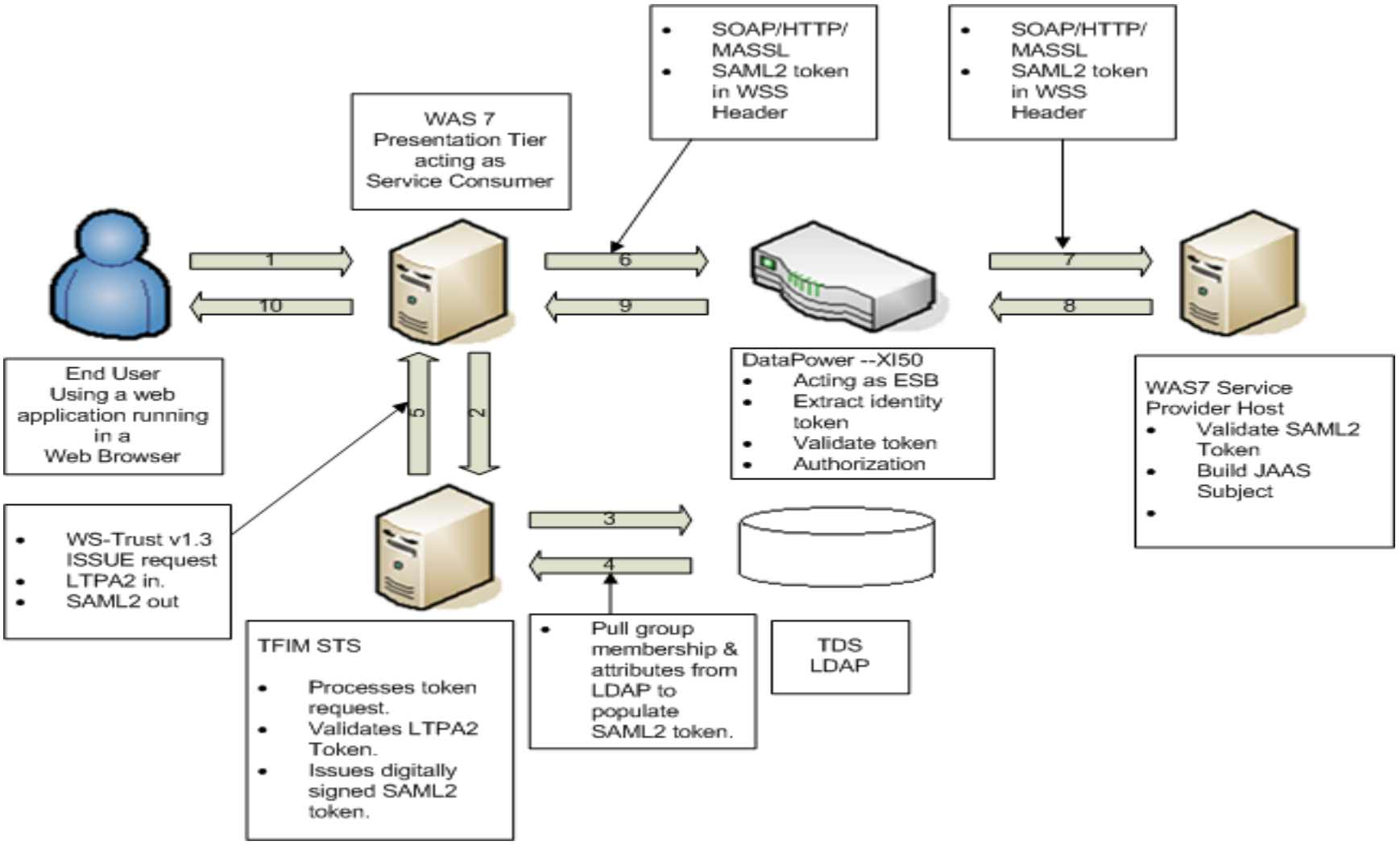
Identity Propagation

- Process by which one system transmits identity of a requestor to another system.
- Identity Propagation usually achieved through some form of token.
  - Token contains username, group membership, other LDAP attributes.
  - Digital signature
  - Confidentiality
- We are using SAML2 tokens in this discussion.

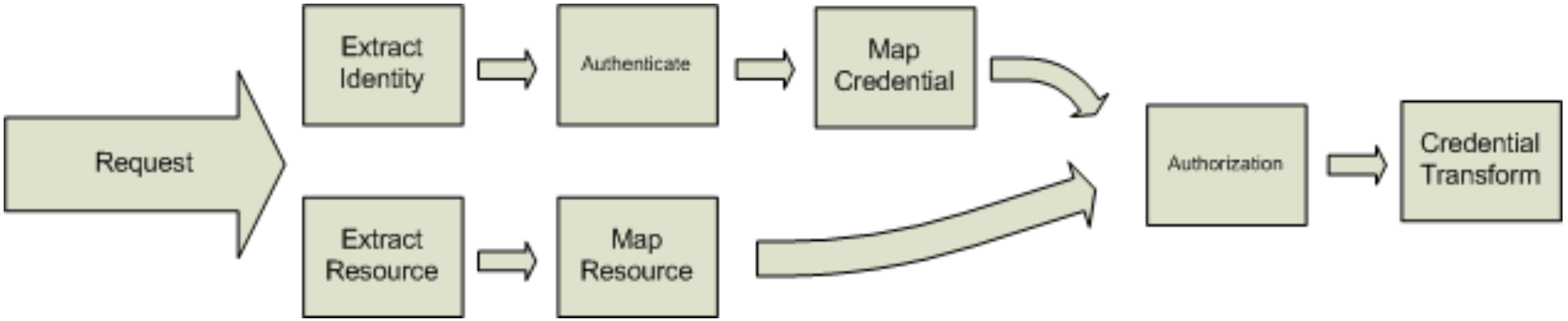
Security Token Service(STS)

- Defined by WS-Trust spec.
- Composed of Web Service(s) that perform operations on Security Tokens.
- Client trusts STS.
  - MASSL
  - Shared key
  - WS-Security
  - Other mechanisms
- Requires a client to provide credentials to prove trust & an identity to be represented in the output token.
  - We'll call these the input credentials.
- Provides assertions about the input credentials in the form of a Security Token.
  - We'll call this the output credential.
- In our example,
  - Input credential is an LTPA2 Token
  - Output credential is a SAML2 token
- Using STS for all token transformations.
  - Central management of digital signature keys/certificates for security tokens.
  - Central management of
    - token generation.
    - token transformations.
- See [11] for more information.

Identity Propagation--Visualized



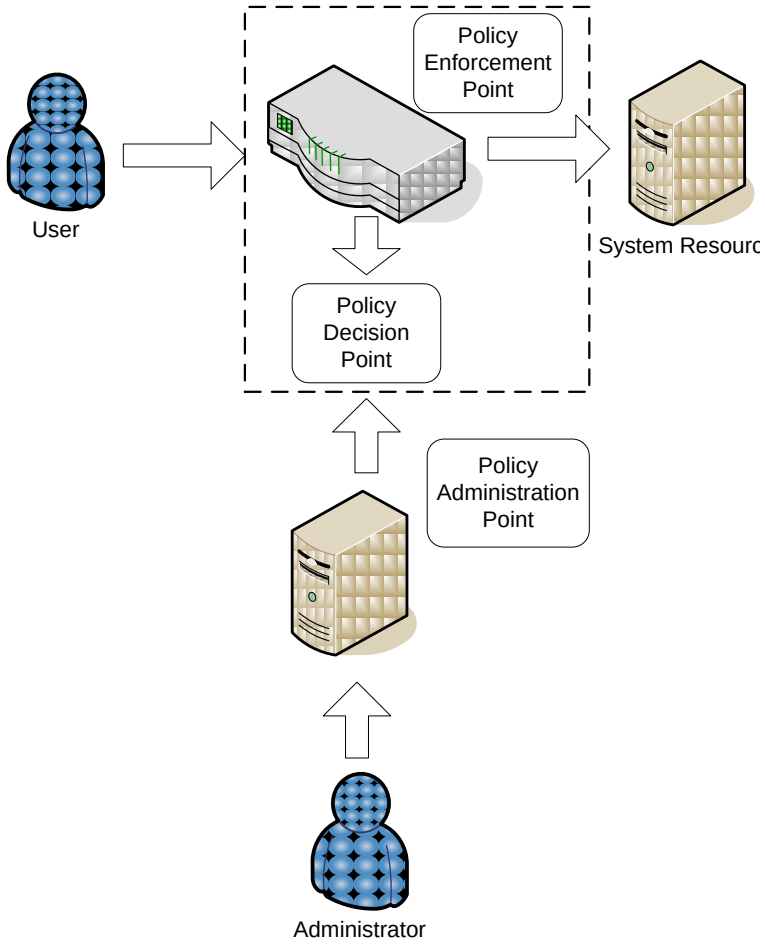
Security Within the ESB



- ESB:
  - Extracts Security Token
  - Validates Security Token
  - Checks Authorization
- We will revisit this shortly.

Policy Driven Security

- **Policy Administration Point (PAP) –** component used for the creation, maintenance, change, and deletion of security policy regarding system resources
- **Policy Decision Point (PDP) –** component responsible for providing a response to an authorization request to a protected resource
- **Policy Enforcement Point (PEP) –** component which manages access to system resources
- See [18]



Mutually Authenticated SSL

- Secure Sockets Layer (SSL) provides transport-layer security between each tier of the system.
- Provides message integrity and confidentiality
- Mutually Authenticated SSL refers to the requirement of the client presenting a valid x509v3 certificate.
- In our case, all communication is over MASSL connections at each network hop.
- Could also use WS-Security Integrity & Confidentiality.
- See [13], [14], [15].

Relevant Specifications

Relevant Specifications

- WS-Security
- WS-Trust
- SAML2

WS-Security

- Provides the basic mechanisms for securing SOAP messages.
- Standard provides for the following to SOAP messages.
  - Integrity (XML Digital Signature with SOAP)
  - Confidentiality(XML Encryption with SOAP)
  - Transmitting identity tokens (SAML2, others)
- See [16].

WS-Trust

- Builds on WS-Security base.
- Provides additional mechanisms for working with security tokens.
- Defines communication with a Security Token Service.
- WS-Trust clients can make the following types of calls:
  - ISSUE
  - VALIDATE
  - RENEW
  - CANCEL
- See [17].

SAML v2

- OASIS standard for exchanging authentication and authorization information.
- Information is propagated as tokens that contain “assertions” about an entity or person.
- Snippet of XML
- WS-Security provides for passing a SAML2 token in a SOAP Header.
- SAML2 spec defines use of XML Digital Signature and XML Encryption with SAML tokens.
- See [19].

Technology

Technology

- Websphere Application Server 7
- TFIM 6.2
  - Security Token Service
- DataPower XI50 (acting as an ESB)

Websphere Application Server(WAS) 7.0

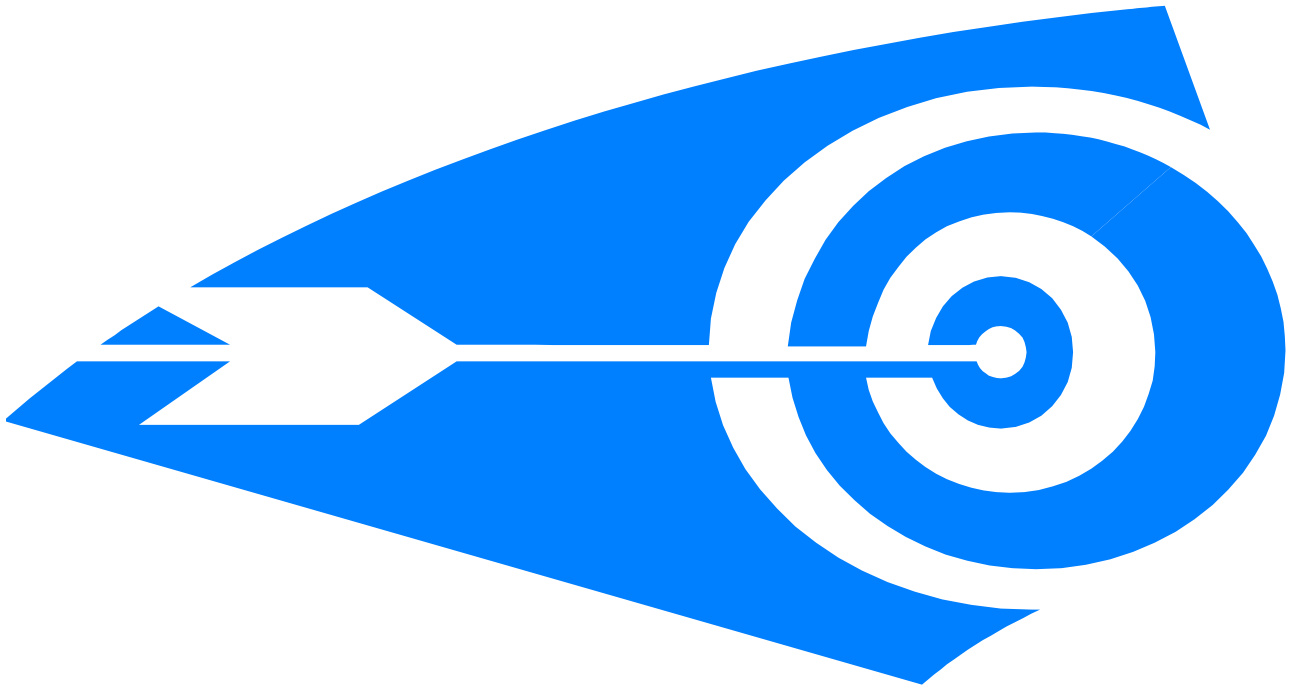
- Description
  - JEE v1.5 compliant Application Server
  - Service Provider Platform
  - Service Consumer Platform
- Using JAX-WS SOAP Runtime.
- Out-of-box functionality (relevant to this discussion) provides
  - WS-Security Support(propagates SAML2 token in SOAP Header)
  - WS-Trust (STS client) support.
  - Local validation of SAML2 tokens(on Service Provider)
  - Dynamic Endpoints can be used to setup client x509v3 cert/key to be used with MASSL connection.
- See [22] for more information.

Tivoli Federated Identity Manager 6.2

- WS-Trust v1.3-compliant Security Token Service(STS)
- Only interested in Security Token Service, but offers solutions for a variety of Federated Single Sign On scenarios.
- Note, token validations will be done locally whenever possible.
- See [21] for more information.

DataPower XI50

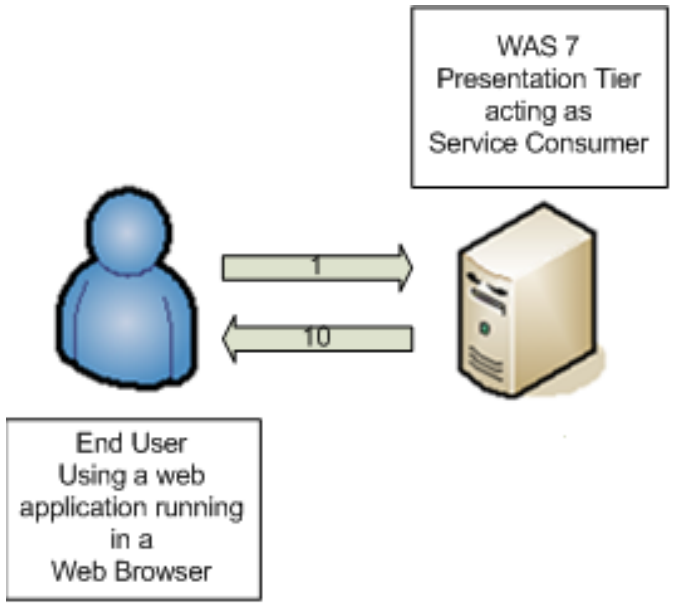
- SOA Appliance
- Acting as an ESB (together with WMQ 7.0).
- Support for:
  - SOAP
  - REST
  - XML (XML Acceleration)
  - WS-* support.
  - Security(authentication, authorization, etc)
  - Many others
- Add on features support:
  - ODBC
  - TAM
  - TIBCO
  - HSM module
- See [20] for more information.



Flow of credentials Through System

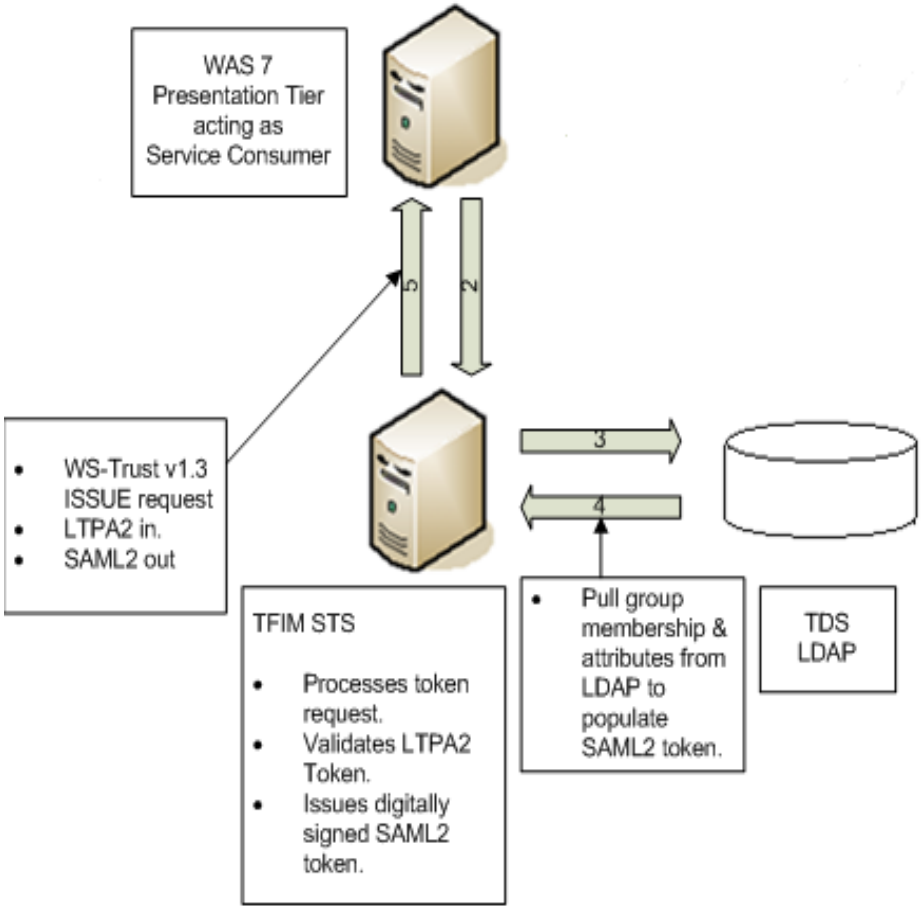
End User Authenticates To System (Service Consumer)

- End user could authenticate to Service Consumer system numerous ways
  - WebSEAL scenario (identity assertion to WAS via LTPA2 token, TAI++, or other methods)
  - JEE Security(Form-Based)—our scenario does this.
  - Other
- Websphere container (Service Consumer) knows the end user by the user session's Security Context.
  - JAAS Subject describes the user's identity



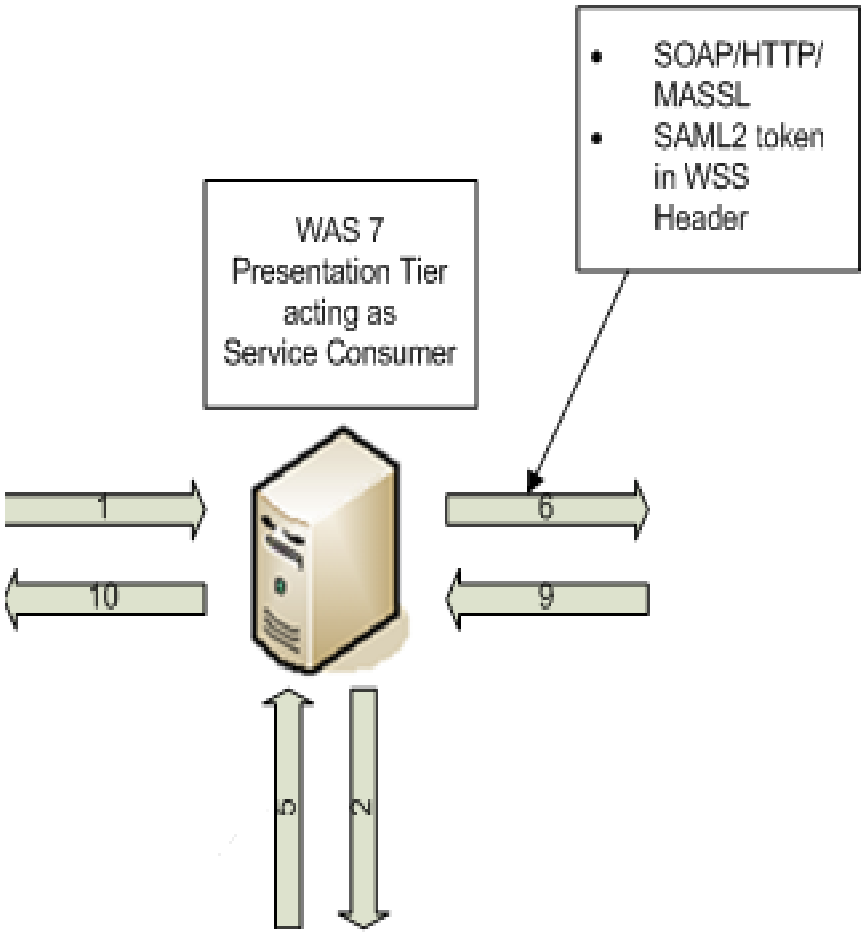
Service Consumer obtains SAML2 token

- WAS7 SOAP/WS-Security runtime interacts with the TFIM STS via a WS-Trust ISSUE request.
  - Input credential: LTPA2 BST
  - Output credential: SAML2token
- SAML2 token is digitally signed by the STS (XML Digital Signature).
- Mutually authenticated SSL for WS-Trust calls.
- Custom module in TFIM STS queries LDAP for user information.
- Token cached locally in Service Consumer WAS container with patch IBM recently created.



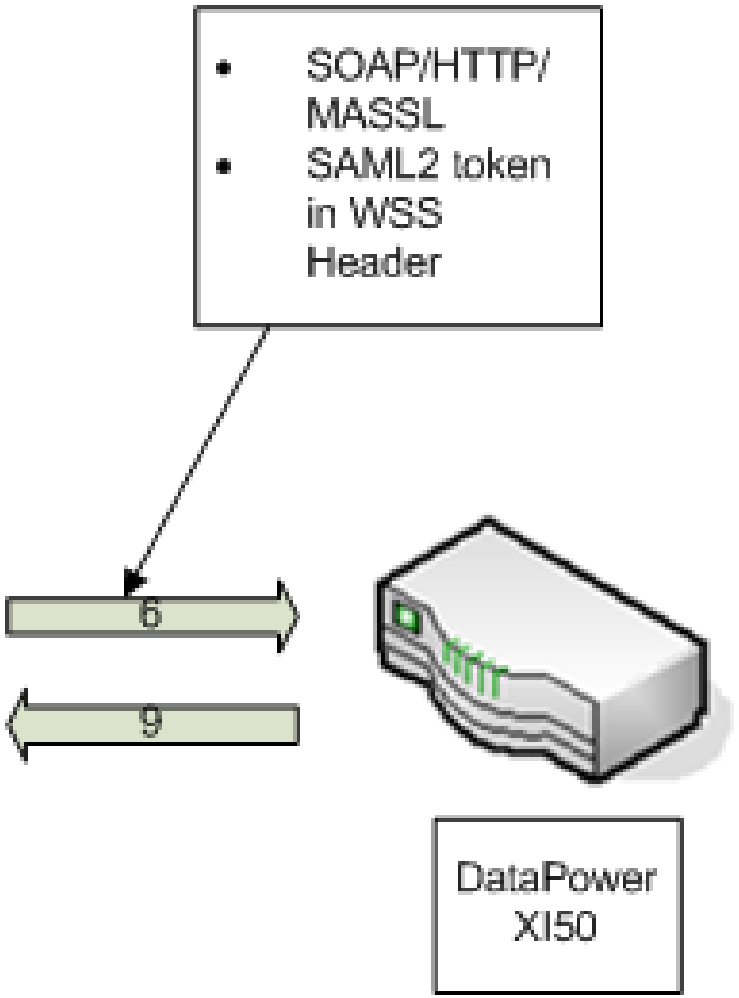
Service Consumer Issues SOAP Call

- Service Consumer uses JAX-WS SOAP Runtime and client stub to issue SOAP call.
- SAML2 token obtained from STS.
- Transport-level security (Mutually Authenticated SSL) for SOAP call.
- WS-Security Runtime injects SAML2 token into WS-Security SOAP Header.
- HTTP POST passes request SOAP message to ESB.
- Wait for response.



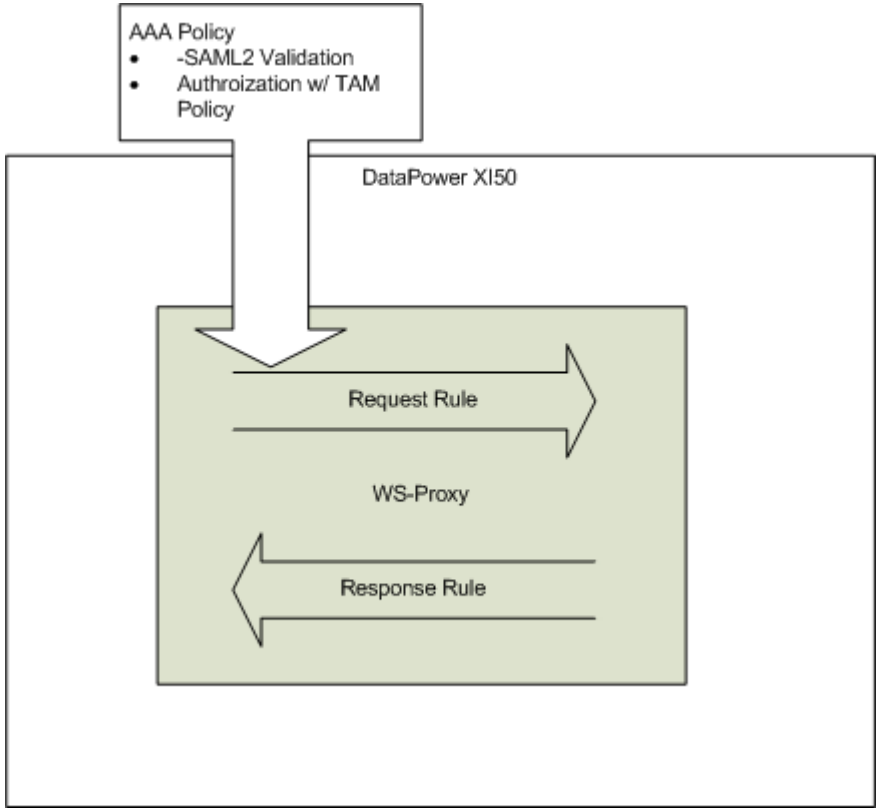
SOAP Request Arrives at DataPower

- HTTPS Front Side Handler advertises service.
- Request is routed to a configured WS-Proxy.



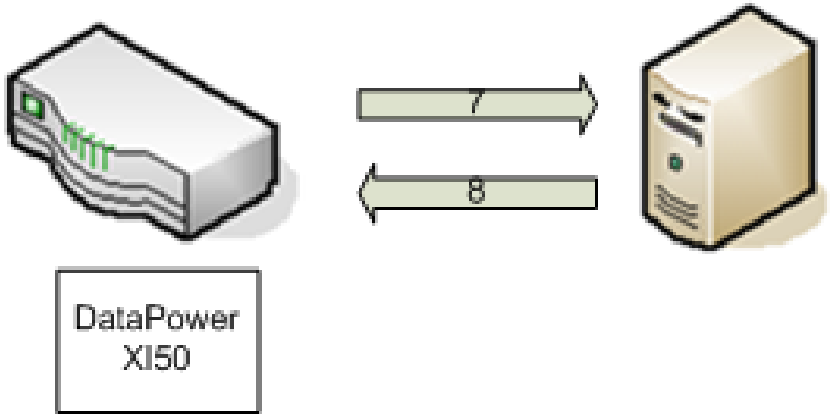
Internal ESB Security Processing

- WS-Proxy has a Request Rule defining policy.
- Service's WS-Proxy Security configuration
  - AAA Policy
    - SAML2 token validation
    - TAM Authorization
  - Exception Handler
  - Audit Logger
  - Other configuration, not relevant to security
- If Service Consumer isn't capable of passing a SAML2 token, DataPower could make a WS-Trust call to TFIM to obtain one or generate token locally.



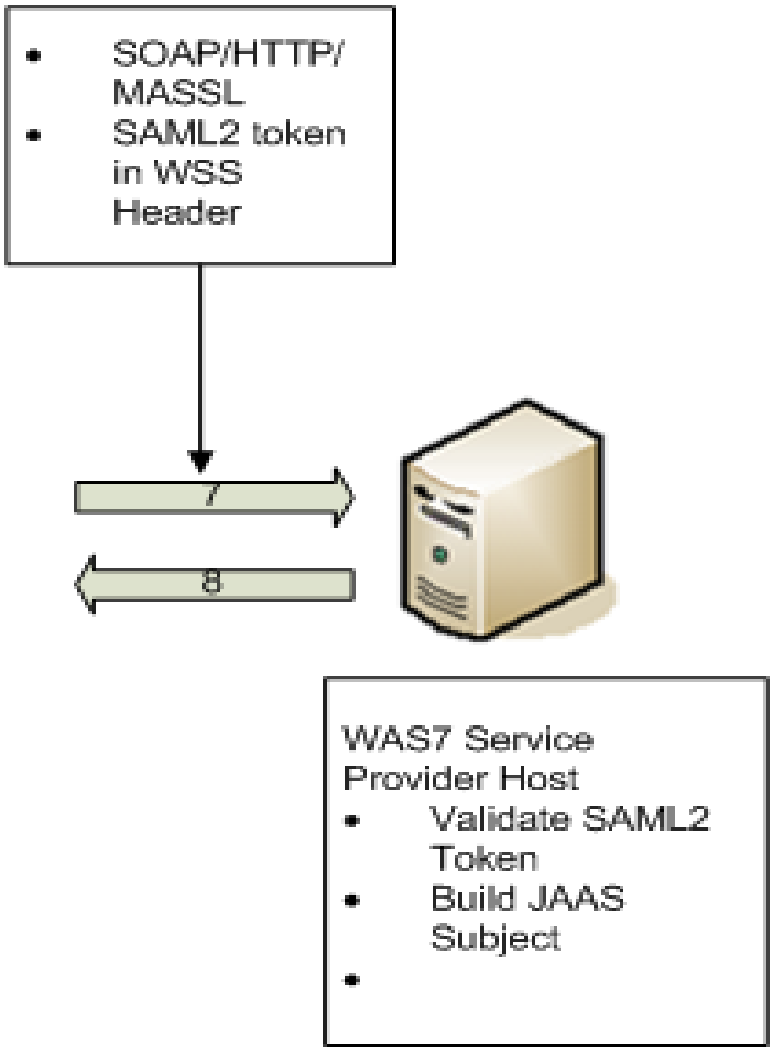
ESB routes request to Service Provider

- WS-Proxy configuration routes SOAP request to Service Provider
  - Could be based upon
    - URL in WSDL.
    - Determined dynamically , based upon message content
- Service Provider Endpoint defined in WSDL stored in Websphere Service Registry & Repository.
- DataPower routes request to this Service Provider Endpoint.



Request arrives at Service Provider

- WAS WS-Security runtime:
  - Extracts token from request message.
  - Validates SAML2 token locally
    - XML Digital Signature validation.
    - Check timestamp.
- Security Context created
  - JAAS Subject
    - Only valid for this one service invocation.
    - Original SAML2 token stored in JAAS Subject.



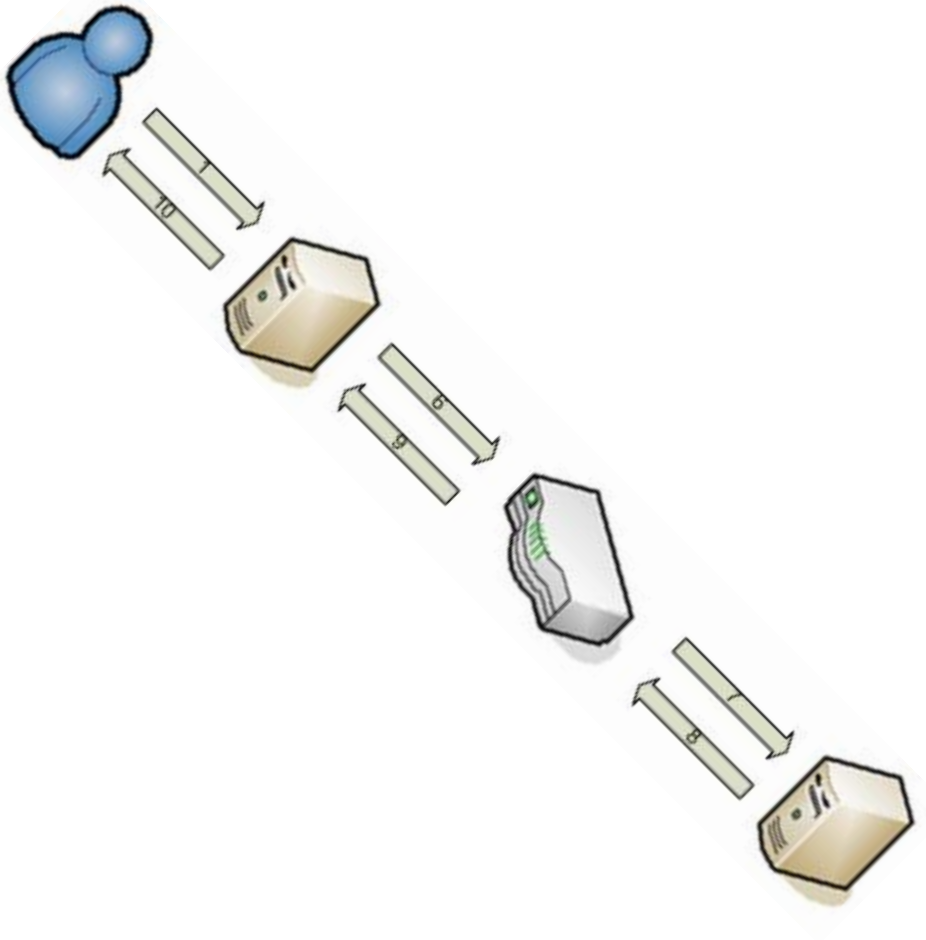
Service Execution

- POJO method exposed as a JAX-WS Web Service operation intercepts SOAP call.
- Service logic is executed.

```
@WebService
public class Echo {
  @WebMethod
   public String echo(String
    str)    {    ...      }
}
```

Response

- SOAP Runtime returns response to ESB.
- ESB intercepts response and passes it to Service Consumer.
- Service Consumer intercepts response.
- For synchronous, request-respond Message Exchange Patterns, identity tokens will generally only be passed in the Consumer->Provider direction.



Third-Party Products & Identity Propagation

- WS-Security, WS-Trust, and SAML2 can be used as the bases for secure identity propagation across compliant platforms.
- For example:
  - JBoss
  - .NET
  - Layer 7

Thank You…

- Questions???

Reference

[1]http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.soafep.multiplatform.doc/info/ae/ae/cwbs_jaxws.html

[2] http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/cwbs_wstruststd.html

[3] http://www.ibm.com/developerworks/library/specification/ws-trust/

[4]http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.iseries.doc/info/iseries/ae/cwbs_sectokenv6.html

[5] http://www.redbooks.ibm.com/redbooks/pdfs/sg246346.pdf

[6] http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[7] http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html

[8] http://en.wikipedia.org/wiki/Security_principal

[9] http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/aes/ae/tsec_pofolo.html

[10] http://publib.boulder.ibm.com/infocenter/wsdatap/v3r8m1/topic/xs40/webapplicationfirewalldevelopersguide132.htm

[11]

[12]

[13[http://www.freesoft.org/CIE/Topics/ssl-draft/3-SPEC.HTM

[14] http://www.ietf.org/rfc/rfc2246.txt

[15] http://www.ietf.org/rfc/rfc2459.txt

[16] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[17] http://docs.oasis-open.org/ws-sx/ws-trust/200512

[18] **http://www.redbooks.ibm.com/redpapers/pdfs/redp4233.pdf**

[19] http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[20] http://publib.boulder.ibm.com/infocenter/wsdatap/v3r8m1/index.jsp?topic=/xi50/welcome.htm

[21] http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc_6.2/welcome.htm

[22] http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/welcome_nd.html