



**System Wide Information Management (SWIM)
eXtensible Markup Language (XML) Gateway
Requirements**

**April 8, 2010
Version 2.2a**

Federal Aviation Administration
800 Independence Avenue SW
Washington, DC 20591

SWIM XML Gateway Requirements

TABLE OF CONTENTS

1.0	BACKGROUND	1
2.0	OPERATIONAL CONCEPT	3
2.1	Operations	3
2.2	XML Gateway Service Descriptions	6
3.0	TECHNICAL REQUIREMENTS.....	9
Appendix A – Desirable Features		A-1
Appendix B – Desirable Performance Features.....		B-1

SWIM XML Gateway Requirements

1.0 BACKGROUND

Today's National Airspace System (NAS) comprises systems that have been developed over time for specific purposes. In general, they are connected discretely to support specific data and information exchange needs. Each of these interfaces is custom designed, developed, managed, and maintained individually at a significant cost to the Federal Aviation Administration (FAA). The Next Generation Air Transportation System (NextGen) relies upon a new decision construct that will bring more data, systems, customers, and service providers into the process. Data will be needed at more places, for more purposes, in a timely manner, and in common formats and structures to ensure consistent use. The resulting decisions must be distributed to the affected parties efficiently and reliably to support timely execution.

The mission of the System Wide Information Management (SWIM) Program is to provide the means for greater information sharing among NAS stakeholders, both FAA and non-FAA users, to support the NextGen operations. This includes, but is not limited to, aeronautical information, flight data, traffic flow management data, surveillance, and weather information. To achieve this mission, the SWIM strategy is to migrate NAS applications toward a loosely-coupled, open, distributed processing environment focused on information sharing (where loosely-coupled systems tend to be highly scalable, robust and agile). These open architecture principles will provide value by reducing costs, reducing risks, enabling new services, and extending and therefore adding value to existing services.

Specifically, SWIM will use Service Oriented Architecture (SOA) principles in designing and specifying NAS Air Traffic Management (ATM) services. Key functional elements of the SWIM SOA are the SWIM Core Capabilities and SWIM Business services. The SWIM Program Office is responsible for definition of the Core Capabilities, whereas the FAA NAS Programs will define the business services in SWIM Segment 1.

As NAS applications are developed using SOA principles and implemented using XML and Web services, the increasing sophistication of these applications stresses the supporting infrastructure. Network traffic increases to support distributed application components; integration with identity and access management systems is required to identify and secure loosely coupled services and other elements; and monitoring, auditing, and control systems are needed to understand and manage dynamically changing application systems. Additionally, using XML can create challenges for SWIM developers and managers alike. Specifically, those challenges include:

- High processing overheads
- The difficulty of managing XML interoperability
- Increased security risk

SWIM XML Gateway Requirements

This document describes the requirements for an XML Gateway that could help to ensure that the opportunities and benefits of SOA are achieved. The XML Gateway does this by providing XML infrastructure services to SWIM developers and users. For the purposes of this document, “infrastructure services” represent the supporting capabilities that allow semi-autonomous, loosely coupled elements to operate together.

XML infrastructure services include the following:

- Performance acceleration and optimization
- Security
 - Identity
 - Access control
 - Mask the internal details of the application
 - Protect against denial-of-service attacks
 - Protect message contents via encryption
 - Provide non-repudiation of messages
 - Threat mitigation
 - Data confidentiality and compliance
- Transformation
- Audit
 - Usage
 - Traffic
 - Services
- Service virtualization
- Mediation

An XML gateway offloads XML processing from the applications. This usually accelerates XML processing and increases system throughput. The gateway takes care of the lower-level tasks of XML processing, including XML Schema Validation, XSLT for transformation, and XPath processing.

As the FAA increases the use of XML across the organization, issues of policy management will arise. One way of providing consistent policy enforcement is through the use of an XML Gateway. As a supplement to the SWIM policy servers, the policy management functions provided by an XML gateway allow for policies to be provisioned, versioned, and archived in an organized and consistent manner.

SWIM XML Gateway Requirements

Finally, security is a major issue when XML is used. The use of XML in SWIM may introduce new security risks. It is possible for an attacker to craft an XML message which will cause denial-of-service to an XML-processing application, due to the processing overhead which XML brings. The FAA must protect against these threats.

An XML gateway server may act as a security proxy for the Application Servers and Services Containers by performing the security-related functionality in their place. If Application Servers and Services Containers delegate security to an XML gateway they should ensure all Web Services transactions go through the XML gateway and that there is no Web Services communications path to the application system that would bypass the XML gateway. The Application Server and Service Container should use an authenticated cryptographic-based security association with the XML gateway server.

2.0 OPERATIONAL CONCEPT

2.1 Operations

For SWIM Segment 1, XML Gateways are used only at the service edge, as shown in Figure 2-1. In this configuration XML Gateways provide a number of functions. Among them are,

- Message validation
- Authentication proxying
- Policy enforcement points
- XML acceleration

These functions are described in greater detail in the following sections. As can be seen in Figure 2-1, SIP applications choose whether or not to offload XML processing to a gateway. Should the SIP choose not to do so, the application will connect directly with other gateways or with other applications. Participating SIPs should carefully weigh the advantages to implementing an XML Gateway for their application. The benefits are numerous, and risks are minimal; consequently, the use of an XML Gateway is **STRONGLY** encouraged by FAA. SIPs that choose to use gateways may share them with other SIPs or not, and may choose to implement a high-availability installation via clustering or failover technology.

SWIM XML Gateway Requirements

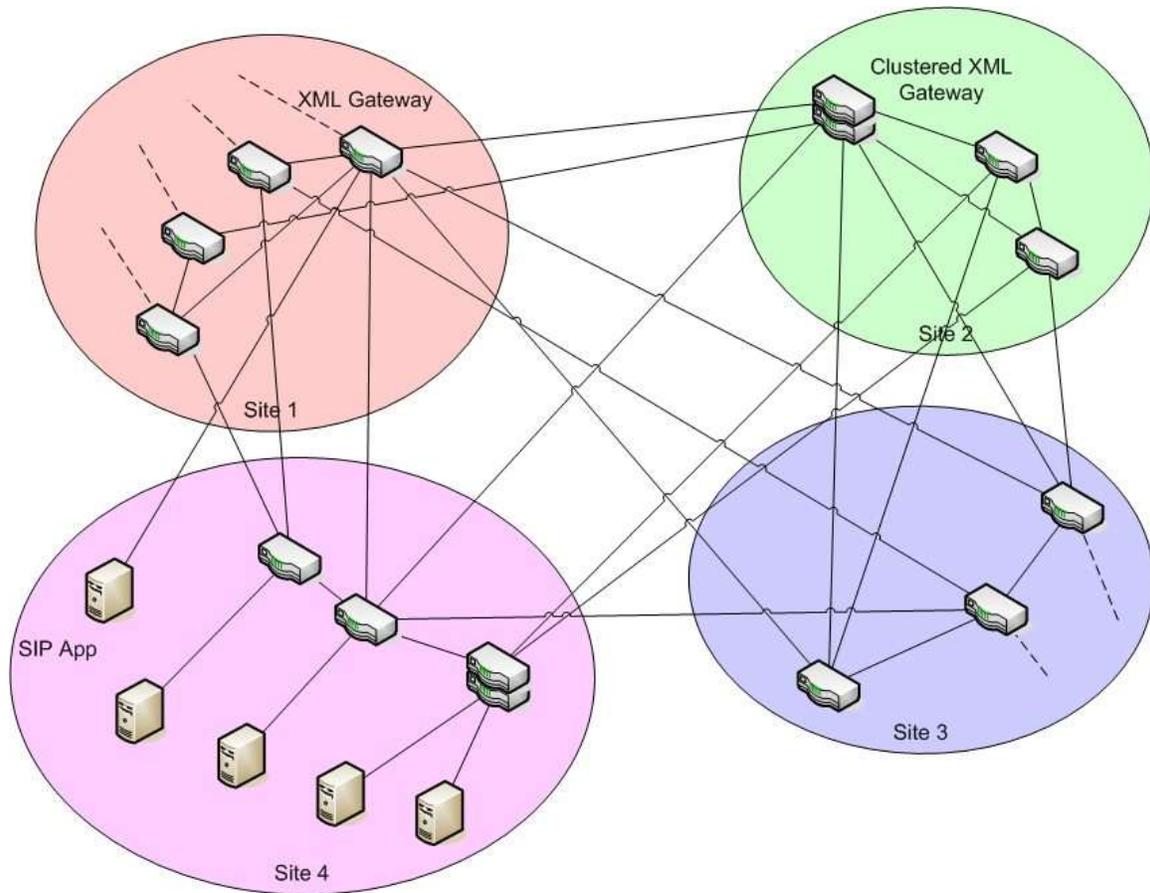


Figure 2-1 Segment 1 XML Gateway Concept of Operations

In future SWIM Segments, shared XML Gateways may be used to consolidate XML traffic within physical sites. In this optional configuration (Shown in Figure 2-2), they communicate with one another to efficiently route XML messages across the SWIM infrastructure, thereby marginally (on the order of one to two percent) reducing the load on FTI. Shared Gateways are also used to perform message enhancement and transformation where necessary and to provide additional policy enforcement and audit points for improved SWIM security. XML Gateways will also play a role in SWIM federation in the near term, since they support queries from multiple databases of authentication and authorization data. SIPs that choose not to participate in this enhanced design will continue to make all necessary connections manually (not shown for clarity).

SWIM XML Gateway Requirements

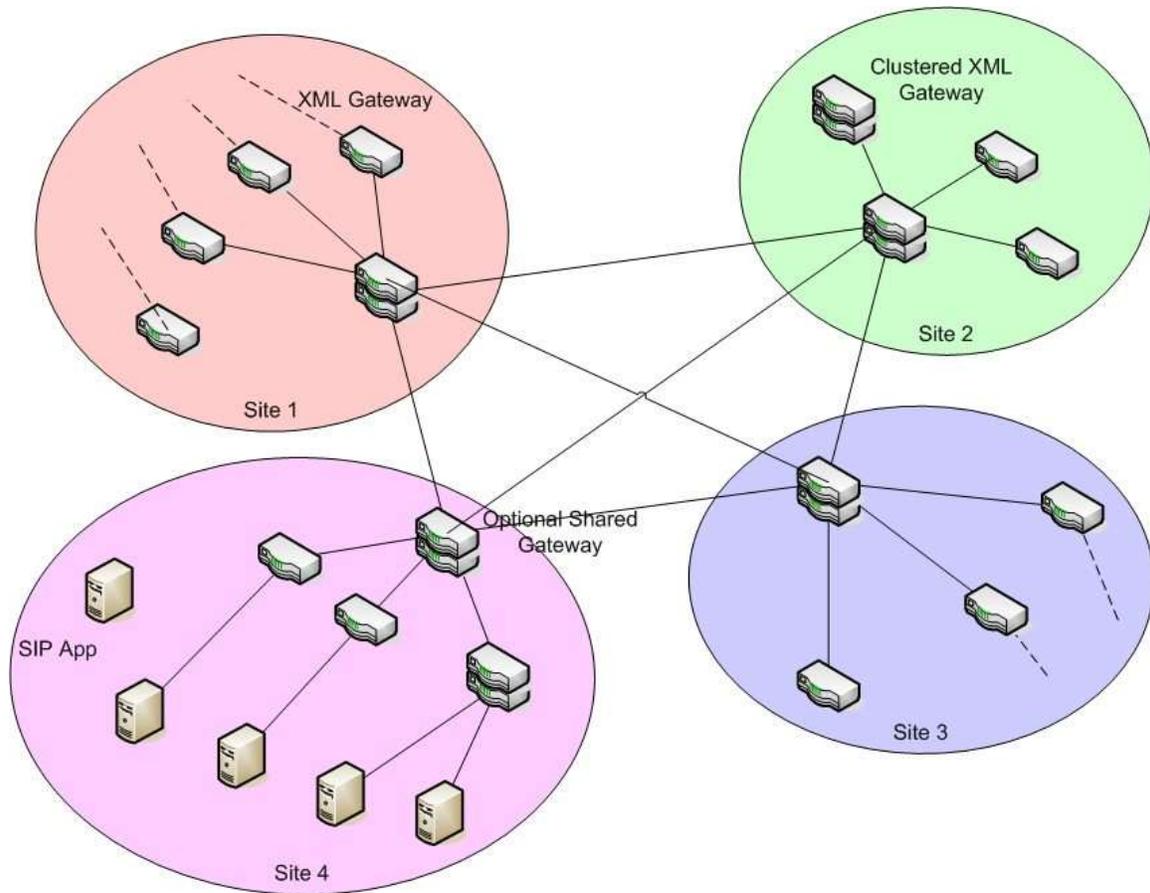


Figure 2-2 Optional XML Gateway Concept of Operations

Finally, an additional hierarchical layer of Gateways may be shared in the “center” of SWIM, to further enhance efficiency. This configuration is shown in Figure 2-3. In this case, the shared gateway (actually an interconnected cluster of gateways, not shown for clarity) spans geographic sites to route messages between sites. These can also be used as “SWIM-wide” policy enforcement points, as well as a point at which to perform testing of QoS and auditing.

SWIM XML Gateway Requirements

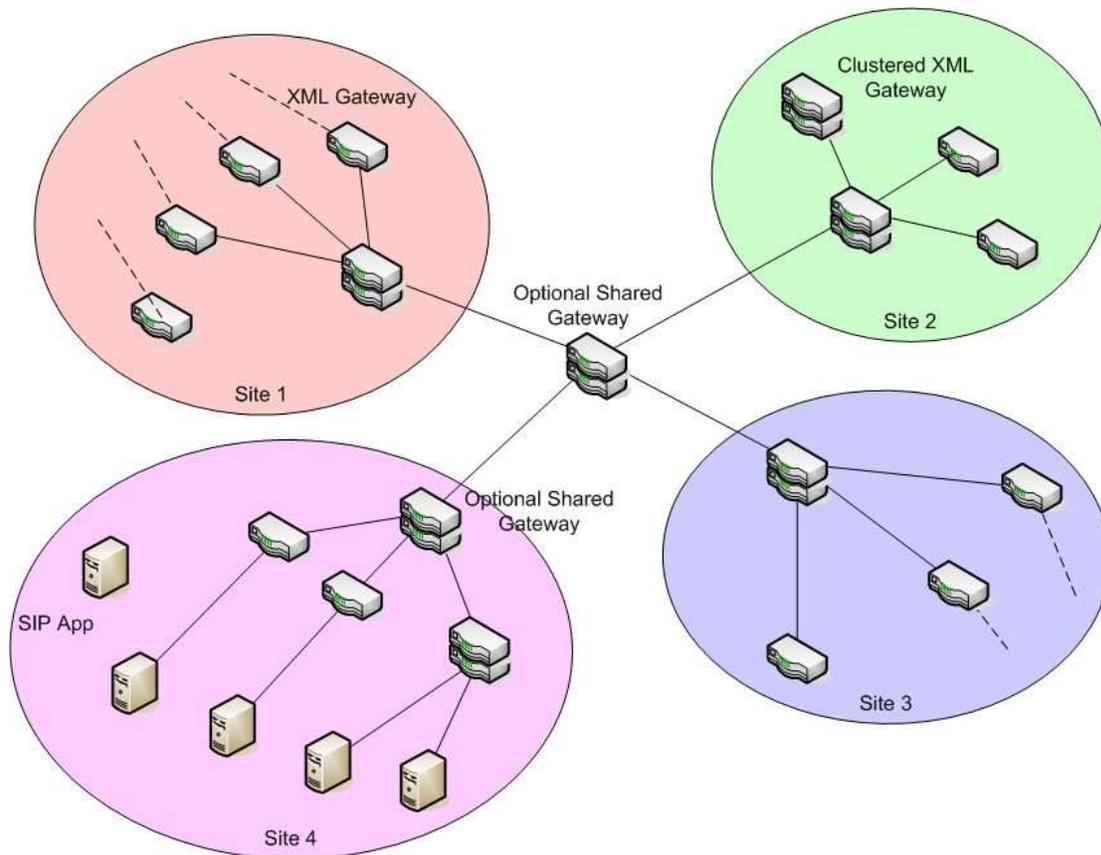


Figure 2-3 Complete XML Gateway Concept of Operations

2.2 XML Gateway Service Descriptions

The following provides a top-level description of services provided by a SWIM XML Gateway. The use of specific services is dependent on the business needs for each service. Every SIP developed NAS service will not necessarily require every Gateway Service or the same capability level. Consequently, Gateway Services will be subject to incremental implementation as requirements and programmatic priorities dictate.

2.2.1 SIP-Consistent Enforcement of Message Level Security

An XML Gateway provides a means for consistent enforcement of policy in SWIM at a granular XML or SOAP (XML message conforming to Web services standards) message level. For example, the XML Gateway will inspect XML message streams for anomalies or known attacks, accelerate XML cryptographic operations, manage access requests, encrypt and decrypt, validate message schemas and signatures, effect credential transformations, and capture message level audit logs. It can also assure standards conformance by enforcing compliance with Web Services standards or ensuring improved interoperability by automatically applying Web Services standards to non-standards based XML messages. By offloading granular XML security operations to the

SWIM XML Gateway Requirements

Gateway, SIPs can expand their security control and flexibility while freeing processor cycles at service endpoints to execute business logic.

2.2.2 Policy Governance

By combining policy enforcement with policy management, the SWIM XML Gateway enables runtime policy governance across SWIM. Governance is a combination of consistent policy definition, execution and conformance validation. At runtime this may mean that every service endpoint responds the same way to a denial-of-service attack or it may mean that precise SLAs can be defined and enforced across a set of services with an automatic response in the event of a violation.

2.2.3 SLA Enforcement of XML-specific functions

Service Level Agreements (SLAs) address situations when compliance to one or more defined service level benchmarks needs to be verified. While XML Gateways cannot support all of the aspects of SLA enforcement, they can measure XML-oriented SLA factors, including XML processing time, messages per hour, rejected transaction counts and queries per day. These measurements are then typically compared by an enforcement process or application to the level desired, the result of which drives an action by SWIM governance bodies. This action can be simply gathering and reporting results, identifying and forwarding SLA violations, or changing service behavior based on current SLA conformance.

2.2.4 Protocol Switching and Reliable Messaging

XML and SOAP messages are transported over many protocols. These transport protocols may include HTTP, HTTPS, FTP, JMS, MQ and others. Mediating XML messages between these transports and middleware protocols is a function that can be handled by the SWIM XML gateway.

2.2.5 Data Translation

One of the most critical functions of the XML gateway in SWIM is the transformation and normalization of data formats. This plays two vital roles: it simplifies interoperation between NAS applications using distinct XML data formats, and it allows for credentials to be transformed and remapped, making cross-application authentication and authorization possible.

Looking forward to future SWIM Segments, advancements will almost certainly be made in the area of the transfer of binary and unstructured data from the SWIM endpoints. In these cases the XML Gateway's role in the data translation process may be reduced or eliminated. The XML Gateway will still offer many other advantages, as listed in previous sections.

SWIM XML Gateway Requirements

2.2.6 Service Virtualization

Another optional feature that the SIP may wish to implement is service virtualization. Service virtualization is the capability of easily personalizing the view of services presented to requestors based on their identity and capabilities. For example, a policy can be defined to generate separate service views based on the identity and associated role of the requesting client application. These identity tailored service views can proscribe different security expectations, standards support, or SLA for each requestor, simplifying the process of service delivery and assuring service reuse. Using service virtualization, SIPs can compose new virtual services based on multiple existing services or discrete operations of a single service. The ability to virtualize services at runtime also simplifies service lifecycle management. In practice, service definitions evolve based on changing business demands, usage policies or standards. The XML Gateway insulates client applications from service changes by ensuring that multiple views of the same underlying service can be maintained simultaneously. The SWIM registry describes multiple instances and versions of a service that may actually only have one instance in the NAS application. This ensures client access isn't broken when service definitions changes. It also guarantees services can be staged before deployment.

2.2.7 Acceleration of XML Processing

XML processing is extremely resource intensive. Parsing, query, and transformation of messages is computationally expensive. The XML Gateway can accelerate performance through specialized parsing software and/or hardware capable of accelerating complex XML operations like XPath, XSLT and Schema Validation

2.2.8 Server Virtualization

XML Gateways can be virtualized by partitioning a single physical platform into a number of logical gateways. Each logical gateway can be configured individually to meet the server load balancing, acceleration, and security requirements of a single application or a cluster of applications. Therefore, each virtualized gateway can consolidate the functions of a number of physical gateways dedicated to the support of single applications (within the limits of the platform processing power). Virtualization adds significantly to the flexibility of SWIM by allowing NAS applications to be easily moved from one physical server to another. For example, with a virtual gateway mapped to a virtual machine, the gateway does not need to be reconfigured when an application is moved or automatically fails over to a new physical machine. Benefits of virtualization include reduced costs through the consolidation of physical devices, higher availability in the event of application failover, and associated savings in management costs and power and cooling costs.

The use of server virtualization does introduce additional complexities (and therefore requirements) with regard to information assurance. These additional requirements are addressed in NIST Special Publication SP800-95 as referenced in the SWIM Governance Policy document.

SWIM XML Gateway Requirements

3.0 TECHNICAL REQUIREMENTS

The mandatory requirements for the SWIM Segment One XML Gateway are provided in Table 3.1, below.

Requirement Text
Standards Support
1. XML Gateway support of Web Services standards and protocols
1.1. The XML Gateway shall comply with the SOAP 1.1 standard
1.2. The XML Gateway shall comply with the SOAP 1.2 standard
1.3. The XML Gateway shall comply with the SOAP with Attachments standard
1.4. The XML Gateway shall comply with the WSDL 1.1 standard
1.5. The XML Gateway shall comply with the WSDL 2.0 standard
1.6. The XML Gateway shall comply with the XPath 2.0 standard
1.7. The XML Gateway shall integrate with UDDI v3
1.8. The XML Gateway shall comply with the XSLT 2.0 standard
1.9. The XML Gateway shall comply with the WS-Addressing 1.0 standards
1.9.1. The XML Gateway shall validate that addresses contained in WS-Addressing headers, including wsa:From, wsa:To, wsa:ReplyTo, and wsa:FaultTo, are valid, reachable addresses.
1.9.2. The XML Gateway shall provide the capability to rewrite addresses contained in WS-Addressing headers, including wsa:From, wsa:To, wsa:ReplyTo, and wsa:FaultTo headers.
1.9.3. The XML Gateway shall provide the capability to enforce the presence of WS-Addressing headers, including wsa:From, wsa:To, wsa:ReplyTo, wsa:FaultTo, wsa:Action, and wsa:MessageID.

SWIM XML Gateway Requirements

Requirement Text
1.9.4. The XML Gateway shall provide the capability to strip WS-Addressing headers.
1.10. The XML Gateway shall comply with the Plain XML (POX) standard(s)
2. XML Gateway support of network transport protocols
2.1. The XML Gateway shall communicate using the HTTP 1.1 network transport protocol
2.2. The XML Gateway shall communicate using the HTTPS network transport protocol
2.3. The XML Gateway shall communicate using FTP
2.4. The XML Gateway shall communicate using JMS
2.4.1. The XML Gateway shall communicate using the Apache ActiveMQ implementation of JMS
2.4.2. The XML Gateway shall communicate using the Fuse Message Broker implementation of JMS
2.5. The XML Gateway shall support all JMS transport protocols that are supported by Apache ActiveMQ and Fuse Message Broker, including but not limited to:
2.5.1. Openwire
2.5.2. HTTP
2.5.3. Stomp
3. The XML Gateway shall support the transfer of binary data as a response to a request.
4. The XML Gateway shall support the use of unstructured data as a response to a request.
5. The XML Gateway shall support the use of non-XML structured text formats as a response to a request .
Networking and Infrastructure Services
6. The XML Gateway shall route based on XML message content except where encryption prevents inspection of the message
7. XML Gateway message routing
7.1. The XML Gateway shall route messages by evaluating XPath expressions

SWIM XML Gateway Requirements

Requirement Text
7.2. The XML Gateway shall route messages by evaluating SOAP Attachments, where present. Note: The evaluation is based on attachment type, attachment size and/or attachment content.
7.3. The XML Gateway shall block messages by evaluating SOAP Attachments, where present
8. The XML Gateway shall enforce minimum and maximum attachment sizes
9. The XML Gateway shall perform schema validation
Policy Control
10. The XML Gateway shall import policies from policy servers
11. The XML Gateway shall export policies to policy servers
12. The XML Gateway shall apply policy obtained from a 3 rd Party Policy server such as AmberPoint , Actional, SOA Manager, etc.
13. The XML Gateway shall transport policies to other policy enforcement points
14. The XML Gateway shall transport policies from other policy enforcement points
15. The XML Gateway shall implement XML policy control via a graphical user interface
16. The XML Gateway shall provide the capability to manually archive policies
17. The XML Gateway shall implement policy provisioning
18. The XML Gateway shall maintain versioning of policies
19. The XML Gateway shall allow the management of policies based on role
20. The XML Gateway shall allow the management of policies based on policy
21. The XML Gateway shall allow wildcard values within policy
22. The XML Gateway shall allow conditional branching within policies
23. The XML Gateway shall allow looping within complex policies.
24. The XML Gateway shall provide the capability to make multiple service calls within a single policy statement.

SWIM XML Gateway Requirements

Requirement Text
Encryption and Signing
25. The XML Gateway shall implement Secure Socket Layer (SSL) encryption
26. The XML Gateway shall implement Transport Layer Security (TLS) encryption
27. The XML Gateway shall comply with WS-I Basic Security Profile
27.1. The XML Gateway shall comply with the W3C XML Signature recommendation (Second Edition)
27.2. The XML Gateway shall implement WS-Security version 1.1
27.3. The XML Gateway shall utilize XML Encryption as defined in the W3C recommendation dated 10 December 2002
27.4. The XML Gateway shall encrypt SSL traffic using AES-128
27.5. The XML Gateway shall encrypt TLS traffic using AES-128
Logging and Audit
28. The XML Gateway shall log all XML activity that transits the gateway to syslog.
Threat Detection

SWIM XML Gateway Requirements

Requirement Text
29. XML Gateway threat detection
29.1. The XML Gateway shall detect threats listed in NIST 800-95 Appendix A
29.2. The XML Gateway shall alert detected threats listed in NIST 800-95 Appendix A
29.3. The XML Gateway shall mitigate threats listed in NIST 800-95 Appendix A
30. The XML Gateway shall provide threat protection from inbound messages.
31. The XML Gateway shall provide white listing threat protection on inbound messages (e.g., schema validation, OGC validation, etc.).
32. The XML Gateway shall provide black listing (signature files) threat protection on inbound messages.
33. XML Gateway threat detection extensibility
33.1. The XML Gateway shall provide an extensible threat detection mechanism.
33.2. The XML Gateway shall provide an extensible threat alerting mechanism.
33.3. The XML Gateway shall provide an extensible threat mitigation mechanism.
Authentication
34. The XML Gateway shall authenticate via HTTP Basic authentication
35. The XML Gateway shall authenticate via WS-Security Username tokens
36. The XML Gateway shall authenticate via WS-Security X.509 certificate tokens
Certificate Management
37. The XML Gateway shall utilize X.509 certificate revocation lists
38. The XML Gateway shall meet or exceed Federal PKI Common Policy Framework V1.7 recommendations
39. The XML Gateway shall utilize XKMS
Authorization

SWIM XML Gateway Requirements

Requirement Text
40. The XML Gateway shall authorize users based on database query
41. The XML Gateway shall authorize users based on role.
42. The XML Gateway shall authorize using content-based authorization
43. The XML Gateway shall be capable of delegating authorization decisions to XACML-based policy decision points in accordance with the XACML standard
Identity Integration
44. XML Gateway interfaces for integrating with systems to maintain identity data
44.1. The XML Gateway shall interface with Microsoft Active Directory for accessing user identity data for purposes of authentication and authorization.
44.2. The XML Gateway shall interface with CA SiteMinder for accessing user identity data for purposes of authentication and authorization.
44.3. The XML Gateway shall interface with RSA Access Manager for accessing user identity data for purposes of authentication and authorization.
44.4. The XML Gateway shall interface with IBM Tivoli Access Manager for accessing user identity data for purposes of authentication and authorization.
44.5. The XML Gateway shall interface with Oracle Identity Manager for accessing user identity data for purposes of authentication and authorization.
44.6. The XML Gateway shall interface with Oracle Access Manager for accessing user identity data for purposes of authentication and authorization.
Alerting
45. The XML Gateway logging subsystem shall synchronize log entries using NTP
46. XML Gateway means of alerting
46.1. The XML Gateway shall provide a capability to alert using email

SWIM XML Gateway Requirements

Requirement Text
46.2. The XML Gateway shall provide a capability to alert using SNMP trap
46.3. The XML Gateway shall provide a capability to alert using a Syslog entry
47. The XML Gateway shall allow the customization of alerts
48. The XML gateway shall allow the full customization of SOAP faults
Security and Identity Mediation
49. The XML Gateway shall perform token mapping (X.509 to SAML, etc)
Transformation
50. The XML Gateway shall perform transformations using XSLT
51. The XML Gateway shall perform transformations using XPath
Confidentiality
52. The XML Gateway shall meet confidentiality standards as defined in NIST 800-21
Availability
53. The XML Gateway shall provide the capability to utilize a clustered architecture
54. The XML Gateway shall detect service outage conditions
55. The XML Gateway shall alarm upon the detection of a service outage condition
Interoperability
56. The XML Gateway shall successfully pass interoperability testing with existing FAA network communications infrastructure and boundary protection mechanisms
56.1. The XML Gateway shall successfully pass interoperability testing with external virus protection devices (e.g., Cisco ASA)
56.2. The XML Gateway shall successfully pass interoperability testing with external proxy devices (e.g., Sidewinder)

SWIM XML Gateway Requirements

The optional features for the SWIM Segment One XML Gateway are provided in Appendix A. Performance Requirements are provided in Appendix B. These features will be considered as product differentiators in trade studies of competing products.

SWIM XML Gateway Desirable Features

Appendix A – Desirable Features

Requirement Text
Standards Support
1. The XML Gateway shall optionally use the following Web Services standards and protocols:
1.1. ebXML
2. The XML Gateway shall optionally use the following network transport protocols:
2.1. MQ
2.2. Tibco RMS
2.3. TCP
2.4. UDP
2.5. Multicast
Networking and Infrastructure Services
3. The XML Gateway shall perform service virtualization
4. The XML Gateway shall perform server virtualization
5. The XML Gateway shall act as a broker for ActiveMQ messages
6. The XML Gateway shall route XML messages based on:
6.1. Identity
6.2. Message Source
6.3. Message format (schema)
6.4. WS-Security token
6.5. Client Authentication method
6.6. SAML authentication method
6.7. Time of Day
6.8. Web Service response time
6.9. Database lookup
6.10. LDAP directory lookup
6.11. Client certificate attributes
6.12. Message size
7. The XML Gateway shall perform protocol conversion between SOAP and Plain XML (POX)
8. The XML Gateway shall perform conversion of REST URIs to
8.1. SOAP messages
8.2. POX messages
9. The XML Gateway shall perform protocol conversion between JMS and

SWIM XML Gateway Desirable Features

Requirement Text
9.1. HTTP
9.2. HTTPS
Policy Control
10. The XML Gateway shall provide for policy chaining
11. The XML Gateway shall facilitate the orderly migration of policies from test to production.
Logging and Audit
12. The XML Gateway shall enable the logging of all XML activity that transits the gateway.
12.1. To SQL database
12.2. To XML database
12.3. To Windows Event log
13. The XML Gateway log shall be searchable based on user criteria
14. The XML Gateway shall enable a method of cryptographically signing a log so that the log integrity is assured.
Authentication
15. The XML Gateway shall utilize SSL mutual authentication
Authorization
16. The XML Gateway shall utilize attribute-based access control
Identity Integration
17. The XML Gateway shall interface with the following systems for accessing user identity data for purposes of authentication and authorization.
17.1. SQL Database
17.2. LDAP Directory
17.3. X.509 certificate
Logging
18. The XML Gateway shall perform logging on a per-service basis
19. The XML Gateway shall perform logging on a per-client basis
20. The XML Gateway shall perform logging on a per-message-type basis
Security and Identity Mediation
21. The XML Gateway shall utilize WS-Trust
22. The XML Gateway shall manage SAML tokens

SWIM XML Gateway Desirable Features

Requirement Text
23. The XML Gateway shall perform conversion between SAML tokens, WS-Security tokens and PKI tokens
24. The XML Gateway shall allow the dynamic creation of WS-Security headers
25. The XML Gateway shall allow the dynamic creation of SAML tokens
SLA Enforcement
26. The XML Gateway shall provide a means for detecting and alerting service outage conditions
27. The XML Gateway shall offer a means for comparing service levels to SLAs and reporting the result.
Availability
28. The XML Gateway shall, at a minimum, meet an availability criteria of 0.9995
29. The XML Gateway shall compare service levels to SLAs.
30. The XML Gateway shall report the comparisons of service levels to SLAs.
31. The XML Gateway shall failover automatically to a clustered Gateway in the event of failure
32. The XML Gateway shall detect a failover condition
33. The XML Gateway shall alert upon a failover condition
34. If the XML Gateway supports hot failover, failover to the standby Gateway shall occur without human intervention
35. The XML Gateway failback mode, if offered, shall be configurable

SWIM XML Gateway Desirable Performance Features

Appendix B – Desirable Performance Features

Performance requirements in an XML Gateway are highly dependent on the specific use cases developed by individual SIPs. Accordingly, they are presented here as “desirable features”, and are to be used as guidance only, until such time as accumulated experience dictates their codification as requirements.

For the purposes of these performance features, “*nominal load*” is defined as follows:

- Up to 10 registered services, being accessed by 50 simultaneous clients via the XML Gateway.
- All message traffic is in the form of SOAP requests and SOAP responses. Message size is to range from 1000 bytes to 30,000 bytes, with an average message size of 5000 bytes.
- All clients generate the next SOAP request immediately on receipt of the previous SOAP response
- Latency is defined as the average increase in the round trip time of a request/response pair divided by 2, measured at the point of request initiation, over a fixed number of requests.

Requirement Text
Performance
1. The XML Gateway shall have a latency of no more than 10 milliseconds when processing the nominal load, with no further processing performed on requests and/or responses.
2. The XML Gateway shall have a latency of no more than 350 milliseconds when processing the nominal load while performing XSLT transformations on requests and/or responses. <i>Note:</i> XSLT transformations include, e.g., changing the SOAP operation of a SOAP request, or transforming a SOAP response to HTML.
3. The XML Gateway shall have a latency of no more than 350 milliseconds when processing the nominal load while performing XML encryption (AES-256) on requests and/or responses.
4. The XML Gateway shall have a latency of no more than 500 milliseconds when processing the nominal load while performing schema validation on requests and responses. <i>Note:</i> Schemas are expected to be lightweight, single reference schemas.