# Web Services Security: Moving up the stack

New specifications improve the WS-Security model

Maryann Hondo (mhondo@us.ibm.com), Sr. Technical Staff Member, IBM
David Melgar (dmelgar@us.ibm.com), Advisory software engineer, IBM
Anthony Nadalin (drsecure@us.ibm.com), Lead Architect, IBM

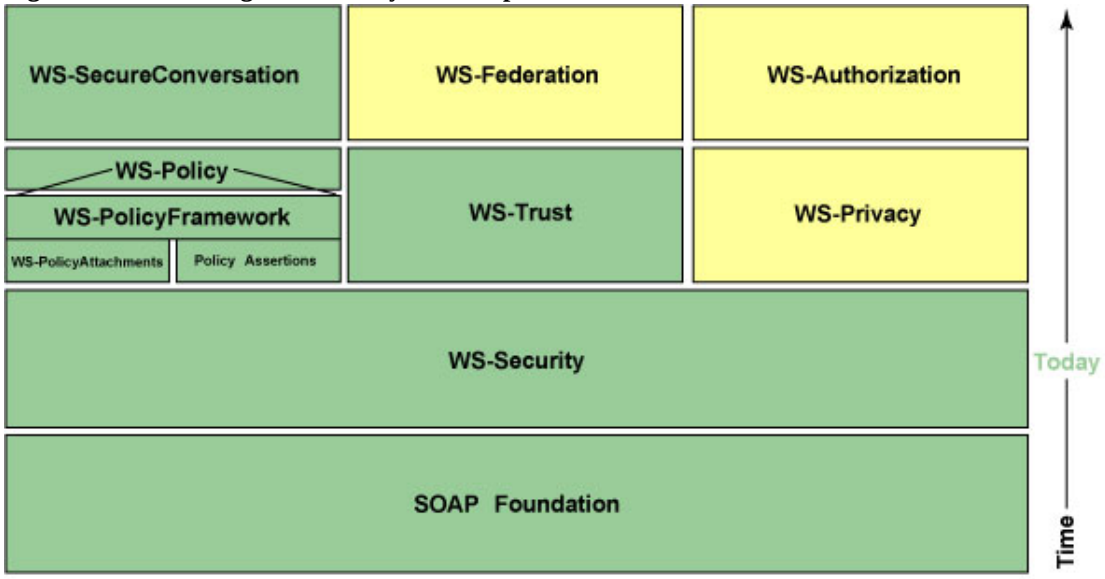
December 2002

In April, IBM, MS, and Verisign jointly published a specification for Web Services Security (WS-Security) that provides a set of mechanisms to help developers of Web services secure SOAP message exchanges. This specification has been accepted by OASIS and a new Web Services Technical Committee (The WSS TC) has been formed to move WS-Security to an open standard. The WS-Security specification has been explained in some detail in an earlier paper, *Security in a Web Services World: A Proposed Architecture and Roadmap* (see Resources).

Additionally in April, IBM and Microsoft provided a roadmap document that included a conceptual stack identifying additional elements that are important to building security into Web services.

The focus of this announcement is the delivery of three more parts of the roadmap; two elements in the policy layer and one in the federation layer (as shown in Figure 1).

**Figure 1. The evolving WS-Security Roadmap**






**Related content:**
Security in a Web Services World
Subscribe to the developerWorks newsletter

**Also in the Web services zone:**
Tutorials
Tools and products
Articles

## Policy

The policy element in the Roadmap labeled WS-Policy has been further refined to include four documents:

- A Policy Framework(WS-Policy) document that defines a grammar for expressing Web services policies.
- A Policy Attachment (WS-Policy-Attachment) document that defines how to attach these policies to Web services.
- A set of general policy assertions (WS-Policy-Assertions).
- A set of security policy assertions (WS-Security Policy) .

The Policy Framework is designed to allow extensibility. Policy is a broad term and encompasses a range of disciplines like security, reliability, transactions, privacy, etc. Similarly, the ability to express policies is not limited to the expression of general policies or security policies. The intent is for the basic policy framework to accommodate the expression of domain specific policy

languages in a way that leverages different domain knowledge within a consistent Web Services Framework.

WS-PolicyAttachment offers several ways to advertise policy assertions with Web services. It builds on the existing WSDL and UDDI specifications but also supports extensibility.

While there will be domain specific policies (for example, security policy) along with common policies for Web services. A common example is a requesting service that may look for a service provider that offers processing in a particular human language (for example, German). The requesting service thus applies a policy assertion, that is, the need for German language support. The provider could also make this assertion by advertising it can offer its service in German. The WS-Policy Assertions Language offers this type of common policy expression. It defines a generic set of policy assertions for Web services.

Security is one domain and to illustrate the expression of security policies, a separate document, **WS-Security-Policy** proposes a language to express policies needed to communicate such policies related to supporting the WS-Security specification.

Trust

In the Web services paradigm the trust between a service requester and a service provider is established through the exchange of information between the two parties in an expected and understood manner. The WS-Security specification already defines the basic mechanisms to securely exchange messages using security tokens. The WS-Trust specification builds on this model by defining how such security tokens are issued and exchanged.

**WS-Trust** starts the work of defining trust relationships by defining a set of interfaces that a secure token service may provide for the issuance, exchange, and validation of security tokens. It is designed to support the creation of multiple security token formats to accommodate a variety of authentication and authorization mechanisms. An issuing security token service takes an input request and typically proof of identity and responds with a token that the named identity has requested (that is, a particular business certification).

The description of this expected behavior within the security space can also be expressed as a *trust policy* and the WS-Policy framework supports trust partners expressing and exchanging their statements of trust.

Secure Conversation

**WS-Secure Conversation** builds on this concept of trust based on security tokens. It describes how security tokens can be used within the context of policy-defined trust relationships to allow multiple service requesters and service providers to securely exchange information over the duration of a conversation. While WS-Trust defines the behavior of overall trust relationships, WS-SecureConversation focuses on defining a security context (security token) for secure communications.

Applying WS-Policy, WS-Trust and WS-SecureConversation

Let’s take the example of a travel agency scenario to illustrate some of these concepts. Acme Travel Service offers its travel services through several different business portals to provide air, hotel and car rental services to its customers. Acme needs to establish different **trust** relationships with its partners through these portals.

Acme would like to offer an integrated set of services to its customers where a requester could submit a single request for hotel, airline and vehicles. Acme also would like the flexibility to extend the services for different partners based on a variety of criteria (gold service, preferred customers, etc). The **policy** for one of its partners, Cars-R-Us might include a security policy requirement for a Cars-R-Us username token and a business application requirement that states the cancellation policy for reservations. The policy for another partner, UnitedCars might include a requirement for UnitedCars Preferred customer numbers.

Since Acme supports different business relationships it needs to be able to determine which travel services to invoke for which customer. How can standards help in automating the trust relationships for Acme to quickly and securely offer integrated travel services as part of the customer’s trusted portal environment?

Acme could assume the registration tasks for all its partners and issue customers an AcmeUsername and Identifier. In this case, Acme provides a veneer for its partners. Before a request is processed, Acme checks the policy for a partner, Cars-R-Us and notifies the user of the cancellation policy, and asks if the request should be processed. Once approved, the request could be augmented (by Acme) with additional security tokens based on this identity, privacy and other business policies (Juser is an employee of xyz company, qualifies for gold service and has a credit limit of US$10,000). Acme needs to have established trusted relationships with the travel service companies and establish which additional tokens need to be supplied with each reservation request.

Alternatively, Acme could just act like a clearinghouse and redirect all requests passing the request from each user on to any partner and let the partner challenge the user for authentication and notify the customer of the policy. While Acme might earn advertising revenue as a clearinghouse, as a travel service it needs to provide value. In this second scenario, Acme could offer a

Security Token Service for its new business partners. Cars-R-Us, who prefers to outsource the management of user information sees the advantage of not doing its own credit-processing and may chose to take advantage of Acmes additional service taking each authenticated request and calling the Security Token Service to retrieve credit approval from Acme.

Credit services might require additional security measures and WS-SecurityPolicy assertions give Acme and Cars-R-Us the ability to express additional security policies that the messages between Acme and Cars-R-Us (that adhere to the WS-Security specification) must provide. For example, Cars-R-Us may require that all credit assertions be digitally signed and contain an expiration time.

Resources

- Security in a Web Services World: A proposed architecture and roadmap provides a detailed explanation of the Web Services Security model.
- Download the Web Services Security specification.
- Download the Web Services Policy Framework specification.
- Download the Web Services Policy Attachments specification.
- Download the Web Services Policy Assertions Language specification.
- Download the Web Services Trust Language specification.
- Download the Web Services Security Policy Language specification.
- Download the Web Services Secure Conversation Language specification.
- IBM offers solutions for secure Business Integration.

About the authors

Maryann Hondo joined Lotus in 1996. She is currently the IBM Web Services Security Standards Lead. Previously, she was the security architect for emerging technology and part of the IBM/Iris Jonah team which provided an IETF PKIX reference implementation. At Lotus she was the security architect for the Lotus Java e-Suite product. Before joining Lotus/IBM her background included working for HP on DCE and PKI based Single SignOn, Digital working on B1/CMW operating system and AT&T Bell Labs working on B2 Unix. She can be reached at mhondo@us.ibm.com.

David Melgar has had a diverse career with a background in the retail industry, systems management, Java technology, and Web services. The original author of UDDI4J, David is a member of the Emerging Technologies division of IBM software group, and is currently focusing on Web services security within the Web Services Toolkit. You can contact David at dmelgar@us.ibm.com.

Anthony Nadalin is the lead architect for the IBM Java Security project. As senior architect, he is responsible for infrastructure design and development across IBM. He serves as the primary security liaison to JavaSoft for security design and development collaboration. You can contact him at drsecure@us.ibm.com.