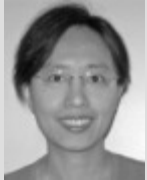


# The SOA Magazine

## Feature Article



### Identity Services for SOA

by Tinny Ng

Published: December 3, 2009 (SOA Magazine Issue XXXIV: November 2009)

[Download PDF](#)

[Digg This](#) • [De.licio.us](#) • [Slashdot](#) • [Technorati](#) • [StumbleUpon](#) • [Google Bookmark](#)

*Abstract: A critical asset that SOA brings to users is the ability to integrate with business partners effectively. It enables a loosely coupled way of linking applications within organizations, across enterprises, and across the Internet. However, the loose coupling of services and their operations across trust boundaries creates challenges in service security. This article, excerpted from her new book [REF-1], discusses how identity services can help service security.*

### Introduction

Service orientation aims to provide services that can be interconnected and reused as appropriate to fulfill a particular business process. These services must be connected and implemented in a secure and auditable manner, according to a defined security policy. However, the loose coupling of services and their operations across trust boundaries create challenges in service security. A number of areas need to be considered regarding service security. One is transaction security. It is essential for services to provide a sufficient level of security to support business transactions. Ensuring the integrity, confidentiality, and security of services through the application of a comprehensive security model is critical, both for organizations and their customers. Another consideration for service security is identity.

### Identity Services

Several forms of inter-organization interaction may occur in a service-oriented deployment. Regardless of the form of the interaction, establishing a trust relationship between the organizations is a key step in allowing inter-organization cooperation. This involves establishing rules around the interaction, such as defining identity information that should be propagated between organizations. It is unlikely that user identities will be the same in all of the service components in a business process flow that spans different organizations.

For example, when service Z of company C in Figure 1.1 receives a request from Tinny Ng, how does it know this Tinny Ng is the one it trusted? If the identity of Tinny Ng has changed, how can company A inform company B and company C about the change? Identity services, therefore, will be required to validate the identity of the requesting users, confirm that they are authorized to perform the requested operation, and map their identity to one that the target service can understand and use to identify the users or services making the request.

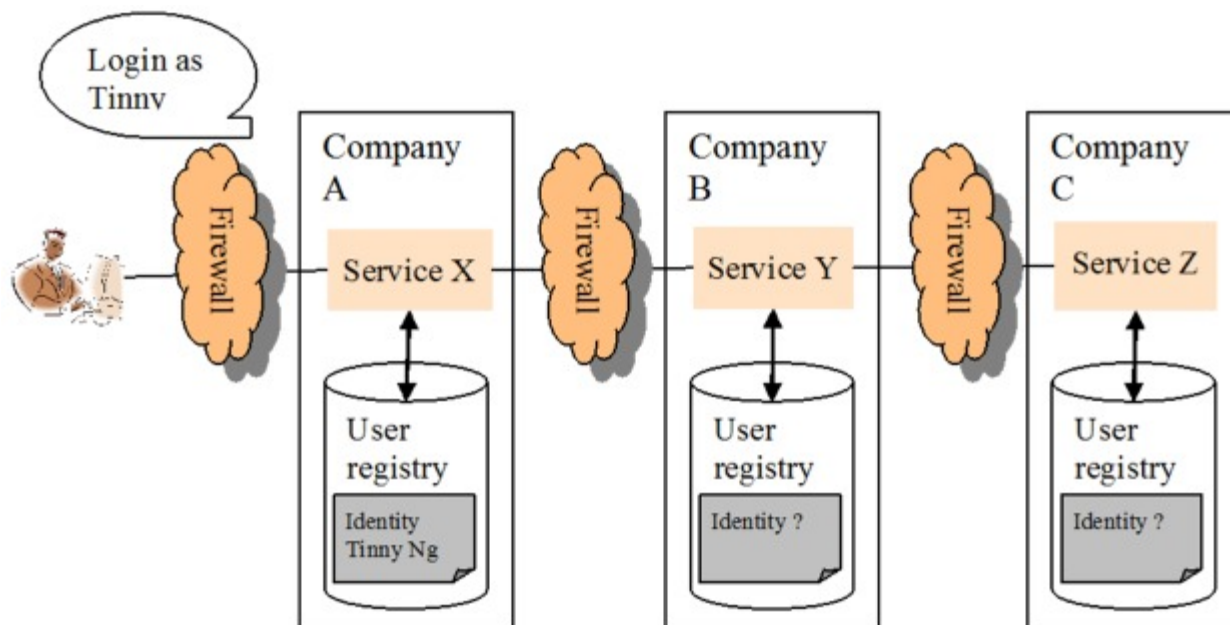


Figure 1.1: Identity propagation in SOA

A security token is like a security-pass or an identity card that you are required to show if you want to enter a restricted access area. Several types of electronic security tokens exist. In many cases, service implementations may restrict the options and formats available for propagating a user's identity to/from the service. In a heterogeneous environment, it is likely that different token types will be supported by different middleware infrastructure components. Identity services are therefore required in the infrastructure, as shown in Figure 1.2, to deal with these identity mediation issues so that services can be easily interconnected without worrying about how to map and propagate user identity from one service to the next.

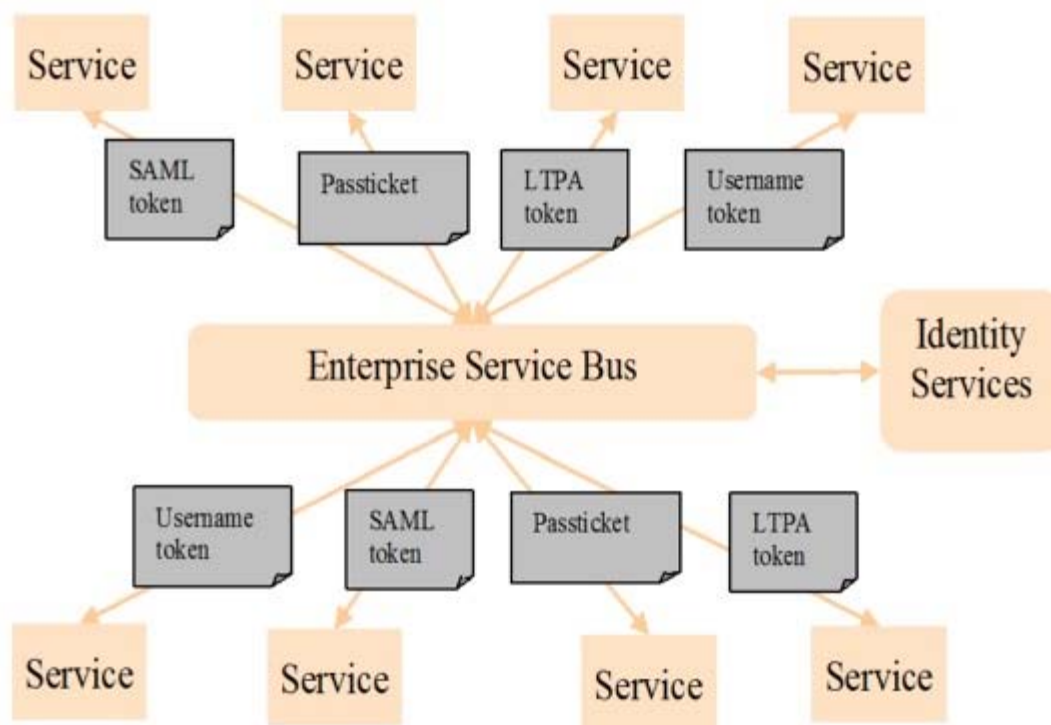


Figure 1.2: Integrating Identity Services with an ESB to support for token transformation

For example, illustrated in Figure 1.3, service requester A, service requester B, and service requester C are making requests for a service from company A. Each of the service requests has associated with it a different type of security token, including a Security Assertion Markup Language (SAML) token, a username token, and an X.509 token, respectively. However, the infrastructure of company A uses a security framework that can only accept a Lightweight Third-Party Authentication (LTPA) security token. As a result, token transformation is required. An identity service can be integrated with an enterprise service bus so that services can be easily interconnected without worrying about how to map and propagate user identity from one service to the next. Mediation flows are to be developed to dynamically transform all these heterogeneous types of security tokens to the one expected by the service provider, which is an LTPA token in this case.

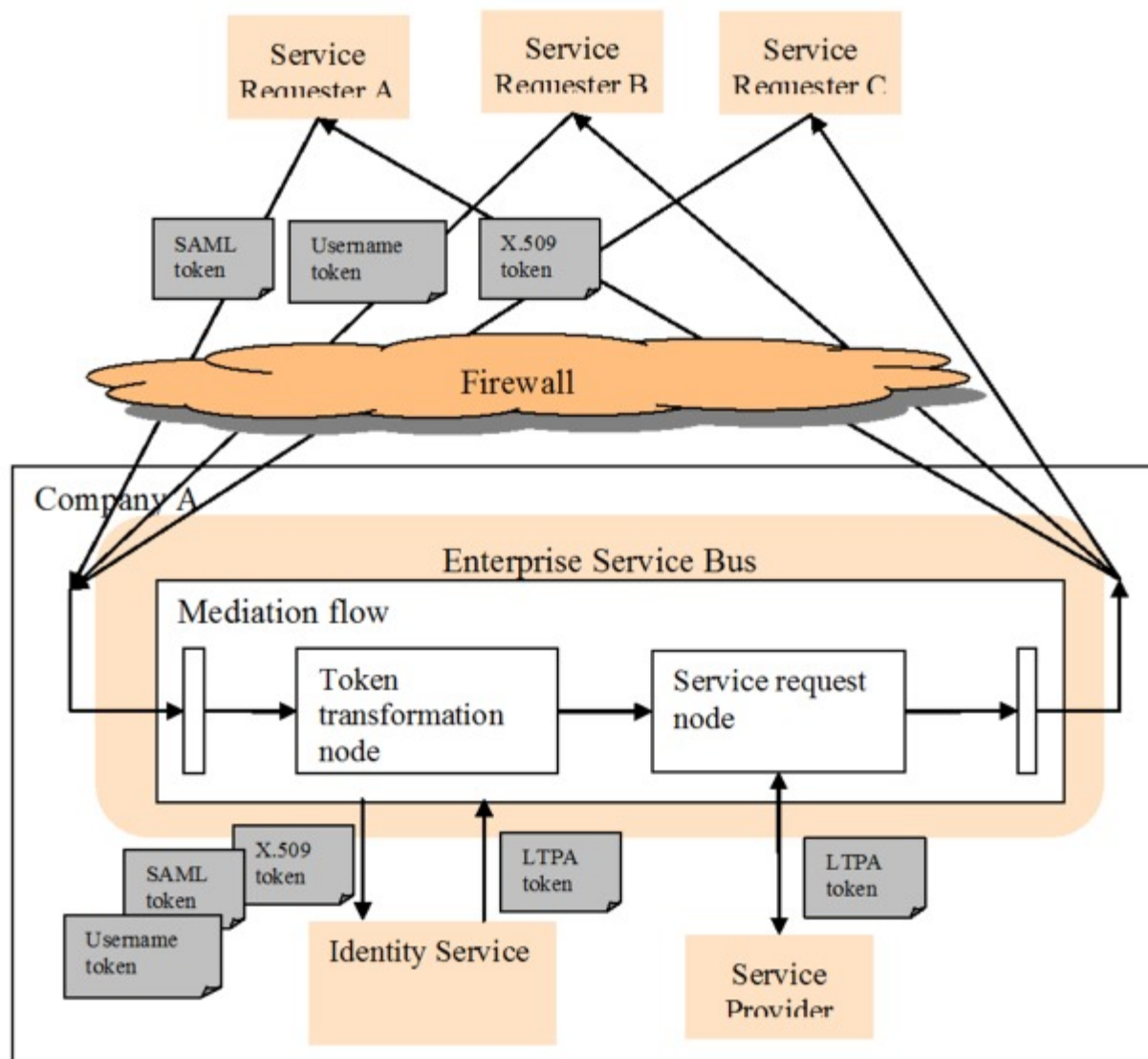


Figure 1.3: Heterogeneous types of security token receiving from service requesters

Identity services in the infrastructure can help increase the responsiveness to new business opportunities by providing security integration with heterogeneous types of service consumers. It also ensures Web services communications are managed in a secured and trusted manner by providing a WS-Trust approach to the management of WS-Security tokens that are sent within Web services requests.

### What are WS-Security and WS-Trust?

WS-Security (Web Services Security) was first released by the Organization for the Advancement of

Structured Information Standards (OASIS) in April 2004. It is a communication protocol that specifies how to provide integrity, confidentiality, and security to Web services messages. It defines how to attach a security token to messages to ensure end-to-end security. Several supported token types can be used for Web services transactions, such as Security Assertion Markup Language (SAML), Lightweight Third-Party Authentication (LTPA), and Username. Listing 1.1 shows a snippet of a Simple Object Access Protocol (SOAP) message that has a SAML security token attached. (SOAP is a protocol specification for applications to exchange information over the Web.)

```
<env:Envelope xmlns:env="...">
  <env:Header>
    <wsse:Security
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <saml:Assertion ...>
        :
      </saml:Assertion>
    </wsse:Security>
  </env:Header>
  <soapenv:Body>
    :
  </soapenv:Body>
</env:Envelope>
```

*Listing 1.1: WS-Security Message Snippet*

WS-Trust (Web Services Trust Language) was approved by OASIS as a standard in March 2007. It provides extensions to WS-Security and specifies how to validate, renew, and issue security tokens. WS-Trust defines the concept of a Security Token Service (STS), a Web service that responds to WS-Trust requests and issues security tokens. Listing 1.2 is a snippet of a WS-Trust message that requests security tokens.

```
<wst:RequestSecurityToken xmlns:wst="...">
  :
  <wst:Issuer xmlns:was="..." xmlns:wst="...">
    <wsa:Address>urn:itfim:wsm:tokenconsumer</wsa:Address>
  </wst:Issuer>
  <wsp:AppliesTo xmlns:wst="...">
    <wsa:EndpointReference>
      <wsa:Address>http://example.com/</wsa:Address>
    </wsa:EndpointReference>
  </wsp:AppliesTo>
  <wst:Base>
    <wss:BinarySecurityToken ..>
      :
    </wss:BinarySecurityToken>
  </wst:Base>
  :
  <wst:RequestType>http://schemas.xmlsoap.org
/ws/2005/02/trust/Validate</wst:RequestType>
</wst:RequestSecurityToken>
```

*Listing 1.2: RequestSecurityToken WS-Trust Message Snippet*

The `<wst:RequestSecurityToken>` element is used to send a request. `<wst:Issuer>` specifies the issuer of the security token that is presented in the message. `<wst:AppliesTo>` defines the scope of the security token, `<wst:Base>` contains the security token, and `<wst:RequestType>` specifies the type of the request. In the example, the message requests validating the embedded binary security token. Listing 1.3 shows a snippet of a WS-Trust message that responds to a `RequestSecurityToken`.

```
<wst:RequestSecurityTokenResponse xmlns:wst="...">
  <wst:Status>
```

```

<wst:Code>http://schemas.xmlsoap.org/ws/
2005/02/trust/status/valid</wst:Code>
</wst:Status>
<wst:RequestedSecurityToken>
  <saml:Assertion ... >
  :
  </saml:Assertion>
</wst:RequestedSecurityToken>
:
</wst:RequestSecurityTokenResponse>

```

*Listing 1.3: RequestSecurityTokenResponse WS-Trust Message Snippet*

The `<wst:RequestSecurityTokenResponse>` element is used to return a security token or response to a request. `<wst:Status>` is used specifically for responding to a validation request to indicate the validation result. `<wst:RequestedSecurityToken>` returns the requested token. In the example, the message indicates that the received token was valid and has returned a SAML token.

### What are Security Token Service (STS) and Web Services Security Management (WSSM)?

Security Token Service (STS) and Web services security management (WSSM) are two key components of identity services. Together, they allow the establishment and management of trust relationships between applications. STS is a Web service that responds to WS-Trust requests. It uses trust service chains to validate, transform, and issue security tokens. STS enables the management of security tokens across trust boundaries. The WSSM component provides functions for service requesters to create outbound security tokens using a token generator, and for service providers to process inbound security tokens using a token consumer. It enhances the WS-Security support and provides a WS-Trust approach to the management of security tokens that STS supports.

Figure 1.4 and Figure 1.5 illustrate how these two components work together to securely send a service request at the requester side and to securely process a service request at the provider side.

### Generating a Security Token for a Web Services Request

Figure 1.4 shows how the WSSM and STS work together to include a security token in a service request.

1. When a service request is created at the requester side, the token generator is called as part of the WS-Security authentication processing.
2. It works with the callback handler to generate a security token.
3. If the service requester is configured to call STS, the token generator then creates a `RequestSecurityToken` WS-Trust message with the security token included along with an `AppliesTo` and an `Issuer`.
4. The `AppliesTo` and `Issuer` are elements of a WS-Trust message. STS uses these two values to uniquely locate the right trust service chain, which then validates, maps, authorizes, and issues a security token.
5. Once processed, STS responds with a `RequestSecurityTokenResponse` WS-Trust message with a security token issued.
6. The token generator includes the security token from STS in the security header of the Web services request message, which is then sent to the service provider.

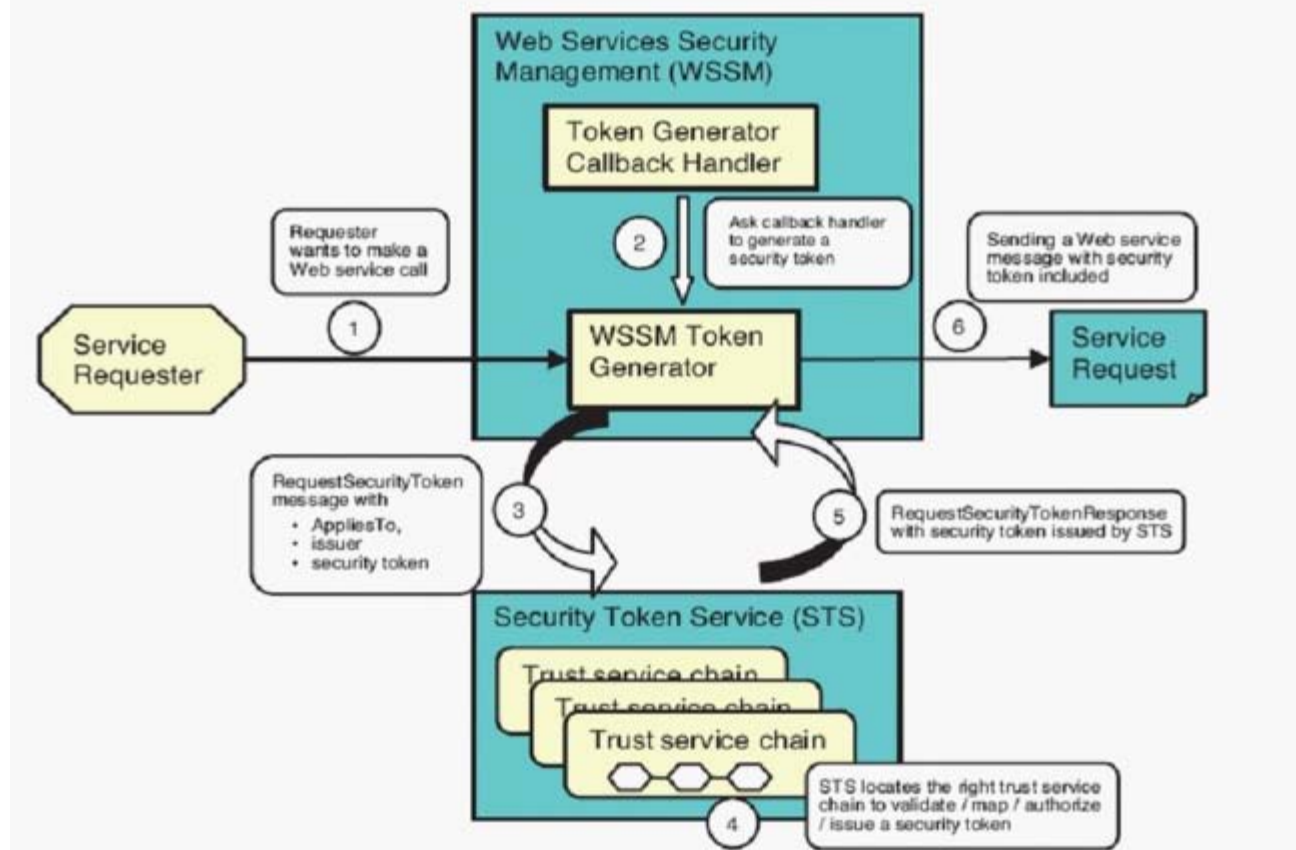


Figure 1.4: Generating a security token for a Web service request

### Consuming a Security Token from a Web Services Request

Figure 1.5 illustrates how to consume the security token that is included in a service request.

1. When a service request is received at the provider side, the token consumer is called as part of WS-Security authentication processing.
2. If the service provider is configured to call STS, the token consumer then creates a RequestSecurityToken WS-Trust message with the security token included along with the AppliesTo and Issuer values.
3. STS uses these two values to uniquely locate the right trust service chain, which then validates, maps, authorizes, and issues a security token.
4. Once processed, STS responds with a RequestSecurityTokenResponse WS-Trust message with a security token issued.
5. The token consumer then works with the callback handler, which passes the security token to the appropriate login module for authentication.
6. Once the credential within the security token is validated, the service provider is accessed.

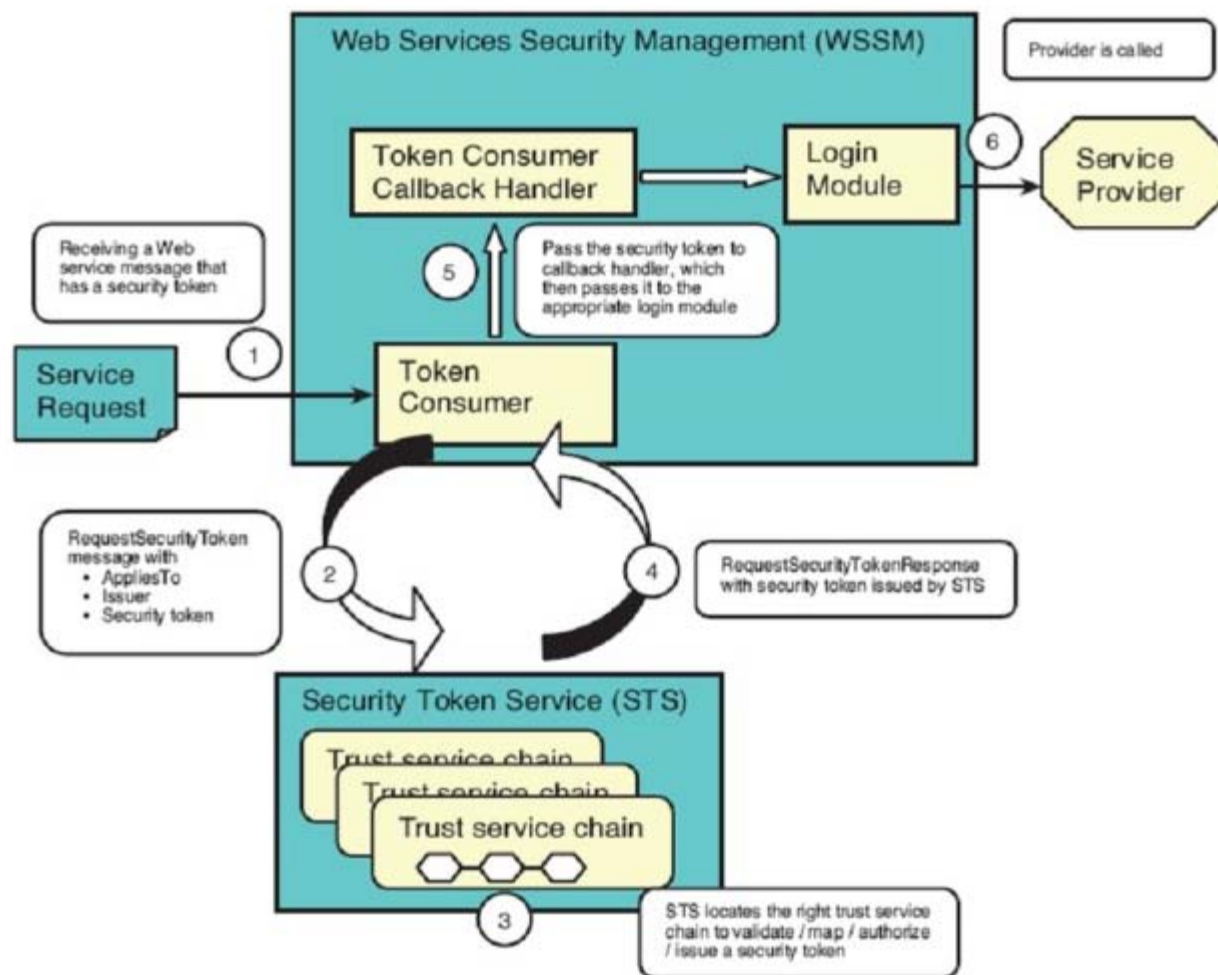


Figure 1.5: Consuming a security token from a Web service request

STS and WSSM form a significant part of Identity Services and play a key role in an SOA scenario. STS enables the management of a trust relationship between security domains; and WSSM adds the ability for message level authentication and authorization. They together facilitate businesses to seamlessly and dynamically interact with each other in a secured, trusted and federated context. It helps increase the responsiveness to new business opportunities by providing security integration with heterogeneous types of service consumers.

## Conclusion

SOA is scalable. Companies who are interested in SOA can choose to begin with certain focus areas and progress through the transformation gradually as requirements come. Service design, service creation, service integration and collaboration, service connectivity, service registry and governance, service security and management are the various focus areas that can be considered. SOA transformation is a journey.

Transforming to SOA is a journey; learning it is also a journey. As organizations advance through the SOA journey, educating the developers on SOA products becomes one of the main focus areas. SOA has been around for years and there is no doubt that its meaning, industry's interpretation, and approaches have evolved. No one will argue that the gist of SOA will continue to be true in the years to come. What remains unchanged is the importance of possessing a good knowledge base for SOA. A solid foundation is essential for building any great structure. A pyramid without a wide base would shatter over time and the Great Wall of China without strong groundwork would not last thousands of years. Similarly for a SOA implementer, having a good knowledge foundation on SOA is crucial for building any IT architecture or project.

## References

[REF-1] "Understanding IBM SOA Foundation Suite: Learning Visually with Examples", IBM Press, ISBN: 0138150400, (by Tinny Ng, Jane Fung, Laura Chan, Vivian Mak) [www.ibmpressbooks.com](http://www.ibmpressbooks.com)

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL



[Home](#) [Past Issues](#) [Contributors](#) [What is SOA?](#) [SOA Glossary](#) [SOA School](#) [SOA Books](#) [About/RSS](#)  
[Legal](#)

Copyright © 2006-2009  
SOA Systems Inc.