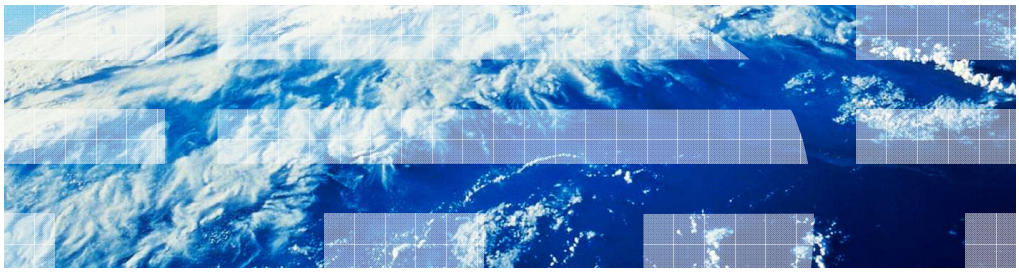
IBM WebSphere Application Server V8

Web services security token exchange support

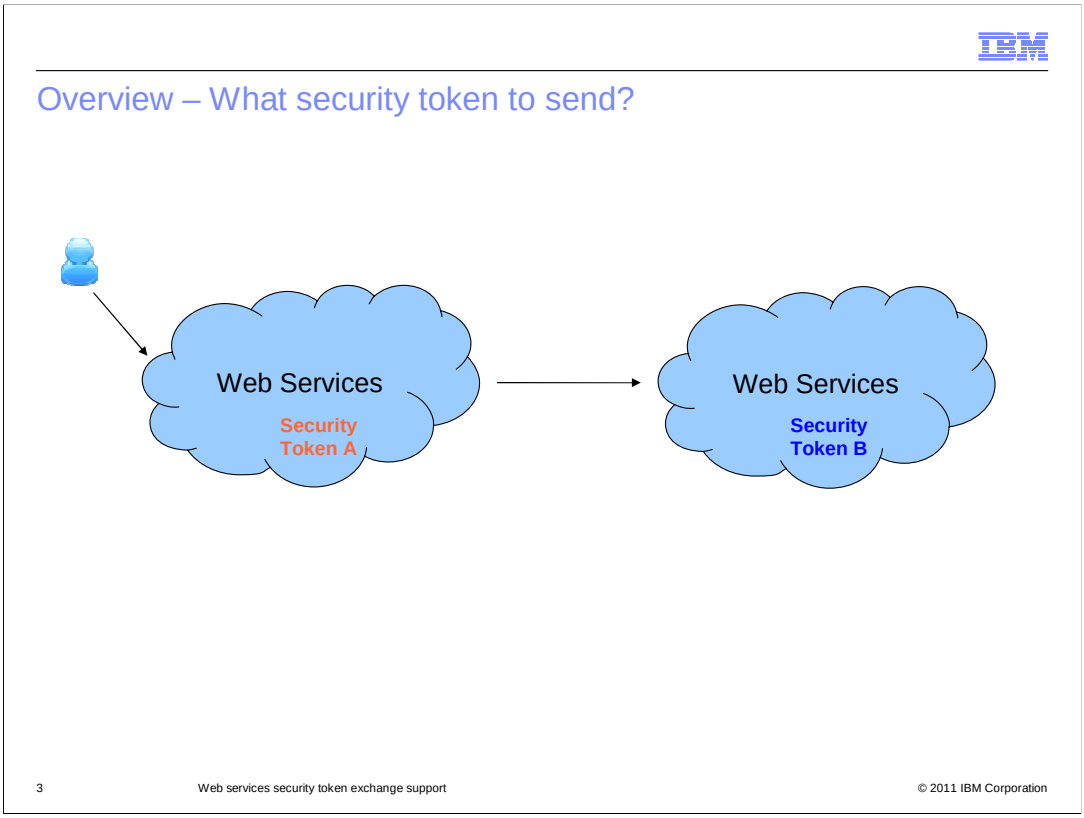


WebSphere. software

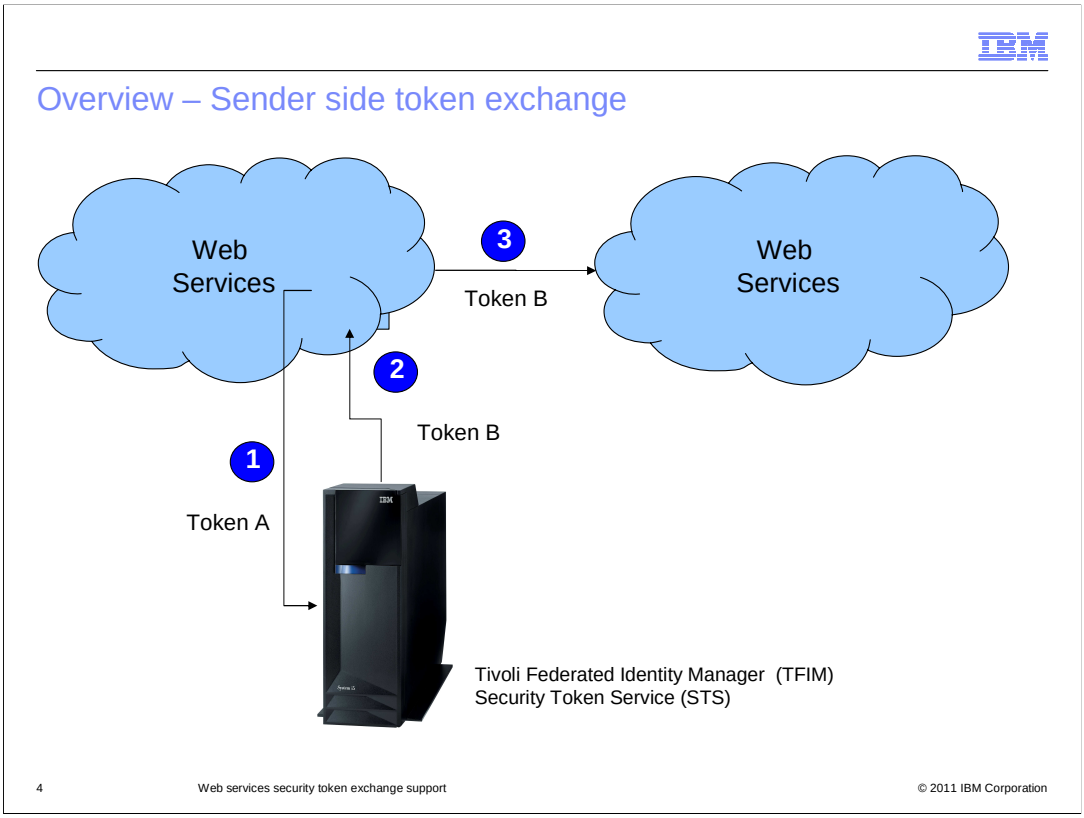
This presentation describes support for the web services security token exchange feature included in IBM WebSphere application Server V8.

Table of contents

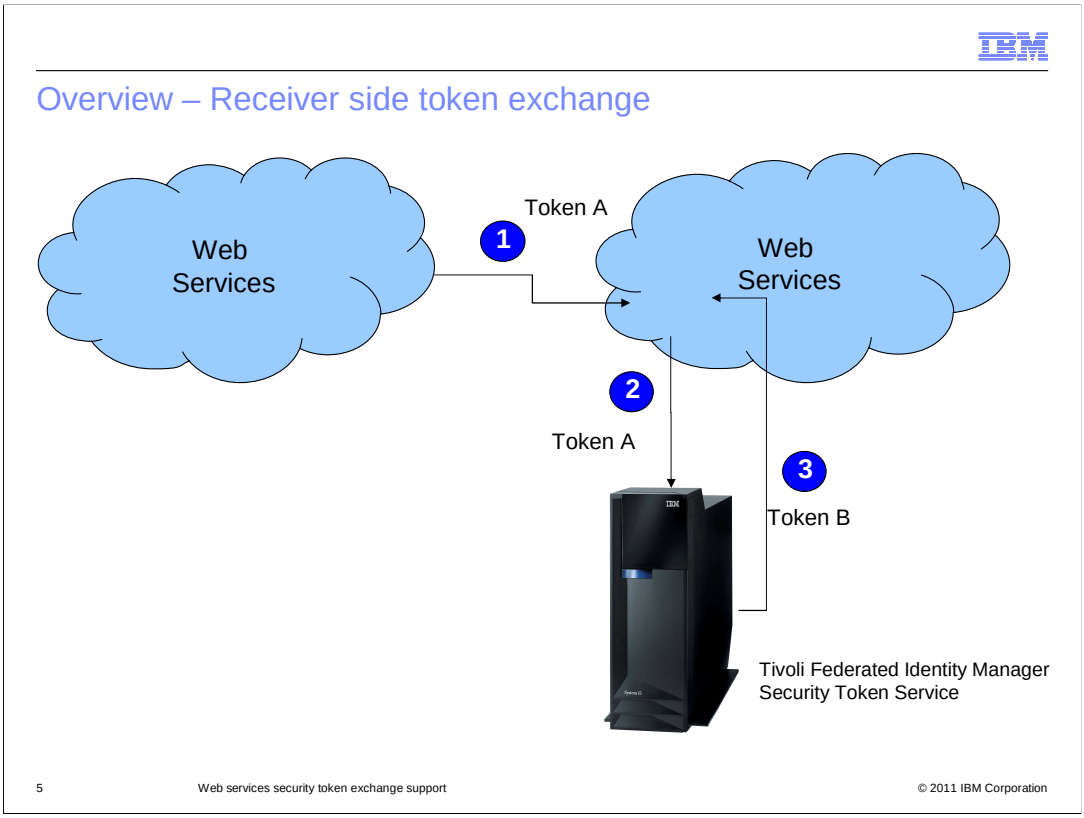
This module will cover an overview of the web services security token exchange feature, the typical usage scenarios, and the solution that this feature provides. It also discusses some pertinent things to consider regarding this feature.



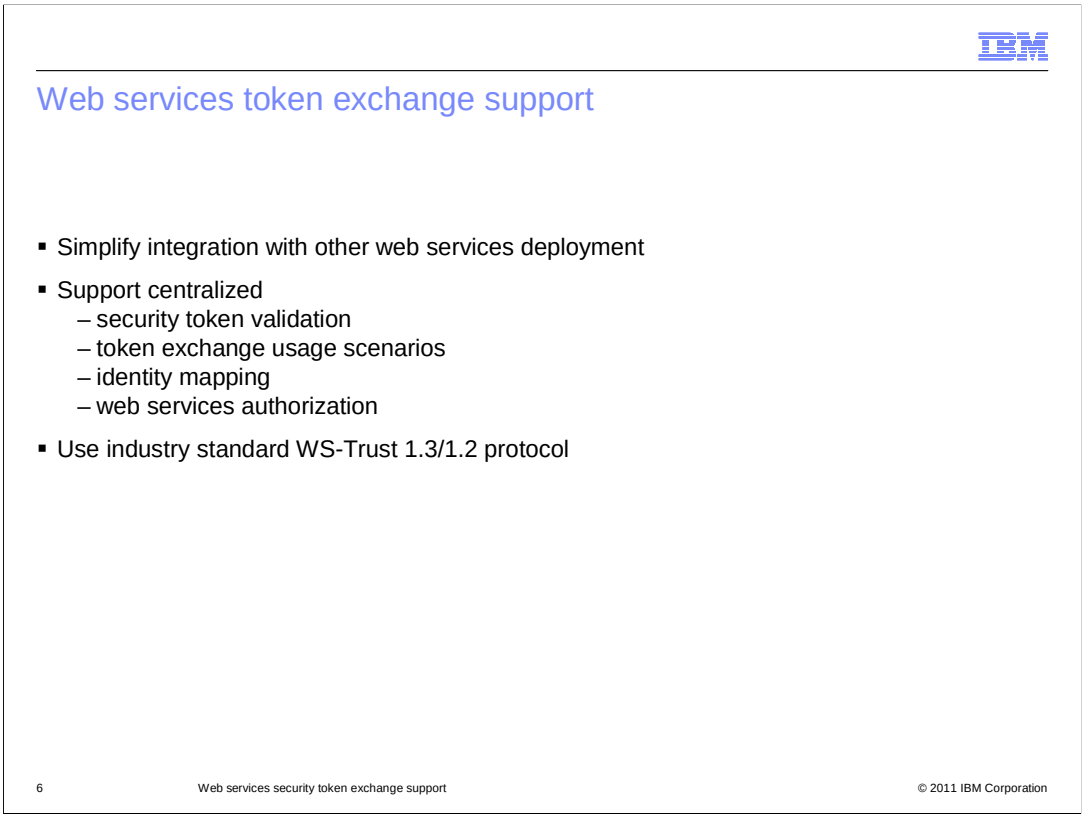
With SOAP web services, security tokens can be sent from a web services client to a web services provider using the web services security standard. When you have two web services that need to communicate, each one might have its own security token requirements. This may be due to platform restrictions, differences in implementations or restrictions on legacy systems. For example, the client web service may have a security token of type “A” and the provider web service may require a security token of type “B”. Before this feature, you might have needed to write custom code in order to convert or exchange the token type at one of the web services. With this feature there is now a mechanism to do the token exchange without having to write custom code.



Given the scenario where the client web service has a token of type “A” and the provider requires a token of type “B”, one option is to have the sender perform the token exchange. In this case, the sender can use a Security Token Service to exchange token “A” for token “B”. Then the sender can send token “B” to the provider web service. An STS is a service that can issue, validate or exchange security tokens. The Tivoli Federated Identity Manager is one such product that provides an STS.



Alternatively, the token can be exchanged at the receiver side. In this case, the web service client sends token “A” to the receiver. The receiver then uses the STS to exchange token “A” for token “B”. Once the receiver has token “B” from the STS, it can authenticate the token. Note that the receiver does not need to be able to understand or authenticate token “A”; it sends it along to the STS to be exchanged for the token type it does understand.

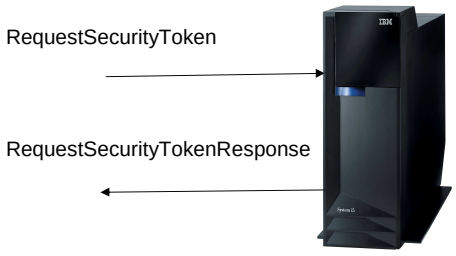


The web services security token exchange feature simplifies integration with other web services deployments by allowing token exchange without having to write custom code.

And by using an STS, the token validation and exchange, identity mapping and web services authorization functions are centralized. This removes the need for each web services instance to have the ability to perform the exchange itself. To perform the exchange itself, each web service would have to understand each of the token types and have access to the user registries required. The centralized STS can perform identity mapping between a user in one realm or registry to another. The STS can also perform a check that the user is authorized to access the particular web service.

Communication with the STS uses the standardized web Services Trust (WS-Trust) protocol version 1.3. The use of WS-Trust 1.2 is also supported.

WS-Trust protocol and security token service



- Issue operation- issue a new security token
- Validate operation- is a security token valid?
- Validate operation- is a security token valid? If so, issue a new security token

Runtime support in V8 and V7.0.0.7

Runtime support in V8 and V7.0.0.11

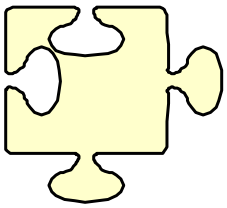
The security token service is itself a web service. And it receives “Request Security Token” messages and replies with “Request Security Token Response” messages. The “Request Security Token” message can request different operations – an “Issue” operation or a “Validate” operation.

The “Issue” operation asks that a new security token be issued and the new security token is returned in the response message.

The “Validate” operation asks for confirmation that a security token is valid. The criteria for validity will vary, but may include checks that the token is not expired, that the signer of the token is trusted, and so on.

The “Validate” operation also has the option of issuing a new token once the original token has been validated. This is effectively the token exchange scenario.

Generic login modules

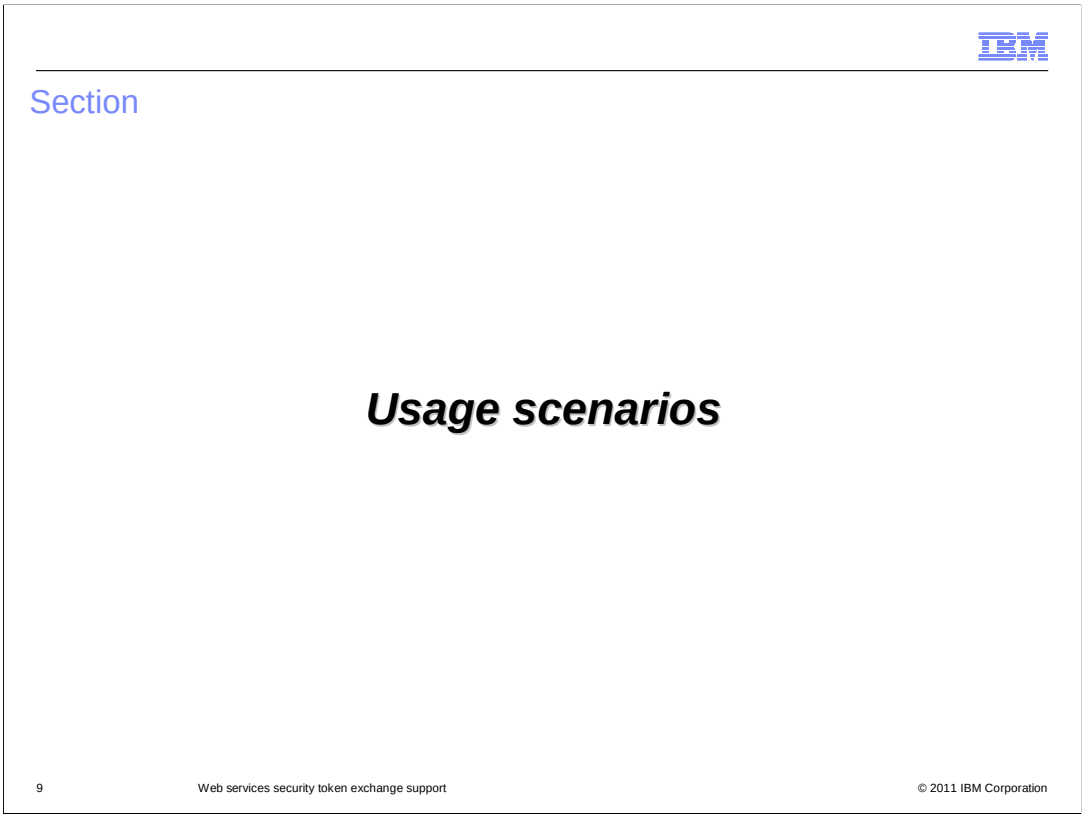


- **GenericIssuedTokenGenerateLoginModule**
  - Send validate requests to STS
  - Validate
  - Exchange
  - Use security token in RunAs Subject
  - Send issue requests to STS
- **GenericIssuedTokenConsumeLoginModule**
  - Validate tokens using STS
  - Validate
  - Exchange

The web services security token exchange function is implemented in the application server as a pair of Java Authentication and Authorization Service (JAAS) login modules. These login modules are referred to as “generic login modules” because they can interact with a variety of token types rather than just a single token type.

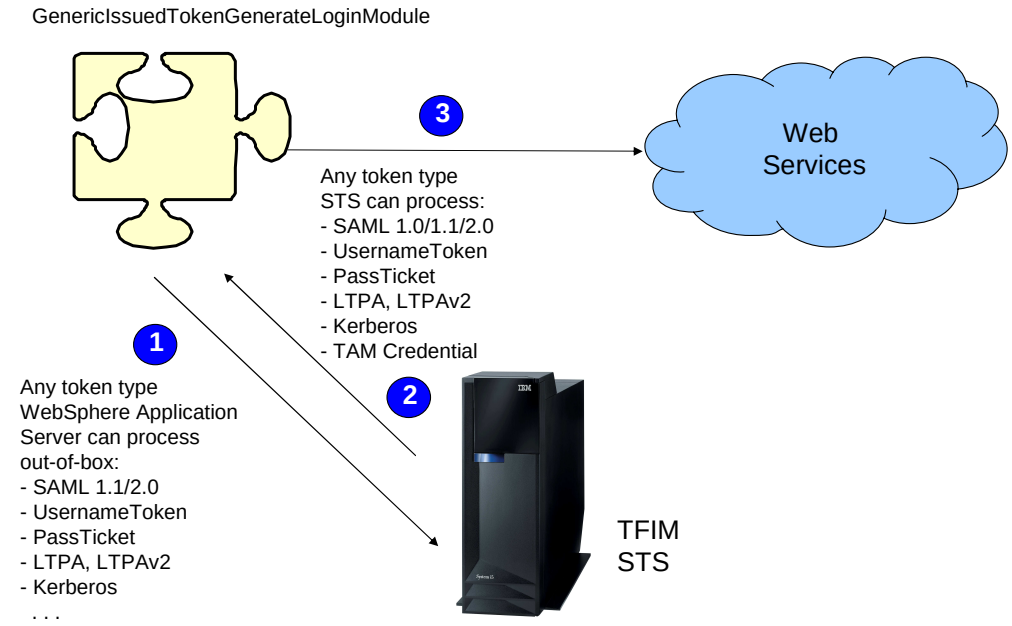
On the client, or generator side, the “generic issued token generate login module” is used. It can perform the token issue, validation or exchange operations described on the previous slide. The security token it sends to the STS for validation or exchange is obtained from the “run as” subject.

On the provider, or consumer side, the “generic issued token consume login module” is used. It can perform the validate or exchange operations described on the previous slide.



This section will get into more detail on the web services security token exchange usage scenarios.

Sender token exchange usage scenario



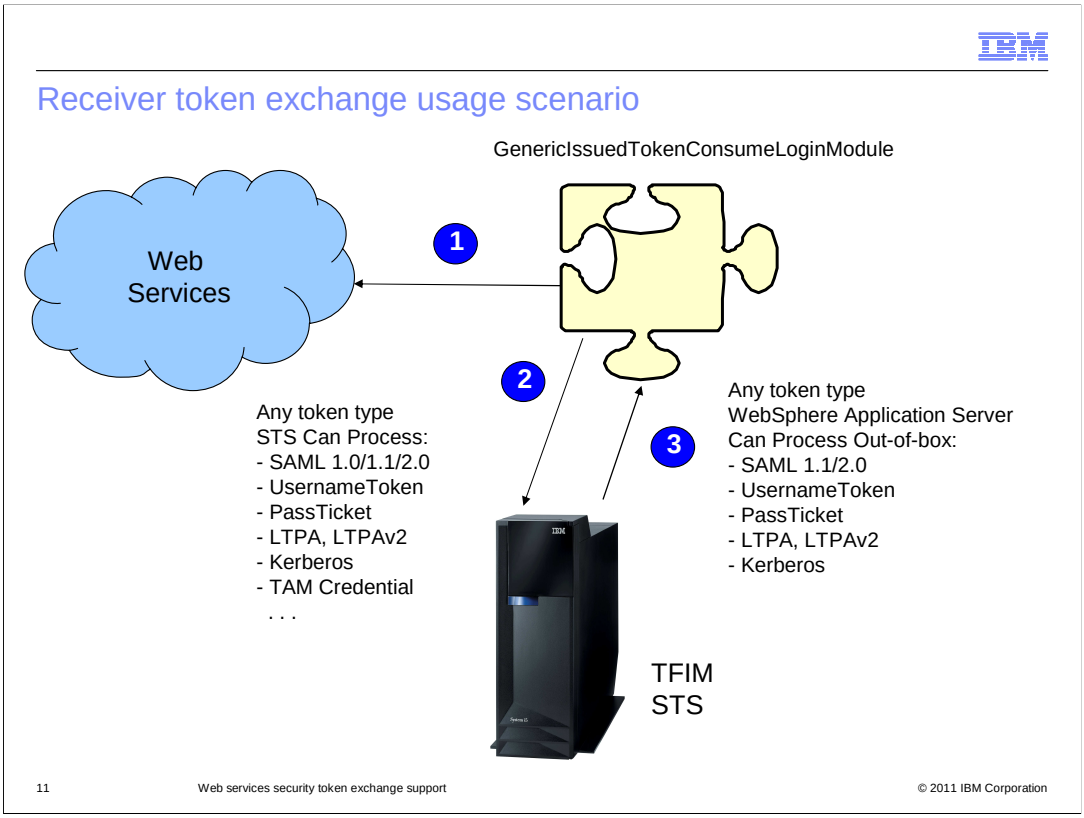
10 Web services security token exchange support

In this scenario, the token exchange is performed at the sender.

In the first part of the flow, labeled “1” in the diagram, the application server extracts the token from the “Run As” subject and sends it in a WS-Trust request to the STS. This token type can be any token that the web services security runtime in the WebSphere Application Server supports. The request also includes the type of token to be returned in the exchange.

In “2”, the STS returns the new token to the sender. This token can be any type that the STS supports.

In “3”, the sender then sends the outbound SOAP message to the target web service with the newly exchanged security token.

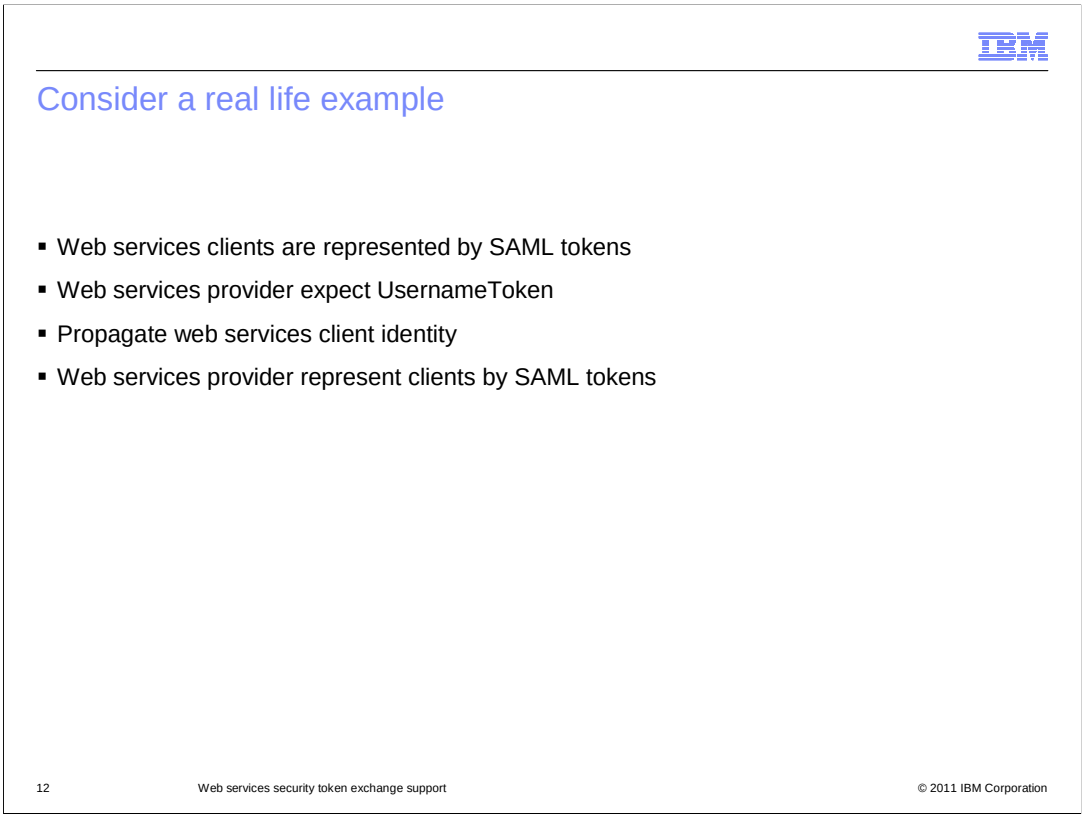


In this scenario, the token exchange is performed at the receiver.

In the first part of the flow, labeled “1” in the diagram, a SOAP message arrives at the receiver containing a security token. This security token can be any token type that the STS can process. The WebSphere Application Server doesn’t need to understand how to process or validate this token type.

In “2”, the application server extracts the token from the SOAP message and sends it in a WS-Trust request to the STS. The request also includes the type of token to be returned in the exchange.

In “3”, the STS returns the new token to the receiver. This token type can be any token that the web services security runtime in the WebSphere Application Server supports.

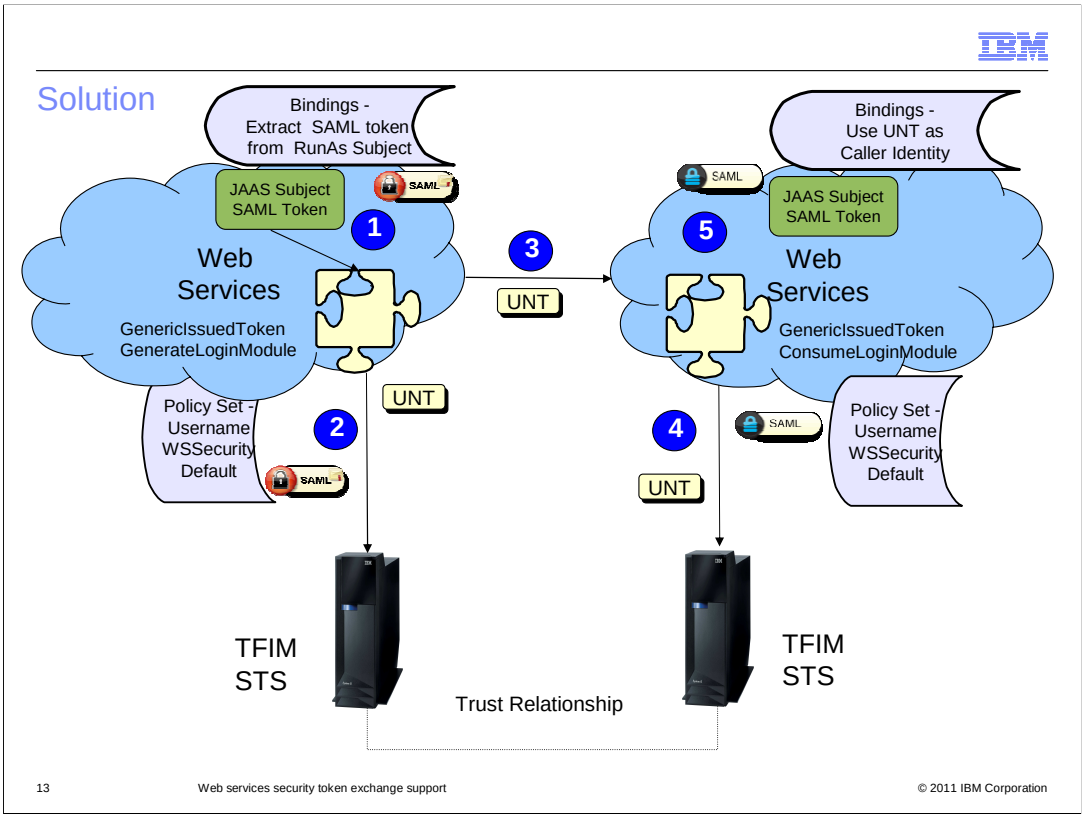


Consider the requirements of a real life example using concrete token types.

The web services client is represented with a SAML (security assertion markup language) token.

The web services provider expects the SOAP message to contain a username token. The user name in the username token must represent the same user as that in the client's SAML token.

Although the provider requires that the incoming SOAP message contain a username token, it ultimately needs to have a SAML token representing the incoming user.



And here is how you can put together the components of this feature to provide a solution to this scenario.

On the left side of this diagram, the web services client will use the “generic issued token generate login module” to exchange the SAML token in the “run as” subject for the username token it will send in the SOAP message to the provider.

Since the token type to be sent in the SOAP message will be a username token, the “policy set” to use will specify a username token. This may be the “username WS-Security default” “policy set” that is shipped with the WebSphere Application Server.

The details of how this username token is to be obtained are specified in the bindings attached to this web services client. The bindings will specify that a SAML token is to be extracted from the “run as” subject and sent to the STS to be exchanged for a username token.

On the right side of the diagram, the web services provider will use the “generic issued token consume login module” to validate the incoming username token and exchange it for a SAML token.

As on the client side, since the token type sent in the SOAP message is a username token, the provider side will also use a username token “policy set”.

And the bindings attached to the web services provider will specify that the username token will be sent to the STS, validated and exchanged for a SAML token. The SAML token received will then be set in the subject of the security context.

This diagram also shows a trust relationship between the two STS servers. This means that they each have been configured to trust and be able to validate the tokens issued by the other STS.

Things to consider

- Exchanged tokens are authentication tokens in SOAP messages
  - Not as protection tokens
- GenericIssuedTokenConsumeLoginModule creates system default security token types
- Generic login modules cannot be used in WSS API

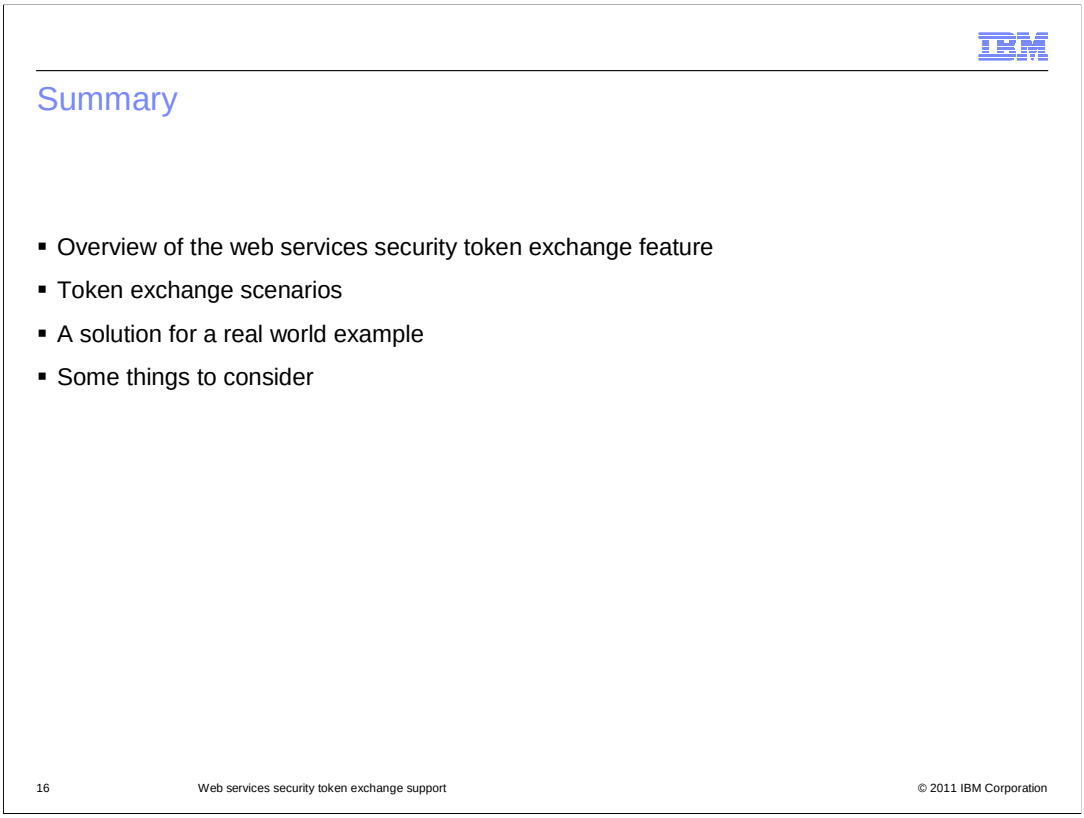
This slide shows some things to consider when using this feature in your environment.

The first is that if this security token exchange feature is used on either the sender or receiver side, the exchanged security token sent in SOAP message can only be an authentication token. It cannot be used as a protection token to sign or encrypt parts of the SOAP message. Though other security tokens not involved in the token exchange can be used for protection of the SOAP message.

Secondly, the token type returned to the “generic issued token consume login module” from the STS must be one of the token types supported by the WebSphere Application Server.

And finally, the “generic login modules” can only be used with “policy sets” and bindings. They cannot be called from the client side web services security APIs.

This section provides a summary of this presentation.



In this presentation you have seen an overview of the web services security token exchange feature and some typical token exchange scenarios.

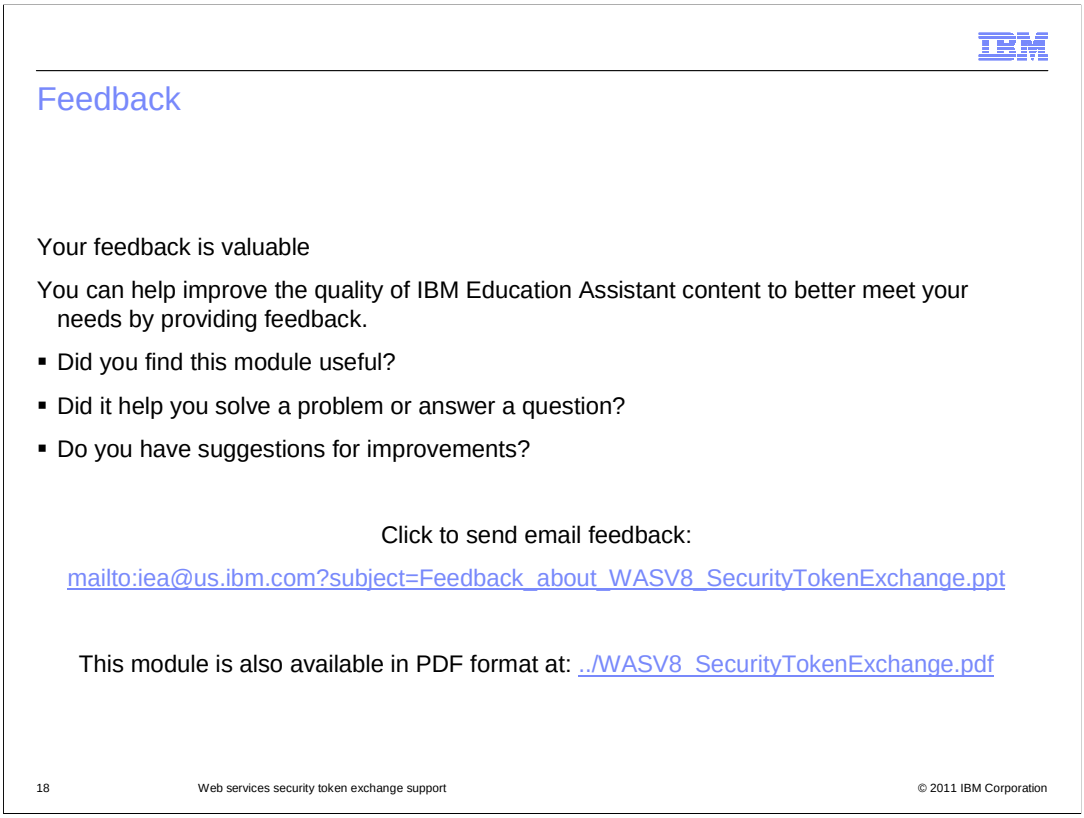
You have also seen a solution to a real world example using this feature.

And finally, some things to consider when using this feature.

References

- Authenticating web services using a generic security token login modules
http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/twbs_secureglm.html
- Feature focus week: Web services token exchange support
http://www.ibm.com/developerworks/forums/thread.jspa?threadID=340953&tstart=0

This slide contains links to useful information.



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information