Introduction to QoS & WSIT

Asst. Prof. Dr. Kanda Runapongsa Saikaew
(krunapon@kku.ac.th)
Mr.Pongsakorn Poosankam
(pongsakorn@gmail.com)

1

Agenda

- What is WSIT?
- How the WSIT Technologies Work
- Bootstrapping and Configuration
- Message Optimization
- Using Reliable Messaging
- Using WSIT Security

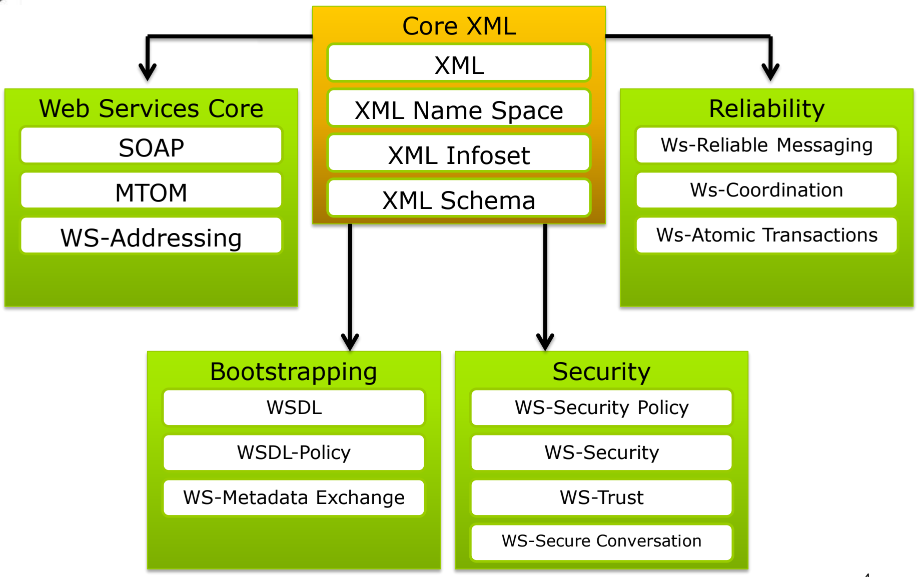
2

What is WSIT?

- WSIT stand for "Web Services Interoperability Technologies"
- Sun Microsystems Web Services interoperate with Microsoft .NET 3.0
  - Reliable Messaging
  - Message Optimization
  - Security
- WSIT is open web services specifications to support enterprise features.
- WSIT is a ***set*** of Quality of Service (QoS)

3

WSIT Web Services Features



4

Agenda

- What is WSIT?
- <u>How the WSIT Technologies Work</u>
- Bootstrapping and Configuration
- Message Optimization
- Reliable Messaging
- WSIT Security

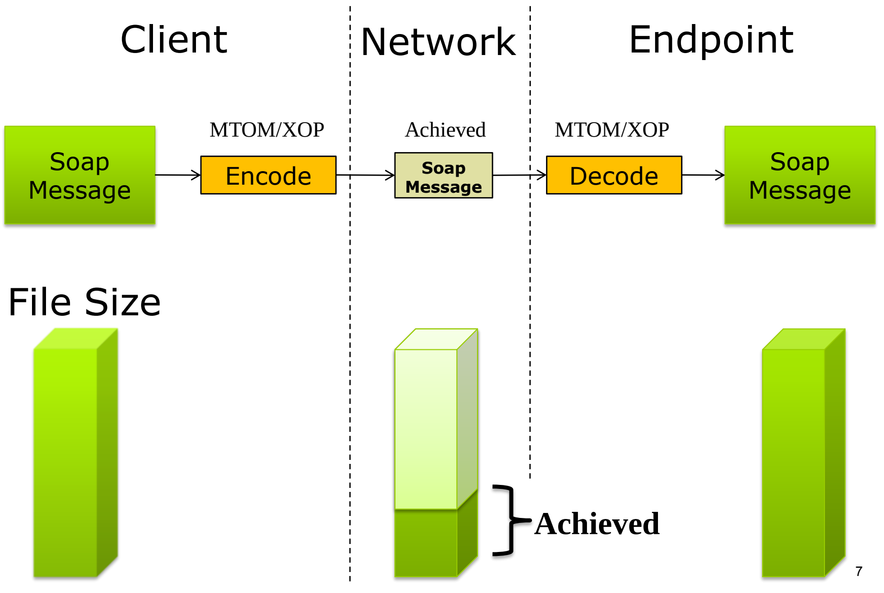
5

How the WSIT Technologies Work

- **<u>How Message Optimization Works</u>**
- How Reliable Messaging Works
- How Security Works

6

How Message Optimization Works (1/4)



7

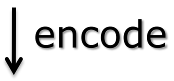
How Message Optimization Works (2/4)

Original Message

```
Content-Type: type="text/xml" ;
-----------------
<soap>
  <a>1</a><b>2</b><c>3</c>
</soap>
```

Soap on HTTP Protocol

```
HTTP header
-----------------
Attachment Part
-----------------
<soap>
  content ..
</soap>
```



Message Optimized

```
Content-Type: type="application/xop+xml" ;
-----------------
Attachment Id=1 (binary)
-----------------
<soap>
  <xop:Include>(binary id)</xop:Include>
</soap>
```

MTOM/XOP

8

How Message Optimization Works (3/4)

- Two approaches of sending binary data via XML
  - Embedding - Base64 encoding
  - Referencing - SOAP with Attachment
- Problem of Base64 encoding
  - Increased size
  - Added overhead
- Problem of SOAP with Attachment
  - Data is external to the document, and it isn't part of the message Infoset, thus requires two different ways of processing data

9

How Message Optimization Works (4/4)

- **XOP (XML-Binary Optimized Packaging)** Is an alternate serialization of XML that just happens to look like a MIME multipart/related package, with an XML document as the root part
- MTOM (Message Transmission Optimization Mechanism)
  - Is a description of how XOP is layered into the SOAP HTTP transport

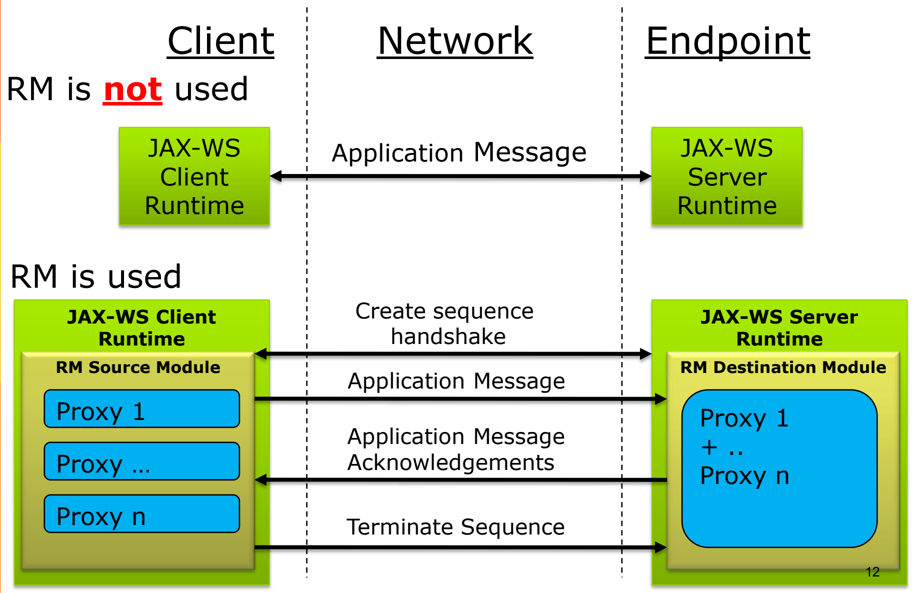
10

How the WSIT Technologies Work

- How Message Optimization Works
- **<u>How Reliable Messaging Works</u>**
- How Security Works

11

How Reliable Messaging Works (1/3)



12

How Reliable Messaging Works (2/3)

- RM source module is plugged into the JAX-WS web service client.
- RM source module transmits the application messages.
- RM source module keeps copies of the messages until their receipt is acknowledged by the destination module via the exchange of protocol messages
- RM destination module acknowledges messages

13

How Reliable Messaging Works (3/3)

- If RM source module not receive acknowledged , it resends the message and requests an acknowledgement.
- RM source terminates the sequence.

14

How the WSIT Technologies Work

- How Message Optimization Works
- How Reliable Messaging Works
- **How Security Works**

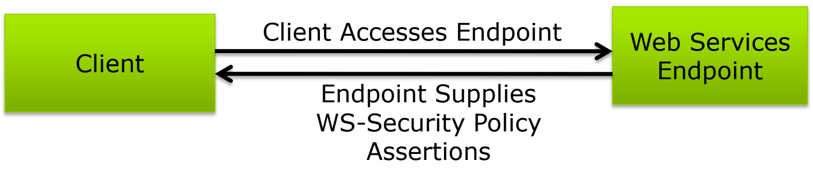
15

How Security Works

- WSIT security technologies following 3 parts
  - **How Security policy work?**
  - How trust work?
  - How secure conversation?

16

How Security policy work?



- Security policy specified in the WSDL.
- The following types of assertions are supported
  - Protection assertions
  - Conditional assertions
  - Security binding assertions
  - Supporting token assertions
  - Web Services Security and Trust assertions

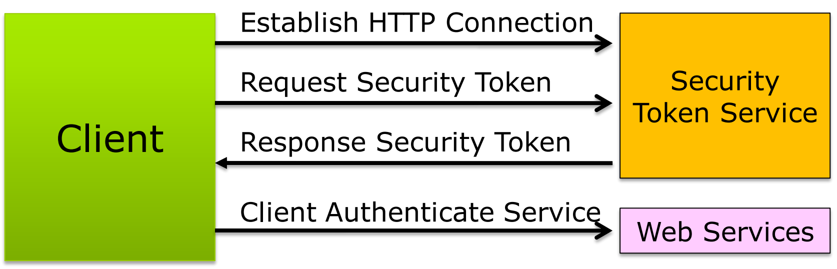
17

How Security Works

- WSIT security technologies following 3 parts
  - How Security policy work?
  - <u>**How trust work?**</u>
  - How secure conversation?

18

How trust work? (1/2)



- The client establishes an HTTPS connection with the Secure Token Service using one of the following methods
  - Username Authentication and Transport Security
  - Mutual Authentication : X509

19

How trust work? (2/2)

- The client sends a RequestSecurityToken message to the Security Token Service
- The Security Token Service sends a Security Assertion Markup Language (SAML) token to the Client.
- The client uses the SAML token to authenticate itself to the web service and trust is established.

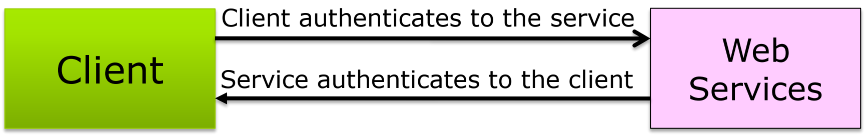
20

How Security Works

- WSIT security technologies following 3 parts
  - How Security policy work?
  - How trust work?
  - **<u>How secure conversation?</u>**

21

How Secure Conversation?


Client
Client authenticates to the service
Service authenticates to the client
Web Services

- The client sends a X509 Certificate to authenticate itself to the web service.
- The web service sends a X509 Certificate to authenticate itself to the client.

22

Agenda

- What is WSIT?
- How the WSIT Technologies Work
- **Bootstrapping and Configuration**
- Message Optimization
- Reliable Messaging
- WSIT Security

23

Bootstrapping and Configuration

- How to create a WSIT client from a Web Service Description Language (WSDL)
- What is a Server-Side Endpoint?
  - A web service endpoint is an entity, processor, or resource that can be referenced and to which web services messages can be addressed
  - Clients use the web service endpoint description to generate code that can send SOAP messages to and receive SOAP messages from the web service endpoint.

24

Creating a Client from WSDL (1/2)

- A WSDL file contains descriptions of the following
  - **Network services**: includes the name of the service, the location of the service, and ways to communicate with the service, that is, what transport to use.
  - **Web services policies**: Policies express the capabilities, requirements, and general characteristics of a web service. Web service providers use policies to specify policy information in a standardized way.

25

Creating a Client from WSDL (2/2)

- Web Services Metadata Exchange (WS-MEX)
  - protocol for requesting and transferring the WSDL from the provider to the client.
  - This protocol is a *bootstrap mechanism* for communication.

26

Agenda

- What is WSIT?
- How the WSIT Technologies Work
- Bootstrapping and Configuration
- <u>**Message Optimization**</u>
- Reliable Messaging
- WSIT Security

27

Message Optimization (1/7)

- How to configure web service providers and clients to use message optimization
  - Creating a Web Service
  - Configuring Message Optimization in a Web Service
  - Deploying and Testing a Web Service
  - Creating a Client to Consume a WSIT-enabled Web Service

28

Message Optimization (2/7)

- Creating a Web Service from Netbeans.
  1. Create “CalculatorApplication” Project
  2. Create “CalculatorWS” Service
  3. Create “add” web metod with 2 parameter and return summation of parameters.

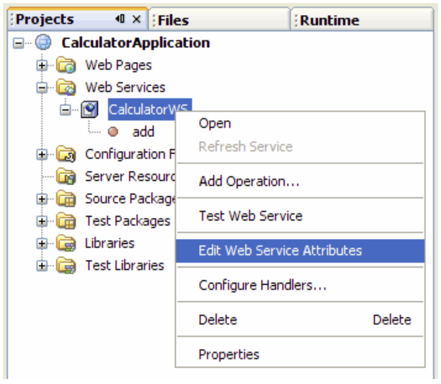
```
@WebService()
public class CalculatorWS {
@WebMethod(operationName = "add")
    public int add(@WebParam(name = "a")
    int a, @WebParam(name = "b")
    int b) {
        return a+b;
    }
}
```

“CalculatorApplication” used for all sample

29

Message Optimization (3/7)

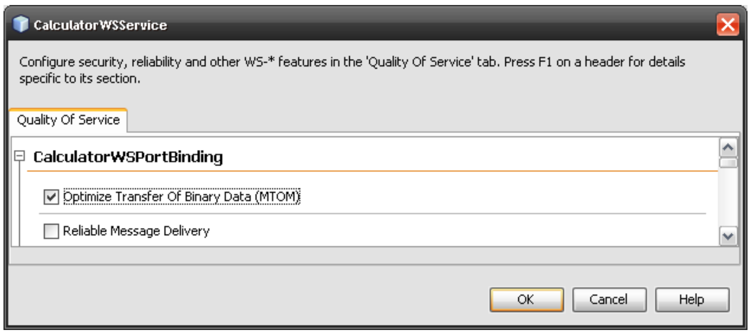
- Configuring Message Optimization in a Web Service
  1. Right-click the CalculatorWS node, and choose Edit Web Service Attributes.



30

Message Optimization (4/7)

2. Select the Optimize Transfer of Binary Data (MTOM) check box,


CalculatorWSService

Configure security, reliability and other WS-* features in the 'Quality Of Service' tab. Press F1 on a header for details specific to its section.

Quality Of Service

CalculatorWSPortBinding

☑ Optimize Transfer Of Binary Data (MTOM)

☐ Reliable Message Delivery

This setting configures the web service to optimize messages that it transmits and to decode optimized messages that it receives.

31

Message Optimization (5/7)

- Deploying and Testing a Web Service

1. Right-click the project node, select Properties, and select Run.
2. Type /CalculatorWSService?wsdl in the Relative URL field and click OK.
3. Right-click the project node and choose Run Project. The IDE starts the web container, builds the application, and displays the WSDL file page in your browser.

32

Message Optimization (6/7)

- Deploying and Testing a Web Service Output.

ws-security added into document namespace

```
xmlns:wsu="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
```

Notice that the WSDL file includes the following WSIT tags

```
<ns1:Policy
xmlns:ns1="http://schemas.xmlsoap.org/ws/2004/09/policy"
wsu:Id="CalculatorWSPortBindingPolicy">
<ns1:ExactlyOne>
    <ns1:All>
            <ns2:OptimizedMimeSerialization
            xmlns:ns2="http://schemas.xmlsoap.org/
            ws/2004/09/policy/optimizedmimeserialization">
        </ns2:OptimizedMimeSerialization>
    </ns1:All>
</ns1:ExactlyOne>
</ns1:Policy>
```

Message Optimization (7/7)

- Creating a Client to Consume a WSIT-enabled Web Service
  - Client can create web service client with normally.
  - The IDE (Netbeans 6.1) use WSDL to create the functionality necessary to satisfy the interoperability requirements of the web service.
  - Monitor HTTP Request/Response with TCPMon

Agenda

- What is WSIT?
- How the WSIT Technologies Work
- Bootstrapping and Configuration
- Message Optimization
- **<u>Reliable Messaging</u>**
- WSIT Security

Reliable Messaging (1/10)

- How to configure reliable messaging in web service endpoint and clients.
  - Reliable Messaging Options
  - Creating Web Services Endpoint
  - Configuring Message Optimization in a Web Service
  - Deploying and Testing a Web Service
  - Creating a Client to Consume a WSIT enabled Web Service

Reliable Messaging (2/10)

- Reliable Messaging Options

Option	Description
Reliable Messaging	Specifies whether reliable messaging is enabled.
Ordered Delivery	Message sequence are delivered to the endpoint application in the order indicated by the message numbers.
Flow Control	Specifies whether the Flow Control feature is enabled.
Max Buffer Size	• Specifies the number of messages that will be buffered for a message sequence • Default 32 kb.
Inactivity Timeout	• Specifies the time interval beyond which either source or destination may terminate any message sequence due to inactivity • Default 10 min.

37

Reliable Messaging (3/10)

- Creating a Web Service from Netbeans.
  1. Create “CalculatorApplication” Project
  2. Create “CalculatorWS” Service
  3. Create “add” web metod with 2 parameter and return summation of parameters.

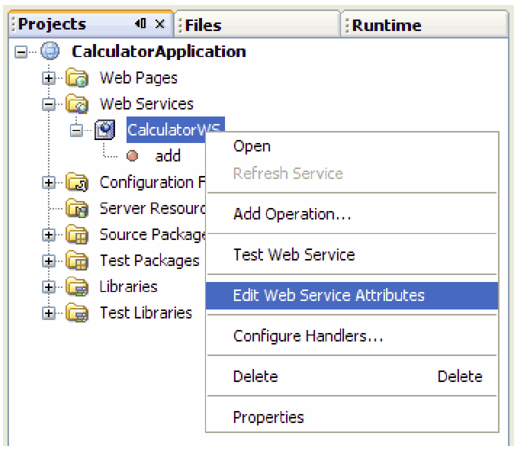
```
@WebService()
public class CalculatorWS {
@WebMethod(operationName = "add")
    public int add(@WebParam(name = "a")
    int a, @WebParam(name = "b")
    int b) {
        return a+b;
    }

}
```

38

Reliable Messaging (4/10)

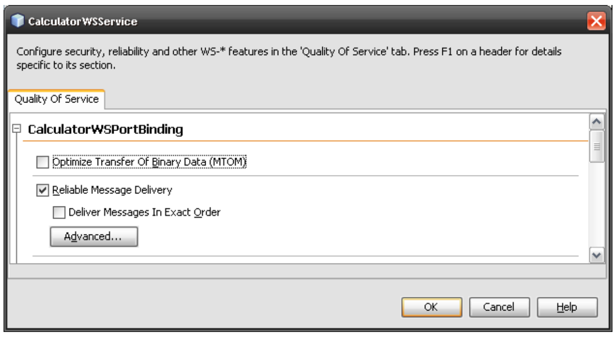
- Configuring Message Optimization in a Web Service
  - Right-click the CalculatorWS node, and choose Edit Web Service Attributes



39

Reliable Messaging (5/10)

- Select the Reliable Message Delivery check box and click ok.


CalculatorWSService

Configure security, reliability and other WS-* features in the 'Quality Of Service' tab. Press F1 on a header for details specific to its section.

Quality Of Service

CalculatorWSPortBinding

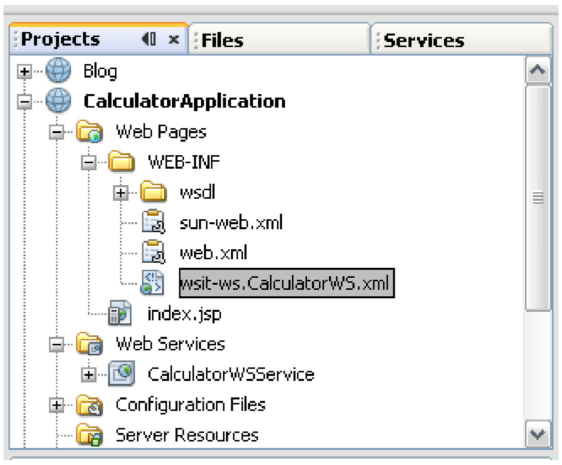
- [ ] Optimize Transfer Of Binary Data (MTOM)
- [x] Reliable Message Delivery
  - [ ] Deliver Messages In Exact Order

This setting ensures that the service sends an acknowledgement to the clients for each message that is delivered,

40

Reliable Messaging (6/10)

- In the left pane, expand the Web Pages node and the WEB-INF node, and open the wsit-ws.CalculatorWS.xml file in the Source Editor



41

Reliable Messaging (7/10)

- Notice that the IDE has added the following tags to the file to enable reliable messaging

```
 <wsp:Policy wsu:Id="CalculatorWSPortBindingPolicy">
      <wsp:ExactlyOne>
         <wsp:All>
            <wsaws:UsingAddressing

        xmlns:wsaws="http://www.w3.org/2006/05/addressi
ng/wsdl"/>
            <wsrm:RMAssertion/>
         </wsp:All>
      </wsp:ExactlyOne>
   </wsp:Policy>
```

42

Reliable Messaging (8/10)

- Deploying and Testing a Web Service

1. Right-click the project node, select Properties, and select Run.
2. Type /CalculatorWSService?wsdl in the Relative URL field and click OK.
3. Right-click the project node and choose Run Project. The IDE starts the web container, builds the application, and displays the WSDL file page in your browser.

43

Reliable Messaging (9/10)

- Deploying and Testing a Web Service Output.

ws-security added into document namespace

```
xmlns:wsu="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
```

Notice that the WSDL file includes the following WSIT tags

```
<ns1:Policy xmlns:ns1="http://schemas.xmlsoap.org/ws/2004/09/policy"
wsu:Id="CalculatorWSPortBindingPolicy">
    <ns1:ExactlyOne>
        <ns1:All>
            <ns2:RMAssertion xmlns:ns2="http://schemas.xmlsoap.org
            /ws/2005/02/rm/policy"></ns2:RMAssertion>
            <ns3:UsingAddressing xmlns:ns3="http://www.w3.org
            /2006/05/addressing/wsdl"></ns3:UsingAddressing>
        </ns1:All>
    </ns1:ExactlyOne>
</ns1:Policy>
```

44

Reliable Messaging (10/10)

- Creating a Client to Consume a WSIT-enabled Web Service
  - Client can create web service client with normally.
  - The IDE (Netbeans 6.1) use WSDL to create the functionality necessary to satisfy the interoperability requirements of the web service.

45

Agenda

- What is WSIT?
- How the WSIT Technologies Work
- Bootstrapping and Configuration
- Message Optimization
- Reliable Messaging
- **WSIT Security**

46

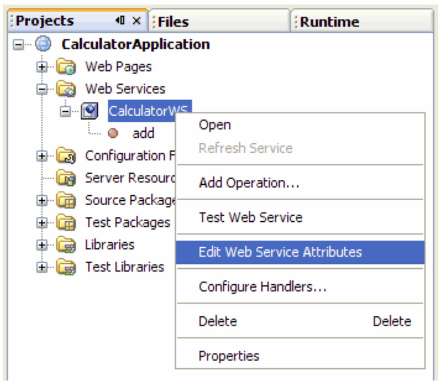
WSIT Security

- **Configuring Security Using NetBeans IDE**
  - Securing the Service
  - Securing the Client
- Summary of Configuration Requirements
  - Service Side
  - Client Side
- Security Mechanisms
- Configuring SSL and Authorized Users
- Configuring Keystores and Truststores
- Securing an Operation
- Configuring A Secure Token Service

47

Configuring Security Using NetBeans IDE (1/4)

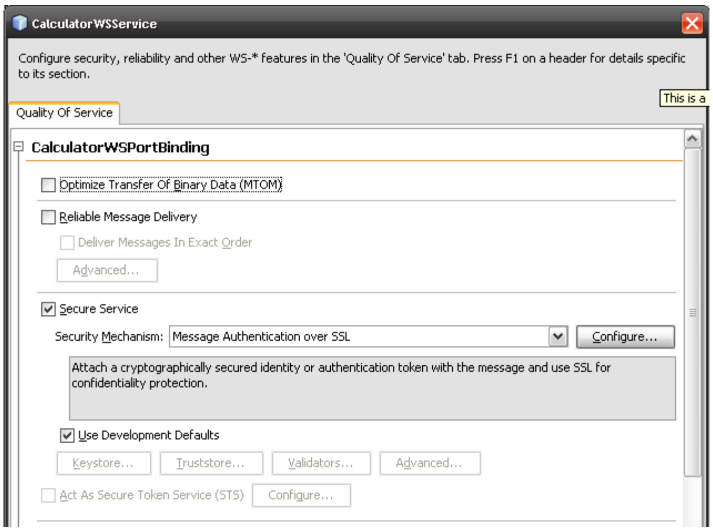
- Securing the Service
  1. Right-click the CalculatorWS node, and choose Edit Web Service Attributes.



48

Configuring Security Using NetBeans IDE (2/4)

2. Select Secure Service And Security Mechanism from the list



CalculatorWSService

Configure security, reliability and other WS-* features in the 'Quality Of Service' tab. Press F1 on a header for details specific to its section.

Quality Of Service

CalculatorWSPortBinding

- [ ] Optimize Transfer Of Binary Data (MTOM)
- [ ] Reliable Message Delivery
  - [ ] Deliver Messages In Exact Order
  - Advanced...
- [x] Secure Service

Security Mechanism: Message Authentication over SSL — Configure...

Attach a cryptographically secured identity or authentication token with the message and use SSL for confidentiality protection.

- [x] Use Development Defaults

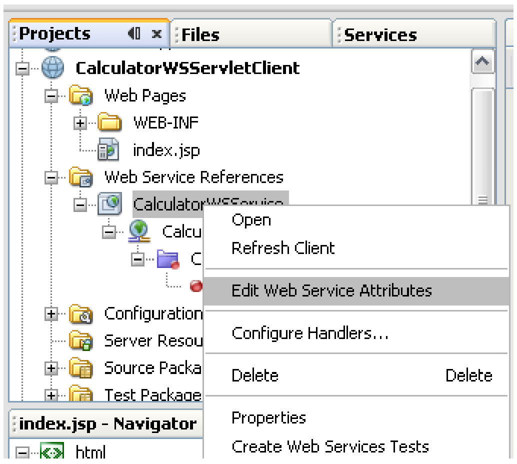
Keystore... Truststore... Validators... Advanced...

- [ ] Act As Secure Token Service (STS) — Configure...

49

Configuring Security Using NetBeans IDE (3/4)

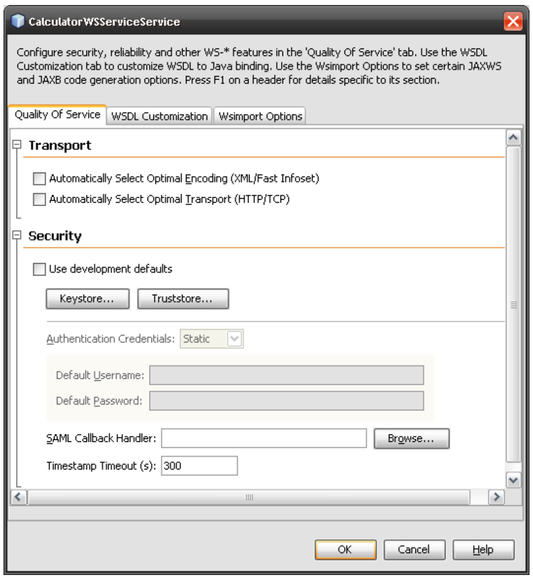
- Securing the Client
  1. Right-click the CalculatorWS node, and choose Edit Web Service Attributes.



50

Configuring Security Using NetBeans IDE (4/4)

2. Select the Quality Of Service tab options


CalculatorWSServiceService

Configure security, reliability and other WS-* features in the 'Quality Of Service' tab. Use the WSDL Customization tab to customize WSDL to Java binding. Use the Wsimport Options to set certain JAXWS and JAXB code generation options. Press F1 on a header for details specific to its section.

Quality Of Service | WSDL Customization | Wsimport Options

Transport

- [ ] Automatically Select Optimal Encoding (XML/Fast Infoset)
- [ ] Automatically Select Optimal Transport (HTTP/TCP)

Security

- [ ] Use development defaults

Keystore... Truststore...

Authentication Credentials: Static

Default Username:

Default Password:

SAML Callback Handler: Browse...

Timestamp Timeout (s): 300

OK Cancel Help

51

WSIT Security

- Configuring Security Using NetBeans IDE
  - Securing the Service
  - Securing the Client
- Security Mechanisms
- Summary of Configuration Requirements
  - Service Side
  - Client Side

52

Security Mechanisms

- The following lists the possible choices for security mechanisms.
  - Username Authentication with Symmetric Keys
  - Mutual Certificates Security
  - Transport Security (SSL)
  - Message Authentication over SSL
  - SAML Authorization over SSL
  - Endorsing Certificate
  - SAML Sender Vouches with Certificates
  - SAML Holder of Key
  - STS Issued Token
  - STS Issued Token with Service Certificate
  - STS Issued Endorsing Token

53

Username Authentication with Symmetric Keys

- Supports integrity and confidentiality using symmetric key
  - Symmetric key cryptography relies on a single, shared secret key that is used to encrypt a message
  - The shared, symmetric key is generated at runtime and encrypted using the service's certificate.
  - The client must specify the alias in the truststore by identifying the server's certificate alias.
- Supports authentication using username/password
  - The client does not possess any certificate/key of his own

54

Mutual Certificates Security

- Supports integrity and confidentiality messages,supports authentication
- When using mutual certificates, a keystore and truststore file must be configured for both the client and server sides of the application
  - A keystore is a database of private keys and their associated X.509 certificate chains authenticating the corresponding public keys.
  - A truststore is a database of trusted entities and their associated X.509 certificate chains authenticating the corresponding public keys.

55

Transport Security (SSL) (1/3)

- Security is provided by the transport mechanisms used to transmit data over the wire between clients and providers
  - Relies on secure HTTP transport (HTTPS) using Secure Sockets Layer (SSL)
- Transport security is a point-to-point security mechanism that can be used for authentication, integrity, and confidentiality.
- When running over an SSL-protected session, the server and client negotiate an encryption algorithm and cryptographic keys

56

Transport Security (SSL) (2/3)

- Security is "live" from the time the data leaves the consumer until it arrives at the provider, or vice versa.
  - The problem is that it is not protected once it gets to its destination.
  - For protection of data after it reaches its destination, use one of the security mechanisms that uses SSL and also secures data at the message level.

57

Transport Security (SSL) (3/3)

- Digital certificates are necessary when running secure HTTP transport (HTTPS) using Secure Sockets Layer (SSL).
  - The HTTPS service of most web servers will not run unless a digital certificate has been installed.
  - Digital certificates have already been created for GlassFish, and the default certificates are sufficient for running this mechanism, and are required when using Atomic Transactions.

58

Message Authentication over SSL

- Attaches a cryptographically secured identity or authentication token with the message and use SSL for confidentiality
- By default, a Username Supporting Token will be used for message authentication
  - To use an X.509 Supporting Token instead, click the Configure button and select X509.

SAML Sender Vouches with Certificates (1/4)

- Protects messages with mutual certificates for integrity and confidentiality and with a **Sender Vouches SAML token** for authorization.
  - The Sender Vouches method establishes the correspondence between a SOAP message and the SAML assertions added to the SOAP message.
  - The attesting entity provides the confirmation evidence that will be used to establish the correspondence between the subject of the SAML subject statements (in SAML assertions) and SOAP message content.
  - The attesting entity, presumed to be different from the subject, vouches for the verification of the subject.

SAML Sender Vouches with Certificates (2/4)

- The receiver has an existing trust relationship with the attesting entity.
- The attesting entity protects the assertions (containing the subject statements) in combination with the message content against modification by another party.
- For this mechanism, the SAML token is included as part of the message signature as an authorization token and is sent only to the recipient.

61

SAML Sender Vouches with Certificates (3/4)

- The message payload needs to be signed and encrypted
- The requestor is vouching for the credentials (present in the SAML assertion) of the entity on behalf of which the requestor is acting.
- The initiator token, which is an X.509 token, is used for signature.

62

SAML Sender Vouches with Certificates (4/4)

- The recipient token, which is also an X.509 token, is used for encryption.
- For the server, this is reversed, the recipient token is the signature token and the initiator token is the encryption token. A SAML token is used for authorization.

63

SAML Holder of Key (1/3)

- Protects messages with a signed SAML assertion (issued by a trusted authority) carrying client public key and authorization information with integrity and confidentiality protection using mutual certificates.
- The Holder-of-Key (HOK) method establishes the correspondence between a SOAP message and the SAML assertions added to the SOAP message.
- The attesting entity includes a signature that can be verified with the key information in the confirmation method of the subject statements of the SAML assertion referenced for key info for the signature.

64

SAML Holder of Key (2/3)

- Under this scenario, the service does not trust the client directly, but requires the client to send a SAML assertion issued by a particular SAML authority.
- The client knows the recipient's public key, but does not share a direct trust relationship with the recipient.

SAML Holder of Key (3/3)

- The recipient has a trust relationship with the authority that issues the SAML token.
- The request is signed with the client's private key and encrypted with the server certificate. The response is signed using the server's private key and encrypted using the key provided within the HOK SAML assertion.

STS Issued Token (1/2)

- This security mechanism protects messages using a token issued by a trusted Secure Token Service (STS) for message integrity and confidentiality protection.
- An STS is a service that implements the protocol defined in the WS-Trust specification
  - This protocol defines message formats and message exchange patterns for issuing, renewing, canceling, and validating security tokens.

STS Issued Token (2/2)

- Service providers and consumers are in potentially different managed environments but use a single STS to establish a chain of trust.
- The service does not trust the client directly, but instead trusts tokens issued by a designated STS.
  - In other words, the STS is taking on the role of a second service with which the client has to securely authenticate.
  - The issued tokens contain a key, which is encrypted for the server and which is used for deriving new keys for signing and encrypting.

STS Issued Token with Service Certificates

- This security mechanism is similar to the one discussed in STS Issued Token, with the difference being that in addition to the service requiring the client to authenticate using a SAML token issued by a designated STS, confidentiality protection is achieved using a service certificate.
- A service certificate is used by a client to authenticate the service and provide message protection.
  - For GlassFish, a default certificate of s1as is installed.

69

STS Issued Endorsing Token

- This security mechanism is similar to the one discussed in STS Issued Token, with the difference being that the client authenticates using a SAML token that is issued by a designated STS.
- An endorsing token is used to sign the message signature.
- In this mechanism, message integrity and confidentiality areprotected using ephemeral keys encrypted for the service. Ephemeral keys use an algorithm where the exchange key value is purged from the cryptographic service provider (CSP) when the key handle is destroyed. The service requires messages to be endorsed by a SAML token issued by a designated STS.

70

WSIT Security

- Configuring Security Using NetBeans IDE
  - Securing the Service
  - Securing the Client
- Security Mechanisms
- **Summary of Configuration Requirements**
  - Service Side
  - Client Side

71

Server-side Configuration Requirements (1/2)

• Summary of Service-side Configuration Requirements

Mechanism	Keystore	Truststore (no alias)	STS	SSL	User in GlassFish
Username Auth. w/Symmetric Keys	Yes				Yes
Mutual Certs.	Yes	Yes			
Transport Sec.				Yes	Yes
Message Auth. over SSL - Username Token				Yes	Yes
Message Auth. over SSL - X.509 Token		YES		Yes	
SAML Auth. over SSL	Yes	Yes		Yes	Yes

72

Server-side Configuration Requirements (2/2)

•Summary of Service-side Configuration Requirements

Mechanism	Keystore	Truststore (no alias)	STS	SSL	User in GlassFish
Endorsing Cert.	Yes				
SAML Sender Vouches with Cert.	Yes	Yes			Yes
SAML Holder of Key	Yes	Yes			Yes
STS Issued Token			Yes		
STS Issued Token with Service Cert.			Yes		
STS Issued Endorsing Token			Yes		

73

Client-side Configuration Requirements (1/2)

•Summary of Client-side Configuration Requirements

Mechanism	Keystore	Truststore	Default User	SAML Callback Handle	STS	SSL	User in GlassFish
Username Auth. w/Symmetric Keys		Yes	Yes				Yes
Mutual Certs.	Yes	Yes					
Transport Sec.						Yes	Yes
Message Auth. over SSL - Username Token			Yes			Yes	Yes
Message Auth. over SSL - X.509 Token	Yes					Yes	
SAML Auth. over SSL	Yes	Yes		Yes		Yes	Yes

74

Client-side Configuration Requirements (2/2)

•Summary of Client-side Configuration Requirements

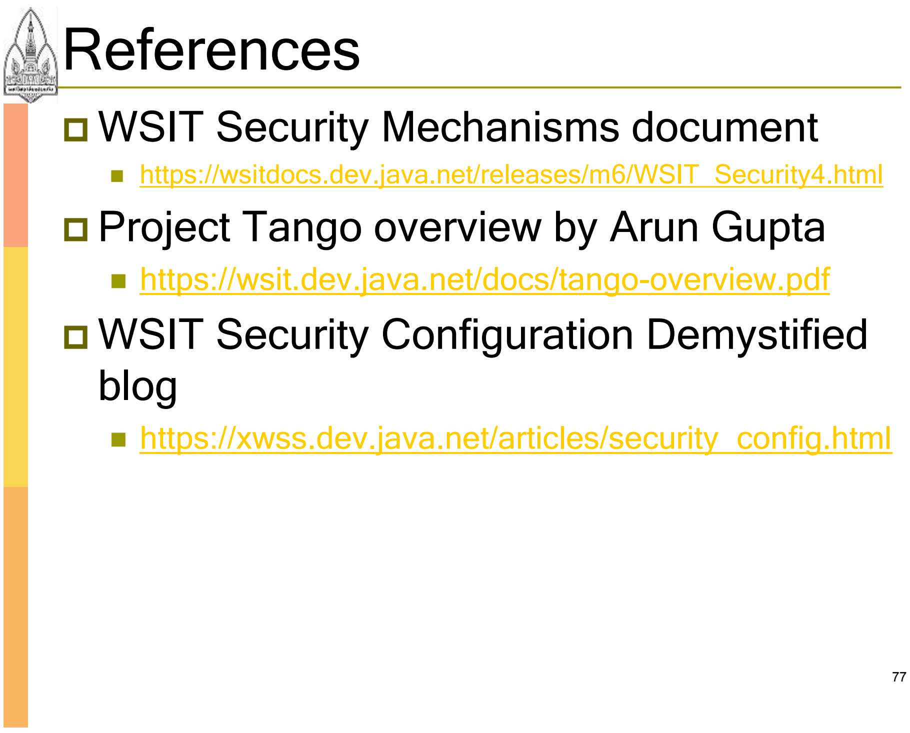
Mechanism	Keystore	Truststore	Default User	SAML Callback Handle	STS	SSL	User in GlassFish
Endorsing Cert.	Yes	Yes					
SAML Sender Vouches with Cert.	Yes	Yes		Yes			Yes
SAML Holder of Key	Yes	Yes		Yes			Yes
STS Issued Token	Yes	Yes			Yes		
STS Issued Token with Service Cert.	Yes	Yes			Yes		
STS Issued Endorsing Token	Yes	Yes			Yes		

75

Summary

- Security layer
  - Transport or Message Layer
- Type of client credentials
  - Username/Password,x509 certificate, SAML assertion or issue token from a third party authority (STS)
- The role of the client credential played in securing the messages
  - As a supporting token or as a primary securing token.
  - In the case of supporting token, the message are usually secured with server's X509 certificate.

76



References

- WSIT Security Mechanisms document
  - https://wsitdocs.dev.java.net/releases/m6/WSIT_Security4.html
- Project Tango overview by Arun Gupta
  - https://wsit.dev.java.net/docs/tango-overview.pdf
- WSIT Security Configuration Demystified blog
  - https://xwss.dev.java.net/articles/security_config.html

77