

Securing Web Services with Apache WS

Nandana Mihindukulasooriya
Blekinge Institute of Technology



Agenda

- WS Security
- Apache Rampart
- How to secure web services with Rampart
- WS-Trust / WS-SecureConversation



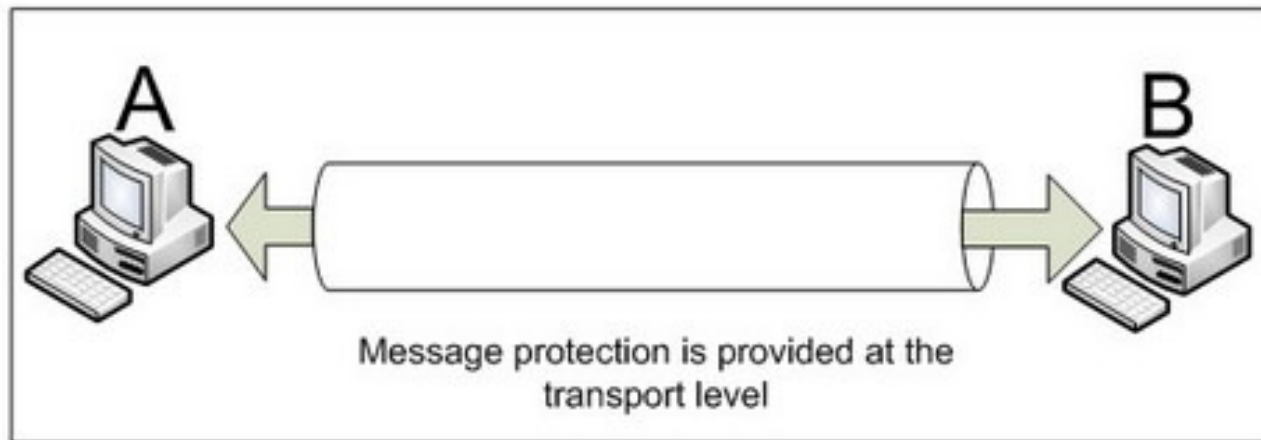
Why ??

- Authentication
- Authorization
- Integrity
- Non repudiation
- Confidentiality
- Replay detection



HTTPS (Transport level)

- Simple
- Fast
- Widely used



HTTPS Problems

- Point-to-Point
- Securing parts of the message
- Auditing
- Federation



Signing

```
<soap:Envelope >  
  <soap:Body>  
    <ns1:report >  
      <ssn></ ssn>  
      <feedback></ feedback>  
    </ ns1:report >  
  </soap:Body>  
</soap:Envelope>
```



Encrypting

```
<soap:Envelope >
  <soap:Body>
    <ns1:order >
      <itemInfo></ itemInfo>
      <creditCard></creditCard >
    </ ns1:order >
  </soap:Body>
</soap:Envelope>
```



XML Security

- Confidentiality of XML documents
 - XML Encryption by W3C
 - <http://www.w3.org/TR/xmlenc-core/>
- Integrity and non-repudiation
 - XML Signature by W3C
 - <http://www.w3.org/TR/xmlsig-core>



Apache XML Security

- Java / C++
- <http://santuario.apache.org/>



WS Security

- Standard set of SOAP [SOAP11, SOAP12] extensions that can be used when building secure Web services
- Standard way to send security meta data



WS Security

- How to attach / embed security tokens to SOAP messages
- How to reference to those tokens
- How to use XML Signature inside a SOAP message
- How to use XML Encryption inside a SOAP message
- How to send all these meta data in a standard way



Security Header

```
<soapenv:Envelope>
  <soapenv:Header>
    <wsse:Security
      soapenv:mustUnderstand="1">
      <wsu:Timestamp
        wsu:Id="Timestamp-31497899">
        <wsu:Created>2008-02-06T13:39:50.943Z</wsu:Created>
        <wsu:Expires>2008-02-06T13:44:50.943Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:UsernameToken
        wsu:Id="UsernameToken-10697954">
        <wsse:Username>apache</wsse:Username>
        <wsse:Password
          Type="http://...#PasswordText">password</wsse:Password>
        </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>
  <soapenv:Body/>
</soapenv:Envelope>
```



Apache WSS4J

- Implements WS Security Specification
- Builders / Processors
- Widely used
 - CXF, Rampart, Spring WS



WS-Policy

- General framework for endpoints to express requirements
- Basic operators
 - `wsp:All`
 - `wsp:ExactlyOne`
- Domain assertions
 - WS-SecurityPolicy Language
 - Just XML elements



WS-SecurityPolicy

- To express security requirements of a Web service according to the WS-Policy spec
 - What needs to be protected
 - What tokens to use
 - Algorithms, reference types, etc..
- Covers all WS-Sec* specifications



Assertion Types

- Protection assertions
- Token assertions
- Binding assertions
- Supporting token assertions
- Protocol assertions



Example

```
<wsp:All>  
  <sp:SupportingTokens>  
    <wsp:Policy>  
      <sp:UsernameToken/>  
    </wsp:Policy>  
  </sp:SupportingTokens>  
  <sp:SignedParts xmlns:sp="http://...securitypolicy">  
    <sp:Body/>  
  </sp:SignedParts>  
</wsp:All>
```

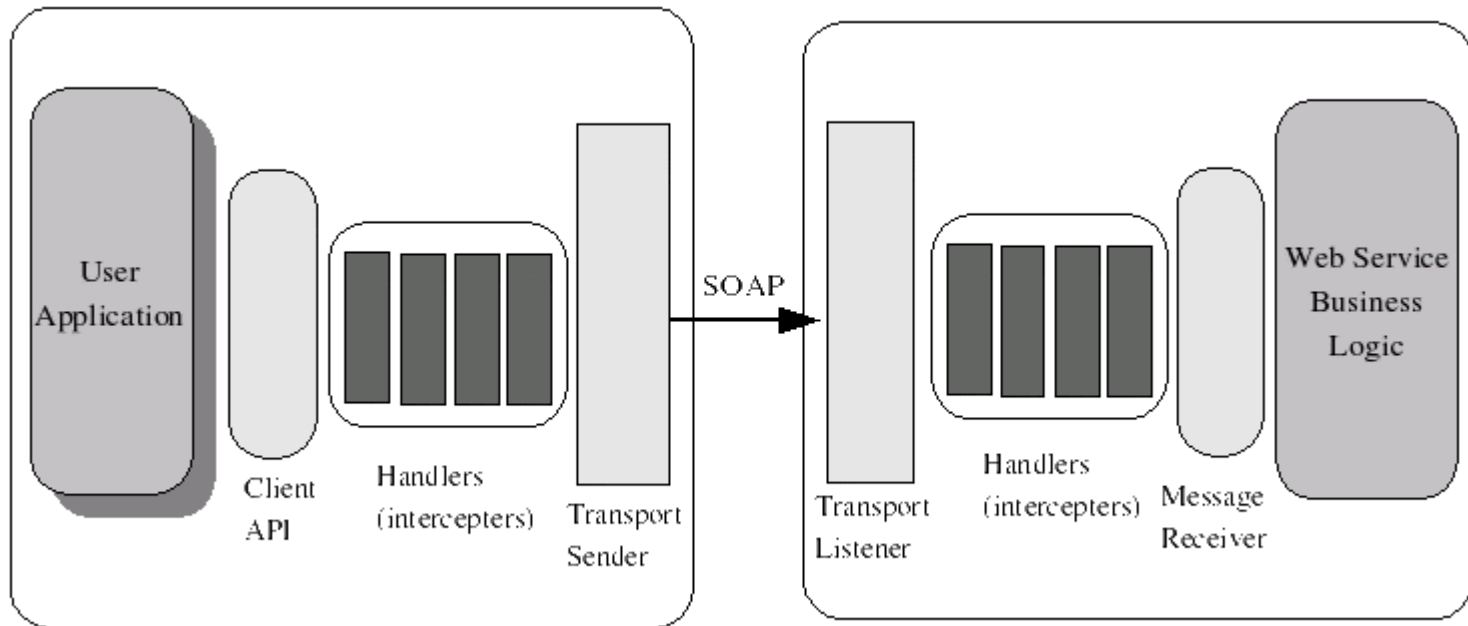


Apache Rampart / Rahas

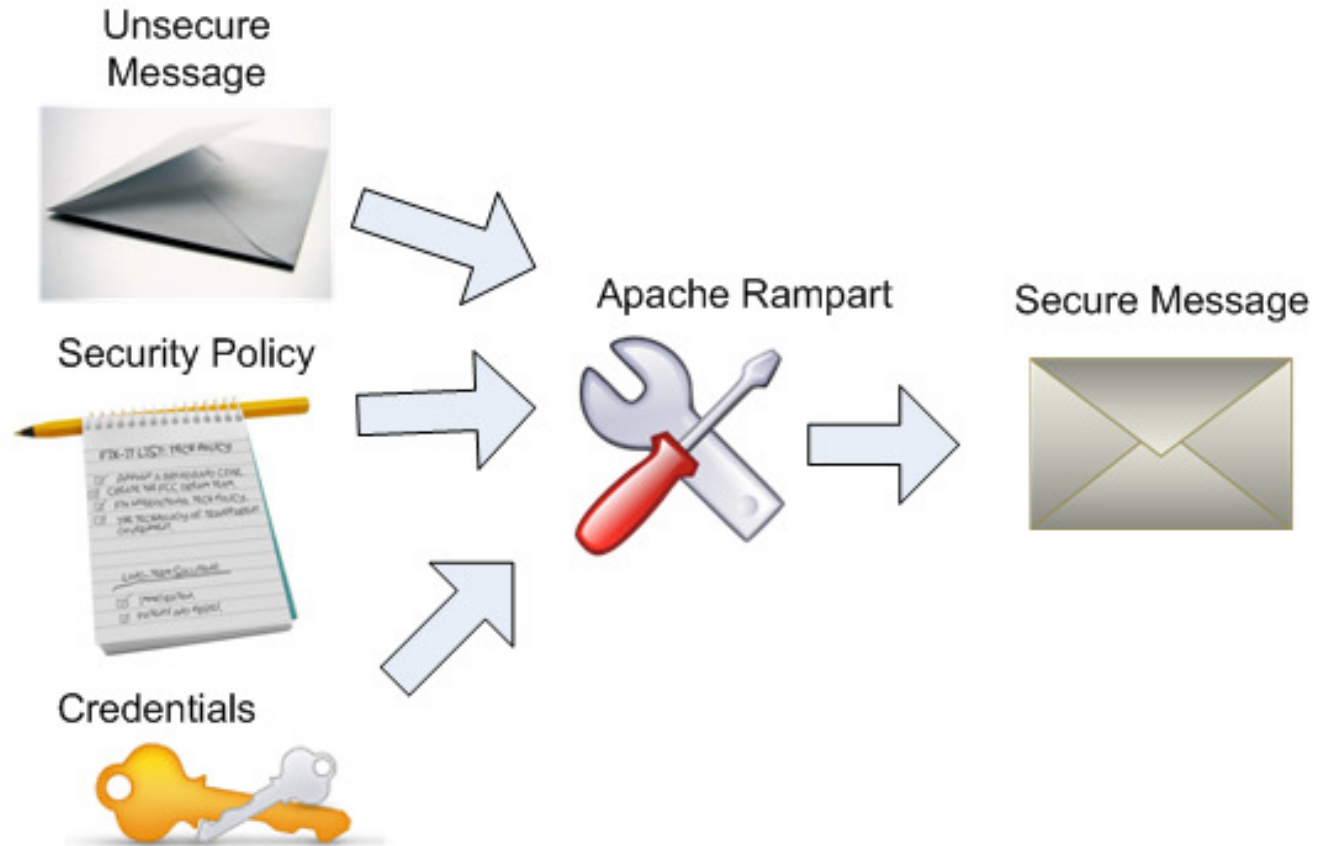
- Security modules of Axis2
- Supports WS- Sec*
 - WS Security
 - WS SecurityPolicy
 - WS SecureConversation
 - WS Trust



Out / In Security handlers



Apache Rampart



Securing a web service

- Deploying Rampart
 - Dependencies
 - Modules
 - Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files



Securing a web service

- Defining the policy
 - What to protect, how to protect
- Engaging Rampart
- Providing credentials
 - Username/passwords
 - keys



Calculator service

- CalculatorService
 - META-INF
 - services.xml (including the policy)
 - tutorial\rampart\service\CalculatorService.class
 - tutorial\rampart\service\PWCBHandler.class
 - services.jks

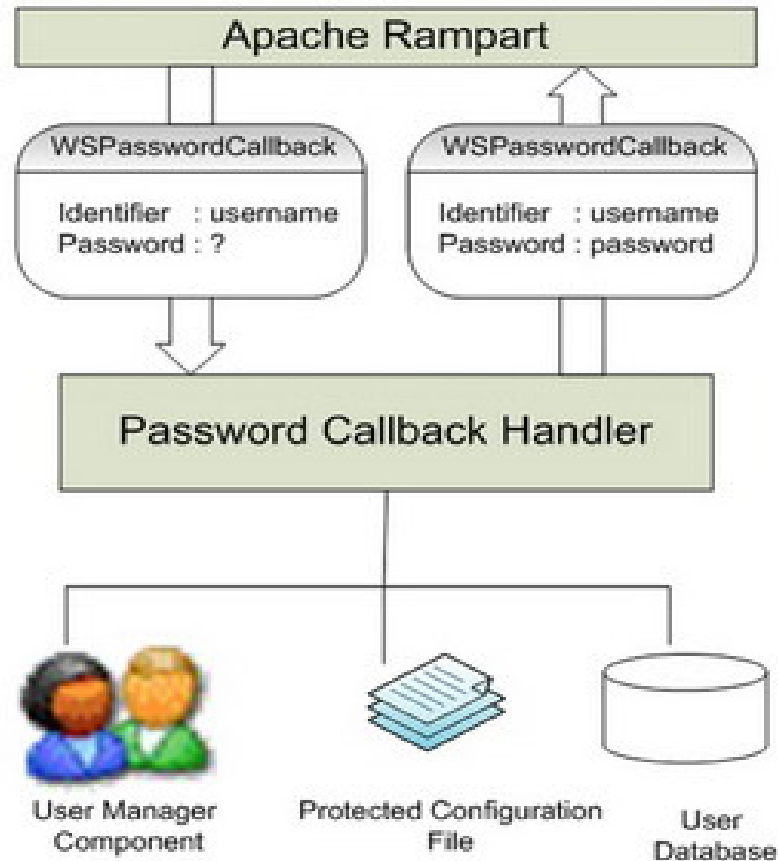


Demo

- http://people.apache.org/~nandana/apachecon_us_09/demo.zip

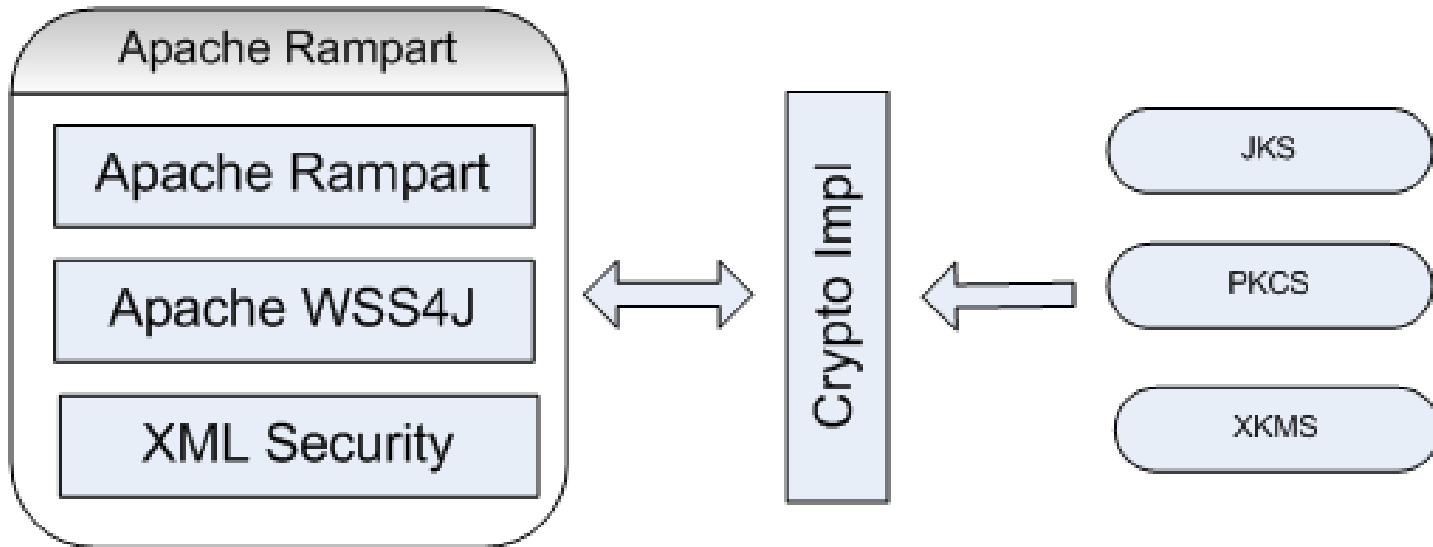


Password Callbacks





Keys



Securing a client

- Code generation
 - `axis2-1.5.1\bin\wsdl2java.sh`
- Policy
 - `WSDL → Stub`
- Engaging rampart
- Provide credentials



WS SecureConversation

- Performance problems
 - Asymmetric crypto operations
 - Password/Credential retrieval overhead
- Establishes a shared security context/session
 - Key/secret associated with the context
- Key Derivation



New Token Types

- SecurityContextToken (SCT)
 - Identifier
 - Associated key
- DerivedKeyToken (DKT)
 - Based on the security context token
 - derived keys used to secure actual payload messages



Apache Rahas

- Axis2 module for WS SecureConversation
- Handles the shared context (SCT) establishment and token derivation



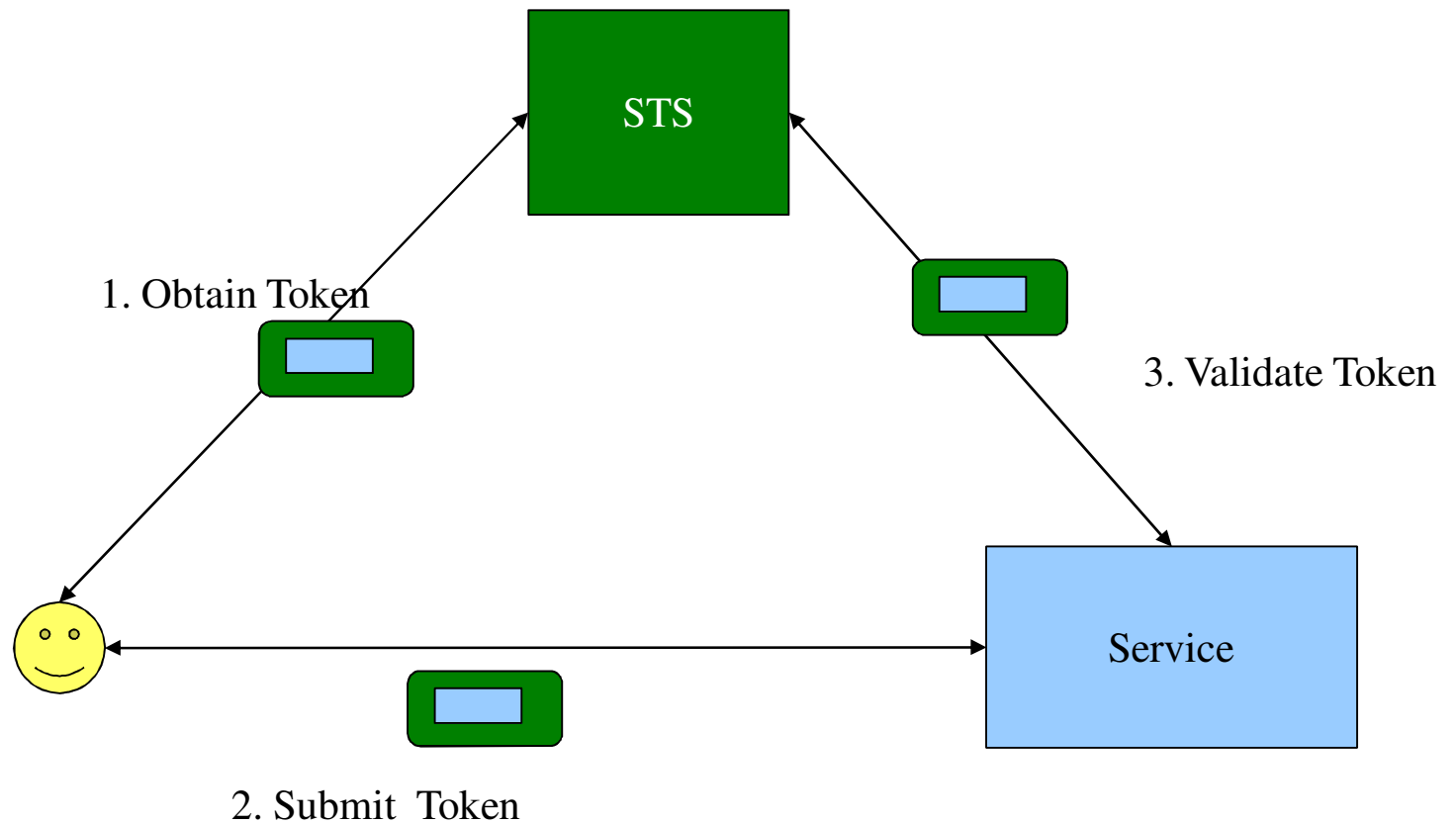
WS Trust

- Defines mechanisms for brokering trust
 - Still needs to bootstrap trust
- Passport
- Security Token Service





WS Trust



Future of Rampart

- Rampart 1.5 will be released soon
- Rampart 2



Getting involved

- <http://ws.apache.org/rampart/>
- rampart-dev@ws.apache.org

