WS-Policy

Sang Shin
Michèle Garoche
www.javapassion.com
"Learning is fun!"

Agenda

- What is and Why WS-Policy?
- WS-Policy terminology
- WS-Policy information model
- WS-Policy examples
- WS-Policy processing

What is & Why WS-Policy

What is WS-Policy?

- Defines syntax and semantics for service providers and clients to describe their requirements, preferences, and capabilities.
- The syntax provides a flexible and concise way of expressing the needs of each domain in the form of policies.
- A domain in this context is a generic field of interest that applies to the service. Example domains are
  - Security, Privacy
  - Reliable messaging
  - Application priorities

What is WS-Policy?

- Also defines processing models for these policies that operate independently of the domains.
- There are three defined operations for processing policies
  - Normalize
  - Merge
  - Intersect

How is WS-Policy Used?

- A Web service specifies, through WS-Policy, a set of rules that must be met by the client before it accesses the Web service.
- Clients that access the Web service check to see whether or not they can adhere to these policies.
  - Example: Web service has a policy “all messages be encrypted or signed in a certain way” - client cannot access the service without meeting this policy requirement
  - Example: Web service has a policy requiring that every message has to have a timestamp

WS-Policy Terminology

WS-Policy Terms

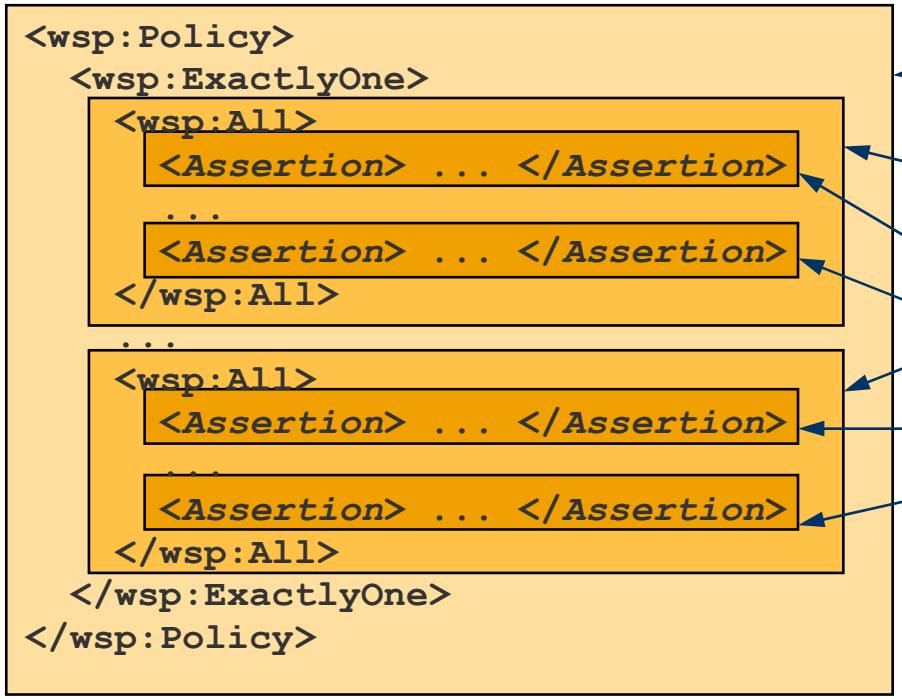
- Policy
  - A collection of policy alternatives
- Policy alternative
  - A collection of assertions
  - In normal form, a policy contains a list of policy alternatives specified in *wsp:All* tags
- Policy assertion
  - Represents a requirement or capability.
  - For example, a policy assertion could require that a certain type of encryption be used in encrypting transmitted data.

WS-Policy Terms

- Policy assertion type
  - A class of policy assertions
- Policy expression
  - An XML representation of a policy
- Policy subject
  - An entity with which a policy can be applied
  - Examples: an endpoint, a message, a resource, an interaction
- Policy scope
  - A set of policy subjects
- Policy attachment
  - A mechanism for associating policy with one or more policy scope

WS-Policy Information Model

Normal Form



„Policy Normal Form“

Policy

Collection of alternatives

(„pick one“)

Policy Alternative

Collection of assertions

(„do all“)

Policy Assertion

Domain-specific behavior

Compact Form – Nested Operators

<wsp:All> and <wsp:ExactlyOne> operators can be nested arbitrarily

- Transformation to normal form using set theory (commutativity, associativity, idempotency, distributivity)

Example

- <All> distributes over <ExactlyOne>

```
<All>
  <ExactlyOne>
    <Assertion1>
    <Assertion2>
  </ExactlyOne>
  <ExactlyOne>
    <Assertion3>
    <Assertion4>
  </ExactlyOne>
</All>
```

Is equivalent to ⇔

```
<ExactlyOne>
  <All><Assertion1><Assertion3></All>
  <All><Assertion1><Assertion4></All>
  <All><Assertion2><Assertion3></All>
  <All><Assertion2><Assertion4></All>
</ExactlyOne>
```

Compact Form – Optional Attribute

Assertions may carry the `wsp:Optional` attribute

- An alternative with and an alternative without the assertion
- Simplification of prior `wsp:Usage` attribute

Example

```
<Assertion wsp:Optional="true" >
  ...
</Assertion>
```

Is equivalent to ⇔

```
<ExactlyOne>
  <All>
    <Assertion>
      ...
    </Assertion>
  </All>
  <All />
</ExactlyOne>
```

Compact Form – Reference Mechanism

URI-based policy identification mechanism

- `wsu:ID` attribute is used to specify a fragment identifier

```
<wsp:Policy xml:base="http://fabrikam123.com/policies" wsu:Id="AUDIT" >
  <wssx:Audit wsp:Optional="true" />
</wsp:Policy>
```

`<PolicyReference>` element to reuse common expressions

-

```
<wsp:Policy>
  <wsp:PolicyReference URI="#AUDIT" />
  <wsse:SecurityToken>
    <wsse:TokenType>wsse:X509v3</wsse:TokenType>
  </wsse:SecurityToken>
</wsp:Policy>
```

```
<wsp:Policy>
  <wssx:Audit wsp:Optional="true" />
  <wsse:SecurityToken>
    <wsse:TokenType>wsse:X509v3</wsse:TokenType>
  </wsse:SecurityToken>
</wsp:Policy>
```

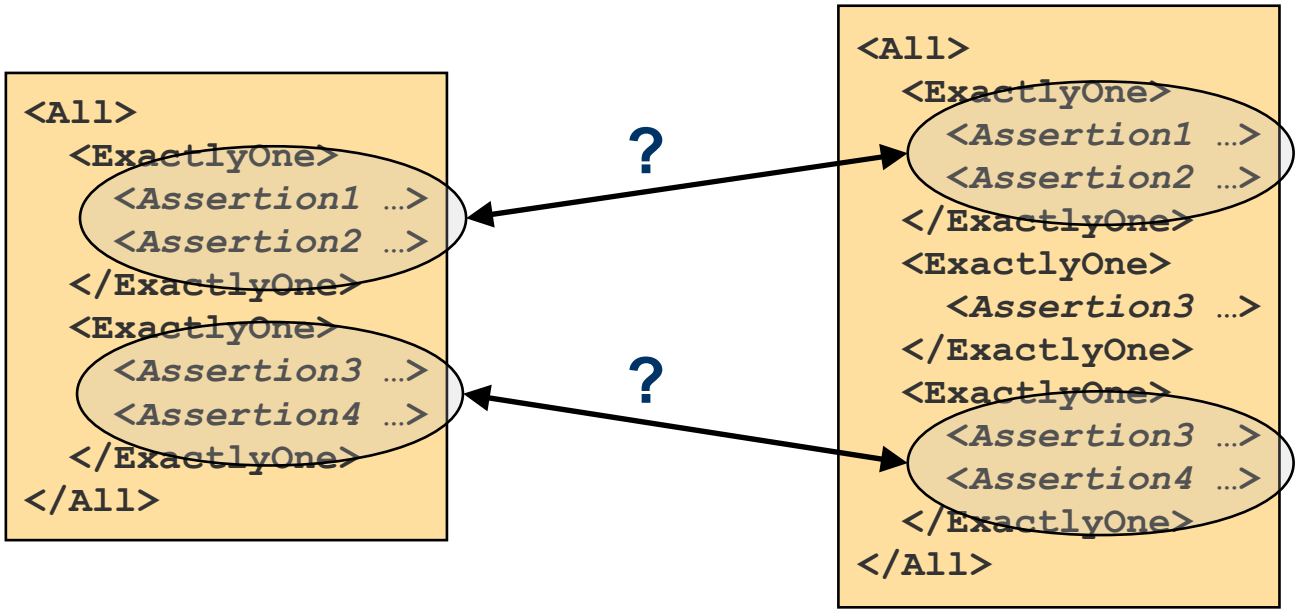
Intersection

Do two Web service endpoints have compatible policy?

- At design time to “wire together” compatible services
- At runtime to select compatible options

Compatibility of alternatives

- It is necessary that alternatives at least have the same assertion types
- Assertion-specific intersection needs to be determined per assertion type



WS-Policy Examples

Example: WS-Policy Document

- Specifies a policy with two policy alternatives
- Choose exactly one from the two policy alternatives
- All policy assertions specified in a policy alternative needs to be met

```
<wsp:Policy ...>
  <wsp:ExactlyOne>
    <wsp:All>                    ← Policy alternative 1
    </wsp:All>
    <wsp:All>                    ← Policy alternative 2
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Example WS-Policy: Security

- Policy contains a single policy assertion requiring the inclusion of a certain security token. If this policy were attached to a Web service, clients wanting to access the Web service are required to send security tokens of the type specified in the policy, X509v3 in this case.

```
<wsp:ExactlyOne>
  <wsp:All>
    <sec:SecurityToken>
      <sec:TokenType>sec:X509v3</sec:TokenType>
    </sec:SecurityToken>
  </wsp:All>
 </wsp:ExactlyOne>
</wsp:Policy>
```

Example WS-Policy: Reliable Messaging

```
<wsp:UsingPolicy></wsp:UsingPolicy>
<wsp:Policy wsu:Id="HelloPortBindingPolicy">
   <wsp:ExactlyOne>
      <wsp:All>
         <ns1:RMAssertion
   xmlns:ns1="http://schemas.xmlsoap.org/ws/2005/02/rm/policy">
         </ns1:RMAssertion>
         <ns2:Ordered xmlns:ns2="http://sun.com/2006/03/rm">
         </ns2:Ordered>
         <ns3:UsingAddressing

xmlns:ns3="http://www.w3.org/2006/05/addressing/wsdl">
         </ns3:UsingAddressing>
      </wsp:All>
   </wsp:ExactlyOne>
</wsp:Policy>
```

Thank you!

Sang Shin
Michèle Garoche
www.javapassion.com
“Learning is fun!”