XACML profile for Web-services

Working draft 03, 5 Sep 2003

Document identifier: draft-xacml-wspl-03.doc
Send comments to: xacml-comment@lists.oasis-open.org
Editors:
Tim Moses, Entrust (tim.moses@entrust.com)

Contributors:
Anne Anderson, Sun Microsystems
Seth Proctor, Sun Microsystems
Simon Godik, Overxeer

Abstract:
This working draft specifies a profile of XACML for expressing policy associated with Web-service end-points.

Status:
This version of the specification is a working draft of the committee. As such, it is expected to change prior to adoption as an OASIS standard.
If you are on the xacml@lists.oasis-open.org list for committee members, send comments there. If you are not on that list, subscribe to the xacml-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to xacml-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Copyright (C) OASIS Open 2003 All Rights Reserved.
# Table of contents

1. Introduction (non-normative) ......................................................... 4
   1.1 Glossary ................................................................................. 4
   1.2 Notation ................................................................................. 4
   1.3 Schema organization and namespaces ........................................ 5
   1.4 Background ............................................................................. 5
2. Model (Normative) ........................................................................... 5
3. Example (Non-normative) ............................................................... 9
4. Instructions to standards developers ............................................... 11
   4.1 Procedure (Normative) .......................................................... 11
   4.2 Example (Non-normative) ....................................................... 11
5. Definitions (Normative) ............................................................... 12
6. End-point policy combination (Normative) ....................................... 12
   6.1 Combine top-level <PolicySet> elements ................................... 13
   6.2 Combine second-level <PolicySet> elements ............................. 13
   6.3 Combine <Policy> elements ................................................... 13
   6.4 Combine <Rule> elements ...................................................... 13
   6.5 Combine <Apply> elements .................................................... 13
   6.6 Eliminate <Policy> elements .................................................. 15
   6.7 Substitute <Apply> elements .................................................. 16
   6.8 Result .................................................................................... 16
7. Security considerations (Non-normative) .......................................... 16
8. Bindings ....................................................................................... 17
   8.1 WSDL 1.1 (Normative) .......................................................... 17
      8.1.1. Introduction ................................................................. 17
      8.1.2. Attachment .................................................................. 17
      8.1.3. Structure .................................................................... 17
      8.1.4. Integrity/authenticity protection ..................................... 18
      8.1.5. Schema ...................................................................... 18
   8.2 WSDL 1.2 draft (Non-normative) .............................................. 20
   8.3 SOAP 1.1 (Normative) ............................................................ 21
      8.3.1. Introduction ................................................................. 21
      8.3.2. Structure .................................................................... 21
      8.3.3. Integrity/authenticity protection ..................................... 21
      8.3.4. Schema ...................................................................... 21
9. References (Non-normative) ........................................................... 22
   Appendix A. Worked example (Non-normative) ................................. 23
   Consumer policy ........................................................................... 23
   A.1.1. Plain-language policy ....................................................... 23
   A.1.2. XACML policy .................................................................. 23
   Combining process ........................................................................ 25
   draft-xacml-wspl-03.doc ................................................................ 2
<table>
<thead>
<tr>
<th>Page</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>67</td>
<td>A.1.3. Combine &lt;PolicySet&gt; elements</td>
</tr>
<tr>
<td>68</td>
<td>A.1.4. Combine &lt;Policy&gt; elements</td>
</tr>
<tr>
<td>69</td>
<td>A.1.5. Combine &lt;Rule&gt; elements</td>
</tr>
<tr>
<td>70</td>
<td>A.1.6. Combine &lt;Apply&gt; elements</td>
</tr>
<tr>
<td>71</td>
<td>A.1.7. Substitute &lt;Apply&gt; elements</td>
</tr>
<tr>
<td>72</td>
<td>Appendix B. Revision history</td>
</tr>
<tr>
<td>73</td>
<td>Appendix C. Notices</td>
</tr>
<tr>
<td></td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>36</td>
</tr>
</tbody>
</table>
1. Introduction (non-normative)

1.1 Glossary

**Aspect** – An independent set of technical features and parameters associated with use of a Web-service. In most cases, an aspect is identified with a single member of the suite of Web-service specifications for which policy provisions must be described, such as WS-Reliable Messaging or WS-Security. In the former case, policy provisions may include such items as: maximum time to live, maximum number of retries and minimum interval between retries.

**Authorized attribute** – An attribute whose value must be assigned by an authority, not a policy-user.

**Coincidence** – The property of pairs of predicates, strategies, objectives and end-point policies that enables them to be combined.

**Combiner** – An entity that combines two or more end-point policies.

**Constrained attribute** - An attribute whose value cannot be assigned by the policy-user.

**End-point policy** – 1. The set of provisions governing all aspects of a Web-service end-point. 2. A conjunctive set of objectives. 3. An XACML <PolicySet> element.

**Objective** – 1. The set of provisions governing a single aspect of a Web-service end-point. 2. A disjunctive list of strategies, in order of preference. 3. An XACML <Policy> element.

**Solution** – The set of features and parameter values that satisfy an end-point’s requirements for successful invocation.

**Strategy** – 1. One solution to a single aspect of a Web-service end-point. 2. A conjunctive set of predicates. 3. An XACML <Rule> element.

**Unconstrained attribute** - An attribute whose value can be assigned by the policy-user within a certain range.

1.2 Notation

This specification contains schema conforming to W3C XML Schema and normative text to describe the syntax and semantics of XML-encoded policy statements.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119]

"they MUST only be used where it is actually required for interoperability or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)"

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

Listings of schemas appear like this.

Example code listings appear like this.
Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

- The prefix `xacml:` stands for the XACML policy namespace.
- The prefix `xs:` stands for the W3C XML Schema namespace [XS].
- The prefix `xf:` stands for the XQuery 1.0 and XPath 2.0 Function and Operators specification namespace [XF].

This specification uses the following typographical conventions in text: `<XACMLElement>`, `<ns:ForeignElement>`, `Attribute`, `Datatype`, `OtherCode`. Terms in italic bold-face are intended to have the meaning defined in the Glossary of this document or [XACML v1.0].

### 1.3 Schema organization and namespaces

The XACML policy syntax is defined in a schema associated with the following XML namespace:

```
urn:oasis:names:tc:xacml:1.0:policy
```

### 1.4 Background

Access to a standard-conformant Web-service end-point involves a number of aspects, such as: reliable messaging, privacy, authorization, trust, authentication and cryptographic security. Each aspect addresses a number of optional features and parameters, which must be coordinated between communicating end-points if the service invocation is to be successful. The provider and consumer of the service likely have different preferences amongst the available choices of features and parameters. Therefore, a mechanism is required by which end-points may describe the mandatory features of service invocation, optional features that they support and the order of their preference amongst such features. Additionally, a procedure is required for combining and reducing these feature descriptions into a service invocation instance that respects both end-points’ requirements. These requirements are explained in [WSPL Req].

This specification defines a profile of XACML that enables it to be used for describing policy associated with Web-service end-points and using them in a successful invocation.

### 2. Model (Normative)

In this profile, an XACML `<PolicySet>` element is associated with a concrete Web-service end-point definition. To that end, its `<Target>` element MUST identify the WSDL 1.1 port whose features and parameters it describes. In the case that a policy must be targeted more finely than a port, a second level of `<PolicySet>` whose `<Target>` element identifies the port’s operations and messages MUST be inserted. The `<PolicySet>` elements MUST contain `<Policy>` elements that define the objective of each aspect of policy associated with the port.

An XACML `<Policy>` element is associated with a single aspect of an end-point policy. The `<Target>` element of a `<Policy>` MUST identify the one objective of the end-point policy to which it applies. Developers of Web-service specifications that make use of XACML MUST define a name and type for its objective. In order for an end-point to be successfully invoked, all of its objectives MUST be achieved by the service invocation. The `<Policy>` element MUST contain `<Rule>` elements that define acceptable alternative strategies for achieving the objective.
An XACML `<Rule>` element MUST describe one alternative strategy for achieving an objective. At least one strategy MUST be successful if its objective is to be achieved. The lexical order of the strategies in the objective SHOULD reflect the policy-writer’s preferences. For example, the policy writer’s preferred strategy should appear first. The `<Rule>` element MUST contain a set of `<Apply>` elements that define predicates.

An XACML `<Apply>` element MUST contain exactly one predicate. All predicates MUST be satisfied by a service invocation if the associated strategy is to be successful.

An `<Apply>` element SHALL NOT contain another `<Apply>` element. It is RECOMMENDED that `<Apply>` elements be structured as follows:

```xml
<Apply functionId="...">
  <AttributeSelector RequestContextPath="..." DataType="..."/>
  <AttributeValue DataType="..."> ... </AttributeValue>
</Apply>
```

In cases where the policy constrains the relationship between attribute values, as opposed to the literal value of an attribute, it will be necessary to substitute a second `<AttributeSelector>` element for the `<AttributeValue>` element in the above fragment. The order of the `<AttributeSelector>` element and the `<AttributeValue>` element in the above fragment MAY be reversed to achieve the required constraint if the applied function has no inverse (e.g. subset). Any of the following elements MAY be used in place of the `<AttributeSelector>` element in either position: `<SubjectAttributeDesignator>`, `<ResourceAttributeDesignator>`, `<ActionAttributeDesignator>` or `<EnvironmentAttributeDesignator>`.

The relevant portion of the WSDL 1.1 data model is hierarchical, as shown in Figure 1.

```
Port
  Operation
  Message
```

Figure 1 - WSDL 1.1 hierarchical data model

This structure is reflected in the end-point policy model, as shown in Figure 2.

The name attribute values of objects in the WSDL 1.1 model SHALL be used in `<Target>` elements of the end-point policy to associate policy statements with those objects. The names SHALL be matched using string equality. Nevertheless, a `<Target>` element used to associate a policy statement with a non-root object in the WSDL 1.1 model is intended to identify the object within the context established by the `<Target>` elements of its enclosing `<PolicySet>` element(s). So, target matching SHALL be performed on the set of objects that has been successively refined by the enclosing layers of the end-point policy.
This model has been chosen to facilitate combining of end-point policies.

The following consequences flow from the model:


2. The contents of all <PolicySet/Target/Subjects> elements SHALL be <AnySubject/>.

3. The contents of the top-level <PolicySet/Target/Resources> element SHALL be the name attribute of the end-point’s port definition.

4. The MatchId for the <PolicySet/Target/Resources> element SHALL be “urn:oasis:names:tc:xacml:1.0:function:string-equal”.

5. The contents of the top-level <PolicySet/Target/Actions> element SHALL be <AnyAction/>.

6. If present, the contents of the second-level <PolicySet/Target/Resources> element SHALL either be the name attribute of the end-point’s message definition or the element <AnyResource/>.

7. In the case that the <PolicySet/Target/Resources> element is the name attribute, the MatchId SHALL be “urn:oasis:names:tc:xacml:1.0:function:string-equal”.
8. If present, the contents of the second-level <PolicySet/Target/Actions> element SHALL be the name attribute of the end-point's operation definition or the element <AnyAction/>.

9. In the case that the <PolicySet/Target/Actions> element is the name attribute, the MatchId SHALL be "urn:oasis:names:tc:xacml:1.0:function:string-equal".

10. If the contents of the second-level <PolicySet/Target/Resources> element is the element <AnyResource/>, then the contents of the <PolicySet/Target/Actions> element SHALL NOT be the element <AnyAction/>, and vice-versa. Otherwise, its <Policy> elements should be placed immediately subordinate to the top-level <PolicySet> element.


12. The MatchId for the <Policy/Target/Resources> element SHALL be "urn:oasis:names:tc:xacml:1.0:function:anyURI-equal".

13. The Effect attribute of all <Rule> elements SHALL be "Permit".

14. The contents of the <Policy/Target/Subjects> element SHALL be <AnySubject/>.

15. The contents of the <Policy/Target/Resources> element SHALL be <AnyResource/>.

16. The contents of the <Policy/Target/Actions> element SHALL identify the objective (see Section 4).

17. The <Rule/Target> element SHALL be omitted.

18. The FunctionId attribute of a <Condition> element SHALL be "urn:oasis:names:tc:xacml:1.0:function:and".

19. The FunctionId attribute of an <Apply> element SHALL identify one of the matching functions specified in XACML.

In order to be considered conformant with this profile, a <PolicySet> element MUST satisfy all of these conditions.

Predicates express constraints on attributes. Attributes fall into three classes:

- Unconstrained attributes,
- Constrained attributes and
- Authorized attributes.

An unconstrained attribute is one whose value can be assigned by the policy-user. For instance, the minimum time between re-transmissions of an unacknowledged message is an attribute that should be under the control of the sender (within certain limits). This is, therefore, in the class of unconstrained attributes.

A constrained attribute, on the other hand, is one whose value is outside the control of the policy-user. This may be because it is an environmental attribute or a subject attribute whose value is assigned by someone other than the policy-user. The emergency condition code is an example of an environmental attribute over which a policy-user has no control; if this attribute is used in a predicate, then the predicate either evaluates "True" or "False", regardless of any action that the policy-user might take. An example of a subject attribute over which the policy-user has no control is his or her status in a customer loyalty program. If this attribute is used in a predicate, then the predicate either evaluates "True" or "False", regardless of any action that the
policy-user might take. Some constrained attributes vary with time either in a predictable or unpredictable manner. In the case of the environmental attribute “time”, it will never again adopt values in the past, whereas, values in the future will arise in a predictable manner. In this case, the policy-user may choose to wait until the predicate involving time evaluates “True”.

An authorized attribute is one whose value has to be asserted by an authority, for instance the policy-user’s role. While the other party will not accept the policy-user’s own assertion that he or she occupies a particular role, the policy-user may be able to take action to obtain the necessary assertion about the attribute from a suitable authority.

3. Example (Non-normative)

This section contains an example of a service-provider policy on the aspect of data-rate allocation.

Here is a plain-language description of the policy.

Clients paying $150/minute are allocated a guaranteed minimum data-rate of 64kb/s.

Clients paying $45/minute are allocated a guaranteed minimum data-rate between 6pm and midnight of 40kb/s.

In order to make the example somewhat easier to read, several abbreviations have been introduced. For instance:

The  element has been omitted from all the  elements.

Only  elements have been retained in  elements.

URIs have been abbreviated.

“*one-and-only” bag functions have been omitted around  elements in  elements.

DataType and FunctionId prefixes have been omitted. A reader familiar with XACML should be able to reconstruct a syntactically correct policy from the information provided.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet PolicySetId="Provider Policy"
    PolicyCombiningAlgId="deny-overrides">
    <Target>
        <Resources>
            <ResourceMatch MatchId="equal"
                <AttributeValue DataType="anyURI">
                    serviceX:portX
                </AttributeValue>
                <ResourceAttributeDesignator AttributeId="portID"
                    DataType="anyURI"/>
            </ResourceMatch>
        </Resources>
    </Target>
    <Actions>
        <AnyAction/>
    </Actions>
</Policy>
```

<ActionMatch MatchId="equal">
  <AttributeValue DataType="anyURI">
    data-rate-allocation
  </AttributeValue>
  <ActionAttributeDesignator AttributeId="objectiveId" DataType="anyURI"/>
</ActionMatch>
</Actions>
</Target>
</Rule>
</Policy>
</PolicySet>

<Rule RuleId="Provider Data-Rate Allocation Objective Strategy 1" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="equal">
      <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
      <AttributeValue DataType="integer">
        150
      </AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
      <AttributeValue DataType="integer">
        64000
      </AttributeValue>
    </Apply>
  </Condition>
</Rule>

<Rule RuleId="Provider Data-Rate Allocation Objective Strategy 2" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="equal">
      <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
      <AttributeValue DataType="integer">
        45
      </AttributeValue>
    </Apply>
    <Apply FunctionId="equal">
      <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
      <AttributeValue DataType="integer">
        40000
      </AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <EnvironmentAttributeDesignator DataType="time" AttributeId="timeOfDay"/>
      <AttributeValue DataType="time">
        18:00
      </AttributeValue>
    </Apply>
  </Condition>
</Rule>
4. Instructions to standards developers

Developers of Web-services standards that are intended to conform with this profile MUST define standard-specific policy parameters.

4.1 Procedure (Normative)

Developers of Web-services standards MUST complete the following steps.

1. Assign a URI for at least one objectiveId attribute. In the event that the specification document-identifier is a URI, it MAY be used as the objectiveId URI.

2. Define a set of attribute names, types and semantics. Classify the attributes as unconstrained, constrained or authorized.

3. Select one or more matching functions on the attributes from the matching functions defined in [XACML]. The functions MUST be type-consistent with the attributes. For every individual attribute, its associated matching functions MUST be combinable, as defined in Table 1. It is STRONGLY RECOMMENDED to use type-greater-than-or-equal or type-less-than-or-equal matching functions in preference to type-greater-than or type-less-than matching functions, respectively. If it is, nonetheless, necessary to use type-greater-than or type-less-than matching functions, then ceiling and floor operations (respectively) MUST be defined for the corresponding attribute. This merely involves defining a resolution for the attribute value. For instance, the attribute “minimum time between re-transmissions of an unacknowledged message” may be assigned a resolution of 1 minute. Then, if this attribute were to be used as the second operand in a duration-greater-than function, the ceiling operation on this attribute would return the shortest value greater than the specified value with a resolution of 1 minute.

These attributes and functions MAY be used in predicates.

4.2 Example (Non-normative)

A committee defining the reliable messaging aspect of Web-service invocation might assign the URI:

urn:oasis:names:tc:wsrm:1.0:objectiveId

as the objectiveId.

It might identify the maximum-time-to-live attribute as a parameter of policy. It might assign the identifier:

urn:oasis:names:tc:wsrm:1.0:maximum-time-to-live

to this attribute. Then it might identify the attribute type to be

It might define its meaning to be the maximum value permitted to be assigned by the requestor to the “time-to-live” parameter associated with a service request. Because the attribute value can be assigned by the requestor, this is an unconstrained attribute.

Then it might identify

\[ \text{urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than-or-equal} \]

as the matching function associated with the attribute. Because this function is neither a type-greater-than nor a type-less-than matching function, there is no need to define a ceiling or floor operation.

The committee MUST specify all relevant parameters in a similar way.

---

## 5. Definitions (Normative)

This profile defines one attribute.

Name: urn:oasis:names:tc:xacml:1.0:attribute:objectiveId.

Type: xs:anyURI.

Meaning: the value of this attribute indicates the aspect of policy addressed by a *Policy* element. The

\[ \text{Policy/Target/Actions/ActionMatch/ActionAttributeDesignator/@AttributeId} \]

attribute MUST be assigned this value.

---

## 6. End-point policy combination (Normative)

The need to combine two or more policies is described in [WSPL Req].

The procedure for combining two top-level *PolicySet* elements is described here. More than two *PolicySet* elements MAY be combined by repeating this procedure. Alternative procedures that achieve the same result under all circumstances SHALL be considered conformant.

The combining procedure involves combining coincident top-level *PolicySet* elements, then combining coincident second-level *PolicySet* elements within the combined top-level *PolicySet* elements, then combining coincident *Policy* elements within the combined

\[ \text{Policy} \]

elements and finally combining coincident *Rule* elements within the combined

\[ \text{Rule} \]

elements. Finally, elimination and substitution steps are applied.

The detailed steps are described below.

The effect of this procedure is to identify a single *Rule* element for each objective that represents the contract between the parties. The contract is compatible with both of the original end-point policies, while reflecting the preferences of the combiner.
6.1 Combine top-level <PolicySet> elements

Combine coincident top-level <PolicySet> elements. <PolicySet> elements are coincident if and only if their <Target> elements are identical.

In order to combine two top-level <PolicySet> elements, append the foreign <Policy> and second-level <PolicySet> elements to the combiner's <Policy> and second-level <PolicySet> elements and assign a new unique PolicySetId attribute.

6.2 Combine second-level <PolicySet> elements

If second-level <PolicySet> elements are present, then all coincident pairs of these MUST be combined in the same way. If a second-level <PolicySet/Target/Resources> element contains the <AnyResource/> element, then it is coincident with another second-level <PolicySet> element if and only if their <Target/Actions> elements are identical. The converse is the case if the <AnyAction> element is present.

6.3 Combine <Policy> elements

Within the resulting <PolicySet> elements, combine all coincident <Policy> elements. <Policy> elements are coincident if and only if their <Target> elements are identical.

In order to combine two <Policy> elements, append the foreign <Rule> elements to the combiner's <Rule> elements and assign a new unique PolicyId attribute.

6.4 Combine <Rule> elements

Within each resulting <Policy> element, combine <Rule> elements in all possible pairings, taking one from the combiner's set and one from the foreign set. The combiner's first <Policy> element SHOULD be paired with each of the foreign <Policy> elements, starting with the first, then the combiner's second <Policy> element SHOULD be paired with each of the foreign <Policy> elements, etc.. This procedure respects the preferences of each policy writer, while giving priority to those of the combiner.

In order to combine two <Rule> elements, append the <Apply> elements from the foreign <Rule> element to the combiner's <Apply> elements and assign a new unique RuleId attribute.

6.5 Combine <Apply> elements

Within each resulting <Rule> element, combine all coincident <Apply> elements. <Apply> elements are coincident if they constrain the same attribute. If there exists no attribute value for which both <Apply> elements evaluate to "True", then their strategies are incompatible and the <Rule> element MUST be discarded. The test for compatible strategies is shown in the third column of Table 1. If no <Rule> elements remain, then the procedure SHALL terminate in failure. Note that in the case where the same attribute is constrained by different aspects, this procedure will not detect incompatible constraints.

Coincident <Apply> elements SHALL be combined as shown in the fourth column of Table 1.

Table 1 is to be interpreted according to the following key.

1. Columns one, two and four contain shorthand versions of an XACML <Apply> element. The portion before the open parenthesis (e.g. “type-equal” in the first row) represents the
<Apply> element’s FunctionId attribute value. The “type-” portion represents any of
the type-specific parts of the standard XACML function identifiers.

2. Alphabetic symbols (e.g. “a” in the first row) represent XACML
    <AttributeDesignator>, <AttributeSelector> or <AttributeValue>
elements.

3. Where N/A appears in the fourth column there is no single replacement <Apply>
element: the predicates are compatible, but not combinable. In these cases, the original
    <Apply> elements MUST NOT be modified by this step in the procedure.

4. ∩ means set intersection.

5. ⊆ means “is a proper subset of”.

<table>
<thead>
<tr>
<th>First &lt;Apply&gt; element</th>
<th>Second &lt;Apply&gt; element</th>
<th>Compatible strategies</th>
<th>Replacement &lt;Apply&gt; element</th>
</tr>
</thead>
<tbody>
<tr>
<td>type-equal(a,b)</td>
<td>type-equal(a,c)</td>
<td>b == c</td>
<td>type-equal(a,b)</td>
</tr>
<tr>
<td>type-equal(a,b)</td>
<td>type-greater-than(a,c)</td>
<td>b &gt; c</td>
<td>type-equal(a,b)</td>
</tr>
<tr>
<td>type-equal(a,b)</td>
<td>type-greater-than-or-equal(a,c)</td>
<td>b ≥ c</td>
<td>type-equal(a,b)</td>
</tr>
<tr>
<td>type-equal(a,b)</td>
<td>type-less-than(a,c)</td>
<td>b &lt; c</td>
<td>type-equal(a,b)</td>
</tr>
<tr>
<td>type-equal(a,b)</td>
<td>type-less-than-or-equal(a,c)</td>
<td>b ≤ c</td>
<td>type-equal(a,b)</td>
</tr>
<tr>
<td>type-greater-than(a,b)</td>
<td>type-greater-than(a,c)</td>
<td></td>
<td>type-greater-than(a,max(b,c))</td>
</tr>
<tr>
<td>type-greater-than(a,b)</td>
<td>type-greater-than-or-equal(a,c)</td>
<td>Where b ≥ c</td>
<td>type-greater-than(a,b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Where b &lt; c</td>
<td>type-greater-than-or-equal(a,c)</td>
</tr>
<tr>
<td>type-greater-than-or-equal(a,b)</td>
<td>type-greater-than-or-equal(a,c)</td>
<td></td>
<td>type-greater-than-or-equal(a,max(b,c))</td>
</tr>
<tr>
<td>type-less-than(a,b)</td>
<td>type-less-than(a,c)</td>
<td></td>
<td>type-less-than(a,min(b,c))</td>
</tr>
<tr>
<td>type-less-than(a,b)</td>
<td>type-less-than-or-equal(a,c)</td>
<td>Where b &gt; c</td>
<td>type-less-than-or-equal(a,c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Where b ≤ c</td>
<td>type-less-than(a,b)</td>
</tr>
<tr>
<td>type-less-than-or-equal(a,b)</td>
<td>type-less-than-or-equal(a,c)</td>
<td></td>
<td>type-less-than-or-equal(a,min(b,c))</td>
</tr>
<tr>
<td>type-greater-than(a,b)</td>
<td>type-less-than(a,c)</td>
<td>b &lt; c</td>
<td>N/A</td>
</tr>
<tr>
<td>type-greater-than(a,b)</td>
<td>type-less-than-or-equal(a,c)</td>
<td>b &lt; c</td>
<td>N/A</td>
</tr>
<tr>
<td>predicate combination</td>
<td>equal(a,c)</td>
<td>type-less-than(a,c)</td>
<td>b &lt; c</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>---------------------</td>
<td>-------</td>
</tr>
<tr>
<td>type-greater-than-or-equal(a,b)</td>
<td>type-less-than(a,c)</td>
<td>b &lt; c</td>
<td>N/A</td>
</tr>
<tr>
<td>set-equals(a,b)</td>
<td>set-equals(a,c)</td>
<td>b == c</td>
<td>set-equals(a,b)</td>
</tr>
<tr>
<td>set-equals(a,b)</td>
<td>subset(a,c)</td>
<td>b ≤ c</td>
<td>set-equals(a,b)</td>
</tr>
<tr>
<td>subset(a,b)</td>
<td>subset(a,c)</td>
<td>∩ (b,c) ≠ 0</td>
<td>subset (a, (b,c))</td>
</tr>
</tbody>
</table>

Table 1 - Predicate combination

6.6 Eliminate <Policy> elements

Following combination, an elimination step MUST be applied. The <Rule> elements represent the available strategies in order of preference for each aspect. Ideally, the policy-user would adopt the first <Rule> element as its strategy for invoking the service. However, some strategies may place constraints on attributes that are not within the control of the policy-user. Such strategies MUST be eliminated.

Elimination proceeds by examining each <Apply> element, as described below.

1. If the <Apply> element places a literal constraint on a constrained attribute, then the policy-user SHALL test whether the constraint is satisfied by the attribute. If it is, then it SHALL proceed. If it is not, then the enclosing <Rule> element SHALL be eliminated.
2. If the <Apply> element places a literal constraint on an unconstrained attribute, then the policy-user SHALL assign a value to the attribute that satisfies the constraint. If the required value is not in the available range, then the enclosing <Rule> element SHALL be eliminated.
3. If the <Apply> element constrains the relationship between two constrained attributes, then the policy-user SHALL test whether the constraint is satisfied by the attributes. If it is, then it SHALL proceed. If it is not, then the enclosing <Rule> element SHALL be eliminated.
4. If the <Apply> element constrains the relationship between two unconstrained attributes, then the policy-user SHALL assign a value to one or both of the attributes that satisfies the constraint. If the required value is not in the available range, then the enclosing <Rule> element SHALL be eliminated.
5. If the <Apply> element constrains the relationship between a constrained attribute and an unconstrained attribute, then the policy-user SHALL assign a value to the unconstrained attribute that satisfies the constraint. If the required value is not in the available range, then the enclosing <Rule> element SHALL be eliminated.
6. If the <Apply> element constrains authorized attributes, then the policy-user SHALL obtain the required attribute from an acceptable authority. The strategy containing the attribute constraint should also indicate what constitutes an acceptable authority. If the required attribute cannot be obtained, then the enclosing <Rule> element SHALL be eliminated.

<Rule> elements MUST be examined in order until one survives the elimination procedure. This represents the highest preference strategy with which the policy-user is able to comply. Therefore, this (and only this) one SHALL be retained.
If, after completing the elimination step, no `<Rule>` elements remain, then the procedure SHALL terminate in failure.

### 6.7 Substitute `<Apply>` elements

Following elimination, a substitution step MAY be applied to the `<Apply>` elements of the remaining `<Rule>` element. Substitution proceeds by the following steps. Predicates that only express constraints between constrained attributes MAY be eliminated, as it has been determined by the previous step that these evaluate “True”. The substitutions shown in Table 2 SHALL be applied.

<table>
<thead>
<tr>
<th><code>&lt;Apply&gt;</code> element</th>
<th>Replacement <code>&lt;Apply&gt;</code> element</th>
</tr>
</thead>
<tbody>
<tr>
<td>type-greater-than(a,b)</td>
<td>type-equal(a, ⌈b)</td>
</tr>
<tr>
<td>type-greater-than-or-equal(a,b)</td>
<td>type-equal(a,b)</td>
</tr>
<tr>
<td>type-less-than(a,b)</td>
<td>type-equal(a, ⌈b)</td>
</tr>
<tr>
<td>type-less-than-or-equal(a,b)</td>
<td>type-equal(a,b)</td>
</tr>
<tr>
<td>type-subset(a,b)</td>
<td>set-equals(a,b)</td>
</tr>
</tbody>
</table>

Table 2 – Substitution procedure

Where ⌈ represents the ceiling operation defined for the attribute and ⌊ represents the floor operation.

In the case of a strategy that contains compatible, but non-combinable, predicates (see note to Table 1) the `<Rule>` element will contain more than one `<Apply>` element constraining the same attribute. In such cases, all but one of these `<Apply>` elements MUST be eliminated. The choice of element to retain is left to the implementer. However, it is RECOMMENDED to retain the final one, as this gives priority to the combiner’s preference.

In the case that one `<Apply>` element expresses a relational constraint between two attributes, and another `<Apply>` element expresses a literal constraint on one of those attributes, then the value of this attribute, as dictated by the literal constraint, MAY be substituted for its designator in the other `<Apply>` element. This procedure MAY be applied recursively until as many of the relational constraints as possible have been replaced by literal constraints.

### 6.8 Result

The result of this procedure is a set of strategies, one for each aspect of policy, and each containing value assignments for attributes that are under the control of the policy-user. A service invocation using these attribute assignments conforms with the applicable policy of both the consumer and the provider.

### 7. Security considerations (Non-normative)

Policies must be integrity protected. The policy-user must confirm that the author of the policy is an entity that is authoritative for the target end-point. How this is achieved is outside the scope of this specification.
8. Bindings

<PolicySet> elements MAY be distributed in a [WSDL 1.1] or WSDL 1.2 service description or in a [SOAP 1.1] message. When they are distributed by one of these means, they MUST be distributed as defined in this section.

8.1 WSDL 1.1 (Normative)

This section defines how <PolicySet> elements SHALL be included in a WSDL 1.1 service description for a Web-service end-point.

8.1.1 Introduction

As a precursor to invoking a WSDL 1.1 operation of a WSDL 1.1 port, certain consumer configuration steps are likely to be required, and these configuration steps are likely to be associated with the port, rather than with an individual operation. Locating, retrieving, validating and combining policy are appropriate functions to perform as one of these configuration steps.

Different aspects of policy may be most applicable to different objects within the WSDL 1.1 data model, see Figure 1. For instance, privacy policy may apply to a WSDL 1.1 message definition, regardless of which WSDL 1.1 operation uses the message. Crypto-security policy, on the other hand, may apply to a message definition, differently, according to which operation uses the message. And, trust policy may apply to the port, independent of which operation or message is used.

8.1.2 Attachment

For the reasons stated in Section 8.1.1, a top-level <PolicySet> element SHALL be targeted only at a WSDL 1.1 port. However, it MUST be possible to associate a policy statement with any object (port, operation or message) either alone or in combination, see Figure 2. For this reason, policy statements MUST be capable of differentiating between the various WSDL 1.1 operation and message definitions of the WSDL 1.1 port at which they are targeted.

The WSDL 1.1 schema requires that <wsdl/port>, <wsdl/operation> and <wsdl/message> elements have a name attribute of type NCName. This attribute is used to associate policies with a particular port, operation or message or combinations thereof. URLs are a form of NCName.

8.1.3 Structure

Conformant <PolicySet> elements SHALL be structured as follows:

The top-level element SHALL be a <PolicySet> element whose <PolicySet/Target/Resources> element identifies the WSDL 1.1 port to which it is applicable, by means of the wsdl/port@name attribute.

Policies that apply to the WSDL 1.1 port, regardless of the particular operation or message SHALL be contained in <Policy> elements immediately subordinate to the top-level <PolicySet> element.

Policies that apply to some combination of WSDL 1.1 port, operation and message SHALL be contained in <PolicySet> elements subordinate to the top-level <PolicySet> element.

These second-level <PolicySet> elements SHALL have <PolicySet/Target/Actions> elements that identify the WSDL 1.1 operation, and <PolicySet/Target/Resources> elements.
elements that identify the WSDL 1.1 message to which they are applicable, by means of the
wsdl/operation@name and wsdl/message@name attributes, respectively. Only WSDL 1.1
message definitions of the “input” type SHALL be identified.

The <Policy/Target/Resources> element SHALL identify the aspect of policy to which it
applies.

8.1.4. Integrity/authenticity protection

If the <wsdl/definitions> element is integrity-protected, then the <PolicySet> elements
SHOULD be included within the integrity-protection of that element.

Where it is not possible to do this, either because the <wsdl/definitions> element is not
integrity-protected, or for other reasons, <PolicySet> elements SHALL be enclosed in a
<saml/Assertion> element wrapper [SAML]. This allows supporting information, such as the
saml/Assertion@Issuer attribute to be attached. The <saml/Assertion> element SHALL
be integrity-protected.

The policy-user SHALL ignore the PolicySet@PolicySetId attribute.

The WSDL 1.1 port to which a policy applies SHALL be identified in the top-level
<PolicySet/Target/Resources> element, by means of the wsdl/port@name attribute.

The policy-user SHALL confirm that it has located the correct policy by examining the policy’s top-
level <PolicySet/Target/Resources> element. Furthermore, if they are present, the policy-
user SHALL confirm that the policy is current, by examining the
saml/Assertion/Conditions@NotBefore and
saml/Assertion/Conditions@NotOnOrAfter attributes.

The wsdl/port@name attribute SHALL contain a URL. In the case where a policy is wrapped in
a <saml/Assertion>, the host and domain parts of the wsdl/port@name URL SHALL be
identical to the saml/Assertion@Issuer attribute value. The saml/Assertion@Issuer
attribute value SHALL be identical to the CN attribute value in the subject field of the certificate
[X509] that validates the <saml/Assertion> element, whether integrity protection is provided
by SSL or XML Digital Signature.

8.1.5. Schema

A <PolicySet> element SHALL be included in a <wsdl/definitions> element in
accordance with the following schema. Additions to the WSDL 1.1 SOAP binding are highlighted.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://schemas.xmlsoap.org/wsdl/policy-conformant-soap/
xmlns:xacml="urn:oasis:names:tc:xacml:1.0:policy">
<import namespace="urn:oasis:names:tc:xacml:1.0:policy"
schemaLocation="http://www.oasis-open.org/committees/download.php/915/cs-xacml-schema-policy-
xmlns:xacml="urn:oasis:names:tc:xacml:1.0:policy">
<element name="EndPointPolicy" type="xacml:PolicySetType"/>
<complexType name="bindingType">
<attribute name="transport" type="anyURI" use="optional"/>
</complexType>
</schema>
```
<complexType name="styleChoice">
  <restriction base="string">
    <enumeration value="rpc"/>
    <enumeration value="document"/>
  </restriction>
</complexType>

<element name="operation" type="policy-conformant-soap:operationType">
  <complexType name="operationType">
    <attribute name="soapAction" type="anyURI" use="optional"/>
    <attribute name="style" type="policy-conformant-soap:styleChoice" use="optional"/>
  </complexType>
</element>

<element name="body" type="policy-conformant-soap:bodyType">
  <complexType name="bodyType">
    <attribute name="encodingStyle" type="anyURI" use="optional"/>
    <attribute name="parts" type="NMTOKENS" use="optional"/>
    <attribute name="use" type="policy-conformant-soap:useChoice" use="optional"/>
    <attribute name="namespace" type="anyURI" use="optional"/>
  </complexType>
</element>

<element name="fault" type="policy-conformant-soap:faultType">
  <complexType name="faultType">
    <complexContent>
      <restriction base="policy-conformant-soap:bodyType">
        <attribute name="parts" type="NMTOKENS" use="prohibited"/>
      </restriction>
    </complexContent>
  </complexType>
</element>

<element name="header" type="policy-conformant-soap:headerType">
  <complexType name="headerType">
    <all>
      <element ref="policy-conformant-soap:headerfault"/>
    </all>
    <attribute name="message" type="QName" use="required"/>
    <attribute name="parts" type="NMTOKENS" use="required"/>
    <attribute name="use" type="policy-conformant-soap:useChoice" use="required"/>
    <attribute name="encodingStyle" type="anyURI" use="optional"/>
    <attribute name="namespace" type="anyURI" use="optional"/>
  </complexType>
</element>

<element name="headerfault" type="policy-conformant-soap:headerfaultType">
  <complexType name="headerfaultType">
  </complexType>
</element>
8.2 WSDL 1.2 draft (Non-normative)

Version 1.2 of WSDL is currently in draft form. Therefore, this specification does not provide a normative binding for XACML to WSDL 1.2. However, in the current draft of WSDL 1.2, the counterpart of the WSDL 1.1 port component is the WSDL 1.2 binding component (see Figure 3). Therefore, it is anticipated that, with the exception of swapping the roles of port and binding, the standard method of attaching a <PolicySet> to a WSDL 1.2 definition will be identical to the standard method of attaching a <PolicySet> to a WSDL 1.1 definition (see Section 8.1).

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{wSDL_1_2_draft_data_model.png}
\caption{WSDL 1.2 draft data model}
\end{figure}
8.3 SOAP 1.1 (Normative)

8.3.1 Introduction

In the case of a WSDL request-response-operation, consumer policies for the response message
MAY be conveyed in a SOAP header of the corresponding request message. The names
assigned to objects by the consumer are not guaranteed to match those assigned by the provider
to the equivalent objects. Therefore, the consumer MUST use the names assigned by the
provider to associate consumer policy with WSDL objects. This means that response policies
MUST be tailored to the particular provider, and the consumer may require a different policy for
each provider of the same service.

In the case of the WSDL solicit-response-operation and the notification-operation, the WSDL
technique, described above, SHALL be used to disseminate consumer policy.

8.3.2 Structure

Conformant <PolicySet> elements SHALL be structured as described in Section 8.1.3, above.
Only WSDL message definitions of the “output” or “fault” types SHALL be targeted by policies.

8.3.3 Integrity/authenticity protection

If the <soap/header> element is integrity-protected, then the <PolicySet> elements
SHOULD be included within the integrity-protection of that element.

Where it is not possible to do this, either because the <soap/header> element is not integrity-
protected, or for other reasons, <PolicySet> elements SHALL be enclosed in a
<saml/Assertion> element wrapper [SAML]. The <saml/Assertion> element SHALL be
integrity protected.

The policy-user SHALL ignore the PolicySet@PolicySetId attribute.

The policy-user SHALL verify that the <PolicySet/Target> element identifies the
wsdl/port@name attribute of the WSDL 1.1 port that originated the request.

In the case where a policy is wrapped in a <saml/Assertion>, the host and domain parts of
the authenticated name of the originating end-point SHALL be identical to the
saml/Assertion@Issuer attribute value. The saml/Assertion@Issuer attribute value
SHALL be identical to the CN attribute value in the subject field of the certificate [X509] that
validates the <saml/Assertion> element, whether integrity protection is provided by SSL or
XML Digital Signature.

If they are present, the policy-user SHALL confirm that the policy is current, by examining the
saml/Assertion/Conditions@NotBefore and
saml/Assertion/Conditions@NotOnOrAfter attributes.

8.3.4 Schema

An XACML <PolicySet> element SHALL be included in a SOAP header in accordance with the
following schema.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema

targetNamespace="urn:oasis:names:tc:xacml:wspl:draft:02"
xmlns:EndPointPolicy="urn:oasis:names:tc:xacml:wspl:draft:02"
xmlns:xsi=http://www.w3.org/2001/XMLSchema
```
9. References (Non-normative)


Appendix A. Worked example (Non-normative)

This appendix contains a worked example to illustrate the process of combining and reducing XACML policies that conform with this profile, using two simple policy instances. The example is drawn from the realm of data-rate allocation, and builds on the example given in Section 0.

Consumer policy

This section describes the service consumer's requirements for the data-rate allocation aspect of service invocation.

A.1.1. Plain-language policy

The plain language description of the policy is as follows.

The service-consumer's first choice is to pay a maximum of $100/minute for a minimum guaranteed data-rate of 64kb/s.

The second choice is to pay a maximum of $50/minute for a minimum guaranteed data-rate between 9pm and midnight of 32kb/s.

A.1.2. XACML policy

```xml
<PolicySet PolicySetId="Consumer Policy"
PolicyCombiningAlgId="deny-overrides">
  <Target>
    <Resources>
      <ResourceMatch MatchId="equal"
      <AttributeValue DataType="anyURI">
        serviceX:portX
      </AttributeValue>
      <ResourceAttributeDesignator AttributeId="portID"
      <AttributeValue DataType="anyURI"/>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Policy PolicyId="Consumer Data-Rate Allocation Objective"
  RuleCombiningAlgId="permit-overrides">
    <Target>
      <AnyAction/>
    </Target>
    <Actions>
      <ActionMatch MatchId="equal">
        <AttributeValue DataType="anyURI">
          data-rate-allocation
          </AttributeValue>
        </ActionMatch>
      <ActionMatch MatchId="equal">
        <AttributeValue DataType="anyURI">
          data-rate-allocation
          </AttributeValue>
      </ActionMatch>
    </Actions>
  </Policy>
</PolicySet>
```
<ActionAttributeDesignator AttributeId="objectiveID"
DataType="anyURI"/>
</ActionMatch>
</Actions>
</Target>

<Rule RuleId="Consumer Data-Rate Allocation Objective Strategy 1" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="less-than-or-equal">
      <SubjectAttributeDesignator DataType="integer"
AttributeId="fee"/>
      <AttributeValue DataType="integer">
        100
      </AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer"
AttributeId="data-rate"/>
      <AttributeValue DataType="integer">
        64000
      </AttributeValue>
    </Apply>
  </Condition>
</Rule>

<Rule RuleId="Consumer Data-rate Allocation Objective Strategy 2" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="less-than-or-equal">
      <SubjectAttributeDesignator DataType="integer"
AttributeId="fee"/>
      <AttributeValue DataType="integer">
        50
      </AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer"
AttributeId="data-rate"/>
      <AttributeValue DataType="integer">
        32000
      </AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <EnvironmentAttributeDesignator DataType="time"
AttributeId="timeOfDay"/>
      <AttributeValue DataType="time">
        21:00
      </AttributeValue>
    </Apply>
  </Condition>
</Rule>
Combining process

A.1.3. Combine <PolicySet> elements

The <Target> elements of the two <PolicySet> elements are identical. Therefore, they may be combined. Append the provider <Policy> elements and assign a new PolicySetId value.

```
<PolicySet PolicySetId="Consumer Policy"
PolicyCombiningAlgId="deny-overrides">
  <Target>
    <Resources>
      ResourceMatch MatchId="equal"
      <AttributeValue DataType="anyURI">
        serviceX:portX
      </AttributeValue>
      </ResourceMatch>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Policy PolicyId="Consumer Data-rate Allocation Objective"
  RuleCombiningAlgId="permit-overrides">
    <Target>
      <ActionMatch MatchId="equal">
        <AttributeValue DataType="anyURI">
          data-rate-allocation
        </AttributeValue>
        </ActionMatch>
      </Actions>
    </Target>
    <Rule RuleId="Consumer Data-rate Allocation Objective Strategy 1" Effect="Permit">
      <Condition FunctionId="and">
        <Apply FunctionId="less-than-or-equal">
          <SubjectAttributeDesignator DataType="integer"
          AttributeId="fee"/>
          <AttributeValue DataType="integer">1.00</AttributeValue>
        </Apply>
        <Apply FunctionId="greater-than-or-equal">
          <ResourceAttributeDesignator DataType="integer"
          AttributeId="data-rate"/>
          <AttributeValue DataType="integer">64000</AttributeValue>
        </Apply>
      </Condition>
    </Rule>
    <Rule RuleId="Consumer Data-rate Allocation Objective Strategy 2" Effect="Permit">
      <Condition FunctionId="and">
        <Apply FunctionId="less-than-or-equal">
          <ResourceAttributeDesignator DataType="integer"
          AttributeId="data-rate"/>
          <AttributeValue DataType="integer">64000</AttributeValue>
        </Apply>
        <Apply FunctionId="greater-than-or-equal">
          <SubjectAttributeDesignator DataType="integer"
          AttributeId="fee"/>
          <AttributeValue DataType="integer">1.00</AttributeValue>
        </Apply>
      </Condition>
    </Rule>
  </Policy>
</PolicySet>
```
<Condition FunctionId="and">
<Apply FunctionId="less-than-or-equal">
<SubjectAttributeDesignator DataType="integer"
AttributeId="fee"/>
<AttributeValue DataType="integer">0.50</AttributeValue>
</Apply>
<Apply FunctionId="greater-than-or-equal">
<ResourceAttributeDesignator DataType="integer"
AttributeId="data-rate"/>
<AttributeValue DataType="integer">32000</AttributeValue>
</Apply>
</Apply>
<Apply FunctionId="greater-than-or-equal">
<EnviromentAttributeDesignator DataType="time"
AttributeId="timeOfDay"/>
<AttributeValue DataType="time">9:00</AttributeValue>
</Apply>
</Condition>
</Rule>
</Policy>
<Policy PolicyId="Provider Data-rate Allocation Objective"
RuleCombiningAlgId="permit-overrides">
<Target>
<Actions>
<ActionMatch MatchId="equal">
<AttributeValue DataType="anyURI">
<AttributeValue>
<data-rate-allocation</AttributeValue>
</AttributeValue>
<ActionAttributeDesignator AttributeId="objectiveID"
DataType="anyURI"/>
</ActionMatch>
</Actions>
</Target>
</Policy>
<Rule RuleId="Provider Data-rate Allocation Objective Strategy 1"
Effect="Permit">
<Condition FunctionId="and">
<Apply FunctionId="equal">
<SubjectAttributeDesignator DataType="integer"
AttributeId="fee"/>
<AttributeValue DataType="integer">1.50</AttributeValue>
</Apply>
<Apply FunctionId="greater-than-or-equal">
<ResourceAttributeDesignator DataType="integer"
AttributeId="data-rate"/>
<AttributeValue DataType="integer">64000</AttributeValue>
</Apply>
</Condition>
</Rule>
<Rule RuleId="Provider Data-rate Allocation Objective Strategy 2"
Effect="Permit">
<Condition FunctionId="and">
<Apply FunctionId="equal">
<SubjectAttributeDesignator DataType="integer"
AttributeId="fee"/>
<AttributeValue DataType="integer">0.45</AttributeValue>
</Apply>
</Condition>
</Rule>
A.1.4. Combine <Policy> elements

The <Target> elements of the two <Policy> elements are identical. Therefore, they may be combined. Append the provider <Rule> elements and assign a new PolicyId value.
<SubjectAttributeDesignator DataType="integer"
  AttributeId="fee"/>
<AttributeValue DataType="integer">1.00</AttributeValue>
</Apply>

<Apply FunctionId="greater-than-or-equal">
  <ResourceAttributeDesignator DataType="integer"
  AttributeId="data-rate"/>
  <AttributeValue DataType="integer">64000</AttributeValue>
</Apply>
</Condition>
</Rule>

<Rule RuleId="Consumer Data-rate Allocation Objective Strategy 2" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="less-than-or-equal">
      <SubjectAttributeDesignator DataType="integer"
        AttributeId="fee"/>
      <AttributeValue DataType="integer">0.50</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer"
        AttributeId="data-rate"/>
      <AttributeValue DataType="integer">32000</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <EnvironmentAttributeDesignator DataType="time"
        AttributeId="timeOfDay"/>
      <AttributeValue DataType="time">9:00</AttributeValue>
    </Apply>
  </Condition>
</Rule>

<Rule RuleId="Provider Data-rate Allocation Objective Strategy 1" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="equal">
      <SubjectAttributeDesignator DataType="integer"
        AttributeId="fee"/>
      <AttributeValue DataType="integer">1.50</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer"
        AttributeId="data-rate"/>
      <AttributeValue DataType="integer">64000</AttributeValue>
    </Apply>
  </Condition>
</Rule>

<Rule RuleId="Provider Data-rate Allocation Objective Strategy 2" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="equal">
      <SubjectAttributeDesignator DataType="integer"
        AttributeId="fee"/>
      <AttributeValue DataType="integer">0.45</AttributeValue>
    </Apply>
  </Condition>
</Rule>
A.1.5. Combine <Rule> elements

Coincident pairs of <Rule> elements are identified and combined. Coincident pairs of <Rule> elements are combined by appending the provider’s <Apply> elements. Pairs are combined only where each member of the pair originally came from different <PolicySet> elements.

```xml
<PolicySet PolicySetId="Combined Policies"
  PolicyCombiningAlgId="deny-overrides">
  <Resources>
    <ResourceMatch MatchId="equal"
      AttributeValue DataType="anyURI">
      serviceX:portX
    </AttributeValue>
    <ResourceAttributeDesignator AttributeId="portID"
      DataType="anyURI"/>
  </ResourceMatch>
  <Actions>
    <AnyAction/>
  </Actions>
  </Target>
  <Policy PolicyId="Combined Data-rate Allocation Objective"
    RuleCombiningAlgId="permit-overrides">
    <Target>
      <Actions>
        <ActionMatch MatchId="equal">
          <AttributeValue DataType="anyURI">
            data-rate-allocation
          </AttributeValue>
          <ActionAttributeDesignator AttributeId="objectiveID"
            DataType="anyURI"/>
        </ActionMatch>
      </Actions>
    </Target>
    <Rule RuleId="Consumer Strategy 1/Provider Strategy 1"
      Effect="Permit">
      <Condition FunctionId="and">
        <Apply FunctionId="equal">
          <ResourceAttributeDesignator DataType="integer"
            AttributeId="data-rate"/>
          <AttributeValue DataType="integer">
            40000
          </AttributeValue>
        </Apply>
        <Apply FunctionId="greater-than-or-equal">
          <EnvironmentAttributeDesignator DataType="time"
            AttributeId="timeOfDay"/>
          <AttributeValue DataType="time">6:00</AttributeValue>
        </Apply>
      </Condition>
    </Rule>
  </Policy>
</PolicySet>
```
<Apply FunctionId="less-than-or-equal">
  <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
  <AttributeValue DataType="integer">1.00</AttributeValue>
</Apply>

<Apply FunctionId="greater-than-or-equal">
  <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
  <AttributeValue DataType="integer">64000</AttributeValue>
</Apply>

<Apply FunctionId="equal">
  <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
  <AttributeValue DataType="integer">1.50</AttributeValue>
</Apply>

<Apply FunctionId="greater-than-or-equal">
  <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
  <AttributeValue DataType="integer">64000</AttributeValue>
</Apply>

<Apply FunctionId="equal">
  <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
  <AttributeValue DataType="integer">40000</AttributeValue>
</Apply>

<Apply FunctionId="greater-than-or-equal">
  <EnvironmentAttributeDesignator DataType="time" AttributeId="timeOfDay"/>
  <AttributeValue DataType="time">6:00</AttributeValue>
</Apply>

<Rule RuleId="Consumer Strategy 1/Provider Strategy 2" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="less-than-or-equal">
      <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
      <AttributeValue DataType="integer">1.00</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
      <AttributeValue DataType="integer">64000</AttributeValue>
    </Apply>
    <Apply FunctionId="equal">
      <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
      <AttributeValue DataType="integer">0.45</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
      <AttributeValue DataType="integer">40000</AttributeValue>
    </Apply>
    <Apply FunctionId="equal">
      <EnvironmentAttributeDesignator DataType="time" AttributeId="timeOfDay"/>
      <AttributeValue DataType="time">6:00</AttributeValue>
    </Apply>
  </Condition>
</Rule>

<Rule RuleId="Consumer Strategy 2/Provider Strategy 1" Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="less-than-or-equal">
      <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
      <AttributeValue DataType="integer">1.00</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
      <AttributeValue DataType="integer">64000</AttributeValue>
    </Apply>
    <Apply FunctionId="equal">
      <SubjectAttributeDesignator DataType="integer" AttributeId="fee"/>
      <AttributeValue DataType="integer">0.45</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer" AttributeId="data-rate"/>
      <AttributeValue DataType="integer">40000</AttributeValue>
    </Apply>
    <Apply FunctionId="equal">
      <EnvironmentAttributeDesignator DataType="time" AttributeId="timeOfDay"/>
      <AttributeValue DataType="time">6:00</AttributeValue>
    </Apply>
  </Condition>
</Rule>
<Condition FunctionId="and">
  <Apply FunctionId="less-than-or-equal">
    <SubjectAttributeDesignator DataType="integer"
      AttributeId="fee"/>
    <AttributeValue DataType="integer">0.50</AttributeValue>
  </Apply>
  <Apply FunctionId="greater-than-or-equal">
    <ResourceAttributeDesignator DataType="integer"
      AttributeId="data-rate"/>
    <AttributeValue DataType="integer">32000</AttributeValue>
  </Apply>
  <Apply FunctionId="greater-than-or-equal">
    <EnvironmentAttributeDesignator DataType="time"
      AttributeId="timeOfDay"/>
    <AttributeValue DataType="time">9:00</AttributeValue>
  </Apply>
  <Apply FunctionId="equal">
    <SubjectAttributeDesignator DataType="integer"
      AttributeId="fee"/>
    <AttributeValue DataType="integer">1.50</AttributeValue>
  </Apply>
  <Apply FunctionId="greater-than-or-equal">
    <ResourceAttributeDesignator DataType="integer"
      AttributeId="data-rate"/>
    <AttributeValue DataType="integer">64000</AttributeValue>
  </Apply>
</Condition>

<Rule RuleId="Consumer Strategy 2/Provider Strategy 2"
  Effect="Permit">
  <Condition FunctionId="and">
    <Apply FunctionId="less-than-or-equal">
      <SubjectAttributeDesignator DataType="integer"
        AttributeId="fee"/>
      <AttributeValue DataType="integer">0.50</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <ResourceAttributeDesignator DataType="integer"
        AttributeId="data-rate"/>
      <AttributeValue DataType="integer">32000</AttributeValue>
    </Apply>
    <Apply FunctionId="greater-than-or-equal">
      <EnvironmentAttributeDesignator DataType="time"
        AttributeId="timeOfDay"/>
      <AttributeValue DataType="time">9:00</AttributeValue>
    </Apply>
    <Apply FunctionId="equal">
      <SubjectAttributeDesignator DataType="integer"
        AttributeId="fee"/>
      <AttributeValue DataType="integer">0.45</AttributeValue>
    </Apply>
    <Apply FunctionId="equal">
      <ResourceAttributeDesignator DataType="integer"
        AttributeId="data-rate"/>
      <AttributeValue DataType="integer">40000</AttributeValue>
    </Apply>
  </Condition>
</Rule>
A.1.6. Combine <Apply> elements

<Apply> elements are combined if they are combinable according to Table 1.
<AttributeValue DataType="integer">0.45</AttributeValue>
</Apply>
</Apply>
</Apply FunctionId="equal">
<ResourceAttributeDesignator DataType="integer"
AttributeId="data-rate"/>
<AttributeValue DataType="integer">
40000
</AttributeValue>
</Apply>
</Apply FunctionId="greater-than-or-equal">
<EnvironmentAttributeDesignator DataType="time"
AttributeId="timeOfDay"/>
<AttributeValue DataType="time">9:00</AttributeValue>
</Apply>
</Condition>
</Rule>
</Policy>
</PolicySet>

A.1.7. Substitute <Apply> elements

Elimination is achieved by deleting all except the first <Rule> element.

PolicySet PolicySetId="Combined Policies"
PolicyCombiningAlgId="deny-overrides">
<Target>
<Resources>
<ResourceMatch MatchId="equal">
<AttributeValue DataType="anyURI">
serviceX:portX
</AttributeValue>
</ResourceMatch>
</Resources>
<Actions>
</Actions>
</Target>
Policy PolicyId="Combined Data-rate Allocation Objective"
RuleCombiningAlgId="permit-overrides">
<Target>
<Actions>
<ActionMatch MatchId="equal">
<AttributeValue DataType="anyURI">
data-rate-allocation
</AttributeValue>
</ActionMatch>
</Actions>
</Target>
<Rule RuleId="Consumer Strategy 2/Provider Strategy 2"
Effect="Permit">
</Rule>
</PolicySet>
<SubjectAttributeDesignator DataType="integer"
AttributeId="fee"/>

<AttributeValue DataType="integer">0.45</AttributeValue>

<Apply FunctionId="equal">
    <ResourceAttributeDesignator DataType="integer"
AttributeId="data-rate"/>
    <AttributeValue DataType="integer">40000</AttributeValue>
</Apply>

<Apply FunctionId="equal">
    <EnvironmentAttributeDesignator DataType="time"
AttributeId="timeOfDay"/>
    <AttributeValue DataType="time">9:00</AttributeValue>
</Apply>

</Condition>

</Rule>

</Policy>

</PolicySet>
## Appendix B. Revision history

<table>
<thead>
<tr>
<th>Rev</th>
<th>Date</th>
<th>By whom</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft 02</td>
<td>23 July 2003</td>
<td>Tim Moses</td>
<td>Limited functions and data-types to those defined by XACML. Prohibited the nesting of <code>&lt;Apply&gt;</code> elements. In the WSDL binding, targeted top-level policy statements at <code>&lt;wsdl:port&gt;</code> elements. Introduced two levels of <code>&lt;PolicySet&gt;</code> elements to allow finer targeting of policy statements. Added a “Security Considerations” section. Introduced the elimination step.</td>
</tr>
<tr>
<td>Draft 03</td>
<td>5 Sep 2003</td>
<td>Tim Moses</td>
<td>Added text clarifying <code>attribute</code> classification. Modified approach to combining involving greater-than and less-than operations to eliminate floor and ceiling functions. Clarified the procedure when compatible, but non-combinable, <code>predicates</code> are present. Added text in WSDL 1.2 binding section.</td>
</tr>
</tbody>
</table>
Appendix C. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights. OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

Copyright (C) OASIS Open 2003. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.