SAML 2.0 Profile of XACML, Version 2.0

Committee Draft 1

16 April 2009

Specification URIs:

This Version:

http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-1-en.html

http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-1-en.odt (Authoritative)

http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-1-en.pdf

Previous Version:

N/A

Latest Version:

http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.html

http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.odt (Authoritative)

http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.pdf

Technical Committee:

OASIS eXtensible Access Control Markup Language (XACML) TC

Chair(s):

Hal Lockhart <hal.lockhart@oracle.com>

Bill Parducci <bill@parducci.net>

Editors:

Erik Rissanen <erik@axiomatics.com>
Hal Lockhart <hal.lockhart@oracle.com>

Related Work:

This specification replaces and supersedes:

- SAML 2.0 profile of XACML 2.0

This specification is related to:

- SAML 2.0 OASIS Standard
- XACML 1.0, 2.0, 3.0 OASIS Standards
- XACML 1.1 Committee Draft

Declared XML Namespace(s):

urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:assertion:cd-01
urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:protocol:cd-01
urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:assertion:cd-01
urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:protocol:cd-01
urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:cd-01
urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:assertion:cd-01
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:protocol:cd-01

Abstract:

This specification defines a profile for the integration of the OASIS Security Assertion Markup Language (SAML) Version 2.0 with all versions of XACML. SAML 2.0 complements XACML functionality in many ways, so a number of somewhat independent functions are described in this profile: 1) use of SAML 2.0 Attribute Assertions with XACML, including the use of SAML Attribute Assertions in a SOAP Header to convey Attributes that can be consumed by an XACML PDP, 2) use of SAML to carry XACML authorization decisions, authorization decision queries, and authorization decision responses, 3)use of SAML to carry XACML policies, policy queries, and policy query responses, 4) use of XACML authorization decisions or policies as Advice in SAML Assertions, and 5) use of XACML responses in SAML Assertions as authorization tokens. Particular implementations may provide only a subset of these functions.

Status:

This document was last revised or approved by the OASIS eXtensible Access Control Markup Language (XACML) TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at http://www.oasis-open.org/committees/xacml/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page http://www.oasis-open.org/committees/xacmlhttp://www.oasis-open.org/committees/xacml/ipr.php.

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/xacml/.

Notices

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

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' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on 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 implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", “SAML” and “XACML” are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

1 Introduction

[Except for schema fragments, all text is normative unless otherwise indicated.]

Non-normative through Section 1.3

The OASIS eXtensible Access Control Markup Language [XACML] is a powerful, standard language that specifies schemas for authorization policies and for authorization decision requests and responses. It also specifies how to evaluate policies against requests to compute a response. A brief non-normative overview of XACML is available in [XACMLIntro].

The non-normative XACML usage model assumes that a Policy Enforcement Point (PEP) is responsible for protecting access to one or more resources. When a resource access is attempted, the PEP sends a description of the attempted access to a Policy Decision Point (PDP) in the form of an authorization decision request. The PDP evaluates this request against its available policies and attributes and produces an authorization decision that is returned to the PEP. The PEP is responsible for enforcing the decision.

In producing its description of the access request, the PEP may obtain attributes from on-line Attribute Authorities (AA) or from Attribute Repositories into which AAs have stored attributes. The PDP (or, more precisely, its Context Handler component) may augment the PEP's description of the access request with additional attributes obtained from AAs or Attribute Repositories.

The PDP may obtain policies from on-line Policy Administration Points (PAP) or from Policy Repositories into which PAPs have stored policies.

XACML itself defines the content of some of the messages necessary to implement this model, but deliberately confines its scope to the language elements used directly by the PDP and does not define protocols or transport mechanisms. Full implementation of the usage model depends on use of other standards to specify assertions, protocols, and transport mechanisms. XACML also does not specify how to implement a Policy Enforcement Point, Policy Administration Point, Attribute Authority, Context Handler, or Repository, but XACML artifacts can serve as a standard format for exchanging information between these entities when combined with other standards.

One standard suitable for providing the assertion and protocol mechanisms needed by XACML is the OASIS Security Assertion Markup Language (SAML), Version 2.0 [SAML]. SAML defines schemas intended for use in requesting and responding with various types of security assertions. The SAML schemas include information needed to identify, validate, and authenticate the contents of the assertions, such as the identity of the assertion issuer, the validity period of the assertion, and the digital signature of the assertion. The SAML specification describes how these elements are to be used. In addition, SAML has associated specifications that define bindings to other standards. These other standards provide transport mechanisms and specify how digital signatures should be created and verified.

1.1 Organization of this Profile

This Profile defines how to use SAML 2.0 to protect, store, transport, request, and respond with XACML schema instances and other information needed by an XACML implementation. The remaining Sections of this Profile describe the following aspects of SAML 2.0 usage.

Section 2 describes how to use SAML Attributes in an XACML system. It describes the use of the following elements:

1. <saml:`Attribute>` – A standard SAML element that MAY be used in an XACML system for storing and transmitting attribute values. The <saml:`Attribute>` must be at least conceptually transformed into an `<xacml-context:Attribute>` before it can be used in an XACML Request Context.

2. `<saml:AttributeStatement>` – A standard SAML element that MUST be used to hold `<saml:Attribute>` instances in an XACML system.

3. `<saml:Assertion>` – A standard SAML element that MUST be used to hold `<saml:AttributeStatement>` instances in an XACML system, either in an Attribute Repository or in a SAML Attribute Response. The `<saml:Assertion>` contains information that is required in order to transform a `<saml:Attribute>` into an `<xacml-context:Attribute>`. An instance of such a `<saml:Assertion>` element is called a SAML Attribute Assertion in this Profile.

4. `<samlp:AttributeQuery>` – A standard SAML protocol element that MAY be used by an XACML PDP or PEP to request `<saml:Attribute>` instances from an Attribute Authority for use in an XACML Request Context.

5. `<samlp:Response>` – A standard SAML protocol element that MUST be used to return SAML Attribute Assertions in response to a `<samlp:AttributeQuery>` in an XACML system. An instance of such a `<samlp:Response>` element is called a SAML Attribute Response in this Profile.

Section 3 describes ways to convey XACML Attributes in a SOAP message.

Section 4 describes the use of SAML in requesting, responding with, storing, and transmitting authorization decisions in an XACML system. The following types and elements are described:

1. `xacml-saml:XACMLAuthzDecisionStatementType` – A new SAML extension type defined in this Profile that MAY be used in an XACML system to create XACMLAuthzDecision Statements that hold XACML authorization decisions for storage or transmission.

2. `<saml:Statement>` – A standard SAML element that MUST be used to contain instances of the `<xacml-saml:XACMLAuthzDecisionStatementType>` . An instance of such a `<saml:Statement>` element is called an XACMLAuthzDecision Statement in this Profile.

3. `<saml:Assertion>` – A standard SAML element that MUST be used to hold XACMLAuthzDecision Statements in an XACML system, either in a repository or in a XACMLAuthzDecision Response. An instance of such a `<saml:Assertion>` element is called an XACMLAuthzDecision Assertion in this Profile.

4. `<xacml-samlp:XACMLAuthzDecisionQuery>` – A new SAML extension protocol element defined in this Profile that MAY be used by a PEP to request an authorization decision from an XACML PDP.

5. `<samlp:Response>` – A standard SAML protocol element that MUST be used to return XACMLAuthzDecision Assertions from an XACML PDP in response to an `<xacml-samlp:XACMLAuthzDecisionQuery>`. An instance of such a `<samlp:Response>` element is called an XACMLAuthzDecision Response in this Profile.

Section 6 describes the use of SAML in requesting, responding with, storing, and transmitting XACML policies. The following types and elements are described:

1. `xacml-saml:XACMLPolicyStatementType` – A new SAML extension type defined in this Profile that MAY be used in an XACML system to create XACMLPolicy Statements that hold XACML policies for storage or transmission.

2. `<saml:Statement>` – A standard SAML element that MUST be used to contain instances of the `xacml-saml:XACMLPolicyStatementType`. An instance of such a `<saml:Statement>` element is called an XACMLPolicy Statement in this Profile.

3. <saml:Assertion> – A standard SAML element that MUST be used to hold XACMLPolicy Statement instances in an XACML system, either in a repository or in an XACMLPolicy Response. An instance of such a <saml:Assertion> element is called an XACMLPolicy Assertion in this Profile.

4. <xacml-samlp:XACMLPolicyQuery> – A new SAML extension protocol element defined in this Profile that MAY be used by a PDP or other application to request XACML policies from a Policy Administration Point (PAP).

5. <samlp:Response> – A standard SAML protocol element that MUST be used to return XACMLPolicy Assertions in response to an <xacml-samlp:XACMLPolicyQuery>. An instance of such a <samlp:Response> element is called an XACMLPolicy Response in this Profile.

Section 7 describes the use of XACMLAuthzDecision Assertion and XACMLPolicy Assertion instances as advice in other SAML Assertions. The following element is described:

1. <saml:Advice> – A standard SAML element that MAY be used to convey XACMLPolicy Assertions or XACMLAuthzDecision Assertions as advice in other <saml:Assertion> instances.

Section 8 describes the use of XACMLAuthzDecision Assertions as authorization tokens in a SOAP message exchange.

Section Error: Reference source not found describes recommended non-normative SAML metadata for use with these XACML-related protocols.

Section 9 describes requirements for conformance with various aspects of this Profile.

1.1 Diagram of SAML integration with XACML

Figure 1 illustrates the XACML use model and the messages that can be used to communicate between the various components. Not all components or messages will be used in every implementation. Not shown, but described in this Profile, is the ability to use an XACMLPolicy Assertion or an XACMLAuthzDecision Assertion in a <saml:Advice> instance.

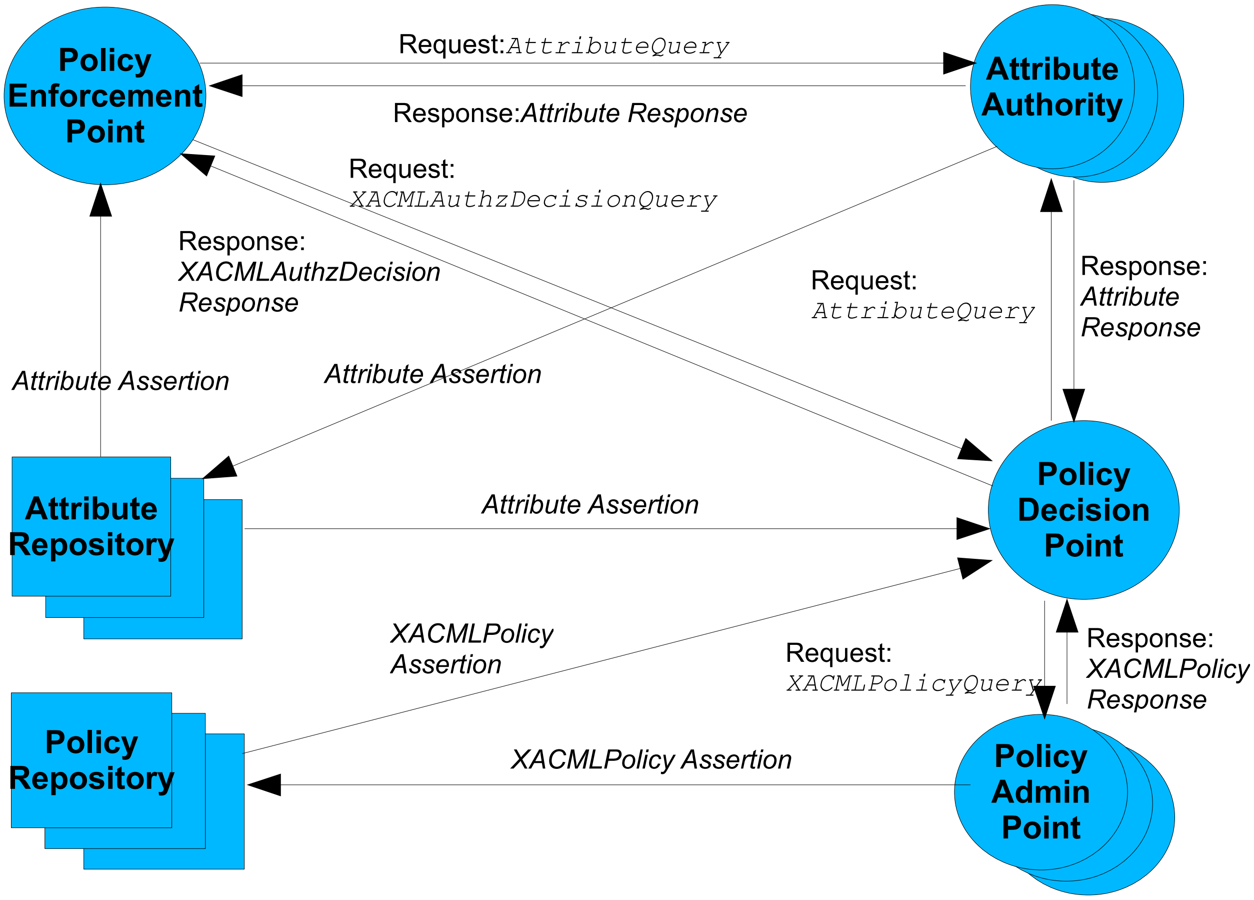

Figure 1: Components and messages in a integration of SAML with XACML

This Profile describes all these message elements, and describes how to use them, along with other aspects of using SAML with XACML.

1.2 Backwards compatibility

This Profile requires no changes or extensions to XACML, but does define extensions to SAML. The Profile may be used with XACML 1.0 , 1.1, 2.0, or 3.0. Separate versions of the Profile schemas are used with each version of XACML as described in Section 1.1.

1.3 Terminology

The keywords "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 [RFC 2119]

AA – Attribute Authority. An entity that binds attributes to identities. Such a binding may be expressed using a SAML Attribute Assertion with the Attribute Authority as the issuer.

Attribute - In this Profile, the term “Attribute”, when the initial letter is capitalized, may refer to either an XACML Attribute or to a SAML Attribute. The term will always be preceded with the type of Attribute intended.

- An XACML Attribute is a typed name/value pair, with other optional information, specified using an `<xacml-context:Attribute>` instance. An XACML Attribute is associated with an entity or topic identity by the XACML Attribute's position within a particular Attribute group in the XACML Request.
- A SAML Attribute is a name/value pair, with other optional information, specified using a `<saml:Attribute>` instance. A SAML Attribute is associated with a particular subject by its inclusion in a SAML Attribute Assertion that contains a `<saml:Subject>` instance. The SAML `Subject` may correspond to any XACML Attribute group.

Attribute group – In this Profile, the term “Attribute group” is used to describe a collection of XACML Attributes in an XACML Request Context that are associated with a particular entity. In XACML 1.0, 1.1, and 2.0, there is a fixed number of such collections, called Subject Attributes, Resource Attributes, Action Attributes, and Environment Attributes. In XACML 3.0, the number and identifiers of such collections is extensible, but there are standard identifiers that correspond to the fixed collections defined in previous versions of XACML.

attribute – In this Profile, the term “attribute”, when not capitalized, refers to a generic attribute or characteristic unless it is preceded by the term “XML”. An “XML attribute” is a syntactic component in XML that occurs inside the opening tag of an XML element.

Attribute Assertion – A `<saml:Assertion>` instance that contains a `<saml:AttributeStatement>` instance.

Attribute Response – A `<samlp:Response>` instance that contains a SAML Attribute Assertion.

PAP – Policy Administration Point. An abstract entity that issues authorization policies that are used by a Policy Decision Point (PDP).

PDP - Policy Decision Point. An abstract entity that evaluates an authorization decision request against one or more policies to produce an authorization decision.

PEP – Policy Enforcement Point. An abstract entity that enforces access control for one or more resources. When a resource access is attempted, a PEP sends an access request describing the attempted access to a PDP. The PDP returns an access decision that the PEP then enforces.

policy – A set of rules indicating the conditions under which an access is permitted or denied. XACML has two different schema elements used for policies: `<xacml:Policy>` and `<xacml:PolicySet>`. An `<xacml:PolicySet>` is a collection of other `<xacml:Policy>` and `<xacml:PolicySet>` elements. An `<xacml:Policy>` contains actual access control rules.

XACMLAuthzDecision Assertion – A `<saml:Assertion>` instance that contains an XACMLAuthzDecision Statement.

XACMLAuthzDecision Response – A **<**`samlp:Response>` instance that contains an XACMLAuthzDecision Assertion.

XACMLAuthzDecision Statement – A `<saml:Statement>` instance that is of type `xacml-saml:XACMLAuthzDecisionStatementType`.

XACMLPolicy Assertion – A `<saml:Assertion>` instance that contains an XACMLPolicy Statement.

XACMLPolicy Response – A `<samlp:Response>` instance that contains an XACMLPolicy Assertion.

XACMLPolicy Statement – A `<saml:Statement>` instance that is of type `xacml-saml:XACMLPolicyStatementType`.

1.1 Namespaces

Normative

The following namespace prefixes are used in the schema fragments:

Prefix	Namespace
`xacml`	The XACML policy namespace.
`xacml-context`	The XACML context namespace.
`xacml-saml`	XACML extensions to the SAML 2.0 Assertion schema namespace.
`xacml-samlp`	XACML extensions to the SAML 2.0 Protocol schema namespace.
`xacml-samlm`	urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:metadata
`saml`	urn:oasis:names:tc:SAML:2.0:assertion
`samlp`	urn:oasis:names:tc:SAML:2.0:protocol
`md`	urn:oasis:names:tc:SAML:2.0:metadata
`ds`	http://www.w3.org/2000/09/xmldsig#
`xsi`	`http://www.w3.org/2001/XMLSchema-instance`
`wsse`	`http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd` or `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.1.xsd`
`wst`	http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3.xsd

This Profile is written for use with XACML 1.0 [XACML1], 1.1 [XACML1.1], 2.0 [XACML2], or 3.0 [XACML3]. Depending on the version of XACML being used, the `xacml`, `xacml-context`, `xacml-saml`, and `xacml-samlp` namespace prefixes have the following values in the schemas:

XACML 1.0:

```
   xacml="urn:oasis:names:tc:xacml:1.0:policy”
   xacml-context="urn:oasis:names:tc:xacml:1.0:context”
   xacml-saml=
"urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:assertion:wd-08”
   xacml-samlp=
"urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:protocol:wd-08”
```

XACML 1.1:

```
   xacml="urn:oasis:names:tc:xacml:1.0:policy”
   xacml-context="urn:oasis:names:tc:xacml:1.0:context”
   xacml-
saml="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:assertion:wd-08”
   xacml-
samlp="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:protocol:wd-08”
```

XACML 2.0:

```
   xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os”
   xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os”
   xacml-
saml="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:wd-08”
   xacml-
samlp="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:wd-08”
```

XACML 3.0:

```
   xacml="urn:oasis:names:tc:xacml:3.0:schema:os”
   xacml-context="urn:oasis:names:tc:xacml:3.0:schema:os”
      NOTE: XACML 3.0 uses a single schema for both policies and context.
   xacml-
saml="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:assertion:wd-08”
   xacml-
samlp="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:protocol:wd-08”
```

1.2 Normative References

[ADMIN] E. Rissanen, ed., *XACML v3.0 Administrative Policy Version 1.0*

[RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[SAML] S. Cantor, et al., eds., *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0,* http://www.oasis-open.org/committees/documents.php?wg_abbrev=security.

[SAML-PROFILE] J. Hughes, et al., eds., *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*, http://www.oasis-open.org/committees/documents.php?wg_abbrev=security**.**

[XACML1] *OASIS eXtensible Access Control Markup Language (XACML) Version 1.0,* http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf

[XACML1.1] *OASIS eXtensible Access Control Markup Language (XACML) Version 1.1,* http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf

[XACML2] T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML) Version 2.0,* OASIS Standard, 1 February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

[XACML3] E. Rissanen, ed., *OASIS eXtensible Access Control Markup Language (XACML) Version 3.0 working draft 11, 5 April 2009, FIXME URL*

[XACML-SAML] OASIS, the schemas associated with namespace `<xacml-saml>` that are a normative part of this Profile.

[XACML-SAMLP] OASIS, the schemas associated with namespace `<xacml-samlp>` that are a normative part of this Profile.

[WSFED] OASIS, Web Services Federation Language (WS-Federation) Version 1.2 Committee Draft 02 January 7, 2009 http://docs.oasis-open.org/wsfed/federation/v1.2/cd/ws-federation-1.2-spec-cd-02.doc

[WSS] OASIS, *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004),* OASIS Standard December 2004, and *WS-Security Core Specification 1.1,* OASIS Standard February 2006, http://www.oasis-open.org/specs/index.php.

WSTRUST] OASIS, WS-Trust 1.4 **FIXME**

1.3 Non-normative References

[XACMLIntro] S. Proctor, *A Brief Introduction to XACML*, http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html, 14 March 2003.

2 Attributes

In an XACML system, PEPs and PDP Context Handlers often need to retrieve attributes from on-line Attribute Authorities or from Attribute Repositories. SAML provides assertion and protocol elements that MAY be used for retrieval of attributes for use in an XACML Request Context. These elements include a `<saml:Attribute>` element for expressing a named attribute value, a `<saml:AttributeStatement>` for holding a collection of `<saml:Attribute>` elements, and a `<saml:Assertion>` element that can hold various kinds of statements, including a `<saml:AttributeStatement>`. A `<saml:Assertion>` instance containing a `<saml:AttributeStatement>` is called a SAML Attribute Assertion in this Profile. A SAML Attribute Assertion includes the name of the attribute issuer, an optional digital signature for authenticating the attribute, an optional subject identity to which the attribute is bound, and optional conditions for use of the assertion that may include a validity period during which the attribute is to be considered valid. Such an assertion is suitable for storing attributes in an Attribute Repository, for transmitting attributes between an Attribute Authority and an Attribute Repository, and for transmitting attributes between an Attribute Repository and a PEP or XACML Context Handler. For querying an on-line Attribute Authority for attributes, and for holding the response to that query, SAML defines `<samlp:AttributeQuery>` and `<samlp:Response>` elements. In this Profile, an instance of such a `<samlp:Response>` element is called a SAML Attribute Response. This Section describes the use of these SAML elements in an XACML system.

Since the format of a `<saml:Attribute>` differs from that of an `<xacml-context:Attribute>`, a mapping operation is required. This Section describes how to transform information contained in a SAML Attribute Assertion into one or more `<xacml-context:Attribute>` instances.

2.1 Element `<saml:Attribute>`

The standard `<saml:Attribute>` element MAY be used in an XACML system for storing and transmitting attribute values.

In order to be used in an XACML Request Context, each `<saml:Attribute>` instance MUST comply with the *SAML XACML Attribute Profile*, associated with namespace `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`, in Section 8.5 of the *Profiles for the OASIS Security Assertion Markup Language (SAML 2.0)* [SAML-PROFILE].

2.1.1 Mapping a `<saml:Attribute>` to an `<xacml-context:Attribute>`

An `<xacml-context:Attribute>` instance MUST be constructed from the corresponding `<saml:Attribute>` instance contained in a SAML Attribute Assertion as follows. An XACML implementation is NOT REQUIRED to instantiate the `<xacml-context:Attribute>` instances physically so long as the XACML PDP can obtain values for the XACML Attributes as if they had been instantiated in this way.

- XACML `AttributeId` XML attribute

  The fully-qualified value of the `<saml:Attribute>` `Name` XML attribute MUST be used.

- XACML `DataType` XML attribute

  The fully-qualified value of the `<saml:Attribute>` `DataType` XML attribute MUST be used. If the `<saml:Attribute>` `DataType` XML attribute is missing, the XACML `DataType` XML attribute MUST be `http://www.w3.org/2001/XMLSchema#string`.

- XACML `Issuer` XML attribute

The string value of the <saml:Issuer> instance from the SAML Attribute Assertion MUST be used.

- <xacml-context:AttributeValue>

  The <saml:AttributeValue> value MUST be used as the value of the <xacml-context:AttributeValue> instance.

Each <saml:Attribute> instance MUST be mapped to no more than one <xacml-context:Attribute> instance. Not all <saml:Attribute> instances in a SAML Attribute Assertion need to be mapped; a subset of <saml:Attribute> instances MAY be selected by a mechanism not specified in this Profile. The Issuer of the SAML Attribute Assertion MUST be used as the Issuer for each <xacml-context:Attribute> instance that is created from <saml:Attribute> instances in that SAML Attribute Assertion.

The <xacml-context:Attribute> created from the SAML Attribute Assertion MUST be placed into the Attribute group of the XACML Request Context that corresponds to the entity that is represented by the <saml:Subject> in the SAML Attribute Assertion.

> *Non-normative Example:* For example, if the SAML Attribute Assertion <saml:Subject> contains a <saml:NameIdentifier> instance, and the value of that NameIdentifier matches the value of the <xacml-context:Attribute> having an AttributeId of urn:oasis:names:tc:xacml:1.0:resource:resource-id, then <xacml-context:Attribute> instances created from <saml:Attribute> instances in that SAML Attribute Assertion MUST be placed into the <xacml-context:Resource> Attribute group or its corresponding XACML 3.0 Attribute group.

If a mapped <saml:Attribute> is placed into an <xacml-context:Subject> instance, then the XACML SubjectCategory XML attribute MUST also be consistent with the conceptual “subject category” of the entity that corresponds to the <saml:Subject> of the SAML Attribute Assertion that contained the <saml:Attribute>. The <saml:Subject> itself is NOT translated into an <xacml-context:Attribute> as part of processing a SAML Attribute Assertion; the <saml:Subject> identity is used only to determine the Attribute group in the XACML Request Context to which the <saml:Attribute> values should be added.

The mapping MUST be done in such a way that the semantics defined by SAML for the elements in a SAML Attribute Assertion have been adhered to. The mapping entity need not perform these semantic checks itself, but the system in which it operates MUST be such that the checks have been done before any <xacml:Attribute> created from a SAML Attribute Assertion is used by an XACML PDP. These semantic checks include, but are not limited to the following.

- Any NotBefore and NotOnOrAfter XML attributes in the SAML Attribute Assertion MUST be valid with respect to the <xacml:Request> in which the SAML-derived <xacml:Attribute> is used. This means that the XACML Attributes associated with the following AttributeId values in the <xacml:Request> MUST represent times and dates that are not before the NotBefore XML attribute value and not on or after the NotOnOrAfter XML attribute value:
  urn:oasis:names:tc:xacml:1.0:environment:current-time
  urn:oasis:names:tc:xacml:1.0:environment:current-date
  urn:oasis:names:tc:xacml:1.0:environment:current-dateTime

  The time period during which SAML Attribute Assertions are considered valid in XACML 3.0 depends on whether the PDP is configured to retrieve XACML Attributes that were valid at the time a policy was issued or at the time the policy is being evaluated.

- The semantics defined by SAML for any <saml:AudienceRestrictionCondition> or <saml:DoNotCacheCondition> elements MUST be adhered to.

2.1 Element <saml:AttributeStatement>

When a <saml:Attribute> instance is stored or transmitted in an XACML system, the instance MUST be enclosed in a standard SAML <saml:AttributeStatement>. The definition and use of the <saml:AttributeStatement> element MUST be as described in the SAML 2.0 standard [SAML].

2.2 Element <saml:Assertion>: SAML Attribute Assertion

When a <saml:AttributeStatement> instance is stored or transmitted in an XACML system, the instance MUST be enclosed in a <saml:Assertion>. An instance of such a <saml:Assertion> element is called a SAML Attribute Assertion in this Profile.

When used as a SAML Attribute Assertion in an XACML system, the definition and use of the <saml:Assertion> element MUST be as specified in the SAML 2.0 standard, augmented with the following requirements. Except as specified here, this Profile imposes no requirements or restrictions on the SAML Attribute Assertion element and its contents beyond those specified in SAML 2.0.

<saml:Issuer> [Required]

The <saml:Issuer> element is a required element for holding information about “the SAML authority that is making the claim(s) in the assertion” [SAML].

In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> element refer to the entity that signs the SAML Attribute Assertion.. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the SAML Attribute Assertion.

When a SAML Attribute Assertion containing a <saml:Attribute> is used to construct an <xacml-context:Attribute>, the string value of the <saml:Issuer> instance MUST be used as the value of the <xacml-context:Attribute> Issuer XML attribute, so the <saml:Issuer> value SHOULD be specified with this in mind.

<ds:Signature> [Optional]

The <ds:Signature> element is an optional element for holding “An XML Signature that authenticates the assertion, as described in Section 5 of the SAML 2.0 specification [SAML].”

A <ds:Signature> instance MAY be used in a SAML Attribute Assertion. In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> instance refer to the entity that signs the SAML Attribute Assertion. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the SAML Attribute Assertion.

A relying party SHOULD verify any signature included in the SAML Attribute Assertion and SHOULD NOT use information derived from the SAML Attribute Assertion unless the signature is verified successfully.

<saml:Subject> [Optional]

The <saml:Subject> element is an optional element used for holding “The subject of the statement(s) in the assertion” [SAML]. Each SAML Attribute Assertion used in an XACML system MUST contain a <saml:Subject> element.

In a SAML Attribute Assertion containing a <saml:Attribute> that is to be mapped to an <xacml-context:Attribute>, the <saml:Subject> instance MUST contain the identity of the entity to which the <saml:Attribute> and its value are bound. For a mapped <saml:Attribute> to be placed in a given XACML Attribute group, this identity SHOULD refer to the same entity as any

XACML Attribute that serves as an entity identifier in the Attribute group. For example, the `<saml:Subject> associated with` a mapped SAML->XACML Attribute to be placed in the XACML `<xacml-context:Resource>` Attribute group SHOULD refer to the same entity as the value of any XACML Attribute having an `AttributeId` of `urn:oasis:names:tc:xacml:1.0:resource:resource-id` that occurs in the same `<xacml-context:Resource>` instance. See Section 2.1 for more information.

`<saml:Conditions>` [Optional]

The `<saml:Conditions>` element is an optional element that is used for "conditions that MUST be taken into account in assessing the validity of and/or using the assertion" [SAML].

The `<saml:Conditions>` instance SHOULD contain `NotBefore` and `NotOnOrAfter` XML attributes to specify the limits on the validity of the SAML Attribute Assertion. If these XML attributes are present, the relying party SHOULD ensure that an `<xacml-context:Attribute>` derived from the SAML Attribute Assertion is used by a PDP for evaluating policies only when the value of the `<xacml-context:Attribute>` in the XACML Request Context having an `AttributeId` of `urn:oasis:names:tc:xacml:1.0:environment:current-dateTime` is contained within the SAML Attribute Assertion's specified validity period. The time period during which SAML Attribute Assertions are considered valid in XACML 3.0 depends on whether the PDP is configured to retrieve XACML Attributes that were valid at the time a policy was issued or at the time the policy is being evaluated.

2.3 Element `<samlp:AttributeQuery>`

The standard SAML `<samlp:AttributeQuery>` element MAY be used in an XACML system by a PEP or XACML Context Handler to request SAML Attribute Assertions from an on-line Attribute Authority for use in an XACML Request Context. The definition and use of the `<samlp:AttributeQuery>` element MUST be as described in the SAML 2.0 standard [SAML].

Note that the SAML-defined `ID` XML attribute is a required component of a `<samlp:AttributeQuery>`and can be used to correlate the `<samlp:AttributeQuery>` with the corresponding SAML Attribute Response.

2.4 Element `<samlp:Response>`: SAML Attribute Response

The response to a `<samlp:AttributeQuery>` MUST be a `<samlp:Response>` instance containing a SAML Attribute Assertion that holds any `<saml:AttributeStatement>` instances that match the query. An instance of such a `<samlp:Response>` element is called a SAML Attribute Response in this Profile. The definition and use of the SAML Attribute Response MUST be as described in the SAML 2.0 standard, augmented with the following requirements. Except as specified here, this Profile imposes no requirements or restrictions on the SAML Attribute Response and its contents beyond those specified in SAML 2.0.

`<saml:Issuer>` [Optional]

The `<saml:Issuer>` element is an optional element that "Identifies the entity that generated the response message" [SAML].

In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the `<saml:Issuer>` element refer to the entity that signs the SAML Attribute Response. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the SAML Attribute Response.

`<ds:Signature>` [Optional]

The <ds:Signature> element is an optional element for holding “An XML Signature that authenticates the responder and provides message integrity” [SAML].

A <ds:Signature> instance MAY be used in a Attribute Response. In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> refer to the entity that signs the SAML Attribute Response. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the SAML Attribute Response .

A relying party SHOULD verify any signature included in the SAML Attribute Response and SHOULD NOT use information derived from the SAML Attribute Response unless the signature is verified successfully.

3 Conveying XACML Attributes in a SOAP Message

At the time a Web Service is invoked, the service MAY need to determine whether the client is authorized to invoke the service or to access resources that are involved in the service invocation. A Web service MAY use an XACML PDP to make such an authorization decision.

When a service evaluates an XACML authorization, access control, or privacy policy related to a SOAP message, it MAY obtain the XACML Attributes required for the evaluation from various sources, including databases, registries, trusted Attribute Authorities, and so on. This work is done in the application-dependent XACML Context Handler that provides XACML Attributes to the PDP on request. A Web Services client or intermediary MAY include XACML `<xacml-context:Attribute>` instances in a `wsse:Security` SOAP Header for use by this Context Handler. This Section of this Profile describes two ways in which such `<xacml-context:Attribute>` instances MAY be provided.

3.1 <xacml-samlp:XACMLAuthzDecisionQuery>

The first way in which XACML Attributes MAY be provided to a service is by including an instance of the `<xacml-samlp:XACMLAuthzDecisionQuery>` (see Section 4.4) in the `wsse:Security` Header of a SOAP message. This query contains an XACML Request Context that SHOULD contain `<xacml-context:Attribute>` instances related to any resource access that the client will need in order to interact successfully with the service. The `<xacml-samlp:XACMLAuthzDecisionQuery>` SHOULD be signed by an entity that the Web Service trusts to authenticate the enclosed `<xacml-context:Attribute>` instances.

The Web Service MAY provide the `<xacml-context:Attribute>` instances in such an `<xacml-samlp:XACMLAuthzDecisionQuery>` to an XACML PDP as part of evaluating XACML policies related to the Web Service interaction. The service SHOULD verify that the query is signed by an entity that the service trusts to authenticate the enclosed `<xacml-context:Attribute>` instances. It SHOULD verify that the `IssueInstant` of the `<xacml-samlp:XACMLAuthzDecisionQuery>` is close enough the the current time to meet the validity requirements of the service.

3.2 SAML Attribute Assertion

A second way in which XACML Attributes MAY be provided to a service is in the form of a SAML Attribute Assertion in the `wsse:`Security Header of a SOAP message. The SAML Attributes contained in the SAML Attribute Assertion MAY be converted to XACML Attributes as described in Section 2.1 of this Profile by an XACML Context Handler for use by a PDP associated with the Web Service in evaluating XACML policies related to the Web Service interaction.

4 Authorization Decisions

XACML defines <xacml-context:Request> and <xacml-context:Response> elements for describing an authorization decision request and the corresponding response from a PDP. In many environments, instances of these elements need to be signed or associated with a validity period in order to be used in an actual protocol between entities. Although SAML 2.0 defines a rudimentary <samlp:AuthzDecisionQuery> in the SAML Protocol Schema and a rudimentary <saml:AuthzDecisionStatement> in the SAML Assertion Schema, these elements are not able to convey all the information that an XACML PDP is capable of accepting as part of its Request Context or conveying as part of its XACML Response Context. In order to allow a PEP to use the SAML protocol with full support for the XACML Request Context and XACML Response Context syntax, this Profile defines one SAML extension type and one SAML extension element, and describes how they are used with other standard SAML elements.

- <xacml-saml:XACMLAuthzDecisionStatementType> is a new SAML extension type that includes an XACML <xacml-context:Response> along with other optional information.
- A <saml:Statement> of type <xacml-saml:XACMLAuthzDecisionStatementType> (defined using xsi:type) MAY be used by a PDP Context Handler to convey an XACML <xacml-context:Response> along with other optional information. An instance of such a <saml:Statement> element is called an XACMLAuthzDecision Statement in this Profile.
- A <saml:Assertion> MUST be used to hold XACMLAuthzDecision Statements. An instance of such a <saml:Assertion> element is called an XACMLAuthzDecision Assertion in this Profile.
- A <xacml-samlp:XACMLAuthzDecisionQuery> is a new SAML extension element that MAY be used by a PEP to submit an XACML Request Context, along with other optional information, as a SAML protocol query to an XACML Context Handler.
- A <samlp:Response> containing an XACMLAuthzDecision Assertion MUST be used by an XACML Context Handler as the response to an <saml-samlp:XACMLAuthzDecisionQuery>. An instance of such a <samlp:Response> element is called an XACMLAuthzDecision Response in this Profile.

This Section defines and describes the usage of these types and elements.. The schemas for the new type and element are contained in the [XACML-SAML] and [XACML-SAMLP] schema documents.

4.1 Type <xacml-saml:XACMLAuthzDecisionStatementType>

The new <xacml-saml:XACMLAuthzDecisionStatementType> complex type contains an XACML Response Context along with related information. Use of this type is an alternative to use of the SAML-defined <saml:AuthzDecisionStatementType>; this alternative allows an XACML Context Handler to use SAML with full support for XACML authorization decisions. An instance of a <saml:Statement> element that is of this type (defined using xsi:type="xacml-saml:XACMLAuthzDecisionStatementType") is called an XACMLAuthzDecision Statement in this Profile.

```
<complexType name="XACMLAuthzDecisionStatementType">
   <complexContent>
       <extension base="saml:StatementAbstractType">
           <sequence>
               <element ref="xacml-context:Response"/>
               <element ref="xacml-context:Request" minOccurs="0"/>
           </sequence>
       </extension>
   </complexContent>
</complexType>
```

The <xacml-saml:XACMLAuthzDecisionStatementType> complex type is an extension to the SAML-defined <saml:StatementAbstractType>. It contains the following elements:

<xacml-context:Response> [Required]

An XACML Response Context created by an XACML PDP. This Response MAY be the result of evaluating an XACML Request Context from an <xacml-samlp:XACMLAuthzDecisionQuery>.

<xacml-context:Request> [Optional]

An <xacml-context:Request> element containing <xacml-context:Attribute> instances that were used by the XACML PDP in evaluating policies to obtain the corresponding<xacml-context:Response>.

If the XACMLAuthzDecision Statement represents a response to an <xacml-samlp:XACMLAuthzDecisionQuery>, and if the ReturnContext XML attribute in the <xacml-samlp:XACMLAuthzDecisionQuery> instance is "true", then this element MUST be included; if the ReturnContext XML attribute in the <xacml-samlp:XACMLAuthzDecisionQuery> instance is "false", then this element MUST NOT be included. See the description of the ReturnContext XML attribute in Section 4.4 for a specification of the <xacml-context:Attribute> instances that MUST be returned in this element when it is part of a response to an <xacml-samlp:XACMLAuthzDecisionQuery>.

If the XACMLAuthzDecision Statement does not represent the response to an <xacml-samlp:XACMLAuthzDecisionQuery>, then this element MAY be included. In this case, the PDP MUST determine which <xacml-context:Attribute> instances are included using criteria that are outside the scope of this Profile.

4.2 Element <saml:Statement>: XACMLAuthzDecision Statement

A <saml:Statement> instance MAY be of type <xacml-saml:XACMLAuthzDecisionStatementType> by using xsi:type as shown in the example in Section 4.3. An instance of a <saml:Statement> element that is of type <xacml-saml:XACMLAuthzDecisionStatementType> is called an XACMLAuthzDecision Statement in this Profile. Any instance of an XACMLAuthzDecision Statement in an XACML system MUST be enclosed in a <saml:Assertion>.

4.3 Element <saml:Assertion>: XACMLAuthzDecision Assertion

A <saml:Assertion> instance MAY contain an XACMLAuthzDecision Statement as shown in the following non-normative example:

```
<saml:Assertion Version="2.0" ID="9812368"
       IssueInstant="2006-05-31T13:20:00.000">
    <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
    <saml:Statement
          xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
        <xacml-context:Response>
            <xacml-context:Result>
                <xacml-context:Decision>
                   NotApplicable
                </xacml-context:Decision>
            </xacml-context:Result>
        </xacml-context:Response>
        <xacml-context:Request>
            ....
        </xacml-context:Request>
    </saml:Statement>
</saml:Assertion>
```

An instance of a `<saml:Assertion>` element containing an XACMLAuthzDecision Statement is called an XACMLAuthzDecision Assertion in this Profile.

This Profile imposes the following requirements and restrictions on the `<saml:Assertion>` element beyond those specified in SAML 2.0 when used as an XACMLAuthzDecision Assertion.

`<saml:Issuer>` [Required]

The `<saml:Issuer>` element is a required element for holding information about “the SAML authority that is making the claim(s) in the assertion” [SAML].

In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the `<saml:Issuer>` element refer to the entity that signs the XACMLAuthzDecision Assertion. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the XACMLAuthzDecision Assertion.

`<ds:Signature>` [Optional]

The `<ds:Signature>` element is an optional element for holding “An XML Signature that authenticates the assertion, as described in Section 5 of the SAML 2.0 core specification [SAML].”

A `<ds:Signature>` instance MAY be used in a `<saml:Assertion>`. In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the `<saml:Issuer>` instance refer to the entity that signs the XACMLAuthzDecision Assertion. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the Assertion .

A relying party SHOULD verify any signature included in the XACMLAuthzDecision Assertion and SHOULD NOT use information derived from the Assertion unless the signature is verified successfully.

`<saml:Subject>` [Optional]

The `<saml:Subject>` element MUST NOT be included in an XACMLAuthzDecision Assertion. Instead, the Subject of an XACMLAuthzDecision Assertion is specified in the XACML Request Context of the corresponding authorization decision request. This corresponding XACML Request Context MAY be included in the XACMLAuthzDecision Statement as described in Section 4.1.

`<saml:Conditions>` [Optional]

The `<saml:Conditions>` element is an optional element that is used for "conditions that MUST be taken into account in assessing the validity of and/or using the assertion" [SAML].

The `<saml:Conditions>` instance SHOULD contain `NotBefore` and `NotOnOrAfter` XML attributes to specify the limits on the validity of the XACMLAuthzDecision Assertion. If these XML attributes are present, the relying party SHOULD ensure that an `<xacml-context:Response>` taken from the XACMLAuthzDecision Assertion is used only during the Assertion's specified validity period.

4.4 Element `<xacml-samlp:XACMLAuthzDecisionQuery>`

The `<xacml-samlp:XACMLAuthzDecisionQuery>` protocol element MAY be used by a PEP to request an authorization decision from an XACML PDP. This element is an alternative to the SAML-defined `<samlp:AuthzDecisionQuery>`; this alternative allows the PEP to use the full capabilities of an XACML PDP. It allows use of the SAML query protocol to convey an XACML Request Context along with related information.

```
<element name="XACMLAuthzDecisionQuery"
         xsi:type="xacml-samlp:XACMLAuthzDecisionQueryType" />
<complexType name="XACMLAuthzDecisionQueryType">
    <complexContent>
        <extension base="samlp:RequestAbstractType">
            <sequence>
                <element ref="xacml-context:Request"/>
                <element ref="xacml-samlp:AdditionalAttributes"
minOccurs="0" maxOccurs="1"/>
                <element ref="xacml:Policy"
                    minOccurs="0" maxOccurs="unbounded" />
                <element ref="xacml:PolicySet"
                    minOccurs="0" maxOccurs="unbounded" />
                <element ref="xacml-saml:ReferencedPolicies"
minOccurs="0" maxOccurs="1" />
           </sequence>
           <attribute name="InputContextOnly"
                          type="boolean"
                          use="optional"
                          default="false"/>
           <attribute name="ReturnContext"
                          type="boolean"
                          use="optional"
                          default="false"/>
           <attribute name=”CombinePolicies”
                          type="boolean"
                          use="optional"
                          default="true"/>
        </extension>
    </complexContent>
</complexType>
```

The <xacml-samlp:XACMLAuthzDecisionQuery> element is of <xacml-samlp:XACMLAuthzDecisionQueryType> complex type, which is an extension to the SAML-defined <samlp:RequestAbstractType>.

The <xacml-samlp:XACMLAuthzDecisionQuery> element contains the following XML attributes and elements in addition to those defined for the <samlp:RequestAbstractType>:

InputContextOnly [Default “false”]

This XML attribute governs the sources of information that the PDP is allowed to use in making its authorization decision. If the value of this XML attribute is “true”, then the authorization decision MUST be made solely on the basis of information contained in the <xacml-samlp:XACMLAuthzDecisionQuery>; external XACML Attributes MUST NOT be used. If the value of this XML attribute is “false”, then the authorization decision MAY be made on the basis of XACML Attributes not contained in the <xacml-samlp:XACMLAuthzDecisionQuery>.

ReturnContext [Default “false”]

This XML attribute allows the PEP to request that an <xacml-context:Request> instance be included in the XACMLAuthzDecision Statement resulting from the query. It also governs the contents of that <xacml-context:Request> instance.

If this attribute is "True", then the PDP SHALL include the <xacml-context:Request> element in the <XACMLAuthzDecisionStatement> element in the <XACMLResponse>. This <xacml-context:Request> element SHALL include all those XACML Attributes supplied by the PEP in the

<XACMLAuthzDecisionQuery> that were used in making the authorization decision. A conforming PDP MAY omit those XACML Attributes which were not referenced in any policy which was evaluated in making the decision. If the value of the InputContextOnly Attribute in the Request is "False", the PDP MAY include additional XACML Attributes in this <xacml-context:Request> element, which were obtained by the PDP and used in making the authorization decision.

If this XML attribute is “false”, then the PDP MUST NOT include an <xacml-context:Request> instance in the XACMLAuthzDecision Statement in the XACMLAuthzDecision Response.

CombinePolicies [Default “true”]

This XML attribute allows the PEP to specify whether policies supplied in <xacml:Policy> and <xacml:PolicySet> elements of the <xacml-samlp:XACMLAuthzDecisionQuery> are to be combined with other policies available to the PDP during evaluation.

If the attribute value is “true”, then the PDP MUST insert all policies passed in the<xacml:Policy> and <xacml:PolicySet> elements in the <xacml-samlp:XACMLAuthzDecisionQuery> into the set of policies or policy sets that define the PDP as specified in Section 7.11 of the XACML 3.0 core specification [XACML3]. They MUST be combined with the other policies using the policy combining algorithm that defines the PDP as specified in Section 7.11 of the XACML 3.0 core specification [XACML3]. If the policy combining algorithm that defines the PDP is one in which element order is considered, then the policies passed in the XACMLAuthzDecision Query MUST be considered in the order in which they appear in the <xacml-samlp:XACMLAuthzDecisionQuery> and MUST be considered as preceding all other policies that define the PDP.

If the attribute value is “false”, then there MUST be no more than one <xacml:Policy> or <xacml:PolicySet> passed in the <xacml-samlp:XACMLAuthzDecisionQuery>. This policy MUST be treated as the policy that defines the PDP as specified in Section 7.11 of the XACML 3.0 core specification [XACML3] for evaluation of the <xacml-context:Request> passed in the <xacml-samlp:XACMLAuthzDecisionQuery>. It MUST NOT be used to evaluate any other <xacml-context:Request> instances unless provided to the PDP independent of the particular<xacml-context:Request>.

<xacml-context:Request> [Required]

An XACML Request Context that is to be evaluated.

<xacml-samlp:AdditionalAttributes> [Zero or One]

Entity descriptions and corresponding <xacml-context:Attribute> instances that apply to them. This element is used only with XACML 3.0 Administrative Policy [ADMIN] functionality.

<xacml:Policy> [Any Number]

Optional XACML Policy instances that MUST be used only for evaluating this authorization decision request.

If the CombinePolicies XML attribute is “true”, then the PDP MUST use such XACML Policy instances.

If the CombinePolicies XML attribute is “false”, then the PDP MUST use this XACML Policy instance. There MUST be only one such XACML Policy instance and there MUST NOT be any XACML PolicySet instances in this <xacml-samlp:XACMLAuthzDecisionQuery> instance.

<xacml:PolicySet> [Any Number]

Optional XACML PolicySet instances that MUST be used only for evaluating this authorization decision request.

If the CombinePolicies XML attribute is “true”, then the PDP MUST use such XACML PolicySet instances.

If the CombinePolicies XML attribute is “false”, then the PDP MUST use this XACML PolicySet instance. There MUST be only one such XACML PolicySet instance and there MUST NOT be any XACML Policy instances in this XACMLAuthzDecision Query.

<xacml-saml:ReferencedPolicies> [Zero or One]

With the exception of XACML Policy and PolicySet instances that the receiver of the XACMLAuthzDecision Statement is not authorized to view, this element MAY contain XACML Policy and PolicySet instances required to resolve <xacml:PolicySetIdReference> or <xacml:PolicyIdReference> instances contained in the XACMLAuthzDecision Statement, including those in the <xacml-saml:ReferencedPolicies> instance itself, or contained in the policies already available to the PDP. The values of the PolicyId and PolicySetId XML attributes of the policies included in the <xacml-saml:ReferencedPolicies> instance MUST exactly match the values contained in the corresponding <xacml:PolicySetIdReference> or <xacml:PolicyIdReference> instances.

4.5 Element <xacml-samlp:AdditionalAttributes>

This element applies only for use with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML 3.0 PDP.

In some cases it may be useful for the PEP to provide attributes for delegates with the authorization decision request. Since the Request Contexts used in reduction are not formed until after the access request is submitted to the PDP, the delegate attributes need to be treated differently from the attributes part of the access **Request Context**. The following defines elements that MAY be used to submit XACML Attributes for this purpose. The XACML Attributes MUST be made available by the Context Handler when the reduction Request Contexts are created.

```
<element name="AdditionalAttributes"
  type="xacml-samlp: AdditionalAttributesType"/>
<complexType name="AdditionalAttributesType">
  <sequence>
    <element ref="xacml-samlp:AssignedAttributes" minOccurs="0"
maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

The <AdditionalAttributes> element is of AdditionalAttributesType complex type.

The <AdditionalAttributes> element contains the following elements:

<AssignedAttributes> [Required]

Assignment of a set of XACML Attributes to specified delegate entities.

4.6 Element <xacml-samlp:AssignedAttributes>

This element is used only with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML 3.0 PDP.

The `<AssignedAttributes>` element MUST contain XACML Attributes that apply to delegate entities identified by the `<xacml-samlp:Holders>` element.

```
<element name="AssignedAttributes" type="xacml-samlp:AssignedAttributesType"/>
<complexType name="AssignedAttributesType">
  <sequence>
    <element ref="xacml-samlp:Holders"/>
    <element ref="xacml-samlp:HolderAttributes"/>
  </sequence>
</complexType>
```

The `<AssignedAttributes>` element is of `AssignedAttributesType` complex type.

The `<AssignedAttributes>` element contains the following elements:

`<xacml-samlp:Holders>` [Required]

The identities of the delegate entities to which the provided XACML Attributes apply.

`<xacml-samlp:HolderAttributes>` [Required]

The XACML Attributes of the delegate entity.

4.7 Element `<xacml-samlp:Holders>`

This element is used only with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML 3.0 PDP.

The `<Holders>` element MUST identify the delegate entities to which the provided `<xacml-samlp:HolderAttributes>` elements apply.

```
<element name="Holders" type="xacml-samlp:HoldersType"/>
<complexType name="HoldersType">
  <sequence>
    <element ref="xacml:Match" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

The `<xacml-samlp:Holders>` element is of `<xacml-samlp:HoldersType>` complex type.

The `<xacml-samlp:Holders>` element contains the following elements:

`<xacml:Match>` [One to many, required]

Matches the delegate entities to which the XACML Attributes in the associated `<xacml-samlp:HolderAttributes>` element apply. The `<Match>` elements shall be evaluated according to the XACML schema against the `<Attributes>` elements in a `<Request>` during reduction. If any `<Match>` element evaluates to "Match" then the supplied attributes shall apply to the <Attributes> element which was referenced by the attribute designator or selector contained in the `<Match>` element

4.8 Element `<xacml-samlp:HolderAttributes>`

This element is used only with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML 3.0 PDP.

The <xacml-samlp:HolderAttributes> element MUST contain XACML Attributes that apply to the delegate entities identified in the corresponding <xacml-samlp:Holders> element.

```
<element name="HolderAttributes" type="xacml-samlp:HolderAttributesType"/>
<complexType name="HolderAttributesType">
  <sequence>
    <element ref="xacml-context:Attribute"
        minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

The <xacml-samlp:HolderAttributes> element is of <xacml-samlp:HolderAttributesType> complex type.

The <xacml-samlp:HolderAttributes> element contains the following elements:

<xacml-context:Attribute> [any number]

> An XACML Attribute of the delegate entities identified in the corresponding <xacml-samlp:Holders> element.

4.9 Element <xacml-saml:ReferencedPolicies>

An instance of this element MAY be used to contain copies of policies referenced from <xacml:Policy> or <xacml:PolicySet> instances included in an XACMLAuthzDecision Statement or in an XACMLPolicy Statement, as well as copies of all policies referenced from other policies included in the <xacml-saml:ReferencedPolicies> instance or policies already present in the PDP If a <xacml:Policy> or <xacml:PolicySet> instance would match a policy both among the policies already present to the PDP as well as a policy contained in the supplied <xacml-saml:ReferencedPolicies> instance, then the supplied policy takes precedence.

```
<element name="ReferencedPolicies"
     type="xacml-saml:ReferencedPoliciesType"/>
<complexType name="ReferencedPoliciesType">
    <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xacml:Policy"/>
            <element ref="xacml:PolicySet"/>
        </choice>
    </sequence>
</complexType>
```

The <xacml-saml:ReferencedPolicies> element is of <xacml-saml:ReferencedPoliciesType> complex type.

The <xacml-saml:ReferencedPolicies> element contains the following elements:

<xacml:Policy> [any number]

> A single <xacml:Policy> that is referenced using an <xacml:PolicyIdReference> from another <xacml:Policy> or <xacml:PolicySet> instance. The value of the PolicyId XML attribute in the <xacml:Policy> MUST be equal to the value of the corresponding <xacml:PolicyIdReference> element.

<xacml:PolicySet> [any number]

> A single <xacml:PolicySet> that is referenced using an <xacml:PolicySetIdReference> from another <xacml:Policy> or <xacml:PolicySet> instance. The value of the PolicySetId

XML attribute in the <xacml:PolicySet> MUST be equal to the value of the corresponding <xacml:PolicySetIdReference> element.

4.10 Element <samlp:Response>: XACMLAuthzDecision Response

A <samlp:Response> instance MAY contain an XACMLAuthzDecision Assertion as shown in the following non-normative example:

```
<samlp:Response Version="2.0" ID="9812368"
       IssueInstant="2006-05-31T13:20:00.000">
   <saml:Assertion Version="2.0" ID="9812368"
       IssueInstant="2006-05-31T13:20:00.000">
      <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
      <saml:Statement
          xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
        <xacml-context:Response>
            <xacml-context:Result>
                <xacml-context:Decision>
                   NotApplicable
                </xacml-context:Decision>
            </xacml-context:Result>
        </xacml-context:Response>
        <xacml-context:Request>
            ....
        </xacml-context:Request>
      </saml:Statement>
   </saml:Assertion>
</samlp:Response>
```

An instance of a <samlp:Response> element containing an XACMLAuthzDecision Assertion is called an XACMLAuthzDecision Response in this Profile. Such a Response MUST be used as the response to an <xacml-samlp:XACMLAuthzDecisionQuery>.

This Profile imposes the following requirements or restrictions on the <samlp:Response> element in addition to those specified in SAML 2.0 when used as an XACMLAuthzDecision Response.

<saml:Issuer> [Optional]

The <saml:Issuer> element is an optional element that “Identifies the entity that generated the response message” [SAML].

In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> element refer to the entity that signs the XACMLAuthzDecision Response. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the Response.

<ds:Signature> [Optional]

The <ds:Signature> element is an optional element for holding “An XML Signature that authenticates the responder and provides message integrity” [SAML].

A <ds:Signature> instance MAY be used in a XACMLAuthzDecision Response. In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> instance refer to the entity that signs the XACMLAuthzDecision Response. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the Response.

A relying party SHOULD verify any signature included in the XACMLAuthzDecision Response and SHOULD NOT use information derived from the Response unless the signature is verified successfully.

`<saml:Assertion>` [Any Number]

`<saml:Assertion>` instances that MAY include one or more XACMLAuthzDecision Assertions that represent responses to associated queries.

`<samlp:StatusCode>` [Required]

The `<samlp:StatusCode>` element is a component of the `<samlp:Status>` element in the `<samlp:Response>`.

In the response to an `<xacml-samlp:XACMLAuthzDecisionQuery>`, the `<samlp:StatusCode>` `Value` XML attribute MUST depend on the value of the `<xacml-context:StatusCode>` instance of the XACML Response Context `<xacml-context:Status>` instance as follows:

`urn:oasis:names:tc:SAML:2.0:status:Success`

This value for the `<samlp:StatusCode>` `Value` XML attribute MUST be used if and only if the `<xacml-context:StatusCode>` value is `urn:oasis:names:tc:xacml:1.0:status:ok`.

`urn:oasis:names:tc:SAML:2.0:status:Requester`

This value for the `<samlp:StatusCode>` `Value` XML attribute MUST be used when the `<xacml-context:StatusCode>` value is `urn:oasis:names:tc:xacml:1.0:status:missing-attribute` or when the `<xacml-context:StatusCode>` value is `urn:oasis:names:tc:xacml:1.0:status:syntax-error` due to a syntax error in the `<xacml-context:Request>`.

`urn:oasis:names:tc:SAML:2.0:status:Responder`

This value for the `<samlp:StatusCode>` `Value` XML attribute MUST be used when the `<xacml-context:StatusCode>` value is `urn:oasis:names:tc:xacml:1.0:status:syntax-error` due to a syntax error in an `<xacml:Policy>` or `<xacml:PolicySet>`. Note that not all syntax errors in policies will be detected in conjunction with the processing of a particular query, so not all policy syntax errors will be reported this way.

`urn:oasis:names:tc:SAML:2.0:status:VersionMismatch`

This value for the `<samlp:StatusCode>` `Value` XML attribute MUST be used only when the SAML interface at the PDP does not support the version of the SAML schema used in the query.

`InResponseTo` [Optional]

This optional XML attribute is “A reference to the identifier of the request to which the response corresponds.” When the XACMLAuthzDecision Response is issued in response to an XACMLAuthzDecision Query, this XML attribute MUST contain the value of the `ID` XML attribute from the XACMLAuthzDecision Query to which this is a response. This allows the receiver to correlate the XACMLAuthzDecision Response with the corresponding XACMLAuthzDecision Query. The SAML-defined `ID` XML attribute is a required component of an instance of the `<samlp:RequestAbstractType>` of which the `<xacml-samlp:XACMLAuthzDecisionQuery>` is an extension.

4.11 Functional Requirements for the <xacml-samlp:AssignedAttributes> Element

During processing of the provided access request, if the <xacml-samlp:Holders> element of a provided <xacml-samlp:AssignedAttributes> element matches a section of the XACML Request Context, then the XACML Context Handler MUST make the XACML Attributes in the <xacml-samlp:HolderAttributes> element appear in that section of the XACML Request Context. Any inheritance between <xacml-samlp:AssignedAttributes> elements is not deduced.

The matching of additional XACML Attributes MUST be made against all Request Contexts involved in the processing of the XACMLAuthzDecision Query, including the provided access request itself and any Request Contexts formed as part of reduction.

The provided XACML Attributes MUST be used only in the evaluation of the provided access request and any derived Request Contexts, including reduction, and MUST NOT be used in evaluation of requests not related to the provided access request unless associated with those other requests independent of the <xacml-samlp:XACMLAuthzDecisionQuery>.

The implementation MUST match the <xacml-samlp:Holders> element against all the attributes available to the context handler, but MUST NOT use any matching <xacml-samlp:HolderAttributes> to find even more attributes through the context handler or even more supplied attributes through other <xacml-samlp:Holders> elements. This implies that there can be no inheritance between <xacml-samlp:AssignedAttributes> elements.

5 XACML Decision Queries using WS-Trust

In some environments, it may be desirable to obtain an XACML authorization decision from a Security Token Service (STS) using the WS-Trust protocol WSTRUST].

5.1 Common Claims Dialect

One method of doing this is to support the Common Claim Dialect as defined in WS-Federation [WSFED], chapter 9. In this case the implementation must map the contents of an incoming <RequestSecurityToken> element into a XACML <Request> element and map the XACML <Response> into an outgoing <RequestSecurityTokenResponseCollection> element. When this approach is taken, there is no explicit reference to XACML in the wire protocol and in general a requestijg party will not be aware whether or not an XACML-based PDP was used to make the decision.

5.2 XACML Dialect

This section defines a WS-Trust-based protocol which is intended to easier and more efficient for XACML PDP to implement. It is based directly on the constructs previously defined in Section 4. It uses the <saml:Assertion> element and <saml:Statement> of type xacml-saml:XACMLAuthzDecisionStatementType to wrap the XACML <Request> and <Response> elements. However, the <xacml-samlp:XACMLDecisionQuery> and <samlp:Response> elements are not used. Instead the request is conveyed in a <wst:RequestSecurityToken> element and the response is carried in a <wst:RequestSecurityTokenResponseCollection> element containing a <wst:RequestSecurityTokenResponse> element.

Except for the outer protocol layer, described in more detail below, the syntax and functional requirements for this protocol is exactly as described above in section 4. In fact, it is possible for a server which contains an XACML PDP to support both protocols, using distinct web service endpoints, with only a small amount of distinct code to handle each request type.

5.3 Decision Request

The decision request is contained in a <wst:RequestSecurityToken> element. This element contains the following attributes and elements from the WS-Trust schema.

- Context This URI specifies an identifier for this request. Its value will be returned in the corresponding response to allow them to be correlated.
- <wst:TokenType> This element contains the value: urn:oasis:names:tc:xacml:3.0:core:schema, to indicate that an XACML decision token will be returned.
- <wst:RequestType> This element contains the value: http://docs.oasis-open.org/ws-sx-ws-trust/200512/Issue

In addition, the <wst:RequestSecurityToken> element MAY contain any of the attributes and elements defined in section 4.4 above as being contained in the <xacml-samlp:XACMLAuthzDecisionQuery> element. Specifically these are the attributes:

- InputContextOnly,
- ReturnContext, and
- CombinePolicies.

These are the elements:

- <xacml-context:Request>,
- <xacml-samlp:AdditionalAttributes>,
- <xacml:Policy>,
- <xacml:PolicySet>, and
- <xacml-saml:ReferencedPolicies>.

The functional requirements for processing these attributes and elements are exactly as set forth in section 4 above.

5.4 Decision Response

The decision response is contained in a <wst:RequestTokenResponseCollection> element. It contains exactly one <wst:RequestTokenResponse> element. This element contains the following attributes and elements.

- Context This element contains the same URI provided in the Context attribute of the request.
- <wst:RequestedSecurityToken> This element contains a <saml:Assertion which in turn contains a <saml:Statement of type xacml-saml:XACMLAuthzDecisionStatementType as described in secitons 4.1, 4.2, and 4.3 above. The functional requirements for processing these attributges and elements are exactly as set forth in section 4 above.

6 Policies

XACML defines the <xacml:Policy> and <xacml:PolicySet> elements for expressing policies. In many environments, instances of these elements need to be stored or transmitted between entities in an XACML system. Such instances may need to be signed or associated with a validity period. SAML is intended to provide this functionality for security-related assertions, but SAML does not define any Protocol or Assertion elements for policies. In order to allow entities in an XACML system to use SAML assertions and protocols to store, transmit, and query for XACML policies, this Profile defines one SAML extension type and one SAML extension element, and describes how they are used with other standard SAML elements.

- <xacml-saml:XACMLPolicyStatementType> is a new SAML extension type that includes XACML policies.
- A <saml:Statement> defined using xsi:type="xacml-saml:XACMLPolicyStatementType" MAY be used in an XACML system to store or convey XACML policies. An instance of a <saml:Statement> element defined using this type is called an XACMLPolicy Statement in this Profile.
- A <saml:Assertion> MUST be used to hold XACMLPolicy Statements. An instance of such a <saml:Assertion> element is called an XACMLPolicy Assertion in this Profile.
- An <xacml-samlp:XACMLPolicyQuery> is a new SAML extension element that MAY be used by a PDP or other entity to request XACML policies as a SAML protocol query.
- A <samlp:Response> containing an XACMLPolicy Assertion that MUST be used in response to an <xacml-samlp:XACMLPolicyQuery>. It MAY be used to transmit XACML policies in other contexts. An instance of such a <samlp:Response> is called an XACMLPolicy Response in this Profile.

This Section defines and describes the usage of these types and elements. The schemas for the new type and element are contained in the [XACML-SAML] and [XACML-SAMLP] schema documents.

6.1 Type <xacml-saml:XACMLPolicyStatementType>

The <xacml-saml:XACMLPolicyStatementType> complex type contains XACML Policy and or XACML PolicySet elements. An instance of a <saml:Statement> element that is of this type is called an XACMLPolicy Statement in this Profile.

```
    <complexType name="XACMLPolicyStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <sequence>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="xacml:Policy"/>
                        <element ref="xacml:PolicySet"/>
                    </choice>
               <element ref="xacml-saml:ReferencedPolicies"
    minOccurs="0" maxOccurs="1" />
                </sequence>
            </extension>
        </complexContent>
    </complexType>
```

The <xacml-saml:XACMLPolicyStatementType> complex type is an extension to the SAML-defined <saml:StatementAbstractType>. It contains the following elements.

<xacml:Policy> [Any Number]

If the XACMLPolicy Statement represents a response to an <xacml-samlp:XACMLPolicyQuery>, then this element MUST contain one of the <xacml:Policy> instances that meet the specifications of the associated <xacml-samlp:XACMLPolicyQuery>. Otherwise, this element MAY contain an arbitrary <xacml:Policy> instance.

<xacml:PolicySet> [Any Number]

If the XACMLPolicy Statement represents a response to an <xacml-samlp:XACMLPolicyQuery>, then this element MUST contain one of the <xacml:PolicySet> instances that meet the specifications of the associated <xacml-samlp:XACMLPolicyQuery>. Otherwise, this element MAY contain an arbitrary <xacml:PolicySet> instance.

<xacml-saml:ReferencedPolicies> [Zero or One]

With the exception of XACML Policy and PolicySet instances that the receiver of the XACMLPolicy Statement is not authorized to view, this element MAY contain XACML Policy and PolicySet instances required to resolve <xacml:PolicySetIdReference> or <xacml:PolicyIdReference> instances contained in the XACMLPolicy Statement, including those in the <xacml-saml:ReferencedPolicies> instance itself. The values of the PolicyId and PolicySetId XML attributes of the policies included in the <xacml-saml:ReferencedPolicies> instance MUST exactly match the values contained in the corresponding <xacml:PolicySetIdReference> or <xacml:PolicyIdReference> instances.

Subject to authorization and availability, if the XACMLPolicy Statement is issued in response to an <xacml-samlp:XACMLPolicyQuery>, there MUST be exactly one <xacml:Policy> element included for every XACML Policy that satisfies the XACMLPolicy Query, and there MUST be exactly one <xacml:PolicySet> element included for every XACML PolicySet that satisfies the XACMLPolicy Query . The responder MUST return all XACML policies available to the responder that satisfy the <xacml-samlp:XACMLPolicyQuery> and that the requester is authorized to receive.

If the XACMLPolicy Statement is issued in response to an <xacml-samlp:XACMLPolicyQuery>, and there are no <xacml:Policy> or <xacml:PolicySet> instances that meet the specifications of the associated <xacml-samlp:XACMLPolicyQuery>, then there MUST be exactly one empty XACMLPolicy Statement included in the response.

An XACMLPolicy Statement enclosed in a signed SAML assertion MAY be used as a method of authentication of XACML policies. In this case the Policy or PolicySet MUST NOT contain an XACML <PolicyIssuer> element. Instead the PDP MAY generate a <PolicyIssuer> element from the certificate or other security token associated with the signature of the SAML assertion before using the policy for XACML request evaluation. In this case the issuer of the SAML assertion SHALL be translated into an XACML attribute with id urn:oasis:names:tc:xacml:1.0:subject:subject-id. This does that mean that the issuer name must be taken directly from the security token, merely that the PDP perform some mapping on the claims in the token to determine the issuer.

6.2 Element <xacml-saml:ReferencedPolicies>

An instance of this element MAY be used to contain copies of policies referenced from <xacml:Policy> or <xacml:PolicySet> instances included in the <xacml-samlp:XACMLPolicyQuery>, as well as copies of policies referenced from other policies included in the <xacml-saml:ReferencedPolicies> instance.

See Section 4.9 for a description of the <xacml-saml:ReferencedPolicies> element.

6.3 Element <saml:Statement>: XACMLPolicy Statement

A <saml:Statement> instance MAY be of defined to be of type <xacml-saml:XACMLPolicyStatementType> by using xsi:type="xacml-saml:XACMLPolicyStatementType" as shown in the example in Section 6.4. such an instance of a <saml:Statement> element is called an XACMLPolicy Statement in this Profile. Any instance of an XACMLPolicy Statement in an XACML system MUST be enclosed in a <saml:Assertion>.

6.4 Element <saml:Assertion>: XACMLPolicy Assertion

A <saml:Assertion> instance MAY contain an XACMLPolicy Statement as shown in the following non-normative example:

```
<saml:Assertion Version="2.0" ID="9812368"
       IssueInstant="2006-05-31T13:20:00.000">
    <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
    <saml:Statement
          xsi:type="xacml-saml:XACMLPolicyStatementType">
        <xacml:Policy PolicyId="policy:1" RuleCombiningAlgId="..">
            ....
        </xacml:Policy>
        <xacml:PolicySet PolicySetId="policyset:5" ... >
            ...
        </xacml:PolicySet>
    </saml:Statement>
</saml:Assertion>
```

An instance of a <saml:Assertion> element containing an XACMLPolicy Statement is called an XACMLPolicy Assertion in this Profile.

When an XACMLPolicy Assertion is part of a response to an <xacml-samlp:XACMLPolicyQuery>, then the XACMLPolicy Assertion MUST contain exactly one XACMLPolicy Statement, which in turn MAY contain any number of XACML Policy and PolicySet instances.

This Profile imposes the following requirements and restrictions on the <saml:Assertion> element beyond those specified in SAML 2.0 when used as an XACMLPolicy Assertion.

<saml:Issuer> [Required]

The <saml:Issuer> element is a required element for holding information about “the SAML authority that is making the claim(s) in the assertion” [SAML].

In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> element refer to the entity that signs the XACMLPolicy Assertion. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the XACMLPolicy Assertion.

<ds:Signature> [Optional]

The <ds:Signature> element is an optional element for holding “An XML Signature that authenticates the assertion, as described [in Section 5 of the SAML 2.0 core specification[SAML]].”

A <ds:Signature> instance MAY be used in an XACMLPolicy Assertion. In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Assertion. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the XACMLPolicy Assertion.

A relying party SHOULD verify any signature included in the XACMLPolicy Assertion and SHOULD NOT use information derived from the XACMLPolicy Assertion unless the signature is verified successfully.

`<saml:Subject>` [Optional]

The `<saml:Subject>` element MUST NOT be included in an XACMLPolicy Assertion. Instead, the Subjects of an XACMLPolicy Assertion are specified in the XACML Policy and PolicySet elements contained in the enclosed XACMLPolicy Statement.

`<saml:Conditions>` [Optional]

The `<saml:Conditions>` element is an optional element that is used for “conditions that MUST be taken into account in assessing the validity of and/or using the assertion” [SAML].

The `<saml:Conditions>` instance SHOULD contain `NotBefore` and `NotOnOrAfter` XML attributes to specify the limits on the validity of the XACMLPolicy Assertion. If these XML attributes are present, the relying party SHOULD ensure that an `<xacml-context:Response>` taken from the XACMLPolicy Assertion is used only during the XACMLPolicy Assertion's specified validity period.

6.5 Element `<xacml-samlp:XACMLPolicyQuery>`

An instance of the `<xacml-samlp:XACMLPolicyQuery>` protocol element MAY be used by a PDP or application to request XACML `<xacml:Policy>` or `<xacml:PolicySet>` instances from an on-line Policy Administration Point.

```
    <element name="XACMLPolicyQuery"
        xsi:type="xacml-samlp:XACMLPolicyQueryType" />
    <complexType name="XACMLPolicyQueryType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <choice minOccurs="1" maxOccurs="unbounded">
                    <element ref="xacml-context:Request"/>
                    <element ref="xacml:PolicySetIdReference"/>
                    <element ref="xacml:PolicyIdReference"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>
```

The `<xacml-samlp:XACMLPolicyQuery>` element is of `<xacml-samlp:XACMLPolicyQueryType>` complex type, which is an extension to the SAML-defined `<samlp:RequestAbstractType>`.

The `<xacml-samlp:XACMLPolicyQuery>` element contains zero or more of the following elements in addition to those defined for the `<samlp:RequestAbstractType>`:

`<xacml-context:Request>` [Any Number]

An XACML Request Context. All XACML `<xacml:Policy>` and `<xacml:PolicySet>` instances potentially applicable to this Request that the requester is authorized to receive MUST be returned. The concept of “applicability” in the XACML context is defined in the XACML 3.0 Specification **[XACML3]**]. Any superset of applicable policies MAY be returned; for example, all policies having top-level Target elements that match the Request MAY be returned.

`<xacml:PolicySetIdReference>` [Any Number]

Identifies an XACML `<xacml:PolicySet>` instance to be returned.

`<xacml:PolicyIdReference>` [Any Number]

Identifies an XACML <xacml:Policy> instance to be returned.

Non-normative note: The <xacml-samlp:XACMLPolicyQuery> is not intended as a robust provisioning protocol. Users requiring such a protocol may consider using the OASIS Service Provisioning Markup Language (SPML). Note that the SAML-defined ID XML attribute is a required component of an instance of <samlp:RequestAbstractType> that the <xacml-samlp:XACMLPolicyQuery> extends and MAY be used to correlate the <xacml-samlp:XACMLPolicyQuery> with the corresponding XACMLPolicy Response.

6.6 Element <samlp:Response>: XACMLPolicy Response

A <samlp:Response> instance MAY contain an XACMLPolicy Assertion. An instance of such a <samlp:Response> element is called an XACMLPolicy Response in this Profile. An XACMLPolicy Response is shown in the following non-normative example:

```
<samlp:Response Version="2.0" ID="x9812368"
       IssueInstant="2006-05-31T13:20:00.000">
   <saml:Assertion Version="2.0" ID="x9812369"
       IssueInstant="2006-05-31T13:20:00.000">
      <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
      <saml:Statement
          xsi:type="xacml-saml:XACMLPolicyStatementType">
        <xacml:PolicySet PolicySetId="policyset:1" ... >
            ....
        </xacml:PolicySet>
      </saml:Statement>
   </saml:Assertion>
</samlp:Response>
```

An instance of a <samlp:Response> element that contains an XACMLPolicy Assertion is called an XACMLPolicy Response in this Profile. Such a Response MUST be used as the response to an <xacml-samlp:XACMLPolicyQuery>. It MAY be used to convey or store XACML policies for other purposes.

This Profile imposes the following requirements and restrictions on the <samlp:Response> element in addition to those specified in SAML 2.0 when used as an XACMLPolicy Response.

<saml:Issuer> [Optional]

The <saml:Issuer> element Identifies the originator of the contained XACML Policy, which MAY be the entity that generated the XACMLPolicy Response message. [SAML].

In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> element refer to the entity that signs the XACMLPolicy Response. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the XACMLPolicy Response.

<ds:Signature> [Optional]

The <ds:Signature> element is an optional element for holding "An XML Signature that authenticates the responder and provides message integrity" [SAML].

A <ds:Signature> instance MAY be used in an XACMLPolicy Response. In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Response. It is up to the relying party to determine whether it has an appropriate trust relationship with the authority that signs the XACMLPolicy Response.

A relying party SHOULD verify any signature included in the XACMLPolicy Response and SHOULD NOT use information derived from the XACMLPolicy Response unless the signature is verified successfully.

`<saml:Assertion>` [Any Number]

If the XACMLPolicy Response is issued in response to an `<xacml-samlp:XACMLPolicyQuery>`, then there MUST be exactly one instance of this element that contains an XACMLPolicy Assertion representing the response to the associated XACMLPolicy Query. If the XACMLPolicy Response is not issued in response to an `<xacml-samlp:XACMLPolicyQuery>`, it MAY contain one or more XACMLPolicy Assertions as well as other SAML or XACML Assertions.

`<saml:Status>` [Required]

If the XACMLPolicy Response is issued in response to an `<xacml-samlp:XACMLPolicyQuery>`, and if it is not possible to return all policies that satisfy the <xacml-samlp:XACMLPolicyQuery>, then a `<samlp:StatusCode>` value of `urn:oasis:names:tc:saml:2.0:status:TooManyResponses` MUST be returned in the `<samlp:Status>` element of the Response.

`InResponseTo` [Optional]

This optional XML attribute is “A reference to the identifier of the request to which the response corresponds.” When the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>, this XML attribute MUST contain the value of the `ID` XML attribute from the `<xacml-samlp:XACMLPolicyQuery>` to which this is a response. This allows the receiver to correlate the XACMLPolicy Response with the corresponding XACMLPolicy Query.

6.7 Policy references and Policy assertions

It may be noted that in relation to a policy assertion, there are three broad classes of policies to consider when resolving policy references: the top level policy in the policy assertion, the policies in the <xacml-samlp:ReferencedPolicies> element and policies external to the policy assertion, available to a PDP by other means.

How policy references are resolved across these three classes of policies depends on the particular case and problem for which the policy assertion is used. Therefore policy reference resolving is implementation defined with respect to policy assertions.

7 Advice

This Section describes how to include XACMLAuthzDecision Assertion and XACMLPolicy Assertion instances as advice in another SAML Assertion instance.

7.1 Element <saml:Advice>

A SAML Assertion MAY include a `<saml:Advice>` element containing “Additional information related to the assertion that assists processing in certain situations but which MAY be ignored [without affecting either the semantics or the validity of the assertion] by applications that do not understand the advice or do not wish to make use of it.” [SAML] An XACMLAuthzDecision Assertion or XACMLPolicy Assertion may be used in the Advice element as shown in the following non-normative example:

```
<saml:Advice>
    <saml:Assertion Version="2.0" ID="200606231640"
            IssueInstant="2006-05-31T13:20:00:000">
        <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
        <saml:Statement
            xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
            <xacml-context:Response>
                ....
            </xacml-context:Response>
            <xacml-context:Request>
                ....
            </xacml-context:Request>
        </saml:Statement>
    </saml:Assertion>
</saml:Advice>
```

8 Using an XACML Authorization Decision as an Authorization Token

This Section of the Profile describes how to use an XACMLAuthzDecision Statement as a security and privacy authorization token as part of a SOAP message exchange in a Web Services context. This token MAY be used by a client to convey an authorization decision from a trusted 3rd party to a service. A Web Service MAY use such a token to determine that the client is authorized to access information involved in the Web Services interaction.

In a Web Services context, an instance of an XACMLAuthzDecision Assertion MAY be used as an authorization token in the Web Services Security [WSS] `wsse:Security` Header of a SOAP message. When used in this way, the XACMLAuthzDecision Statement in the XACMLAuthzDecision Assertion MUST include the corresponding XACML Request Context. This allows the Web service to determine whether the `<xacml-context:Attribute>` instances in the Request correspond to the access that the client requires as part of the Web Service interaction. The XACMLAuthzDecision Assertion SHOULD be signed by a Policy Decision Point trusted by the Web Service.

A Web Service MAY use this token to determine that a trusted 3rd party has evaluated an XACML Request Context that is relevant to the invocation of the service, and has reported an authorization decision. The service SHOULD verify that the signature on the XACMLAuthzDecision Assertion is from a Policy Decision Point that the service trusts. The service SHOULD verify that the validity period of the XACMLAuthzDecision Assertion includes the time at which the Web Service interaction will access the information or resource to which the Request Context applies. The service SHOULD verify that the `<xacml-context:Attribute>` instances contained in the XACML `<xacml-context:Request>` element correctly describe the information or resource access that needs to be authorized as part of this Web Service interaction.

9 Conformance

Implementations of this Profile MAY implement certain subsets of the described functionality. Each implementation MUST clearly identify the subsets it implements using the following identifiers.

An implementation of this Profile is a conforming *SAML Attribute* implementation if the implementation conforms to Section 2 of this Profile. The following URI MUST be used as the identifier for this functionality:

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:attrs:all
```

An implementation of this Profile is a conforming *SOAP Attributes as XACMLAuthzDecisionQuery* implementation if the implementation conforms to Section 3.1 of this Profile. The following URI MUST be used as the identifier for this functionality:

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:authzQuery
```

An implementation of this Profile is a conforming *SOAP Attributes as SAML Attribute Assertion* implementation if the implementation conforms to Section 3.2 of this Profile. The following URI MUST be used as the identifier for this functionality:

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:attrAssertion
```

An implementation of this Profile is a conforming *XACML Authz Decision without Policies* implementation if the implementation conforms to all parts of Section 4 of this Profile excluding the `<xacml:Policy>`, `<xacml:PolicySet>`, and `<xacml-samlp:ReferencedPolicies>` elements and their sub-elements and the `CombinePolicies` XML attribute in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. XACML 3.0 implementations MUST support the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0 implementations MUST NOT support the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be used as the identifier for this functionality:

```
 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:noPolicies
```

An implementation of this Profile is a conforming *XACML Authz Decision with Policies* implementation if the implementation conforms to all parts of Section 4 of this Profile. XACML 3.0 implementations MUST support the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0 implementations MUST NOT support the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be used as the identifier for this functionality:

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:withPolicies
```

An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust with Policies* implementation if it conforms to section 5 in its entirety as described in the previous paragraqph. The following URI MUST be used as the identifier for this functionality.

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:withP
olicies
```

An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust without Policies implementation if it conforms to section 5, with the exceptions relating to policies and additioanl attribues noted above. The following URI MUST be used as the identifier for this functionality.*

urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:noPolicies

An implementation of this Profile is a conforming *XACML Policies* implementation if the implementation conforms to Section 6 of this Profile. The following URI MUST be used as the identifier for this functionality:

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:policies
```

An implementation of this Profile is a conforming *SAML Advice* implementation if the implementation conforms to Section 7 of this Profile. The following URI MUST be used as the identifier for this functionality:

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:adviceSAML
```

An implementation of this Profile is a conforming *XACML Authz Token* implementation if the implementation conforms to Section 8 of this Profile. The following URI MUST be used as the identifier for this functionality:

```
urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzToken
```

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged

Participants:

- Anne Anderson, Sun Microsystems
- Anthony Nadalin, IBM
- Bill Parducci,
- Carlisle Adams, University of Ottawa
- Daniel Engovatov, BEA
- Don Flinn,
- Ed Coyne
- Erik Rissanen
- Ernesto Damiani
- Frank Siebenlist
- Gerald Brose
- Hal Lockhart
- Haruyuki Kawabe
- James MacLean
- John Merrells
- Ken Yagen
- Konstantin Beznosov
- Michiharu Kudo
- Michael McIntosh
- Pierangela Samarati
- Pirasenna Velandai Thiyagarajan
- Polar Humenn
- Rebekah Metz
- Ron Jacobson
- Satoshi Hada
- Sekhar Vajjhala
- Seth Proctor
- Simon Godik
- Steve Anderson
- Steve Crocker
- Suresh Damodaran
- Tim Moses
- Von Welch
- Frederic Deleon
- Argyn Kuketayev

Appendix A. Revision History

Rev	Date	By whom	What
WD 1	12 April 2006	Anne Anderson	Create from SAML Profile errata document. <XACMLAuthzDecisionStatementType>: replace “ReturnResponse” with “ReturnContext” in description. Authorization Decisions: replaced “in the Response to an <XACMLAuthzDecisionStatement>” with “...<XACMLAuthzDecisionQuery>”. Create new types for SAML elements that will need to include XACML extensions. Create new elements for each extended type. Allow an XACMLAuthzDecisionQuery to include XACML policies for use in evaluating that query. Allow an XACMLAssertion to contain an XACMLAdvice element that in turn can contain an XACMLAssertion.
WD 2	23 June 2006	Anne Anderson	Changed name to “xacml-2.0-profile-saml2.0-v2-spec.... Removed specifications for all new elements except the XACMLAuthzDecisionQuery and XACMLPolicyQuery and all new types except for XACMLAuthzDecisionStatementType and XACMLPolicyStatementType and the two new Query types. Added descriptions of each standard SAML element in which XACML types might occur, and gave examples of use of xsi:type. Described use of the ID and InResponseTo attributes to correlate Queries and Responses.
WD 3	5 March 2007	Anne Anderson	-change boilerplate to conform to new OASIS template -Title: change to reflect that this profile applies to all versions of XACML -1.3 Added section on backwards compatibility -1.4 Removed notation section -1.5 Added namespaces section -2.6 Insert the “Conveying XACML Attributes in a SOAP Message” section from the WS-XACML profile -2.1.1 Clarify that <saml:Subject> is not translated into an XACML -id Attribute -3.5 and following,3.13: add syntax for passing additional Attributes in XACMLAuthzDecisionQuery from Admin Policy. 3.9 and following: add syntax for passing references policies. -4.4 XACMLPolicyQuery: clarify it returns all **potentially** applicable policies; remove Target element; change Choice lower bound from 0 to 1 and remove case where no elements included; add non-normative note to consider SPML for provisioning protocol -4.5 Response: Use valid ID values in example; add <samlp:Status> element saying to use SAML TooManyResponses StatusCode if unable to return all applicable policies -7 Insert the “XACML Authorization Token” section from the WS-XACML profile -Schemas: create versions specific to each XACML version -Protocol schema: remove XACMLPolicyQuery Target element, change Choice lower bound from 0 to 1 -Protocol schema: add Administrative Policy elements.
WD 4	15 June 2007	Anne Anderson	-throughout: used actual schema elements rather than invented names except when speaking about instances embedded in other instances (e.g. <saml:Attribute> rather than SAML Attribute, but SAML Attribute Response rather than <samlp:Response>). -throughout: changed SHALL to MUST -throughout: added namespace designators to schema items

Rev	Date	By whom	What
			and added additional namespace prefixes to list in Section 1.4 -Figure 1 updated the “Components and messages diagram to use same names as text -2.1.1 Clarified that implementations need not create actual <xacml-context:Attribute> instances so long as PDP can obtain corresponding values as if such instances existed. -2.1.1 Reworded description of NotBefore, NotOnOrAfter relationship to XACML date/time Attributes to be more clear -3.4,7,B.1 Inserted non-normative notes referring to open issues in relevant places -3.4,4.1 Clarified that the ReferencedPolicies element need not contain policies that receiver is not authorized to view -3.9 Clarified that Policy[Set]IdReference values must exactly match corresponding Policy[Set]Id values in the ReferencedPolicies element. -3.7 Changed “AttributeMatch” to “Match” to fit 3.0 schema -3.9,schemas:Fixed schema for ReferencedPolicies so it validates -3.4,4.1 Reworded AssignedAttributes and XACMLAuthzDecisionQuery Policy[Set] descriptions to clarify that the values must not be used except with the given Request “unless associated with the ... independently of the Request” -4.1,4.2 Add ReferencedPolicies element to XACMLPolicyStatementType -4.6 Reworded so to allow Response that is not issued in response to a specific Query -7 Added first draft of SAML Metadata -8 Added urn for SAML Metadata functionality
WD 5	19 July 2007	Anne Anderson	-Import XACML 1.0 schemas from local copies -Import XACML 2.0 schemas from http://docs.oasis-open.org/xacml/ directory -Import XACML 3.0 WD3 schema -Add OASIS copyright to all schemas -Made “Conveying XACML Attributes in a SOAP Message” a separate Section for easier reference in Conformance Section -Revised Conformance Section to refer to current document sections and to include previously omitted elements. -Made Introduction non-normative except for Namespaces and Normative References sections. -Made SAML Metadata section normative but RECOMMENDED
WD 6		Erik Rissanen	Added wording about deriving a policy issuer element from a saml assertion. Reworded requirements on the ReturnContext attribute. Changed some MAY/MUST statements. Fixed some TBDs. Changed order in which supplied policies are combined. Removed section about metadata. Fixed typos. Don't allow inheritance between supplied attributes in an

Rev	Date	By whom	What
			authz query. Relax the constraints on the <ReferencedPolicies> element.
WD 7	23 March 2009	Hal Lockhart	Improved some wording from previous changes. Added WS-Trust based decision request and response. Removed Metadata conformance clause.