XACML Profile for SAML 2.0

Working Draft 03, 27 July 2004

Document identifier:
oasis-xacml-profile-saml-wd-03

Location:
http://www.oasis-open.org/committees/xacml/

Editors:
Anne Anderson, Sun Microsystems (anne.anderson@sun.com)
Hal Lockhart, BEA (hlockhar@bea.com)

Abstract:

This specification defines a profile for the use of the OASIS Security Assertion Markup Language (SAML) Version 2.0 to carry XACML 2.0 policies, policy queries and responses, authorization decisions, and authorization decision queries and responses. It also describes the use of SAML 2.0 Attribute Assertions with XACML. Using XACML with SAML 2.0, XACML document instances can be protected using the SAML guidelines for use of digital signatures and can be transported using SAML bindings to transport mechanisms.

Status:

This version of the specification is a working draft within the OASIS XACML TC. As such, it is expected to change prior to adoption as an OASIS standard.

Committee members should send comments on this specification to the xacml@lists.oasis-open.org list. Others should subscribe to and send comments to the xacml-comment@lists.oasis-open.org list. To subscribe, send an email message to xacml-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

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 XACML TC web page (http://www.oasis-open.org/committees/xacml/).

For any errata page for this specification, please refer to the XACML SAML Profile section of the XACML TC web page (http://www.oasis-open.org/committees/xacml/).

Table of Contents

1 Introduction (non-normative)

The OASIS eXtensible Access Control Markup Language [XACML-SAMLP] 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 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 can serve as a standard format for exchanging information with these entities when combined with other standards.

One standard suitable for providing the assertion and protocol mechanisms needed by XACML is the OASIS Security Markup Assertion 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 and validate 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.

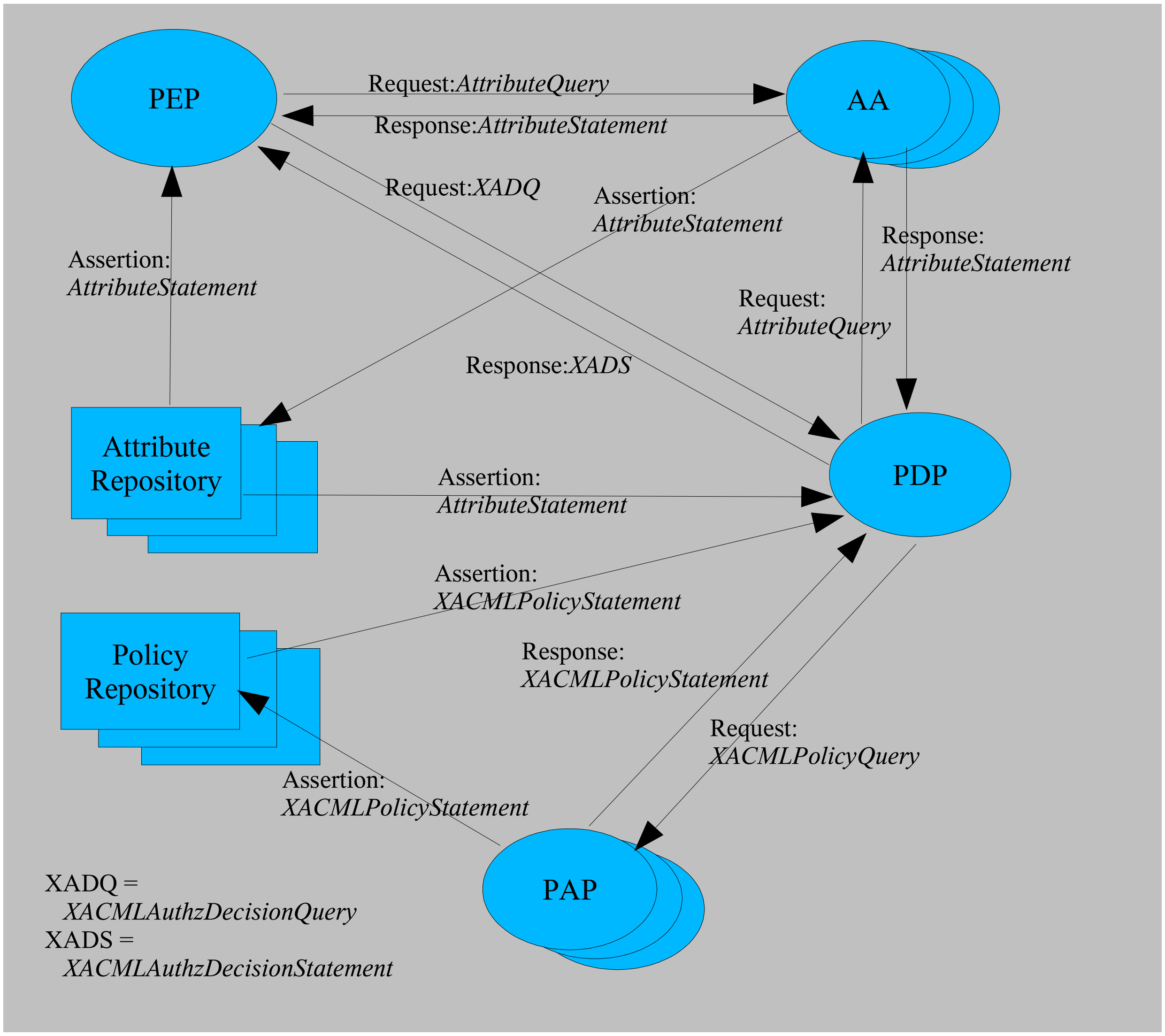
This profile defines how to use SAML 2.0 to protect, transport, and request XACML schema instances and other information needed by an XACML implementation.

There are 6 types of queries and statements used in this profile:

1. AttributeQuery – A standard SAML Request used for requesting one or more attributes from an Attribute Authority.
2. AttributeStatement – A standard SAML Statement that contains one or more attributes. This statement may be used in a SAML Response from an Attribute Authority, or it may be used in a SAML Assertion as a format for storing attributes in an Attribute Repository.
3. XACMLPolicyQuery – A SAML Request extension, defined in this profile. It is used for requesting one or more policies from a Policy Administration Point.
4. XACMLPolicyStatement – A SAML Statement extension, defined in this profile. It may be used in a SAML Response from a Policy Administration Point, or it may be used in a SAML Assertion as a format for storing policies in a Policy Repository.

5. XACMLAuthzDecisionQuery – A SAML Request extension, defined in this profile. It is used by a PEP to request an authorization decision from an XACML PDP.
6. XACMLAuthzDecisionStatement – A SAML Statement extension, defined in this profile. It may be used in a SAML Response from an XACML PDP. It might also be used in a SAML Assertion that is used as a credential, but this is not part of the currently defined XACML use model.

The following diagram illustrates the XACML use model and the messages that are used to communicate between the various components. Not all components will be used in every implementation.


PEP
AA
Request:*AttributeQuery*
Response:*AttributeStatement*
Request:*XADQ*
Assertion: *AttributeStatement*
Response: *AttributeStatement*
Assertion: *AttributeStatement*
Request: *AttributeQuery*
Response:*XADS*
Attribute Repository
PDP
Assertion: *AttributeStatement*
Assertion: *XACMLPolicyStatement*
Policy Repository
Response: *XACMLPolicyStatement*
Request: *XACMLPolicyQuery*
Assertion: *XACMLPolicyStatement*
PAP
XADQ = *XACMLAuthzDecisionQuery*
XADS = *XACMLAuthzDecisionStatement*

This specification describes all these query and statement schema elements, and describes how to use them. It also describes some other aspects of using SAML with XACML. This specification requires no changes or extensions to XACML, but does define extensions to SAML.

1.1 Notation

In order to improve readability, the examples in this profile assume use of the following XML

Internal Entity declarations:

```
^lt;!ENTITY saml "urn:oasis:names:tc:SAML:2.0:assertion"
^lt;!ENTITY samlp "urn:oasis:names:tc:SAML:2.0:protocol"
^lt;!ENTITY xacml "urn:oasis:names:tc:xacml:2.0:">
^lt;!ENTITY xacml-context "urn:oasis:names:tc:xacml:2.0:context">

^lt;!ENTITY xml "http://www.w3.org/2001/XMLSchema#">
^lt;!ENTITY subject-category
 "urn:oasis:names:tc:xacml:1.0:subject-category:">
^lt;!ENTITY subject "urn:oasis:names:tc:xacml:1.0:subject:">
^lt;!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:">
^lt;!ENTITY action "urn:oasis:names:tc:xacml:1.0:action:">
^lt;!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:">
```

For example, `&xml;#string` is equivalent to `http://www.w3.org/2001/XMLSchema#string`.

The namespace associated with the XACML schema [XACML-SAML] that extends the SAML Assertion schema is

```
xacml-saml="urn:oasis:names:tc:xacml:2.0:saml-profile:assertion"
```

The namespace associated with the XACML schema [XACML-SAMLP] that extends the SAML Protocol schema is

```
xacml-samlp="urn:oasis:names:tc:xacml:2.0:saml-profile:protocol"
```

1.2 Terminology

The key words *must, must not, required, shall, shall not, should, should not, recommended, may*, and *optional* in this document are to be interpreted as described in IETF RFC 2119 [RFC2119].

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 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 Request Context `<xacml-context:Attribute>` element. An XACML Attribute is associated with an identity by the Attribute's position within the XACML Request; for example, an XACML Attribute contained within the `<xacml-context:Resource>` element is an attribute of that resource.
- A SAML Attribute is name/value pair, with other optional information, specified using a SAML Assertion `<saml:Attribute>` element. A SAML Attribute is associated with a particular subject by its inclusion in a `<saml:SubjectStatement>` element. The SAML subject may correspond to an XACML subject, resource, action, or even environment.

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.

PAP – Policy Administration Point. An entity that issues authorization policies. Such policies may be expressed using a SAML Policy Assertion with the Policy Administration Point as the issuer.

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

PEP – Policy Enforcement Point. An 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 which subjects are permitted to access which resources using which actions under which conditions. XACML has two different schema elements used for policies: `<Policy>` and `<PolicySet>`. A `<PolicySet>` is a collection of other `<Policy>` and `<PolicySet>` elements. A `<Policy>` contains actual access control rules.

2 Attributes (normative)

The SAML assertion schema defines an Attribute Assertion. The SAML protocol schema defines an AttributeQuery used for requesting instances of Attribute Assertions, and a Response that contains the requested instances. Systems using XACML MAY use instances of these SAML elements transmit and store SAML Attributes. Systems using XACML MAY use the SAML AttributeQuery protocol to request instances of SAML Attributes. In order to be used in an XACML Request Context, the SAML Attribute SHALL be mapped to an XACML Attribute. This Section describes that mapping.

2.1 Mapping a SAML Attribute Assertion to XACML Attributes

A SAML Attribute Assertion is a `<saml:Assertion>` instance that contains one or more `<saml:AttributeStatement>` instances, each of which may contain one or more `<saml:Attribute>` instances.

In order to be used in an XACML Request Context, each SAML Attribute in the SAML Attribute Assertion SHALL comply with the *XACML Attribute Profile*, Identification `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`, of the *Profiles for the OASIS Security Assertion Markup Language* [SAML-PROFILE].

An `<xacml-context:Attribute>` SHALL be constructed from the corresponding `<saml:Attribute>` element in a SAML Attribute Assertion as follows.

- XACML `AttributeId` XML attribute

  The value of the `<saml:Attribute>` `Name` XML attribute SHALL be used.

- XACML `DataType` XML attribute

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

- XACML `Issuer` XML attribute

  The string value of the `<saml:Issuer>` element from the SAML Attribute Assertion SHALL be used.

- <xacml-context:AttributeValue>

  The `<saml:AttributeValue>` value SHALL be used as the value of the `<xacml-context:AttributeValue>` element.

Each `<saml:Attribute>` instance is mapped to a single `<xacml-context:Attribute>` element. Not all `<saml:Attribute>` instances in a SAML Attribute Assertion need to be mapped; the SAML Attribute instances to be mapped may be selected by a mechanism not specified here. The `Issuer` of the `<saml:Assertion>` element is used as the `Issuer` for each `<xacml-context:Attribute>` element that is created.

The `<xacml-context:Attribute>` created from the `<saml:Assertion>` SHALL be placed into the `<xacml-context:Resource>`, `<xacml-context:Subject>`, `<xacml-context:Action>`, or `<xacml-context:Environment>` element that corresponds to the entity that is the `<saml:Subject>` in the `SAML Attribute Assertion`. For example, if the SAML Attribute Assertion Subject contains a `<saml:NameIdentifier>` element, and the value of that `NameIdentifier` matches the value of the `<xacml-context:Attribute>` having an `AttributeId` of `&resource;resource-id`, then `<xacml-context:Attribute>` instances created from `<saml:Attribute>` instances in that SAML Attribute Assertion SHALL be placed into the `<xacml-context:Resource>` element. If the `<xacml-context:Attribute>` is placed into an `<xacml-context:Subject>` element, then the XACML `SubjectCategory` XML element SHALL also be consistent with the entity that is the Subject of the

`<saml:Assertion>`.

The entity performing the mapping SHALL ensure that the semantics defined by SAML for the elements in the `<saml:Assertion>` have been adhered to. The mapping entity need not perform these semantic checks itself, but it SHALL ensure that the checks have been done before any `<xacml:Attribute>` created from the `<saml: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:Assertion>` SHALL be valid with respect to the `<xacml:Request>` in which the SAML-derived `<xacml:Attribute>` is used. This means that the `NotBefore` and `NotOnOrAfter` XML attribute values SHALL be consistent with the `&environment;current-time`, `&environment;current-date`, and `&environment:current-dateTime` `<xacml:Attribute>` values associated with the `<xacml:Request>`.
- The entity doing the mapping SHALL ensure that the semantics defined by SAML for any `<saml:AudienceRestrictionCondition>` or `<saml:DoNotCacheCondition>` elements have been adhered to.
- If a `<ds:Signature>` element occurs in the `<saml:Assertion>`, then the entity performing the mapping SHALL ensure that the signature is valid and that the SAML `<Issuer>` element is consistent with any `<ds:X509IssuerName>` value in the signature. The guidelines regarding digital signatures in Section 5: *SAML and XML Signature Syntax and Processing* of the SAML core specification [SAML] SHALL be adhered to.

3 Authorization Decisions (normative)

SAML 2.0 defines a rudimentary AuthzDecisionQuery in the SAML Protocol Schema and a rudimentary AuthzDecisionStatement in the SAML Assertion Schema. A SAML AuthzDecisionQuery is unable to convey all the information that an XACML PDP is capable of accepting as part of its Request Context. Likewise, the SAML AuthzDecisionStatement is unable to convey all the information contained in an XACML Response Context.

In order to allow a PEP to use the SAML Request and Response syntax with full support for the XACML Request Context and Response Context syntax, this specification defines two SAML extensions:

- `<xacml-samlp:XACMLAuthzDecisionQuery>` is a SAML Query that extends the SAML Protocol Schema. It allows a PEP to submit an XACML Request Context in a SAML Request, along with other information.
- `<xacml-saml:XACMLAuthzDecisionStatement>` is a SAML Statement that extends the SAML Assertion schema. It allows an XACML PDP to return an XACML Response Context in the Response to an `<XACMLAuthzDecisionStatement>`, along with other information. It also allows an XACML Response Context to be stored or transmitted in the form of a SAML Assertion.

This Section defines these extensions. The extensions are contained in [XACML-SAML] and [XACML-SAMLP].

3.1 Element `<XACMLAuthzDecisionQuery>`

The `<XACMLAuthzDecisionQuery>` element MAY be used by a PEP to request an authorization decision from an XACML PDP. It allows a SAML Request to convey an XACML Request Context instance.

```
<xs:element name=”XACMLAuthzDecisionQuery”
            type=”XACMLAuthzDecisionQueryType”/>
<xs:complexType name=”XACMLAuthzDecisionQueryType”>
    <xs:complexContent>
        <xs:extension base=”samlp:RequestAbstractType”>
            <xs:sequence>
                <xs:element ref=”xacml-context:Request”/>
            </xs:sequence>
            <xs:attribute name=”InputContextOnly”
                          type=”boolean”
                          use=”required”/>
            <xs:attribute name=”ReturnContext”
                          type=”boolean”
                          use=”required”/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The `<XACMLAuthzDecisionQuery>` element is of XACMLAuthzDecisionQueryType complex type. This element is an alternative to the SAML-defined `<samlp:AuthzDecisionQuery>` that allows a PEP to use the full capabilities of an XACML PDP.

The `<XACMLAuthzDecisionQuery>` element contains the following attributes and elements:

`InputContextOnly` [Required]

This attribute governs the sources of information that the PDP is allowed to use in making its authorization decision. If this attribute is “True”, then the authorization decision SHALL be made solely on the basis of information contained in the `<XACMLAuthzDecisionQuery>`; no external attributes MAY be used. If this attribute is

"False", then the authorization decision MAY be made on the basis of external attributes not contained in the `<XACMLAuthzDecisionQuery>`.

`ReturnContext` [Required]

This attribute allows the PEP to request that an `<xacml-context:Request>` element be included in the `<XACMLAuthzDecisionStatement>` resulting from the request. It also governs the contents of that `<xacml-context:Request>` element.

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. The PDP MAY include additional XACML Attributes in this `<xacml-context:Request>` element, such as external attributes obtained by the PDP and used in making the authorization decision, or other attributes known by the PDP that may be useful to the PEP in making subsequent `<XACMLAuthzDecisionQuery>` requests.

If this element is "False", then the PDP SHALL NOT include the `<xacml-context:Request>` element in the `<XACMLAuthzDecisionStatement>` element of the `<XACMLResponse>` .

`<xacml-context:Request>` [Required]

An XACML Request Context.

3.2 Element <XACMLAuthzDecisionStatement>

The `<XACMLAuthzDecisionStatement>` MAY be used by an XACML PDP to return a SAML Response containing an XACML Response Context to a PEP in response to an `<XACMLAuthzDecisionQuery>`. It may also be used in a SAML Assertion as a format for storage of an authorization decision in a repository.

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

The `<XACMLAuthzDecisionStatement>` element is of XACMLAuthzDecisionStatementType complex type. This element is an alternative to the SAML-defined `<samlp:AuthzDecisionStatement>` that allows a SAML Assertion to contain the full content of the response from an XACML PDP.

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

`<xacml-context:Response>` [Required]

The XACML Response Context created by the XACML PDP in response to the `<XACMLAuthzDecisionQuery>`.

`<xacml-context:Request>` [Optional]

An `<xacml-context:Request>` containing XACML Attributes returned by the XACML PDP in response to the `<XACMLAuthzDecisionQuery>`. This element SHALL be included if the `ReturnResponse` XML attribute in the `<XACMLAuthzDecisionQuery>`

is “True”. This element SHALL NOT be included if the `ReturnResponse` XML attribute in the `<XACMLAuthzDecisionQuery>` is “False”.

See the description of the `ReturnContext` attribute in Section 3.1: *Element `<XACMLAuthzDecisionQuery>`* for a description of the XACML `<Attribute>` values that SHALL be returned in this element.

4 Policies (normative)

XACML defines two policy schema elements: `<Policy>` and `<PolicySet>`. SAML does not define any Protocol or Assertion schemas for policies. This Section defines new SAML extensions for `<XACMLPolicyQuery>` and `<XACMLPolicyStatement>` elements. Instances of these new elements can be used to request, transmit, and store XACML `<Policy>` and `<PolicySet>` instances. The new extensions are contained in [XACML-SAML] and [XACML-SAMLP].

4.1 Element `<XACMLPolicyQuery>`

The <XACMLPolicyQuery> element is used by an PDP to request one or more XACML Policy or PolicySet instances from an on-line Policy Administration Point as part of a SAML Request.

```
<xs:element name=”XACMLPolicyQuery”
            type=”XACMLPolicyQueryType”/>
<xs:complexType name=”XACMLPolicyQueryType”>
    <complexContent>
        <xs:extension base=”samlp:RequestAbstractType”>
            <xs:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xs:element ref=”xacml-context:Request”/>
                <xs:element ref=”xacml:PolicySetIdReference”/>
                <xs:element ref=”xacml:PolicyIdReference”/>
            </xs:choice>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The `<XACMLPolicyQuery>` element is of XACMLPolicyQueryType complex type.

The `<XACMLPolicyQuery>` element contains one or more of the following elements:

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

Supplies an XACML Request Context. All XACML Policy and PolicySet instances applicable to this Request SHALL be returned. The concept of “applicability” in the XACML context is defined in the XACML 2.0 Specification [XACML-SAMLP].

`<xacml:PolicySetIdReference>` [Any Number]

Identifies an XACML `<PolicySet>` to be returned.

`<xacml:PolicyIdReference>` [Any Number]

Identifies an XACML `<Policy>` to be returned.

4.2 Element <XACMLPolicyStatement>

The `<XACMLPolicyStatement>` is used by a Policy Administration Point to return one or more XACML `<Policy>` or `<PolicySet>` instances in a SAML Response to an `<XACMLPolicyQuery>` SAML Request. The `<XACMLPolicyStatement>` may also be used in a SAML Assertion as a format for storing the `<XACMLPolicyStatement>` in a repository.

```
<xs:element name=”XACMLPolicyStatement”
            type=”xacml-saml:XACMLPolicyStatementType”/>
<xs:complexType name=”XACMLPolicyStatementType”>
    <xs:complexContent>
        <xs:extension base=”saml:StatementAbstractType”>
          <xs:choice minOccurs=”0” maxOccurs=unbounded”>
            <xs:element ref=”xacml:Policy”/>
            <xs:element ref=”xacmlPolicySet”/>
          </xs:choice>
```

```
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
```

The `<XACMLPolicyStatement>` element is of XACMLPolicyStatementType complex type.

The `<XACMLPolicyStatement>` element contains the following elements. If the `<XACMLPolicyStatement>` is issued in response to an `<XACMLPolicyQuery>`, and there are no `<xacml:Policy>` or `<xacml:PolicySet>` instances that meet the specifications of the associated `<XACMLPolicyQuery>`, then there SHALL be no elements in the `<XACMLPolicyStatement>`.

`<xacml:Policy>` [Any Number]

> An `<xacml:Policy>` instance that meets the specifications of the associated `<XACMLPolicyQuery>`, if any.

`<xacml:PolicySet>` [Any Number]

> An `<xacml:PolicySet>` instance that meets the specifications of the associated `<XACMLPolicyQuery>`, if any.

5 References

5.1 Normative References

[RFC2119] 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, J. Kemp, E. Maler, eds., *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, http://www.oasis-open.org/committees/security*

[SAML-PROFILE] F. Hirsch, et al., eds., *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*, *http://www.oasis-open.org/committees/security*

[XACML] T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML) Versions 1.0, 1.1, and 2.0*,http://www.oasis-open.org/committees/xacml/

[XACML-SAML] A. Anderson, ed., *xacml-profile-saml-schema-assertion.xsd*, http://www.oasis-open.org/committees/xacml/

[XACML-SAMLP] A. Anderson, ed., *xacml-profile-saml-schema-profile.xsd*, http://www.oasis-open.org/committees/xacml/

5.2 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.

A. Acknowledgments

The editors would like to acknowledge the contributions of the OASIS XXX Technical Committee, whose voting members at the time of publication were:

- Frank Siebenlist, Argonne National Laboratory
- Daniel Engovatov, BEA Systems, Inc.
- Hal Lockhart, BEA Systems, Inc.
- Ronald Jacobson, Computer Associates
- Tim Moses, Entrust
- Simon Godik, GlueCode Software
- Bill Parducci, GlueCode Software
- Michiharu Kudo, IBM
- Michael McIntosh, IBM
- Anthony Nadalin, IBM
- Steve Anderson, OpenNetwork
- Anne Anderson, Sun Microsystems
- Seth Proctor, Sun Microsystems
- Polar Humenn, Syracuse University

B. Revision History

Rev	Date	By Whom	What
01	20 Mar 2003	Anne Anderson	Initial Working Draft.
02	25 Feb 2004	Anne Anderson	Added proposed extension schemas and normative text. Makes use of sstc-maler-w28a-attribute-draft-02, which has not been approved by SSTC. Based on SAML 2.0 Draft 07 core and schemas.
03	27 July 2004	Anne Anderson	Changed Bill and Simon affiliation to GlueCode Software. Changed Attribute description to match SAML Profiles section on XACML. Changed AuthorizationDecision to AuthzDecision to be consistent with SAML 2.0. Made Queries extend SAML RequestAbstractType, consistent with 2.0. Changed Policy Authority to Policy Administration Point, consistent with XACML specification.

C. Notices