



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/>).

33 **Table of Contents**

34 1 Introduction (non-normative).....3

35 1.1 Notation.....4

36 1.2 Terminology.....5

37 2 Attributes (normative).....7

38 2.1 Mapping a SAML Attribute Assertion to XACML Attributes.....7

39 3 Authorization Decisions (normative).....9

40 3.1 Element <XACMLAuthzDecisionQuery>.....9

41 3.2 Element <XACMLAuthzDecisionStatement>.....10

42 4 Policies (normative).....12

43 4.1 Element <XACMLPolicyQuery>.....12

44 4.2 Element <XACMLPolicyStatement>.....12

45 5 References.....14

46 5.1 Normative References.....14

47 5.2 Non-normative References.....14

1 Introduction (non-normative)

48

49

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

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

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

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

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

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

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

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

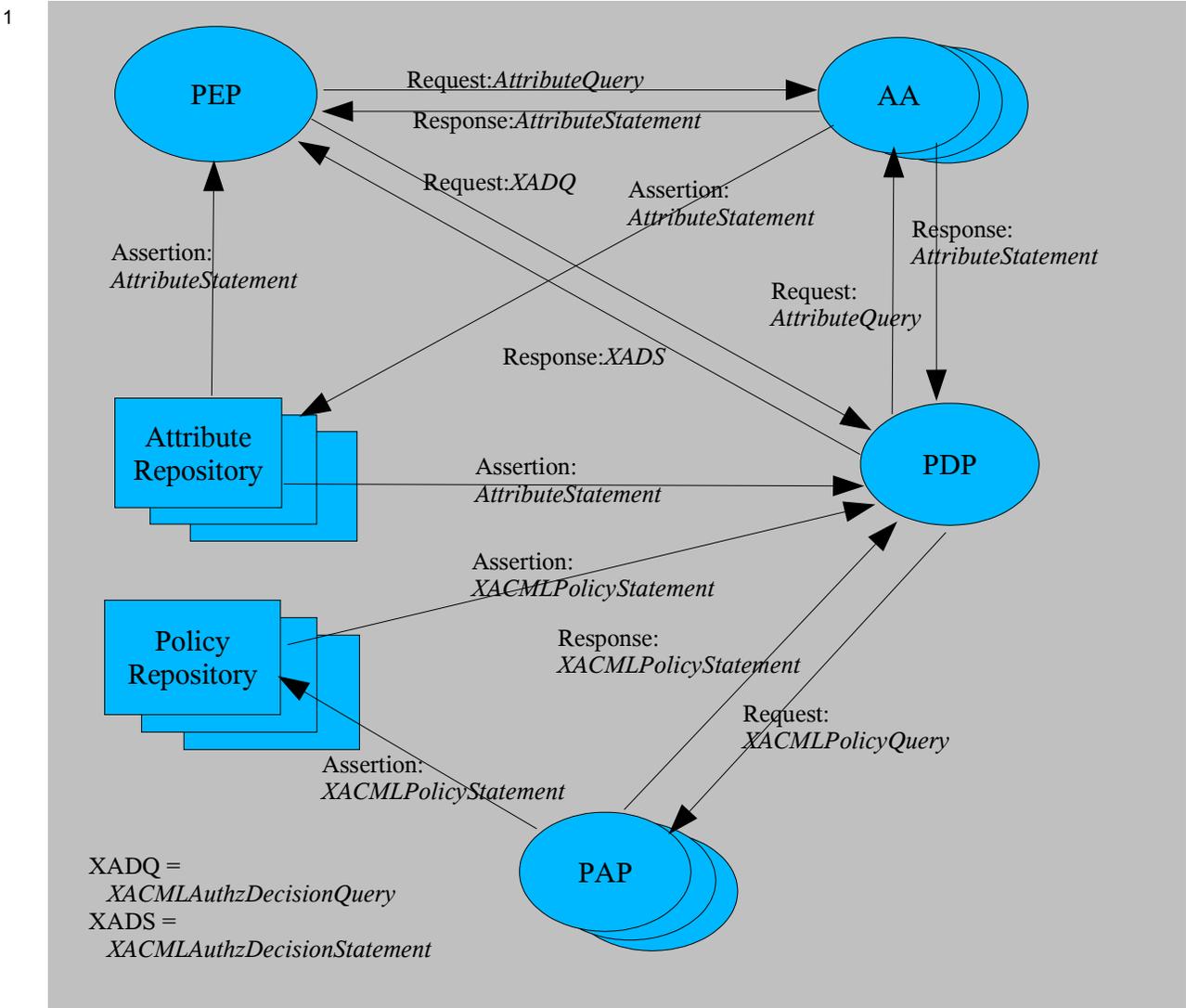
- 85 1. *AttributeQuery* – A standard SAML Request used for requesting one or more attributes from an
86 *Attribute Authority*.
- 87 2. *AttributeStatement* – A standard SAML Statement that contains one or more attributes. This
88 statement may be used in a SAML Response from an *Attribute Authority*,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

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

101 The following diagram illustrates the XACML use model and the messages that are used to

102 communicate between the various components. Not all components will be used in every

103 implementation.



105 This specification describes all these query and statement schema elements, and describes how

106 to use them. It also describes some other aspects of using SAML with XACML. This specification

107 requires no changes or extensions to XACML, but does define extensions to SAML.

108 **1.1 Notation**

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

110 Internal Entity declarations:

```
111 ^lt;!ENTITY saml "urn:oasis:names:tc:SAML:2.0:assertion"
112 ^lt;!ENTITY samlp "urn:oasis:names:tc:SAML:2.0:protocol"
113 ^lt;!ENTITY xacml "urn:oasis:names:tc:xacml:2.0:">
114 ^lt;!ENTITY xacml-context "urn:oasis:names:tc:xacml:2.0:context">
115
116 ^lt;!ENTITY xml "http://www.w3.org/2001/XMLSchema#">
117 ^lt;!ENTITY subject-category
118   "urn:oasis:names:tc:xacml:1.0:subject-category:">
119 ^lt;!ENTITY subject "urn:oasis:names:tc:xacml:1.0:subject:">
120 ^lt;!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:">
121 ^lt;!ENTITY action "urn:oasis:names:tc:xacml:1.0:action:">
122 ^lt;!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:">
```

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

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

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

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

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

131 1.2 Terminology

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

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

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

139 • An XACML Attribute is a typed name/value pair, with other optional information, specified using
140 an XACML Request Context `<xacml-context:Attribute>` element. An XACML Attribute
141 is associated with an identity by the Attribute's position within the XACML Request; for
142 example, an XACML Attribute contained within the `<xacml-context:Resource>` element is
143 an attribute of that resource.

144 • A SAML Attribute is name/value pair, with other optional information, specified using a SAML
145 Assertion `<saml:Attribute>` element. A SAML Attribute is associated with a particular
146 subject by its inclusion in a `<saml:SubjectStatement>` element. The SAML subject may
147 correspond to an XACML subject, resource, action, or even environment.

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

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

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

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

158 **policy** – A set of rules indicating which subjects are permitted to access which resources using

159 which actions under which conditions. XACML has two different schema elements used for
160 policies: <Policy> and <PolicySet>. A <PolicySet> is a collection of other <Policy> and
161 <PolicySet> elements. A <Policy> contains actual access control rules.

2 Attributes (normative)

161

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

2.1 Mapping a SAML Attribute Assertion to XACML Attributes

169

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

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

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

- 179 • XACML `AttributeId` XML attribute

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

- 181 • XACML `DataType` XML attribute

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

- 185 • XACML `Issuer` XML attribute

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

- 188 • `<xacml-context:AttributeValue>`

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

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

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

207 <saml:Assertion>.

208 The entity performing the mapping SHALL ensure that the semantics defined by SAML for the
209 elements in the <saml:Assertion> have been adhered to. The mapping entity need not
210 perform these semantic checks itself, but it SHALL ensure that the checks have been done before
211 any <xacml:Attribute> created from the <saml:Assertion> is used by an XACML PDP.
212 These semantic checks include, but are not limited to, the following.

- 213 • Any NotBefore and NotOnOrAfter XML attributes in the <saml:Assertion> SHALL be
214 valid with respect to the <xacml:Request> in which the SAML-derived
215 <xacml:Attribute> is used. This means that the NotBefore and NotOnOrAfter XML
216 attribute values SHALL be consistent with the &environment;current-time,
217 &environment;current-date, and &environment:current-dateTime
218 <xacml:Attribute> values associated with the <xacml:Request>.
- 219 • The entity doing the mapping SHALL ensure that the semantics defined by SAML for any
220 <saml:AudienceRestrictionCondition> or <saml:DoNotCacheCondition>
221 elements have been adhered to.
- 222 • If a <ds:Signature> element occurs in the <saml:Assertion>, then the entity performing
223 the mapping SHALL ensure that the signature is valid and that the SAML <Issuer> element is
224 consistent with any <ds:X509IssuerName> value in the signature. The guidelines regarding
225 digital signatures in Section 5: *SAML and XML Signature Syntax and Processing* of the SAML
226 core specification [SAML] SHALL be adhered to.

227

3 Authorization Decisions (normative)

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

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

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

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

3.1 Element `<XACMLAuthzDecisionQuery>`

247 The `<XACMLAuthzDecisionQuery>` element MAY be used by a PEP to request an
248 authorization decision from an XACML PDP. It allows a SAML Request to convey an XACML
249 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>
```

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

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

254 `InputContextOnly` [Required]

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

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

261 ReturnContext [Required]

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

265 If this attribute is "True", then the PDP SHALL include the <xacml-context:Request>
266 element in the <XACMLAuthzDecisionStatement> element in the
267 <XACMLResponse>. This <xacml-context:Request> element SHALL include all
268 those XACML Attributes supplied by the PEP in the <XACMLAuthzDecisionQuery> that
269 were used in making the authorization decision. The PDP MAY include additional
270 XACML Attributes in this <xacml-context:Request> element, such as external
271 attributes obtained by the PDP and used in making the authorization decision, or other
272 attributes known by the PDP that may be useful to the PEP in making subsequent
273 <XACMLAuthzDecisionQuery> requests.

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

277 <xacml-context:Request> [Required]

278 An XACML Request Context.

279 3.2 Element <XACMLAuthzDecisionStatement>

280 The <XACMLAuthzDecisionStatement> MAY be used by an XACML PDP to return a SAML
281 Response containing an XACML Response Context to a PEP in response to an
282 <XACMLAuthzDecisionQuery>. It may also be used in a SAML Assertion as a format for
283 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>
```

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

288 The <XACMLAuthzDecisionStatement> element contains the following elements:

289 <xacml-context:Response> [Required]

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

292 <xacml-context:Request> [Optional]

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

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

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

301 4 Policies (normative)

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

308 4.1 Element <XACMLPolicyQuery>

309 The <XACMLPolicyQuery> element is used by an PDP to request one or more XACML Policy or
310 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>
  </complexContent>
</xs:complexType>
```

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

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

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

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

317 <xacml:PolicySetIdReference> [Any Number]

318 Identifies an XACML <PolicySet> to be returned.

319 <xacml:PolicyIdReference> [Any Number]

320 Identifies an XACML <Policy> to be returned.

321 4.2 Element <XACMLPolicyStatement>

322 The <XACMLPolicyStatement> is used by a Policy Administration Point to return one or more
323 XACML <Policy> or <PolicySet> instances in a SAML Response to an
324 <XACMLPolicyQuery> SAML Request. The <XACMLPolicyStatement> may also be used in
325 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">
  <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>
  </complexContent>
</xs:complexType>
```

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

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

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

332 <xacml:Policy> [Any Number]

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

335 <xacml:PolicySet> [Any Number]

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

338

5 References

339

5.1 Normative References

340

341 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
342 IETF RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

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

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

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

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

353 **[XACML-SAMPLP]** A. Anderson, ed., *xacml-profile-saml-schema-profile.xsd*,
354 <http://www.oasis-open.org/committees/xacml/>

355

5.2 Non-normative References

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

359 **A. Acknowledgments**

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

- 362 • Frank Siebenlist, Argonne National Laboratory
- 363 • Daniel Engovatov, BEA Systems, Inc.
- 364 • Hal Lockhart, BEA Systems, Inc.
- 365 • Ronald Jacobson, Computer Associates
- 366 • Tim Moses, Entrust
- 367 • Simon Godik, GlueCode Software
- 368 • Bill Parducci, GlueCode Software
- 369 • Michiharu Kudo, IBM
- 370 • Michael McIntosh, IBM
- 371 • Anthony Nadalin, IBM
- 372 • Steve Anderson, OpenNetwork
- 373 • Anne Anderson, Sun Microsystems
- 374 • Seth Proctor, Sun Microsystems
- 375 • Polar Humenn, Syracuse University

376

B. Revision History

377

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.

378

379

C. Notices

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

389 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
390 applications, or other proprietary rights which may cover technology that may be required to
391 implement this specification. Please address the information to the OASIS Executive Director.

392 **Copyright © OASIS Open 2004. All Rights Reserved.**

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

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

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