



2 XACML Profile for SAML 2.0

3 Working Draft 04, 19 August 2004

4 Document identifier:

5 oasis-xacml-profile-saml-wd-04

6 Location:

7 <http://www.oasis-open.org/committees/xacml/>

8 Editors:

9 Anne Anderson, Sun Microsystems (anne.anderson@sun.com)

10 Hal Lockhart, BEA (hlockhar@bea.com)

11 Abstract:

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

19 Status:

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

22 Committee members should send comments on this specification to the
23 xacml@lists.oasis-open.org list. Others should subscribe to and send comments to the
24 xacml-comment@lists.oasis-open.org list. To subscribe, send an email message to [xacml-](mailto:xacml-comment-request@lists.oasis-open.org)
25 comment-request@lists.oasis-open.org with the word "subscribe" as the body of the
26 message.

27 For information on whether any patents have been disclosed that may be essential to
28 implementing this specification, and any offers of patent licensing terms, please refer to
29 the Intellectual Property Rights section of the XACML TC web page ([http://www.oasis-](http://www.oasis-open.org/committees/xacml/)
30 [open.org/committees/xacml/](http://www.oasis-open.org/committees/xacml/)).

31 For any errata page for this specification, please refer to the XACML SAML Profile section
32 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 Element <saml:Assertion> (normative).....14

46 5.1 Element <saml:Issuer>.....14

47 5.2 Element <ds:Signature>.....14

48 5.3 Element <saml:Subject>.....14

49 5.4 Element <saml:Conditions>.....15

50 6 Element <samlp:Request> (normative).....16

51 6.1 Element <saml:Issuer>.....16

52 6.2 Element <ds:Signature>.....16

53 7 Element <samlp:Response> (normative).....17

54 7.1 Element <samlp:Issuer>.....17

55 7.2 Element <ds:Signature>.....17

56 7.3 Element <samlp:StatusCode>.....17

57 8 References.....19

58 8.1 Normative References.....19

59 8.2 Non-normative References.....19

1 Introduction (non-normative)

60

61
62 The OASIS eXtensible Access Control Markup Language [XACML-SAML] is a powerful,
63 standard language that specifies schemas for authorization policies and for authorization decision
64 requests and responses. It also specifies how to evaluate policies against requests to compute a
65 response. A brief overview of XACML is available in [XACMLIntro].

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

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

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

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

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

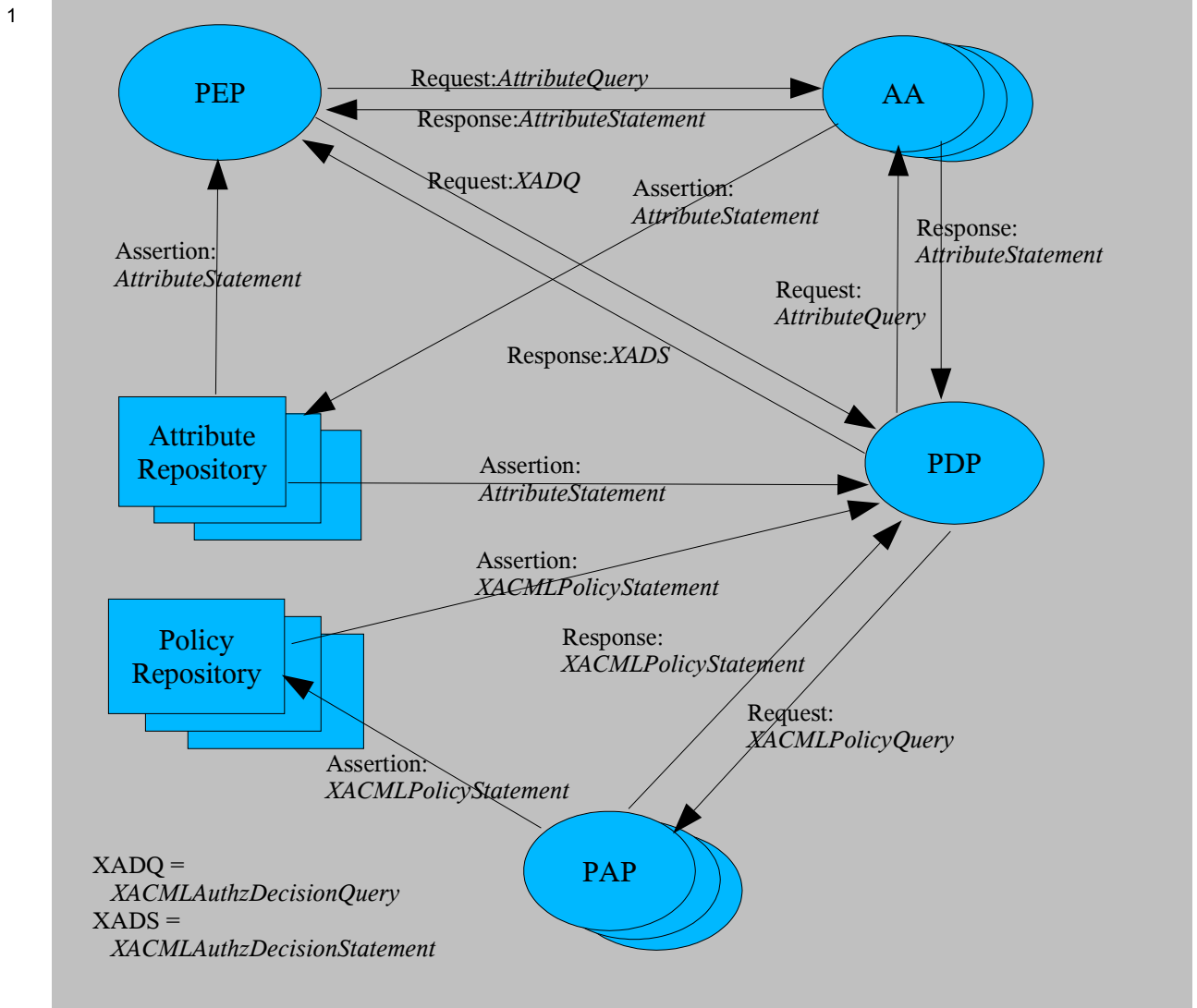
94 This profile defines how to use SAML 2.0 to protect, transport, and request XACML schema
95 instances and other information needed by an XACML implementation.

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

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

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

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



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

120 1.1 Notation

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

122 Internal Entity declarations:

```
123 ^lt;!ENTITY saml "urn:oasis:names:tc:SAML:2.0:assertion"
124 ^lt;!ENTITY samlp "urn:oasis:names:tc:SAML:2.0:protocol"
125 ^lt;!ENTITY xacml "urn:oasis:names:tc:xacml:2.0:">
126 ^lt;!ENTITY xacml-context "urn:oasis:names:tc:xacml:2.0:context">
127
128 ^lt;!ENTITY xml "http://www.w3.org/2001/XMLSchema#">
129 ^lt;!ENTITY subject-id
130     "urn:oasis:names:tc:xacml:1.0:subject:subject-id">
131 ^lt;!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:">
132 ^lt;!ENTITY resource-id
133     "urn:oasis:names:tc:xacml:1.0:resource:resource-id">
134 ^lt;!ENTITY action-id "urn:oasis:names:tc:xacml:1.0:action:action-id">
135 ^lt;!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:">
136 ^lt;!ENTITY current-dateTime
137     "urn:oasis:names:tc:xacml:1.0:environment:current-dateTime">
```

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

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

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

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

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

146 1.2 Terminology

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

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

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

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

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

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

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

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

170 **PEP** – Policy Enforcement Point. An entity that enforces access control for one or more
171 resources. When a resource access is attempted, a PEP sends an access request describing the

172 attempted access to a PDP. The PDP returns an access decision that the PEP then enforces.
173 **policy** – A set of rules indicating which subjects are permitted to access which resources using
174 which actions under which conditions. XACML has two different schema elements used for
175 policies: <Policy> and <PolicySet>. A <PolicySet> is a collection of other <Policy> and
176 <PolicySet> elements. A <Policy> contains actual access control rules.

177

2 Attributes (normative)

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

2.1 Mapping a SAML Attribute Assertion to XACML Attributes

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

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

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

- 195 • XACML `AttributeId` XML attribute

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

- 197 • XACML `DataType` XML attribute

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

- 201 • XACML `Issuer` XML attribute

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

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

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

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

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

223 <saml:Assertion>.

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

- 229 • Any NotBefore and NotOnOrAfter XML attributes in the <saml:Assertion> SHALL be
230 valid with respect to the <xacml:Request> in which the SAML-derived
231 <xacml:Attribute> is used. This means that the NotBefore and NotOnOrAfter XML
232 attribute values SHALL be consistent with the &environment;current-time,
233 &environment;current-date, and &environment:current-dateTime
234 <xacml:Attribute> values associated with the <xacml:Request>.
- 235 • The entity doing the mapping SHALL ensure that the semantics defined by SAML for any
236 <saml:AudienceRestrictionCondition> or <saml:DoNotCacheCondition>
237 elements have been adhered to.
- 238 • If a <ds:Signature> element occurs in the <saml:Assertion>, then the entity performing
239 the mapping SHALL ensure that the signature is valid and that the SAML <Issuer> element is
240 consistent with any <ds:X509IssuerName> value in the signature. The guidelines regarding
241 digital signatures in Section 5: *SAML and XML Signature Syntax and Processing* of the SAML
242 core specification [SAML] SHALL be adhered to.

243

3 Authorization Decisions (normative)

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

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

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

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

3.1 Element `<XACMLAuthzDecisionQuery>`

262 The `<XACMLAuthzDecisionQuery>` element MAY be used by a PEP to request an
263 authorization decision from an XACML PDP. It allows a SAML Request to convey an XACML
264 Request Context instance.
265

```
<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>
```

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

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

270 `InputContextOnly` [Required]

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

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

277 ReturnContext [Required]

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

281 If this attribute is "True", then the PDP SHALL include the <xacml-context:Request>
282 element in the <XACMLAuthzDecisionStatement> element in the
283 <XACMLResponse>. This <xacml-context:Request> element SHALL include all
284 those XACML Attributes supplied by the PEP in the <XACMLAuthzDecisionQuery> that
285 were used in making the authorization decision. The PDP MAY include additional
286 XACML Attributes in this <xacml-context:Request> element, such as external
287 attributes obtained by the PDP and used in making the authorization decision, or other
288 attributes known by the PDP that may be useful to the PEP in making subsequent
289 <XACMLAuthzDecisionQuery> requests.

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

293 <xacml-context:Request> [Required]

294 An XACML Request Context.

295 3.2 Element <XACMLAuthzDecisionStatement>

296 The <XACMLAuthzDecisionStatement> MAY be used by an XACML PDP to return a SAML
297 Response containing an XACML Response Context to a PEP in response to an
298 <XACMLAuthzDecisionQuery>. It may also be used in a SAML Assertion as a format for
299 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>
```

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

304 The <XACMLAuthzDecisionStatement> element contains the following elements:

305 <xacml-context:Response> [Required]

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

308 <xacml-context:Request> [Optional]

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

312 is "True". This element SHALL NOT be included if the `ReturnResponse` XML attribute in
313 the `<XACMLAuthzDecisionQuery>` is "False".
314 See the description of the `ReturnContext` attribute in Section 3.1: *Element*
315 `<XACMLAuthzDecisionQuery>` for a description of the XACML `<Attribute>` values
316 that SHALL be returned in this element.

317

4 Policies (normative)

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

324 4.1 Element <XACMLPolicyQuery>

325 The <XACMLPolicyQuery> element is used by an PDP to request one or more XACML Policy or
326 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>
```

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

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

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

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

333 <xacml:PolicySetIdReference> [Any Number]

334 Identifies an XACML <PolicySet> to be returned.

335 <xacml:PolicyIdReference> [Any Number]

336 Identifies an XACML <Policy> to be returned.

337 4.2 Element <XACMLPolicyStatement>

338 The <XACMLPolicyStatement> is used by a Policy Administration Point to return one or more
339 XACML <Policy> or <PolicySet> instances in a SAML Response to an
340 <XACMLPolicyQuery> SAML Request. The <XACMLPolicyStatement> may also be used in
341 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>
```

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

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

348 <xacml:Policy> [Any Number]

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

351 <xacml:PolicySet> [Any Number]

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

354 5 Element <saml:Assertion> (normative)

355 An <XACMLAuthzDecisionStatement>, <XACMLPolicyStatement>, or SAML standard
356 <saml:AttributeStatement> is encapsulated in a <saml:Assertion>, which may be
357 signed.

358 Most components of a <saml:Assertion> are fully specified in the SAML 2.0 specification
359 [SAML]. The following elements and attributes are further specified here for use with the SAML
360 statement types defined and used in this Profile.

361 Except as specified here, this Profile imposes no requirements or restrictions on information in the
362 <saml:Assertion> element.

363 5.1 Element <saml:Issuer>

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

366 In order to support 3rd party digital signatures, this Profile does NOT require that the identity
367 provided in the <saml:Issuer> element be consistent with the identity of the signer. It is up to
368 the relying party to have an appropriate trust relationship with the authority that signs the
369 <saml:Assertion>.

370 When a <saml:AttributeAssertion> is used to construct an XACML **attribute**, the string
371 value of the <saml:Issuer> element SHALL be used as the value of the XACML Issuer XML
372 attribute, so the SAML value SHOULD be specified with this in mind. See *Section 2.1: Mapping a*
373 *SAML Attribute Assertion to XACML Attributes* for more information.

374 5.2 Element <ds:Signature>

375 The <ds:Signature> element is an optional element for holding “An XML Signature that
376 authenticates the assertion, as described in Section 5.”

377 A <ds:Signature> element MAY be used in an assertion used with an XACML Statement. In
378 order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
379 in the <saml:Issuer> element be consistent with the identity of the signer. It is up to the relying
380 party to have an appropriate trust relationship with the authority that signs the
381 <saml:Assertion>.

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

384 5.3 Element <saml:Subject>

385 The <saml:Subject> element is an optional element for holding “The subject of the statement
386 (s) in the assertion” [SAML].

387 The <saml:Subject> element SHALL NOT be included in an assertion that contains an
388 <XACMLAuthzDecision> or <XACMLPolicy>.

389 In a <saml:AttributeAssertion> that is to be mapped to an XACML **attribute**, the
390 <saml:Subject> element SHALL contain the identity of the entity to which the attribute and its
391 value are bound. For an XACML <Subject> **attribute**, this identity SHOULD be consistent with
392 the value of any XACML &subject-id; **attribute** that occurs in the same <Subject> element.
393 For an XACML <Resource> **attribute**, this identity SHOULD be consistent with the value of any
394 XACML &resource-id; **attribute** that occurs in the same <Resource> element. For an
395 XACML <Action> **attribute**, this identity SHOULD be consistent with the value of any XACML
396 &action-id; **attribute** that occurs in the same <Action> element. For an XACML
397 <Environment> **attribute**, this identity SHOULD be consistent with the value of any XACML

398 **attribute** that occurs in the same <Environment> element and provides an environment identity.

399 **5.4 Element <saml:Conditions>**

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

402 The <saml:Conditions> element SHOULD contain `NotBefore` and `NotOnOrAfter` attributes
403 to specify the limits on the validity of the assertion. If these attributes are present, the relying party
404 SHOULD ensure that information derived from the assertion is used by a **PDP** for evaluating
405 policies only when the value of the request **context** `¤t-dateTime`; **resource attribute**
406 in the is contained within the assertion's specified validity period.

407 **6 Element <samlp:Request> (normative)**

408 An <XACMLAuthzDecisionQuery> or <XACMLPolicyQuery> is usually encapsulated in a
409 <samlp:Request> element, which may be signed.

410 Most components of a <samlp:Request> are fully specified in the SAML 2.0 specification
411 [SAML]. The following elements and attributes are further specified here for use with the SAML
412 query types defined and used in this Profile. Except as specified here, this Profile imposes no
413 requirements or restrictions on information in the <samlp:Request> element.

414 **6.1 Element <saml:Issuer>**

415 See *Section 5.1: Element <saml:Issuer>*.

416 **6.2 Element <ds:Signature>**

417 See *Section 5.2: Element <ds:Signature>*.

418 7 Element <samlp:Response> (normative)

419 An <XACMLAuthzDecisionStatement> or <XACMLPolicyStatement> is usually
420 encapsulated in a <samlp:Request> element, which may be signed.

421 Most components of a <samlp:Response> are fully specified in the SAML 2.0 specification
422 [SAML]. The following elements and attributes are further specified here for use with the SAML
423 statement types defined and used in this Profile. Except as specified here, this Profile imposes no
424 requirements or restrictions on information in the <samlp:Response> element.

425 7.1 Element <samlp:Issuer>

426 See Section 5.1: Element <saml:Issuer>.

427 7.2 Element <ds:Signature>

428 See Section 5.2: Element <ds:Signature>.

429 7.3 Element <samlp:StatusCode>

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

432 7.3.1 Response to <XACMLAuthzDecisionQuery>

433 In the response to an <XACMLAuthzDecisionQuery> request, the <samlp:StatusCode>
434 VALUE attribute SHALL depend on the <xacml:StatusCode> element of the authorization
435 decision <xacml:Status> element as follows:

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

437 This value for the <samlp:StatusCode> VALUE SHALL be used if and only if the
438 <xacml:StatusCode> value is urn:oasis:names:tc:xacml:1.0:status:ok.

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

440 This value for the <samlp:StatusCode> VALUE SHALL be used when the
441 <xacml:StatusCode> value is
442 urn:oasis:names:tc:xacml:1.0:status:missing-attribute or the when the
443 <xacml:StatusCode> value is
444 urn:oasis:names:tc:xacml:1.0:status:syntax-error due to a syntax error in
445 the <xacml:Request>.

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

447 This value for the <samlp:StatusCode> VALUE SHALL be used when the
448 <xacml:StatusCode> value is
449 urn:oasis:names:tc:xacml:1.0:status:syntax-error due to a syntax error in
450 an <xacml:Policy> or <xacml:PolicySet>. Note that not all syntax errors in
451 policies will be detected in conjunction with the processing of a particular query, so not all
452 policy syntax errors will be reported this way.

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

454 This value for the <samlp:StatusCode> VALUE SHALL be used only when the SAML
455 interface at the PDP does not support the version of the SAML request message used in
456 the query.

457 **7.3.2 Response to <XACMLPolicyQuery>**

458 In the response to an <XACMLPolicyQuery> request, the <samlp:StatusCode> value SHALL
459 be as specified in the SAML specification.

460

8 References

461

8.1 Normative References

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

[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-SAMPLP]

A. Anderson, ed., *xacml-profile-saml-schema-profile.xsd*, <http://www.oasis-open.org/committees/xacml/>

477

8.2 Non-normative References

478

479

480

[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.

481 **A. Acknowledgments**

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

- 484 • Frank Siebenlist, Argonne National Laboratory
- 485 • Daniel Engovatov, BEA Systems, Inc.
- 486 • Hal Lockhart, BEA Systems, Inc.
- 487 • Ronald Jacobson, Computer Associates
- 488 • Tim Moses, Entrust
- 489 • Simon Godik, GlueCode Software
- 490 • Bill Parducci, GlueCode Software
- 491 • Michiharu Kudo, IBM
- 492 • Michael McIntosh, IBM
- 493 • Anthony Nadalin, IBM
- 494 • Steve Anderson, OpenNetwork
- 495 • Anne Anderson, Sun Microsystems
- 496 • Seth Proctor, Sun Microsystems
- 497 • Polar Humenn, Syracuse University

498

B. Revision History

499

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.
04	04 Aug 2004	Anne Anderson	Added new Sections describing how to populate elements in saml:Assertion (Issuer, Signature, Subject, Conditions), Request (Issuer, Signature), and Response (Issuer, Signature, StatusCode).

500

501

C. Notices

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

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

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

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

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

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