



2 SAML 2.0 profile of XACML

3 Committee Draft 01, 16 September 2004

4 Document identifier:

5 access_control-xacml-2.0-saml_profile-spec-cd-01

6 Location:

7 http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml_profile-spec-cd-01.pdf

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.

16 Status:

17 This version of the specification is an approved Committee Draft within the OASIS Access
18 Control TC.

19 Access Control TC members should send comments on this specification to the
20 xacml@lists.oasis-open.org list. Others may use the following link and complete the
21 comment form: http://oasis-open.org/committees/comments/form.php?wg_abbrev=xacml.

22 For information on whether any patents have been disclosed that may be essential to
23 implementing this specification, and any offers of patent licensing terms, please refer to
24 the Intellectual Property Rights section of the Access Control TC web page
25 (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml).

26 For any errata page for this specification, please refer to the Access Control TC web page
27 (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml).

28 Copyright © OASIS Open 2004 All Rights Reserved.

Table of Contents

29		
30	1 Introduction (non-normative).....	3
31	1.1 Notation.....	4
32	1.2 Terminology.....	5
33	2 Attributes (normative).....	7
34	2.1 Mapping a SAML Attribute Assertion to XACML Attributes.....	7
35	3 Authorization Decisions (normative).....	9
36	3.1 Element <XACMLAuthzDecisionQuery>.....	9
37	3.2 Element <XACMLAuthzDecisionStatement>.....	10
38	4 Policies (normative).....	12
39	4.1 Element <XACMLPolicyQuery>.....	12
40	4.2 Element <XACMLPolicyStatement>.....	12
41	5 Element <saml:Assertion> (normative).....	14
42	5.1 Element <saml:Issuer>.....	14
43	5.2 Element <ds:Signature>.....	14
44	5.3 Element <saml:Subject>.....	14
45	5.4 Element <saml:Conditions>.....	15
46	6 Element <samlp:Request> (normative).....	16
47	6.1 Element <saml:Issuer>.....	16
48	6.2 Element <ds:Signature>.....	16
49	7 Element <samlp:Response> (normative).....	17
50	7.1 Element <samlp:Issuer>.....	17
51	7.2 Element <ds:Signature>.....	17
52	7.3 Element <samlp:StatusCode>.....	17
53	8 References.....	19
54	8.1 Normative References.....	19
55	8.2 Non-normative References.....	19

1 Introduction (non-normative)

56

57

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

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

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

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

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

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

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

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

93 1. AttributeQuery – A standard SAML Request used for requesting one or more attributes from an
94 Attribute Authority.

95 2. AttributeStatement – A standard SAML Statement that contains one or more attributes. This
96 statement may be used in a SAML Response from an Attribute Authority, or it may be used in a
97 SAML Assertion as a format for storing attributes in an Attribute Repository.

98 3. XACMLPolicyQuery – A SAML Request extension, defined in this profile. It is used for
99 requesting one or more policies from a Policy Administration Point.

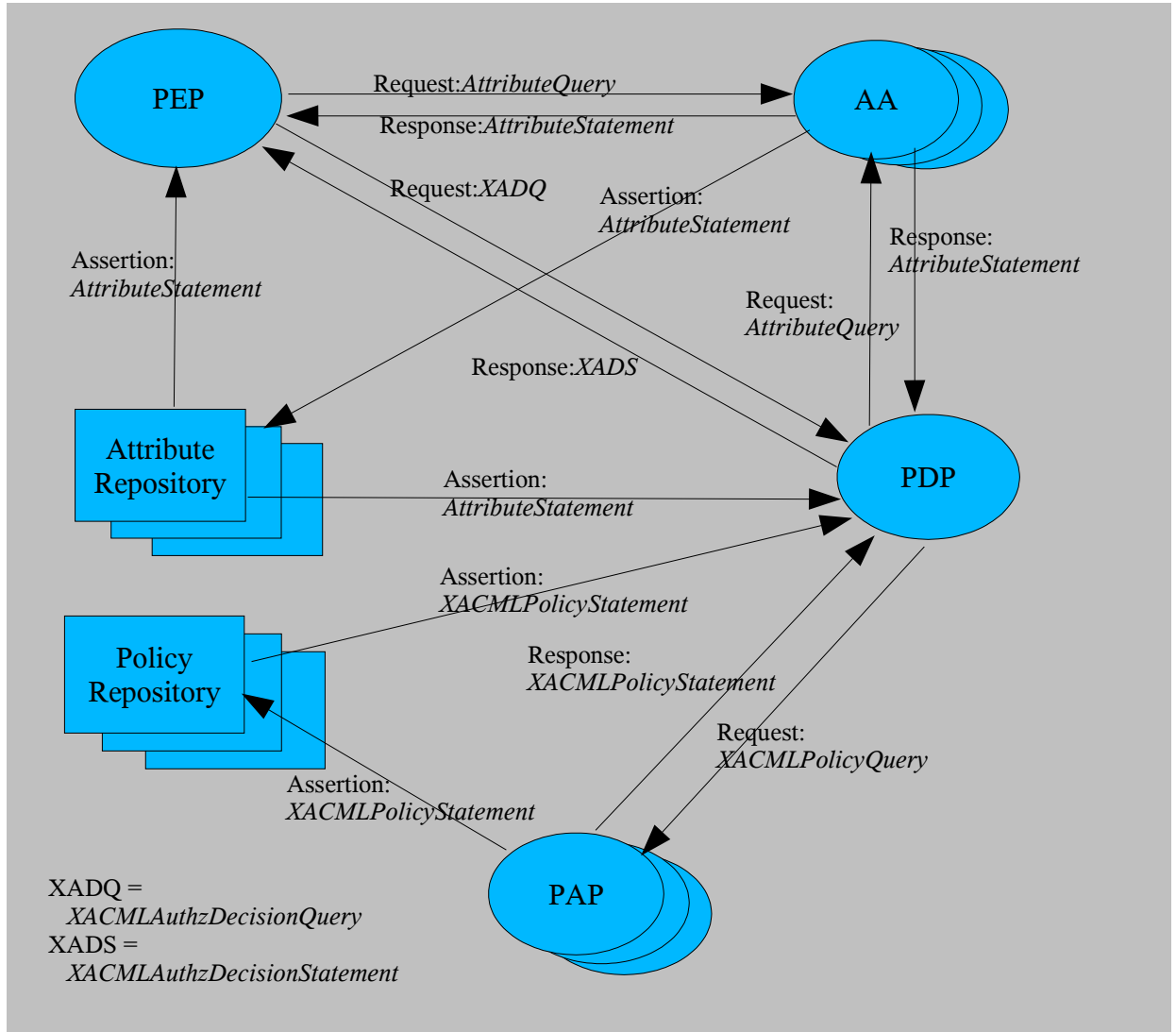
100 4. XACMLPolicyStatement – A SAML Statement extension, defined in this profile. It may be used
101 in a SAML Response from a Policy Administration Point, or it may be used in a SAML
102 Assertion as a format for storing policies in a Policy Repository.

103 5. XACMLAuthzDecisionQuery – A SAML Request extension, defined in this profile. It is used by
104 a PEP to request an authorization decision from an XACML PDP.

105 6. XACMLAuthzDecisionStatement – A SAML Statement extension, defined in this profile. It may
106 be used in a SAML Response from an XACML PDP. It might also be used in a SAML
107 Assertion that is used as a credential, but this is not part of the currently defined XACML use
108 model.

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

1



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

116 1.1 Notation

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

118 Internal Entity declarations:

```
119 ^lt;!ENTITY saml "urn:oasis:names:tc:SAML:2.0:assertion"
120 ^lt;!ENTITY samlp "urn:oasis:names:tc:SAML:2.0:protocol"
121 ^lt;!ENTITY xacml "urn:oasis:names:tc:xacml:2.0:"
122 ^lt;!ENTITY xacml-context
123     "urn:oasis:names:tc:xacml:2.0:context:schema:cd-01"
124 ^lt;!ENTITY xml "http://www.w3.org/2001/XMLSchema#"
125 ^lt;!ENTITY subject-id
126     "urn:oasis:names:tc:xacml:1.0:subject:subject-id"
127 ^lt;!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:"
128 ^lt;!ENTITY resource-id
129     "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
130 ^lt;!ENTITY action-id "urn:oasis:names:tc:xacml:1.0:action:action-id"
131 ^lt;!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:"
132 ^lt;!ENTITY current-dateTime
133     "urn:oasis:names:tc:xacml:1.0:environment:current-dateTime"
```

134 For example, “&xml;#string” is equivalent to
135 <http://www.w3.org/2001/XMLSchema#string>.

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

```
138     xacml-saml="urn:oasis:names:tc:xacml:2.0:saml:assertion:schema:cd-01"
```

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

```
141     xacml-samlp="urn:oasis:names:tc:xacml:2.0:saml:protocol:schema:cd-01"
```

142 1.2 Terminology

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

146 *“they MUST only be used where it is actually required for interoperation or to limit*
147 *behavior which has potential for causing harm (e.g., limiting retransmissions)”*

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

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

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

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

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

166 **attribute** – In this Profile, the term “attribute”, when not capitalized, refers to a generic attribute or
167 characteristic unless it is preceded by the term “XML”. An “XML attribute” is a syntactic

168 component in XML that occurs inside the opening tag of an XML element.

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

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

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

176 **policy** – A set of rules indicating which subjects are permitted to access which resources using
177 which actions under which conditions. XACML has two different schema elements used for
178 policies: <Policy> and <PolicySet>. A <PolicySet> is a collection of other <Policy> and
179 <PolicySet> elements. A <Policy> contains actual access control rules.

2 Attributes (normative)

180

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

2.1 Mapping a SAML Attribute Assertion to XACML Attributes

188

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

192 In order to be used in an XACML Request Context, each SAML Attribute in the SAML Attribute
193 Assertion SHALL comply with *XACML Attribute Profile* (Section 8.5), namespace
194 `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`, in the *Profiles for the*
195 *OASIS Security Assertion Markup Language* [SAML-PROFILE].

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

- 198 • XACML `AttributeId` XML attribute

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

- 200 • XACML `DataType` XML attribute

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

- 204 • XACML `Issuer` XML attribute

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

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

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

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

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

226 <saml:Assertion>.

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

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

246

3 Authorization Decisions (normative)

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

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

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

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

3.1 Element `<XACMLAuthzDecisionQuery>`

265 The `<XACMLAuthzDecisionQuery>` element MAY be used by a PEP to request an
266 authorization decision from an XACML PDP. It allows a SAML Request to convey an XACML
267 Request Context instance.
268

```
<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="optional"
                    default="false"/>
      <xs:attribute name="ReturnContext"
                    type="boolean"
                    use="optional"
                    default="false"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

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

274 `InputContextOnly` [Default "false"]

275 This XML attribute governs the sources of information that the PDP is allowed to use in

276 making its authorization decision. If this XML attribute is “true”, then the authorization
277 decision SHALL be made solely on the basis of information contained in the
278 <XACMLAuthzDecisionQuery>; no external attributes MAY be used. If this XML
279 attribute is “false”, then the authorization decision MAY be made on the basis of external
280 attributes not contained in the <XACMLAuthzDecisionQuery>.

281 ReturnContext [Default “false”]

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

285 If this XML attribute is “true”, then the PDP SHALL include the <xacml-
286 context:Request> element in the <XACMLAuthzDecisionStatement> element in
287 the <XACMLResponse>. This <xacml-context:Request> element SHALL include all
288 those attributes supplied by the PEP in the <XACMLAuthzDecisionQuery> that were
289 used in making the authorization decision. The PDP MAY include additional attributes in
290 this <xacml-context:Request> element, such as external attributes obtained by the
291 PDP and used in making the authorization decision, or other attributes known by the PDP
292 that may be useful to the PEP in making subsequent <XACMLAuthzDecisionQuery>
293 requests.

294 If this XML attribute is “false”, then the PDP SHALL NOT include the <xacml-
295 context:Request> element in the <XACMLAuthzDecisionStatement> element of
296 the <XACMLResponse> .

297 <xacml-context:Request> [Required]

298 An XACML Request Context.

299 3.2 Element <XACMLAuthzDecisionStatement>

300 The <XACMLAuthzDecisionStatement> MAY be used by an XACML PDP to return a SAML
301 Response containing an XACML Response Context to a PEP in response to an
302 <XACMLAuthzDecisionQuery>. It may also be used in a SAML Assertion as a format for
303 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>
```

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

308 The <XACMLAuthzDecisionStatement> element contains the following elements:

309 <xacml-context:Response> [Required]

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

312 <xacml-context:Request> [Optional]

313 An <xacml-context:Request> containing XACML Attributes returned by the XACML
314 PDP in response to the <XACMLAuthzDecisionQuery>. This element SHALL be
315 included if the ReturnResponse XML attribute in the <XACMLAuthzDecisionQuery>
316 is "true". This element SHALL NOT be included if the ReturnResponse XML attribute in
317 the <XACMLAuthzDecisionQuery> is "false".

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

321

4 Policies (normative)

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

328

4.1 Element <XACMLPolicyQuery>

329 The <XACMLPolicyQuery> element is used by a PDP to request one or more XACML Policy or
330 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:Target"/>
        <xs:element ref="xacml:PolicySetIdReference"/>
        <xs:element ref="xacml:PolicyIdReference"/>
      </xs:choice>
    </xs:extension>
  </complexContent>
</xs:complexType>
```

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

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

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

334 Supplies an XACML Request Context. All XACML Policy and PolicySet instances
335 applicable to this Request SHALL be returned. The concept of "applicability" in the
336 XACML context is defined in the XACML 2.0 Specification [XACML].

337 <xacml:Target> [Any Number]

338 Supplies an XACML <Target> element. All XACML Policy and PolicySet instances
339 applicable to this <Target> SHALL be returned.

340 <xacml:PolicySetIdReference> [Any Number]

341 Identifies an XACML <PolicySet> to be returned.

342 <xacml:PolicyIdReference> [Any Number]

343 Identifies an XACML <Policy> to be returned.

4.2 Element <XACMLPolicyStatement>

344
345 The <XACMLPolicyStatement> is used by a Policy Administration Point to return one or more
346 XACML <Policy> or <PolicySet> instances in a SAML Response to an
347 <XACMLPolicyQuery> SAML Request. The <XACMLPolicyStatement> may also be used in
348 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>
```

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

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

355 <xacml:Policy> [Any Number]

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

358 <xacml:PolicySet> [Any Number]

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

361 5 Element <saml:Assertion> (normative)

362 An <XACMLAuthzDecisionStatement>, <XACMLPolicyStatement>, or SAML standard
363 <saml:AttributeStatement> SHALL be encapsulated in a <saml:Assertion>, which MAY
364 be signed.

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

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

370 5.1 Element <saml:Issuer>

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

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

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

381 5.2 Element <ds:Signature>

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

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

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

391 5.3 Element <saml:Subject>

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

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

396 In a <saml:AttributeAssertion> that is to be mapped to an XACML Attribute, the
397 <saml:Subject> element SHALL contain the identity of the entity to which the attribute and its
398 value are bound. For an XACML <Subject> Attribute, this identity SHOULD be consistent with
399 the value of any XACML &subject-id; Attribute that occurs in the same <Subject> element.
400 For an XACML <Resource> Attribute, this identity SHOULD be consistent with the value of any
401 XACML &resource-id; Attribute that occurs in the same <Resource> element. For an
402 XACML <Action> Attribute, this identity SHOULD be consistent with the value of any XACML
403 &action-id; Attribute that occurs in the same <Action> element. For an XACML
404 <Environment> Attribute, this identity SHOULD be consistent with the value of any XACML

405 Attribute that occurs in the same `<Environment>` element and provides an environment identity.

406 **5.4 Element `<saml:Conditions>`**

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

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

414 **6 Element <samlp:Request> (normative)**

415 An <XACMLAuthzDecisionQuery> or <XACMLPolicyQuery> SHALL be encapsulated in a
416 <samlp:Request> element, which MAY be signed.

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

421 **6.1 Element <saml:Issuer>**

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

423 **6.2 Element <ds:Signature>**

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

425 7 Element <samlp:Response> (normative)

426 An <XACMLAuthzDecisionStatement> or <XACMLPolicyStatement> SHALL be
427 encapsulated in a <samlp:Response> element, which MAY be signed.

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

432 7.1 Element <samlp:Issuer>

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

434 7.2 Element <ds:Signature>

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

436 7.3 Element <samlp:StatusCode>

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

439 7.3.1 Response to <XACMLAuthzDecisionQuery>

440 In the response to an <XACMLAuthzDecisionQuery> request, the <samlp:StatusCode>
441 Value XML attribute SHALL depend on the <xacml:StatusCode> element of the authorization
442 decision <xacml:Status> element as follows:

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

444 This value for the <samlp:StatusCode> Value XML attribute SHALL be used if and
445 only if the <xacml:StatusCode> value is
446 urn:oasis:names:tc:xacml:1.0:status:ok.

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

448 This value for the <samlp:StatusCode> Value XML attribute SHALL be used when the
449 <xacml:StatusCode> value is
450 urn:oasis:names:tc:xacml:1.0:status:missing-attribute or the when the
451 <xacml:StatusCode> value is
452 urn:oasis:names:tc:xacml:1.0:status:syntax-error due to a syntax error in
453 the <xacml:Request>.

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

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

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

462 This value for the <samlp:StatusCode> Value XML attribute SHALL be used only when
463 the SAML interface at the PDP does not support the version of the SAML request
464 message used in the query.

465 **7.3.2 Response to <XACMLPolicyQuery>**

466 In the response to an <XACMLPolicyQuery> request, the <samlp:StatusCode> Value XML
467 attribute SHALL be as specified in the SAML specification.

468

8 References

469

8.1 Normative References

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

[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, et al., eds., *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, Committee Draft 01c, 18 September 2004, http://www.oasis-open.org/committees/documents.php?wg_abbrev=security.

[SAML-PROFILE]

J. Hughes, et al., eds., *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*, Committee Draft 01b, 14 September 2004, http://www.oasis-open.org/committees/documents.php?wg_abbrev=security.

[XACML]

S. Godik, T. Moses, eds., *OASIS eXtensible Access Control Markup Language (XACML) Version 2.0*, Committee Draft 01, 16 September 2004, http://docs.oasis-open.org/xacml/access_control-xacml-2.0-core-spec-cd-01.pdf.

[XACML-SAML]

A. Anderson, ed., *access_control-xacml-2.0-saml-assertion-schema-cd-01.xsd*, http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml-assertion-schema-cd-01.xsd.

[XACML-SAML P]

A. Anderson, ed., *access_control-xacml-2.0-saml-protocol-schema-cd-01.xsd*, http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml-protocol-schema-cd-01.xsd.

491

8.2 Non-normative References

492

493

494

[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.

495 **A. Acknowledgments**

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

- 498 • Frank Siebenlist, Argonne National Laboratory
- 499 • Daniel Engovatov, BEA Systems, Inc.
- 500 • Hal Lockhart, BEA Systems, Inc.
- 501 • Rebekah Metz, Booz Allen Hamilton
- 502 • Ronald Jacobson, Computer Associates
- 503 • Tim Moses, Entrust
- 504 • Simon Godik, GlueCode Software
- 505 • Bill Parducci, GlueCode Software
- 506 • Michiharu Kudo, IBM
- 507 • Michael McIntosh, IBM
- 508 • Anthony Nadalin, IBM
- 509 • Steve Anderson, OpenNetwork
- 510 • Anne Anderson, Sun Microsystems
- 511 • Seth Proctor, Sun Microsystems
- 512 • Polar Humenn, Syracuse University
- 513 • Edward Coyne, Veterans Health Administration

514

B. Revision History

515

Rev	Date	By Whom	What
CD-01	16 Sept 2004	XACML committee	Committee Draft

516

C. Notices

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

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

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

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

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

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