



SAML 2.0 Profile of XACML 2.1

Working Draft 1, 12 April 2006

Document identifier:

xacml-2.1-profile-saml2.0-spec-wd-1

OASIS identifier:

[OASIS document number]

Location:

Persistent: <http://docs.oasis-open.org/xacml/2.1/xacml-2.1-profile-saml2.0.zip>

This Version:

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#CURRENT

Previous Version:

<http://www.oasis-open.org/committees/download.php/15447/xacml-2.0-saml-errata-wd.zip>

Technical Committee:

OASIS eXtensible Access Control Markup Language (XACML) TC

Chair(s):

Hal Lockhart

Bill Parducci

Editor:

Anne Anderson

Subject / Keywords:

XACML, SAML, access control, security assertions, policy

OASIS Conceptual Model Topic Area:

Security

Related Work:

This specification replaces or supercedes:

- SAML 2.0 profile of XACML 2.0

This specification is related to:

- SAML 2.0 OASIS Standard
- XACML 2.0 OASIS Standard

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

34 decisions, and authorization decision queries and responses. It also describes the use of SAML
35 2.0 Attribute Assertions with XACML. This version defines a more complete extension of SAML
36 for use in XACML systems than in the previous version.

37 **Status:**

38 This document was last revised or approved by the eXtensible Access Control Markup Language
39 (XACML) TC on the above date. The level of approval is also listed above. Check the current
40 location noted above for possible later revisions of this document. This document is updated
41 periodically on no particular schedule.

42 Technical Committee members should send comments on this specification to the
43 Technical Committee's email list. Others should send comments to the Technical
44 Committee by using the "Send A Comment" button on the Technical Committee's
45 web page at www.oasis-open.org/committees/xacml.

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

50 The non-normative errata page for this specification is located at [www.oasis-](http://www.oasis-open.org/committees/xacml)
51 [open.org/committees/xacml](http://www.oasis-open.org/committees/xacml).

52

Notices

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

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

65 Copyright © OASIS Open 2006. All Rights Reserved.

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

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

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

81

82

83

84

85 Table of Contents

86	1 Introduction.....	5
87	1.1 Notation.....	7
88	1.2 Terminology.....	8
89	1.3 Normative References.....	9
90	1.4 Non-normative References.....	9
91	2 Attributes.....	10
92	2.1 Element <saml:Attribute>.....	10
93	2.2 Element <saml:AttributeStatement>.....	11
94	2.3 Element <saml:Assertion>.....	12
95	2.4 Element <xacml-saml:XACMLAssertion>.....	13
96	2.5 Element <samlp:AttributeQuery>.....	13
97	2.6 Element <samlp:Response>.....	13
98	3 Authorization Decisions.....	14
99	3.1 Element <XACMLAuthzDecisionStatement>.....	14
100	3.2 Element <XACMLAssertion>.....	15
101	3.3 Element <XACMLAuthzDecisionQuery>.....	16
102	3.4 Element <XACMLResponse>.....	18
103	4 Policies.....	20
104	4.1 Element <XACMLPolicyStatement>.....	20
105	4.2 Element <XACMLAssertion>.....	21
106	4.3 Element <XACMLPolicyQuery>.....	21
107	4.4 Element <XACMLResponse>.....	22
108	5 Advice.....	23
109	5.1 Element <XACMLAdvice>.....	23
110	5.2 Element <XACMLAssertion>.....	23
111		

1 Introduction

112

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

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

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

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

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

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

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

145 This Profile defines how to use SAML 2.0 to protect, store, transport, request, and respond with XACML
146 schema instances and other information needed by an XACML implementation.

147 This Profile starts by describing how to use SAML Attributes in an XACML system. It describes the use
148 of the following elements:

- 149 1. `SAML Attribute` – A standard SAML element that MAY be used in an XACML system for
150 storing and transmitting attribute values. The `SAML Attribute` must be transformed into an
151 `XACML Attribute` before it can be used in an XACML Request Context.
- 152 2. `SAML AttributeStatement` – A standard SAML element that SHALL be used to hold SAML
153 `Attribute` instances in an XACML system.
- 154 3. `SAML Assertion` – A standard SAML element that MAY be used to hold SAML
155 `AttributeStatement` instances in an XACML system, either in an Attribute Repository or in a

156 SAML Response to a SAML AttributeQuery. The SAML Assertion contains information
157 that is required in order to transform a SAML Attribute into an XACML Attribute. A SAML
158 Attribute SHALL be contained in either a SAML Assertion instance or in an
159 XACMLAssertion instance when used in an XACML system.

160 4. XACMLAssertion – A new SAML extension element that is an alternative to the SAML
161 Assertion element and allows inclusion of XACML Statement instances and inclusion of other
162 XACMLAssertion instances as advice. A SAML Attribute SHALL be contained in either a
163 SAML Assertion instance or in an XACMLAssertion instance when used in an XACML
164 system.

165 5. SAML AttributeQuery – A standard SAML protocol element that MAY be used by an XACML
166 PDP or PEP to request SAML Attribute instances from an Attribute Authority for use in an
167 XACML Request Context.

168 6. SAML Response – A standard SAML protocol element that SHALL be used to return SAML
169 Attribute instances in response to a SAML AttributeQuery in an XACML system.

170 Next, this Profile describes the use of SAML for use in requesting, responding with, storing, and
171 transmitting authorization decisions in an XACML system. The following elements are described:

172 1. XACMLAuthzDecisionStatement – A new SAML extension element defined in this Profile
173 that MAY be used in an XACML system to hold XACML authorization decisions for storage or
174 transmission.

175 2. XACMLAssertion – A new SAML extension element defined in this Profile that MAY be used in
176 an XACML system to hold XACMLAuthzDecisionStatement instances for storage or
177 transmission.

178 3. XACMLAuthzDecisionQuery – A new SAML extension protocol element defined in this Profile
179 that MAY be used by a PEP to request an authorization decision from an XACML PDP.

180 4. XACMLResponse – A new SAML extension protocol element defined in this Profile that SHALL
181 be used to return authorization decisions from an XACML PDP in response to an
182 XACMLAuthzDecisionQuery.

183 Then, this Profile describes the use of SAML for use in requesting, responding with, storing, and
184 transmitting XACML policies. The following elements are described:

185 1. XACMLPolicyStatement – A new SAML extension element defined in this Profile that MAY be
186 used in an XACML system to hold XACML policies for storage or transmission.

187 2. XACMLAssertion – A new SAML extension element defined in this Profile that MAY be used in
188 an XACML system to hold XACMLPolicyStatement instances for storage or transmission.

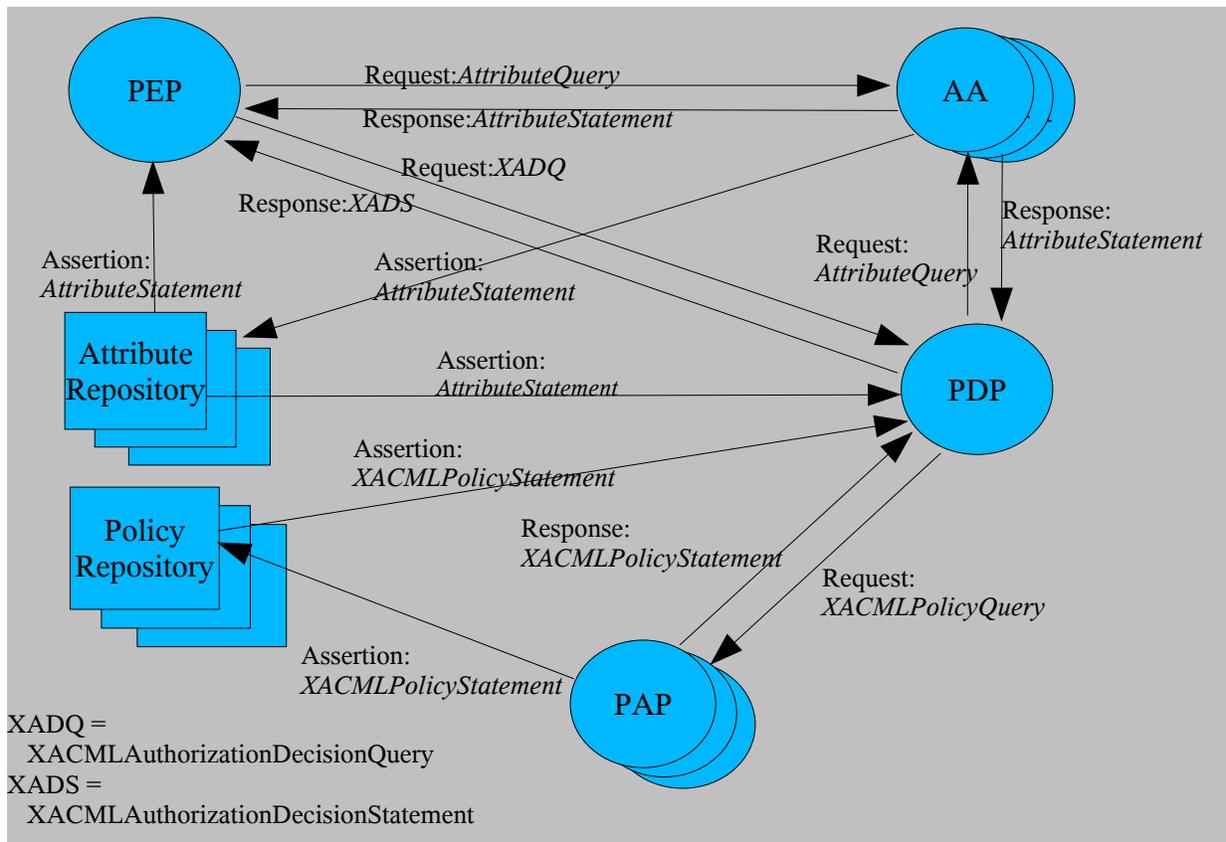
189 3. XACMLPolicyQuery – A new SAML extension protocol element defined in this Profile that MAY
190 be used by a PDP or other application to request XACML policies from a Policy Administration
191 Point.

192 4. XACMLResponse – A new SAML extension protocol element defined in this Profile that SHALL
193 be used to return policies from a Policy Administration Point in response to an
194 XACMLPolicyQuery.

195 Finally, this Profile describes the use of XACMLAssertion instances as advice in other Assertions. The
196 following elements are described:

- 197 1. XACMLAdvice – A new SAML extension element defined in this Profile that MAY be used for
 198 including XACMLAssertion instances as advice in another XACMLAssertion.
- 199 2. XACMLAssertion – A new SAML extension element that MAY be used to hold an
 200 XACMLAdvice instance along with SAML Statement or XACML extension Statement instances.

201 Figure 1 illustrates the XACML use model and the messages that can be used to communicate between
 202 the various components. Statements are carried in SAML or XACML Assertions, and Assertions are
 203 carried in SAML or XACML Responses as appropriate. Not all components or messages will be used in
 204 every implementation. Not shown, but described in this specification, is the ability to use an XACML
 205 Assertion in a SAML Advice instance.



206 Figure 1: Components and messages in an integration of SAML with an XACML system

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

210 1.1 Notation

211 In order to improve readability, the examples in this Profile assume use of the following XML Internal
 212 Entity declarations:

```

213 <!ENTITY saml "urn:oasis:names:tc:SAML:2.0:assertion">
214 <!ENTITY samlp "urn:oasis:names:tc:SAML:2.0:protocol">
215 <!ENTITY xacml "urn:oasis:names:tc:xacml:2.0:">
216 <!ENTITY xacml-context "urn:oasis:names:tc:xacml:2.0:context:schema:os">

```

```
217 <!ENTITY xs "http://www.w3.org/2001/XMLSchema#">
218 <!ENTITY subject-id "urn:oasis:names:tc:xacml:1.0:subject:subject-id">
219 <!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:">
220 <!ENTITY resource-id "urn:oasis:names:tc:xacml:1.0:resource:resource-id">
221 <!ENTITY action-id "urn:oasis:names:tc:xacml:1.0:action:action-id">
222 <!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:">
223 <!ENTITY current-dateTime "urn:oasis:names:tc:xacml:1.0:environment:current-
224 dateTime">
```

225 For example, “&xml:string” is equivalent to “http://www.w3.org/2001/XMLSchema#string”.

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

```
228 xacml-saml="urn:oasis:names:tc:xacml:2.1:profile:saml2.0:schema:assertion"
```

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

```
231 xacml-samlp="urn:oasis:names:tc:xacml:2.1:profile:saml2.0:schema:protocol"
```

232 1.2 Terminology

233 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
234 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as
235 described in IETF RFC 2119 [RFC 2119]

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

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

239 • An XACML Attribute is a typed name/value pair, with other optional information, specified using an
240 XACML Request Context <xacml-context:Attribute> element. An XACML Attribute is
241 associated with a subject, resource, action, or environment identity by the XACML Attribute's position
242 within the XACML Request; for example, an XACML Attribute contained within a <xacml-
243 context:Resource> instance is an attribute of that resource.

240 • A SAML Attribute is a name/value pair, with other optional information, specified using a SAML
241 Assertion <saml:Attribute> instance. A SAML Attribute is associated with a particular subject by
242 its inclusion in a <saml:Assertion> or an <xacml-saml:XACMLAssertion> instance that
243 contains a <saml:Subject> instance. The SAML Subject may correspond to an XACML
244 Subject, Resource, Action, or even Environment.

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

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

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

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

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

252 1.3 Normative References

- 253 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
254 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 255 **[SAML]** S. Cantor, et al., eds., *Assertions and Protocols for the OASIS Security
256 Assertion Markup Language (SAML) V2.0*, [http://www.oasis-
257 open.org/committees/documents.php?wg_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security).
- 258 **[SAML-PROFILE]** J. Hughes, et al., eds., *Profiles for the OASIS Security Assertion Markup
259 Language (SAML) V2.0*, [http://www.oasis-
260 open.org/committees/documents.php?wg_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security).
- 261 **[XACML]** T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML)
262 Version 2.0*, OASIS Standard, 1 February 2005, [http://docs.oasis-
263 open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf).
- 264 **[XACML-SAML]** A. Anderson, ed., *xacml-2.1-profile-saml2.0-schema-assertion.xsd*,
265 [http://docs.oasis-open.org/xacml/2.1/xacml-2.1-profile-saml2.0-schema-
266 assertion.xsd](http://docs.oasis-open.org/xacml/2.1/xacml-2.1-profile-saml2.0-schema-assertion.xsd)
- 267 **[XACML-SAML-P]** A. Anderson, ed., *xacml-2.1-profile-saml2.0-schema-protocol.xsd*,
268 [http://docs.oasis-open.org/xacml/2.1/xacml-2.1-profile-saml2.0-schema-
269 protocol.xsd](http://docs.oasis-open.org/xacml/2.1/xacml-2.1-profile-saml2.0-schema-protocol.xsd).

270 1.4 Non-normative References

- 271 **[XACMLIntro]** S. Proctor, *A Brief Introduction to XACML*, [http://www.oasis-
272 open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html](http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html), 14
273 March 2003.
274

275

2 Attributes

276 In an XACML system, PEPs and PDP Context Handlers often need to retrieve attributes from on-line
277 Attribute Authorities or from Attribute Repositories. SAML provides assertion and protocol elements that
278 MAY be used for retrieval of attributes for use in an XACML Request Context. These elements include a
279 `<saml:Attribute>` element for expressing a named attribute value, a
280 `<saml:AttributeStatement>` for holding a collection of `<saml:Attribute>` elements, and a
281 `<saml:Assertion>` element that can hold various kinds of statements, including a
282 `<saml:AttributeStatement>`. A `<saml:Assertion>` includes the name of the attribute issuer, an
283 optional digital signature for authenticating the attribute, an optional subject identity to which the attribute
284 is bound, and optional conditions for use of the assertion that include an optional validity period for use of
285 the attribute. Such an assertion is suitable for storing attributes in a Repository and for transmitting
286 attributes between an Attribute Authority and an Attribute Repository, and between an Attribute
287 Repository and a PEP or Context Handler. The `<xacml-saml:XACMLAssertion>` element defined in
288 this Profile is an alternative to the `<saml:Assertion>` element that MAY be used to hold
289 `<saml:Attribute>` instances. In the remainder of this Section, "Attribute Assertion" SHALL refer to
290 either a `<saml:Assertion>` or an `<xacml-saml:XACMLAssertion>` that contains one or more
291 `<saml:Attribute>` instances. For querying an on-line Attribute Authority for attributes, and for holding
292 the response to that query, SAML defines `<samlp:AttributeQuery>` and `<samlp:Response>`
293 elements. This Section describes the use of these SAML elements in an XACML system.

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

2.1 Element `<saml:Attribute>`

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

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

304 An `<xacml-context:Attribute>` instance SHALL be constructed from the corresponding
305 `<saml:Attribute>` instance contained in an Attribute Assertion as follows.

- 306 • XACML `AttributeId` XML attribute

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

- 308 • XACML `DataType` XML attribute

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

- 312 • XACML `Issuer` XML attribute

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

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

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

317 Each <saml:Attribute> instance SHALL be mapped to no more than one <xacml-
318 context:Attribute> instance. Not all <saml:Attribute> instances in an Attribute Assertion need
319 to be mapped; a subset of SAML Attribute instances MAY be selected by a mechanism not specified
320 here. The Issuer of the Attribute Assertion SHALL be used as the Issuer for each <xacml-
321 context:Attribute> instance that is created from <saml:Attribute> instances in that Attribute
322 Assertion.

323 The <xacml-context:Attribute> created from the Attribute Assertion SHALL be placed into the
324 <xacml-context:Resource>, <xacml-context:Subject>, <xacml-context:Action>, or
325 <xacml-context:Environment> instance that corresponds to the entity that is represented by the
326 <saml:Subject> in the Attribute Assertion. For example, if the Attribute Assertion Subject contains a
327 <saml:NameIdentifier> instance, and the value of that NameIdentifier matches the value of the
328 <xacml-context:Attribute> having an AttributeId of &resource;resource-id, then
329 <xacml-context:Attribute> instances created from <saml:Attribute> instances in that
330 Attribute Assertion SHALL be placed into the <xacml-context:Resource> instance. If a mapped
331 <saml:Attribute> is placed into an <xacml-context:Subject> instance, then the XACML
332 SubjectCategory XML attribute SHALL also be consistent with the entity that is the Subject of the
333 Attribute Assertion that contained the <saml:Attribute>.

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

- 339 • Any NotBefore and NotOnOrAfter XML attributes in the Attribute Assertion SHALL be valid with
340 respect to the <xacml:Request> in which the SAML-derived <xacml:Attribute> is used. This
341 means that the NotBefore and NotOnOrAfter XML attribute values SHALL be consistent with the
342 &environment;current-time, &environment;current-date, and
343 &environment;current-dateTime <xacml:Attribute> values associated with the
344 <xacml:Request>.
- 340 • The entity doing the mapping SHALL ensure that the semantics defined by SAML for any
341 <saml:AudienceRestrictionCondition> or <saml:DoNotCacheCondition> elements have
342 been adhered to.
- 341 • If a <ds:Signature> instance occurs in the Attribute Assertion, then the entity performing the
342 mapping SHALL ensure that the signature is valid and that the SAML <Issuer> instance is
343 consistent with any <ds:X509IssuerName> value in the signature. The guidelines regarding digital
344 signatures in Section 5: SAML and XML Signature Syntax and Processing of the SAML core
345 specification [SAML] SHALL be adhered to.

342 2.2 Element <saml:AttributeStatement>

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

344 2.3 Element <saml:Assertion>

345 When a <saml:AttributeStatement> instance is stored or transmitted in an XACML system, the
346 instance SHALL be enclosed in either a <saml:Assertion> or <xacml-saml:XACMLAssertion>.
347 The definition and use of the <saml:Assertion> element SHALL be as specified in the SAML 2.0
348 standard, augmented with the following requirements. Except as specified here, this Profile imposes no
349 requirements or restrictions on the <saml:Assertion> element and its contents beyond those
350 specified in SAML 2.0.

346 <saml:Issuer> [Required]

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

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

349 When a <saml:Assertion> containing a <saml:Attribute> is used to construct an XACML
350 Attribute, the string value of the <saml:Issuer> instance will be used as the value of the XACML
351 Issuer XML attribute, so the SAML value SHOULD be specified with this in mind.

350 <ds:Signature> [Optional]

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

352 A <ds:Signature> instance MAY be used in a <saml:Assertion>. In order to support 3rd party
353 digital signatures, this Profile does NOT require that the identity provided in the <saml:Issuer>
354 instance be consistent with the identity of the signer. It is up to the relying party to have an
355 appropriate trust relationship with the authority that signs the <saml:Assertion>.

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

354 <saml:Subject> [Optional]

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

356 In a <saml:Assertion> containing a <saml:Attribute> that is to be mapped to an XACML
357 Attribute, the <saml:Subject> instance SHALL contain the identity of the entity to which the
358 attribute and its value are bound. For a mapped Attribute to be placed in the XACML <xacml-
359 context:Subject> instance, this identity SHOULD be consistent with the value of any XACML
360 Attribute having an AttributeId of &subject-id; that occurs in the same <xacml-
361 context:Subject> instance. For a mapped Attribute to be placed in the XACML <xacml-
362 context:Resource> instance, this identity SHOULD be consistent with the value of any XACML
363 Attribute having an AttributeId of &resource-id; that occurs in the same <xacml-
364 context:Resource> instance. For a mapped Attribute to be placed in the XACML <xacml-
365 context:Action> instance, this identity SHOULD be consistent with the value of any XACML
366 Attribute having an AttributeId of &action-id; that occurs in the same <xacml-
367 context:Action> instance. For a mapped Attribute to be placed in the XACML <xacml-
368 context:Environment> instance, this identity SHOULD be consistent with the value of any
369 XACML Attribute that occurs in the same <xacml-context:Environment> instance and provides
370 an environment identity. See Section 2.1 for more information.

357 <saml:Conditions> [Optional]

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

359 The <saml:Conditions> instance SHOULD contain NotBefore and NotOnOrAfter XML
360 attributes to specify the limits on the validity of the Assertion. If these XML attributes are present,
361 the relying party SHOULD ensure that an <xacml-context:Attribute> derived from the
362 Assertion is used by a PDP for evaluating policies only when the value of the XACML Attribute in the
363 <xacml-context:Environment> instance having an AttributeId of ¤t-dateTime; is
364 contained within the Assertion's specified validity period.

360 2.4 Element <xacml-saml:XACMLAssertion>

361 The <xacml-saml:XACMLAssertion> element is an extension to the <saml:Assertion> element
362 that MAY be used to hold <saml:AttributeStatement> instances and <saml:Attribute>
363 instances in an XACML system. When used in an XACML system, <saml:Attribute> instances
364 SHALL be contained in either a <saml:Assertion> or <xacml-saml:Assertion> instance.

365 The <xacml-saml:XACMLAssertion> element allows the inclusion of other XACML extension
366 elements. All the requirements for use of a <saml:Assertion> described in Section 2.3 apply also to
367 an <xacml-saml:XACMLAssertion> when used as an Attribute Assertion. See Section 3.2 for a
368 description of the syntax of an <xacml-saml:XACMLAssertion>.

369 2.5 Element <samlp:AttributeQuery>

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

374 2.6 Element <samlp:Response>

375 The response to a <samlp:AttributeQuery> SHALL be either a <samlp:Response> instance or an
376 <xacml-samlp:XACMLResponse> instance containing an Attribute Assertion that holds any
377 <saml:AttributeStatement> instances that match the query. The definition and use of the
378 <samlp:Response> element SHALL be as described in the SAML 2.0 standard, augmented with the
379 following requirements. Except as specified here, this Profile imposes no requirements or restrictions on
380 the <samlp:Response> element and its contents beyond those specified in SAML 2.0.

381 <samlp:Issuer> [Required]

382 This element SHALL be handled as specified in Section 2.3 for the <saml:Issuer> element in a
383 <saml:Assertion>.

384 <ds:Signature> [Required]

385 This element SHALL be handled as specified in Section 2.3 for the <ds:Signature> in a
386 <saml:Assertion>.

387

3 Authorization Decisions

388 XACML defines `<xacml-context:Request>` and `<xacml-context:Response>` elements for
389 describing an authorization decision request and the corresponding response from a PDP. In many
390 environments, these instances of these elements need to be signed or associated with a validity period
391 in order to be used in an actual protocol between entities. SAML 2.0 defines a rudimentary
392 `<saml:AuthzDecisionQuery>` in the SAML Protocol Schema and a rudimentary
393 `<saml:AuthzDecisionStatement>` in the SAML Assertion Schema, but these elements are not able
394 to convey all the information that an XACML PDP is capable of accepting as part of its Request Context.
395 Likewise, the SAML `<saml:AuthzDecisionStatement>` is unable to convey all the information
396 contained in an XACML Response Context. In order to allow a PEP to use the SAML protocol with full
397 support for the XACML Request Context and XACML Response Context syntax, this specification
398 defines four SAML extension elements:

- 399 • `<xacml-saml:XACMLAuthzDecisionStatement>` allows a PDP Context Handler to include an
400 XACML `<xacml-context:Response>`, along with other optional information, in a SAML Statement.
- 401 • `<xacml-saml:XACMLAssertion>` allows a PDP Context Handler to include an `<xacml-`
402 `saml:XACMLAuthzDecisionStatement>` in a SAML Assertion.
- 403 • `<xacml-samlp:XACMLAuthzDecisionQuery>` allows a PEP to submit an XACML Request
404 Context, along with other optional information, as a SAML protocol query.
- 405 • `<xacml-samlp:XACMLResponse>` allows a PDP to include `<xacml-`
406 `saml:XACMLAuthzDecisionStatement>` instances in a SAML protocol response.

407 This Section defines these elements. All these elements are contained in the [XACML-SAML] and
408 [XACML-SAML P] schema documents.

3.1 Element `<XACMLAuthzDecisionStatement>`

410 The `<xacml-saml:XACMLAuthzDecisionStatement>` element MAY be used by an XACML PDP to
411 hold an authorization decision that provides full support for XACML functionality. It allows a SAML
412 Statement to contain an XACML Response Context along with related information. This element is an
413 alternative to the SAML-defined `<samlp:AuthzDecisionStatement>`. This element SHALL be used
414 as part of a response to an `<xacml-samlp:XACMLAuthzDecisionQuery>`. It MAY also be used in an
415 `<xacml-saml:XACMLAssertion>` as a format for storage of an authorization decision in a Repository.

```

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

```

416 The `<xacml-saml:XACMLAuthzDecisionStatement>` element is of `<xacml-`
 417 `saml:XACMLAuthzDecisionStatementType>` complexType, which is an extension to the SAML-
 418 defined `<saml:StatementAbstractType>`.

419 The `<xacml-saml:XACMLAuthzDecisionStatement>` element contains the following elements:

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

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

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

424 An `<xacml-context:Request>` element containing XACML Attributes that were used by the
 425 XACML PDP in evaluating policies to obtain the Response.

426 If the statement represents a response to an `<xacml-samlp:XACMLAuthzDecisionQuery>`, and
 427 if the `ReturnContext` XML attribute in the query is "true", then this element SHALL be included;
 428 if the `ReturnContext` XML attribute in the query is "false", then this element SHALL NOT be
 429 included. See the description of the `ReturnContext` XML attribute in Section 3.3 for a specification
 430 of the XACML Attribute values that SHALL be returned in this element when it is part of a response
 431 to an `<xacml-samlp:XACMLAuthzDecisionQuery>`.

432 If this statement does not represent the response to an `<xacml-`
 433 `samlp:XACMLAuthzDecisionQuery>`, then this element MAY be included. In this case, the PDP
 434 SHALL determine which XACML Attributes are included.

435 3.2 Element `<XACMLAssertion>`

436 The `<xacml-saml:XACMLAssertion>` element allows XACML or SAML Statements to be carried in a
 437 SAML Assertion, which MAY be signed. The `<xacml-saml:XACMLAssertion>` element MAY be used
 438 by a PDP, Policy Administration Authority, or Attribute Authority to hold `<xacml-`
 439 `saml:XACMLAuthzDecisionStatement>`, `<xacml-saml:XACMLPolicyStatement>`, or any of the
 440 standard SAML Statement instances, including `<saml:AttributeStatement>` instances.

```

<element name="XACMLAssertion"
  xsi:type="xacml-saml:XACMLAssertionType"
  <complexType name="XACMLAssertionType">
    <complexContent>
      <extension base="saml:AssertionType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element
            ref="xacml-saml:XACMLAdvice" maxOccurs="1" />
          <element
            ref="xacml-saml:XACMLAuthzDecisionStatement"/>
          <element
            ref="xacml-saml:XACMLPolicyStatement"/>
        </choice>
      </extension>
    </complexContent>
  </complexType>

```

441 The `<xacml-saml:XACMLAssertion>` element is of `<xacml-saml:XACMLAssertionType>`
 442 `complexType`, which is an extension of the SAML-defined `<saml:AssertionType>`.

443 The specification of the components of a `<saml:Assertion>` in SAML 2.0 SHALL apply to an
 444 `<xacml-saml:XACMLAssertion>`. The additional requirements and restrictions on a
 445 `<saml:Assertion>` specified in Section 2.3 SHALL also apply to an `<xacml-`
 446 `saml:XACMLAssertion>`.

447 The following elements are defined or further specified here for use with the extended SAML statement
 448 types defined and used in this Profile. These elements are in addition to the elements and attributes
 449 defined for the `<saml:AssertionType>`:

450 `<xacml-saml:XACMLAdvice>` [Optional]

451 "Additional information related to the assertion that assists processing in certain situations but which
 452 MAY be ignored by applications that do not understand the advice or do not wish to make use of it."
 453 [SAML] The `<xacml-saml:XACMLAdvice>` element allows the use of `<xacml-`
 454 `saml:XACMLAssertion>` instances as advice in assertions. See Section 5.1 for the definition of
 455 the `<xacml-saml:XACMLAdvice>` element.

456 `<saml:Subject>` [Optional]

457 The `<saml:Subject>` element SHALL NOT be included in an assertion that contains an `<xacml-`
 458 `saml:XACMLAuthzDecisionStatement>`. Instead, the Subject of an `<xacml-`
 459 `saml:XACMLAuthzDecision>` is specified in the XACML Request Context of the corresponding
 460 authorization decision request. This corresponding XACML Request Context MAY be included in the
 461 `<xacml-samlp:XACMLAuthzDecisionStatement>` as described in Section 3.1.

462 The `<saml:Subject>` element MAY be included in an assertion that contains an `<xacml-`
 463 `saml:XACMLPolicyStatement>` instance. There is usually not a unique Subject for an `<xacml-`
 464 `saml:XACMLPolicyStatement>`, as typically an XACML policy applies to multiple Subjects.

465 **3.3 Element `<XACMLAuthzDecisionQuery>`**

466 The `<xacml-samlp:XACMLAuthzDecisionQuery>` protocol element MAY be used by a PEP to
 467 request an authorization decision from an XACML PDP. This element is an alternative to the SAML-
 468 defined `<samlp:AuthzDecisionQuery>` and allows the PEP to use the full capabilities of an XACML

469 PDP. It allows the SAML query protocol to convey an XACML Request Context along with related
470 information.

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

470 The `<xacml-samlp:XACMLAuthzDecisionQuery>` element is of `<xacml-`
471 `samlp:XACMLAuthzDecisionQueryType>` complexType, which is an extension to the SAML-defined
472 `<samlp:RequestAbstractType>`.

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

472 InputContextOnly [Default "false"]

473 This XML attribute governs the sources of information that the PDP is allowed to use in making its
474 authorization decision. If this XML attribute is "true", then the authorization decision SHALL be
475 made solely on the basis of information contained in the `<xacml-`
476 `samlp:XACMLAuthzDecisionQuery>`; no external attributes MAY be used. If this XML attribute is
477 "false", then the authorization decision MAY be made on the basis of external attributes not
478 contained in the `<xacml-samlp:XACMLAuthzDecisionQuery>`.

474 ReturnContext [Default "false"]

475 This XML attribute allows the PEP to request that an `<xacml-context:Request>` instance be
476 included in the `<xacml-saml:XACMLAuthzDecisionStatement>` resulting from the query. It
477 also governs the contents of that `<xacml-context:Request>` instance.

476 If this XML attribute is "true", then the PDP SHALL include an `<xacml-context:Request>`
477 instance in the `<xacml-saml:XACMLAuthzDecisionStatement>` in the `<xacml-`
478 `samlp:XACMLResponse>` used in the response to the query. This `<xacml-context:Request>`
479 instance SHALL include all those attributes supplied by the PEP in the `<xacml-`
480 `samlp:XACMLAuthzDecisionQuery>` that were used in making the authorization decision. The
481 PDP MAY include additional attributes in this `<xacml-context:Request>` instance, such as
482 external attributes obtained by the PDP and used in making the authorization decision, or other

477 attributes known by the PDP that may be useful to the PEP in making subsequent authorization
478 decision queries.

478 If this XML attribute is "false", then the PDP SHALL NOT include an <xacml-
479 context:Request> instance in the <xacml-saml:XACMLAuthzDecisionStatement>
480 contained in the <xacml-samlp:XACMLResponse> that is used in the query response.

479 <xacml-context:Request> [Required]

480 An XACML Request Context.

481 <xacml:Policy> [0 to Many]

482 Optional XACML Policy instances that MAY be used in evaluating this authorization decision request
483 only. The PDP MAY choose to use such Policy instances. If used, the PDP SHALL determine the
484 combining algorithm.

483 <xacml:PolicySet> [Any Number]

484 Optional XACML PolicySet instances that MAY be used in evaluating this authorization decision
485 request only. The PDP MAY choose to use such PolicySet instances . If used, the PDP SHALL
486 determine the combining algorithm.

485 3.4 Element <XACMLResponse>

486 The <xacml-samlp:XACMLResponse> element allows XACML Statements to be carried in a SAML
487 response, which MAY be signed. An <xacml-samlp:XACMLResponse> instance SHALL be used
488 when <xacml-saml:XACMLAssertion> instances are included in SAML protocol responses. The
489 <xacml-samlp:XACMLResponse> element MAY be used to carry other SAML <saml:Assertion>
490 instances, including those containing <saml:AttributeStatement> instances to be mapped to
491 XACML Attributes.

```
<element name="XACMLResponse"  
  xsi:type="xacml-samlp:XACMLResponseType">  
  <complexType name="XACMLResponseType">  
    <complexContent>  
      <extension base="samlp:ResponseType">  
        <choice minOccurs="0" maxOccurs="unbounded">  
          <element  
            ref="xacml-saml:XACMLAssertion"/>  
        </choice>  
      </extension>  
    </complexContent>  
  </complexType>
```

492 The <xacml-samlp:XACMLResponse> element is of <xacml-samlp:XACMLResponseType>
493 complexType, which is an extension of the SAML-defined <samlp:ResponseType>.

494 The specification of the components of a <samlp:Response> in SAML 2.0 SHALL apply to an
495 <xacml-samlp:XACMLResponse>.

496 The following additional elements and requirements are specified here for use in responses to an
497 <xacml-samlp:XACMLAuthzDecisionQuery>.

498 <samlp:Issuer> [Required]

499 This element SHALL be handled as specified in Section 2.3 for the <saml:Issuer> element in a
500 <saml:Assertion>.

501 <ds:Signature> [Required]

502 This element SHALL be handled as specified in Section 2.3 for the <ds:Signature> in a
503 <saml:Assertion>.

504 <xacml-saml:XACMLAssertion> [Any Number]

505 <xacml-saml:XACMLAssertion> instances containing SAML Statement or XACML extended
506 Statement instances that represent the response to the associated query.

507 <samlp:StatusCode> [Required]

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

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

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

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

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

517 This value for the <samlp:StatusCode> Value XML attribute SHALL be used when the
518 <xacml-context:StatusCode> value is
519 urn:oasis:names:tc:xacml:1.0:status:missing-attribute or the when the
520 <xacml-context:StatusCode> value is
521 urn:oasis:names:tc:xacml:1.0:status:syntax-error due to a syntax error in the
522 <xacml-context:Request>.

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

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

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

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

4 Policies

533

534 XACML defines two policy elements: `<xacml:Policy>` and `<xacml:PolicySet>`. These may need
535 to be transmitted between entities in an XACML system, but SAML does not define any Protocol or
536 Assertion elements for policies. In order to query for and make assertions about policies, this
537 specification defines four SAML extension elements:

- 538 • `<xacml-saml:XACMLPolicyStatement>` allows one or more XACML policies to be included in a
539 SAML statement.
- 540 • `<xacml-saml:XACMLAssertion>` allows an `<xacml-saml:XACMLPolicyStatement>` to be
541 included in a SAML assertion.
- 542 • `<xacml-samlp:XACMLPolicyQuery>` allows a PDP or application to request XACML policies as a
543 SAML protocol query.
- 544 • `<xacml-samlp:XACMLResponse>` allows an `<xacml-saml:XACMLAssertion>`, which may
545 contain an `<xacml-saml:XACMLPolicyStatement>`, to be included in a SAML protocol response.

546 This Section defines these elements for use with XACML policies. All these elements are contained in
547 the [XACML-SAML] and [XACML-SAML] schema documents.

4.1 Element `<XACMLPolicyStatement>`

548

549 The `<xacml-saml:XACMLPolicyStatement>` element MAY be used by a Policy Administration Point
550 to hold one or more XACML policies. This element MAY be used as part of a response to an `<xacml-
551 samlp:XACMLPolicyQuery>`. This element may also be used in a SAML Assertion as a format for
552 storing an XACML policy in a Repository.

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

553 The `<xacml-saml:XACMLPolicyStatement>` element is of `<xacml-
554 saml:XACMLPolicyStatementType>` `complexType`, which is an extension to the SAML-defined
555 `<saml:StatementAbstractType>`.

554 An `<xacml-saml:XACMLPolicyStatement>` element contains the following elements.

555 `<xacml:Policy>` [Any Number]

556 If the statement represents a response to an `<xacml-samlp:XACMLPolicyQuery>`, then this
557 element SHALL contain all `<xacml:Policy>` instances that meet the specifications of the
558 associated `<xacml-samlp:XACMLPolicyQuery>`. Otherwise, this element MAY contain arbitrary
559 `<xacml:Policy>` instances.

560 <xacml:PolicySet> [Any Number]

561 If the statement represents a response to an <xacml-samlp:XACMLPolicyQuery>, then this
562 element SHALL contain all <xacml:PolicySet> instances that meet the specifications of the
563 associated <xacml-samlp:XACMLPolicyQuery>. Otherwise, this element MAY contain arbitrary
564 <xacml:PolicySet> instances.

565 If the <xacml-saml:XACMLPolicyStatement> is issued in response to an <xacml-
566 samlp:XACMLPolicyQuery>, and there are no <xacml:Policy> or <xacml:PolicySet> instances
567 that meet the specifications of the associated <xacml-samlp:XACMLPolicyQuery>, then there
568 SHALL be exactly one empty <xacml-saml:XACMLPolicyStatement> included in the response.

569 4.2 Element <XACMLAssertion>

570 When an <xacml-saml:XACMLPolicyStatement> instance is stored or transmitted in an XACML
571 system, the instance SHALL be enclosed in an <xacml-saml:XACMLAssertion> instance. When an
572 <xacml-saml:XACMLAssertion> is part of a response to an <xacml-
573 samlp:XACMLPolicyQuery>, then the <xacml-saml:XACMLAssertion> SHALL contain exactly
574 one <xacml-saml:XACMLPolicyStatement>. The definition and use of the <xacml-
575 saml:XACMLAssertion> element is described in Section 3.2.

576 4.3 Element <XACMLPolicyQuery>

577 The <xacml-samlp:XACMLPolicyQuery> element MAY be used by a PDP or application to request
578 XACML <xacml:Policy> or <xacml:PolicySet> instances from an on-line Policy Administration
579 Point.

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

580 The <xacml-samlp:XACMLPolicyQuery> element is of <xacml-samlp:XACMLPolicyQueryType>
581 complexType, which is an extension to the SAML-defined <samlp:RequestAbstractType>.

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

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

585 Supplies an XACML Request Context. All XACML <xacml:Policy> and <xacml:PolicySet>
586 instances applicable to this Request SHALL be returned. The concept of "applicability" in the
587 XACML context is defined in the XACML 2.0 Specification [XACML].

588 <xacml:Target> [Any Number]

589 Supplies an XACML `<xacml:Target>` instance. All XACML `<xacml:Policy>` and
590 `<xacml:PolicySet>` instances applicable to this `<Target>` SHALL be returned.

591 `<xacml:PolicySetIdReference>` [Any Number]

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

593 `<xacml:PolicyIdReference>` [Any Number]

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

595 If the `<xacml-samlp:XACMLPolicyQuery>` contains no element instances, then the Policy
596 Administration Point SHOULD return all policies that are authorized and appropriate for use by the
597 requester.

598 **4.4 Element `<XACMLResponse>`**

599 The response to an `<xacml-samlp:XACMLPolicyQuery>` SHALL be an `<xacml-samlp:Response>`
600 instance containing exactly one `<xacml-saml:XACMLAssertion>` instance that contains exactly one
601 `<xacml-saml:XACMLPolicyStatement>` instance.

602 The specification of the components of a `<saml:Response>` in SAML 2.0 SHALL apply to an `<xacml-samlp:XACMLResponse>`. The following additional elements and requirements are specified here for
603 use in responses to an `<xacml-samlp:XACMLPolicyQuery>`.
604

605 `<samlp:Issuer>` [Required]

606 This element SHALL be handled as specified in Section 2.3 for the `<saml:Issuer>` element in a
607 `<saml:Assertion>`.

608 `<ds:Signature>` [Required]

609 This element SHALL be handled as specified in Section 2.3 for the `<ds:Signature>` in a
610 `<saml:Assertion>`.

611 `<xacml-saml:XACMLAssertion>` [Any Number]

612 This element SHALL contain exactly one `<xacml-saml:XACMLPolicyStatement>` that
613 represents the response to the associated query. It MAY contain other SAML or XACML Assertions.

614 5 Advice

615 This Section describes how to include `<xacml-saml:XACMLAssertion>` instances as advice to
616 accompany any SAML Statement or XACML extended Statement.

617 5.1 Element `<XACMLAdvice>`

618 A SAML Assertion includes an optional `<saml:Advice>` element containing “Additional information
619 related to the assertion that assists processing in certain situations but which MAY be ignored [without
620 affecting either the semantics or the validity of the assertion] by applications that do not understand the
621 advice or do not wish to make use of it.” [SAML] The `<xacml-saml:XACMLAdvice>` element extends
622 `<saml:Advice>` to allow the inclusion of `<xacml-saml:XACMLAssertion>` instances containing
623 `<xacml-saml:XACMLAuthzDecisionStatement>` or `<xacml-saml:XACMLPolicyStatement>`
624 instances.

```
<element name="XACMLAdvice"
  xsi:type="xacml-saml:XACMLAdviceType"
  <complexType name="XACMLAdviceType">
    <complexContent>
      <extension base="saml:AdviceType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element
            ref="xacml-saml:XACMLAssertion"/>
        </choice>
      </extension>
    </complexContent>
  </complexType>
```

625 The `<xacml-saml:XACMLAdvice>` element is of `<xacml-saml:XACMLAdviceType>` complexType,
626 which is an extension to the SAML-defined `<saml:AdviceType>`.

627 The `<xacml-saml:XACMLAdvice>` element contains the following elements in addition to those
628 defined in the `<saml:AdviceType>`:

629 `<xacml-saml:XACMLAssertion>` [Any Number]

630 An assertion representing advice for the use of the outer assertion. It MAY contain any number of
631 `<xacml-saml:XACMLAuthzDecisionStatement>` or `<xacml-`
632 `saml:XACMLPolicyStatement>` instances.

633 5.2 Element `<XACMLAssertion>`

634 The `<xacml-saml:XACMLAssertion>` element includes an optional `<xacml-saml:XACMLAdvice>`
635 element. Since the `<xacml-saml:XACMLAssertion>` extends the standard `<saml:Assertion>`,
636 this means that not only `<xacml-saml:XACMLAuthzDecisionStatement>` and `<xacml-`
637 `saml:XACMLPolicyStatement>` instances, but any instance of a SAML Statement, may be used in an
638 `<xacml-saml:XACMLAssertion>`, and may be associated there with an `<xacml-`
639 `saml:XACMLAdvice>` instance that includes an `<xacml-saml:XACMLAssertion>` to be used as
640 advice for the outer `<xacml-saml:XACMLAssertion>`.

641

Appendix A. Acknowledgments

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

644 **Participants:**

- 645 • Anne Anderson, Sun Microsystems
- 646 • Anthony Nadalin, IBM
- 647 • Bill Parducci,
- 648 • Carlisle Adams, University of Ottawa
- 649 • Daniel Engovatov, BEA
- 650 • Don Flinn,
- 651 • Ed Coyne
- 652 • Ernesto Damiani
- 653 • Frank Siebenlist
- 654 • Gerald Brose
- 655 • Hal Lockhart
- 656 • Haruyuki Kawabe
- 657 • James MacLean
- 658 • John Merrells
- 659 • Ken Yagen
- 660 • Konstantin Beznosov
- 661 • Michiharu Kudo
- 662 • Michael McIntosh
- 663 • Pierangela Samarati
- 664 • Pirasenna Velandai Thiyagarajan
- 665 • Polar Humenn
- 666 • Rebekah Metz
- 667 • Ron Jacobson
- 668 • Satoshi Hada
- 669 • Sekhar Vajjhala
- 670 • Seth Proctor
- 671 • Simon Godik
- 672 • Steve Anderson
- 673 • Steve Crocker
- 674 • Suresh Damodaran
- 675 • Tim Moses
- 676 • Von Welch
- 677 • Frederic Deleon
- 678 • Argyn Kuketayev

679

Appendix B. Revision History

Rev	Date	By whom	What
WD 1	12 April 2006	Anne Anderson	Create from SAML Profile errata document. <XACMLAuthzDecisionStatementType>: replace "ReturnResponse" with "ReturnContext" in description. Authorization Decisions: replaced "in the Response to an <XACMLAuthzDecisionStatement>" with "...<XACMLAuthzDecisionQuery>". Create new types for SAML elements that will need to include XACML extensions. Create new elements for each extended type. Allow an XACMLAuthzDecisionQuery to include XACML policies for use in evaluating that query. Allow an XACMLAssertion to contain an XACMLAdvice element that in turn can contain an XACMLAssertion.