



SAML 2.0 Profile of XACML, Version 2

Working Draft 5

19 July 2007

Specification URIs:

[document identifier as per OASIS Artifact Naming Guidelines]

This Version:

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].html](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].html)

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].pdf](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].pdf)

Previous Version:

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].html](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].html)

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].pdf](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].pdf)

Latest Version:

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].html](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].html)

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].pdf](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].pdf)

Latest Approved Version:

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].html](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].html)

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].pdf](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].pdf)

Technical Committee:

OASIS eXtensible Access Control Markup Language (XACML) TC

Chair(s):

Hal Lockhart

Bill Parducci

Editor:

Anne Anderson

Related Work:

This specification replaces and supersedes:

- SAML 2.0 profile of XACML 2.0

This specification is related to:

- SAML 2.0 OASIS Standard

- 32 • XACML 1.0, 2.0, 3.0 OASIS Standards
33 • XACML 1.1 Committee Draft

34 **Declared XML Namespace(s):**

35 [list namespaces here]

36 [list namespaces here]

37 **Abstract:**

38 This specification defines a profile for the integration of the OASIS Security Assertion Markup
39 Language (SAML) Version 2.0 with all versions of XACML. SAML 2.0 complements XACML
40 functionality in many ways, so a number of somewhat independent functions are described in
41 this profile: 1) use of SAML 2.0 Attribute Assertions with XACML, including the use of SAML
42 Attribute Assertions in a SOAP Header to convey Attributes that can be consumed by an XACML
43 PDP, 2) use of SAML to carry XACML authorization decisions, authorization decision queries,
44 and authorization decision responses, 3) use of SAML to carry XACML policies, policy queries,
45 and policy query responses, 4) use of XACML authorization decisions or policies as Advice in
46 SAML Assertions, and 5) use of XACML responses in SAML Assertions as authorization tokens.
47 Particular implementations may provide only a subset of these functions.

48 **Status:**

49
50 This document was last revised or approved by the [TC name | membership of OASIS] on the
51 above date. The level of approval is also listed above. Check the "Latest Version" or "Latest
52 Approved Version" location noted above for possible later revisions of this document.

53 Technical Committee members should send comments on this specification to the Technical
54 Committee's email list. Others should send comments to the Technical Committee by using the
55 "Send A Comment" button on the Technical Committee's web page at [http://www.oasis-](http://www.oasis-open.org/committees/[specific location]/)
56 [open.org/committees/\[specific location\]/](http://www.oasis-open.org/committees/[specific location]/).

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

61 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/[specific location]/)
62 [open.org/committees/\[specific location\]/](http://www.oasis-open.org/committees/[specific location]/).

63 Notices

64 Copyright © OASIS® 1993–2007. All Rights Reserved. OASIS trademark, IPR and other policies apply.

65 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
66 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

67 This document and translations of it may be copied and furnished to others, and derivative works that
68 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
69 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
70 notice and this section are included on all such copies and derivative works. However, this document
71 itself may not be modified in any way, including by removing the copyright notice or references to OASIS,
72 except as needed for the purpose of developing any document or deliverable produced by an OASIS
73 Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR
74 Policy, must be followed) or as required to translate it into languages other than English.

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

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

82 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
83 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard,
84 to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to
85 such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that
86 produced this specification.

87 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of
88 any patent claims that would necessarily be infringed by implementations of this specification by a patent
89 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
90 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
91 claims on its website, but disclaims any obligation to do so.

92 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
93 might be claimed to pertain to the implementation or use of the technology described in this document or
94 the extent to which any license under such rights might or might not be available; neither does it
95 represent that it has made any effort to identify any such rights. Information on OASIS' procedures with
96 respect to rights in any document or deliverable produced by an OASIS Technical Committee can be
97 found on the OASIS website. Copies of claims of rights made available for publication and any
98 assurances of licenses to be made available, or the result of an attempt made to obtain a general license
99 or permission for the use of such proprietary rights by implementers or users of this OASIS Committee
100 Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no
101 representation that any information or list of intellectual property rights will at any time be complete, or
102 that any claims in such list are, in fact, Essential Claims.

103 The names "OASIS", [insert specific trademarked names, abbreviations, etc. here] are trademarks of
104 OASIS, the owner and developer of this specification, and should be used only to refer to the
105 organization and its official outputs. OASIS welcomes reference to, and implementation and use of,
106 specifications, while reserving the right to enforce its marks against misleading uses. Please see
107 <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

109	1 Introduction.....	6
110	1.1 Organization of this Profile.....	6
111	1.2 Diagram of SAML integration with XACML.....	8
112	1.3 Backwards compatibility.....	9
113	1.4 Namespaces.....	11
114	1.5 Normative References.....	12
115	1.6 Non-normative References.....	12
116	2 Attributes.....	13
117	2.1 Element <saml:Attribute>.....	13
118	2.2 Element <saml:AttributeStatement>.....	15
119	2.3 Element <saml:Assertion>: SAML Attribute Assertion.....	15
120	2.4 Element <samlp:AttributeQuery>.....	16
121	2.5 Element <samlp:Response>: SAML Attribute Response.....	17
122	2.6 Conveying XACML Attributes in a SOAP Message.....	17
123	3 Authorization Decisions.....	19
124	3.1 Type <xacml-saml:XACMLAuthzDecisionStatementType>.....	19
125	3.2 Element <saml:Statement>: XACMLAuthzDecision Statement.....	20
126	3.3 Element <saml:Assertion>: XACMLAuthzDecision Assertion.....	21
127	3.4 Element <xacml-samlp:XACMLAuthzDecisionQuery>.....	22
128	3.5 Element <xacml-samlp:AdditionalAttributes>.....	25
129	3.6 Element <xacml-samlp:AssignedAttributes>.....	26
130	3.7 Element <xacml-samlp: HOLDERS>.....	26
131	3.8 Element <xacml-samlp:HolderAttributes>.....	27
132	3.9 Element <xacml-saml:ReferencedPolicies>.....	27
133	3.10 Element <samlp:Response>: XACMLAuthzDecision Response.....	28
134	3.11 Functional Requirements for the <xacml-samlp:AssignedAttributes> Element.....	31
135	4 Policies.....	32
136	4.1 Type <xacml-saml:XACMLPolicyStatementType>.....	32
137	4.2 Element <xacml-saml:ReferencedPolicies>.....	34
138	4.3 Element <saml:Statement>: XACMLPolicy Statement.....	34
139	4.4 Element <saml:Assertion>: XACMLPolicy Assertion.....	34
140	4.5 Element <xacml-samlp:XACMLPolicyQuery>.....	35
141	4.6 Element <samlp:Response>: XACMLPolicy Response.....	36
142	5 Advice.....	39
143	5.1 Element <saml:Advice>.....	39
144	6 Using an XACML Authorization Decision as an Authorization Token.....	40
145	7 SAML Metadata.....	41
146	7.1 Type <xacml-samlm:XACMLPDPDescriptorType>.....	42

147	7.2 Type <xacml-samlm:XACMLPDPConfigType>.....	43
148	7.3 Type <xacml-samlm:XACMLAuthzDecisionQueryDescriptorType>.....	43
149	7.4 Type <xacml-samlm:XACMLAuthzDecisionQueryConfigType>.....	44
150	8 Conformance.....	45
151		

1 Introduction

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185

186
187
188
189
190
191
192
193
194
195

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

Non-normative through Section 1.4

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

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

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

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

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

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

1.1 Organization of this Profile

This Profile defines how to use SAML 2.0 to protect, store, transport, request, and respond with XACML schema instances and other information needed by an XACML implementation. The remaining Sections of this Profile describe the following aspects of SAML 2.0 usage.

Section 2 describes how to use SAML Attributes in an XACML system. It describes the use of the following elements:

1. `<saml:Attribute>` – A standard SAML element that MAY be used in an XACML system for storing and transmitting attribute values. The `<saml:Attribute>` must be at least conceptually transformed into an `<xacml-context:Attribute>` before it can be used in an XACML Request Context.

- 196 2. <saml:AttributeStatement> – A standard SAML element that MUST be used to hold
197 <saml:Attribute> instances in an XACML system.
- 198 3. <saml:Assertion> – A standard SAML element that MUST be used to hold
199 <saml:AttributeStatement> instances in an XACML system, either in an Attribute
200 Repository or in a SAML Attribute Response. The <saml:Assertion> contains information
201 that is required in order to transform a <saml:Attribute> into an <xacml-
202 context:Attribute>. An instance of such a <saml:Assertion> element is called a SAML
203 Attribute Assertion in this Profile.
- 204 4. <samlp:AttributeQuery> – A standard SAML protocol element that MAY be used by an
205 XACML PDP or PEP to request <saml:Attribute> instances from an Attribute Authority for
206 use in an XACML Request Context.
- 207 5. <samlp:Response> – A standard SAML protocol element that MUST be used to return SAML
208 Attribute Assertions in response to a <samlp:AttributeQuery> in an XACML system. An
209 instance of such a <samlp:Response> element is called a SAML Attribute Response in this
210 Profile.

211 Section 3 describes ways to convey XACML Attributes in a SOAP message.

212 Section 4 describes the use of SAML in requesting, responding with, storing, and transmitting
213 authorization decisions in an XACML system. The following types and elements are described:

- 214 1. `xacml-saml:XACMLAuthzDecisionStatementType` – A new SAML extension type defined
215 in this Profile that MAY be used in an XACML system to create XACMLAuthzDecision
216 Statements that hold XACML authorization decisions for storage or transmission.
- 217 2. <saml:Statement> – A standard SAML element that MUST be used to contain instances of
218 the <xacml-saml:XACMLAuthzDecisionStatementType>. An instance of such a
219 <saml:Statement> element is called an XACMLAuthzDecision Statement in this Profile.
- 220 3. <saml:Assertion> – A standard SAML element that MUST be used to hold
221 XACMLAuthzDecision Statements in an XACML system, either in a repository or in a
222 XACMLAuthzDecision Response. An instance of such a <saml:Assertion> element is called
223 an XACMLAuthzDecision Assertion in this Profile.
- 224 4. <xacml-samlp:XACMLAuthzDecisionQuery> – A new SAML extension protocol element
225 defined in this Profile that MAY be used by a PEP to request an authorization decision from an
226 XACML PDP.
- 227 5. <samlp:Response> – A standard SAML protocol element that MUST be used to return
228 XACMLAuthzDecision Assertions from an XACML PDP in response to an <xacml-
229 samlp:XACMLAuthzDecisionQuery>. An instance of such a <samlp:Response> element
230 is called an XACMLAuthzDecision Response in this Profile.

231 Section 5 describes the use of SAML in requesting, responding with, storing, and transmitting XACML
232 policies. The following types and elements are described:

- 233 1. `xacml-saml:XACMLPolicyStatementType` – A new SAML extension type defined in this
234 Profile that MAY be used in an XACML system to create XACMLPolicy Statements that hold
235 XACML policies for storage or transmission.
- 236 2. <saml:Statement> – A standard SAML element that MUST be used to contain instances of
237 the `xacml-saml:XACMLPolicyStatementType`. An instance of such a <saml:Statement>
238 element is called an XACMLPolicy Statement in this Profile.

- 239 3. `<saml:Assertion>` – A standard SAML element that MUST be used to hold XACMLPolicy
240 Statement instances in an XACML system, either in a repository or in an XACMLPolicy
241 Response. An instance of such a `<saml:Assertion>` element is called an XACMLPolicy
242 Assertion in this Profile.
- 243 4. `<xacml-samlp:XACMLPolicyQuery>` – A new SAML extension protocol element defined in
244 this Profile that MAY be used by a PDP or other application to request XACML policies from a
245 Policy Administration Point (PAP).
- 246 5. `<samlp:Response>` – A standard SAML protocol element that MUST be used to return
247 XACMLPolicy Assertions in response to an `<xacml-samlp:XACMLPolicyQuery>`. An
248 instance of such a `<samlp:Response>` element is called an XACMLPolicy Response in this
249 Profile.

250 Section 6 describes the use of XACMLAuthzDecision Assertion and XACMLPolicy Assertion instances
251 as advice in other SAML Assertions. The following element is described:

- 252 1. `<saml:Advice>` – A standard SAML element that MAY be used to convey XACMLPolicy
253 Assertions or XACMLAuthzDecision Assertions as advice in other `<saml:Assertion>`
254 instances.

255 Section 7 describes the use of XACMLAuthzDecision Assertions as authorization tokens in a SOAP
256 message exchange.

257 Section 8 describes recommended non-normative SAML metadata for use with these XACML-related
258 protocols.

259 Section 9 describes requirements for conformance with various aspects of this Profile.

260 **1.2 Diagram of SAML integration with XACML**

261 Figure 1 illustrates the XACML use model and the messages that can be used to communicate between
262 the various components. Not all components or messages will be used in every implementation. Not
263 shown, but described in this Profile, is the ability to use an XACMLPolicy Assertion or an
264 XACMLAuthzDecision Assertion in a `<saml:Advice>` instance.

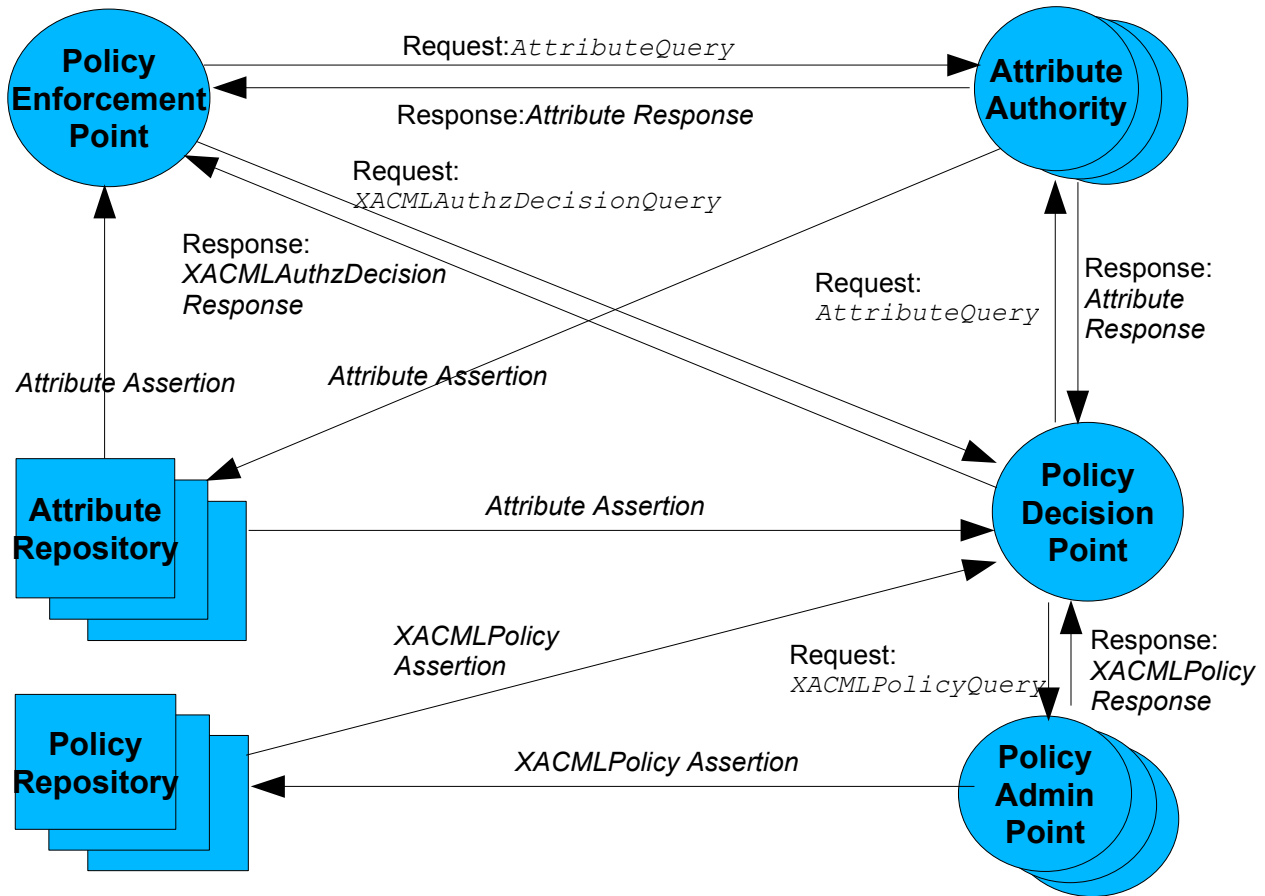


Figure 1: Components and messages in a integration of SAML with XACML

265 This Profile describes all these message elements, and describes how to use them, along with other
 266 aspects of using SAML with XACML.

267 1.3 Backwards compatibility

268 This Profile requires no changes or extensions to XACML, but does define extensions to SAML. The
 269 Profile may be used with XACML 1.0 , 1.1, 2.0, or 3.0. Separate versions of the Profile schemas are
 270 used with each version of XACML as described in Section 1.5.

271

272 1.4 Terminology

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

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

278 **Attribute** - In this Profile, the term "Attribute", when the initial letter is capitalized, may refer to either an
 279 XACML Attribute or to a SAML Attribute. The term will always be preceded with the type of Attribute
 280 intended.

- 281 • An XACML Attribute is a typed name/value pair, with other optional information, specified using an
282 `<xacml-context:Attribute>` instance. An XACML Attribute is associated with an entity or topic
283 identity by the XACML Attribute's position within a particular Attribute group in the XACML Request.
- 284 • A SAML Attribute is a name/value pair, with other optional information, specified using a
285 `<saml:Attribute>` instance. A SAML Attribute is associated with a particular subject by its
286 inclusion in a SAML Attribute Assertion that contains a `<saml:Subject>` instance. The SAML
287 Subject may correspond to any XACML Attribute group.

288 **Attribute group** – In this Profile, the term “Attribute group” is used to describe a collection of XACML
289 Attributes in an XACML Request Context that are associated with a particular entity. In XACML 1.0, 1.1,
290 and 2.0, there is a fixed number of such collections, called Subject Attributes, Resource Attributes,
291 Action Attributes, and Environment Attributes. In XACML 3.0, the number and identifiers of such
292 collections is extensible, but there are standard identifiers that correspond to the fixed collections defined
293 in previous versions of XACML.

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

297 **Attribute Assertion** – A `<saml:Assertion>` instance that contains a
298 `<saml:AttributeStatement>` instance.

299 **Attribute Response** – A `<samlp:Response>` instance that contains a SAML Attribute Assertion.

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

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

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

307 **policy** – A set of rules indicating the conditions under which an access is permitted or denied. XACML
308 has two different schema elements used for policies: `<xacml:Policy>` and `<xacml:PolicySet>`. An
309 `<xacml:PolicySet>` is a collection of other `<xacml:Policy>` and `<xacml:PolicySet>` elements.
310 An `<xacml:Policy>` contains actual access control rules.

311 **XACMLAuthzDecision Assertion** – A `<saml:Assertion>` instance that contains an
312 XACMLAuthzDecision Statement.

313 **XACMLAuthzDecision Response** – A `<samlp:Response>` instance that contains an
314 XACMLAuthzDecision Assertion.

315 **XACMLAuthzDecision Statement** – A `<saml:Statement>` instance that is of type `xacml-`
316 `saml:XACMLAuthzDecisionStatementType`.

317 **XACMLPolicy Assertion** – A `<saml:Assertion>` instance that contains an XACMLPolicy Statement.

318 **XACMLPolicy Response** – A `<samlp:Response>` instance that contains an XACMLPolicy Assertion.

319 **XACMLPolicy Statement** – A `<saml:Statement>` instance that is of type `xacml-`
320 `saml:XACMLPolicyStatementType`.

321 1.5 Namespaces

322 *Normative*

323 The following namespace prefixes are used in the schema fragments:

Prefix	Namespace
xacml	The XACML policy namespace.
xacml-context	The XACML context namespace.
xacml-saml	XACML extensions to the SAML 2.0 Assertion schema namespace.
xacml-samlp	XACML extensions to the SAML 2.0 Protocol schema namespace.
xacml-samlm	urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:metadata
saml	urn:oasis:names:tc:SAML:2.0:assertion
samlp	urn:oasis:names:tc:SAML:2.0:protocol
md	urn:oasis:names:tc:SAML:2.0:metadata
ds	http://www.w3.org/2000/09/xmldsig#
xsi	http://www.w3.org/2001/XMLSchema-instance
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd or http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.1.xsd

324 This Profile is written for use with XACML 1.0 [XACML1], 1.1 [XACML 1.1], 2.0 [XACML2], or 3.0
 325 [XACML3]. Depending on the version of XACML being used, the xacml, xacml-context, xacml-
 326 saml, and xacml-samlp namespace prefixes have the following values in the schemas:

327 XACML 1.0:

```
328 xacml="urn:oasis:names:tc:xacml:1.0:policy"
329 xacml-context="urn:oasis:names:tc:xacml:1.0:context"
330 xacml-saml=
331 "urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:assertion"
332 xacml-samlp=
333 "urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:protocol"
```

335 XACML 1.1:

```
336 xacml="urn:oasis:names:tc:xacml:1.0:policy"
337 xacml-context="urn:oasis:names:tc:xacml:1.0:context"
338 xacml-
339 saml="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:assertion"
340 xacml-
341 samlp="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:protocol"
```

343 XACML 2.0:

```
344 xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
345 xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
346 xacml-
347 saml="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion"
348 xacml-
349 samlp="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol"
```

351 XACML 3.0:
 352 xacml="urn:oasis:names:tc:xacml:3.0:schema:os"
 353 xacml-context="urn:oasis:names:tc:xacml:3.0:schema:os"

354 NOTE: XACML 3.0 uses a single schema for both policies and context.
 355 xacml-
 356 saml="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:assertion"
 357 xacml-
 358 sampl="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:protocol"

360 1.6 Normative References

- 361 **[ADMIN]** E. Rissanen, ed., *XACML v3.0 Administrative Policy Version 1.0*
- 362 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
 363 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 364 **[SAML]** S. Cantor, et al., eds., *Assertions and Protocols for the OASIS Security*
 365 *Assertion Markup Language (SAML) V2.0*, [http://www.oasis-](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security)
 366 [open.org/committees/documents.php?wg_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security).
- 367 **[SAML-PROFILE]** J. Hughes, et al., eds., *Profiles for the OASIS Security Assertion Markup*
 368 *Language (SAML) V2.0*, [http://www.oasis-](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security)
 369 [open.org/committees/documents.php?wg_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security).
- 370 **[XACML1]** OASIS *eXtensible Access Control Markup Language (XACML) Version 1.0*
- 371 **[XACML1.1]** OASIS *eXtensible Access Control Markup Language (XACML) Version 1.1*
- 372 **[XACML2]** T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML)*
 373 *Version 2.0*, OASIS Standard, 1 February 2005, [http://docs.oasis-](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf)
 374 [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).
- 375 **[XACML3]** E. Rissanen, ed., *OASIS eXtensible Access Control Markup Language*
 376 *(XACML) Version 3.0*
- 377 **[XACML-SAML]** OASIS, the schemas associated with namespace <xacml-saml> that are a
 378 normative part of this Profile.
- 379 **[XACML-SAMPL]** OASIS, the schemas associated with namespace <xacml-sampl> that are a
 380 normative part of this Profile.
- 381 **[WSS]** OASIS, *Web Services Security: SOAP Message Security 1.0 (WS-Security*
 382 *2004)*, OASIS Standard December 2004, and *WS-Security Core Specification*
 383 *1.1*, OASIS Standard February 2006, [http://www.oasis-](http://www.oasis-open.org/specs/index.php)
 384 [open.org/specs/index.php](http://www.oasis-open.org/specs/index.php).
- 385

386 1.7 Non-normative References

- 387 **[XACMLIntro]** S. Proctor, *A Brief Introduction to XACML*, [http://www.oasis-](http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html)
 388 [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
 389 March 2003.
- 390

391

2 Attributes

392 In an XACML system, PEPs and PDP Context Handlers often need to retrieve attributes from on-line
393 Attribute Authorities or from Attribute Repositories. SAML provides assertion and protocol elements that
394 MAY be used for retrieval of attributes for use in an XACML Request Context. These elements include a
395 `<saml:Attribute>` element for expressing a named attribute value, a
396 `<saml:AttributeStatement>` for holding a collection of `<saml:Attribute>` elements, and a
397 `<saml:Assertion>` element that can hold various kinds of statements, including a
398 `<saml:AttributeStatement>`. A `<saml:Assertion>` instance containing a
399 `<saml:AttributeStatement>` is called a SAML Attribute Assertion in this Profile. A SAML Attribute
400 Assertion includes the name of the attribute issuer, an optional digital signature for authenticating the
401 attribute, an optional subject identity to which the attribute is bound, and optional conditions for use of the
402 assertion that may include a validity period during which the attribute is to be considered valid. Such an
403 assertion is suitable for storing attributes in an Attribute Repository, for transmitting attributes between an
404 Attribute Authority and an Attribute Repository, and for transmitting attributes between an Attribute
405 Repository and a PEP or XACML Context Handler. For querying an on-line Attribute Authority for
406 attributes, and for holding the response to that query, SAML defines `<samlp:AttributeQuery>` and
407 `<samlp:Response>` elements. In this Profile, an instance of such a `<samlp:Response>` element is
408 called a SAML Attribute Response. This Section describes the use of these SAML elements in an
409 XACML system.

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

2.1 Element `<saml:Attribute>`

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

416 In order to be used in an XACML Request Context, each `<saml:Attribute>` instance MUST comply
417 with the *SAML XACML Attribute Profile*, associated with namespace
418 `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`, in Section 8.5 of the *Profiles for*
419 *the OASIS Security Assertion Markup Language (SAML 2.0)* [SAML-PROFILE].

2.1.1 Mapping a `<saml:Attribute>` to an `<xacml-context:Attribute>`

421 An `<xacml-context:Attribute>` instance MUST be constructed from the corresponding
422 `<saml:Attribute>` instance contained in a SAML Attribute Assertion as follows. An XACML
423 implementation is NOT REQUIRED to instantiate the `<xacml-context:Attribute>` instances
424 physically so long as the XACML PDP can obtain values for the XACML Attributes as if they had been
425 instantiated in this way.

- 426 • XACML `AttributeId` XML attribute

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

- 428 • XACML `DataType` XML attribute

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

432 • XACML Issuer XML attribute

433 The string value of the `<saml:Issuer>` instance from the SAML Attribute Assertion MUST be used.

434 • `<xacml-context:AttributeValue>`

435 The `<saml:AttributeValue>` value MUST be used as the value of the `<xacml-`
436 `context:AttributeValue>` instance.

437 Each `<saml:Attribute>` instance MUST be mapped to no more than one `<xacml-`
438 `context:Attribute>` instance. Not all `<saml:Attribute>` instances in a SAML Attribute Assertion
439 need to be mapped; a subset of `<saml:Attribute>` instances MAY be selected by a mechanism not
440 specified in this Profile. The Issuer of the SAML Attribute Assertion MUST be used as the Issuer for
441 each `<xacml-context:Attribute>` instance that is created from `<saml:Attribute>` instances in
442 that SAML Attribute Assertion.

443 The `<xacml-context:Attribute>` created from the SAML Attribute Assertion MUST be placed into
444 the Attribute group of the XACML Request Context that corresponds to the entity that is represented by
445 the `<saml:Subject>` in the SAML Attribute Assertion.

446 *Non-normative Example:* For example, if the SAML Attribute Assertion `<saml:Subject>` contains
447 a `<saml:NameIdentifier>` instance, and the value of that `NameIdentifier` matches the value
448 of the `<xacml-context:Attribute>` having an `AttributeId` of
449 `urn:oasis:names:tc:xacml:1.0:resource:resource-id`, then `<xacml-`
450 `context:Attribute>` instances created from `<saml:Attribute>` instances in that SAML
451 Attribute Assertion MUST be placed into the `<xacml-context:Resource>` Attribute group or its
452 corresponding XACML 3.0 Attribute group.

453 If a mapped `<saml:Attribute>` is placed into an `<xacml-context:Subject>` instance, then the
454 XACML `SubjectCategory` XML attribute MUST also be consistent with the conceptual “subject
455 category” of the entity that corresponds to the `<saml:Subject>` of the SAML Attribute Assertion that
456 contained the `<saml:Attribute>`. The `<saml:Subject>` itself is NOT translated into an `<xacml-`
457 `context:Attribute>` as part of processing a SAML Attribute Assertion; the `<saml:Subject>`
458 identity is used only to determine the Attribute group in the XACML Request Context to which the
459 `<saml:Attribute>` values should be added.

460 The mapping MUST be done in such a way that the semantics defined by SAML for the elements in a
461 SAML Attribute Assertion have been adhered to. The mapping entity need not perform these semantic
462 checks itself, but the system in which it operates MUST be such that the checks have been done before
463 any `<xacml:Attribute>` created from a SAML Attribute Assertion is used by an XACML PDP. These
464 semantic checks include, but are not limited to the following.

- 465 • Any `NotBefore` and `NotOnOrAfter` XML attributes in the SAML Attribute Assertion MUST be valid
466 with respect to the `<xacml:Request>` in which the SAML-derived `<xacml:Attribute>` is used.
467 This means that the XACML Attributes associated with the following `AttributeId` values in the
468 `<xacml:Request>` MUST represent times and dates that are not before the `NotBefore` XML
469 attribute value and not on or after the `NotOnOrAfter` XML attribute value:

470 `urn:oasis:names:tc:xacml:1.0:environment:current-time`

471 `urn:oasis:names:tc:xacml:1.0:environment:current-date`

472 `urn:oasis:names:tc:xacml:1.0:environment:current-dateTime`

473 The time period during which SAML Attribute Assertions are considered valid in XACML 3.0 depends
474 on whether the PDP is configured to retrieve XACML Attributes that were valid at the time a policy
475 was issued or at the time the policy is being evaluated.

- 476 • The semantics defined by SAML for any `<saml:AudienceRestrictionCondition>` or
477 `<saml:DoNotCacheCondition>` elements MUST be adhered to.

478 **2.2 Element `<saml:AttributeStatement>`**

479 When a `<saml:Attribute>` instance is stored or transmitted in an XACML system, the instance MUST
480 be enclosed in a standard SAML `<saml:AttributeStatement>`. The definition and use of the
481 `<saml:AttributeStatement>` element MUST be as described in the SAML 2.0 standard [SAML].

482 **2.3 Element `<saml:Assertion>`: SAML Attribute Assertion**

483 When a `<saml:AttributeStatement>` instance is stored or transmitted in an XACML system, the
484 instance MUST be enclosed in a `<saml:Assertion>`. An instance of such a `<saml:Assertion>`
485 element is called a SAML Attribute Assertion in this Profile.

486 When used as a SAML Attribute Assertion in an XACML system, the definition and use of the
487 `<saml:Assertion>` element MUST be as specified in the SAML 2.0 standard, augmented with the
488 following requirements. Except as specified here, this Profile imposes no requirements or restrictions on
489 the SAML Attribute Assertion element and its contents beyond those specified in SAML 2.0.

490 `<saml:Issuer>` [Required]

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

493 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
494 in the `<saml:Issuer>` element refer to the entity that signs the SAML Attribute Assertion.. It is up
495 to the relying party to determine whether it has an appropriate trust relationship with the authority
496 that signs the SAML Attribute Assertion.

497 When a SAML Attribute Assertion containing a `<saml:Attribute>` is used to construct an
498 `<xacml-context:Attribute>`, the string value of the `<saml:Issuer>` instance MUST be used
499 as the value of the `<xacml-context:Attribute>` Issuer XML attribute, so the
500 `<saml:Issuer>` value SHOULD be specified with this in mind.

501 `<ds:Signature>` [Optional]

502 The `<ds:Signature>` element is an optional element for holding “An XML Signature that
503 authenticates the assertion, as described in Section 5 of the SAML 2.0 specification [SAML].”

504 A `<ds:Signature>` instance MAY be used in a SAML Attribute Assertion. In order to support 3rd
505 party digital signatures, this Profile does NOT require that the identity provided in the
506 `<saml:Issuer>` instance refer to the entity that signs the SAML Attribute Assertion. It is up to the
507 relying party to determine whether it has an appropriate trust relationship with the authority that signs
508 the SAML Attribute Assertion.

509 A relying party SHOULD verify any signature included in the SAML Attribute Assertion and SHOULD
510 NOT use information derived from the SAML Attribute Assertion unless the signature is verified
511 successfully.

512 `<saml:Subject>` [Optional]

513 The `<saml:Subject>` element is an optional element used for holding “The subject of the
514 statement(s) in the assertion” [SAML]. Each SAML Attribute Assertion used in an XACML system
515 MUST contain a `<saml:Subject>` element.

516 In a SAML Attribute Assertion containing a `<saml:Attribute>` that is to be mapped to an
517 `<xacml-context:Attribute>`, the `<saml:Subject>` instance MUST contain the identity of the
518 entity to which the `<saml:Attribute>` and its value are bound. For a mapped
519 `<saml:Attribute>` to be placed in a given XACML Attribute group, this identity SHOULD refer to
520 the same entity as any XACML Attribute that serves as an entity identifier in the Attribute group. For
521 example, the `<saml:Subject>` associated with a mapped SAML->XACML Attribute to be
522 placed in the XACML `<xacml-context:Resource>` Attribute group SHOULD refer to the same
523 entity as the value of any XACML Attribute having an `AttributeId` of
524 `urn:oasis:names:tc:xacml:1.0:resource:resource-id` that occurs in the same `<xacml-`
525 `context:Resource>` instance. See Section 2.1 for more information.

526 `<saml:Conditions>` [Optional]

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

529 The `<saml:Conditions>` instance SHOULD contain `NotBefore` and `NotOnOrAfter` XML
530 attributes to specify the limits on the validity of the SAML Attribute Assertion. If these XML attributes
531 are present, the relying party SHOULD ensure that an `<xacml-context:Attribute>` derived
532 from the SAML Attribute Assertion is used by a PDP for evaluating policies only when the value of
533 the `<xacml-context:Attribute>` in the XACML Request Context having an `AttributeId` of
534 `urn:oasis:names:tc:xacml:1.0:environment:current-dateTime` is contained within the
535 SAML Attribute Assertion's specified validity period. The time period during which SAML Attribute
536 Assertions are considered valid in XACML 3.0 depends on whether the PDP is configured to retrieve
537 XACML Attributes that were valid at the time a policy was issued or at the time the policy is being
538 evaluated.

539 **2.4 Element `<samlp:AttributeQuery>`**

540 The standard SAML `<samlp:AttributeQuery>` element MAY be used in an XACML system by a
541 PEP or XACML Context Handler to request SAML Attribute Assertions from an on-line Attribute Authority
542 for use in an XACML Request Context. The definition and use of the `<samlp:AttributeQuery>`
543 element MUST be as described in the SAML 2.0 standard [SAML].

544 Note that the SAML-defined `ID` XML attribute is a required component of a
545 `<samlp:AttributeQuery>` and can be used to correlate the `<samlp:AttributeQuery>` with the
546 corresponding SAML Attribute Response.

547 **2.5 Element `<samlp:Response>`: SAML Attribute Response**

548 The response to a `<samlp:AttributeQuery>` MUST be a `<samlp:Response>` instance containing a
549 SAML Attribute Assertion that holds any `<saml:AttributeStatement>` instances that match the
550 query. An instance of such a `<samlp:Response>` element is called a SAML Attribute Response in this
551 Profile. The definition and use of the SAML Attribute Response MUST be as described in the SAML 2.0
552 standard, augmented with the following requirements. Except as specified here, this Profile imposes no
553 requirements or restrictions on the SAML Attribute Response and its contents beyond those specified in
554 SAML 2.0.

555 `<saml:Issuer>` [Optional]

556 The `<saml:Issuer>` element is an optional element that “Identifies the entity that generated the
557 response message” [SAML].

558 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
559 in the `<saml:Issuer>` element refer to the entity that signs the SAML Attribute Response. It is up
560 to the relying party to determine whether it has an appropriate trust relationship with the authority
561 that signs the SAML Attribute Response.

562 `<ds:Signature>` [Optional]

563 The `<ds:Signature>` element is an optional element for holding “An XML Signature that
564 authenticates the responder and provides message integrity” [SAML].

565 A `<ds:Signature>` instance MAY be used in a Attribute Response. In order to support 3rd party
566 digital signatures, this Profile does NOT require that the identity provided in the `<saml:Issuer>`
567 refer to the entity that signs the SAML Attribute Response. It is up to the relying party to determine
568 whether it has an appropriate trust relationship with the authority that signs the SAML Attribute
569 Response .

570 A relying party SHOULD verify any signature included in the SAML Attribute Response and
571 SHOULD NOT use information derived from the SAML Attribute Response unless the signature is
572 verified successfully.

573

3 Conveying XACML Attributes in a SOAP Message

574 At the time a Web Service is invoked, the service MAY need to determine whether the client is
575 authorized to invoke the service or to access resources that are involved in the service invocation. A
576 Web service MAY use an XACML PDP to make such an authorization decision.

577 When a service evaluates an XACML authorization, access control, or privacy policy related to a SOAP
578 message, it MAY obtain the XACML Attributes required for the evaluation from various sources, including
579 databases, registries, trusted Attribute Authorities, and so on. This work is done in the application-
580 dependent XACML Context Handler that provides XACML Attributes to the PDP on request. A Web
581 Services client or intermediary MAY include XACML `<xacml-context:Attribute>` instances in a
582 `wsse:Security` SOAP Header for use by this Context Handler. This Section of this Profile describes
583 two ways in which such `<xacml-context:Attribute>` instances MAY be provided.

3.1 `<xacml-samlp:XACMLAuthzDecisionQuery>`

584
585 The first way in which XACML Attributes MAY be provided to a service is by including an instance of the
586 `<xacml-samlp:XACMLAuthzDecisionQuery>` (see Section 4.4) in the `wsse:Security` Header of a
587 SOAP message. This query contains an XACML Request Context that SHOULD contain `<xacml-`
588 `context:Attribute>` instances related to any resource access that the client will need in order to
589 interact successfully with the service. The `<xacml-samlp:XACMLAuthzDecisionQuery>` SHOULD
590 be signed by an entity that the Web Service trusts to authenticate the enclosed `<xacml-`
591 `context:Attribute>` instances.

592 The Web Service MAY provide the `<xacml-context:Attribute>` instances in such an `<xacml-`
593 `samlp:XACMLAuthzDecisionQuery>` to an XACML PDP as part of evaluating XACML policies related
594 to the Web Service interaction. The service SHOULD verify that the query is signed by an entity that the
595 service trusts to authenticate the enclosed `<xacml-context:Attribute>` instances. It SHOULD
596 verify that the `IssueInstant` of the `<xacml-samlp:XACMLAuthzDecisionQuery>` is close enough
597 the the current time to meet the validity requirements of the service.

3.2 SAML Attribute Assertion

598
599 A second way in which XACML Attributes MAY be provided to a service is in the form of a SAML
600 Attribute Assertion in the `wsse:Security` Header of a SOAP message. The SAML Attributes contained in
601 the SAML Attribute Assertion MAY be converted to XACML Attributes as described in Section 2.1 of this
602 Profile by an XACML Context Handler for use by a PDP associated with the Web Service in evaluating
603 XACML policies related to the Web Service interaction.

604

4 Authorization Decisions

605 XACML defines `<xacml-context:Request>` and `<xacml-context:Response>` elements for
606 describing an authorization decision request and the corresponding response from a PDP. In many
607 environments, instances of these elements need to be signed or associated with a validity period in order
608 to be used in an actual protocol between entities. Although SAML 2.0 defines a rudimentary
609 `<samlp:AuthzDecisionQuery>` in the SAML Protocol Schema and a rudimentary
610 `<saml:AuthzDecisionStatement>` in the SAML Assertion Schema, these elements are not able to
611 convey all the information that an XACML PDP is capable of accepting as part of its Request Context or
612 conveying as part of its XACML Response Context. In order to allow a PEP to use the SAML protocol
613 with full support for the XACML Request Context and XACML Response Context syntax, this Profile
614 defines one SAML extension type and one SAML extension element, and describes how they are used
615 with other standard SAML elements.

- 616 • `<xacml-saml:XACMLAuthzDecisionStatementType>` is a new SAML extension type that
617 includes an XACML `<xacml-context:Response>` along with other optional information.
- 618 • A `<saml:Statement>` of type `<xacml-saml:XACMLAuthzDecisionStatementType>` (defined
619 using `xsi:type`) MAY be used by a PDP Context Handler to convey an XACML `<xacml-
620 context:Response>` along with other optional information. An instance of such a
621 `<saml:Statement>` element is called an XACMLAuthzDecision Statement in this Profile.
- 622 • A `<saml:Assertion>` MUST be used to hold XACMLAuthzDecision Statements. An instance of
623 such a `<saml:Assertion>` element is called an XACMLAuthzDecision Assertion in this Profile.
- 624 • A `<xacml-samlp:XACMLAuthzDecisionQuery>` is a new SAML extension element that MAY be
625 used by a PEP to submit an XACML Request Context, along with other optional information, as a
626 SAML protocol query to an XACML Context Handler.
- 627 • A `<samlp:Response>` containing an XACMLAuthzDecision Assertion MUST be used by an XACML
628 Context Handler as the response to an `<saml-samlp:XACMLAuthzDecisionQuery>`. An instance
629 of such a `<samlp:Response>` element is called an XACMLAuthzDecision Response in this Profile.

630 This Section defines and describes the usage of these types and elements.. The schemas for the new
631 type and element are contained in the [XACML-SAML] and [XACML-SAML] schema documents.

632 4.1 Type `<xacml-saml:XACMLAuthzDecisionStatementType>`

633 The new `<xacml-saml:XACMLAuthzDecisionStatementType>` complex type contains an XACML
634 Response Context along with related information. Use of this type is an alternative to use of the SAML-
635 defined `<saml:AuthzDecisionStatementType>`; this alternative allows an XACML Context Handler
636 to use SAML with full support for XACML authorization decisions. An instance of a
637 `<saml:Statement>` element that is of this type (defined using `xsi:type="xacml-
638 saml:XACMLAuthzDecisionStatementType"`) is called an XACMLAuthzDecision Statement in this
639 Profile.

```

<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>

```

640 The `<xacml-saml:XACMLAuthzDecisionStatementType>` complex type is an extension to the
 641 SAML-defined `<saml:StatementAbstractType>`. It contains the following elements:

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

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

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

646 An `<xacml-context:Request>` element containing `<xacml-context:Attribute>` instances
 647 that were used by the XACML PDP in evaluating policies to obtain the corresponding `<xacml-`
 648 `context:Response>`.

649 If the XACMLAuthzDecision Statement represents a response to an `<xacml-`
 650 `samlp:XACMLAuthzDecisionQuery>`, and if the ReturnContext XML attribute in the `<xacml-`
 651 `samlp:XACMLAuthzDecisionQuery>` instance is "true", then this element MUST be included; if
 652 the ReturnContext XML attribute in the `<xacml-samlp:XACMLAuthzDecisionQuery>`
 653 instance is "false", then this element MUST NOT be included. See the description of the
 654 ReturnContext XML attribute in Section 4.4 for a specification of the `<xacml-`
 655 `context:Attribute>` instances that MUST be returned in this element when it is part of a
 656 response to an `<xacml-samlp:XACMLAuthzDecisionQuery>`.

657 If the XACMLAuthzDecision Statement does not represent the response to an `<xacml-`
 658 `samlp:XACMLAuthzDecisionQuery>`, then this element MAY be included. In this case, the PDP
 659 MUST determine which `<xacml-context:Attribute>` instances are included using criteria that
 660 are outside the scope of this Profile.

661 4.2 Element `<saml:Statement>`: XACMLAuthzDecision Statement

662 A `<saml:Statement>` instance MAY be of type `<xacml-`
 663 `saml:XACMLAuthzDecisionStatementType>` by using `xsi:type` as shown in the example in
 664 Section 4.3. An instance of a `<saml:Statement>` element that is of type `<xacml-`
 665 `saml:XACMLAuthzDecisionStatementType>` is called an XACMLAuthzDecision Statement in this
 666 Profile. Any instance of an XACMLAuthzDecision Statement in an XACML system MUST be enclosed in
 667 a `<saml:Assertion>`.

668 4.3 Element `<saml:Assertion>`: XACMLAuthzDecision Assertion

669 A `<saml:Assertion>` instance MAY contain an XACMLAuthzDecision Statement as shown in the
 670 following non-normative example:

```

<saml:Assertion Version="2.0" ID="9812368"
  IssueInstant="2006-05-31T13:20:00.000">
  <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
  <saml:Statement
    xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
    <xacml-context:Response>
      <xacml-context:Result>
        <xacml-context:Decision>
          NotApplicable
        </xacml-context:Decision>
      </xacml-context:Result>
    </xacml-context:Response>
    <xacml-context:Request>
      . . . .
    </xacml-context:Request>
  </saml:Statement>
</saml:Assertion>

```

671 An instance of a `<saml:Assertion>` element containing an XACMLAuthzDecision Statement is called
 672 an XACMLAuthzDecision Assertion in this Profile.

673 This Profile imposes the following requirements and restrictions on the `<saml:Assertion>` element
 674 beyond those specified in SAML 2.0 when used as an XACMLAuthzDecision Assertion.

675 `<saml:Issuer>` [Required]

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

678 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
 679 in the `<saml:Issuer>` element refer to the entity that signs the XACMLAuthzDecision Assertion. It
 680 is up to the relying party to determine whether it has an appropriate trust relationship with the
 681 authority that signs the XACMLAuthzDecision Assertion.

682 `<ds:Signature>` [Optional]

683 The `<ds:Signature>` element is an optional element for holding “An XML Signature that
 684 authenticates the assertion, as described in Section 5 of the SAML 2.0 core specification [SAML].”

685 A `<ds:Signature>` instance MAY be used in a `<saml:Assertion>`. In order to support 3rd party
 686 digital signatures, this Profile does NOT require that the identity provided in the `<saml:Issuer>`
 687 instance refer to the entity that signs the XACMLAuthzDecision Assertion. It is up to the relying party
 688 to determine whether it has an appropriate trust relationship with the authority that signs the
 689 Assertion .

690 A relying party SHOULD verify any signature included in the XACMLAuthzDecision Assertion and
 691 SHOULD NOT use information derived from the Assertion unless the signature is verified
 692 successfully.

693 `<saml:Subject>` [Optional]

694 The `<saml:Subject>` element MUST NOT be included in an XACMLAuthzDecision Assertion.
 695 Instead, the Subject of an XACMLAuthzDecision Assertion is specified in the XACML Request
 696 Context of the corresponding authorization decision request. This corresponding XACML Request
 697 Context MAY be included in the XACMLAuthzDecision Statement as described in Section 4.1.

698 `<saml:Conditions>` [Optional]

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

701 The <saml:Conditions> instance SHOULD contain NotBefore and NotOnOrAfter XML
702 attributes to specify the limits on the validity of the XACMLAuthzDecision Assertion. If these XML
703 attributes are present, the relying party SHOULD ensure that an <xacml-context:Response>
704 taken from the XACMLAuthzDecision Assertion is used only during the Assertion's specified validity
705 period.

706 **4.4 Element <xacml-samlp:XACMLAuthzDecisionQuery>**

707 The <xacml-samlp:XACMLAuthzDecisionQuery> protocol element MAY be used by a PEP to
708 request an authorization decision from an XACML PDP. This element is an alternative to the SAML-
709 defined <samlp:AuthzDecisionQuery>; this alternative allows the PEP to use the full capabilities of
710 an XACML PDP. It allows use of the SAML query protocol to convey an XACML Request Context along
711 with related 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-samlp:AdditionalAttributes"
minOccurs="0" maxOccurs="1"/>
        <element ref="xacml:Policy"
minOccurs="0" maxOccurs="unbounded" />
        <element ref="xacml:PolicySet"
minOccurs="0" maxOccurs="unbounded" />
        <element ref="xacml-saml:ReferencedPolicies"
minOccurs="0" maxOccurs="1" />
      </sequence>
      <attribute name="InputContextOnly"
type="boolean"
use="optional"
default="false"/>
      <attribute name="ReturnContext"
type="boolean"
use="optional"
default="false"/>
      <attribute name="CombinePolicies"
type="boolean"
use="optional"
default="true"/>
    </extension>
  </complexContent>
</complexType>

```

712 The `<xacml-samlp:XACMLAuthzDecisionQuery>` element is of `<xacml-`
713 `samlp:XACMLAuthzDecisionQueryType>` complex type, which is an extension to the SAML-defined
714 `<samlp:RequestAbstractType>`.

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

717 `InputContextOnly` [Default "false"]

718 This XML attribute governs the sources of information that the PDP is allowed to use in making its
719 authorization decision. If the value of this XML attribute is "true", then the authorization decision
720 MUST be made solely on the basis of information contained in the `<xacml-`
721 `samlp:XACMLAuthzDecisionQuery>`; external XACML Attributes MUST NOT be used. If the
722 value of this XML attribute is "false", then the authorization decision MAY be made on the basis of
723 XACML Attributes not contained in the `<xacml-samlp:XACMLAuthzDecisionQuery>`.

724 `ReturnContext` [Default "false"]

725 This XML attribute allows the PEP to request that an `<xacml-context:Request>` instance be
726 included in the XACMLAuthzDecision Statement resulting from the query. It also governs the
727 contents of that `<xacml-context:Request>` instance.

728 If the value of this XML attribute is "true", then the PDP MUST include an `<xacml-`
729 `context:Request>` instance in the XACMLAuthzDecision Statement in the XACMLAuthzDecision

730 Response. This `<xacml-context:Request>` instance MUST include all those attributes supplied
731 by the PEP in the `<xacml-sampl:XACMLAuthzDecisionQuery>` that were used in making the
732 authorization decision. The PDP MAY include additional attributes in this `<xacml-
733 context:Request>` instance, such as external attributes obtained by the PDP and used in making
734 the authorization decision, or other attributes known by the PDP that may be useful to the PEP in
735 making subsequent authorization decision queries.

736 If this XML attribute is “false”, then the PDP MUST NOT include an `<xacml-context:Request>`
737 instance in the XACMLAuthzDecision Statement in the XACMLAuthzDecision Response.

738 `CombinePolicies` [Default “true”]

739 This XML attribute allows the PEP to specify whether policies supplied in `<xacml:Policy>` and
740 `<xacml:PolicySet>` elements of the `<xacml-sampl:XACMLAuthzDecisionQuery>` are to be
741 combined with other policies available to the PDP during evaluation.

742 If the attribute value is “true”, then the PDP MUST insert all policies passed in the `<xacml-
743 sampl:XACMLAuthzDecisionQuery>` into the set of policies or policy sets that define the PDP as
744 specified in Section 7.13 of the XACML 2.0 core specification [XACML2]. They MUST be combined
745 with the other policies using the policy combining algorithm that defines the PDP as specified in
746 Section 7.13 of the XACML 2.0 core specification [XACML2]. If the policy combining algorithm that
747 defines the PDP is one in which element order is considered, then the policies passed in the
748 XACMLAuthzDecision Query MUST be considered in the order in which they appear in the `<xacml-
749 sampl:XACMLAuthzDecisionQuery>` and MUST be considered as following all other policies that
750 define the PDP.

751 *TBD: Issue#72 describes a problem in combining policies passed in this way in connection with*
752 *XACML 3.0 policy reduction.*

753 If the attribute value is “false”, then there MUST be no more than one `<xacml:Policy>` or
754 `<xacml:PolicySet>` passed in the `<xacml-sampl:XACMLAuthzDecisionQuery>`. This policy
755 MUST be treated as the policy that defines the PDP as specified in Section 7.13 of the XACML 2.0
756 core specification [XACML2] for evaluation of the `<xacml-context:Request>` passed in the
757 `<xacml-sampl:XACMLAuthzDecisionQuery>`. It MUST NOT be used to evaluate any other
758 `<xacml-context:Request>` instances unless provided to the PDP independent of the particular
759 `<xacml-context:Request>`.

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

761 An XACML Request Context that is to be evaluated.

762 `<xacml-sampl:AdditionalAttributes>` [Zero or One]

763 Entity descriptions and corresponding `<xacml-context:Attribute>` instances that apply to
764 them. This element is used only with XACML 3.0 Administrative Policy [ADMIN] functionality.

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

766 Optional XACML Policy instances that MUST be used only for evaluating this authorization decision
767 request.

768 If the `CombinePolicies` XML attribute is “true”, then the PDP MAY choose to use such XACML
769 Policy instances.

770 If the `CombinePolicies` XML attribute is “false”, then the PDP MUST use this XACML Policy
771 instance. There MUST be only one such XACML Policy instance and there MUST NOT be any
772 XACML PolicySet instances in this `<xacml-sampl:XACMLAuthzDecisionQuery>` instance.

773 <xacml:PolicySet> [Any Number]

774 Optional XACML PolicySet instances that MUST be used only for evaluating this authorization
775 decision request.

776 If the CombinePolicies XML attribute is "true", then the PDP MAY choose to use such XACML
777 PolicySet instances.

778 If the CombinePolicies XML attribute is "false", then the PDP MUST use this XACML PolicySet
779 instance. There MUST be only one such XACML PolicySet instance and there MUST NOT be any
780 XACML Policy instances in this XACMLAuthzDecision Query.

781 <xacml-saml:ReferencedPolicies> [Zero or One]

782 With the exception of XACML Policy and PolicySet instances that the receiver of the
783 XACMLAuthzDecision Statement is not authorized to view, this element MUST contain all XACML
784 Policy and PolicySet instances required to resolve all <xacml:PolicySetIdReference> or
785 <xacml:PolicyIdReference> instances contained in the XACMLAuthzDecision Statement,
786 including those in the <xacml-saml:ReferencedPolicies> instance itself. The values of the
787 PolicyId and PolicySetId XML attributes of the policies included in the <xacml-
788 saml:ReferencedPolicies> instance MUST exactly match the values contained in the
789 corresponding <xacml:PolicySetIdReference> or <xacml:PolicyIdReference>
790 instances.

791 4.5 Element <xacml-samlp:AdditionalAttributes>

792 This element applies only for use with XACML 3.0 Administrative Policy [ADMIN], and requires an
793 XACML 3.0 PDP.

794 In some cases it may be useful for the PEP to provide attributes for delegates with the authorization
795 decision request. Since the Request Contexts used in reduction are not formed until after the access
796 request is submitted to the PDP, the delegate attributes need to be treated differently from the attributes
797 part of the access **Request Context**. The following defines elements that MAY be used to submit
798 XACML Attributes for this purpose. The XACML Attributes MUST be made available by the Context
799 Handler when the reduction Request Contexts are created.

```
800 <element name="AdditionalAttributes"  
801   type="xacml-samlp: AdditionalAttributesType"/>  
802 <complexType name="AdditionalAttributesType">  
803   <sequence>  
804     <element ref="xacml-samlp:AssignedAttributes" minOccurs="0"  
805     maxOccurs="unbounded"/>  
806   </sequence>  
807 </complexType>
```

808 The <AdditionalAttributes> element is of AdditionalAttributesType complex type.

809 The <AdditionalAttributes> element contains the following elements:

810 <AssignedAttributes> [Required]

811 Assignment of a set of XACML Attributes to specified delegate entities.

812 **4.6 Element <xacml-samlp:AssignedAttributes>**

813 This element is used only with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML 3.0
814 PDP.

815 The <AssignedAttributes> element MUST contain XACML Attributes that apply to delegate entities
816 identified by the <xacml-samlp:HolderAttributes> element.

```
817 <element name="AssignedAttributes" type="xacml-  
818 samlp:AssignedAttributesType"/>  
819 <complexType name="AssignedAttributesType">  
820 <sequence>  
821 <element ref="xacml-samlp:HolderAttributes"/>  
822 <element ref="xacml-samlp:HolderAttributes"/>  
823 </sequence>  
824 </complexType>
```

825 The <AssignedAttributes> element is of AssignedAttributesType complex type.

826 The <AssignedAttributes> element contains the following elements:

827 <xacml-samlp:HolderAttributes> [Required]

828 The identities of the delegate entities to which the provided XACML Attributes apply.

829 <xacml-samlp:HolderAttributes> [Required]

830 The XACML Attributes of the delegate entity.

831 **4.7 Element <xacml-samlp:HolderAttributes>**

832 This element is used only with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML 3.0
833 PDP.

834 The <HolderAttributes> element MUST identify the delegate entities to which the provided <xacml-
835 samlp:HolderAttributes> elements apply.

```
836 <element name="HolderAttributes" type="xacml-samlp:HolderAttributesType"/>  
837 <complexType name="HolderAttributesType">  
838 <sequence>  
839 <element ref="xacml:Match" maxOccurs="unbounded"/>  
840 </sequence>  
841 </complexType>
```

842 The <xacml-samlp:HolderAttributes> element is of <xacml-samlp:HolderAttributesType> complex type.

843 The <xacml-samlp:HolderAttributes> element contains the following elements:

844 <xacml:Match> [One to many, required]

845 Matches the delegate entities to which the XACML Attributes in the associated <xacml-
846 samlp:HolderAttributes> element apply.

847 *TBD: the details of the <HolderAttributes> element are not specified yet since the core schema is in the*
848 *process of being rewritten.*

849 4.8 Element `<xacml-samlp:HolderAttributes>`

850 This element is used only with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML 3.0
851 PDP.

852 The `<xacml-samlp:HolderAttributes>` element MUST contain XACML Attributes that apply to the
853 delegate entities identified in the corresponding `<xacml-samlp:HolderAttributes>` element.

```
854 <element name="HolderAttributes" type="xacml-samlp:HolderAttributesType"/>  
855 <complexType name="HolderAttributesType">  
856   <sequence>  
857     <element ref="xacml-context:Attribute"  
858       minOccurs="0" maxOccurs="unbounded"/>  
859   </sequence>  
860 </complexType>
```

861 The `<xacml-samlp:HolderAttributes>` element is of `<xacml-samlp:HolderAttributesType>`
862 complex type.

863 The `<xacml-samlp:HolderAttributes>` element contains the following elements:

864 `<xacml-context:Attribute>` [any number]

865 An XACML Attribute of the delegate entities identified in the corresponding `<xacml-`
866 `samlp:HolderAttributes>` element.

867 4.9 Element `<xacml-saml:ReferencedPolicies>`

868 An instance of this element MUST be used to contain copies of all policies referenced from
869 `<xacml:Policy>` or `<xacml:PolicySet>` instances included in an XACMLAuthzDecision Statement
870 or in an XACMLPolicy Statement, as well as copies of all policies referenced from other policies included
871 in the `<xacml-saml:ReferencedPolicies>` instance..

```
872 <element name="ReferencedPolicies"  
873   type="xacml-saml:ReferencedPoliciesType"/>  
874 <complexType name="ReferencedPoliciesType">  
875   <sequence>  
876     <choice minOccurs="0" maxOccurs="unbounded">  
877       <element ref="xacml:Policy"/>  
878       <element ref="xacml:PolicySet"/>  
879     </choice>  
880   </sequence>  
881 </complexType>
```

882 The `<xacml-saml:ReferencedPolicies>` element is of `<xacml-`
883 `saml:ReferencedPoliciesType>` complex type.

884 The `<xacml-saml:ReferencedPolicies>` element contains the following elements:

885 `<xacml:Policy>` [any number]

886 A single `<xacml:Policy>` that is referenced using an `<xacml:PolicyIdReference>` from
887 another `<xacml:Policy>` or `<xacml:PolicySet>` instance included in an XACMLAuthzDecision
888 Statement or XACMLPolicy Statement. The value of the `PolicyId` XML attribute in the
889 `<xacml:Policy>` MUST be equal to the value of the corresponding
890 `<xacml:PolicyIdReference>` element.

891 `<xacml:PolicySet>` [any number]

892 A single `<xacml:PolicySet>` that is referenced using an `<xacml:PolicySetIdReference>`
893 from another `<xacml:Policy>` or `<xacml:PolicySet>` instance included in an
894 XACMLAuthzDecision Statement or XACMLPolicy Statement. The value of the `PolicySetId` XML
895 attribute in the `<xacml:PolicySet>` MUST be equal to the value of the corresponding
896 `<xacml:PolicySetIdReference>` element.

897 **4.10 Element `<samlp:Response>`: XACMLAuthzDecision Response**

898 A `<samlp:Response>` instance MAY contain an XACMLAuthzDecision Assertion as shown in the
899 following non-normative example:

```
<samlp:Response Version="2.0" ID="9812368"
  IssueInstant="2006-05-31T13:20:00.000">
  <saml:Assertion Version="2.0" ID="9812368"
    IssueInstant="2006-05-31T13:20:00.000">
    <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
    <saml:Statement
      xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
      <xacml-context:Response>
        <xacml-context:Result>
          <xacml-context:Decision>
            NotApplicable
          </xacml-context:Decision>
        </xacml-context:Result>
      </xacml-context:Response>
      <xacml-context:Request>
        . . . .
      </xacml-context:Request>
    </saml:Statement>
  </saml:Assertion>
</samlp:Response>
```

900 An instance of a `<samlp:Response>` element containing an XACMLAuthzDecision Assertion is called
901 an XACMLAuthzDecision Response in this Profile. Such a Response MUST be used as the response to
902 an `<xacml-samlp:XACMLAuthzDecisionQuery>`.

903 This Profile imposes the following requirements or restrictions on the `<samlp:Response>` element in
904 addition to those specified in SAML 2.0 when used as an XACMLAuthzDecision Response.

905 `<saml:Issuer>` [Optional]

906 The `<saml:Issuer>` element is an optional element that “Identifies the entity that generated the
907 response message” [SAML].

908 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
909 in the `<saml:Issuer>` element refer to the entity that signs the XACMLAuthzDecision Response. It
910 is up to the relying party to determine whether it has an appropriate trust relationship with the
911 authority that signs the Response.

912 `<ds:Signature>` [Optional]

913 The `<ds:Signature>` element is an optional element for holding “An XML Signature that
914 authenticates the responder and provides message integrity” [SAML].

915 A `<ds:Signature>` instance MAY be used in a XACMLAuthzDecision Response. In order to
916 support 3rd party digital signatures, this Profile does NOT require that the identity provided in the

917 <saml:Issuer> instance refer to the entity that signs the XACMLAuthzDecision Response. It is up
918 to the relying party to determine whether it has an appropriate trust relationship with the authority
919 that signs the Response.

920 A relying party SHOULD verify any signature included in the XACMLAuthzDecision Response and
921 SHOULD NOT use information derived from the Response unless the signature is verified
922 successfully.

923 <saml:Assertion> [Any Number]

924 <saml:Assertion> instances that MAY include one or more XACMLAuthzDecision Assertions that
925 represent responses to associated queries.

926 <samlp:StatusCode> [Required]

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

929 In the response to an <xacml-samlp:XACMLAuthzDecisionQuery>, the <samlp:StatusCode>
930 Value XML attribute MUST depend on the value of the <xacml-context:StatusCode> instance
931 of the XACML Response Context <xacml-context:Status> instance as follows:

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

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

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

936 This value for the <samlp:StatusCode> Value XML attribute MUST be used when the
937 <xacml-context:StatusCode> value is
938 urn:oasis:names:tc:xacml:1.0:status:missing-attribute or when the <xacml-
939 context:StatusCode> value is urn:oasis:names:tc:xacml:1.0:status:syntax-
940 error due to a syntax error in the <xacml-context:Request>.

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

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

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

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

951 InResponseTo [Optional]

952 This optional XML attribute is “A reference to the identifier of the request to which the response
953 corresponds.” When the XACMLAuthzDecision Response is issued in response to an
954 XACMLAuthzDecision Query, this XML attribute MUST contain the value of the ID XML attribute
955 from the XACMLAuthzDecision Query to which this is a response. This allows the receiver to
956 correlate the XACMLAuthzDecision Response with the corresponding XACMLAuthzDecision
957 Query. The SAML-defined ID XML attribute is a required component of an instance of the

958 <samlp:RequestAbstractType> of which the <xacml-
959 samlp:XACMLAuthzDecisionQuery> is an extension.

960 **4.11 Functional Requirements for the <xacml- 961 samlp:AssignedAttributes> Element**

962 *TBD: the matching of the <Holders> element against the Request Context is not defined yet since*
963 *the core schema (including the Request Context) is being rewritten.*

964

965 During processing of the provided access request, if the <xacml-samlp:Holders> element of a
966 provided <xacml-samlp:AssignedAttributes> element matches a section of the XACML Request
967 Context, then the XACML Context Handler MUST make the XACML Attributes in the <xacml-
968 samlp:HolderAttributes> element appear in that section of the XACML Request Context. Any
969 inheritance between <xacml-samlp:AssignedAttributes> elements is not deduced.

970 The matching of additional XACML Attributes MUST be made against all Request Contexts involved in
971 the processing of the XACMLAuthzDecision Query, including the provided access request itself and any
972 Request Contexts formed as part of reduction.

973 The provided XACML Attributes MUST be used only in the evaluation of the provided access request
974 and any derived Request Contexts, including reduction, and MUST NOT be used in evaluation of
975 requests not related to the provided access request unless associated with those other requests
976 independent of the <xacml-samlp:XACMLAuthzDecisionQuery>.

977 Note that, to implement this functionality, if additional XACML Attributes are fetched by the Context
978 Handler during processing, the implementation MUST test whether those additional XACML Attributes
979 provide a match for a <xacml-samlp:Holders> element. It is also conceivable that the XACML
980 Attributes provided in the <xacml-samlp:HolderAttributes> element may trigger XACML
981 Attributes from other attribute sources available to the Context Handler. An implementation MUST be
982 prepared to handle any circular dependencies that may arise.

983

5 Policies

984 XACML defines the `<xacml:Policy>` and `<xacml:PolicySet>` elements for expressing policies. In
985 many environments, instances of these elements need to be stored or transmitted between entities in an
986 XACML system. Such instances may need to be signed or associated with a validity period. SAML is
987 intended to provide this functionality for security-related assertions, but SAML does not define any
988 Protocol or Assertion elements for policies. In order to allow entities in an XACML system to use SAML
989 assertions and protocols to store, transmit, and query for XACML policies, this Profile defines one SAML
990 extension type and one SAML extension element, and describes how they are used with other standard
991 SAML elements.

- 992 • `<xacml-saml:XACMLPolicyStatementType>` is a new SAML extension type that includes
993 XACML policies.
- 994 • A `<saml:Statement>` defined using `xsi:type="xacml-saml:XACMLPolicyStatementType"`
995 MAY be used in an XACML system to store or convey XACML policies. An instance of a
996 `<saml:Statement>` element defined using this type is called an XACMLPolicy Statement in this
997 Profile.
- 998 • A `<saml:Assertion>` MUST be used to hold XACMLPolicy Statements. An instance of such a
999 `<saml:Assertion>` element is called an XACMLPolicy Assertion in this Profile.
- 1000 • An `<xacml-samlp:XACMLPolicyQuery>` is a new SAML extension element that MAY be used by
1001 a PDP or other entity to request XACML policies as a SAML protocol query.
- 1002 • A `<samlp:Response>` containing an XACMLPolicy Assertion that MUST be used in response to an
1003 `<xacml-samlp:XACMLPolicyQuery>`. It MAY be used to transmit XACML policies in other
1004 contexts. An instance of such a `<samlp:Response>` is called an XACMLPolicy Response in this
1005 Profile.

1006 This Section defines and describes the usage of these types and elements. The schemas for the new
1007 type and element are contained in the [XACML-SAML] and [XACML-SAML]P schema documents.

1008 5.1 Type `<xacml-saml:XACMLPolicyStatementType>`

1009 The `<xacml-saml:XACMLPolicyStatementType>` complex type contains XACML Policy and or
1010 XACML PolicySet elements. An instance of a `<saml:Statement>` element that is of this type is called
1011 an XACMLPolicy Statement in this Profile.

```

<complexType name="XACMLPolicyStatementType">
  <complexContent>
    <extension base="saml:StatementAbstractType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xacml:Policy"/>
          <element ref="xacml:PolicySet"/>
        </choice>
        <element ref="xacml-saml:ReferencedPolicies"
minOccurs="0" maxOccurs="1" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

1012 The `<xacml-saml:XACMLPolicyStatementType>` complex type is an extension to the SAML-
1013 defined `<saml:StatementAbstractType>`. It contains the following elements.

1014 `<xacml:Policy>` [Any Number]

1015 If the XACMLPolicy Statement represents a response to an `<xacml-samlp:XACMLPolicyQuery>`,
1016 then this element MUST contain one of the `<xacml:Policy>` instances that meet the specifications
1017 of the associated `<xacml-samlp:XACMLPolicyQuery>`. Otherwise, this element MAY contain an
1018 arbitrary `<xacml:Policy>` instance.

1019 `<xacml:PolicySet>` [Any Number]

1020 If the XACMLPolicy Statement represents a response to an `<xacml-samlp:XACMLPolicyQuery>`,
1021 then this element MUST contain one of the `<xacml:PolicySet>` instances that meet the
1022 specifications of the associated `<xacml-samlp:XACMLPolicyQuery>`. Otherwise, this element
1023 MAY contain an arbitrary `<xacml:PolicySet>` instance.

1024 `<xacml-saml:ReferencedPolicies>` [Zero or One]

1025 With the exception of XACML Policy and PolicySet instances that the receiver of the XACMLPolicy
1026 Statement is not authorized to view, this element MUST contain all XACML Policy and PolicySet
1027 instances required to resolve all `<xacml:PolicySetIdReference>` or
1028 `<xacml:PolicyIdReference>` instances contained in the XACMLPolicy Statement, including
1029 those in the `<xacml-saml:ReferencedPolicies>` instance itself. The values of the `PolicyId`
1030 and `PolicySetId` XML attributes of the policies included in the `<xacml-`
1031 `saml:ReferencedPolicies>` instance MUST exactly match the values contained in the
1032 corresponding `<xacml:PolicySetIdReference>` or `<xacml:PolicyIdReference>`
1033 instances.

1034 Subject to authorization and availability, if the XACMLPolicy Statement is issued in response to an
1035 `<xacml-samlp:XACMLPolicyQuery>`, there MUST be exactly one `<xacml:Policy>` element
1036 included for every XACML Policy that satisfies the XACMLPolicy Query, and there MUST be exactly one
1037 `<xacml:PolicySet>` element included for every XACML PolicySet that satisfies the XACMLPolicy
1038 Query . The responder MUST return all XACML policies available to the responder that satisfy the
1039 `<xacml-samlp:XACMLPolicyQuery>` and that the requester is authorized to receive.

1040 If the XACMLPolicy Statement is issued in response to an `<xacml-samlp:XACMLPolicyQuery>`, and
1041 there are no `<xacml:Policy>` or `<xacml:PolicySet>` instances that meet the specifications of the
1042 associated `<xacml-samlp:XACMLPolicyQuery>`, then there MUST be exactly one empty
1043 XACMLPolicy Statement included in the response.

1044 **5.2 Element <xacml-saml:ReferencedPolicies>**

1045 An instance of this element MUST be used to contain copies of all policies referenced from
1046 <xacml:Policy> or <xacml:PolicySet> instances included in the <xacml-
1047 samlp:XACMLPolicyQuery>, as well as copies of all policies referenced from other policies included in
1048 the <xacml-saml:ReferencedPolicies> instance.

1049 See Section 4.9 for a description of the <xacml-saml:ReferencedPolicies> element.

1050 **5.3 Element <saml:Statement>: XACMLPolicy Statement**

1051 A <saml:Statement> instance MAY be defined to be of type <xacml-
1052 samlp:XACMLPolicyStatementType> by using xsi:type="xacml-
1053 samlp:XACMLPolicyStatementType" as shown in the example in Section 5.4. such an instance of a
1054 <saml:Statement> element is called an XACMLPolicy Statement in this Profile. Any instance of an
1055 XACMLPolicy Statement in an XACML system MUST be enclosed in a <saml:Assertion>.

1056 **5.4 Element <saml:Assertion>: XACMLPolicy Assertion**

1057 A <saml:Assertion> instance MAY contain an XACMLPolicy Statement as shown in the following
1058 non-normative example:

```
<saml:Assertion Version="2.0" ID="9812368"  
  IssueInstant="2006-05-31T13:20:00.000">  
  <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>  
  <saml:Statement  
    xsi:type="xacml-saml:XACMLPolicyStatementType">  
    <xacml:Policy PolicyId="policy:1" RuleCombiningAlgId="..">  
      ....  
    </xacml:Policy>  
    <xacml:PolicySet PolicySetId="policyset:5" ... >  
      ...  
    </xacml:PolicySet>  
  </saml:Statement>  
</saml:Assertion>
```

1059 An instance of a <saml:Assertion> element containing an XACMLPolicy Statement is called an
1060 XACMLPolicy Assertion in this Profile.

1061 When an XACMLPolicy Assertion is part of a response to an <xacml-samlp:XACMLPolicyQuery>,
1062 then the XACMLPolicy Assertion MUST contain exactly one XACMLPolicy Statement, which in turn MAY
1063 contain any number of XACML Policy and PolicySet instances.

1064 This Profile imposes the following requirements and restrictions on the <saml:Assertion> element
1065 beyond those specified in SAML 2.0 when used as an XACMLPolicy Assertion.

1066 <saml:Issuer> [Required]

1067 The <saml:Issuer> element is a required element for holding information about “the SAML
1068 authority that is making the claim(s) in the assertion” [SAML].

1069 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
1070 in the <saml:Issuer> element refer to the entity that signs the XACMLPolicy Assertion. It is up to
1071 the relying party to determine whether it has an appropriate trust relationship with the authority that
1072 signs the XACMLPolicy Assertion.

1073 <ds:Signature> [Optional]

1074 The <ds:Signature> element is an optional element for holding “An XML Signature that
1075 authenticates the assertion, as described [in Section 5 of the SAML 2.0 core specification[SAML]].”

1076 A <ds:Signature> instance MAY be used in an XACMLPolicy Assertion. In order to support 3^d
1077 party digital signatures, this Profile does NOT require that the identity provided in the
1078 <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Assertion. It is up to the
1079 relying party to determine whether it has an appropriate trust relationship with the authority that signs
1080 the XACMLPolicy Assertion.

1081 A relying party SHOULD verify any signature included in the XACMLPolicy Assertion and SHOULD
1082 NOT use information derived from the XACMLPolicy Assertion unless the signature is verified
1083 successfully.

1084 <saml:Subject> [Optional]

1085 The <saml:Subject> element MUST NOT be included in an XACMLPolicy Assertion. Instead,
1086 the Subjects of an XACMLPolicy Assertion are specified in the XACML Policy and PolicySet
1087 elements contained in the enclosed XACMLPolicy Statement.

1088 <saml:Conditions> [Optional]

1089 The <saml:Conditions> element is an optional element that is used for “conditions that MUST be
1090 taken into account in assessing the validity of and/or using the assertion” [SAML].

1091 The <saml:Conditions> instance SHOULD contain NotBefore and NotOnOrAfter XML
1092 attributes to specify the limits on the validity of the XACMLPolicy Assertion. If these XML attributes
1093 are present, the relying party SHOULD ensure that an <xacml-context:Response> taken from
1094 the XACMLPolicy Assertion is used only during the XACMLPolicy Assertion's specified validity
1095 period.

1096 **5.5 Element <xacml-samlp:XACMLPolicyQuery>**

1097 An instance of the new <xacml-samlp:XACMLPolicyQuery> protocol element MAY be used by a
1098 PDP or application to request XACML <xacml:Policy> or <xacml:PolicySet> instances from an
1099 on-line Policy Administration Point.

```
<element name="XACMLPolicyQuery"
  xsi:type="xacml-samlp:XACMLPolicyQueryType" />
<complexType name="XACMLPolicyQueryType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element ref="xacml-context:Request"/>
        <element ref="xacml:PolicySetIdReference"/>
        <element ref="xacml:PolicyIdReference"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

1100 The <xacml-samlp:XACMLPolicyQuery> element is of <xacml-samlp:XACMLPolicyQueryType>
1101 complex type, which is an extension to the SAML-defined <samlp:RequestAbstractType>.

1102 The <xacml-samlp:XACMLPolicyQuery> element contains zero or more of the following elements in
1103 addition to those defined for the <samlp:RequestAbstractType>:

1104 <xacml-context:Request> [Any Number]

1105 An XACML Request Context. All XACML <xacml:Policy> and <xacml:PolicySet> instances
1106 potentially applicable to this Request that the requester is authorized to receive MUST be returned.
1107 The concept of “applicability” in the XACML context is defined in the XACML 2.0 Specification
1108 [XACML]. Any superset of applicable policies MAY be returned; for example, all policies having top-
1109 level Target elements that match the Request MAY be returned.

1110 <xacml:PolicySetIdReference> [Any Number]

1111 Identifies an XACML <xacml:PolicySet> instance to be returned.

1112 <xacml:PolicyIdReference> [Any Number]

1113 Identifies an XACML <xacml:Policy> instance to be returned.

1114 *Non-normative note: The <xacml-samlp:XACMLPolicyQuery> is not intended as a robust*
1115 *provisioning protocol. Users requiring such a protocol may consider using the OASIS Service*
1116 *Provisioning Markup Language (SPML). Note that the SAML-defined ID XML attribute is a required*
1117 *component of an instance of <samlp:RequestAbstractType> that the <xacml-*
1118 *samlp:XACMLPolicyQuery> extends and MAY be used to correlate the <xacml-*
1119 *samlp:XACMLPolicyQuery> with the corresponding XACMLPolicy Response.*

1120 **5.6 Element <samlp:Response>: XACMLPolicy Response**

1121 A <samlp:Response> instance MAY contain an XACMLPolicy Assertion. An instance of such a
1122 <samlp:Response> element is called an XACMLPolicy Response in this Profile. An XACMLPolicy
1123 Response is shown in the following non-normative example:

```
<samlp:Response Version="2.0" ID="x9812368"  
  IssueInstant="2006-05-31T13:20:00.000">  
  <saml:Assertion Version="2.0" ID="x9812369"  
    IssueInstant="2006-05-31T13:20:00.000">  
    <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>  
    <saml:Statement  
      xsi:type="xacml-saml:XACMLPolicyStatementType">  
      <xacml:PolicySet PolicySetId="policyset:1" ... >  
        ....  
      </xacml:PolicySet>  
    </saml:Statement>  
  </saml:Assertion>  
</samlp:Response>
```

1124 An instance of a <samlp:Response> element that contains an XACMLPolicy Assertion is called an
1125 XACMLPolicy Response in this Profile. Such a Response MUST be used as the response to an
1126 <xacml-samlp:XACMLPolicyQuery>. It MAY be used to convey or store XACML policies for other
1127 purposes.

1128 This Profile imposes the following requirements and restrictions on the <samlp:Response> element in
1129 addition to those specified in SAML 2.0 when used as an XACMLPolicy Response.

1130 <saml:Issuer> [Optional]

1131 The <saml:Issuer> element Identifies the entity that generated the XACMLPolicy Response
1132 message.” [SAML].

1133 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
1134 in the <saml:Issuer> element refer to the entity that signs the XACMLPolicy Response. It is up to
1135 the relying party to determine whether it has an appropriate trust relationship with the authority that
1136 signs the XACMLPolicy Response.

1137 <ds:Signature> [Optional]

1138 The <ds:Signature> element is an optional element for holding “An XML Signature that
1139 authenticates the responder and provides message integrity” [SAML].

1140 A <ds:Signature> instance MAY be used in an XACMLPolicy Response. In order to support 3rd
1141 party digital signatures, this Profile does NOT require that the identity provided in the
1142 <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Response. It is up to the
1143 relying party to determine whether it has an appropriate trust relationship with the authority that signs
1144 the XACMLPolicy Response.

1145 A relying party SHOULD verify any signature included in the XACMLPolicy Response and SHOULD
1146 NOT use information derived from the XACMLPolicy Response unless the signature is verified
1147 successfully.

1148 <saml:Assertion> [Any Number]

1149 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,
1150 then there MUST be exactly one instance of this element that contains an XACMLPolicy Assertion
1151 representing the response to the associated XACMLPolicy Query. If the XACMLPolicy Response is
1152 not issued in response to an <xacml-samlp:XACMLPolicyQuery>, it MAY contain one or more
1153 XACMLPolicy Assertions as well as other SAML or XACML Assertions.

1154 <saml>Status> [Required]

1155 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,
1156 and if it is not possible to return all policies that satisfy the <xacml-samlp:XACMLPolicyQuery>, then
1157 a <samlp:StatusCode> value of
1158 urn:oasis:names:tc:saml:2.0:status:TooManyResponses MUST be returned in the
1159 <samlp>Status> element of the Response.

1160 InResponseTo [Optional]

1161 This optional XML attribute is “A reference to the identifier of the request to which the response
1162 corresponds.” When the XACMLPolicy Response is issued in response to an <xacml-
1163 samlp:XACMLPolicyQuery>, this XML attribute MUST contain the value of the ID XML attribute
1164 from the <xacml-samlp:XACMLPolicyQuery> to which this is a response. This allows the
1165 receiver to correlate the XACMLPolicy Response with the corresponding XACMLPolicy Query.

1166

6 Advice

1167 This Section describes how to include XACMLAuthzDecision Assertion and XACMLPolicy Assertion
1168 instances as advice in another SAML Assertion instance.

1169 6.1 Element `<saml:Advice>`

1170 A SAML Assertion MAY include a `<saml:Advice>` element containing “Additional information related to
1171 the assertion that assists processing in certain situations but which MAY be ignored [without affecting
1172 either the semantics or the validity of the assertion] by applications that do not understand the advice or
1173 do not wish to make use of it.” [SAML] An XACMLAuthzDecision Assertion or XACMLPolicy Assertion
1174 may be used in the Advice element as shown in the following non-normative example:

```
<saml:Advice>
  <saml:Assertion Version="2.0" ID="200606231640"
    IssueInstant="2006-05-31T13:20:00:000">
    <saml:Issuer>https://XACMLPDP.example.com</saml:Issuer>
    <saml:Statement
      xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
      <xacml-context:Response>
        . . . .
      </xacml-context:Response>
      <xacml-context:Request>
        . . . .
      </xacml-context:Request>
    </saml:Statement>
  </saml:Assertion>
</saml:Advice>
```

1175
1176

7 Using an XACML Authorization Decision as an Authorization Token

1177
1178
1179
1180
1181

This Section of the Profile describes how to use an XACMLAuthzDecision Statement as a security and privacy authorization token as part of a SOAP message exchange in a Web Services context. This token MAY be used by a client to convey an authorization decision from a trusted 3rd party to a service. A Web Service MAY use such a token to determine that the client is authorized to access information involved in the Web Services interaction.

1182
1183
1184
1185
1186
1187
1188

In a Web Services context, an instance of an XACMLAuthzDecision Assertion MAY be used as an authorization token in the Web Services Security [WSS] `wsse:Security` Header of a SOAP message. When used in this way, the XACMLAuthzDecision Statement in the XACMLAuthzDecision Assertion MUST include the corresponding XACML Request Context. This allows the Web service to determine whether the `<xacml-context:Attribute>` instances in the Request correspond to the access that the client requires as part of the Web Service interaction. The XACMLAuthzDecision Assertion SHOULD be signed by a Policy Decision Point trusted by the Web Service.

1189
1190
1191
1192
1193
1194
1195
1196
1197

A Web Service MAY use this token to determine that a trusted 3rd party has evaluated an XACML Request Context that is relevant to the invocation of the service, and has reported an authorization decision. The service SHOULD verify that the signature on the XACMLAuthzDecision Assertion is from a Policy Decision Point that the service trusts. The service SHOULD verify that the validity period of the XACMLAuthzDecision Assertion includes the time at which the Web Service interaction will access the information or resource to which the Request Context applies. The service SHOULD verify that the `<xacml-context:Attribute>` instances contained in the XACML `<xacml-context:Request>` element correctly describe the information or resource access that needs to be authorized as part of this Web Service interaction.

1198

8 SAML Metadata

1199 The following SAML metadata extensions are RECOMMENDED.

1200 *TBD: this Section is under development. Contributions from developers who have implemented the*
1201 *Profile are invited. See <http://wiki.oasis-open.org/xacml/IssuesList>, Issue#74 for more information on*
1202 *current contributions to this topic.*

1203 These SAML metadata extensions are used to create XACML SAML versions of the standard SAML
1204 metadata information. The namespace for these metadata extensions is

```
1205 xmlns:xacml-samlm=  
1206 "urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:metadata
```

1207 The types defined in this Section of the Profile are used as in the following example, where an `xacml-`
1208 `samlm:XACMLPDPDescriptorType` is used to instantiate a standard SAML `md:RoleDescriptor` in a
1209 standard SAML `md:EntityDescriptor` by means of the `xsi:type` XML attribute: example:

```
<md:EntityDescriptor entityID="..." validUntil="..."  
  cacheDuration="..." ID="..." >  
  <ds:Signature>...</ds:Signature>  
  <md:RoleDescriptor xsi:type="xacml-samlm:XACMLPDPDescriptorType"  
    ...any std RoleDescriptor attributes... >  
    <xacml-samlm:XACMLAuthzService/>  
  </md:RoleDescriptor>  
  <md:Organization>...</md:Organization>  
  <md:ContactPerson>...</md:ContactPerson>  
  <md:AdditionalMetadataLocation>...</md:AdditionalMetadataLocation>  
</md:EntityDescriptor>
```

1210

1211 **8.1 Type <xacml-samlm:XACMLPDPDescriptorType>**

1212 PDP information: standard SAML metadata. Proposed syntax:


```

<complexType name="XACMLPDPDescriptorType">
  <complexContent>
    <extension base="md:RoleDescriptorType">
      <sequence>
        <element ref="xacml-samlm:XACMLAuthzService"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="XACMLAuthzService" type="md:EndpointType"/>

```

1213 **8.2 Type <xacml-samlm:XACMLPDPConfigType>**

1214 Extended PDP information. Attributes which are not defined in SAML standard metadata. No proposed
 1215 syntax yet.

1216 **8.3 Type <xacml-**
 1217 **samlm:XACMLAuthzDecisionQueryDescriptorType>**

1218 PEP endpoint information. Proposed syntax:

```

<complexType name="XACMLAuthzDecisionQueryDescriptorType">
  <complexContent>
    <extension base="md:QueryDescriptorType">
    </extension>
  </complexContent>
</complexType>

```

1219

```
<element name="XACMLAuthzDecisionQueryDescriptor"
  type="xacml-samlm:XACMLAuthzDecisionQueryDescriptorType"/>
<complexType name="XACMLAuthzDecisionQueryDescriptorType">
  <complexContent>
    <extension base="md:QueryDescriptorType">
    </extension>
  </complexContent>
</complexType>
```

1220 **8.4 Type <xacml-samlm:XACMLAuthzDecisionQueryConfigType>**

1221 PEP extended metadata. No proposed syntax yet.

9 Conformance

1222

1223 Implementations of this Profile MAY implement certain subsets of the described functionality. Each
1224 implementation MUST clearly identify the subsets it implements using the following identifiers.

1225 An implementation of this Profile is a conforming *SAML Attribute* implementation if the implementation
1226 conforms to Section 2 of this Profile. The following URI MUST be used as the identifier for this
1227 functionality:

1228 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:attrs:all

1229 An implementation of this Profile is a conforming *SOAP Attributes as XACML Authz Decision Query*
1230 implementation if the implementation conforms to Section 3.1 of this Profile. The following URI MUST be
1231 used as the identifier for this functionality:

1232 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:authzQuery

1233 An implementation of this Profile is a conforming *SOAP Attributes as SAML Attribute Assertion*
1234 implementation if the implementation conforms to Section 3.2 of this Profile. The following URI MUST be
1235 used as the identifier for this functionality:

1236 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:attrAssertion

1237

1238 An implementation of this Profile is a conforming *XACML Authz Decision without Policies* implementation
1239 if the implementation conforms to all parts of Section 4 of this Profile excluding the `<xacml:Policy>`,
1240 `<xacml:PolicySet>`, and `<xacml-samlp:ReferencedPolicies>` elements and their sub-
1241 elements and the `CombinePolicies` XML attribute in the `<xacml-
1242 samlp:XACMLAuthzDecisionQuery>`. XACML 3.0 implementations MUST support the `<xacml-
1243 samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-
1244 samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0 implementations MUST NOT support
1245 the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-
1246 samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be used as the identifier for this
1247 functionality:

1248 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionurn:oasis:nam
1249 es:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:noPolicies

1250 An implementation of this Profile is a conforming *XACML Authz Decision with Policies* implementation if
1251 the implementation conforms to all parts of Section 4 of this Profile. XACML 3.0 implementations MUST
1252 support the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-
1253 samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0 implementations MUST NOT support
1254 the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-
1255 samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be used as the identifier for this
1256 functionality:

1257 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:withPolicies

1258 An implementation of this Profile is a conforming *XACML Policies* implementation if the implementation
1259 conforms to Section 5 of this Profile. The following URI MUST be used as the identifier for this
1260 functionality:

1261 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:policies

1262 An implementation of this Profile is a conforming *SAML Advice* implementation if the implementation
1263 conforms to Section 6 of this Profile. The following URI MUST be used as the identifier for this
1264 functionality:

1265 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:adviceSAML

1266 An implementation of this Profile is a conforming *XACML Authz Token* implementation if the
1267 implementation conforms to Section 7 of this Profile. The following URI MUST be used as the identifier
1268 for this functionality:

1269 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzToken

1270 An implementation of this Profile is a conforming *SAML Metadata* implementation if the implementation
1271 conforms to Section 8 of this Profile. The following URI MUST be used as the identifier for this
1272 functionality:

1273 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:metadata

1274

Appendix A. Acknowledgments

1275 The following individuals have participated in the creation of this specification and are gratefully
1276 acknowledged

1276 **Participants:**

- 1277 • Anne Anderson, Sun Microsystems
- 1278 • Anthony Nadalin, IBM
- 1279 • Bill Parducci,
- 1280 • Carlisle Adams, University of Ottawa
- 1281 • Daniel Engovatov, BEA
- 1282 • Don Flinn,
- 1283 • Ed Coyne
- 1284 • Ernesto Damiani
- 1285 • Frank Siebenlist
- 1286 • Gerald Brose
- 1287 • Hal Lockhart
- 1288 • Haruyuki Kawabe
- 1289 • James MacLean
- 1290 • John Merrells
- 1291 • Ken Yagen
- 1292 • Konstantin Beznosov
- 1293 • Michiharu Kudo
- 1294 • Michael McIntosh
- 1295 • Pierangela Samarati
- 1296 • Pirasenna Velandai Thiyagarajan
- 1297 • Polar Humenn
- 1298 • Rebekah Metz
- 1299 • Ron Jacobson
- 1300 • Satoshi Hada
- 1301 • Sekhar Vajjhala
- 1302 • Seth Proctor
- 1303 • Simon Godik
- 1304 • Steve Anderson
- 1305 • Steve Crocker
- 1306 • Suresh Damodaran
- 1307 • Tim Moses
- 1308 • Von Welch
- 1309 • Frederic Deleon
- 1310 • Argyn Kuketayev

1311

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.
WD 2	23 June 2006	Anne Anderson	Changed name to "xacml-2.0-profile-saml2.0-v2-spec.... Removed specifications for all new elements except the XACMLAuthzDecisionQuery and XACMLPolicyQuery and all new types except for XACMLAuthzDecisionStatementType and XACMLPolicyStatementType and the two new Query types. Added descriptions of each standard SAML element in which XACML types might occur, and gave examples of use of xsi:type. Described use of the ID and InResponseTo attributes to correlate Queries and Responses.
WD 3	5 March 2007	Anne Anderson	-change boilerplate to conform to new OASIS template -Title: change to reflect that this profile applies to all versions of XACML -1.3 Added section on backwards compatibility -1.4 Removed notation section -1.5 Added namespaces section -2.6 Insert the "Conveying XACML Attributes in a SOAP Message" section from the WS-XACML profile -2.1.1 Clarify that <saml:Subject> is not translated into an XACML -id Attribute -3.5 and following,3.13: add syntax for passing additional Attributes in XACMLAuthzDecisionQuery from Admin Policy. 3.9 and following: add syntax for passing references policies. -4.4 XACMLPolicyQuery: clarify it returns all potentially applicable policies; remove Target element; change Choice lower bound from 0 to 1 and remove case where no elements included; add non-normative note to consider SPML for provisioning protocol -4.5 Response: Use valid ID values in example; add <samp:Status> element saying to use SAML TooManyResponses StatusCode if unable to return all applicable policies -7 Insert the "XACML Authorization Token" section from the WS-XACML profile -Schemas: create versions specific to each XACML version -Protocol schema: remove XACMLPolicyQuery Target element, change Choice lower bound from 0 to 1 -Protocol schema: add Administrative Policy elements.
WD 4	15 June 2007	Anne Anderson	-throughout: used actual schema elements rather than

Rev	Date	By whom	What
			<p>invented names except when speaking about instances embedded in other instances (e.g. <saml:Attribute> rather than SAML Attribute, but SAML Attribute Response rather than <samlp:Response>).</p> <ul style="list-style-type: none"> -throughout: changed SHALL to MUST -throughout: added namespace designators to schema items and added additional namespace prefixes to list in Section 1.4 -Figure 1 updated the "Components and messages diagram to use same names as text -2.1.1 Clarified that implementations need not create actual <xacml-context:Attribute> instances so long as PDP can obtain corresponding values as if such instances existed. -2.1.1 Reworded description of NotBefore, NotOnOrAfter relationship to XACML date/time Attributes to be more clear -3.4.7,B.1 Inserted non-normative notes referring to open issues in relevant places -3.4.4.1 Clarified that the ReferencedPolicies element need not contain policies that receiver is not authorized to view -3.9 Clarified that Policy[Set]IdReference values must exactly match corresponding Policy[Set]Id values in the ReferencedPolicies element. -3.7 Changed "AttributeMatch" to "Match" to fit 3.0 schema -3.9,schemas:Fixed schema for ReferencedPolicies so it validates -3.4.4.1 Reworded AssignedAttributes and XACMLAuthzDecisionQuery Policy[Set] descriptions to clarify that the values must not be used except with the given Request "unless associated with the ... independently of the Request" -4.1,4.2 Add ReferencedPolicies element to XACMLPolicyStatementType -4.6 Reworded so to allow Response that is not issued in response to a specific Query -7 Added first draft of SAML Metadata -8 Added urn for SAML Metadata functionality
WD 5	19 July 2007	Anne Anderson	<ul style="list-style-type: none"> -Import XACML 1.0 schemas from local copies -Import XACML 2.0 schemas from http://docs.oasis-open.org/xacml/ directory -Import XACML 3.0 WD3 schema -Add OASIS copyright to all schemas -Made "Conveying XACML Attributes in a SOAP Message" a separate Section for easier reference in Conformance Section -Revised Conformance Section to refer to current document sections and to include previously omitted elements. -Made Introduction non-normative except for Namespaces and Normative References sections. -Made SAML Metadata section normative but RECOMMENDED

