



SAML 2.0 Profile of XACML, Version 2.0

Committee Draft 1

16 April 2009

Specification URIs:

This Version:

- <http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-1-en.html>
- <http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-1-en.odt> (Authoritative)
- <http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-1-en.pdf>

Previous Version:

N/A

Latest Version:

- <http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.html>
- <http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.odt> (Authoritative)
- <http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.pdf>

Technical Committee:

OASIS eXtensible Access Control Markup Language (XACML) TC

Chair(s):

Hal Lockhart <hal.lockhart@oracle.com>
Bill Parducci <bill@parducci.net>

Editors:

Erik Rissanen <erik@axiomatics.com>
Hal Lockhart <hal.lockhart@oracle.com>

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
- XACML 1.0, 2.0, 3.0 OASIS Standards
- XACML 1.1 Committee Draft

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

33 urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:assertion:cd-01
34 urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:protocol:cd-01
35 urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:assertion:cd-01
36 urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:protocol:cd-01
37 urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:cd-01
38 urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01
39 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:assertion:cd-01
40 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:protocol:cd-01

41 **Abstract:**

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

52 **Status:**

53 This document was last revised or approved by the OASIS eXtensible Access Control Markup
54 Language (XACML) TC on the above date. The level of approval is also listed above. Check the
55 "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of
56 this document.

57 Technical Committee members should send comments on this specification to the Technical
58 Committee's email list. Others should send comments to the Technical Committee by using the
59 "Send A Comment" button on the Technical Committee's web page at [http://www.oasis-](http://www.oasis-open.org/committees/xacml/)
60 [open.org/committees/xacml/](http://www.oasis-open.org/committees/xacml/).

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

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

67 Notices

68 Copyright © OASIS® 2009. All Rights Reserved.

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

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

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

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

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

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

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

107 The names "OASIS", "SAML" and "XACML" are trademarks of [OASIS](http://www.oasis-open.org), the owner and developer of this
108 specification, and should be used only to refer to the organization and its official outputs. OASIS
109 welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce
110 its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above
111 guidance.

112 Table of Contents

113	1 Introduction.....	6
114	1.1 Organization of this Profile.....	6
115	1.1 Diagram of SAML integration with XACML.....	8
116	1.2 Backwards compatibility.....	9
117	1.3 Terminology.....	9
118	1.1 Namespaces.....	11
119	1.2 Normative References.....	12
120	1.3 Non-normative References.....	12
121	2 Attributes.....	13
122	2.1 Element <saml:Attribute>.....	13
123	2.1 Element <saml:AttributeStatement>.....	15
124	2.2 Element <saml:Assertion>: SAML Attribute Assertion.....	15
125	2.3 Element <samlp:AttributeQuery>.....	16
126	2.4 Element <samlp:Response>: SAML Attribute Response.....	16
127	3 Conveying XACML Attributes in a SOAP Message.....	18
128	3.1 <xacml-samlp:XACMLAuthzDecisionQuery>.....	18
129	3.2 SAML Attribute Assertion.....	18
130	4 Authorization Decisions.....	19
131	4.1 Type <xacml-saml:XACMLAuthzDecisionStatementType>.....	19
132	4.2 Element <saml:Statement>: XACMLAuthzDecision Statement.....	20
133	4.3 Element <saml:Assertion>: XACMLAuthzDecision Assertion.....	20
134	4.4 Element <xacml-samlp:XACMLAuthzDecisionQuery>.....	22
135	4.5 Element <xacml-samlp:AdditionalAttributes>.....	25
136	4.6 Element <xacml-samlp:AssignedAttributes>.....	25
137	4.7 Element <xacml-samlp:Holders>.....	26
138	4.8 Element <xacml-samlp:HolderAttributes>.....	26
139	4.9 Element <xacml-saml:ReferencedPolicies>.....	27
140	4.10 Element <samlp:Response>: XACMLAuthzDecision Response.....	28
141	4.11 Functional Requirements for the <xacml-samlp:AssignedAttributes> Element.....	30
142	5 XACML Decision Queries using WS-Trust.....	31
143	5.1 Common Claims Dialect.....	31
144	5.2 XACML Dialect.....	31
145	5.3 Decision Request.....	31
146	5.4 Decision Response.....	32
147	6 Policies.....	33
148	6.1 Type <xacml-saml:XACMLPolicyStatementType>.....	33
149	6.2 Element <xacml-saml:ReferencedPolicies>.....	34
150	6.3 Element <saml:Statement>: XACMLPolicy Statement.....	35
151	6.4 Element <saml:Assertion>: XACMLPolicy Assertion.....	35

152 6.5 Element <xacml-samlp:XACMLPolicyQuery>.....36
153 6.6 Element <samlp:Response>: XACMLPolicy Response.....37
154 6.7 Policy references and Policy assertions..... 38
155 7 Advice..... 39
156 7.1 Element <saml:Advice>..... 39
157 8 Using an XACML Authorization Decision as an Authorization Token..... 40
158 9 Conformance..... 41

1 Introduction

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
196
197
198
199
200
201
202

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

Non-normative through Section 1.3

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.

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

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

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

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

238 Section 6 describes the use of SAML in requesting, responding with, storing, and transmitting XACML
239 policies. The following types and elements are described:

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

- 246 3. <saml:Assertion> – A standard SAML element that MUST be used to hold XACMLPolicy
247 Statement instances in an XACML system, either in a repository or in an XACMLPolicy
248 Response. An instance of such a <saml:Assertion> element is called an XACMLPolicy
249 Assertion in this Profile.
- 250 4. <xacml-samlp:XACMLPolicyQuery> – A new SAML extension protocol element defined in
251 this Profile that MAY be used by a PDP or other application to request XACML policies from a
252 Policy Administration Point (PAP).
- 253 5. <samlp:Response> – A standard SAML protocol element that MUST be used to return
254 XACMLPolicy Assertions in response to an <xacml-samlp:XACMLPolicyQuery>. An
255 instance of such a <samlp:Response> element is called an XACMLPolicy Response in this
256 Profile.
- 257 Section 7 describes the use of XACMLAuthzDecision Assertion and XACMLPolicy Assertion instances as
258 advice in other SAML Assertions. The following element is described:
- 259 1. <saml:Advice> – A standard SAML element that MAY be used to convey XACMLPolicy
260 Assertions or XACMLAuthzDecision Assertions as advice in other <saml:Assertion>
261 instances.
- 262 Section 8 describes the use of XACMLAuthzDecision Assertions as authorization tokens in a SOAP
263 message exchange.
- 264 Section Error: Reference source not found describes recommended non-normative SAML metadata for
265 use with these XACML-related protocols.
- 266 Section 9 describes requirements for conformance with various aspects of this Profile.

267 **1.1 Diagram of SAML integration with XACML**

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

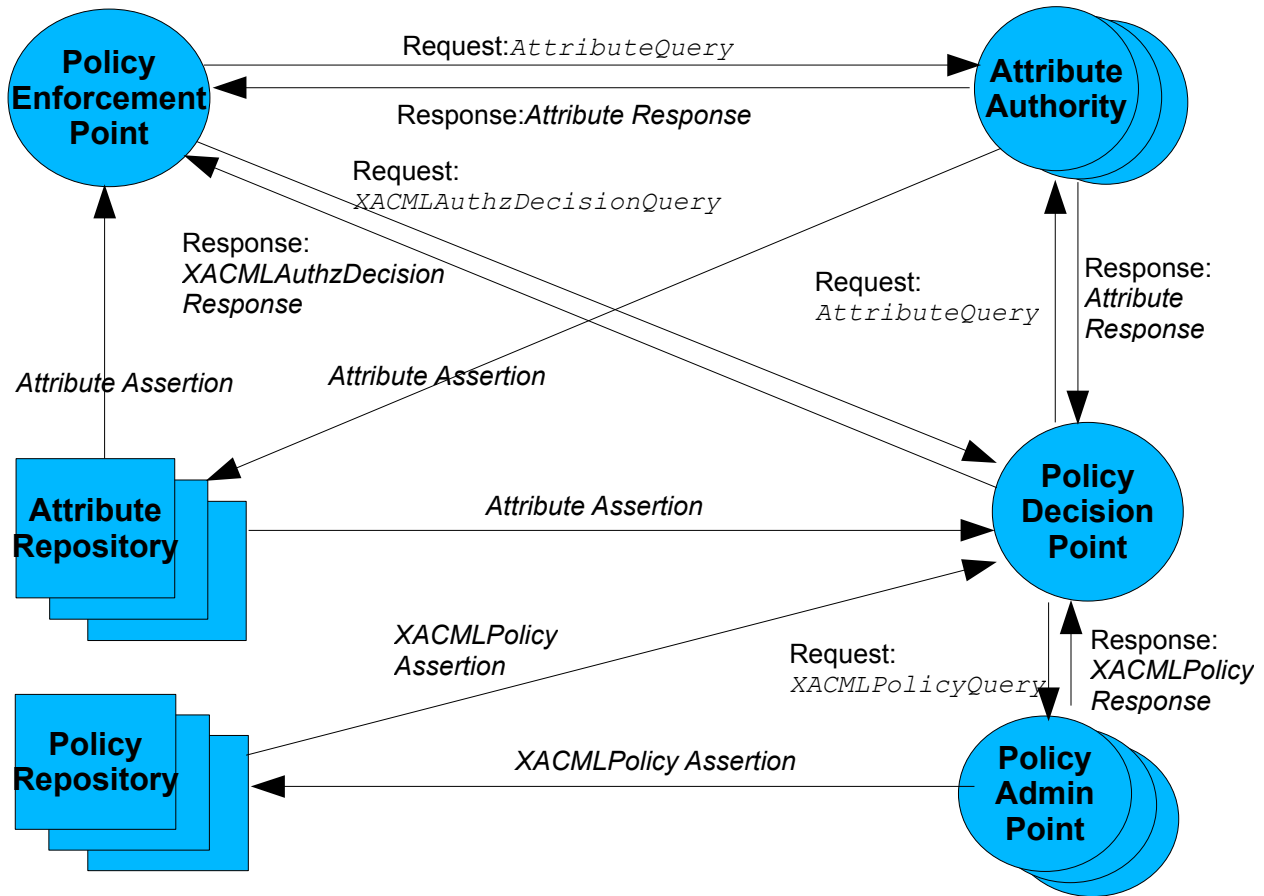


Figure 1: Components and messages in a integration of SAML with XACML

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

274 1.2 Backwards compatibility

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

278 1.3 Terminology

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

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

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

287 • An XACML Attribute is a typed name/value pair, with other optional information, specified using an
288 `<xacml-context:Attribute>` instance. An XACML Attribute is associated with an entity or topic
289 identity by the XACML Attribute's position within a particular Attribute group in the XACML Request.

290 • A SAML Attribute is a name/value pair, with other optional information, specified using a
291 `<saml:Attribute>` instance. A SAML Attribute is associated with a particular subject by its
292 inclusion in a SAML Attribute Assertion that contains a `<saml:Subject>` instance. The SAML
293 Subject may correspond to any XACML Attribute group.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

327 1.1 Namespaces

328 *Normative*

329 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
wst	http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3.xsd

330 This Profile is written for use with XACML 1.0 [XACML1], 1.1 [XACML1.1], 2.0 [XACML2], or 3.0
 331 [XACML3]. Depending on the version of XACML being used, the xacml, xacml-context, xacml-
 332 saml, and xacml-samlp namespace prefixes have the following values in the schemas:

333 **XACML 1.0:**

```
334   xacml="urn:oasis:names:tc:xacml:1.0:policy"
335   xacml-context="urn:oasis:names:tc:xacml:1.0:context"
336   xacml-saml=
337   "urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:assertion:wd-08"
338   xacml-samlp=
339   "urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:protocol:wd-08"
```

341 **XACML 1.1:**

```
342   xacml="urn:oasis:names:tc:xacml:1.0:policy"
343   xacml-context="urn:oasis:names:tc:xacml:1.0:context"
344   xacml-
345   saml="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:assertion:wd-08"
346   xacml-
347   samlp="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:protocol:wd-08"
```

349 **XACML 2.0:**

```
350   xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
351   xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
352   xacml-
353   saml="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:wd-08"
354   xacml-
355   samlp="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:wd-08"
```

357 XACML 3.0:
 358 xacml="urn:oasis:names:tc:xacml:3.0:schema:os"
 359 xacml-context="urn:oasis:names:tc:xacml:3.0:schema:os"

360 NOTE: XACML 3.0 uses a single schema for both policies and context.
 361 xacml-
 362 saml="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:assertion:wd-08"
 363 xacml-
 364 samlp="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:protocol:wd-08"

366 1.2 Normative References

- 367 **[ADMIN]** E. Rissanen, ed., *XACML v3.0 Administrative Policy Version 1.0*
- 368 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
 369 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 370 **[SAML]** S. Cantor, et al., eds., *Assertions and Protocols for the OASIS Security Assertion
 371 Markup Language (SAML) V2.0*, [http://www.oasis-
 372 open.org/committees/documents.php?wg_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security).
- 373 **[SAML-PROFILE]** J. Hughes, et al., eds., *Profiles for the OASIS Security Assertion Markup
 374 Language (SAML) V2.0*, [http://www.oasis-open.org/committees/documents.php?
 375 wg_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security).
- 376 **[XACML1]** *OASIS eXtensible Access Control Markup Language (XACML) Version 1.0*,
 377 <http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf>
- 378 **[XACML1.1]** *OASIS eXtensible Access Control Markup Language (XACML) Version 1.1*,
 379 [http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-
 380 1.1.pdf](http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf)
- 381 **[XACML2]** T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML)
 382 Version 2.0*, OASIS Standard, 1 February 2005, [http://docs.oasis-
 383 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).
- 384 **[XACML3]** E. Rissanen, ed., *OASIS eXtensible Access Control Markup Language (XACML)
 385 Version 3.0 working draft 11, 5 April 2009, FIXME URL*
- 386 **[XACML-SAML]** OASIS, the schemas associated with namespace <xacml-saml> that are a
 387 normative part of this Profile.
- 388 **[XACML-SAMPLP]** OASIS, the schemas associated with namespace <xacml-samlp> that are a
 389 normative part of this Profile.
- 390 **[WSFED]** OASIS, *Web Services Federation Language (WS-Federation) Version 1.2*
 391 Committee Draft 02 January 7, 2009 [http://docs.oasis-open.org/wsfed/federation/
 392 v1.2/cd/ws-federation-1.2-spec-cd-02.doc](http://docs.oasis-open.org/wsfed/federation/v1.2/cd/ws-federation-1.2-spec-cd-02.doc)
- 393 **[WSS]** OASIS, *Web Services Security: SOAP Message Security 1.0 (WS-Security
 394 2004)*, OASIS Standard December 2004, and *WS-Security Core Specification
 395 1.1*, OASIS Standard February 2006, <http://www.oasis-open.org/specs/index.php>.
- 396 **WSTRUST** OASIS, *WS-Trust 1.4 **FIXME***
- 397

398 1.3 Non-normative References

- 399 **[XACMLIntro]** S. Proctor, *A Brief Introduction to XACML*, [http://www.oasis-open.org/committees/
 400 download.php/2713/Brief_Introduction_to_XACML.html](http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html), 14 March 2003.
- 401

2 Attributes

402

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

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

2.1 Element `<saml:Attribute>`

424

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

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

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

431

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

- 437 • XACML `AttributeId` XML attribute

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

- 439 • XACML `DataType` XML attribute

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

- 443 • XACML `Issuer` XML attribute

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

445 • `<xacml-context:AttributeValue>`

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

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

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

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

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

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

476 • Any `NotBefore` and `NotOnOrAfter` XML attributes in the SAML Attribute Assertion MUST be valid
477 with respect to the `<xacml:Request>` in which the SAML-derived `<xacml:Attribute>` is used.
478 This means that the XACML Attributes associated with the following `AttributeId` values in the
479 `<xacml:Request>` MUST represent times and dates that are not before the `NotBefore` XML
480 attribute value and not on or after the `NotOnOrAfter` XML attribute value:
481 `urn:oasis:names:tc:xacml:1.0:environment:current-time`
482 `urn:oasis:names:tc:xacml:1.0:environment:current-date`
483 `urn:oasis:names:tc:xacml:1.0:environment:current-dateTime`

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

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

489 2.1 Element `<saml:AttributeStatement>`

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

493 2.2 Element `<saml:Assertion>`: SAML Attribute Assertion

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

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

501 `<saml:Issuer>` [Required]

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

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

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

512 `<ds:Signature>` [Optional]

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

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

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

523 `<saml:Subject>` [Optional]

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

527 In a SAML Attribute Assertion containing a `<saml:Attribute>` that is to be mapped to an `<xacml-
528 context:Attribute>`, the `<saml:Subject>` instance MUST contain the identity of the entity to
529 which the `<saml:Attribute>` and its value are bound. For a mapped `<saml:Attribute>` to be
530 placed in a given XACML Attribute group, this identity SHOULD refer to the same entity as any

531 XACML Attribute that serves as an entity identifier in the Attribute group. For example, the
532 <saml:Subject> associated with a mapped SAML->XACML Attribute to be placed in the
533 XACML <xacml-context:Resource> Attribute group SHOULD refer to the same entity as the
534 value of any XACML Attribute having an AttributeId of
535 urn:oasis:names:tc:xacml:1.0:resource:resource-id that occurs in the same <xacml-
536 context:Resource> instance. See Section 2.1 for more information.

537 <saml:Conditions> [Optional]

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

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

550 2.3 Element <samlp:AttributeQuery>

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

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

558 2.4 Element <samlp:Response>: SAML Attribute Response

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

566 <saml:Issuer> [Optional]

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

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

573 <ds:Signature> [Optional]

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

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

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

3 Conveying XACML Attributes in a SOAP Message

584

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

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

3.1 `<xacml-samlp:XACMLAuthzDecisionQuery>`

595

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

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

3.2 SAML Attribute Assertion

609

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

4 Authorization Decisions

615

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

- 627 • `<xacml-saml:XACMLAuthzDecisionStatementType>` is a new SAML extension type that
628 includes an XACML `<xacml-context:Response>` along with other optional information.
- 629 • A `<saml:Statement>` of type `<xacml-saml:XACMLAuthzDecisionStatementType>` (defined
630 using `xsi:type`) MAY be used by a PDP Context Handler to convey an XACML `<xacml-`
631 `context:Response>` along with other optional information. An instance of such a
632 `<saml:Statement>` element is called an XACMLAuthzDecision Statement in this Profile.
- 633 • A `<saml:Assertion>` MUST be used to hold XACMLAuthzDecision Statements. An instance of
634 such a `<saml:Assertion>` element is called an XACMLAuthzDecision Assertion in this Profile.
- 635 • A `<xacml-samlp:XACMLAuthzDecisionQuery>` is a new SAML extension element that MAY be
636 used by a PEP to submit an XACML Request Context, along with other optional information, as a
637 SAML protocol query to an XACML Context Handler.
- 638 • A `<samlp:Response>` containing an XACMLAuthzDecision Assertion MUST be used by an XACML
639 Context Handler as the response to an `<saml-samlp:XACMLAuthzDecisionQuery>`. An instance
640 of such a `<samlp:Response>` element is called an XACMLAuthzDecision Response in this Profile.

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

643 4.1 Type `<xacml-saml:XACMLAuthzDecisionStatementType>`

644 The new `<xacml-saml:XACMLAuthzDecisionStatementType>` complex type contains an XACML
645 Response Context along with related information. Use of this type is an alternative to use of the SAML-
646 defined `<saml:AuthzDecisionStatementType>`; this alternative allows an XACML Context Handler
647 to use SAML with full support for XACML authorization decisions. An instance of a `<saml:Statement>`
648 element that is of this type (defined using `xsi:type="xacml-`
649 `saml:XACMLAuthzDecisionStatementType`") is called an XACMLAuthzDecision Statement in this
650 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>

```

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

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

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

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

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

660 If the XACMLAuthzDecision Statement represents a response to an `<xacml-`
 661 `samlp:XACMLAuthzDecisionQuery>`, and if the `ReturnContext` XML attribute in the `<xacml-`
 662 `samlp:XACMLAuthzDecisionQuery>` instance is "true", then this element MUST be included; if
 663 the `ReturnContext` XML attribute in the `<xacml-samlp:XACMLAuthzDecisionQuery>` instance
 664 is "false", then this element MUST NOT be included. See the description of the `ReturnContext`
 665 XML attribute in Section 4.4 for a specification of the `<xacml-context:Attribute>` instances that
 666 MUST be returned in this element when it is part of a response to an `<xacml-`
 667 `samlp:XACMLAuthzDecisionQuery>`.

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

672 4.2 Element `<saml:Statement>`: XACMLAuthzDecision Statement

673 A `<saml:Statement>` instance MAY be of type `<xacml-`
 674 `saml:XACMLAuthzDecisionStatementType>` by using `xsi:type` as shown in the example in
 675 Section 4.3. An instance of a `<saml:Statement>` element that is of type `<xacml-`
 676 `saml:XACMLAuthzDecisionStatementType>` is called an XACMLAuthzDecision Statement in this
 677 Profile. Any instance of an XACMLAuthzDecision Statement in an XACML system MUST be enclosed in
 678 a `<saml:Assertion>`.

679 4.3 Element `<saml:Assertion>`: XACMLAuthzDecision Assertion

680 A `<saml:Assertion>` instance MAY contain an XACMLAuthzDecision Statement as shown in the
 681 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>

```

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

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

686 `<saml:Issuer>` [Required]

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

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

693 `<ds:Signature>` [Optional]

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

696 A `<ds:Signature>` instance MAY be used in a `<saml:Assertion>`. In order to support 3rd party
 697 digital signatures, this Profile does NOT require that the identity provided in the `<saml:Issuer>`
 698 instance refer to the entity that signs the XACMLAuthzDecision Assertion. It is up to the relying party
 699 to determine whether it has an appropriate trust relationship with the authority that signs the Assertion
 700 .

701 A relying party SHOULD verify any signature included in the XACMLAuthzDecision Assertion and
 702 SHOULD NOT use information derived from the Assertion unless the signature is verified
 703 successfully.

704 `<saml:Subject>` [Optional]

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

709 `<saml:Conditions>` [Optional]

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

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

717 **4.4 Element <xacml-samlp:XACMLAuthzDecisionQuery>**

718 The <xacml-samlp:XACMLAuthzDecisionQuery> protocol element MAY be used by a PEP to
719 request an authorization decision from an XACML PDP. This element is an alternative to the SAML-
720 defined <samlp:AuthzDecisionQuery>; this alternative allows the PEP to use the full capabilities of
721 an XACML PDP. It allows use of the SAML query protocol to convey an XACML Request Context along
722 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>

```

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

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

728 `InputContextOnly` [Default "false"]

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

735 `ReturnContext` [Default "false"]

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

739 If this attribute is "True", then the PDP SHALL include the `<xacml-context:Request>` element in
740 the `<XACMLAuthzDecisionStatement>` element in the `<XACMLResponse>`. This `<xacml-`
741 `context:Request>` element SHALL include all those XACML Attributes supplied by the PEP in the

742 <XACMLAuthzDecisionQuery> that were used in making the authorization decision. A conforming
743 PDP MAY omit those XACML Attributes which were not referenced in any policy which was evaluated
744 in making the decision. If the value of the InputContextOnly Attribute in the Request is "False",
745 the PDP MAY include additional XACML Attributes in this <xacml-context:Request> element,
746 which were obtained by the PDP and used in making the authorization decision.

747

748 If this XML attribute is "false", then the PDP MUST NOT include an <xacml-context:Request>
749 instance in the XACMLAuthzDecision Statement in the XACMLAuthzDecision Response.

750 CombinePolicies [Default "true"]

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

754 If the attribute value is "true", then the PDP MUST insert all policies passed in the<xacml:Policy>
755 and <xacml:PolicySet> elements in the <xacml-sampl:XACMLAuthzDecisionQuery> into
756 the set of policies or policy sets that define the PDP as specified in Section 7.11 of the XACML 3.0
757 core specification [XACML3]. They MUST be combined with the other policies using the policy
758 combining algorithm that defines the PDP as specified in Section 7.11 of the XACML 3.0 core
759 specification [XACML3]. If the policy combining algorithm that defines the PDP is one in which
760 element order is considered, then the policies passed in the XACMLAuthzDecision Query MUST be
761 considered in the order in which they appear in the <xacml-sampl:XACMLAuthzDecisionQuery>
762 and MUST be considered as preceding all other policies that define the PDP.

763

764 If the attribute value is "false", then there MUST be no more than one <xacml:Policy> or
765 <xacml:PolicySet> passed in the <xacml-sampl:XACMLAuthzDecisionQuery>. This policy
766 MUST be treated as the policy that defines the PDP as specified in Section 7.11 of the XACML 3.0
767 core specification [XACML3] for evaluation of the <xacml-context:Request> passed in the
768 <xacml-sampl:XACMLAuthzDecisionQuery>. It MUST NOT be used to evaluate any other <xacml-
769 context:Request> instances unless provided to the PDP independent of the particular<xacml-
770 context:Request>.

771 <xacml-context:Request> [Required]

772 An XACML Request Context that is to be evaluated.

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

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

776 <xacml:Policy> [Any Number]

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

779 If the CombinePolicies XML attribute is "true", then the PDP MUST use such XACML Policy
780 instances.

781 If the CombinePolicies XML attribute is "false", then the PDP MUST use this XACML Policy
782 instance. There MUST be only one such XACML Policy instance and there MUST NOT be any
783 XACML PolicySet instances in this <xacml-sampl:XACMLAuthzDecisionQuery> instance.

784 <xacml:PolicySet> [Any Number]

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

787 If the `CombinePolicies` XML attribute is "true", then the PDP MUST use such XACML PolicySet
788 instances.

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

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

793 With the exception of XACML Policy and PolicySet instances that the receiver of the
794 XACMLAuthzDecision Statement is not authorized to view, this element MAY contain XACML Policy
795 and PolicySet instances required to resolve `<xacml:PolicySetIdReference>` or
796 `<xacml:PolicyIdReference>` instances contained in the XACMLAuthzDecision Statement,
797 including those in the `<xacml-saml:ReferencedPolicies>` instance itself, or contained in the
798 policies already available to the PDP. The values of the `PolicyId` and `PolicySetId` XML
799 attributes of the policies included in the `<xacml-saml:ReferencedPolicies>` instance MUST
800 exactly match the values contained in the corresponding `<xacml:PolicySetIdReference>` or
801 `<xacml:PolicyIdReference>` instances.

802 4.5 Element `<xacml-samlp:AdditionalAttributes>`

803 This element applies only for use with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML
804 3.0 PDP.

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

```
811 <element name="AdditionalAttributes"  
812   type="xacml-samlp: AdditionalAttributesType"/>  
813 <complexType name="AdditionalAttributesType">  
814   <sequence>  
815     <element ref="xacml-samlp:AssignedAttributes" minOccurs="0"  
816     maxOccurs="unbounded"/>  
817   </sequence>  
818 </complexType>
```

819 The `<AdditionalAttributes>` element is of `AdditionalAttributesType` complex type.

820 The `<AdditionalAttributes>` element contains the following elements:

821 `<AssignedAttributes>` [Required]

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

823 4.6 Element `<xacml-samlp:AssignedAttributes>`

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

826 The <AssignedAttributes> element MUST contain XACML Attributes that apply to delegate entities
827 identified by the <xacml-samlp: Holders> element.

```
828 <element name="AssignedAttributes" type="xacml-samlp:AssignedAttributesType"/>  
829 <complexType name="AssignedAttributesType">  
830 <sequence>  
831 <element ref="xacml-samlp: Holders"/>  
832 <element ref="xacml-samlp: HolderAttributes"/>  
833 </sequence>  
834 </complexType>
```

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

836 The <AssignedAttributes> element contains the following elements:

837 <xacml-samlp: Holders> [Required]

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

839 <xacml-samlp: HolderAttributes> [Required]

840 The XACML Attributes of the delegate entity.

841 4.7 Element <xacml-samlp: Holders>

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

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

```
846 <element name=" Holders" type="xacml-samlp: HoldersType"/>  
847 <complexType name=" HoldersType">  
848 <sequence>  
849 <element ref="xacml: Match" maxOccurs="unbounded"/>  
850 </sequence>  
851 </complexType>
```

852 The <xacml-samlp: Holders> element is of <xacml-samlp: HoldersType> complex type.

853 The <xacml-samlp: Holders> element contains the following elements:

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

855 Matches the delegate entities to which the XACML Attributes in the associated <xacml-
856 samlp: HolderAttributes> element apply. The < Match> elements shall be
857 evaluated according to the XACML schema against the < Attributes> elements in a
858 < Request> during reduction. If any < Match> element evaluates to "Match" then the
859 supplied attributes shall apply to the < Attributes> element which was referenced by the
860 attribute designator or selector contained in the < Match> element

861

862 4.8 Element <xacml-samlp: HolderAttributes>

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

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

```
867 <element name="HolderAttributes" type="xacml-samlp:HolderAttributesType"/>  
868 <complexType name="HolderAttributesType">  
869   <sequence>  
870     <element ref="xacml-context:Attribute"  
871       minOccurs="0" maxOccurs="unbounded"/>  
872   </sequence>  
873 </complexType>
```

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

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

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

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

880 **4.9 Element `<xacml-saml:ReferencedPolicies>`**

881 An instance of this element MAY be used to contain copies of policies referenced from `<xacml:Policy>`
882 or `<xacml:PolicySet>` instances included in an XACMLAuthzDecision Statement or in an
883 XACMLPolicy Statement, as well as copies of all policies referenced from other policies included in the
884 `<xacml-saml:ReferencedPolicies>` instance or policies already present in the PDP If a
885 `<xacml:Policy>` or `<xacml:PolicySet>` instance would match a policy both among the policies
886 already present to the PDP as well as a policy contained in the supplied `<xacml-`
887 `saml:ReferencedPolicies>` instance, then the supplied policy takes precedence.

```
888 <element name="ReferencedPolicies"  
889   type="xacml-saml:ReferencedPoliciesType"/>  
890 <complexType name="ReferencedPoliciesType">  
891   <sequence>  
892     <choice minOccurs="0" maxOccurs="unbounded">  
893       <element ref="xacml:Policy"/>  
894       <element ref="xacml:PolicySet"/>  
895     </choice>  
896   </sequence>  
897 </complexType>
```

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

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

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

902 A single `<xacml:Policy>` that is referenced using an `<xacml:PolicyIdReference>` from
903 another `<xacml:Policy>` or `<xacml:PolicySet>` instance. The value of the `PolicyId` XML
904 attribute in the `<xacml:Policy>` MUST be equal to the value of the corresponding
905 `<xacml:PolicyIdReference>` element.

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

907 A single `<xacml:PolicySet>` that is referenced using an `<xacml:PolicySetIdReference>`
908 from another `<xacml:Policy>` or `<xacml:PolicySet>` instance. The value of the `PolicySetId`

909 XML attribute in the `<xacml:PolicySet>` MUST be equal to the value of the corresponding
910 `<xacml:PolicySetIdReference>` element.

911 **4.10 Element `<samlp:Response>`: XACMLAuthzDecision Response**

912 A `<samlp:Response>` instance MAY contain an XACMLAuthzDecision Assertion as shown in the
913 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>
```

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

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

919 `<saml:Issuer>` [Optional]

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

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

926 `<ds:Signature>` [Optional]

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

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

934 A relying party SHOULD verify any signature included in the XACMLAuthzDecision Response and
935 SHOULD NOT use information derived from the Response unless the signature is verified
936 successfully.

937 <saml:Assertion> [Any Number]

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

940 <samlp:StatusCode> [Required]

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

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

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

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

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

950 This value for the <samlp:StatusCode> Value XML attribute MUST be used when the
951 <xacml-context:StatusCode> value is
952 urn:oasis:names:tc:xacml:1.0:status:missing-attribute or when the <xacml-
953 context:StatusCode> value is urn:oasis:names:tc:xacml:1.0:status:syntax-
954 error due to a syntax error in the <xacml-context:Request>.

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

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

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

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

965 InResponseTo [Optional]

966 This optional XML attribute is "A reference to the identifier of the request to which the response
967 corresponds." When the XACMLAuthzDecision Response is issued in response to an
968 XACMLAuthzDecision Query, this XML attribute MUST contain the value of the ID XML attribute
969 from the XACMLAuthzDecision Query to which this is a response. This allows the receiver to
970 correlate the XACMLAuthzDecision Response with the corresponding XACMLAuthzDecision
971 Query. The SAML-defined ID XML attribute is a required component of an instance of the
972 <samlp:RequestAbstractType> of which the <xacml-
973 samlp:XACMLAuthzDecisionQuery> is an extension.

974 **4.11 Functional Requirements for the <xacml-**
975 **samlp:AssignedAttributes> Element**

976 During processing of the provided access request, if the <xacml-samlp:Holderes> element of a
977 provided <xacml-samlp:AssignedAttributes> element matches a section of the XACML Request
978 Context, then the XACML Context Handler MUST make the XACML Attributes in the <xacml-
979 samlp:HolderAttributes> element appear in that section of the XACML Request Context. Any
980 inheritance between <xacml-samlp:AssignedAttributes> elements is not deduced.

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

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

988 The implementation MUST match the <xacml-samlp:Holderes> element against all the attributes
989 available to the context handler, but MUST NOT use any matching <xacml-
990 samlp:HolderAttributes> to find even more attributes through the context handler or even more
991 supplied attributes through other <xacml-samlp:Holderes> elements. This implies that there can be no
992 inheritance between <xacml-samlp:AssignedAttributes> elements.

993 5 XACML Decision Queries using WS-Trust

994 In some environments, it may be desirable to obtain an XACML authorization decision from a Security
995 Token Service (STS) using the WS-Trust protocol [WSTRUST].

996 5.1 Common Claims Dialect

997 One method of doing this is to support the Common Claim Dialect as defined in WS-Federation [WSFED],
998 chapter 9. In this case the implementation must map the contents of an incoming
999 <RequestSecurityToken> element into a XACML <Request> element and map the XACML <Response>
1000 into an outgoing <RequestSecurityTokenResponseCollection> element. When this approach is taken,
1001 there is no explicit reference to XACML in the wire protocol and in general a requestijg party will not be
1002 aware whether or not an XACML-based PDP was used to make the decision.

1003 5.2 XACML Dialect

1004 This section defines a WS-Trust-based protocol which is intended to be easier and more efficient for XACML
1005 PDP to implement. It is based directly on the constructs previously defined in Section 4. It uses the
1006 <saml:Assertion> element and <saml:Statement> of type xacml-
1007 saml:XACMLAuthzDecisionStatementType to wrap the XACML <Request> and <Response> elements.
1008 However, the <xacml-samlp:XACMLDecisionQuery> and <samlp:Response> elements are not used.
1009 Instead the request is conveyed in a <wst:RequestSecurityToken> element and the response is carried in
1010 a <wst:RequestSecurityTokenResponseCollection> element containing a
1011 <wst:RequestSecurityTokenResponse> element.

1012 Except for the outer protocol layer, described in more detail below, the syntax and functional requirements
1013 for this protocol is exactly as described above in section 4. In fact, it is possible for a server which
1014 contains an XACML PDP to support both protocols, using distinct web service endpoints, with only a
1015 small amount of distinct code to handle each request type.

1016 5.3 Decision Request

1017 The decision request is contained in a <wst:RequestSecurityToken> element. This element contains the
1018 following attributes and elements from the WS-Trust schema.

- 1019 • Context This URI specifies an identifier for this request. Its value will be returned in the
1020 corresponding response to allow them to be correlated.
- 1021 • <wst:TokenType> This element contains the value: urn:oasis:names:tc:xacml:3.0:core:schema,
1022 to indicate that an XACML decision token will be returned.
- 1023 • <wst:RequestType> This element contains the value: [http://docs.oasis-open.org/ws-sx/ws-
1024 trust/200512/Issue](http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue)

1025 In addition, the <wst:RequestSecurityToken> element MAY contain any of the attributes and elements
1026 defined in section 4.4 above as being contained in the <xacml-samlp:XACMLAuthzDecisionQuery>
1027 element. Specifically these are the attributes:

- 1028 • InputContextOnly,
- 1029 • ReturnContext, and
- 1030 • CombinePolicies.

1031 These are the elements:

- 1032 • <xacml-context:Request>,
- 1033 • <xacml-samlp:AdditionalAttributes>,
- 1034 • <xacml:Policy>,
- 1035 • <xacml:PolicySet>, and
- 1036 • <xacml-saml:ReferencedPolicies>.

1037 The functional requirements for processing these attributes and elements are exactly as set forth in
1038 section 4 above.

1039 **5.4 Decision Response**

1040 The decision response is contained in a <wst:RequestTokenResponseCollection> element. It contains
1041 exactly one <wst:RequestTokenResponse> element. This element contains the following attributes and
1042 elements.

- 1043 • Context This element contains the same URI provided in the Context attribute of the request.
- 1044 • <wst:RequestedSecurityToken> This element contains a <saml:Assertion which in turn contains a
1045 <saml:Statement of type xacml-saml:XACMLAuthzDecisionStatementType as described in
1046 sections 4.1, 4.2, and 4.3 above. The functional requirements for processing these attributes
1047 and elements are exactly as set forth in section 4 above.

1048

6 Policies

1049 XACML defines the `<xacml:Policy>` and `<xacml:PolicySet>` elements for expressing policies. In
1050 many environments, instances of these elements need to be stored or transmitted between entities in an
1051 XACML system. Such instances may need to be signed or associated with a validity period. SAML is
1052 intended to provide this functionality for security-related assertions, but SAML does not define any
1053 Protocol or Assertion elements for policies. In order to allow entities in an XACML system to use SAML
1054 assertions and protocols to store, transmit, and query for XACML policies, this Profile defines one SAML
1055 extension type and one SAML extension element, and describes how they are used with other standard
1056 SAML elements.

- 1057 • `<xacml-saml:XACMLPolicyStatementType>` is a new SAML extension type that includes
1058 XACML policies.
- 1059 • A `<saml:Statement>` defined using `xsi:type="xacml-saml:XACMLPolicyStatementType"`
1060 MAY be used in an XACML system to store or convey XACML policies. An instance of a
1061 `<saml:Statement>` element defined using this type is called an XACMLPolicy Statement in this
1062 Profile.
- 1063 • A `<saml:Assertion>` MUST be used to hold XACMLPolicy Statements. An instance of such a
1064 `<saml:Assertion>` element is called an XACMLPolicy Assertion in this Profile.
- 1065 • An `<xacml-samlp:XACMLPolicyQuery>` is a new SAML extension element that MAY be used by a
1066 PDP or other entity to request XACML policies as a SAML protocol query.
- 1067 • A `<samlp:Response>` containing an XACMLPolicy Assertion that MUST be used in response to an
1068 `<xacml-samlp:XACMLPolicyQuery>`. It MAY be used to transmit XACML policies in other
1069 contexts. An instance of such a `<samlp:Response>` is called an XACMLPolicy Response in this
1070 Profile.

1071 This Section defines and describes the usage of these types and elements. The schemas for the new
1072 type and element are contained in the [XACML-SAML] and [XACML-SAML]P schema documents.

1073 6.1 Type `<xacml-saml:XACMLPolicyStatementType>`

1074 The `<xacml-saml:XACMLPolicyStatementType>` complex type contains XACML Policy and or
1075 XACML PolicySet elements. An instance of a `<saml:Statement>` element that is of this type is called
1076 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>
```

1077 The `<xacml-saml:XACMLPolicyStatementType>` complex type is an extension to the SAML-defined
1078 `<saml:StatementAbstractType>`. It contains the following elements.

1079 <xacml:Policy> [Any Number]

1080 If the XACMLPolicy Statement represents a response to an <xacml-samlp:XACMLPolicyQuery>,
1081 then this element MUST contain one of the <xacml:Policy> instances that meet the specifications
1082 of the associated <xacml-samlp:XACMLPolicyQuery>. Otherwise, this element MAY contain an
1083 arbitrary <xacml:Policy> instance.

1084 <xacml:PolicySet> [Any Number]

1085 If the XACMLPolicy Statement represents a response to an <xacml-samlp:XACMLPolicyQuery>,
1086 then this element MUST contain one of the <xacml:PolicySet> instances that meet the
1087 specifications of the associated <xacml-samlp:XACMLPolicyQuery>. Otherwise, this element
1088 MAY contain an arbitrary <xacml:PolicySet> instance.

1089 <xacml-saml:ReferencedPolicies> [Zero or One]

1090 With the exception of XACML Policy and PolicySet instances that the receiver of the XACMLPolicy
1091 Statement is not authorized to view, this element MAY contain XACML Policy and PolicySet instances
1092 required to resolve <xacml:PolicySetIdReference> or <xacml:PolicyIdReference>
1093 instances contained in the XACMLPolicy Statement, including those in the <xacml-
1094 saml:ReferencedPolicies> instance itself. The values of the PolicyId and PolicySetId
1095 XML attributes of the policies included in the <xacml-saml:ReferencedPolicies> instance
1096 MUST exactly match the values contained in the corresponding
1097 <xacml:PolicySetIdReference> or <xacml:PolicyIdReference> instances.

1098 Subject to authorization and availability, if the XACMLPolicy Statement is issued in response to an
1099 <xacml-samlp:XACMLPolicyQuery>, there MUST be exactly one <xacml:Policy> element
1100 included for every XACML Policy that satisfies the XACMLPolicy Query, and there MUST be exactly one
1101 <xacml:PolicySet> element included for every XACML PolicySet that satisfies the XACMLPolicy
1102 Query . The responder MUST return all XACML policies available to the responder that satisfy the
1103 <xacml-samlp:XACMLPolicyQuery> and that the requester is authorized to receive.

1104 If the XACMLPolicy Statement is issued in response to an <xacml-samlp:XACMLPolicyQuery>, and
1105 there are no <xacml:Policy> or <xacml:PolicySet> instances that meet the specifications of the
1106 associated <xacml-samlp:XACMLPolicyQuery>, then there MUST be exactly one empty
1107 XACMLPolicy Statement included in the response.

1108 An XACMLPolicy Statement enclosed in a signed SAML assertion MAY be used as a method of
1109 authentication of XACML policies. In this case the Policy or PolicySet MUST NOT contain an XACML
1110 <PolicyIssuer> element. Instead the PDP MAY generate a <PolicyIssuer> element from the certificate or
1111 other security token associated with the signature of the SAML assertion before using the policy for
1112 XACML request evaluation. In this case the issuer of the SAML assertion SHALL be translated into an
1113 XACML attribute with id urn:oasis:names:tc:xacml:1.0:subject:subject-id. This does that
1114 mean that the issuer name must be taken directly from the security token, merely that the PDP perform
1115 some mapping on the claims in the token to determine the issuer.

1116 **6.2 Element <xacml-saml:ReferencedPolicies>**

1117 An instance of this element MAY be used to contain copies of policies referenced from <xacml:Policy>
1118 or <xacml:PolicySet> instances included in the <xacml-samlp:XACMLPolicyQuery>, as well as
1119 copies of policies referenced from other policies included in the <xacml-saml:ReferencedPolicies>
1120 instance.

1121 See Section 4.9 for a description of the <xacml-saml:ReferencedPolicies> element.

1122 6.3 Element <saml:Statement>: XACMLPolicy Statement

1123 A <saml:Statement> instance MAY be defined to be of type <xacml-
1124 saml:XACMLPolicyStatementType> by using xsi:type="xacml-
1125 saml:XACMLPolicyStatementType" as shown in the example in Section 6.4. such an instance of a
1126 <saml:Statement> element is called an XACMLPolicy Statement in this Profile. Any instance of an
1127 XACMLPolicy Statement in an XACML system MUST be enclosed in a <saml:Assertion>.

1128 6.4 Element <saml:Assertion>: XACMLPolicy Assertion

1129 A <saml:Assertion> instance MAY contain an XACMLPolicy Statement as shown in the following non-
1130 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>
```

1131 An instance of a <saml:Assertion> element containing an XACMLPolicy Statement is called an
1132 XACMLPolicy Assertion in this Profile.

1133 When an XACMLPolicy Assertion is part of a response to an <xacml-samlp:XACMLPolicyQuery>,
1134 then the XACMLPolicy Assertion MUST contain exactly one XACMLPolicy Statement, which in turn MAY
1135 contain any number of XACML Policy and PolicySet instances.

1136 This Profile imposes the following requirements and restrictions on the <saml:Assertion> element
1137 beyond those specified in SAML 2.0 when used as an XACMLPolicy Assertion.

1138 <saml:Issuer> [Required]

1139 The <saml:Issuer> element is a required element for holding information about “the SAML
1140 authority that is making the claim(s) in the assertion” [SAML].

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

1145 <ds:Signature> [Optional]

1146 The <ds:Signature> element is an optional element for holding “An XML Signature that
1147 authenticates the assertion, as described [in Section 5 of the SAML 2.0 core specification[SAML]].”

1148 A <ds:Signature> instance MAY be used in an XACMLPolicy Assertion. In order to support 3rd
1149 party digital signatures, this Profile does NOT require that the identity provided in the
1150 <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Assertion. It is up to the
1151 relying party to determine whether it has an appropriate trust relationship with the authority that signs
1152 the XACMLPolicy Assertion.

1153 A relying party SHOULD verify any signature included in the XACMLPolicy Assertion and SHOULD
1154 NOT use information derived from the XACMLPolicy Assertion unless the signature is verified
1155 successfully.

1156 <saml:Subject> [Optional]

1157 The <saml:Subject> element MUST NOT be included in an XACMLPolicy Assertion. Instead, the
1158 Subjects of an XACMLPolicy Assertion are specified in the XACML Policy and PolicySet elements
1159 contained in the enclosed XACMLPolicy Statement.

1160 <saml:Conditions> [Optional]

1161 The <saml:Conditions> element is an optional element that is used for “conditions that MUST be
1162 taken into account in assessing the validity of and/or using the assertion” [SAML].

1163 The <saml:Conditions> instance SHOULD contain NotBefore and NotOnOrAfter XML
1164 attributes to specify the limits on the validity of the XACMLPolicy Assertion. If these XML attributes
1165 are present, the relying party SHOULD ensure that an <xacml-context:Response> taken from
1166 the XACMLPolicy Assertion is used only during the XACMLPolicy Assertion's specified validity period.

1167 6.5 Element <xacml-samlp:XACMLPolicyQuery>

1168 An instance of the <xacml-samlp:XACMLPolicyQuery> protocol element MAY be used by a PDP or
1169 application to request XACML <xacml:Policy> or <xacml:PolicySet> instances from an on-line
1170 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>
```

1171 The <xacml-samlp:XACMLPolicyQuery> element is of <xacml-samlp:XACMLPolicyQueryType>
1172 complex type, which is an extension to the SAML-defined <samlp:RequestAbstractType>.

1173 The <xacml-samlp:XACMLPolicyQuery> element contains zero or more of the following elements in
1174 addition to those defined for the <samlp:RequestAbstractType>:

1175 <xacml-context:Request> [Any Number]

1176 An XACML Request Context. All XACML <xacml:Policy> and <xacml:PolicySet> instances
1177 potentially applicable to this Request that the requester is authorized to receive MUST be returned.
1178 The concept of “applicability” in the XACML context is defined in the XACML 3.0 Specification
1179 [XACML3]. Any superset of applicable policies MAY be returned; for example, all policies having
1180 top-level Target elements that match the Request MAY be returned.

1181 <xacml:PolicySetIdReference> [Any Number]

1182 Identifies an XACML <xacml:PolicySet> instance to be returned.

1183 <xacml:PolicyIdReference> [Any Number]

1184 Identifies an XACML `<xacml:Policy>` instance to be returned.

1185 *Non-normative note: The `<xacml-samlp:XACMLPolicyQuery>` is not intended as a robust*
1186 *provisioning protocol. Users requiring such a protocol may consider using the OASIS Service*
1187 *Provisioning Markup Language (SPML). Note that the SAML-defined ID XML attribute is a required*
1188 *component of an instance of `<samlp:RequestAbstractType>` that the `<xacml-`*
1189 *`samlp:XACMLPolicyQuery>` extends and MAY be used to correlate the `<xacml-`*
1190 *`samlp:XACMLPolicyQuery>` with the corresponding XACMLPolicy Response.*

1191 6.6 Element `<samlp:Response>`: XACMLPolicy Response

1192 A `<samlp:Response>` instance MAY contain an XACMLPolicy Assertion. An instance of such a
1193 `<samlp:Response>` element is called an XACMLPolicy Response in this Profile. An XACMLPolicy
1194 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>
```

1195 An instance of a `<samlp:Response>` element that contains an XACMLPolicy Assertion is called an
1196 XACMLPolicy Response in this Profile. Such a Response MUST be used as the response to an
1197 `<xacml-samlp:XACMLPolicyQuery>`. It MAY be used to convey or store XACML policies for other
1198 purposes.

1199 This Profile imposes the following requirements and restrictions on the `<samlp:Response>` element in
1200 addition to those specified in SAML 2.0 when used as an XACMLPolicy Response.

1201 `<saml:Issuer>` [Optional]

1202 The `<saml:Issuer>` element identifies the originator of the contained XACML Policy, which MAY be
1203 the entity that generated the XACMLPolicy Response message. [SAML].

1204 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
1205 in the `<saml:Issuer>` element refer to the entity that signs the XACMLPolicy Response. It is up to
1206 the relying party to determine whether it has an appropriate trust relationship with the authority that
1207 signs the XACMLPolicy Response.

1208 `<ds:Signature>` [Optional]

1209 The `<ds:Signature>` element is an optional element for holding "An XML Signature that
1210 authenticates the responder and provides message integrity" [SAML].

1211 A `<ds:Signature>` instance MAY be used in an XACMLPolicy Response. In order to support 3rd
1212 party digital signatures, this Profile does NOT require that the identity provided in the
1213 `<saml:Issuer>` instance refer to the entity that signs the XACMLPolicy Response. It is up to the
1214 relying party to determine whether it has an appropriate trust relationship with the authority that signs
1215 the XACMLPolicy Response.

1216 A relying party SHOULD verify any signature included in the XACMLPolicy Response and SHOULD
1217 NOT use information derived from the XACMLPolicy Response unless the signature is verified
1218 successfully.

1219 <saml:Assertion> [Any Number]

1220 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,
1221 then there MUST be exactly one instance of this element that contains an XACMLPolicy Assertion
1222 representing the response to the associated XACMLPolicy Query. If the XACMLPolicy Response is
1223 not issued in response to an <xacml-samlp:XACMLPolicyQuery>, it MAY contain one or more
1224 XACMLPolicy Assertions as well as other SAML or XACML Assertions.

1225 <saml:Status> [Required]

1226 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,
1227 and if it is not possible to return all policies that satisfy the <xacml-samlp:XACMLPolicyQuery>, then a
1228 <samlp:StatusCode> value of
1229 urn:oasis:names:tc:saml:2.0:status:TooManyResponses MUST be returned in the
1230 <samlp:Status> element of the Response.

1231 InResponseTo [Optional]

1232 This optional XML attribute is “A reference to the identifier of the request to which the response
1233 corresponds.” When the XACMLPolicy Response is issued in response to an <xacml-
1234 samlp:XACMLPolicyQuery>, this XML attribute MUST contain the value of the ID XML attribute
1235 from the <xacml-samlp:XACMLPolicyQuery> to which this is a response. This allows the
1236 receiver to correlate the XACMLPolicy Response with the corresponding XACMLPolicy Query.

1237 **6.7 Policy references and Policy assertions**

1238 It may be noted that in relation to a policy assertion, there are three broad classes of policies to consider
1239 when resolving policy references: the top level policy in the policy assertion, the policies in the <xacml-
1240 samlp:ReferencedPolicies> element and policies external to the policy assertion, available to a PDP by
1241 other means.

1242 How policy references are resolved across these three classes of policies depends on the particular case
1243 and problem for which the policy assertion is used. Therefore policy reference resolving is implementation
1244 defined with respect to policy assertions.

1245 7 Advice

1246 This Section describes how to include XACMLAuthzDecision Assertion and XACMLPolicy Assertion
1247 instances as advice in another SAML Assertion instance.

1248 7.1 Element <saml:Advice>

1249 A SAML Assertion MAY include a <saml:Advice> element containing "Additional information related to
1250 the assertion that assists processing in certain situations but which MAY be ignored [without affecting
1251 either the semantics or the validity of the assertion] by applications that do not understand the advice or
1252 do not wish to make use of it." [SAML] An XACMLAuthzDecision Assertion or XACMLPolicy Assertion
1253 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>
```


1254
1255

8 Using an XACML Authorization Decision as an Authorization Token

1256 This Section of the Profile describes how to use an XACMLAuthzDecision Statement as a security and
1257 privacy authorization token as part of a SOAP message exchange in a Web Services context. This token
1258 MAY be used by a client to convey an authorization decision from a trusted 3^d party to a service. A Web
1259 Service MAY use such a token to determine that the client is authorized to access information involved in
1260 the Web Services interaction.

1261 In a Web Services context, an instance of an XACMLAuthzDecision Assertion MAY be used as an
1262 authorization token in the Web Services Security [WSS] `wsse:Security` Header of a SOAP message.
1263 When used in this way, the XACMLAuthzDecision Statement in the XACMLAuthzDecision Assertion
1264 MUST include the corresponding XACML Request Context. This allows the Web service to determine
1265 whether the `<xacml-context:Attribute>` instances in the Request correspond to the access that the
1266 client requires as part of the Web Service interaction. The XACMLAuthzDecision Assertion SHOULD be
1267 signed by a Policy Decision Point trusted by the Web Service.

1268 A Web Service MAY use this token to determine that a trusted 3^d party has evaluated an XACML Request
1269 Context that is relevant to the invocation of the service, and has reported an authorization decision. The
1270 service SHOULD verify that the signature on the XACMLAuthzDecision Assertion is from a Policy
1271 Decision Point that the service trusts. The service SHOULD verify that the validity period of the
1272 XACMLAuthzDecision Assertion includes the time at which the Web Service interaction will access the
1273 information or resource to which the Request Context applies. The service SHOULD verify that the
1274 `<xacml-context:Attribute>` instances contained in the XACML `<xacml-context:Request>`
1275 element correctly describe the information or resource access that needs to be authorized as part of this
1276 Web Service interaction.

9 Conformance

1277

1278 Implementations of this Profile MAY implement certain subsets of the described functionality. Each
1279 implementation MUST clearly identify the subsets it implements using the following identifiers.

1280 An implementation of this Profile is a conforming *SAML Attribute* implementation if the implementation
1281 conforms to Section 2 of this Profile. The following URI MUST be used as the identifier for this
1282 functionality:

1283 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:attrs:all

1284 An implementation of this Profile is a conforming *SOAP Attributes as XACML Authz Decision Query*
1285 implementation if the implementation conforms to Section 3.1 of this Profile. The following URI MUST be
1286 used as the identifier for this functionality:

1287 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:authzQuery

1288 An implementation of this Profile is a conforming *SOAP Attributes as SAML Attribute Assertion*
1289 implementation if the implementation conforms to Section 3.2 of this Profile. The following URI MUST be
1290 used as the identifier for this functionality:

1291 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:attrAssertion

1292

1293 An implementation of this Profile is a conforming *XACML Authz Decision without Policies* implementation
1294 if the implementation conforms to all parts of Section 4 of this Profile excluding the <xacml:Policy>,
1295 <xacml:PolicySet>, and <xacml-samlp:ReferencedPolicies> elements and their sub-elements
1296 and the CombinePolicies XML attribute in the <xacml-samlp:XACMLAuthzDecisionQuery>.
1297 XACML 3.0 implementations MUST support the <xacml-samlp:AdditionalAttributes> element
1298 and its sub-elements in the <xacml-samlp:XACMLAuthzDecisionQuery>. XACML 1.0, 1.1, and 2.0
1299 implementations MUST NOT support the <xacml-samlp:AdditionalAttributes> element and its
1300 sub-elements in the <xacml-samlp:XACMLAuthzDecisionQuery>. The following URI MUST be
1301 used as the identifier for this functionality:

1302 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:noPolicies

1303 An implementation of this Profile is a conforming *XACML Authz Decision with Policies* implementation if
1304 the implementation conforms to all parts of Section 4 of this Profile. XACML 3.0 implementations MUST
1305 support the <xacml-samlp:AdditionalAttributes> element and its sub-elements in the <xacml-
1306 samlp:XACMLAuthzDecisionQuery>. XACML 1.0, 1.1, and 2.0 implementations MUST NOT support
1307 the <xacml-samlp:AdditionalAttributes> element and its sub-elements in the <xacml-
1308 samlp:XACMLAuthzDecisionQuery>. The following URI MUST be used as the identifier for this
1309 functionality:

1310 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:withPolicies

1311 An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust with Policies*
1312 implementation if it conforms to section 5 in its entirety as described in the previous paragraph. The
1313 following URI MUST be used as the identifier for this functionality.

1314 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:withP
1315 olicies

1316 An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust without Policies*
1317 *implementation if it conforms to section 5, with the exceptions relating to policies and additionl attributes*
1318 *noted above. The following URI MUST be used as the identifier for this functionality.*

1319 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:noPol`
1320 `icies`

1321 An implementation of this Profile is a conforming *XACML Policies* implementation if the implementation
1322 conforms to Section 6 of this Profile. The following URI MUST be used as the identifier for this
1323 functionality:

1324 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:policies`

1325 An implementation of this Profile is a conforming *SAML Advice* implementation if the implementation
1326 conforms to Section 7 of this Profile. The following URI MUST be used as the identifier for this
1327 functionality:

1328 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:adviceSAML`

1329 An implementation of this Profile is a conforming *XACML Authz Token* implementation if the
1330 implementation conforms to Section 8 of this Profile. The following URI MUST be used as the identifier
1331 for this functionality:

1332 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzToken`

1333

1334 **Appendix A. Acknowledgments**

1335 The following individuals have participated in the creation of this specification and are gratefully
1336 acknowledged

1337 **Participants:**

- 1338 • Anne Anderson, Sun Microsystems
- 1339 • Anthony Nadalin, IBM
- 1340 • Bill Parducci,
- 1341 • Carlisle Adams, University of Ottawa
- 1342 • Daniel Engovatov, BEA
- 1343 • Don Flinn,
- 1344 • Ed Coyne
- 1345 • Erik Rissanen
- 1346 • Ernesto Damiani
- 1347 • Frank Siebenlist
- 1348 • Gerald Brose
- 1349 • Hal Lockhart
- 1350 • Haruyuki Kawabe
- 1351 • James MacLean
- 1352 • John Merrells
- 1353 • Ken Yagen
- 1354 • Konstantin Beznosov
- 1355 • Michiharu Kudo
- 1356 • Michael McIntosh
- 1357 • Pierangela Samarati
- 1358 • Pirasenna Velandai Thiyagarajan
- 1359 • Polar Humenn
- 1360 • Rebekah Metz
- 1361 • Ron Jacobson
- 1362 • Satoshi Hada
- 1363 • Sekhar Vajjhala
- 1364 • Seth Proctor
- 1365 • Simon Godik
- 1366 • Steve Anderson
- 1367 • Steve Crocker
- 1368 • Suresh Damodaran
- 1369 • Tim Moses
- 1370 • Von Welch
- 1371 • Frederic Deleon
- 1372 • Argyn Kuketayev

1373

Appendix A. 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 <samlp: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 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>). -throughout: changed SHALL to MUST -throughout: added namespace designators to schema items

Rev	Date	By whom	What
			<p>and added additional namespace prefixes to list in Section 1.4</p> <ul style="list-style-type: none"> -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
WD 6		Erik Rissanen	<p>Added wording about deriving a policy issuer element from a saml assertion.</p> <p>Reworded requirements on the ReturnContext attribute.</p> <p>Changed some MAY/MUST statements.</p> <p>Fixed some TBDs.</p> <p>Changed order in which supplied policies are combined.</p> <p>Removed section about metadata.</p> <p>Fixed typos.</p> <p>Don't allow inheritance between supplied attributes in an</p>

Rev	Date	By whom	What
			authz query. Relax the constraints on the <ReferencedPolicies> element.
WD 7	23 March 2009	Hal Lockhart	Improved some wording from previous changes. Added WS-Trust based decision request and response. Removed Metadata conformance clause.

