



SAML 2.0 Profile of XACML, Version 2.0

Committee Draft 03

11 March 2010

Specification URIs:

This Version:

<http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-03-en.html>
<http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-03-en.odt> (Authoritative)
<http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cd-03-en.pdf>

Previous 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>

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:

- [Assertions and Protocols for the OASIS Security Assertion Markup Language\(SAML\)v 2.0 OASIS Standard](#)

- 33 • eXtensible Access Control Markup Language (XACML) Version 1.0, OASIS Standard
- 34 • eXtensible Access Control Markup Language (XACML) Version 2.0, OASIS Standard
- 35 • eXtensible Access Control Markup Language (XACML) Version 3.0, CD 03
- 36 • eXtensible Access Control Markup Language (XACML) Version 1.1, Committee Draft

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

38 urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:assertion:wd-13
39 urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:protocol:wd-13
40 urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:assertion:wd-13
41 urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:protocol:wd-13
42 urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:wd-13
43 urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:wd-13
44 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:assertion:wd-13
45 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:protocol:wd-13

46 **Abstract:**

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

57 **Status:**

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

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

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

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

72

Notices

73 Copyright © OASIS® 2010. All Rights Reserved.

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

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

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

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

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

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

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

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

Table of Contents

118	1 Introduction.....	6
119	1.1 Organization of this Profile.....	6
120	1.1 Diagram of SAML integration with XACML.....	8
121	1.2 Backwards compatibility.....	9
122	1.3 Terminology.....	9
123	1.1 Namespaces.....	11
124	1.2 Normative References.....	12
125	1.3 Non-normative References.....	13
126	2 Attributes.....	14
127	2.1 Element <saml:Attribute>.....	14
128	2.1 Element <saml:AttributeStatement>.....	16
129	2.2 Element <saml:Assertion>: SAML Attribute Assertion.....	16
130	2.3 Element <samlp:AttributeQuery>.....	17
131	2.4 Element <samlp:Response>: SAML Attribute Response.....	17
132	3 Conveying XACML Attributes in a SOAP Message.....	19
133	3.1 <xacml-samlp:XACMLAuthzDecisionQuery>.....	19
134	3.2 SAML Attribute Assertion.....	19
135	4 Authorization Decisions.....	20
136	4.1 Type <xacml-saml:XACMLAuthzDecisionStatementType>.....	20
137	4.2 Element <saml:Statement>: XACMLAuthzDecision Statement.....	21
138	4.3 Element <saml:Assertion>: XACMLAuthzDecision Assertion.....	21
139	4.4 Element <xacml-samlp:XACMLAuthzDecisionQuery>.....	23
140	4.5 Element <xacml-samlp:Extensions>.....	26
141	4.6 Element <xacml-samlp:AdditionalAttributes>.....	26
142	4.7 Element <xacml-samlp:AssignedAttributes>.....	27
143	4.8 Element <xacml-samlp:Holders>.....	27
144	4.9 Element <xacml-samlp:HolderAttributes>.....	28
145	4.10 Element <xacml-saml:ReferencedPolicies>.....	28
146	4.11 Element <samlp:Response>: XACMLAuthzDecision Response.....	29
147	4.12 Functional Requirements for the <xacml-samlp:AssignedAttributes> Element.....	31
148	5 XACML Decision Queries using WS-Trust.....	32
149	5.1 Common Claims Dialect.....	32
150	5.2 XACML Dialect.....	32
151	5.3 Decision Request.....	32
152	5.4 Decision Response.....	33
153	6 Policies.....	34
154	6.1 Type <xacml-saml:XACMLPolicyStatementType>.....	34
155	6.2 Element <xacml-saml:ReferencedPolicies>.....	35
156	6.3 Element <saml:Statement>: XACMLPolicy Statement.....	36

157	6.4 Element <saml:Assertion>: XACMLPolicy Assertion.....	36
158	6.5 Element <xacml-samlp:XACMLPolicyQuery>.....	37
159	6.6 Element <samlp:Response>: XACMLPolicy Response.....	38
160	6.7 Policy references and Policy assertions.....	39
161	7 Advice.....	40
162	7.1 Element <saml:Advice>.....	40
163	8 Using an XACML Authorization Decision as an Authorization Token.....	41
164	9 Conformance.....	42
165	Appendix A.Acknowledgments.....	44
166	Appendix B.Revision History.....	45

1 Introduction

167

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

169 *Non-normative through Section 1.3*

170 The OASIS eXtensible Access Control Markup Language [XACML] is a powerful, standard language that
171 specifies schemas for authorization policies and for authorization decision requests and responses. It
172 also specifies how to evaluate policies against requests to compute a response. A brief non-normative
173 overview of XACML is available in Error: Reference source not found.

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

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

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

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

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

201 1.1 Organization of this Profile

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

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

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

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

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

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

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

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

- 248 1. `xacml-saml:XACMLPolicyStatementType` – A new SAML extension type defined in this
249 Profile that MAY be used in an XACML system to create XACMLPolicy Statements that hold
250 XACML policies for storage or transmission.
- 251 2. <saml:Statement> – A standard SAML element that MUST be used to contain instances of the
252 `xacml-saml:XACMLPolicyStatementType`. An instance of such a <saml:Statement>
253 element is called an XACMLPolicy Statement in this Profile.
- 254 3. <saml:Assertion> – A standard SAML element that MUST be used to hold XACMLPolicy
255 Statement instances in an XACML system, either in a repository or in an XACMLPolicy Response.

256 An instance of such a `<saml:Assertion>` element is called an XACMLPolicy Assertion in this
257 Profile.

258 4. `<xacml-samlp:XACMLPolicyQuery>` – A new SAML extension protocol element defined in
259 this Profile that MAY be used by a PDP or other application to request XACML policies from a
260 Policy Administration Point (PAP).

261 5. `<samlp:Response>` – A standard SAML protocol element that MUST be used to return
262 XACMLPolicy Assertions in response to an `<xacml-samlp:XACMLPolicyQuery>`. An instance
263 of such a `<samlp:Response>` element is called an XACMLPolicy Response in this Profile.

264 Section 7 describes the use of XACMLAuthzDecision Assertion and XACMLPolicy Assertion instances as
265 advice in other SAML Assertions. The following element is described:

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

269 Section 8 describes the use of XACMLAuthzDecision Assertions as authorization tokens in a SOAP
270 message exchange.

271

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

273 **1.1 Diagram of SAML integration with XACML**

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

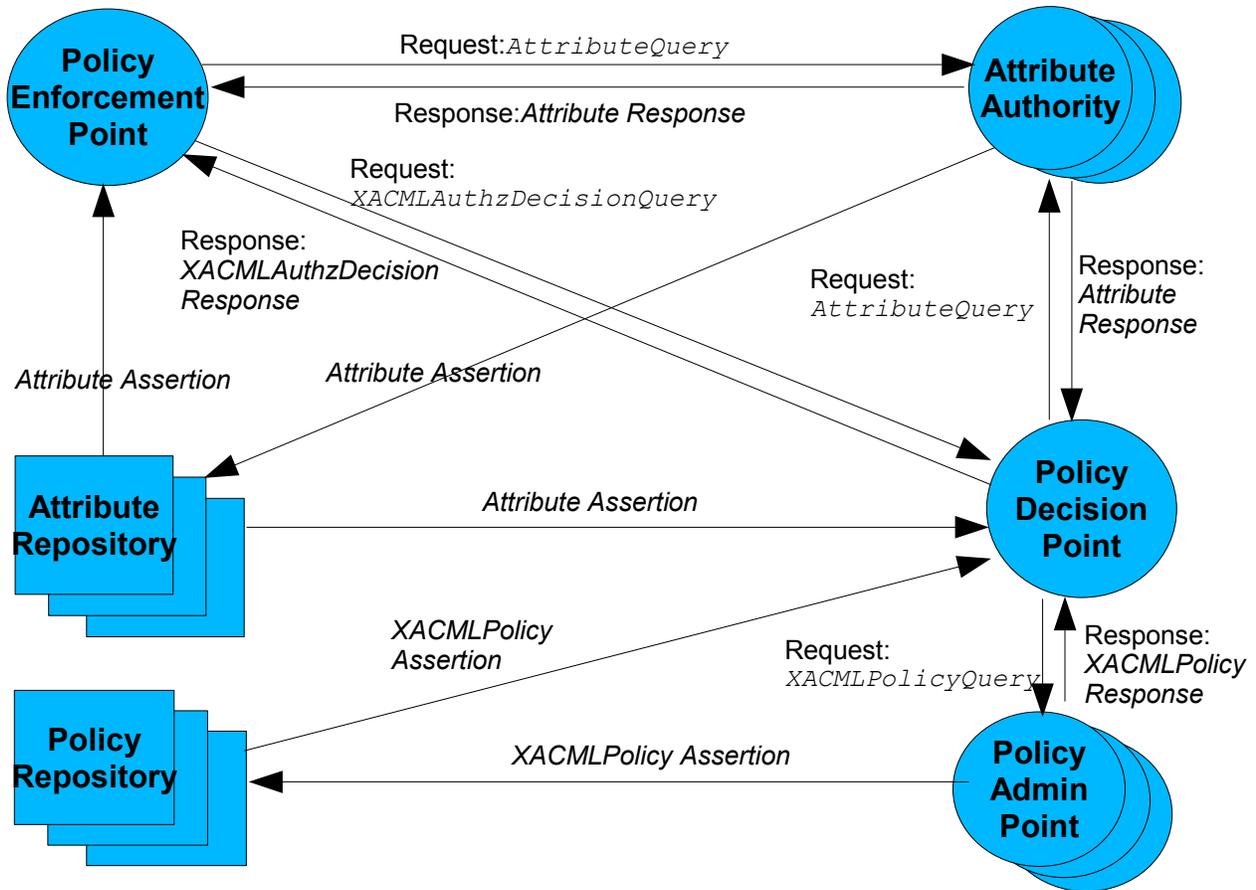


Figure 1: Components and messages in a integration of SAML with XACML

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

280 1.2 Backwards compatibility

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

284 1.3 Terminology

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

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

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

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

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

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

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

- 309 **Attribute Assertion** – A `<saml:Assertion>` instance that contains a `<saml:AttributeStatement>`
310 instance.

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

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

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

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

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

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

- 325 **XACMLAuthzDecision Response** – A `<samlp:Response>` instance that contains an
326 XACMLAuthzDecision Assertion.

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

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

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

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

333 1.1 Namespaces

334 *Normative*

335 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

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

339 XACML 1.0:

```
340 xacml="urn:oasis:names:tc:xacml:1.0:policy"  
341 xacml-context="urn:oasis:names:tc:xacml:1.0:context"  
342 xacml-saml=  
343 "urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:assertion:wd-13"  
344 xacml-samlp=  
345 "urn:oasis:names:tc:xacml:1.0:profile:saml2.0:v2:schema:protocol:wd-13"
```

347 XACML 1.1:

```
348 xacml="urn:oasis:names:tc:xacml:1.0:policy"  
349 xacml-context="urn:oasis:names:tc:xacml:1.0:context"  
350 xacml-  
351 saml="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:assertion:wd-13"  
352 xacml-  
353 samlp="urn:oasis:names:tc:xacml:1.1:profile:saml2.0:v2:schema:protocol:wd-13"
```

355 XACML 2.0:

```
356 xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"  
357 xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"  
358 xacml-  
359 saml="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:wd-13"  
360 xacml-  
361 samlp="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:wd-13"
```

363 XACML 3.0:
 364 xacml="urn:oasis:names:tc:xacml:3.0:schema:os"
 365 xacml-context="urn:oasis:names:tc:xacml:3.0:schema:os"

366 NOTE: XACML 3.0 uses a single schema for both policies and context.
 367 xacml-
 368 saml="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:assertion:wd-13"
 369 xacml-
 370 sampl="urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:schema:protocol:wd-13"

372 1.2 Normative References

- 373 **[ADMIN]** OASIS Committee Draft 03, *XACML v3.0 Administration and Delegation Profile*
 374 *Version 1.0*, 11 March 2010, [http://docs.oasis-open.org/xacml/3.0/xacml-3.0-](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-administration-v1-spec-cd-03-en.doc)
 375 [administration-v1-spec-cd-03-en.doc](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-administration-v1-spec-cd-03-en.doc)
- 376 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
 377 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 378 **[SAML]** OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion*
 379 *Markup Language (SAML) V2.0*, . 15 March 2005, [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
 380 [open.org/security/saml/v2.0/saml-core-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
- 381 **[SAML-PROFILE]** **OASIS Standard, Profiles for the OASIS Security Assertion Markup**
 382 **Language (SAML) V2.0, 15 March 2005**, [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
 383 [open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
- 384 **[XACML1]** OASIS Standard, *eXtensible Access Control Markup Language (XACML)*
 385 *Version 1.0*, 18 February 2003, [http://www.oasis-](http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf)
 386 [open.org/committees/download.php/2406/oasis-xacml-1.0.pdf](http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf)
- 387 **[XACML1.1]** OASIS Standard, *eXtensible Access Control Markup Language (XACML)*
 388 *Version 1.1*, 7 August 2003, [http://www.oasis-](http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf)
 389 [open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf](http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf)
- 390 **[XACML2]** OASIS, Standard, *eXtensible Access Control Markup Language (XACML)*
 391 *Version 2.0*, 1 February 2005, [http://docs.oasis-](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf)
 392 [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).
- 393 **[XACML3]** OASIS Committee Draft 03, *eXtensible Access Control Markup Language*
 394 *(XACML) Version 3.0*, 11 March 2010, [http://docs.oasis-](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cd-03-en.doc)
 395 [open.org/xacml/3.0/xacml-3.0-core-spec-cd-03-en.doc](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cd-03-en.doc)
- 396 **[XACML-SAML]** the schemas associated with namespace <xacml-saml> that are a normative
 397 part of this Profile.
- 398 **[XACML-SAMPLP]** the schemas associated with namespace <xacml-samplp> that are a normative
 399 part of this Profile.
- 400 **[WSFED]** OASIS Committee Draft 02, *Web Services Federation Language (WS-*
 401 *Federation) Version 1.2*, January 7, 2009 [http://docs.oasis-](http://docs.oasis-open.org/wsfed/federation/v1.2/cd/ws-federation-1.2-spec-cd-02.doc)
 402 [open.org/wsfed/federation/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)
- 403 **[WSS]** OASIS Standard, *Web Services Security: SOAP Message Security 1.0 (WS-*
 404 *Security 2004)*, March 2004, [http://docs.oasis-open.org/wss/2004/01/oasis-](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf)
 405 [200401-wss-soap-message-security-1.0.pdf](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf)
- 406 **[WSS-Core]** OASIS Standard, *WS-Security Core Specification 1.1*, February 2006,
 407 [http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)
 408 [SOAPMessageSecurity.pdf](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)
- 409 **[WSTRUST]** OASIS Standard, *WS-Trust 1.4*, 2 February 2009, [http://docs.oasis-open.org/ws-](http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.doc)
 410 [sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.doc](http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.doc)

411

412 **1.3 Non-normative References**

413 None

414

415

416

2 Attributes

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

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

438 2.1 Element `<saml:Attribute>`

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

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

445 2.1.1 Mapping a `<saml:Attribute>` to an `<xacml-context:Attribute>`

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

- 451 • XACML `AttributeId` XML attribute

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

- 453 • XACML `DataType` XML attribute

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

- 457 • XACML `Issuer` XML attribute

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

459 • `<xacml-context:AttributeValue>`

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

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

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

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

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

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

490 • Any `NotBefore` and `NotOnOrAfter` XML attributes in the SAML Attribute Assertion MUST be valid
491 with respect to the `<xacml:Request>` in which the SAML-derived `<xacml:Attribute>` is used.
492 This means that the XACML Attributes associated with the following `AttributeId` values in the
493 `<xacml:Request>` MUST represent times and dates that are not before the `NotBefore` XML
494 attribute value and not on or after the `NotOnOrAfter` XML attribute value:
495 `urn:oasis:names:tc:xacml:1.0:environment:current-time`
496 `urn:oasis:names:tc:xacml:1.0:environment:current-date`
497 `urn:oasis:names:tc:xacml:1.0:environment:current-dateTime`

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

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

503 **2.1 Element <saml:AttributeStatement>**

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

507 **2.2 Element <saml:Assertion>: SAML Attribute Assertion**

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

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

515 <saml:Issuer> [Required]

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

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

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

526 <ds:Signature> [Optional]

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

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

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

537 <saml:Subject> [Optional]

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

541 In a SAML Attribute Assertion containing a <saml:Attribute> that is to be mapped to an <xacml-
542 context:Attribute>, the <saml:Subject> instance MUST contain the identity of the entity to
543 which the <saml:Attribute> and its value are bound. For a mapped <saml:Attribute> to be
544 placed in a given XACML Attribute group, this identity SHOULD refer to the same entity as any
545 XACML Attribute that serves as an entity identifier in the Attribute group. For example, the

546 <saml:Subject> associated with a mapped SAML->XACML Attribute to be placed in the
547 XACML <xacml-context:Resource> Attribute group SHOULD refer to the same entity as the
548 value of any XACML Attribute having an AttributeId of
549 urn:oasis:names:tc:xacml:1.0:resource:resource-id that occurs in the same <xacml-
550 context:Resource> instance. See Section 2.1 for more information.

551 <saml:Conditions> [Optional]

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

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

564 2.3 Element <samlp:AttributeQuery>

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

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

572 2.4 Element <samlp:Response>: SAML Attribute Response

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

580 <saml:Issuer> [Optional]

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

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

587 <ds:Signature> [Optional]

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

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

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

598

3 Conveying XACML Attributes in a SOAP Message

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

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

609 3.1 `<xacml-samlp:XACMLAuthzDecisionQuery>`

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

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

623 3.2 SAML Attribute Assertion

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

629

4 Authorization Decisions

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

- 641 • `<xacml-saml:XACMLAuthzDecisionStatementType>` is a new SAML extension type that
642 includes an XACML `<xacml-context:Response>` along with other optional information.
- 643 • A `<saml:Statement>` of type `<xacml-saml:XACMLAuthzDecisionStatementType>` (defined
644 using `xsi:type`) MAY be used by a PDP Context Handler to convey an XACML `<xacml-`
645 `context:Response>` along with other optional information. An instance of such a
646 `<saml:Statement>` element is called an XACMLAuthzDecision Statement in this Profile.
- 647 • A `<saml:Assertion>` MUST be used to hold XACMLAuthzDecision Statements. An instance of
648 such a `<saml:Assertion>` element is called an XACMLAuthzDecision Assertion in this Profile.
- 649 • A `<xacml-samlp:XACMLAuthzDecisionQuery>` is a new SAML extension element that MAY be
650 used by a PEP to submit an XACML Request Context, along with other optional information, as a
651 SAML protocol query to an XACML Context Handler.
- 652 • A `<samlp:Response>` containing an XACMLAuthzDecision Assertion MUST be used by an XACML
653 Context Handler as the response to an `<saml-samlp:XACMLAuthzDecisionQuery>`. An instance
654 of such a `<samlp:Response>` element is called an XACMLAuthzDecision Response in this Profile.

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

657 4.1 Type `<xacml-saml:XACMLAuthzDecisionStatementType>`

658 The new `<xacml-saml:XACMLAuthzDecisionStatementType>` complex type contains an XACML
659 Response Context along with related information. Use of this type is an alternative to use of the SAML-
660 defined `<saml:AuthzDecisionStatementType>`; this alternative allows an XACML Context Handler
661 to use SAML with full support for XACML authorization decisions. An instance of a `<saml:Statement>`
662 element that is of this type (defined using `xsi:type="xacml-`
663 `saml:XACMLAuthzDecisionStatementType"`) is called an XACMLAuthzDecision Statement in this
664 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>

```

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

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

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

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

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

674 If the XACMLAuthzDecision Statement represents a response to an `<xacml-`
 675 `samlp:XACMLAuthzDecisionQuery>`, and if the `ReturnContext` XML attribute in the `<xacml-`
 676 `samlp:XACMLAuthzDecisionQuery>` instance is "true", then this element MUST be included; if
 677 the `ReturnContext` XML attribute in the `<xacml-samlp:XACMLAuthzDecisionQuery>` instance
 678 is "false", then this element MUST NOT be included. See the description of the `ReturnContext`
 679 XML attribute in Section 4.4 for a specification of the `<xacml-context:Attribute>` instances that
 680 MUST be returned in this element when it is part of a response to an `<xacml-`
 681 `samlp:XACMLAuthzDecisionQuery>`.

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

686 4.2 Element `<saml:Statement>`: XACMLAuthzDecision Statement

687 A `<saml:Statement>` instance MAY be of type `<xacml-`
 688 `saml:XACMLAuthzDecisionStatementType>` by using `xsi:type` as shown in the example in
 689 Section 4.3. An instance of a `<saml:Statement>` element that is of type `<xacml-`
 690 `saml:XACMLAuthzDecisionStatementType>` is called an XACMLAuthzDecision Statement in this
 691 Profile. Any instance of an XACMLAuthzDecision Statement in an XACML system MUST be enclosed in
 692 a `<saml:Assertion>`.

693 4.3 Element `<saml:Assertion>`: XACMLAuthzDecision Assertion

694 A `<saml:Assertion>` instance MAY contain an XACMLAuthzDecision Statement as shown in the
 695 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>

```

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

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

700 `<saml:Issuer>` [Required]

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

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

707 `<ds:Signature>` [Optional]

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

710 A `<ds:Signature>` instance MAY be used in a `<saml:Assertion>`. In order to support 3rd party
 711 digital signatures, this Profile does NOT require that the identity provided in the `<saml:Issuer>`
 712 instance refer to the entity that signs the XACMLAuthzDecision Assertion. It is up to the relying party
 713 to determine whether it has an appropriate trust relationship with the authority that signs the Assertion
 714 .

715 A relying party SHOULD verify any signature included in the XACMLAuthzDecision Assertion and
 716 SHOULD NOT use information derived from the Assertion unless the signature is verified
 717 successfully.

718 `<saml:Subject>` [Optional]

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

723 `<saml:Conditions>` [Optional]

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

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

731 **4.4 Element <xacml-samlp:XACMLAuthzDecisionQuery>**

732 The <xacml-samlp:XACMLAuthzDecisionQuery> protocol element MAY be used by a PEP to
733 request an authorization decision from an XACML PDP. This element is an alternative to the SAML-
734 defined <samlp:AuthzDecisionQuery>; this alternative allows the PEP to use the full capabilities of
735 an XACML PDP. It allows use of the SAML query protocol to convey an XACML Request Context along
736 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" />
        <xs:any namespace="##any" processContents="strict"
minOccurs="0" maxOccurs="unbounded"/>
      </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>

```

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

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

742 `InputContextOnly` [Default "false"]

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

749 `ReturnContext` [Default "false"]

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

753 If this attribute is "True", then the PDP SHALL include the `<xacml-context:Request>` element in
754 the `<XACMLAuthzDecisionStatement>` element in the `<XACMLResponse>`. This `<xacml-`
755 `context:Request>` element SHALL include all those XACML Attributes supplied by the PEP in the
756 `<XACMLAuthzDecisionQuery>` that were used in making the authorization decision. A conforming

757 PDP MAY omit those XACML Attributes which were not referenced in any policy which was evaluated
758 in making the decision. If the value of the `InputContextOnly` Attribute in the Request is "False", the
759 PDP MAY include additional XACML Attributes in this `<xacml-context:Request>` element, which
760 were obtained by the PDP and used in making the authorization decision.

761

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

764 `CombinePolicies` [Default "true"]

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

768 If the attribute value is "true", then the PDP MUST insert all policies passed in the `<xacml:Policy>`
769 and `<xacml:PolicySet>` elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>` into
770 the set of policies or policy sets that define the PDP as specified in Section 7.11 of the XACML 3.0
771 core specification [XACML3]. They MUST be combined with the other policies using the policy
772 combining algorithm that defines the PDP as specified in Section 7.11 of the XACML 3.0 core
773 specification [XACML3]. If the policy combining algorithm that defines the PDP is one in which
774 element order is considered, then the policies passed in the XACMLAuthzDecision Query MUST be
775 considered in the order in which they appear in the `<xacml-samlp:XACMLAuthzDecisionQuery>`
776 and MUST be considered as preceding all other policies that define the PDP.

777

778 If the attribute value is "false", then there MUST be no more than one `<xacml:Policy>` or
779 `<xacml:PolicySet>` passed in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. This policy
780 MUST be treated as the policy that defines the PDP as specified in Section 7.11 of the XACML 3.0
781 core specification [XACML3] for evaluation of the `<xacml-context:Request>` passed in the
782 `<xacml-samlp:XACMLAuthzDecisionQuery>`. It MUST NOT be used to evaluate any other `<xacml-`
783 `context:Request>` instances unless provided to the PDP independent of the particular `<xacml-`
784 `context:Request>`.

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

786 An XACML Request Context that is to be evaluated.

787 `<xacml-samlp:AdditionalAttributes>` [Zero or One]

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

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

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

793 If the `CombinePolicies` XML attribute is "true", then the PDP MUST use such XACML Policy
794 instances.

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

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

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

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

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

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

807 With the exception of XACML Policy and PolicySet instances that the receiver of the
808 XACMLAuthzDecision Statement is not authorized to view, this element MAY contain XACML Policy
809 and PolicySet instances required to resolve `<xacml:PolicySetIdReference>` or
810 `<xacml:PolicyIdReference>` instances contained in the XACMLAuthzDecision Statement,
811 including those in the `<xacml-saml:ReferencedPolicies>` instance itself, or contained in the
812 policies already available to the PDP. The values of the `PolicyId` and `PolicySetId` XML
813 attributes of the policies included in the `<xacml-saml:ReferencedPolicies>` instance MUST
814 exactly match the values contained in the corresponding `<xacml:PolicySetIdReference>` or
815 `<xacml:PolicyIdReference>` instances.

816 `<xacml-saml:Extensions>` [Optional]

817 Contains extension points which MAY be used by profiles which extend this profile.

818 4.5 Element `<xacml-samlp:Extensions>`

819 This element is used to carry an extension point to the protocols.

```
820 <element name="Extensions" xsi:type="xacml-samlp:ExtensionsType" />  
821 <complexType name="ExtensionsType">  
822   <sequence>  
823     <any namespace="##any" processContents="strict" minOccurs="0"  
824       maxOccurs="unbounded"/>  
825   </sequence>  
826 </complexType>
```

827 The `<xacml-samlp:Extensions>` element contains the following XML elements:

828 `xs:any` [Any Number]

829 An extension point which MAY be used by profiles which extend this profile. For instance, this
830 extension point MAY be used to provide policies in other formats than XACML in environments which
831 are not purely XACML based, but want to reuse the query/response protocol of XACML. An
832 implementation MUST reject an instance of an `<XACMLAuthzDecisionQuery>` element if it does
833 not understand all elements which appear at this extension point. A rejected instance MUST be
834 answered with an XACML Indeterminate result with a status code of
835 `urn:oasis:names:tc:xacml:1.0:status:syntax-error`.

836 4.6 Element `<xacml-samlp:AdditionalAttributes>`

837 This element applies only for use with XACML 3.0 Administrative Policy [ADMIN], and requires an XACML
838 3.0 PDP.

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

```

845 <element name="AdditionalAttributes"
846   type="xacml-samlp: AdditionalAttributesType"/>
847 <complexType name="AdditionalAttributesType">
848   <sequence>
849     <element ref="xacml-samlp:AssignedAttributes" minOccurs="0"
850     maxOccurs="unbounded"/>
851   </sequence>
852 </complexType>

```

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

854 The <AdditionalAttributes> element contains the following elements:

855 <AssignedAttributes> [Required]

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

857 4.7 Element <xacml-samlp:AssignedAttributes>

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

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

```

862 <element name="AssignedAttributes" type="xacml-samlp:AssignedAttributesType"/>
863 <complexType name="AssignedAttributesType">
864   <sequence>
865     <element ref="xacml-samlp: Holders"/>
866     <element ref="xacml-samlp: HolderAttributes"/>
867   </sequence>
868 </complexType>

```

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

870 The <AssignedAttributes> element contains the following elements:

871 <xacml-samlp: Holders> [Required]

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

873 <xacml-samlp: HolderAttributes> [Required]

874 The XACML Attributes of the delegate entity.

875 4.8 Element <xacml-samlp: Holders>

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

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

```

880 <element name=" Holders" type="xacml-samlp: HoldersType"/>
881 <complexType name=" HoldersType">
882   <sequence>
883     <element ref="xacml: Match" maxOccurs="unbounded"/>
884   </sequence>
885 </complexType>

```

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

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

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

889 Matches the delegate entities to which the XACML Attributes in the associated `<xacml-samlp:HolderAttributes>` element apply. The `<Match>` elements shall be
890 evaluated according to the XACML schema against the `<Attributes>` elements in a
891 `<Request>` during reduction. If any `<Match>` element evaluates to "Match" then the
892 supplied attributes shall apply to the `<Attributes>` element which was referenced by the
893 attribute designator or selector contained in the `<Match>` element
894

895

896 **4.9 Element `<xacml-samlp:HolderAttributes>`**

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

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

```
901 <element name="HolderAttributes" type="xacml-samlp:HolderAttributesType"/>  
902 <complexType name="HolderAttributesType">  
903   <sequence>  
904     <element ref="xacml-context:Attribute"  
905       minOccurs="0" maxOccurs="unbounded"/>  
906   </sequence>  
907 </complexType>
```

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

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

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

912 An XACML Attribute of the delegate entities identified in the corresponding `<xacml-samlp:HolderAttributes>` element.
913

914 **4.10 Element `<xacml-saml:ReferencedPolicies>`**

915 An instance of this element MAY be used to contain copies of policies referenced from `<xacml:Policy>`
916 or `<xacml:PolicySet>` instances included in an XACML Authz Decision Statement or in an
917 XACML Policy Statement, as well as copies of all policies referenced from other policies included in the
918 `<xacml-saml:ReferencedPolicies>` instance or policies already present in the PDP. If a
919 `<xacml:Policy>` or `<xacml:PolicySet>` instance would match a policy both among the policies
920 already present to the PDP as well as a policy contained in the supplied `<xacml-saml:ReferencedPolicies>`
921 instance, then the supplied policy takes precedence.

```
922 <element name="ReferencedPolicies"  
923   type="xacml-saml:ReferencedPoliciesType"/>  
924 <complexType name="ReferencedPoliciesType">  
925   <sequence>  
926     <choice minOccurs="0" maxOccurs="unbounded">  
927       <element ref="xacml:Policy"/>  
928       <element ref="xacml:PolicySet"/>  
929     </choice>  
930   </sequence>  
931 </complexType>
```

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

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

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

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

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

941 A single `<xacml:PolicySet>` that is referenced using an `<xacml:PolicySetIdReference>`
942 from another `<xacml:Policy>` or `<xacml:PolicySet>` instance. The value of the `PolicySetId`
943 XML attribute in the `<xacml:PolicySet>` MUST be equal to the value of the corresponding
944 `<xacml:PolicySetIdReference>` element.

945 **4.11 Element `<samlp:Response>`: XACMLAuthzDecision Response**

946 A `<samlp:Response>` instance MAY contain an XACMLAuthzDecision Assertion as shown in the
947 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>
```

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

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

953 `<saml:Issuer>` [Optional]

954 The `<saml:Issuer>` element is an optional element that "Identifies the entity that generated the
955 response message" [SAML].

956 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
957 in the `<saml:Issuer>` element refer to the entity that signs the XACMLAuthzDecision Response. It

958 is up to the relying party to determine whether it has an appropriate trust relationship with the authority
959 that signs the Response.

960 <ds:Signature> [Optional]

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

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

968 A relying party SHOULD verify any signature included in the XACMLAuthzDecision Response and
969 SHOULD NOT use information derived from the Response unless the signature is verified
970 successfully.

971 <saml:Assertion> [Any Number]

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

974 <samlp:StatusCode> [Required]

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

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

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

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

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

984 This value for the <samlp:StatusCode> Value XML attribute MUST be used when the
985 <xacml-context:StatusCode> value is
986 urn:oasis:names:tc:xacml:1.0:status:missing-attribute or when the <xacml-
987 context:StatusCode> value is urn:oasis:names:tc:xacml:1.0:status:syntax-
988 error due to a syntax error in the <xacml-context:Request>.

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

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

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

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

999 InResponseTo [Optional]

1000 This optional XML attribute is “A reference to the identifier of the request to which the response
1001 corresponds.” When the XACMLAuthzDecision Response is issued in response to an
1002 XACMLAuthzDecision Query, this XML attribute MUST contain the value of the ID XML attribute
1003 from the XACMLAuthzDecision Query to which this is a response. This allows the receiver to
1004 correlate the XACMLAuthzDecision Response with the corresponding XACMLAuthzDecision
1005 Query. The SAML-defined ID XML attribute is a required component of an instance of the
1006 <samlp:RequestAbstractType> of which the <xacml-
1007 samlp:XACMLAuthzDecisionQuery> is an extension.

1008 **4.12 Functional Requirements for the <xacml- 1009 samlp:AssignedAttributes> Element**

1010 During processing of the provided access request, if the <xacml-samlp: HOLDERS> element of a
1011 provided <xacml-samlp:AssignedAttributes> element matches a section of the XACML Request
1012 Context, then the XACML Context Handler MUST make the XACML Attributes in the <xacml-
1013 samlp:HolderAttributes> element appear in that section of the XACML Request Context. Any
1014 inheritance between <xacml-samlp:AssignedAttributes> elements is not deduced.

1015 The matching of additional XACML Attributes MUST be made against all Request Contexts involved in the
1016 processing of the XACMLAuthzDecision Query, including the provided access request itself and any
1017 Request Contexts formed as part of reduction.

1018 The provided XACML Attributes MUST be used only in the evaluation of the provided access request and
1019 any derived Request Contexts, including reduction, and MUST NOT be used in evaluation of requests not
1020 related to the provided access request unless associated with those other requests independent of the
1021 <xacml-samlp:XACMLAuthzDecisionQuery>.

1022 The implementation MUST match the <xacml-samlp: HOLDERS> element against all the attributes
1023 available to the context handler, but MUST NOT use any matching <xacml-
1024 samlp:HolderAttributes> to find even more attributes through the context handler or even more
1025 supplied attributes through other <xacml-samlp: HOLDERS> elements. This implies that there can be no
1026 inheritance between <xacml-samlp:AssignedAttributes> elements.

1027 5 XACML Decision Queries using WS-Trust

1028 In some environments, it may be desirable to obtain an XACML authorization decision from a Security
1029 Token Service (STS) using the WS-Trust protocol [WSTRUST].

1030 5.1 Common Claims Dialect

1031 One method of doing this is to support the Common Claim Dialect as defined in WS-Federation [WSFED],
1032 chapter 9. In this case the implementation must map the contents of an incoming
1033 <RequestSecurityToken> element into a XACML <Request> element and map the XACML <Response>
1034 into an outgoing <RequestSecurityTokenResponseCollection> element. When this approach is taken,
1035 there is no explicit reference to XACML in the wire protocol and in general a requestijg party will not be
1036 aware whether or not an XACML-based PDP was used to make the decision.

1037 5.2 XACML Dialect

1038 This section defines a WS-Trust-based protocol which is intended to be easier and more efficient for XACML
1039 PDP to implement. It is based directly on the constructs previously defined in Section 4. It uses the
1040 <saml:Assertion> element and <saml:Statement> of type xacml-
1041 saml:XACMLAuthzDecisionStatementType to wrap the XACML <Request> and <Response> elements.
1042 However, the <xacml-samlp:XACMLDecisionQuery> and <samlp:Response> elements are not used.
1043 Instead the request is conveyed in a <wst:RequestSecurityToken> element and the response is carried in
1044 a <wst:RequestSecurityTokenResponseCollection> element containing a
1045 <wst:RequestSecurityTokenResponse> element.

1046 Except for the outer protocol layer, described in more detail below, the syntax and functional requirements
1047 for this protocol is exactly as described above in section 4. In fact, it is possible for a server which contains
1048 an XACML PDP to support both protocols, using distinct web service endpoints, with only a small amount
1049 of distinct code to handle each request type.

1050 5.3 Decision Request

1051 The decision request is contained in a <wst:RequestSecurityToken> element. This element contains the
1052 following attributes and elements from the WS-Trust schema.

- 1053 • Context This URI specifies an identifier for this request. Its value will be returned in the
1054 corresponding response to allow them to be correlated.
- 1055 • <wst:TokenType> This element contains the value: urn:oasis:names:tc:xacml:3.0:core:schema,
1056 to indicate that an XACML decision token will be returned.
- 1057 • <wst:RequestType> This element contains the value: [http://docs.oasis-open.org/ws-sx-ws-
1058 trust/200512/Issue](http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue)

1059 In addition, the <wst:RequestSecurityToken> element MAY contain any of the attributes and elements
1060 defined in section 4.4 above as being contained in the <xacml-samlp:XACMLAuthzDecisionQuery>
1061 element. Specifically these are the attributes:

- 1062 • InputContextOnly,
- 1063 • ReturnContext, and
- 1064 • CombinePolicies.

1065 These are the elements:

- 1066 • <xacml-context:Request>,
- 1067 • <xacml-samlp:AdditionalAttributes>,
- 1068 • <xacml:Policy>,
- 1069 • <xacml:PolicySet>, and
- 1070 • <xacml-saml:ReferencedPolicies>.

1071 The functional requirements for processing these attributes and elements are exactly as set forth in
1072 section 4 above.

1073 **5.4 Decision Response**

1074 The decision response is contained in a <wst:RequestTokenResponseCollection> element. It contains
1075 exactly one <wst:RequestTokenResponse> element. This element contains the following attributes and
1076 elements.

- 1077 • Context This element contains the same URI provided in the Context attribute of the request.
- 1078 • <wst:RequestedSecurityToken> This element contains a <saml:Assertion which in turn contains a
1079 <saml:Statement of type xacml-saml:XACMLAuthzDecisionStatementType as described in
1080 sections 4.1, 4.2, and 4.3 above. The functional requirements for processing these attributes and
1081 elements are exactly as set forth in section 4 above.

1082

6 Policies

1083 XACML defines the `<xacml:Policy>` and `<xacml:PolicySet>` elements for expressing policies. In
1084 many environments, instances of these elements need to be stored or transmitted between entities in an
1085 XACML system. Such instances may need to be signed or associated with a validity period. SAML is
1086 intended to provide this functionality for security-related assertions, but SAML does not define any
1087 Protocol or Assertion elements for policies. In order to allow entities in an XACML system to use SAML
1088 assertions and protocols to store, transmit, and query for XACML policies, this Profile defines one SAML
1089 extension type and one SAML extension element, and describes how they are used with other standard
1090 SAML elements.

1091 • `<xacml-saml:XACMLPolicyStatementType>` is a new SAML extension type that includes XACML
1092 policies.

1093 • A `<saml:Statement>` defined using `xsi:type="xacml-saml:XACMLPolicyStatementType"`
1094 MAY be used in an XACML system to store or convey XACML policies. An instance of a
1095 `<saml:Statement>` element defined using this type is called an XACML Policy Statement in this
1096 Profile.

1097 • A `<saml:Assertion>` MUST be used to hold XACML Policy Statements. An instance of such a
1098 `<saml:Assertion>` element is called an XACML Policy Assertion in this Profile.

1099 • An `<xacml-samlp:XACMLPolicyQuery>` is a new SAML extension element that MAY be used by a
1100 PDP or other entity to request XACML policies as a SAML protocol query.

1101 • A `<samlp:Response>` containing an XACML Policy Assertion that MUST be used in response to an
1102 `<xacml-samlp:XACMLPolicyQuery>`. It MAY be used to transmit XACML policies in other
1103 contexts. An instance of such a `<samlp:Response>` is called an XACML Policy Response in this
1104 Profile.

1105 This Section defines and describes the usage of these types and elements. The schemas for the new
1106 type and element are contained in the [XACML-SAML] and [XACML-SAML] schema documents.

1107 6.1 Type `<xacml-saml:XACMLPolicyStatementType>`

1108 The `<xacml-saml:XACMLPolicyStatementType>` complex type contains XACML Policy and or
1109 XACML PolicySet elements. An instance of a `<saml:Statement>` element that is of this type is called
1110 an XACML Policy 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>
```

1111 The `<xacml-saml:XACMLPolicyStatementType>` complex type is an extension to the SAML-defined
1112 `<saml:StatementAbstractType>`. It contains the following elements.

1113 `<xacml:Policy>` [Any Number]

1114 If the XACMLPolicy Statement represents a response to an `<xacml-sampl:XACMLPolicyQuery>`,
1115 then this element MUST contain one of the `<xacml:Policy>` instances that meet the specifications
1116 of the associated `<xacml-sampl:XACMLPolicyQuery>`. Otherwise, this element MAY contain an
1117 arbitrary `<xacml:Policy>` instance.

1118 `<xacml:PolicySet>` [Any Number]

1119 If the XACMLPolicy Statement represents a response to an `<xacml-sampl:XACMLPolicyQuery>`,
1120 then this element MUST contain one of the `<xacml:PolicySet>` instances that meet the
1121 specifications of the associated `<xacml-sampl:XACMLPolicyQuery>`. Otherwise, this element
1122 MAY contain an arbitrary `<xacml:PolicySet>` instance.

1123 `<xacml-saml:ReferencedPolicies>` [Zero or One]

1124 With the exception of XACML Policy and PolicySet instances that the receiver of the XACMLPolicy
1125 Statement is not authorized to view, this element MAY contain XACML Policy and PolicySet instances
1126 required to resolve `<xacml:PolicySetIdReference>` or `<xacml:PolicyIdReference>`
1127 instances contained in the XACMLPolicy Statement, including those in the `<xacml-
1128 saml:ReferencedPolicies>` instance itself. The values of the `PolicyId` and `PolicySetId`
1129 XML attributes of the policies included in the `<xacml-saml:ReferencedPolicies>` instance
1130 MUST exactly match the values contained in the corresponding
1131 `<xacml:PolicySetIdReference>` or `<xacml:PolicyIdReference>` instances.

1132 Subject to authorization and availability, if the XACMLPolicy Statement is issued in response to an
1133 `<xacml-sampl:XACMLPolicyQuery>`, there MUST be exactly one `<xacml:Policy>` element
1134 included for every XACML Policy that satisfies the XACMLPolicy Query, and there MUST be exactly one
1135 `<xacml:PolicySet>` element included for every XACML PolicySet that satisfies the XACMLPolicy
1136 Query. The responder MUST return all XACML policies available to the responder that satisfy the
1137 `<xacml-sampl:XACMLPolicyQuery>` and that the requester is authorized to receive.

1138 If the XACMLPolicy Statement is issued in response to an `<xacml-sampl:XACMLPolicyQuery>`, and
1139 there are no `<xacml:Policy>` or `<xacml:PolicySet>` instances that meet the specifications of the
1140 associated `<xacml-sampl:XACMLPolicyQuery>`, then there MUST be exactly one empty
1141 XACMLPolicy Statement included in the response.

1142 An XACMLPolicy Statement enclosed in a signed SAML assertion MAY be used as a method of
1143 authentication of XACML policies. In this case the Policy or PolicySet MUST NOT contain an XACML
1144 `<PolicyIssuer>` element. Instead the PDP MAY generate a `<PolicyIssuer>` element from the certificate or
1145 other security token associated with the signature of the SAML assertion before using the policy for
1146 XACML request evaluation. In this case the issuer of the SAML assertion SHALL be translated into an
1147 XACML attribute with id `urn:oasis:names:tc:xacml:1.0:subject:subject-id`. This does that
1148 mean that the issuer name must be taken directly from the security token, merely that the PDP perform
1149 some mapping on the claims in the token to determine the issuer.

1150 **6.2 Element `<xacml-saml:ReferencedPolicies>`**

1151 An instance of this element MAY be used to contain copies of policies referenced from `<xacml:Policy>`
1152 or `<xacml:PolicySet>` instances included in the `<xacml-sampl:XACMLPolicyQuery>`, as well as
1153 copies of policies referenced from other policies included in the `<xacml-saml:ReferencedPolicies>`
1154 instance.

1155 See Section 4.10 for a description of the `<xacml-saml:ReferencedPolicies>` element.

1156 **6.3 Element <saml:Statement>: XACMLPolicy Statement**

1157 A <saml:Statement> instance MAY be of defined to be of type <xacml-
1158 saml:XACMLPolicyStatementType> by using xsi:type="xacml-
1159 saml:XACMLPolicyStatementType" as shown in the example in Section 6.4. such an instance of a
1160 <saml:Statement> element is called an XACMLPolicy Statement in this Profile. Any instance of an
1161 XACMLPolicy Statement in an XACML system MUST be enclosed in a <saml:Assertion>.

1162 **6.4 Element <saml:Assertion>: XACMLPolicy Assertion**

1163 A <saml:Assertion> instance MAY contain an XACMLPolicy Statement as shown in the following non-
1164 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>
```

1165 An instance of a <saml:Assertion> element containing an XACMLPolicy Statement is called an
1166 XACMLPolicy Assertion in this Profile.

1167 When an XACMLPolicy Assertion is part of a response to an <xacml-samlp:XACMLPolicyQuery>,
1168 then the XACMLPolicy Assertion MUST contain exactly one XACMLPolicy Statement, which in turn MAY
1169 contain any number of XACML Policy and PolicySet instances.

1170 This Profile imposes the following requirements and restrictions on the <saml:Assertion> element
1171 beyond those specified in SAML 2.0 when used as an XACMLPolicy Assertion.

1172 <saml:Issuer> [Required]

1173 The <saml:Issuer> element is a required element for holding information about “the SAML
1174 authority that is making the claim(s) in the assertion” [SAML].

1175 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
1176 in the <saml:Issuer> element refer to the entity that signs the XACMLPolicy Assertion. It is up to
1177 the relying party to determine whether it has an appropriate trust relationship with the authority that
1178 signs the XACMLPolicy Assertion.

1179 <ds:Signature> [Optional]

1180 The <ds:Signature> element is an optional element for holding “An XML Signature that
1181 authenticates the assertion, as described [in Section 5 of the SAML 2.0 core specification[SAML]].”

1182 A <ds:Signature> instance MAY be used in an XACMLPolicy Assertion. In order to support 3rd
1183 party digital signatures, this Profile does NOT require that the identity provided in the
1184 <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Assertion. It is up to the
1185 relying party to determine whether it has an appropriate trust relationship with the authority that signs
1186 the XACMLPolicy Assertion.

1187 A relying party SHOULD verify any signature included in the XACMLPolicy Assertion and SHOULD
1188 NOT use information derived from the XACMLPolicy Assertion unless the signature is verified
1189 successfully.

1190 <saml:Subject> [Optional]

1191 The <saml:Subject> element MUST NOT be included in an XACMLPolicy Assertion. Instead, the
1192 Subjects of an XACMLPolicy Assertion are specified in the XACML Policy and PolicySet elements
1193 contained in the enclosed XACMLPolicy Statement.

1194 <saml:Conditions> [Optional]

1195 The <saml:Conditions> element is an optional element that is used for “conditions that MUST be
1196 taken into account in assessing the validity of and/or using the assertion” [SAML].

1197 The <saml:Conditions> instance SHOULD contain NotBefore and NotOnOrAfter XML
1198 attributes to specify the limits on the validity of the XACMLPolicy Assertion. If these XML attributes
1199 are present, the relying party SHOULD ensure that an <xacml-context:Response> taken from
1200 the XACMLPolicy Assertion is used only during the XACMLPolicy Assertion's specified validity period.

1201 6.5 Element <xacml-samlp:XACMLPolicyQuery>

1202 An instance of the <xacml-samlp:XACMLPolicyQuery> protocol element MAY be used by a PDP or
1203 application to request XACML <xacml:Policy> or <xacml:PolicySet> instances from an on-line
1204 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>
```

1205 The <xacml-samlp:XACMLPolicyQuery> element is of <xacml-samlp:XACMLPolicyQueryType>
1206 complex type, which is an extension to the SAML-defined <samlp:RequestAbstractType>.

1207 The <xacml-samlp:XACMLPolicyQuery> element contains zero or more of the following elements in
1208 addition to those defined for the <samlp:RequestAbstractType>:

1209 <xacml-context:Request> [Any Number]

1210 An XACML Request Context. All XACML <xacml:Policy> and <xacml:PolicySet> instances
1211 potentially applicable to this Request that the requester is authorized to receive MUST be returned.
1212 The concept of “applicability” in the XACML context is defined in the XACML 3.0 Specification
1213 [XACML3]. Any superset of applicable policies MAY be returned; for example, all policies having top-
1214 level Target elements that match the Request MAY be returned.

1215 <xacml:PolicySetIdReference> [Any Number]

1216 Identifies an XACML <xacml:PolicySet> instance to be returned.

1217 <xacml:PolicyIdReference> [Any Number]

1218 Identifies an XACML `<xacml:Policy>` instance to be returned.

1219 *Non-normative note: The `<xacml-samlp:XACMLPolicyQuery>` is not intended as a robust provisioning*
1220 *protocol. Users requiring such a protocol may consider using the OASIS Service Provisioning Markup*
1221 *Language (SPML). Note that the SAML-defined ID XML attribute is a required component of an*
1222 *instance of `<samlp:RequestAbstractType>` that the `<xacml-samlp:XACMLPolicyQuery>`*
1223 *extends and MAY be used to correlate the `<xacml-samlp:XACMLPolicyQuery>` with the*
1224 *corresponding XACMLPolicy Response.*

1225 **6.6 Element `<samlp:Response>`: XACMLPolicy Response**

1226 A `<samlp:Response>` instance MAY contain an XACMLPolicy Assertion. An instance of such a
1227 `<samlp:Response>` element is called an XACMLPolicy Response in this Profile. An XACMLPolicy
1228 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>
```

1229 An instance of a `<samlp:Response>` element that contains an XACMLPolicy Assertion is called an
1230 XACMLPolicy Response in this Profile. Such a Response MUST be used as the response to an `<xacml-`
1231 `samlp:XACMLPolicyQuery>`. It MAY be used to convey or store XACML policies for other purposes.

1232 This Profile imposes the following requirements and restrictions on the `<samlp:Response>` element in
1233 addition to those specified in SAML 2.0 when used as an XACMLPolicy Response.

1234 `<saml:Issuer>` [Optional]

1235 The `<saml:Issuer>` element identifies the originator of the contained XACML Policy, which MAY be
1236 the entity that generated the XACMLPolicy Response message. [SAML].

1237 In order to support 3rd party digital signatures, this Profile does NOT require that the identity provided
1238 in the `<saml:Issuer>` element refer to the entity that signs the XACMLPolicy Response. It is up to
1239 the relying party to determine whether it has an appropriate trust relationship with the authority that
1240 signs the XACMLPolicy Response.

1241 `<ds:Signature>` [Optional]

1242 The `<ds:Signature>` element is an optional element for holding “An XML Signature that
1243 authenticates the responder and provides message integrity” [SAML].

1244 A `<ds:Signature>` instance MAY be used in an XACMLPolicy Response. In order to support 3rd
1245 party digital signatures, this Profile does NOT require that the identity provided in the
1246 `<saml:Issuer>` instance refer to the entity that signs the XACMLPolicy Response. It is up to the
1247 relying party to determine whether it has an appropriate trust relationship with the authority that signs
1248 the XACMLPolicy Response.

1249 A relying party SHOULD verify any signature included in the XACMLPolicy Response and SHOULD
1250 NOT use information derived from the XACMLPolicy Response unless the signature is verified
1251 successfully.

1252 <saml:Assertion> [Any Number]

1253 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,
1254 then there MUST be exactly one instance of this element that contains an XACMLPolicy Assertion
1255 representing the response to the associated XACMLPolicy Query. If the XACMLPolicy Response is
1256 not issued in response to an <xacml-samlp:XACMLPolicyQuery>, it MAY contain one or more
1257 XACMLPolicy Assertions as well as other SAML or XACML Assertions.

1258 <saml:Status> [Required]

1259 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,
1260 and if it is not possible to return all policies that satisfy the <xacml-samlp:XACMLPolicyQuery>, then a
1261 <samlp:StatusCode> value of
1262 urn:oasis:names:tc:saml:2.0:status:TooManyResponses MUST be returned in the
1263 <samlp:Status> element of the Response.

1264 InResponseTo [Optional]

1265 This optional XML attribute is "A reference to the identifier of the request to which the response
1266 corresponds." When the XACMLPolicy Response is issued in response to an <xacml-
1267 samlp:XACMLPolicyQuery>, this XML attribute MUST contain the value of the ID XML attribute
1268 from the <xacml-samlp:XACMLPolicyQuery> to which this is a response. This allows the
1269 receiver to correlate the XACMLPolicy Response with the corresponding XACMLPolicy Query.

1270 **6.7 Policy references and Policy assertions**

1271 It may be noted that in relation to a policy assertion, there are three broad classes of policies to consider
1272 when resolving policy references: the top level policy in the policy assertion, the policies in the <xacml-
1273 samlp:ReferencedPolicies> element and policies external to the policy assertion, available to a PDP by
1274 other means.

1275 How policy references are resolved across these three classes of policies depends on the particular case
1276 and problem for which the policy assertion is used. Therefore policy reference resolving is implementation
1277 defined with respect to policy assertions.

1278 **7 Advice**

1279 This Section describes how to include XACMLAuthzDecision Assertion and XACMLPolicy Assertion
1280 instances as advice in another SAML Assertion instance.

1281 **7.1 Element <saml:Advice>**

1282 A SAML Assertion MAY include a <saml:Advice> element containing “Additional information related to
1283 the assertion that assists processing in certain situations but which MAY be ignored [without affecting
1284 either the semantics or the validity of the assertion] by applications that do not understand the advice or do
1285 not wish to make use of it.” [SAML] An XACMLAuthzDecision Assertion or XACMLPolicy Assertion may
1286 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>
```

1287
1288

8 Using an XACML Authorization Decision as an Authorization Token

1289 This Section of the Profile describes how to use an XACMLAuthzDecision Statement as a security and
1290 privacy authorization token as part of a SOAP message exchange in a Web Services context. This token
1291 MAY be used by a client to convey an authorization decision from a trusted 3rd party to a service. A Web
1292 Service MAY use such a token to determine that the client is authorized to access information involved in
1293 the Web Services interaction.

1294 In a Web Services context, an instance of an XACMLAuthzDecision Assertion MAY be used as an
1295 authorization token in the Web Services Security [WSS] and [WSS-Core] `wsse:Security` Header of a
1296 SOAP message. When used in this way, the XACMLAuthzDecision Statement in the
1297 XACMLAuthzDecision Assertion MUST include the corresponding XACML Request Context. This allows
1298 the Web service to determine whether the `<xacml-context:Attribute>` instances in the Request
1299 correspond to the access that the client requires as part of the Web Service interaction. The
1300 XACMLAuthzDecision Assertion SHOULD be signed by a Policy Decision Point trusted by the Web
1301 Service.

1302 A Web Service MAY use this token to determine that a trusted 3rd party has evaluated an XACML Request
1303 Context that is relevant to the invocation of the service, and has reported an authorization decision. The
1304 service SHOULD verify that the signature on the XACMLAuthzDecision Assertion is from a Policy Decision
1305 Point that the service trusts. The service SHOULD verify that the validity period of the
1306 XACMLAuthzDecision Assertion includes the time at which the Web Service interaction will access the
1307 information or resource to which the Request Context applies. The service SHOULD verify that the
1308 `<xacml-context:Attribute>` instances contained in the XACML `<xacml-context:Request>`
1309 element correctly describe the information or resource access that needs to be authorized as part of this
1310 Web Service interaction.

9 Conformance

1311

1312 Implementations of this Profile MAY implement certain subsets of the described functionality. Each
1313 implementation MUST clearly identify the subsets it implements using the following identifiers.

1314 An implementation of this Profile is a conforming *SAML Attribute* implementation if the implementation
1315 conforms to Section 2 of this Profile. The following URI MUST be used as the identifier for this
1316 functionality:

1317 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:attrs:all`

1318 An implementation of this Profile is a conforming *SOAP Attributes as XACML Authz Decision Query*
1319 implementation if the implementation conforms to Section 3.1 of this Profile. The following URI MUST be
1320 used as the identifier for this functionality:

1321 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:authzQuery`

1322 An implementation of this Profile is a conforming *SOAP Attributes as SAML Attribute Assertion*
1323 implementation if the implementation conforms to Section 3.2 of this Profile. The following URI MUST be
1324 used as the identifier for this functionality:

1325 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:attrAssertion`

1326

1327 An implementation of this Profile is a conforming *XACML Authz Decision without Policies* implementation
1328 if the implementation conforms to all parts of Section 4 of this Profile excluding the `<xacml:Policy>`,
1329 `<xacml:PolicySet>`, and `<xacml-samlp:ReferencedPolicies>` elements and their sub-elements
1330 and the `CombinePolicies` XML attribute in the `<xacml-samlp:XACMLAuthzDecisionQuery>`.
1331 XACML 3.0 implementations MUST support the `<xacml-samlp:AdditionalAttributes>` element
1332 and its sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0
1333 implementations MUST NOT support the `<xacml-samlp:AdditionalAttributes>` element and its
1334 sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be used
1335 as the identifier for this functionality:

1336 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:noPolicies`

1337 An implementation of this Profile is a conforming *XACML Authz Decision with Policies* implementation if
1338 the implementation conforms to all parts of Section 4 of this Profile. XACML 3.0 implementations MUST
1339 support the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-
1340 samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0 implementations MUST NOT support
1341 the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-
1342 samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be used as the identifier for this
1343 functionality:

1344 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:withPolicies`

1345 An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust with Policies*
1346 implementation if it conforms to section 5 in its entirety as described in the previous paragraph. The
1347 following URI MUST be used as the identifier for this functionality.

1348 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:withP
1349 olicies`

1350 An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust without Policies*
1351 *implementation if it conforms to section 5, with the exceptions relating to policies and additioanl attribues*
1352 *noted above. The following URI MUST be used as the identifier for this functionality.*

1353 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:noPol
1354 icies`

1355 An implementation of this Profile is a conforming *XACML Policies* implementation if the implementation
1356 conforms to Section 6 of this Profile. The following URI MUST be used as the identifier for this
1357 functionality:

1358 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:policies

1359 An implementation of this Profile is a conforming *SAML Advice* implementation if the implementation
1360 conforms to Section 7 of this Profile. The following URI MUST be used as the identifier for this
1361 functionality:

1362 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:adviceSAML

1363 An implementation of this Profile is a conforming *XACML Authz Token* implementation if the
1364 implementation conforms to Section 8 of this Profile. The following URI MUST be used as the identifier
1365 for this functionality:

1366 urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzToken

1367

1368 **Appendix A. Acknowledgments**

1369 The following individuals have participated in the creation of this specification and are gratefully
1370 acknowledged

1371 Anil Saldhana

1372 Anil Tappetla

1373 Anne Anderson

1374 Anthony Nadalin

1375 Bill Parducci

1376 Craig Forster

1377 David Chadwick

1378 David Staggs

1379 Dilli Arumugam

1380 Duane DeCouteau

1381 Erik Rissanen

1382 Gareth Richards

1383 Hal Lockhart

1384 Jan Herrmann

1385 John Tolbert

1386 Ludwig Seitz

1387 Michiharu Kudo

1388 Naomaru Itoi

1389 Paul Tyson

1390 Prateek Mishra

1391 Rich Levinson

1392 Ronald Jacobson

1393 Seth Proctor

1394 Sridhar Muppidi

1395 Tim Moses

1396 Vernon Murdoch

1397

1398

Appendix B. Revision History

Rev	Date	By whom	What
WD 1	12 April 2006	Anne Anderson	Create from SAML Profile errata document. <XACMLAuthzDecisionStatementType>: replace "ReturnResponse" with "ReturnContext" in description. Authorization Decisions: replaced "in the Response to an <XACMLAuthzDecisionStatement>" with "...<XACMLAuthzDecisionQuery>". Create new types for SAML elements that will need to include XACML extensions. Create new elements for each extended type. Allow an XACMLAuthzDecisionQuery to include XACML policies for use in evaluating that query. Allow an XACMLAssertion to contain an XACMLAdvice element that in turn can contain an XACMLAssertion.
WD 2	23 June 2006	Anne Anderson	Changed name to "xacml-2.0-profile-saml2.0-v2-spec.... Removed specifications for all new elements except the XACMLAuthzDecisionQuery and XACMLPolicyQuery and all new types except for XACMLAuthzDecisionStatementType and XACMLPolicyStatementType and the two new Query types. Added descriptions of each standard SAML element in which XACML types might occur, and gave examples of use of xsi:type. Described use of the ID and InResponseTo attributes to correlate Queries and Responses.
WD 3	5 March 2007	Anne Anderson	-change boilerplate to conform to new OASIS template -Title: change to reflect that this profile applies to all versions of XACML -1.3 Added section on backwards compatibility -1.4 Removed notation section -1.5 Added namespaces section -2.6 Insert the "Conveying XACML Attributes in a SOAP Message" section from the WS-XACML profile -2.1.1 Clarify that <saml:Subject> is not translated into an XACML -id Attribute -3.5 and following, 3.13: add syntax for passing additional Attributes in XACMLAuthzDecisionQuery from Admin Policy. 3.9 and following: add syntax for passing references policies. -4.4 XACMLPolicyQuery: clarify it returns all potentially applicable policies; remove Target element; change Choice lower bound from 0 to 1 and remove case where no elements included; add non-normative note to consider SPML for provisioning protocol -4.5 Response: Use valid ID values in example; add <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

			<p>embedded in other instances (e.g. <saml:Attribute> rather than SAML Attribute, but SAML Attribute Response rather than <samlp:Response>).</p> <p>-throughout: changed SHALL to MUST</p> <p>-throughout: added namespace designators to schema items and added additional namespace prefixes to list in Section 1.4</p> <p>-Figure 1 updated the "Components and messages diagram to use same names as text</p> <p>-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.</p> <p>-2.1.1 Reworded description of NotBefore, NotOnOrAfter relationship to XACML date/time Attributes to be more clear</p> <p>-3.4.7,B.1 Inserted non-normative notes referring to open issues in relevant places</p> <p>-3.4.4.1 Clarified that the ReferencedPolicies element need not contain policies that receiver is not authorized to view</p> <p>-3.9 Clarified that Policy[Set]IdReference values must exactly match corresponding Policy[Set]Id values in the ReferencedPolicies element.</p> <p>-3.7 Changed "AttributeMatch" to "Match" to fit 3.0 schema</p> <p>-3.9.schemas:Fixed schema for ReferencedPolicies so it validates</p> <p>-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"</p> <p>-4.1.4.2 Add ReferencedPolicies element to XACMLPolicyStatementType</p> <p>-4.6 Reworded so to allow Response that is not issued in response to a specific Query</p> <p>-7 Added first draft of SAML Metadata</p> <p>-8 Added urn for SAML Metadata functionality</p>
WD 5	19 July 2007	Anne Anderson	<p>-Import XACML 1.0 schemas from local copies</p> <p>-Import XACML 2.0 schemas from http://docs.oasis-open.org/xacml/ directory</p> <p>-Import XACML 3.0 WD3 schema</p> <p>-Add OASIS copyright to all schemas</p> <p>-Made "Conveying XACML Attributes in a SOAP Message" a separate Section for easier reference in Conformance Section</p> <p>-Revised Conformance Section to refer to current document sections and to include previously omitted elements.</p> <p>-Made Introduction non-normative except for Namespaces and Normative References sections.</p> <p>-Made SAML Metadata section normative but RECOMMENDED</p>
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 authz query.</p> <p>Relax the constraints on the <ReferencedPolicies> element.</p>
WD 7	23 March 2009	Hal Lockhart	<p>Improved some wording from previous changes.</p> <p>Added WS-Trust based decision request and response.</p> <p>Removed Metadata conformance clause.</p>
WD 10	15 Dec 2009	Erik Rissanen	Add xs:any to authz query protocol
WD 11	17 Dec 2009	Erik Rissanen	<p>Update acknowledgments</p> <p>Fix formatting issues</p>
WD 12	12 Jan 2010	Erik Rissanen	<p>Updated cross references</p> <p>Removed reference to non-existing section.</p> <p>Update acknowledgments</p>
WD 13	8 Mar 2010	Erik Rissanen	<p>Updated cross references</p> <p>Fixed OASIS formatting issues</p> <p>Removed unused reference to XACML 2.0 introduction</p>

