



# SAML 2.0 Profile of XACML, Version 2.0

## Committee Specification 01

10 August 2010

### Specification URIs:

#### This Version:

<http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cs-01-en.html>

<http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cs-01-en.odt> (Authoritative)

<http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cs-01-en.pdf>

#### Previous 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>

#### 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](mailto:hal.lockhart@oracle.com)>

Bill Parducci <[bill@parducci.net](mailto:bill@parducci.net)>

### Editors:

Erik Rissanen <[erik@axiomatics.com](mailto:erik@axiomatics.com)>

Hal Lockhart <[hal.lockhart@oracle.com](mailto: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  
50 this profile: 1) use of SAML 2.0 Attribute Assertions with XACML, including the use of SAML  
51 Attribute Assertions in a SOAP Header to convey Attributes that can be consumed by an XACML  
52 PDP, 2) use of SAML to carry XACML authorization decisions, authorization decision queries,  
53 and authorization decision responses, 3) use of SAML to carry XACML policies, policy queries,  
54 and policy query responses, 4) use of XACML authorization decisions or policies as Advice in  
55 SAML 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/)  
69 [open.org/committees/xacml](http://www.oasis-open.org/committees/xacml/)<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/).

---

# Notices

72

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  
79 notice and this section are included on all such copies and derivative works. However, this document  
80 itself may not be modified in any way, including by removing the copyright notice or references to OASIS,  
81 except as needed for the purpose of developing any document or deliverable produced by an OASIS  
82 Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR  
83 Policy, must be 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,  
93 to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to  
94 such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that  
95 produced this specification.

96 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of  
97 any 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  
104 represent that it has made any effort to identify any such rights. Information on OASIS' procedures with  
105 respect to rights in any document or deliverable produced by an OASIS Technical Committee can be  
106 found on the OASIS website. Copies of claims of rights made available for publication and any  
107 assurances of licenses to be made available, or the result of an attempt made to obtain a general license  
108 or permission for the use of such proprietary rights by implementers or users of this OASIS Committee  
109 Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no  
110 representation that any information or list of intellectual property rights will at any time be complete, or  
111 that any claims in such list are, in fact, Essential Claims.

112 The names "OASIS", "SAML" and "XACML" are trademarks of OASIS, 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  
115 enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for  
116 above 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  
190 how to implement a Policy Enforcement Point, Policy Administration Point, Attribute Authority, Context  
191 Handler, or Repository, but XACML artifacts can serve as a standard format for exchanging information  
192 between 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  
197 assertions, such as the identity of the assertion issuer, the validity period of the assertion, and the digital  
198 signature of the assertion. The SAML specification describes how these elements are to be used. In  
199 addition, SAML has associated specifications that define bindings to other standards. These other  
200 standards provide transport mechanisms and specify how digital signatures should be created and  
201 verified.

## 202 1.1 Organization of this Profile

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

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

- 208 1. `<saml:Attribute>` – A standard SAML element that MAY be used in an XACML system for  
209 storing and transmitting attribute values. The `<saml:Attribute>` must be at least conceptually

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

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

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

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

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

- 249 1. `xacml-saml:XACMLPolicyStatementType` – A new SAML extension type defined in this  
250 Profile that MAY be used in an XACML system to create XACMLPolicy Statements that hold  
251 XACML policies for storage or transmission.

- 252 2. `<saml:Statement>` – A standard SAML element that MUST be used to contain instances of  
253 the `xacml-saml:XACMLPolicyStatementType`. An instance of such a `<saml:Statement>`  
254 element is called an XACMLPolicy Statement in this Profile.
- 255 3. `<saml:Assertion>` – A standard SAML element that MUST be used to hold XACMLPolicy  
256 Statement instances in an XACML system, either in a repository or in an XACMLPolicy  
257 Response. An instance of such a `<saml:Assertion>` element is called an XACMLPolicy  
258 Assertion in this Profile.
- 259 4. `<xacml-samlp:XACMLPolicyQuery>` – A new SAML extension protocol element defined in  
260 this Profile that MAY be used by a PDP or other application to request XACML policies from a  
261 Policy Administration Point (PAP).
- 262 5. `<samlp:Response>` – A standard SAML protocol element that MUST be used to return  
263 XACMLPolicy Assertions in response to an `<xacml-samlp:XACMLPolicyQuery>`. An  
264 instance of such a `<samlp:Response>` element is called an XACMLPolicy Response in this  
265 Profile.

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

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

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

273

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

## 275 **1.1 Diagram of SAML integration with XACML**

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



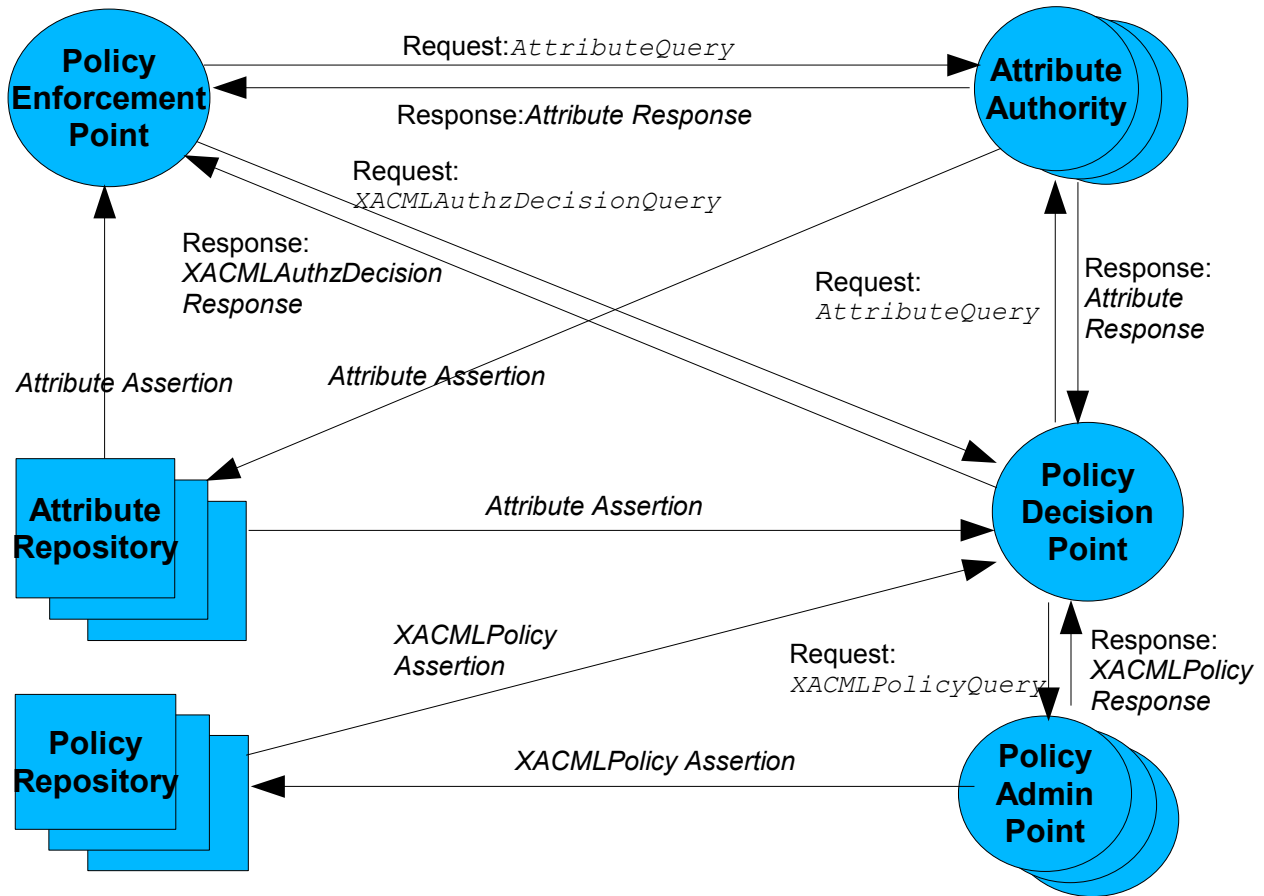


Figure 1: Components and messages in a integration of SAML with XACML

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

## 282 1.2 Backwards compatibility

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

## 286 1.3 Terminology

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 335 1.1 Namespaces

336 *Normative*

337 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	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>
xsi	<a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a>
wsse	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a> or <a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.1.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.1.xsd</a>
wst	<a href="http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3.xsd">http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3.xsd</a>

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

341 XACML 1.0:

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

349 XACML 1.1:

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

357 XACML 2.0:

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

365 XACML 3.0:  
 366     xacml="urn:oasis:names:tc:xacml:3.0:schema:os"  
 367     xacml-context="urn:oasis:names:tc:xacml:3.0:schema:os"

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

## 374 1.2 Normative References

- 375     **[ADMIN]**           OASIS Committee Specification 01, XACML v3.0 Administration and Delegation  
 376                         Profile Version 1.0. 11 March 2010. [http://docs.oasis-open.org/xacml/3.0/xacml-](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-administration-v1-spec-cs-01-en.doc)  
 377                         [3.0-administration-v1-spec-cs-01-en.doc](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-administration-v1-spec-cs-01-en.doc)
- 378     **[RFC 2119]**         S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF  
 379                         RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 380     **[SAML]**             OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion*  
 381                         *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)  
 382                         [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)
- 383     **[SAML-PROFILE]**   **OASIS Standard, Profiles for the OASIS Security Assertion Markup**  
 384                         **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)**  
 385                         **[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)**
- 386     **[XACML1]**           OASIS Standard, *eXtensible Access Control Markup Language (XACML)*  
 387                         *Version 1.0*, 18 February 2003, [http://www.oasis-](http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf)  
 388                         [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)
- 389     **[XACML1.1]**         OASIS Standard, *eXtensible Access Control Markup Language (XACML)*  
 390                         *Version 1.1*, 7 August 2003, [http://www.oasis-](http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf)  
 391                         [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)
- 392     **[XACML2]**           OASIS, Standard, *eXtensible Access Control Markup Language (XACML)*  
 393                         *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)  
 394                         [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).
- 395     **[XACML3]**           OASIS Committee Specification 01, *eXtensible access control markup language*  
 396                         *(XACML) Version 3.0*. August 2010. [http://docs.oasis-open.org/xacml/3.0/xacml-](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.doc)  
 397                         [3.0-core-spec-cs-01-en.doc](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.doc)
- 398     **[XACML-SAML]**       the schemas associated with namespace <xacml-saml> that are a normative  
 399                         part of this Profile.
- 400     **[XACML-SAMPLP]**   the schemas associated with namespace <xacml-samplp> that are a normative  
 401                         part of this Profile.
- 402     **[WSFED]**           OASIS Committee Draft 02, *Web Services Federation Language (WS-*  
 403                         *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)  
 404                         [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)
- 405     **[WSS]**             OASIS Standard, *Web Services Security: SOAP Message Security 1.0 (WS-*  
 406                         *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)  
 407                         [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)
- 408     **[WSS-Core]**       OASIS Standard, *WS-Security Core Specification 1.1*, February 2006,  
 409                         [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)  
 410                         [SOAPMessageSecurity.pdf](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)
- 411     **[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)  
 412                         [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)

413

414 **1.3 Non-normative References**

415 None

416

417

418

## 2 Attributes

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

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

### 440 2.1 Element `<saml:Attribute>`

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

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

#### 447 2.1.1 Mapping a `<saml:Attribute>` to an `<xacml-context:Attribute>`

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

- 453 • XACML `AttributeId` XML attribute

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

- 455 • XACML `DataType` XML attribute

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

- 459 • XACML `Issuer` XML attribute

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

461 • `<xacml-context:AttributeValue>`

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

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

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

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

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

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

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

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

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

## 505 **2.1 Element <saml:AttributeStatement>**

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

## 509 **2.2 Element <saml:Assertion>: SAML Attribute Assertion**

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

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

517 <saml:Issuer> [Required]

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

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

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

528 <ds:Signature> [Optional]

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

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

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

539 <saml:Subject> [Optional]

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

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



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

553 <saml:Conditions> [Optional]

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

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

## 566 2.3 Element <samlp:AttributeQuery>

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

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

## 574 2.4 Element <samlp:Response>: SAML Attribute Response

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

582 <saml:Issuer> [Optional]

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

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

589 <ds:Signature> [Optional]

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

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

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

---

## 600 3 Conveying XACML Attributes in a SOAP Message

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

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

### 611 3.1 `<xacml-samlp:XACMLAuthzDecisionQuery>`

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

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

### 625 3.2 SAML Attribute Assertion

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

---

## 4 Authorization Decisions

631

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

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

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

### 659 4.1 Type `<xacml-saml:XACMLAuthzDecisionStatementType>`

660 The new `<xacml-saml:XACMLAuthzDecisionStatementType>` complex type contains an XACML  
661 Response Context along with related information. Use of this type is an alternative to use of the SAML-  
662 defined `<saml:AuthzDecisionStatementType>`; this alternative allows an XACML Context Handler  
663 to use SAML with full support for XACML authorization decisions. An instance of a  
664 `<saml:Statement>` element that is of this type (defined using `xsi:type="xacml-`  
665 `saml:XACMLAuthzDecisionStatementType"`) is called an XACMLAuthzDecision Statement in this  
666 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>

```

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

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

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

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

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

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

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

## 688 4.2 Element `<saml:Statement>`: XACMLAuthzDecision Statement

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

## 695 4.3 Element `<saml:Assertion>`: XACMLAuthzDecision Assertion

696 A `<saml:Assertion>` instance MAY contain an XACMLAuthzDecision Statement as shown in the  
 697 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>

```

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

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

702 `<saml:Issuer>` [Required]

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

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

709 `<ds:Signature>` [Optional]

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

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

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

720 `<saml:Subject>` [Optional]

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

725 `<saml:Conditions>` [Optional]

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

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

#### 733 **4.4 Element <xacml-samlp:XACMLAuthzDecisionQuery>**

734 The <xacml-samlp:XACMLAuthzDecisionQuery> protocol element MAY be used by a PEP to  
735 request an authorization decision from an XACML PDP. This element is an alternative to the SAML-  
736 defined <samlp:AuthzDecisionQuery>; this alternative allows the PEP to use the full capabilities of  
737 an XACML PDP. It allows use of the SAML query protocol to convey an XACML Request Context along  
738 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>

```

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

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

744 `InputContextOnly` [Default "false"]

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

751 `ReturnContext` [Default "false"]

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

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



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

763

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

766 CombinePolicies [Default "true"]

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

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

780

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

788 <xacml-context:Request> [Required]

789 An XACML Request Context that is to be evaluated.

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

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

793 <xacml:Policy> [Any Number]

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

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

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

801 <xacml:PolicySet> [Any Number]

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

804 If the CombinePolicies XML attribute is "true", then the PDP MUST use such XACML PolicySet  
805 instances.

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

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

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

819 <xacml-saml:Extensions> [Optional]

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

## 821 **4.5 Element <xacml-sampl:Extensions>**

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

```
823 <element name="Extensions" xsi:type="xacml-sampl:ExtensionsType" />  
824 <complexType name="ExtensionsType">  
825   <sequence>  
826     <any namespace="##any" processContents="strict" minOccurs="0"  
827       maxOccurs="unbounded"/>  
828   </sequence>  
829 </complexType>
```

830 The <xacml-sampl:Extensions> element contains the following XML elements:

831 xs:any [Any Number]

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

## 839 **4.6 Element <xacml-sampl:AdditionalAttributes>**

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

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

```
848 <element name="AdditionalAttributes"  
849   type="xacml-samlp: AdditionalAttributesType"/>  
850 <complexType name="AdditionalAttributesType">  
851   <sequence>  
852     <element ref="xacml-samlp:AssignedAttributes" minOccurs="0"  
853     maxOccurs="unbounded"/>  
854   </sequence>  
855 </complexType>
```

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

857 The <AdditionalAttributes> element contains the following elements:

858 <AssignedAttributes> [Required]

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

## 860 **4.7 Element <xacml-samlp:AssignedAttributes>**

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

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

```
865 <element name="AssignedAttributes" type="xacml-samlp:AssignedAttributesType"/>  
866 <complexType name="AssignedAttributesType">  
867   <sequence>  
868     <element ref="xacml-samlp: Holders"/>  
869     <element ref="xacml-samlp: HolderAttributes"/>  
870   </sequence>  
871 </complexType>
```

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

873 The <AssignedAttributes> element contains the following elements:

874 <xacml-samlp: Holders> [Required]

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

876 <xacml-samlp: HolderAttributes> [Required]

877 The XACML Attributes of the delegate entity.

## 878 **4.8 Element <xacml-samlp: Holders>**

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

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

```
883 <element name="Holders" type="xacml-sampl:HolderType"/>
884 <complexType name="HolderType">
885   <sequence>
886     <element ref="xacml:Match" maxOccurs="unbounded"/>
887   </sequence>
888 </complexType>
```

889 The `<xacml-sampl:HolderType>` element is of `<xacml-sampl:HolderType>` complex type.

890 The `<xacml-sampl:HolderType>` element contains the following elements:

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

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

898

## 899 **4.9 Element `<xacml-sampl:HolderAttributes>`**

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

902 The `<xacml-sampl:HolderAttributes>` element MUST contain XACML Attributes that apply to the  
903 delegate entities identified in the corresponding `<xacml-sampl:Holders>` element.

```
904 <element name="HolderAttributes" type="xacml-sampl:HolderAttributesType"/>
905 <complexType name="HolderAttributesType">
906   <sequence>
907     <element ref="xacml-context:Attribute"
908       minOccurs="0" maxOccurs="unbounded"/>
909   </sequence>
910 </complexType>
```

911 The `<xacml-sampl:HolderAttributesType>` element is of `<xacml-sampl:HolderAttributesType>`  
912 complex type.

913 The `<xacml-sampl:HolderAttributes>` element contains the following elements:

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

915 An XACML Attribute of the delegate entities identified in the corresponding `<xacml-`  
916 `sampl:Holders>` element.

## 917 **4.10 Element `<xacml-saml:ReferencedPolicies>`**

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

```

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

```

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

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

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

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

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

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

## 948 4.11 Element `<samlp:Response>`: XACMLAuthzDecision Response

949 A `<samlp:Response>` instance MAY contain an XACMLAuthzDecision Assertion as shown in the  
950 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>

```

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

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

956 `<saml:Issuer>` [Optional]

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

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

963 `<ds:Signature>` [Optional]

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

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

971 A relying party SHOULD verify any signature included in the XACMLAuthzDecision Response and  
972 SHOULD NOT use information derived from the Response unless the signature is verified  
973 successfully.

974 `<saml:Assertion>` [Any Number]

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

977 `<samlp:StatusCode>` [Required]

978 The `<samlp:StatusCode>` element is a component of the `<samlp:Status>` element in the  
979 `<samlp:Response>`.

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

983 `urn:oasis:names:tc:SAML:2.0:status:Success`

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

986 `urn:oasis:names:tc:SAML:2.0:status:Requester`

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

992 `urn:oasis:names:tc:SAML:2.0:status:Responder`

993 This value for the `<samlp:StatusCode>` Value XML attribute MUST be used when the  
994 `<xacml-context:StatusCode>` value is  
995 `urn:oasis:names:tc:xacml:1.0:status:syntax-error` due to a syntax error in an

996 <xacml:Policy> or <xacml:PolicySet>. Note that not all syntax errors in policies will be  
997 detected in conjunction with the processing of a particular query, so not all policy syntax errors  
998 will be reported this way.

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

1000 This value for the <samlp:StatusCode> Value XML attribute MUST be used only when the  
1001 SAML interface at the PDP does not support the version of the SAML schema used in the query.

1002 InResponseTo [Optional]

1003 This optional XML attribute is “A reference to the identifier of the request to which the response  
1004 corresponds.” When the XACMLAuthzDecision Response is issued in response to an  
1005 XACMLAuthzDecision Query, this XML attribute MUST contain the value of the ID XML attribute  
1006 from the XACMLAuthzDecision Query to which this is a response. This allows the receiver to  
1007 correlate the XACMLAuthzDecision Response with the corresponding XACMLAuthzDecision  
1008 Query. The SAML-defined ID XML attribute is a required component of an instance of the  
1009 <samlp:RequestAbstractType> of which the <xacml-  
1010 samlp:XACMLAuthzDecisionQuery> is an extension.

## 1011 **4.12 Functional Requirements for the <xacml- 1012 samlp:AssignedAttributes> Element**

1013 During processing of the provided access request, if the <xacml-samlp: HOLDERS> element of a  
1014 provided <xacml-samlp:AssignedAttributes> element matches a section of the XACML Request  
1015 Context, then the XACML Context Handler MUST make the XACML Attributes in the <xacml-  
1016 samlp:HolderAttributes> element appear in that section of the XACML Request Context. Any  
1017 inheritance between <xacml-samlp:AssignedAttributes> elements is not deduced.

1018 The matching of additional XACML Attributes MUST be made against all Request Contexts involved in  
1019 the processing of the XACMLAuthzDecision Query, including the provided access request itself and any  
1020 Request Contexts formed as part of reduction.

1021 The provided XACML Attributes MUST be used only in the evaluation of the provided access request  
1022 and any derived Request Contexts, including reduction, and MUST NOT be used in evaluation of  
1023 requests not related to the provided access request unless associated with those other requests  
1024 independent of the <xacml-samlp:XACMLAuthzDecisionQuery>.

1025 The implementation MUST match the <xacml-samlp: HOLDERS> element against all the attributes  
1026 available to the context handler, but MUST NOT use any matching <xacml-  
1027 samlp:HolderAttributes> to find even more attributes through the context handler or even more  
1028 supplied attributes through other <xacml-samlp: HOLDERS> elements. This implies that there can be  
1029 no inheritance between <xacml-samlp:AssignedAttributes> elements.

---

## 1030 5 XACML Decision Queries using WS-Trust

1031 In some environments, it may be desirable to obtain an XACML authorization decision from a Security  
1032 Token Service (STS) using the WS-Trust protocol [WSTRUST].

### 1033 5.1 Common Claims Dialect

1034 One method of doing this is to support the Common Claim Dialect as defined in WS-Federation  
1035 [WSFED], chapter 9. In this case the implementation must map the contents of an incoming  
1036 <RequestSecurityToken> element into a XACML <Request> element and map the XACML <Response>  
1037 into an outgoing <RequestSecurityTokenResponseCollection> element. When this approach is taken,  
1038 there is no explicit reference to XACML in the wire protocol and in general a requestijg party will not be  
1039 aware whether or not an XACML-based PDP was used to make the decision.

### 1040 5.2 XACML Dialect

1041 This section defines a WS-Trust-based protocol which is intended to be easier and more efficient for  
1042 XACML PDP to implement. It is based directly on the constructs previously defined in Section 4. It uses  
1043 the <saml:Assertion> element and <saml:Statement> of type xacml-  
1044 saml:XACMLAuthzDecisionStatementType to wrap the XACML <Request> and <Response> elements.  
1045 However, the <xacml-samlp:XACMLDecisionQuery> and <samlp:Response> elements are not used.  
1046 Instead the request is conveyed in a <wst:RequestSecurityToken> element and the response is carried  
1047 in a <wst:RequestSecurityTokenResponseCollection> element containing a  
1048 <wst:RequestSecurityTokenResponse> element.

1049 Except for the outer protocol layer, described in more detail below, the syntax and functional  
1050 requirements for this protocol is exactly as described above in section 4. In fact, it is possible for a server  
1051 which contains an XACML PDP to support both protocols, using distinct web service endpoints, with only  
1052 a small amount of distinct code to handle each request type.

### 1053 5.3 Decision Request

1054 The decision request is contained in a <wst:RequestSecurityToken> element. This element contains the  
1055 following attributes and elements from the WS-Trust schema.

- 1056 • Context This URI specifies an identifier for this request. Its value will be returned in the  
1057 corresponding response to allow them to be correlated.
- 1058 • <wst:TokenType> This element contains the value: urn:oasis:names:tc:xacml:3.0:core:schema,  
1059 to indicate that an XACML decision token will be returned.
- 1060 • <wst:RequestType> This element contains the value: [http://docs.oasis-open.org/ws-sx/ws-  
1061 trust/200512/Issue](http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue)

1062 In addition, the <wst:RequestSecurityToken> element MAY contain any of the attributes and elements  
1063 defined in section 4.4 above as being contained in the <xacml-samlp:XACMLAuthzDecisionQuery>  
1064 element. Specifically these are the attributes:

- 1065 • InputContextOnly,
- 1066 • ReturnContext, and
- 1067 • CombinePolicies.

1068 These are the elements:



- 1069 • <xacml-context:Request>,
- 1070 • <xacml-samlp:AdditionalAttributes>,
- 1071 • <xacml:Policy>,
- 1072 • <xacml:PolicySet>, and
- 1073 • <xacml-saml:ReferencedPolicies>.

1074 The functional requirements for processing these attributes and elements are exactly as set forth in  
1075 section 4 above.

## 1076 **5.4 Decision Response**

1077 The decision response is contained in a <wst:RequestTokenResponseCollection> element. It contains  
1078 exactly one <wst:RequestTokenResponse> element. This element contains the following attributes and  
1079 elements.

- 1080 • Context This element contains the same URI provided in the Context attribute of the request.
- 1081 • <wst:RequestedSecurityToken> This element contains a <saml:Assertion which in turn contains  
1082 a <saml:Statement of type xacml-saml:XACMLAuthzDecisionStatementType as described in  
1083 sections 4.1, 4.2, and 4.3 above. The functional requirements for processing these attributes  
1084 and elements are exactly as set forth in section 4 above.

1085

## 6 Policies

1086 XACML defines the `<xacml:Policy>` and `<xacml:PolicySet>` elements for expressing policies. In  
1087 many environments, instances of these elements need to be stored or transmitted between entities in an  
1088 XACML system. Such instances may need to be signed or associated with a validity period. SAML is  
1089 intended to provide this functionality for security-related assertions, but SAML does not define any  
1090 Protocol or Assertion elements for policies. In order to allow entities in an XACML system to use SAML  
1091 assertions and protocols to store, transmit, and query for XACML policies, this Profile defines one SAML  
1092 extension type and one SAML extension element, and describes how they are used with other standard  
1093 SAML elements.

- 1094 • `<xacml-saml:XACMLPolicyStatementType>` is a new SAML extension type that includes  
1095 XACML policies.
- 1096 • A `<saml:Statement>` defined using `xsi:type="xacml-saml:XACMLPolicyStatementType"`  
1097 MAY be used in an XACML system to store or convey XACML policies. An instance of a  
1098 `<saml:Statement>` element defined using this type is called an XACMLPolicy Statement in this  
1099 Profile.
- 1100 • A `<saml:Assertion>` MUST be used to hold XACMLPolicy Statements. An instance of such a  
1101 `<saml:Assertion>` element is called an XACMLPolicy Assertion in this Profile.
- 1102 • An `<xacml-samlp:XACMLPolicyQuery>` is a new SAML extension element that MAY be used by a  
1103 PDP or other entity to request XACML policies as a SAML protocol query.
- 1104 • A `<samlp:Response>` containing an XACMLPolicy Assertion that MUST be used in response to an  
1105 `<xacml-samlp:XACMLPolicyQuery>`. It MAY be used to transmit XACML policies in other  
1106 contexts. An instance of such a `<samlp:Response>` is called an XACMLPolicy Response in this  
1107 Profile.

1108 This Section defines and describes the usage of these types and elements. The schemas for the new  
1109 type and element are contained in the [XACML-SAML] and [XACML-SAML] schema documents.

### 1110 6.1 Type `<xacml-saml:XACMLPolicyStatementType>`

1111 The `<xacml-saml:XACMLPolicyStatementType>` complex type contains XACML Policy and or  
1112 XACML PolicySet elements. An instance of a `<saml:Statement>` element that is of this type is called  
1113 an XACMLPolicy Statement in this Profile.

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

1114 The `<xacml-saml:XACMLPolicyStatementType>` complex type is an extension to the SAML-  
1115 defined `<saml:StatementAbstractType>`. It contains the following elements.

1116 <xacml:Policy> [Any Number]

1117 If the XACMLPolicy Statement represents a response to an <xacml-samlp:XACMLPolicyQuery>,  
1118 then this element MUST contain one of the <xacml:Policy> instances that meet the specifications  
1119 of the associated <xacml-samlp:XACMLPolicyQuery>. Otherwise, this element MAY contain an  
1120 arbitrary <xacml:Policy> instance.

1121 <xacml:PolicySet> [Any Number]

1122 If the XACMLPolicy Statement represents a response to an <xacml-samlp:XACMLPolicyQuery>,  
1123 then this element MUST contain one of the <xacml:PolicySet> instances that meet the  
1124 specifications of the associated <xacml-samlp:XACMLPolicyQuery>. Otherwise, this element  
1125 MAY contain an arbitrary <xacml:PolicySet> instance.

1126 <xacml-saml:ReferencedPolicies> [Zero or One]

1127 With the exception of XACML Policy and PolicySet instances that the receiver of the XACMLPolicy  
1128 Statement is not authorized to view, this element MAY contain XACML Policy and PolicySet  
1129 instances required to resolve <xacml:PolicySetIdReference> or  
1130 <xacml:PolicyIdReference> instances contained in the XACMLPolicy Statement, including  
1131 those in the <xacml-saml:ReferencedPolicies> instance itself. The values of the PolicyId  
1132 and PolicySetId XML attributes of the policies included in the <xacml-  
1133 saml:ReferencedPolicies> instance MUST exactly match the values contained in the  
1134 corresponding <xacml:PolicySetIdReference> or <xacml:PolicyIdReference>  
1135 instances.

1136 Subject to authorization and availability, if the XACMLPolicy Statement is issued in response to an  
1137 <xacml-samlp:XACMLPolicyQuery>, there MUST be exactly one <xacml:Policy> element  
1138 included for every XACML Policy that satisfies the XACMLPolicy Query, and there MUST be exactly one  
1139 <xacml:PolicySet> element included for every XACML PolicySet that satisfies the XACMLPolicy  
1140 Query . The responder MUST return all XACML policies available to the responder that satisfy the  
1141 <xacml-samlp:XACMLPolicyQuery> and that the requester is authorized to receive.

1142 If the XACMLPolicy Statement is issued in response to an <xacml-samlp:XACMLPolicyQuery>, and  
1143 there are no <xacml:Policy> or <xacml:PolicySet> instances that meet the specifications of the  
1144 associated <xacml-samlp:XACMLPolicyQuery>, then there MUST be exactly one empty  
1145 XACMLPolicy Statement included in the response.

1146 An XACMLPolicy Statement enclosed in a signed SAML assertion MAY be used as a method of  
1147 authentication of XACML policies. In this case the Policy or PolicySet MUST NOT contain an XACML  
1148 <PolicyIssuer> element. Instead the PDP MAY generate a <PolicyIssuer> element from the certificate or  
1149 other security token associated with the signature of the SAML assertion before using the policy for  
1150 XACML request evaluation. In this case the issuer of the SAML assertion SHALL be translated into an  
1151 XACML attribute with id urn:oasis:names:tc:xacml:1.0:subject:subject-id. This does that  
1152 mean that the issuer name must be taken directly from the security token, merely that the PDP perform  
1153 some mapping on the claims in the token to determine the issuer.

## 1154 **6.2 Element <xacml-saml:ReferencedPolicies>**

1155 An instance of this element MAY be used to contain copies of policies referenced from  
1156 <xacml:Policy> or <xacml:PolicySet> instances included in the <xacml-  
1157 samlp:XACMLPolicyQuery>, as well as copies of policies referenced from other policies included in  
1158 the <xacml-saml:ReferencedPolicies> instance.

1159 See Section 4.10 for a description of the <xacml-saml:ReferencedPolicies> element.

### 1160 **6.3 Element <saml:Statement>: XACMLPolicy Statement**

1161 A <saml:Statement> instance MAY be of defined to be of type <xacml-  
1162 saml:XACMLPolicyStatementType> by using xsi:type="xacml-  
1163 saml:XACMLPolicyStatementType" as shown in the example in Section 6.4. such an instance of a  
1164 <saml:Statement> element is called an XACMLPolicy Statement in this Profile. Any instance of an  
1165 XACMLPolicy Statement in an XACML system MUST be enclosed in a <saml:Assertion>.

### 1166 **6.4 Element <saml:Assertion>: XACMLPolicy Assertion**

1167 A <saml:Assertion> instance MAY contain an XACMLPolicy Statement as shown in the following  
1168 non-normative example:

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

1169 An instance of a <saml:Assertion> element containing an XACMLPolicy Statement is called an  
1170 XACMLPolicy Assertion in this Profile.

1171 When an XACMLPolicy Assertion is part of a response to an <xacml-samlp:XACMLPolicyQuery>,  
1172 then the XACMLPolicy Assertion MUST contain exactly one XACMLPolicy Statement, which in turn MAY  
1173 contain any number of XACML Policy and PolicySet instances.

1174 This Profile imposes the following requirements and restrictions on the <saml:Assertion> element  
1175 beyond those specified in SAML 2.0 when used as an XACMLPolicy Assertion.

1176 <saml:Issuer> [Required]

1177 The <saml:Issuer> element is a required element for holding information about “the SAML  
1178 authority that is making the claim(s) in the assertion” [SAML].

1179 In order to support 3<sup>rd</sup> party digital signatures, this Profile does NOT require that the identity provided  
1180 in the <saml:Issuer> element refer to the entity that signs the XACMLPolicy Assertion. It is up to  
1181 the relying party to determine whether it has an appropriate trust relationship with the authority that  
1182 signs the XACMLPolicy Assertion.

1183 <ds:Signature> [Optional]

1184 The <ds:Signature> element is an optional element for holding “An XML Signature that  
1185 authenticates the assertion, as described [in Section 5 of the SAML 2.0 core specification[SAML]].”

1186 A <ds:Signature> instance MAY be used in an XACMLPolicy Assertion. In order to support 3<sup>rd</sup>  
1187 party digital signatures, this Profile does NOT require that the identity provided in the  
1188 <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Assertion. It is up to the  
1189 relying party to determine whether it has an appropriate trust relationship with the authority that signs  
1190 the XACMLPolicy Assertion.

1191 A relying party SHOULD verify any signature included in the XACMLPolicy Assertion and SHOULD  
1192 NOT use information derived from the XACMLPolicy Assertion unless the signature is verified  
1193 successfully.

1194 <saml:Subject> [Optional]

1195 The <saml:Subject> element MUST NOT be included in an XACMLPolicy Assertion. Instead, the  
1196 Subjects of an XACMLPolicy Assertion are specified in the XACML Policy and PolicySet elements  
1197 contained in the enclosed XACMLPolicy Statement.

1198 <saml:Conditions> [Optional]

1199 The <saml:Conditions> element is an optional element that is used for “conditions that MUST be  
1200 taken into account in assessing the validity of and/or using the assertion” [SAML].

1201 The <saml:Conditions> instance SHOULD contain NotBefore and NotOnOrAfter XML  
1202 attributes to specify the limits on the validity of the XACMLPolicy Assertion. If these XML attributes  
1203 are present, the relying party SHOULD ensure that an <xacml-context:Response> taken from  
1204 the XACMLPolicy Assertion is used only during the XACMLPolicy Assertion’s specified validity  
1205 period.

## 1206 6.5 Element <xacml-samlp:XACMLPolicyQuery>

1207 An instance of the <xacml-samlp:XACMLPolicyQuery> protocol element MAY be used by a PDP or  
1208 application to request XACML <xacml:Policy> or <xacml:PolicySet> instances from an on-line  
1209 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>
```

1210 The <xacml-samlp:XACMLPolicyQuery> element is of <xacml-samlp:XACMLPolicyQueryType>  
1211 complex type, which is an extension to the SAML-defined <samlp:RequestAbstractType>.

1212 The <xacml-samlp:XACMLPolicyQuery> element contains zero or more of the following elements in  
1213 addition to those defined for the <samlp:RequestAbstractType>:

1214 <xacml-context:Request> [Any Number]

1215 An XACML Request Context. All XACML <xacml:Policy> and <xacml:PolicySet> instances  
1216 potentially applicable to this Request that the requester is authorized to receive MUST be returned.  
1217 The concept of “applicability” in the XACML context is defined in the XACML 3.0 Specification  
1218 **[XACML3]**. Any superset of applicable policies MAY be returned; for example, all policies having  
1219 top-level Target elements that match the Request MAY be returned.

1220 <xacml:PolicySetIdReference> [Any Number]

1221 Identifies an XACML <xacml:PolicySet> instance to be returned.

1222 <xacml:PolicyIdReference> [Any Number]

1223 Identifies an XACML <xacml:Policy> instance to be returned.

1224 *Non-normative note: The <xacml-samlp:XACMLPolicyQuery> is not intended as a robust*  
1225 *provisioning protocol. Users requiring such a protocol may consider using the OASIS Service*  
1226 *Provisioning Markup Language (SPML). Note that the SAML-defined ID XML attribute is a required*  
1227 *component of an instance of <samlp:RequestAbstractType> that the <xacml-*  
1228 *samlp:XACMLPolicyQuery> extends and MAY be used to correlate the <xacml-*  
1229 *samlp:XACMLPolicyQuery> with the corresponding XACMLPolicy Response.*

## 1230 6.6 Element <samlp:Response>: XACMLPolicy Response

1231 A <samlp:Response> instance MAY contain an XACMLPolicy Assertion. An instance of such a  
1232 <samlp:Response> element is called an XACMLPolicy Response in this Profile. An XACMLPolicy  
1233 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>
```

1234 An instance of a <samlp:Response> element that contains an XACMLPolicy Assertion is called an  
1235 XACMLPolicy Response in this Profile. Such a Response MUST be used as the response to an  
1236 <xacml-samlp:XACMLPolicyQuery>. It MAY be used to convey or store XACML policies for other  
1237 purposes.

1238 This Profile imposes the following requirements and restrictions on the <samlp:Response> element in  
1239 addition to those specified in SAML 2.0 when used as an XACMLPolicy Response.

1240 <saml:Issuer> [Optional]

1241 The <saml:Issuer> element identifies the originator of the contained XACML Policy, which MAY  
1242 be the entity that generated the XACMLPolicy Response message. [SAML].

1243 In order to support 3<sup>rd</sup> party digital signatures, this Profile does NOT require that the identity provided  
1244 in the <saml:Issuer> element refer to the entity that signs the XACMLPolicy Response. It is up to  
1245 the relying party to determine whether it has an appropriate trust relationship with the authority that  
1246 signs the XACMLPolicy Response.

1247 <ds:Signature> [Optional]

1248 The <ds:Signature> element is an optional element for holding "An XML Signature that  
1249 authenticates the responder and provides message integrity" [SAML].

1250 A <ds:Signature> instance MAY be used in an XACMLPolicy Response. In order to support 3<sup>rd</sup>  
1251 party digital signatures, this Profile does NOT require that the identity provided in the  
1252 <saml:Issuer> instance refer to the entity that signs the XACMLPolicy Response. It is up to the  
1253 relying party to determine whether it has an appropriate trust relationship with the authority that signs  
1254 the XACMLPolicy Response.

1255 A relying party SHOULD verify any signature included in the XACMLPolicy Response and SHOULD  
1256 NOT use information derived from the XACMLPolicy Response unless the signature is verified  
1257 successfully.

1258 <saml:Assertion> [Any Number]

1259 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,  
1260 then there MUST be exactly one instance of this element that contains an XACMLPolicy Assertion  
1261 representing the response to the associated XACMLPolicy Query. If the XACMLPolicy Response is  
1262 not issued in response to an <xacml-samlp:XACMLPolicyQuery>, it MAY contain one or more  
1263 XACMLPolicy Assertions as well as other SAML or XACML Assertions.

1264 <saml:Status> [Required]

1265 If the XACMLPolicy Response is issued in response to an <xacml-samlp:XACMLPolicyQuery>,  
1266 and if it is not possible to return all policies that satisfy the <xacml-samlp:XACMLPolicyQuery>, then  
1267 a <samlp:StatusCode> value of  
1268 urn:oasis:names:tc:saml:2.0:status:TooManyResponses MUST be returned in the  
1269 <samlp:Status> element of the Response.

1270 InResponseTo [Optional]

1271 This optional XML attribute is “A reference to the identifier of the request to which the response  
1272 corresponds.” When the XACMLPolicy Response is issued in response to an <xacml-  
1273 samlp:XACMLPolicyQuery>, this XML attribute MUST contain the value of the ID XML attribute  
1274 from the <xacml-samlp:XACMLPolicyQuery> to which this is a response. This allows the  
1275 receiver to correlate the XACMLPolicy Response with the corresponding XACMLPolicy Query.

## 1276 **6.7 Policy references and Policy assertions**

1277 It may be noted that in relation to a policy assertion, there are three broad classes of policies to consider  
1278 when resolving policy references: the top level policy in the policy assertion, the policies in the <xacml-  
1279 samlp:ReferencedPolicies> element and policies external to the policy assertion, available to a PDP by  
1280 other means.

1281 How policy references are resolved across these three classes of policies depends on the particular  
1282 case and problem for which the policy assertion is used. Therefore policy reference resolving is  
1283 implementation defined with respect to policy assertions.

---

1284 **7 Advice**

1285 This Section describes how to include XACMLAuthzDecision Assertion and XACMLPolicy Assertion  
1286 instances as advice in another SAML Assertion instance.

1287 **7.1 Element <saml:Advice>**

1288 A SAML Assertion MAY include a <saml:Advice> element containing “Additional information related to  
1289 the assertion that assists processing in certain situations but which MAY be ignored [without affecting  
1290 either the semantics or the validity of the assertion] by applications that do not understand the advice or  
1291 do not wish to make use of it.” [SAML] An XACMLAuthzDecision Assertion or XACMLPolicy Assertion  
1292 may be used in the Advice element as shown in the following non-normative example:

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



1293  
1294

---

## 8 Using an XACML Authorization Decision as an Authorization Token

1295 This Section of the Profile describes how to use an XACMLAuthzDecision Statement as a security and  
1296 privacy authorization token as part of a SOAP message exchange in a Web Services context. This token  
1297 MAY be used by a client to convey an authorization decision from a trusted 3<sup>rd</sup> party to a service. A Web  
1298 Service MAY use such a token to determine that the client is authorized to access information involved in  
1299 the Web Services interaction.

1300 In a Web Services context, an instance of an XACMLAuthzDecision Assertion MAY be used as an  
1301 authorization token in the Web Services Security [WSS] and [WSS-Core] `wsse:Security` Header of a  
1302 SOAP message. When used in this way, the XACMLAuthzDecision Statement in the  
1303 XACMLAuthzDecision Assertion MUST include the corresponding XACML Request Context. This allows  
1304 the Web service to determine whether the `<xacml-context:Attribute>` instances in the Request  
1305 correspond to the access that the client requires as part of the Web Service interaction. The  
1306 XACMLAuthzDecision Assertion SHOULD be signed by a Policy Decision Point trusted by the Web  
1307 Service.

1308 A Web Service MAY use this token to determine that a trusted 3<sup>rd</sup> party has evaluated an XACML  
1309 Request Context that is relevant to the invocation of the service, and has reported an authorization  
1310 decision. The service SHOULD verify that the signature on the XACMLAuthzDecision Assertion is from  
1311 a Policy Decision Point that the service trusts. The service SHOULD verify that the validity period of the  
1312 XACMLAuthzDecision Assertion includes the time at which the Web Service interaction will access the  
1313 information or resource to which the Request Context applies. The service SHOULD verify that the  
1314 `<xacml-context:Attribute>` instances contained in the XACML `<xacml-context:Request>`  
1315 element correctly describe the information or resource access that needs to be authorized as part of this  
1316 Web Service interaction.

---

## 9 Conformance

1317

1318 Implementations of this Profile MAY implement certain subsets of the described functionality. Each  
1319 implementation MUST clearly identify the subsets it implements using the following identifiers.

1320 An implementation of this Profile is a conforming *SAML Attribute* implementation if the implementation  
1321 conforms to Section 2 of this Profile. The following URI MUST be used as the identifier for this  
1322 functionality:

1323 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:attrs:all`

1324 An implementation of this Profile is a conforming *SOAP Attributes as XACMLAuthzDecisionQuery*  
1325 implementation if the implementation conforms to Section 3.1 of this Profile. The following URI MUST be  
1326 used as the identifier for this functionality:

1327 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:authzQuery`

1328 An implementation of this Profile is a conforming *SOAP Attributes as SAML Attribute Assertion*  
1329 implementation if the implementation conforms to Section 3.2 of this Profile. The following URI MUST be  
1330 used as the identifier for this functionality:

1331 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:SOAP:attrAssertion`

1332

1333 An implementation of this Profile is a conforming *XACML Authz Decision without Policies* implementation  
1334 if the implementation conforms to all parts of Section 4 of this Profile excluding the `<xacml:Policy>`,  
1335 `<xacml:PolicySet>`, and `<xacml-samlp:ReferencedPolicies>` elements and their sub-elements  
1336 and the `CombinePolicies` XML attribute in the `<xacml-samlp:XACMLAuthzDecisionQuery>`.  
1337 XACML 3.0 implementations MUST support the `<xacml-samlp:AdditionalAttributes>` element  
1338 and its sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0  
1339 implementations MUST NOT support the `<xacml-samlp:AdditionalAttributes>` element and its  
1340 sub-elements in the `<xacml-samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be  
1341 used as the identifier for this functionality:

1342 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:noPolicies`

1343 An implementation of this Profile is a conforming *XACML Authz Decision with Policies* implementation if  
1344 the implementation conforms to all parts of Section 4 of this Profile. XACML 3.0 implementations MUST  
1345 support the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-  
1346 samlp:XACMLAuthzDecisionQuery>`. XACML 1.0, 1.1, and 2.0 implementations MUST NOT support  
1347 the `<xacml-samlp:AdditionalAttributes>` element and its sub-elements in the `<xacml-  
1348 samlp:XACMLAuthzDecisionQuery>`. The following URI MUST be used as the identifier for this  
1349 functionality:

1350 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecision:withPolicies`

1351 An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust with Policies*  
1352 implementation if it conforms to section 5 in its entirety as described in the previous paragraph. The  
1353 following URI MUST be used as the identifier for this functionality.

1354 `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:withP`  
1355 `olicies`

1356 An implementation of this Profile is a conforming *XACML Authz Decision using WS-Trust without Policies*  
1357 *implementation if it conforms to section 5, with the exceptions relating to policies and additional attributes*  
1358 *noted above. The following URI MUST be used as the identifier for this functionality.*

1359        `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzDecisionWSTrust:noPol`  
1360        `icies`

1361        An implementation of this Profile is a conforming *XACML Policies* implementation if the implementation  
1362        conforms to Section 6 of this Profile. The following URI MUST be used as the identifier for this  
1363        functionality:

1364        `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:policies`

1365        An implementation of this Profile is a conforming *SAML Advice* implementation if the implementation  
1366        conforms to Section 7 of this Profile. The following URI MUST be used as the identifier for this  
1367        functionality:

1368        `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:adviceSAML`

1369        An implementation of this Profile is a conforming *XACML Authz Token* implementation if the  
1370        implementation conforms to Section 8 of this Profile. The following URI MUST be used as the identifier  
1371        for this functionality:

1372        `urn:oasis:names:tc:xacml:3.0:profile:saml2.0:v2:authzToken`

1373

---

1374 **Appendix A. Acknowledgments**

1375 The following individuals have participated in the creation of this specification and are gratefully  
1376 acknowledged

1377 Anil Saldhana

1378 Anil Tappetla

1379 Anne Anderson

1380 Anthony Nadalin

1381 Bill Parducci

1382 Craig Forster

1383 David Chadwick

1384 David Staggs

1385 Dilli Arumugam

1386 Duane DeCouteau

1387 Erik Rissanen

1388 Gareth Richards

1389 Hal Lockhart

1390 Jan Herrmann

1391 John Tolbert

1392 Ludwig Seitz

1393 Michiharu Kudo

1394 Naomaru Itoi

1395 Paul Tyson

1396 Prateek Mishra

1397 Rich Levinson

1398 Ronald Jacobson

1399 Seth Proctor

1400 Sridhar Muppidi

1401 Tim Moses

1402 Vernon Murdoch

1403

1404

## 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 <b>potentially</b> applicable policies; remove Target element; change Choice lower bound from 0 to 1 and remove case where no elements included; add non-normative note to consider SPML for provisioning protocol -4.5 Response: Use valid ID values in example; add <samp:Status> element saying to use SAML TooManyResponses StatusCode if unable to return all applicable policies -7 Insert the "XACML Authorization Token" section from the WS-XACML profile -Schemas: create versions specific to each XACML version -Protocol schema: remove XACMLPolicyQuery Target element, change Choice lower bound from 0 to 1 -Protocol schema: add Administrative Policy elements.
WD 4	15 June 2007	Anne Anderson	-throughout: used actual schema elements rather than invented names except when speaking about instances

			<p>embedded in other instances (e.g. &lt;saml:Attribute&gt; rather than SAML Attribute, but SAML Attribute Response rather than &lt;samlp:Response&gt;).</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 &lt;xacml-context:Attribute&gt; 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 <a href="http://docs.oasis-open.org/xacml/">http://docs.oasis-open.org/xacml/</a> 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 &lt;ReferencedPolicies&gt; 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>

