

# SAML 2.0 Single Sign-On with Constrained Delegation

## Working Draft 01, 1 October 2005

### Document identifier:

draft-cantor-saml-ss0-delegation-01

### Location:

<http://shibboleth.internet2.edu/shibboleth-documents.html>

### Editors:

Scott Cantor ([cantor.2@osu.edu](mailto:cantor.2@osu.edu)), The Ohio State University

### Contributors:

TBD

### Abstract:

This is a proposed set of profiles for enabling constrained delegation in conjunction with the authentication request protocol defined in the OASIS SAML 2.0 specification (<http://www.oasis-open.org/committees/security>). Readers should be familiar with that specification before reading this document. Of particular interest is the ability to extend SAML-authenticated browser- or other client-based web interactions to encompass authentication to additional back-end services.

This is a **working draft** and the text may change before completion. Please submit comments to the shibboleth-dev mailing list (see <http://shibboleth.internet2.edu/> for subscription details).

**NOTE: Much of this preliminary document has been superseded by discussion and significant work within both the SAML TC and the Liberty Technical Expert Group. It is the author's opinion that a significant set of use cases relevant to Shibboleth can be addressed by incorporating significant portions of Liberty ID-WSF 2.0, which is due for final release early in 2006.**

**Readers should take this into consideration and understand that in any case, NO commitments to support any of the features discussed in this document, nor any alternative mechanisms, are formally part of the Shibboleth 2.0 feature set. Until such time as a set of credible standards exist, any such solutions should be considered proprietary and non-interoperable.**

# Table of Contents

27	1 Introduction.....	4
28	1.1 Notation.....	4
29	2 Architectural Overview.....	6
30	2.1 Single Sign-On with Delegation.....	6
31	3 Profiles.....	8
32	3.1 Authentication Request Delegation Profile.....	8
33	3.1.1 Required Information.....	8
34	3.1.2 Profile Description.....	8
35	3.1.3 <samlp:AuthnRequest> Usage.....	8
36	3.1.3.1 <saml:Subject>.....	8
37	3.1.3.2 <saml:Conditions>.....	9
38	3.1.4 <saml:Assertion> Usage.....	10
39	3.1.4.1 <saml:SubjectConfirmation>.....	10
40	3.1.4.2 <saml:AudienceRestriction>.....	11
41	3.1.5 Processing Rules.....	11
42	3.1.6 Example.....	12
43	3.1.7 Metadata Considerations.....	13
44	3.1.8 Security and Privacy Considerations.....	13
45	3.2 Browser/ECP Single Sign-On Profiles With Delegation.....	14
46	3.2.1 Required Information.....	14
47	3.2.2 Profile Description.....	14
48	3.2.3 Processing Rules.....	14
49	3.2.4 Metadata Considerations.....	15
50	3.2.4.1 Example.....	15
51	3.2.5 Security and Privacy Considerations.....	15
52	3.3 SAML Token Service Profile.....	15
53	3.3.1 Required Information.....	15
54	3.3.2 Profile Overview.....	16
55	3.3.3 Profile Description.....	17
56	3.3.3.1 Requester Determines Identity Provider.....	17
57	3.3.3.2 <samlp:AuthnRequest> issued by Requester to Identity Provider.....	17
58	3.3.3.3 Identity Provider identifies Requester.....	17
59	3.3.3.4 Identity Provider issues and returns <samlp:Response> to Requester.....	17
60	3.3.4 Use of Authentication Request Protocol.....	17
61	3.3.4.1 <samlp:AuthnRequest> Usage.....	17
62	3.3.4.2 <samlp:Response> Usage.....	18
63	3.3.4.3 Processing Rules.....	18
64	3.3.5 Metadata Considerations.....	19
65	3.3.5.1 Example.....	19
66	3.3.6 Security and Privacy Considerations.....	19
67	3.4 SOAP Application Profile.....	20
68	3.4.1 Required Information.....	20
69	3.4.2 Profile Description.....	20
70	3.4.2.1 <wsse:Security> Header Content.....	20
71	3.4.2.2 Originator Processing Rules.....	21
72	3.4.2.3 Recipient Processing Rules.....	21
73	3.4.3 Example.....	22
74	3.4.4 Metadata Considerations.....	23
75	3.4.5 Security and Privacy Considerations.....	23
76	3.5 SAML Authentication to SAML Token Service.....	23
77	3.5.1 Required Information.....	23

78	3.5.2 Profile Description.....	23
79	3.5.3 Processing Rules.....	24
80	3.5.4 Metadata Considerations.....	24
81	3.5.5 Security and Privacy Considerations.....	24
82	4 References.....	25
83	4.1 Normative References.....	25
84	4.2 Non-Normative References.....	26
85		

# 86 1 Introduction

87 The SAML 2.0 specification includes a pair of profiles enabling web single sign-on (SSO) that address  
88 both traditional web browsers and more functional HTTP user agents (termed enhanced clients) that can  
89 participate more directly in the authentication exchange. These profiles are by design restricted to an  
90 exchange between a single identity provider (IdP) and service provider (SP).

91 However, in many web application scenarios, the SP may need to access additional services on the  
92 principal's behalf while carrying out its work. These services may be accessed with a variety of  
93 application protocols, although SOAP-based protocols are increasing in popularity. Concrete use cases  
94 that fit this conceptual model include meta-search applications that access multiple content repositories  
95 and databases, portals that access information or transaction services programmatically on behalf of  
96 their users, grid computing, and many others.

97 Often, the principal, SP, and any additional services share a security domain, which allows various  
98 solutions to be employed, such as permitting the SP to impersonate all principals within the domain,  
99 employing proprietary security protocols, or using symmetric-key-based technology such as Kerberos. In  
100 order to extend this model to a federated scenario, such as is supported by SAML, additional  
101 standardization is required to produce a solution that provides adequate control to all the parties to the  
102 transaction. Various companies have proposed proprietary solutions that could be profiled to address  
103 such use cases for SOAP-based applications. The Liberty Alliance Project  
104 (<http://www.projectliberty.org/>), an open consortium, has also proposed specifications built around SAML  
105 that include possible solutions.

106 Such solutions attack a very general problem space with a great deal of completeness and flexibility, but  
107 this also leads to complexity. This proposal describes a simpler set of profiles built more directly on  
108 SAML 2.0 with additional simplifying assumptions in the hope that the adoption of strong federated  
109 security can be fostered without requiring as much additional software or knowledge. They may also lend  
110 themselves more naturally to the incorporation of non-SOAP-based services, as SAML is generally  
111 agnostic to the application protocols for which it can be profiled.

112 The general approach taken is one of constrained delegation, specifically the use of SAML assertions to  
113 enable an SP to act in a limited but transparent fashion on behalf of a principal when communicating  
114 with an additional SP. Policy can then be enforced that addresses all four parties to a given transaction  
115 (the principal, IdP, delegate SP, and back-end SP).

116 Unless specifically noted, nothing in this document should be taken to conflict with the SAML 2.0  
117 specification, or any bindings and profiles referenced within it. Readers are advised to familiarize  
118 themselves with that specification first.

## 119 1.1 Notation

120 This specification uses normative text to describe the use of SAML 2.0 and additional SAML profiles.

121 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD  
122 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as  
123 described in [RFC 2119]:

124       ...they MUST only be used where it is actually required for interoperation or to limit behavior  
125       which has potential for causing harm (e.g., limiting retransmissions)...

126 These keywords are thus capitalized when used to unambiguously specify requirements over protocol  
127 and application features and behavior that affect the interoperability and security of implementations.  
128 When these words are not capitalized, they are meant in their natural-language sense.

129       Listings of XML schemas appear like this.

130       Example code listings appear like this.

132 Conventional XML namespace prefixes are used throughout the listings in this specification to stand for  
133 their respective namespaces as follows, whether or not a namespace declaration is present in the  
134 example:

- 135 • The prefix `saml:` stands for the SAML 2.0 assertion namespace,  
136 `urn:oasis:names:tc:SAML:2.0:assertion`
- 137 • The prefix `samlp:` stands for the SAML 2.0 protocol namespace,  
138 `urn:oasis:names:tc:SAML:2.0:protocol`
- 139 • The prefix `md:` stands for the SAML 2.0 metadata namespace,  
140 `urn:oasis:names:tc:SAML:2.0:metadata`
- 141 • The prefix `ds:` stands for the W3C XML Signature namespace,  
142 <http://www.w3.org/2000/09/xmldsig#>
- 143 • The prefix `xenc:` stands for the W3C XML Encryption namespace,  
144 <http://www.w3.org/2001/04/xmlenc#>
- 145 • The prefix `wsse:` stands for the WSS 1.0 Security Extension namespace,  
146 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd>
- 147 • The prefix `wsse11:` stands for the WSS 1.1 Security Extension namespace,  
148 <http://docs.oasis-open.org/wss/2005/xx/oasis-2004xx-wss-wssecurity-secext-1.1.xsd>
- 149 • The prefix `wsu:` stands for the WSS 1.0 Utility namespace,  
150 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd>
- 151 • The prefix `S:` stands for the SOAP 1.1 namespace,  
152 <http://schemas.xmlsoap.org/soap/envelope/>
- 153 • The prefix `xsd:` stands for the W3C XML Schema namespace,  
154 <http://www.w3.org/2001/XMLSchema>  
155 in example listings. In schema listings, this is the default namespace and no prefix is shown.
- 156 • The prefix `xsi:` stands for the W3C XML Schema instance namespace,  
157 <http://www.w3.org/2001/XMLSchema-instance>  
158 in example listings.

159 This specification uses the following typographical conventions in text: `<SAMLElement>`,  
160 `<ns:ForeignElement>`, Attribute, **Datatype**, OtherCode.

## 2 Architectural Overview

In order to constrain the problem space, we consider only application scenarios that can be mapped to an interaction between the following set of actors:

- User Agent acting on behalf of a Principal
- Identity Provider (IdP)
- Service Provider A (SPa)
- Service Provider B (SPb)

The SAML 2.0 SSO profiles provide a means by which the principal can authenticate to the IdP by unspecified means and obtain a SAML assertion for use by SPa to authenticate the principal (assuming SPa is offering HTTP-based services). The assertion can contain details about the authentication event as well as attributes about the principal. Here, we add a fourth party, SPb that offers services, usually but not necessarily specific to the principal. For architectural purposes, we do not specify the particular protocol used between SPa and SPb, although a profile for the use of SOAP is included in this document.

The principal would like to allow SPa to access SPb on his or her behalf, but only in a limited context. That is, while SPa can most likely authenticate itself to SPb (for example using TLS or a digital signature), this proposal gives SPa the ability to prove to SPb that it is authorized to act on behalf of the principal at a particular point in time by presenting a SAML assertion as evidence, without requiring that SPb allow SPa to do so in all cases, or indeed ever again.

This interaction should be a natural extension of the use of SAML to authenticate the principal to SPa in the first place. In fact, it may be the same assertion. While this architectural example presumes SPa is HTTP-based, any profile of SAML that results in the issuance of an assertion to authenticate the principal can in theory be combined with the profiles in this document.

Finally, it should be possible to repeat the process, thus enabling additional service providers (SPc, SPd, etc.) to be accessed by SPa, enabling SPb to access additional services on behalf of the principal itself by requesting additional assertions, etc., allowing more complex application choreography.

### 2.1 Single Sign-On with Delegation

The following sequence diagram illustrates the set of required and optional interactions when using the SSO profile to achieve constrained delegation.

We do not constrain the SAML bindings that can be utilized in each step. See [SAML2Bind] for detailed descriptions of bindings that may be used to transmit the SAML protocol messages. We also omit the process of identity provider discovery, which may impact the exact exchange of messages in the first two steps until the identity provider is determined.

TODO: Extend SSO diagram

#### 1. HTTP Request to Service Provider A

In step 1, the principal, via an HTTP user agent, makes an HTTP request for a secured resource at SPa without a security context.

#### 2. <samlp:AuthnRequest> issued by Service Provider A to Identity Provider

In step 2, SPa issues a <samlp:AuthnRequest> message to the IdP, to be delivered by the user agent. This request MAY specify that SPa wishes to be a delegate, and MAY identify additional service providers with which SPa expects to communicate on behalf of the user.

202 **3. Identity Provider identifies Principal**

203 In step 3, the principal is identified by the IdP by some means outside the scope of this  
204 specification. This may require a new act of authentication, or it may reuse an existing  
205 authenticated session.

206 **4. Identity Provider issues <samlp:Response> to Service Provider A**

207 In step 4, the IdP issues a <samlp:Response> message to be delivered by the user agent to  
208 SPa. The response can contain one or more assertions, or an error status to indicate failure.  
209 These assertions will, at minimum, authenticate the principal to SPa, and may also be usable by  
210 SPa to authenticate itself on behalf of the principal to additional service providers, to obtain  
211 additional assertions from the IdP for this purpose, or both.

212 **5. SPa grants or denies access to Principal**

213 In step 5, SPa responds to the principal's user agent with an error, or establishes its own security  
214 context for the principal and returns the requested resource. During this or subsequent  
215 interactions with the principal, SPa may need to access additional service providers on behalf of  
216 the principal.

217 **6. <samlp:AuthnRequest> issued by Service Provider A to Identity Provider**

218 In step 6, SPa discovers that it does not already have an assertion with which it can authenticate  
219 itself to SPb on behalf of the principal. It must ask for an additional assertion from the IdP for this  
220 purpose, possibly using the assertion it obtained in step 4 to demonstrate its right to do so. This  
221 exchange may be directly carried out between SPa and the IdP, or it may be mediated by the  
222 user agent.

223 **7. Identity Provider issues <samlp:Response> to Service Provider A**

224 In step 7, the IdP returns a <samlp:Response> message (either directly, or via the user agent)  
225 that contains the assertion requested by SPa or an error status.

226 **8. Service Provider A accesses Service Provider B**

227 In step 8, SPa uses the assertion it received in step 4 or 7 above to authenticate itself to SPb on  
228 behalf of the principal. This step can make use of any application security profiles that allow for  
229 the use of SAML and proof of key possession. A profile for using SOAP is included in this  
230 document.

231 Note that an IdP can initiate this sequence at step 4 and issue an unsolicited <samlp:Response>  
232 message to SPa without the preceding steps.

## 233 3 Profiles

234 This section defines a set of composable SAML profiles that extend and build on the interactions found in  
235 the existing SAML 2.0 profiles [SAML2Prof], and offer new stand-alone capabilities.

### 236 3.1 Authentication Request Delegation Profile

237 The `<samlp:AuthnRequest>` element [SAML2Core] is a very general request mechanism, with a large  
238 number of optional elements. As profiled for use in requesting SSO [SAML2Prof], some of the more  
239 complex options are ignored or left unspecified (e.g. the use of a `<saml:Conditions>` element). The  
240 following profile makes use of some of this flexibility to describe a mechanism by which an  
241 `<samlp:AuthnRequest>` can include a request to embed delegation support in the resulting assertion.

#### 242 3.1.1 Required Information

243 **Identification:** urn:mace:shibboleth:2.0:profiles:delegation

244 **Contact Information:** [shibboleth-dev@internet2.edu](mailto:shibboleth-dev@internet2.edu)

245 **Description:** Given below.

246 **Updates:** Nothing

#### 247 3.1.2 Profile Description

248 This profile defines the usage of a set of elements in the `<samlp:AuthnRequest>` element that, when  
249 included in a request processed by an identity provider's Single Sign-On Service endpoint, result in an  
250 assertion that contains conditions and subject confirmation rules that, taken together, make it usable as a  
251 delegation token.

252 Referring back to section 2.1, this profile concerns the mechanism by which the issuer of a  
253 `<samlp:AuthnRequest>` can indicate that the resulting `<saml:Assertion>` is intended to be used by  
254 a given service provider (SPa) to access a second service provider (SPb) on behalf of a principal. For  
255 this to be possible, the assertion must include specific content enabling SPb to securely establish SPa's  
256 right to do so. This profile also permits the optimization of requesting a token that is usable  
257 simultaneously by SPa at multiple relying parties (though see section 3.1.8 for some of the privacy  
258 considerations of this approach).

259 The use of the `<samlp:AuthnRequest>` message and the resulting `<samlp:Response>` message  
260 generally occurs in the context of an additional profile of use, such as the Web Browser SSO and ECP  
261 profiles [SAML2Prof] or one of the subsequent profiles described in this document. This profile  
262 specifically addresses only the semantics and processing rules of the elements discussed below.

#### 263 3.1.3 `<samlp:AuthnRequest>` Usage

264 The following sections define semantics of the `<saml:Subject>` and `<saml:Conditions>` elements  
265 related to delegation.

##### 266 3.1.3.1 `<saml:Subject>`

267 The `<saml:Subject>` element in a `<samlp:AuthnRequest>` message specifies the requested  
268 subject of the resulting assertion(s) [SAML2Core]. It serves two purposes within this profile:

- 269       • Specifying the principal's `<saml:NameID>` when the entity presenting the request to the IdP is  
270       not the principal (for example, an SP making a request for an additional assertion it can use to  
271       act on the principal's behalf).
- 272       • Specifying subject confirmation keys or other restrictions on the use of the resulting assertion(s)  
273       if non-default behavior is required.

274       Regardless of its content, including a `<saml:Subject>` element is a request for the IdP to embed a  
275       strongly matching `<saml:Subject>` element in the resulting assertions ([SAML2Core], section 3.3.4).

276       In the context of delegation, this involves a request to include one or more  
277       `<saml:SubjectConfirmation>` elements with a `Method` attribute equal to

278       `urn:oasis:names:tc:SAML:2.0:cm:holder-of-key`

279       with or without designating a specific key. Each such element is a request to permit the specified entity to  
280       confirm itself as, and thus impersonate, the subject.

281       Each `<saml:SubjectConfirmation>` element in the request MUST include a `<saml:BaseID>`,  
282       `<saml:NameID>`, or `<saml:EncryptedID>` element specifying the entity to be given delegation  
283       capability. It MAY also contain a `<saml:SubjectConfirmationData>` element (with an `xsi:type` of  
284       **`saml:KeyInfoConfirmationDataType`**) containing at least one `<ds:KeyInfo>` element specifying a key  
285       belonging to the entity to be given delegation capability.

286       Each `<saml:SubjectConfirmation>` element MAY include a  
287       `<saml:SubjectConfirmationData>` element containing the `Address`, `Recipient`, `NotBefore`,  
288       and/or `NotOnOrAfter` attributes to further restrict the use of the resulting assertion(s), as defined by  
289       [SAML2Core].

290       Together, these elements permit the requester to specify the delegate entity and/or key. The IdP can use  
291       the information to prepare and verify the appropriate `<saml:SubjectConfirmation>` element to  
292       place in the resulting assertion(s). For example, an identifier by itself permits the IdP to select a default  
293       confirmation key associated with that entity, perhaps based on metadata.

### 294       **3.1.3.2 <saml:Conditions>**

295       The `<saml:Conditions>` element in a `<samlp:AuthnRequest>` message specifies the SAML  
296       conditions the requester expects to limit the validity and/or use of the resulting assertion(s)  
297       [SAML2Core]. It serves three purposes within this profile:

- 298       • Specifies that the request is being made in conjunction with this profile
- 299       • Specifying the validity period of the resulting assertion(s) to customize the length of time during  
300       which the assertion(s) can be used.
- 301       • Specifying the relying parties with whom the requester expects to communicate using the  
302       assertion(s).

303       A `<saml:Conditions>` element MUST be included in the `<saml:AuthnRequest>` message and  
304       MUST contain a `<saml:AudienceRestriction>` containing a single `<saml:Audience>` element  
305       with the value

306       `urn:mace:shibboleth:2.0:profiles:delegation`

307       which indicates the request is for an assertion constructed in accordance with this profile. The use of the  
308       elements discussed in this profile in a request that does not contain this `<saml:Audience>` value are  
309       unspecified.

310       The following constructs MAY also be included within the `<saml:Conditions>` element:

311 <saml:AudienceRestriction>

312         Separate from the mandatory element discussed above, this Indicates that the resulting  
313         assertion(s) SHOULD contain a matching <saml:AudienceRestriction> element. Used to  
314         indicate that the resulting assertion(s) are intended to be usable with the relying parties whose  
315         unique identifiers are contained in the enclosed <saml:Audience> elements.

316 NotBefore

317         Indicates that the resulting assertion(s) SHOULD contain a matching NotBefore attribute in the  
318         <saml:Conditions> element.

319 NotOnOrAfter

320         Indicates that the resulting assertion(s) SHOULD contain a matching NotOnOrAfter attribute in  
321         the <saml:Conditions> element.

322 In each case, the IdP is given the final discretion over what to include in the resulting assertion(s). While  
323 it MAY fundamentally alter the corresponding values that it includes if necessary, it SHOULD return an  
324 error if it cannot at least partially fulfill the request.

325 Note that the <saml:AudienceRestriction> element MAY include the unique identifier of the IdP  
326 itself, representing a request for the resulting assertion(s) to be usable as a means of authentication to  
327 the IdP on behalf of the user, enabling additional delegation tokens to be obtained.

### 328 3.1.4 <saml:Assertion> Usage

329 For the purpose of supporting delegation, the <saml:SubjectConfirmation> element is used to  
330 identify the delegate and the key to be used by it to confirm its identity, and the  
331 <saml:AudienceRestriction> element is used to signal the use of this profile and to identify the  
332 relying parties with whom the delegate can use the assertion to authenticate and authorize its activities on  
333 behalf of the principal. Other assertion contents MUST be interpreted as defined by the SAML 2.0  
334 Assertions and Protocols specification [SAML2Core] and any applicable profiles of use.

#### 335 3.1.4.1 <saml:SubjectConfirmation>

336 | A Delegation [permissionrelationship](#) in an assertion [created in accordance with this profile](#) is expressed  
337 by including a <saml:SubjectConfirmation> element with a Method attribute equal to

338         urn:oasis:names:tc:SAML:2.0:cm:holder-of-key

339 in the assertion's <saml:Subject> element. Each delegate is identified with a distinct  
340 <saml:SubjectConfirmation> element that MUST contain:

- 341         • a <saml:BaseID>, <saml:NameID>, or <saml:EncryptedID> element that names the  
342         delegate
- 343         • a <saml:SubjectConfirmationData> element (with an xsi:type of  
344         **saml:KeyInfoConfirmationDataType**) containing at least one <ds:KeyInfo> element  
345         specifying a key belonging to the delegate

346 Multiple keys MAY be included in multiple <ds:KeyInfo> elements, but each <ds:KeyInfo> element  
347 MUST identify a single key.

348 A <saml:SubjectConfirmationData> element MAY include a standard set of XML attributes that  
349 restrict the confirmation process beyond requiring proof of key possession, such as Address,  
350 Recipient, NotBefore, and NotOnOrAfter [SAML2Core]. These attributes can be used to “fine-  
351 tune” the capability of the corresponding delegate to use the assertion.

### 352 **3.1.4.2 <saml:AudienceRestriction>**

353 At least one <saml:AudienceRestriction> element containing a <saml:Audience> element with  
354 the value of

355 urn:mace:shibboleth:2.0:profiles:delegation

356 MUST be included in an assertion created in accordance with this profile. The use of the elements  
357 discussed in this profile in an assertion that does not contain this <saml:Audience> value are  
358 unspecified.

359 Additional <saml:AudienceRestriction> elements MAY be included to constrain the delegation  
360 “scope” of an assertion by identifying a set of one or more relying parties who can accept the assertion.  
361 An SP (or IdP) can be designated by including its unique identifier in an included <saml:Audience>  
362 element. Alternatively, an affiliated set of service providers can be designated as a unit by including the  
363 affiliation's identifier.

### 364 **3.1.5 Processing Rules**

365 The general processing rules for the <saml:AuthnRequest> message [SAML2Core] still apply and  
366 may be further constrained by the specific profile of use. This profile deals only with the use of the  
367 optional request content discussed in section 3.1.3 and the processing of an assertion containing the  
368 content discussed in section 3.1.4 provided that the <saml:Conditions> element is present in the  
369 <saml:AuthnRequest> and <saml:Assertion> and it includes at least one  
370 <saml:AudienceRestriction> element containing a <saml:Audience> element with the value of

371 urn:mace:shibboleth:2.0:profiles:delegation

372 If no <saml:Subject> element is included, then any assertions returned by the IdP MUST contain a  
373 <saml:Subject> that identifies the presenter of the request. Assuming the requester (<Issuer>) is  
374 not the presenter, then at its discretion, the IdP MAY include a <saml:SubjectConfirmation>  
375 element establishing the requester as a delegate of the presenter.

376 If a <saml:Subject> element is included, and if it identifies the principal (using a <saml:BaseID>,  
377 <saml:NameID>, or <saml:EncryptedID> element), then the IdP MUST verify that the presenter of  
378 the request is authorized to receive assertions about that principal, typically by performing some form of  
379 authentication.

380 If a <saml:Subject> element is included, and if it includes <saml:SubjectConfirmation>  
381 elements constructed in accordance with this profile, then the IdP MAY at its discretion (or based on  
382 policies established by the principal) include equivalent <saml:SubjectConfirmation> elements in  
383 the resulting assertion(s) to permit the delegation requested. If none of the requested forms of delegation  
384 are permissible, then the IdP SHOULD generally return an error <samlp:Status> in the  
385 <samlp:Response> rather than return an assertion incapable of meeting the requester's needs. More  
386 specific profiles MAY refine or clarify error handling behavior.

387 Within a given <saml:SubjectConfirmation> element, if no confirmation key is included, then the  
388 IdP MAY utilize any trustworthy means at its disposal, such as SAML metadata [SAML2Meta], to identify  
389 a default key associated with the delegate identified in the <saml:SubjectConfirmation> element.

390 If no additional <saml:AudienceRestriction> elements are included in the <saml:Conditions>  
391 element in the request, then the IdP MAY at its discretion (or based on policies established by the  
392 principal) establish a default scope for the delegation by including such an element in the resulting  
393 assertion(s).

394 Based on the set of relying parties expected to process the resulting assertion(s), the IdP SHOULD  
395 consider the use of XML encryption when including personally identifying information. The use of the

396 transient name identifier format [SAML2Core] in conjunction with the <saml:EncryptedAttribute> element  
397 may be appropriate to conceal information from the requester and the relying parties. See section 3.1.8  
398 for a more detailed discussion of security and privacy considerations.

### 399 3.1.6 Example

400 The example below shows a <samlp:AuthnRequest> that might be used during a browser profile  
401 interaction in which a requesting SP, "https://spa.example.com/shibboleth", asks for a SSO token that  
402 can also be used by it to access a second SP, "https://spb.example.com/shibboleth". The identity of the  
403 principal is derived from authenticating the presenter of the request, as is typical during SSO.

```
404 <samlp:AuthnRequest
405   Issuer="https://spa.example.com/shibboleth"
406   IssueInstant="2003-04-17T00:46:02Z"
407   Version="2.0"
408   Destination="https://idp.example.com/shibboleth/SSO"
409   RequestID="_c7055387-af61-4fce-8b98-e2927324b306">
410   <ds:Signature>...</ds:Signature>
411   <saml:Subject>
412     <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
413       <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
414         https://spa.example.com/shibboleth
415       </saml:NameID>
416       <SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
417         <ds:KeyInfo>
418           <ds:KeyName>spa.example.com</ds:KeyName>
419         </ds:KeyInfo>
420       </SubjectConfirmationData>
421     </saml:SubjectConfirmation>
422   </saml:Subject>
423   <saml:Conditions>
424     <saml:AudienceRestriction>
425       <saml:Audience>urn:mace:shibboleth:2.0:profiles:delegation</saml:Audience>
426     </saml:AudienceRestriction>
427     <saml:AudienceRestriction>
428       <saml:Audience>https://spb.example.com/shibboleth</saml:Audience>
429     </saml:AudienceRestriction>
430   </saml:Conditions>
431 </samlp:AuthnRequest>
```

432 And now the corresponding <samlp:Response> message:

```
433 <samlp:Response
434   IssueInstant="2003-04-17T00:46:02Z"
435   Version="2.0"
436   Destination="https://spa.example.com/Shibboleth.sso/SAML/POST"
437   ResponseID="_d7055387-af61-4fce-8b98-e2927324b306"
438   InResponseTo="_c7055387-af61-4fce-8b98-e2927324b306">
439   <saml:Issuer>https://idp.example.com/shibboleth</saml:Issuer>
440   <samlp:Status>
441     <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
442   </samlp:Status>
443   <saml:Assertion
444     AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
445     IssueInstant="2003-04-17T00:46:02Z">
446     <saml:Issuer>https://idp.example.com/shibboleth</saml:Issuer>
447     <ds:Signature>...</ds:Signature>
448     <saml:Subject>
449       <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
450         3f7b3dcf-1674-4ecd-92c8-1544f346baf8
451       </saml:NameID>
452       <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
453         <saml:SubjectConfirmationData Address="127.0.0.1"
454           InResponseTo="_c7055387-af61-4fce-8b98-e2927324b306"
```

```

455     Recipient="https://spa.example.com/Shibboleth.sso/SAML/POST"
456     NotBefore="2003-04-17T00:46:02Z"
457     NotOnOrAfter="2003-04-17T00:51:02Z"
458     </saml:SubjectConfirmationData>
459   </saml:SubjectConfirmation>
460   <saml:SubjectConfirmation
461     Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
462     <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
463       https://spa.example.com/shibboleth
464     </saml:NameID>
465     <SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
466       <ds:KeyInfo>
467         <ds:KeyName>spa.example.com</ds:KeyName>
468       </ds:KeyInfo>
469     </SubjectConfirmationData>
470   </saml:SubjectConfirmation>
471 </saml:Subject>
472 <saml:Conditions
473   NotBefore="2003-04-17T00:46:02Z"
474   NotOnOrAfter="2003-04-17T01:46:02Z">
475   <saml:AudienceRestriction>
476     <saml:Audience>urn:mace:shibboleth:2.0:profiles:delegation</saml:Audience>
477   </saml:AudienceRestriction>
478   <saml:AudienceRestriction>
479     <saml:Audience>https://spa.example.com/shibboleth</saml:Audience>
480     <saml:Audience>https://spb.example.com/shibboleth</saml:Audience>
481   </saml:AudienceRestriction>
482 </saml:Conditions>
483 <saml:AuthnStatement AuthnInstant="2003-04-17T00:46:00Z" SessionIndex="1">
484   <saml:AuthnContext>
485     <saml:AuthnContextClassRef>
486       urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
487     </saml:AuthnContextClassRef>
488   </saml:AuthnContext>
489   <saml:SubjectLocality Address="127.0.0.1"/>
490 </saml:AuthnStatement>
491 </saml:Assertion>
492 </samlp:Response>

```

### 493 3.1.7 Metadata Considerations

494 As this profile concerns itself with extending the capability of an existing profile of the Authentication  
495 Request protocol, it does not by itself warrant explicit advertisement using metadata. Profiles that  
496 compose with it SHOULD ensure that support for this extension is addressed within their own use of  
497 metadata.

### 498 3.1.8 Security and Privacy Considerations

499 By itself, this profile only concerns itself with the issuance of a single assertion (or set of assertions  
500 sharing significant overlap) to be used by a single requester (for example an SP). Potentially the same  
501 assertion could be used by that SP to access multiple services on behalf of the principal. This raises  
502 privacy considerations that ordinarily are out of scope because the SAML SSO profiles concern  
503 themselves with only a single relying party.

504 For example, if multiple services receive the same assertion during a single session, they can potentially  
505 correlate the principal's activities among themselves by recognizing that the same assertion was used.  
506 This is harder to do if separate assertions are obtained to access each service independently. Of course,  
507 many other means of activity correlation exist, and this is merely one of them.

508 An additional problem is the flow of information about the principal inside the assertion through an active  
509 intermediary (the requesting SP) and on to multiple back-end services. This threat can be addressed

510 through the use of XML encryption over sensitive parts of the assertion. In fact, different parts of the  
511 assertion can be encrypted independently, or to multiple relying parties for decryption.

512 For example, the `<saml:NameID>` and `<saml:Attribute>` elements can be encrypted into  
513 `<saml:EncryptedNameID>` and `<saml:EncryptedAttribute>` elements that include symmetric  
514 keys encrypted to different relying parties. Only the relying parties that can decrypt the encryption key  
515 will be able to recover the original data. In this fashion, a set of SAML attributes for each SP can be  
516 included without necessarily exposing them to the others.

517 Alternatively, multiple assertions might be generated, signed, and then encrypted (using the  
518 `<saml:EncryptedAssertion>` element) separately for each back-end service. The `Recipient`  
519 attribute in the `<xenc:EncryptedKey>` element could be used to enable the intermediate SP to  
520 determine which assertion to supply to a given relying party (provided it had knowledge of which key it  
521 should use to confirm itself as a delegate).

522 Finally, note that identifying the set of relying parties to be included in an assertion provides the IdP with  
523 a significant amount of knowledge about the activities of the principal. This is already a consideration with  
524 the SAML SSO profiles in general, but as additional tiers of service access are added, even more  
525 information is potentially exposed. This problem cannot easily be mitigated while relying on the IdP to  
526 enforce constraints on the scope of delegation and deserves further study. More advanced cryptographic  
527 techniques may help address this problem.

## 528 **3.2 Browser/ECP Single Sign-On Profiles With Delegation**

529 By combining the original SAML 2.0 profiles for web SSO with the delegation profile defined in section  
530 3.1, the standard interaction between a principal's user agent, an IdP and an SP can be extended,  
531 without using additional SAML protocol interactions, to support the delegation scenario described in  
532 section 2.1.

### 533 **3.2.1 Required Information**

534 **Identification:** urn:mace:shibboleth:2.0:profiles:SSO:delegation

535 **Contact Information:** [shibboleth-dev@internet2.edu](mailto:shibboleth-dev@internet2.edu)

536 **Description:** Given below.

537 **Updates:** SAML 2.0 Browser SSO and ECP profiles

### 538 **3.2.2 Profile Description**

539 This profile is a small extension to the SAML 2.0 Web Browser SSO and ECP profiles [SAML2Prof] that  
540 combines them with the Authentication Request Delegation profile defined in section 3.1. It does not alter  
541 the message flows between the SP, IdP, and user agent.

542 However, an SP that supports this extension profile MAY utilize the extended `<samlp:AuthnRequest>`  
543 request semantics defined in section 3.1 in the request it creates and an IdP that supports this extension  
544 profile MAY apply those processing rules in the creation of its response.

545 This extension profile changes nothing pertaining to the single sign-on process. It merely provides an  
546 optional facility for extending the use of the resulting assertion(s) in more advanced scenarios involving  
547 additional communication between service providers.

### 548 3.2.3 Processing Rules

549 All existing message content and processing rules [SAML2Prof] also apply to this extension (specific  
550 rules depending on whether the Web Browser SSO or ECP profile is in use).

551 An SP that wishes to make use of this extended profile MAY include the extended content defined in  
552 section 3.1.3, noting that it MUST include the profile-specific `<saml:Audience>` value defined in that  
553 section in the `<saml:AuthnRequest>` message.

554 An IdP that supports this extended profile MAY follow the processing rules defined in section 3.1.5 and  
555 include extended assertion content as defined by section 3.1.4 and requested by the SP. Such an IdP  
556 MAY also include extended content in the assertions it issues in an unsolicited response, but it SHOULD  
557 do so only if it has reason to believe this is necessary and the SP supports this (for example, based on  
558 an examination of its metadata).

559 Note that an SP that does not support the extended delegation semantics will likely reject such an  
560 assertion on the basis of an unsatisfied `<saml:AudienceRestriction>` element, if it is following  
561 SAML processing rules.

### 562 3.2.4 Metadata Considerations

563 As this profile is an extension of existing profiles that utilize metadata, support for this extension is  
564 expressed in SAML 2.0 metadata by adding a boolean-typed XML attribute to the  
565 `<md:SingleSignOnService>` element(s) in an IdP's `<md:IDPSSODescriptor>` role element and  
566 `<md:AssertionConsumerService>` element(s) in an SP's `<md:SPSSODescriptor>` role element.

567 This XML attribute has the following qualified name:

568     Namespace:   urn:mace:shibboleth:2.0:profiles:SSO:delegation  
569     Local Name:   support

#### 570 3.2.4.1 Example

571 The example below shows a fragmentary `<md:SingleSignOnService>` element that advertises  
572 support for this profile. The namespace declaration must be in scope, but the prefix is of course arbitrary.

```
573 <md:SingleSignOnService  
574   xmlns:dlg="urn:mace:shibboleth:2.0:profiles:SSO:delegation"  
575   dlg:support="true" .../>
```

### 576 3.2.5 Security and Privacy Considerations

577 See section 3.1.8 for a discussion of the new considerations raised by the use of this extended profile  
578 during SSO.

## 579 3.3 SAML Token Service Profile

580 Existing profiles [SAML2Prof] of the SAML Authentication Request protocol [SAML2Core] address a flow  
581 in which the requester (an SP) generally relays its request through an active or passive intermediary, the  
582 presenter, who subsequently authenticates to the IdP in order to receive a bearer assertion used to  
583 authenticate itself to the requester.

584 This profile defines a stand-alone token service designed to enable a requester to exchange arbitrary  
585 authentication credentials for one or more SAML assertions that it can use in one or more subsequent  
586 interactions that are outside the scope of this profile. The resulting assertions can be tailored for their

587 intended use by the requester using the features of the `<samlp:AuthnRequest>` element  
588 [SAML2Core].

### 589 **3.3.1 Required Information**

590 **Identification:** urn:mace:shibboleth:2.0:profiles:SAMLTokenService

591 **Contact Information:** [shibboleth-dev@internet2.edu](mailto:shibboleth-dev@internet2.edu)

592 **Description:** Given below.

593 **Updates:** None.

### 594 **3.3.2 Profile Overview**

595 The Web Browser SSO and ECP profiles [SAML2Prof] define the rules by which an IdP can accept a  
596 `<samlp:AuthnRequest>` message from an SP presented by an intermediary and issue a bearer  
597 assertion to the intermediary inside a `<samlp:Response>` message tailored for immediate delivery  
598 back to the requester. The client is merely a transmission "hop" for the message and is not an active  
599 participant in the SAML protocol exchange (although of course its authentication to the IdP is a critical  
600 piece of the overall profile).

601 This profile defines a stand-alone two-party exchange that is independent of a particular context of use  
602 for the resulting assertions and is more suited to use by client software or devices capable of richer  
603 behavior and configuration, or by system entities that are themselves in need of authentication  
604 credentials in the SAML format. In many cases, the requester may be trusted with additional freedom in  
605 the kinds of assertion content it can ask for because it is (as proven by an authentication process)  
606 representing the wishes of the principal that will be the subject of those assertions.

607 This profile of the Authentication Request protocol REQUIRES that the `<samlp:AuthnRequest>`  
608 requester and presenter are the same entity. In other words, the entity that is asking for the assertion is  
609 the same as the entity that is expected to authenticate to the IdP in order to obtain it. Likewise, the  
610 `<samlp:Response>` is returned directly to that entity, rather than packaged for delivery to an SP. The  
611 SAML SOAP binding [SAML2Bind] would typically be used to exchange the SAML protocol messages.

612 A requester that supports this profile MAY utilize the extended `<samlp:AuthnRequest>` request  
613 semantics defined in section 3.1 in the request it creates and an IdP that supports this extension profile  
614 MAY apply those processing rules in the creation of its response.

615 Finally, while authentication of the requester is mostly out of scope, one specific case of authentication  
616 by means of SAML itself is defined to enable an entity wielding a SAML assertion to use it to obtain an  
617 additional assertion with different properties.

618 Figure @@ shows the simple exchange that makes up this profile.

619 TBD

#### 620 **1. Requester Determines Identity Provider**

621 In step 1, the requester determines the appropriate identity provider to contact.

#### 622 **2. `<samlp:AuthnRequest>` issued by Requester to Identity Provider**

623 In step 2, the requester obtains the location of an endpoint at the selected identity provider and sends a  
624 `<samlp:AuthnRequest>` message to that endpoint using a compatible SAML binding.

#### 625 **3. Identity Provider identifies Requester**

626 In step 3, the requester is identified by the identity provider by some means outside the scope of this  
627 profile. This may require a new act of authentication, or it may reuse an existing authenticated session.

#### 628 **4. Identity Provider issues and returns <samlp:Response> to Requester**

629 In step 4, the identity provider evaluates the request and returns a <samlp:Response> message to the  
630 requester. The message may indicate an error or will include one or more signed assertions that satisfy  
631 the request, including at least one authentication assertion.

### 632 **3.3.3 Profile Description**

#### 633 **3.3.3.1 Requester Determines Identity Provider**

634 This step is implementation-dependent and may involve manual configuration of the requester, user  
635 input, or any other means.

#### 636 **3.3.3.2 <samlp:AuthnRequest> issued by Requester to Identity Provider**

637 Once an identity provider is selected, the location of an endpoint is determined based on the SAML  
638 binding chosen by the requester for sending the <samlp:AuthnRequest> message. Metadata  
639 [SAML2Meta] MAY be used for this purpose.

640 The exact formatting of the message is defined by the SAML binding used. Profile-specific rules for the  
641 contents of the <samlp:AuthnRequest> message are included in section 3.3.4.1. The message MAY  
642 be signed as a means of authenticating the request.

643 The identity provider MUST process the <samlp:AuthnRequest> message as described in  
644 [SAML2Core]. This may constrain the subsequent interactions with the requester, for example if the  
645 `IsPassive` attribute is included.

#### 646 **3.3.3.3 Identity Provider identifies Requester**

647 At any time during the previous step or subsequent to it, the identity provider MUST establish the identity  
648 of the requester (unless it returns an error to the requester). The identity provider may use any means to  
649 authenticate the requester, subject to any requirements included in the <samlp:AuthnRequest> in the  
650 form of the <samlp:RequestedAuthnContext> element.

651 Section 3.5 defines a SAML-based authentication profile that MAY be used for this purpose, allowing one  
652 assertion to be used as the basis for obtaining another.

#### 653 **3.3.3.4 Identity Provider issues and returns <samlp:Response> to Requester**

654 Regardless of the success or failure of processing the <samlp:AuthnRequest>, the identity provider  
655 SHOULD respond to the requester with a <samlp:Response> message.

656 The exact formatting of the message is defined by the SAML binding used. Profile-specific rules for the  
657 contents of the <samlp:Response> message are included in section 3.3.4.2. The message MAY be  
658 signed as a means of authenticating the response.

659 The requester MUST process the <samlp:Response> message as described in [SAML2Core].

## 660 3.3.4 Use of Authentication Request Protocol

661 This profile is based on the Authentication Request protocol defined in [SAML2Core]. In the  
662 nomenclature of actors enumerated in section 3.4 of that document, the requester is (obviously) the  
663 requester and also the presenter. By default, it is also the requested subject, and the attesting entity. The  
664 request can include content that alters or extends this behavior.

### 665 3.3.4.1 <samlp:AuthnRequest> Usage

666 A requester MAY include any message content described in [SAML2Core], section 3.4.1. All processing  
667 rules are as defined in [SAML2Core]. The <saml:Issuer> element MUST be present and MUST  
668 contain the unique identifier of the requester.

669 If the identity provider cannot or will not satisfy the request, it MUST respond with a  
670 <samlp:Response> message containing an appropriate error status code or codes.

671 Note that the requester MAY include a <saml:Subject> element in the request that names the actual  
672 identity about which it wishes to receive an assertion. This identity MUST be that of the authenticated  
673 requester, unless the authentication profile described in section 3.5 is used. In that case, the requester  
674 can attach an existing <saml:Assertion> as a means of authentication to receive additional  
675 assertions about the principal identified in the subject of the attached assertion.

676 The <samlp:AuthnRequest> message MAY be signed, but if not then it MUST be integrity protected  
677 by some other binding-specific means.

678 The requester MAY utilize the extended delegation request profile defined in section 3.1.

### 679 3.3.4.2 <samlp:Response> Usage

680 If the identity provider wishes to return an error, it MUST NOT include any assertions in the  
681 <samlp:Response> message. Otherwise, if the request is successful the <samlp:Response> element  
682 MUST conform to the following:

- 683 • If the <samlp:Response> message is signed or if an enclosed assertion is encrypted, then the  
684 <saml:Issuer> element MUST be present. Otherwise it MAY be omitted. If present it MUST  
685 contain the unique identifier of the issuing identity provider; the `Format` attribute MUST be  
686 omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`
- 687 • It MUST contain at least one assertion. Each assertion's <saml:Issuer> element MUST  
688 contain the unique identifier of the issuing identity provider; the `Format` attribute MUST be  
689 omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`
- 690 • The set of one or more assertions MUST contain at least one <saml:AuthnStatement> that  
691 reflects the authentication of the subject to the identity provider.
- 692 • Other statements MAY be included in the assertion(s) at the discretion of the identity provider. In  
693 particular, <saml:AttributeStatement> elements MAY be included.
- 694 • The subject, confirmation rules, and conditions present in the resulting assertions are dependent  
695 on the content of the request, the discretion of the identity provider, the generic processing rules  
696 for the <samlp:AuthnRequest> message [SAML2Core], and the processing rules in section  
697 3.3.4.3.

698 In all cases, the identity provider MUST ensure that the content of all included assertions are supported  
699 by the authenticated identity of the requester, the contents of the request, and any applicable policies  
700 established by it and the subject.

701 The included assertions MAY utilize the extended delegation request profile defined in section 3.1.

### 702 3.3.4.3 Processing Rules

703 Regardless of the SAML binding used, the identity of the requester MUST be established, generally in a  
704 binding-specific manner. A signature in the `<samlp:AuthnRequest>` element is NOT sufficient to  
705 establish the requester's identity unless the requester can prove possession of the corresponding key.

706 In addition to the rules specified by [SAML2Core], the following rules apply to the processing of requests:

- 707 • If a `<saml:Subject>` element is not present in the request, or contains only  
708 `<saml:SubjectConfirmation>` elements, then the subject of the resulting assertions MUST  
709 **strongly match** the `<saml:Issuer>` element in the request, except that the identifier MAY be  
710 in a different format if specified by a `<samlp:NameIDPolicy>` element. In such a case, the  
711 identifier's physical content MAY be different, but it MUST refer to the same principal.
- 712 • If a `<saml:Subject>` element identifying the intended subject is present in the request, but  
713 does not match the authenticated identity of the requester, then an error MUST be returned  
714 unless the authentication profile described in section 3.5 is used, and supports the alternative  
715 subject requested.
- 716 • If no `<saml:SubjectConfirmation>` elements are included in the request, then the identity  
717 provider MUST have knowledge of one or more appropriate confirmation keys associated with  
718 the requester (perhaps via metadata or the authentication process). If not, an error MUST be  
719 returned. Otherwise, the identity provider MUST include a `<saml:SubjectConfirmation>`  
720 element in the resulting assertions with a `Method` attribute equal to  
721 `urn:oasis:names:tc:SAML:2.0:cm:holder-of-key`, and containing a  
722 `<saml:SubjectConfirmationData>` element identifying the relevant confirmation key(s).

723 In general, broad discretion MAY be granted to the requester in establishing the  
724 `<saml:SubjectConfirmation>` and `<saml:Conditions>` elements to place in the resulting  
725 assertion. The basis of this is the authentication performed to ensure that the entity requesting the  
726 assertions is authorized to receive them. The method of authentication MAY be used to determine what  
727 kinds of assertion content can be included.

728 In particular, the `<saml:SubjectConfirmation>` elements to include SHOULD be carefully  
729 scrutinized by the identity provider, as they determine how the assertions can be subsequently applied.

730 There are no specific processing rules related to the consumption of the response apart from general  
731 rules applying to all SAML protocol messages. The assertions are generally intended for consumption by  
732 other relying parties, and might not be processed by the requester directly.

### 733 3.3.5 Metadata Considerations

734 As this profile is an alternate version of the service offered by endpoints represented by the  
735 `<md:SingleSignOnService>` element, support for this extension is expressed in SAML 2.0 metadata  
736 [SAML2Meta] by adding a boolean-typed XML attribute to the `<md:SingleSignOnService>`  
737 element(s) in an IdP's `<md:IDPSSODescriptor>` role element.

738 This XML attribute has the following qualified name:

739     Namespace:    `urn:mace:shibboleth:2.0:profiles:SAMLTokenService`  
740     Local Name:   `support`

### 741 3.3.5.1 Example

742 The example below shows a fragmentary `<md:SingleSignOnService>` element that advertises  
743 support for this profile. The namespace declaration must be in scope, but the prefix is of course arbitrary.

```
744 <md:SingleSignOnService  
745   xmlns:sts="urn:mace:shibboleth:2.0:profiles:SAMLTokenService"  
746   sts:support="true" .../>
```

### 747 3.3.6 Security and Privacy Considerations

748 TBD

749 As this is a very generic service, a lot of potential security risks exist based on the kinds of assertions  
750 issued.

## 751 3.4 SOAP Application Profile

752 In general, the use of SAML assertions to secure communications between service providers is  
753 dependent on the application protocol and does not in and of itself presume web services. However, in  
754 the case of SOAP, the Web Services Security SOAP Message Security [WSS-SMS] and SAML Token  
755 Profile [WSS-SAML11] specifications define an industry standard means of packaging SAML assertions  
756 together with application data using the `<wsse:Security>` SOAP header block.

757 This profile defines a minimal set of rules for attaching assertions to reduce the complexity that  
758 implementers must support. If additional options are needed, more full-featured security specifications  
759 supporting SAML can be used.

### 760 3.4.1 Required Information

761 **Identification:** urn:mace:shibboleth:2.0:profiles:SAML-authentication:SOAP

762 **Contact Information:** [shibboleth-dev@internet2.edu](mailto:shibboleth-dev@internet2.edu)

763 **Description:** Given below.

764 **Updates:** None.

### 765 3.4.2 Profile Description

766 The core Web Services Security: SOAP Message Security (WSS-SMS) specification [WSS-SMS]  
767 describes a very broad set of syntax components and basic processing rules for signing and encrypting  
768 SOAP messages and for attaching security tokens, such as SAML assertions, to them. The SAML Token  
769 Profile [WSS-SAML11] further defines the specific rules for attaching SAML assertions that use particular  
770 confirmation methods and processing them at relying parties.

771 By themselves, these specifications are necessary but not sufficient to achieve interoperable security for  
772 SOAP-based applications. In particular, there are many ways of accomplishing the same basic goal  
773 using different syntax, and there is no discussion whatsoever of the meaning of the information itself in  
774 the context of an application.

775 To provide a basic level of interoperability such that implementations of security can be effectively  
776 separated from application code, this profile defines a very limited set of constraints on the use of WSS-  
777 SMS and the SAML Token profile with the assertions typically available to service providers participating  
778 in the SAML SSO profiles and the additional profiles defined in this document.

779 Specifically, this profile describes a means by which one or more SAML assertions containing  
780 authentication and attribute information can be attached to a SOAP message, signed at either the  
781 message or transport layer, and then consumed for the purposes of authenticating the SOAP message  
782 and attaching a set of contextual information about the requester.

### 783 **3.4.2.1 <wsse:Security> Header Content**

784 This profile concerns itself exclusively with defining the content of a <wsse:Security> header created  
785 by the originator of the SOAP request to authenticate the message. The header **MUST** contain the  
786 following elements:

787 <wsu:Timestamp> (Optional)

788       Optionally allows a creation and/or expiration timestamp to be attached to the message.

789 <saml:Assertion> or <wsse:SecurityTokenReference> (One or More)

790       One or more SAML assertions (by value or remote URI reference) whose subject refers to the  
791 identity that the message recipient **SHOULD** associate with contents of the message.

792 <ds:Signature> (Optional)

793       A signature created by the originator of the message to ensure the integrity of the message and  
794 to authenticate itself as the holder of the key contained in the SAML assertion(s). Alternatively,  
795 the transport layer **MAY** be used to provide these capabilities.

796 If a <ds:Signature> element is included, then its <ds:SignedInfo> element **MUST** contain  
797 <ds:Reference> elements for the following content:

- 798       • The SOAP message body
- 799       • Any additional SOAP headers that need to be protected
- 800       • The <wsu:Timestamp> element, if one is included
- 801       • Each SAML assertion (or security token reference) associated with the message

### 802 **3.4.2.2 Originator Processing Rules**

803 The originator of a SOAP message to be secured with this profile:

- 804       • **MUST** create a <wsse:Security> header targeted at the intended recipient if one does not  
805 already exist in the message.
- 806       • **MUST** prepend the relevant <saml:Assertion> or <wsse:SecurityTokenReference>  
807 elements to be attached to the message to the header's content.
- 808       • **MAY** prepend a <wsu:Timestamp> element to the header's content.

809 If the confirmation method of the assertion(s) to be satisfied is

810       urn:oasis:names:tc:SAML:2.0:cm:holder-of-key

811 then the originator **MAY** prepend a <ds:Signature> element to the header's content, but it **MUST**  
812 appear after any other content inserted by the originator in accordance with WSS-SMS requirements for  
813 signature processing [WSS-SMS]. The signature's content **MUST** follow the rules outlined in the previous  
814 section.

815 Additionally, the originator **MUST** include a <ds:KeyInfo> containing a  
816 <wsse:SecurityTokenReference> element that refers to the assertion containing the confirmation

817 key that will be used to create the signature [WSS-SAML11]. A direct reference (using the  
818 <wsse:Reference> element) SHOULD be used, but if the assertion is included in the header, then a  
819 key identifier reference (using the <wsse:KeyIdentifier> element) MAY be used.

820 If a <ds:Signature> element is not included, then the transport layer MUST supply the key to  
821 establish the confirmation. TLS client authentication MAY be used for this purpose.

### 822 3.4.2.3 Recipient Processing Rules

823 Any <wsse:Security> headers MUST be processed in accordance with [WSS-SMS] and [WSS-  
824 SAML11]. Any SAML assertions included with the message MUST also be processed in accordance with  
825 [SAML2Core] and any additional SAML profiles governing their use.

826 Valid assertions associated with the message for which the message originator can successfully satisfy  
827 an included <saml:SubjectConfirmation> element MAY be associated with the message by the  
828 recipient and the the subject of those assertions MAY be considered a party to the origination of the  
829 message.

### 830 3.4.3 Example

831 The following example shows a SOAP message secured using this profile, containing a SAML assertion  
832 supplying information about the originator of the message. The actual content of the assertion is largely  
833 omitted.

```
834 <S:Envelope>  
835   <S:Header>  
836     <wsse:Security>  
837       <wsu:Timestamp wsu:Id="timestamp">  
838         <wsu:Created>2005-09-13T08:42:00Z</wsu:Created>  
839       </wsu:Timestamp>  
840       <saml:Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc" ...>  
841         <saml:Issuer>https://idp.example.com/shibboleth</saml:Issuer>  
842         <ds:Signature>...</ds:Signature>  
843         <saml:Subject>  
844           <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">  
845             3f7b3dcf-1674-4ecd-92c8-1544f346baf8  
846           </saml:NameID>  
847           <saml:SubjectConfirmation  
848             Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">  
849             <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">  
850               https://spa.example.com/shibboleth  
851             </saml:NameID>  
852             <SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">  
853               <ds:KeyInfo>  
854                 <ds:KeyName>spa.example.com</ds:KeyName>  
855               </ds:KeyInfo>  
856             </SubjectConfirmationData>  
857           </saml:SubjectConfirmation>  
858         </saml:Subject>  
859         <saml:AuthnStatement>...</saml:AuthnStatement>  
860         <saml:AttributeStatement>...</saml:AttributeStatement>  
861       </saml:Assertion>  
862     </ds:Signature>  
863     <ds:SignedInfo>  
864       <ds:CanonicalizationMethod  
865         Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>  
866       <ds:SignatureMethod  
867         Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>  
868       <ds:Reference URI="#MsgBody">  
869         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
870         <ds:DigestValue>...</ds:DigestValue>
```

```

871     </ds:Reference>
872     <ds:Reference URI="#timestamp">
873       <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
874       <ds:DigestValue>...</ds:DigestValue>
875     </ds:Reference>
876     <ds:Reference URI="#_a75adf55-01d7-40cc-929f-dbd8372ebdfc">
877       <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
878       <ds:DigestValue>...</ds:DigestValue>
879     </ds:Reference>
880   </ds:SignedInfo>
881   <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
882   <ds:KeyInfo>
883     <wsse:SecurityTokenReference
884 wsse1:TokenType="http://docs.oasis-open.org/wss/oasis-wss-samltoken-profile-
885 1.1#SAMLV2.0">
886       <wsse:Reference URI="#_a75adf55-01d7-40cc-929f-dbd8372ebdfc"/>
887     </wsse:SecurityTokenReference>
888   </ds:KeyInfo>
889 </ds:Signature>
890 </wsse:Security>
891 </S:Header>
892 <S:Body wsu:Id="MsgBody">
893   <ReportRequest>
894     <TickerSymbol>SUNW</TickerSymbol>
895   </ReportRequest>
896 </S:Body>
897 </S:Envelope>

```

### 898 3.4.4 Metadata Considerations

899 In general, this profile does not constrain the mechanisms by which specific keys are named, located, or  
900 verified by the parties. Confirmation and signing keys can be identified in a variety of ways; SAML 2.0  
901 metadata [SAML2Meta] is a possible source of information about the binding between SAML entity  
902 identifiers and public keys used by those entities.

### 903 3.4.5 Security and Privacy Considerations

904 TBD

## 905 3.5 SAML Authentication to SAML Token Service

906 Section 3.3 defines a SAML Token Service profile that supports the SAML SOAP binding [SAML2Bind].  
907 Section 3.4 defines a profile by which SAML assertions can be used to authenticate a SOAP application  
908 protocol. Combining these profiles results in a mechanism by which a SAML assertion can be used to  
909 authenticate a request for another SAML assertion.

910 Furthermore, this mechanism can be combined with the delegation profile described in section 3.1 such  
911 that an entity can use an assertion issued to a principal authorized for its use as a delegate to obtain an  
912 additional assertion by authenticating to the SAML Token Service on behalf of the principal. This enables  
913 chains of delegation to be supported across time by issuing new assertions on the basis of older ones.

### 914 3.5.1 Required Information

915 **Identification:** urn:mace:shibboleth:2.0:profiles:SAML-authentication:SAMLTokenService

916 **Contact Information:** [shibboleth-dev@internet2.edu](mailto:shibboleth-dev@internet2.edu)

917 **Description:** Given below.

918 **Updates:** None.

## 919 **3.5.2 Profile Description**

920 The SAML Token Service profile defined in section 3.3 does not define any specific mechanisms by  
921 which the requester can authenticate to the SAML authority, allowing many different technologies to be  
922 supported. The use case of an entity in possession of a SAML assertion acquiring additional assertions  
923 can be supported by defining an interoperable authentication mechanism that can be applied to the  
924 SAML Token Service exchange when the SAML SOAP binding is used.

925 The technical aspects of this profile are defined by a combination of sections 3.3 and 3.4 and their  
926 references. Specifically, when the SAML SOAP binding [SAML2Bind] is used to carry the SAML protocol  
927 exchange, the SOAP Application profile in section 3.4 defines the syntax and processing rules for  
928 attaching a SAML assertion to the message as a means of authentication.

929 Because the token service is security-sensitive, assertions used for authentication to it SHOULD contain  
930 a strong subject confirmation method, such as the Holder of Key mechanism that requires a proof of key  
931 possession using a digital signature or TLS client authentication.

## 932 **3.5.3 Processing Rules**

933 The token service requester MUST adhere to the profile defined section 3.3 to attach one or more  
934 assertions to the request. The `<wsse:Security>` SOAP header MUST be targeted at the ultimate  
935 SOAP recipient. The header MUST include a `<wsu:Timestamp>` element that indicates the creation  
936 time of the message.

937 The identity provider MUST verify the contents of the `<wsse:Security>` SOAP header in accordance  
938 with section 3.3. Having done so, the identity provider MAY consider the requester to be the subject of  
939 the attached assertion(s), or at least acting on that principal's behalf.

940 Note that the assertion(s) attached to the request MAY be issued by the responding identity provider or  
941 MAY be issued by a third party that is acceptable to the responder.

## 942 **3.5.4 Metadata Considerations**

943 In general, this profile does not constrain the mechanisms by which specific keys are named, located, or  
944 verified by the parties. Confirmation and signing keys can be identified in a variety of ways; SAML 2.0  
945 metadata [SAML2Meta] is a possible source of information about the binding between SAML entity  
946 identifiers and public keys used by those entities.

## 947 **3.5.5 Security and Privacy Considerations**

948 TBD

## 949 4 References

950 The following works are referenced directly or indirectly in the body of this specification.

### 951 4.1 Normative References

- 952       **[RFC 2119]**       S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF  
953                               RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 954       **[RFC 2246]**       T. Dierks, C. Allen. *The TLS Protocol Version 1.0*. IETF RFC 2246, January  
955                               1999. <http://www.ietf.org/rfc/rfc2246.txt>.
- 956       **[RFC 2396]**       T. Berners-Lee et al. *Uniform Resource Identifiers (URI): Generic Syntax*. IETF  
957                               RFC 2396, August, 1998. <http://www.ietf.org/rfc/rfc2396.txt>.
- 958       **[SAML2Core]**       S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion  
959                               Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-  
960                               core-2.0-os. <http://www.oasis-open.org/committees/security/>.
- 961       **[SAML2Bind]**       S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language  
962                               (SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-bindings-2.0-os.  
963                               <http://www.oasis-open.org/committees/security/>.
- 964       **[SAML2Prof]**       S. Cantor et al. *Profiles for the OASIS Security Assertion Markup Language  
965                               (SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-profiles-2.0-os.  
966                               <http://www.oasis-open.org/committees/security/>.
- 967       **[SAML2Secure]**    F. Hirsch et al. *Security and Privacy Considerations for the OASIS Security  
968                               Assertion Markup Language (SAML) V2.0*. OASIS SSTC, March 2005.  
969                               Document ID saml-sec-consider-2.0-os. [open.org/committees/security/](http://www.oasis-<br/>970                               open.org/committees/security/).
- 971       **[SAML2Meta]**       S. Cantor et al., *Metadata for the OASIS Security Assertion Markup Language  
972                               (SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-metadata-2.0-os.  
973                               See <http://www.oasis-open.org/committees/security/>.
- 974       **[SAML2-XSD]**       S. Cantor et al. *SAML assertion schema*. OASIS SSTC, March 2005. Document  
975                               ID saml-schema-assertion-2.0.xsd. [open.org/committees/security/](http://www.oasis-<br/>976                               open.org/committees/security/).
- 977       **[SAML2P-XSD]**     S. Cantor et al. *SAML protocol schema*. OASIS SSTC, March 2005. Document  
978                               ID saml-schema-protocol-2.0.xsd. [open.org/committees/security/](http://www.oasis-<br/>979                               open.org/committees/security/).
- 980       **[SAML2Meta-xsd]** S. Cantor et al., *SAML metadata schema*. OASIS SSTC, March 2005. Document  
981                               ID saml-schema-metadata-2.0.xsd. See [open.org/committees/security/](http://www.oasis-<br/>982                               open.org/committees/security/).
- 983       **[WSS-SMS]**       Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds.  
984                               *Web Services Security: SOAP Message Security V1.0*. OASIS WSSTC, January  
985                               2004. [http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-  
986                               security-1.0.pdf](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-<br/>986                               security-1.0.pdf)
- 987       **[WSS-SAML11]**    Monzillo, Ronald, et al. *Web Services Security: SAML Token Profile 1.1*, OASIS  
988                               *Public Review Draft 01*. OASIS WSSTC, June 2005. [http://www.oasis-  
989                               open.org/committees/download.php/13405](http://www.oasis-<br/>989                               open.org/committees/download.php/13405)
- 990       **[Schema2]**       P. V. Biron et al. *XML Schema Part 2: Datatypes*. World Wide Web Consortium  
991                               Recommendation, May 2001. <http://www.w3.org/TR/xmlschema-2/>.

992 **4.2 Non-Normative References**

993 **[SAML2Gloss]** J. Hodges et al., *Glossary for the OASIS Security Assertion Markup Language*  
994 *(SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-glossary-2.0-os.  
995 See <http://www.oasis-open.org/committees/security/>.