

---

# 1 Shibboleth Architecture

## 2 Protocols and Profiles

### 3 Working Draft 05, 23 November 2004

#### 4 Document identifier:

5 draft-mace-shibboleth-arch-protocols-05

#### 6 Location:

7 <http://shibboleth.internet2.edu/>

#### 8 Editors:

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

#### 10 Contributors:

11 Steven Carmody, Brown University  
12 Marlena Erdos, Tivoli Systems, Inc.  
13 Keith Hazelton, University of Wisconsin  
14 Walter Hoehn, University of Memphis  
15 RL "Bob" Morgan, University of Washington  
16 Tom Scavo, Individual  
17 David Wasley, University of California

#### 18 Abstract:

19 This specification defines the general architecture, protocols, and message formats that make up  
20 the Shibboleth web single sign-on and attribute exchange mechanism, which is built on the OASIS  
21 SAML 1.1 specification (<http://www.oasis-open.org/committees/security>). Readers should be  
22 familiar with that specification before reading this document.

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

# Table of Contents

26	1 Introduction.....	3
27	1.1 Notation.....	3
28	2 Architectural Overview.....	4
29	2.1 Single Sign-On Overview.....	4
30	2.2 Identity Provider.....	5
31	2.2.1 Authentication Authority.....	6
32	2.2.2 Attribute Authority.....	6
33	2.2.3 Single Sign-On Service.....	6
34	2.2.4 Inter-Site Transfer Service.....	7
35	2.2.5 Artifact Resolution Service.....	7
36	2.3 Service Provider.....	7
37	2.3.1 Assertion Consumer Service.....	7
38	2.3.2 Attribute Requester.....	8
39	2.4 WAYF.....	8
40	3 Protocols and Profiles.....	9
41	3.1 Authentication Request and Response Profiles.....	9
42	3.1.1 Authentication Request Profile.....	9
43	3.1.1.1 Required Information.....	9
44	3.1.1.2 Message Format and Transmission.....	9
45	3.1.1.3 Processing Rules.....	10
46	3.1.1.4 Example.....	10
47	3.1.2 Browser/POST Authentication Response Profile.....	11
48	3.1.2.1 Example.....	11
49	3.1.3 Browser/Artifact Authentication Response Profile.....	12
50	3.1.3.1 Example.....	13
51	3.2 Attribute Request, Response, and Syntax Profile.....	13
52	3.2.1 Required Information.....	13
53	3.2.2 Attribute Requests.....	13
54	3.2.2.1 Example.....	13
55	3.2.3 Attribute Responses.....	14
56	3.2.3.1 Example.....	14
57	3.2.4 Attribute Naming and Syntax.....	15
58	3.3 Transient NameIdentifier Format.....	15
59	3.4 Metadata Profile.....	16
60	3.4.1 Element <md:EntitiesDescriptor>.....	16
61	3.4.2 Element <md:EntityDescriptor>.....	16
62	3.4.3 Element <md:IDPSSODescriptor>.....	16
63	3.4.4 Element <md:AuthnAuthorityDescriptor>.....	17
64	3.4.5 Element <md:AttributeAuthorityDescriptor>.....	17
65	3.4.6 Element <md:SPSSODescriptor>.....	17
66	4 Security and Privacy Considerations.....	18
67	4.1 Additional Browser Profile Considerations.....	18
68	4.1.1 Information Leakage and Impersonation.....	18
69	4.1.2 Time Synchronization.....	18
70	5 References.....	19
71	5.1 Normative References.....	19
72	5.2 Non-Normative References.....	20
73		

---

# 74 1 Introduction

75 This specification defines a set of related profiles of SAML 1.1 and additional messages and protocols that  
76 make up the Shibboleth architecture. It is functionally a superset of the SAML 1.1 web browser single  
77 sign-on and attribute exchange mechanisms that incorporates additional profiles for user privacy and  
78 service-provider-first access.

79 Unless specifically noted, nothing in this document should be taken to conflict with the SAML 1.1  
80 specification, or any bindings and profiles referenced within it. Readers are advised to familiarize  
81 themselves with that specification first.

## 82 1.1 Notation

83 This specification uses normative text to describe the use of SAML 1.1 and additional SAML profiles.

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

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

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

92       Listings of XML schemas appear like this.

93       Example code listings appear like this.

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

- 98 • The prefix `saml:` stands for the SAML 1.1 assertion namespace,  
99     `urn:oasis:names:tc:SAML:1.0:assertion`
- 100 • The prefix `samlp:` stands for the SAML 1.1 request-response protocol namespace,  
101     `urn:oasis:names:tc:SAML:1.0:protocol`
- 102 • The prefix `md:` stands for the SAML 2.0 metadata namespace,  
103     `urn:oasis:names:tc:SAML:2.0:metadata`
- 104 • The prefix `ds:` stands for the W3C XML Signature namespace,  
105     <http://www.w3.org/2000/09/xmldsig#>
- 106 • The prefix `xsd:` stands for the W3C XML Schema namespace,  
107     <http://www.w3.org/2001/XMLSchema>  
108     in example listings. In schema listings, this is the default namespace and no prefix is shown.

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

## 2 Architectural Overview

111

112 Broadly speaking, the Shibboleth architecture defines a set of interactions between an *identity provider*  
113 and a *service provider* to facilitate web browser single sign-on and attribute exchange.

114 Previous versions of this specification and the SAML 1.1 specification variously refer to these roles of  
115 identity provider and service provider as "source site" or "origin" and "destination site" or "target". This  
116 specification adopts terminology used within the Liberty ID-FF specification [LibertyProt] and the draft  
117 SAML 2.0 specification [SAML2Gloss].

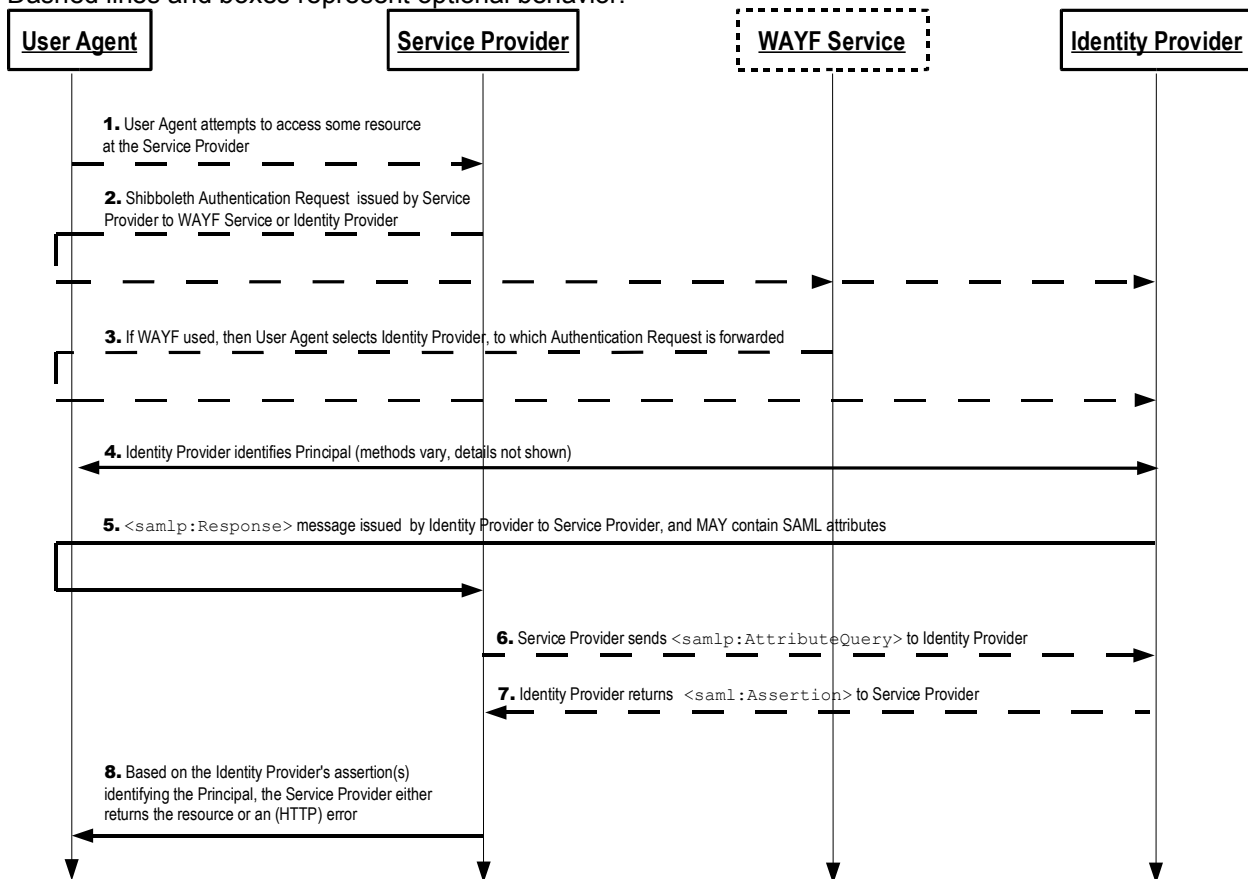
118 An additional, optional component called a *WAYF service* acts independently as a possible means of  
119 identity provider discovery. The role of the WAYF can be, and often is, taken on by a service provider  
120 itself.

### 2.1 Single Sign-On Overview

121

122 The following sequence diagram illustrates the set of required and optional interactions when using the  
123 Browser/POST profile. The Browser/Artifact profile replaces step 5 below with an artifact issued to the  
124 service provider followed by a SAML request/response exchange between the service provider and  
125 identity provider. See [SAMLBind] for detailed descriptions of both profiles.

126 Dashed lines and boxes represent optional behavior.



#### 1. HTTP Request to Service Provider

127

128 In step 1, the principal, via an HTTP User Agent, makes an HTTP request for a secured resource  
129 at the service provider without a security context.

## 130 2. Authentication Request issued by Service Provider to WAYF or Identity Provider

131 In step 2, the service provider redirects the user agent with an Authentication Request to either a  
132 WAYF or directly to an identity provider. A WAYF is typically used if the service provider wants to  
133 delegate the job of identity provider discovery and is working with a sufficiently constrained set of  
134 identity providers.

## 135 3. WAYF forwards Authentication Request to selected Identity Provider

136 If a WAYF is used in step 2, then it interacts via unspecified means with the user agent to select  
137 an identity provider to which to redirect the user agent with the service provider's Authentication  
138 Request.

## 139 4. Identity Provider identifies Principal

140 In step 4, the principal is identified by the identity provider by some means outside the scope of  
141 this specification. This may require a new act of authentication, or it may reuse an existing  
142 authenticated session.

## 143 5. Identity Provider issues <samlp:Response> or SAML Artifact(s) to Service Provider

144 In step 5, the identity provider issues a <samlp:Response> message or one or more SAML  
145 artifacts to be delivered by the user agent to the service provider. Either the SAML 1.1  
146 Browser/POST profile or Browser/Artifact profile can be used. If the Browser/POST profile is used,  
147 then either one or more assertions or an error response is passed directly through the user agent  
148 to the service provider. If the Browser/Artifact profile is used, then one or more SAML artifacts are  
149 passed through the user agent to the service provider, at which point the service provider  
150 communicates directly with the identity provider to resolve the artifact(s) into assertions.

## 151 6. Service Provider sends <samlp:AttributeQuery> to Identity Provider

152 In step 6, the service provider can optionally use the subject of the authentication assertion it  
153 received in step 5 to send a <samlp:AttributeQuery> (inside a <samlp:Request>  
154 message) to an attribute authority associated with the identity provider.

## 155 7. Identity Provider returns <saml:Assertion> to Service Provider

156 In step 6, the attribute authority associated with the identity provider processes the  
157 <samlp:AttributeQuery> and returns a <samlp:Response> message, possibly containing  
158 one or more assertions containing attributes that apply to the principal.

## 159 8. Service Provider grants or denies access to Principal

160 In step 8, the service provider can respond to the principal's user agent with its own error, or can  
161 establish its own security context for the principal and return the requested resource.

162 Note that an identity provider can initiate this sequence at step 5 and issue a <samlp:Response>  
163 message or SAML artifact(s) to a service provider without the preceding steps.

## 164 2.2 Identity Provider

165 An identity provider is an entity that authenticates principals and produces assertions of authentication and  
166 attribute information in accordance with [SAMLCore] and the SAML Browser/POST or Browser/Artifact  
167 profiles in [SAMLBind]. It consists of functional components drawn from the SAML domain model, an  
168 *authentication authority* and an *attribute authority*, along with an *inter-site transfer service*, defined by the  
169 Browser profiles, and a *single sign-on service*, defined by this specification. Note that physically, the single  
170 sign-on service and inter-site transfer service MAY be the same location.

171 Each identity provider MUST be assigned a unique identifier, or *provider ID*. The identifier MUST be a URI  
172 [RFC 2396] of no more than 1024 characters. Use of an "https" URL for this purpose may be  
173 advantageous for metadata publication (see section ).

## 174 **2.2.1 Authentication Authority**

175 The authentication authority is a SAML-defined service that issues authentication assertions about  
176 principals to relying parties (service providers, in the case of Shibboleth). Shibboleth does not specify how  
177 authentication of principals should be performed; the authority works with the principal's authentication  
178 service so that assertions about the authentication event are issued.

179 The only specifically defined use of an authentication assertion in Shibboleth is in accordance with the  
180 Browser/POST and Browser/Artifact profiles. As a result, the authentication authority is NOT REQUIRED  
181 to process SAML `<samlp:Request>` messages containing `<samlp:AuthenticationQuery>` or  
182 `<saml:AssertionIDReference>` elements, but MAY choose to do so. Also note that the  
183 Browser/POST and Browser/Artifact profiles do not specifically require the authentication authority to  
184 remember the assertions that it issues over an extended period of time, though this is also permitted.

## 185 **2.2.2 Attribute Authority**

186 The attribute authority is a SAML-defined service that supports a SAML protocol binding and the  
187 processing of SAML `<samlp:Request>` messages containing the `<samlp:AttributeQuery>`  
188 element. This service issues attribute assertions to service providers in a mutually authenticated fashion.  
189 Implementations typically rely on SSL/TLS [RFC 2246] or SAML message signatures to mutually  
190 authenticate the exchange.

191 Shibboleth additionally requires that control of attribute release to service providers be available to both  
192 administrators and principals. Therefore, a Shibboleth attribute authority MUST have the ability to  
193 authenticate requests and MUST implement some form of access control governing the release of  
194 specific attributes and values belonging to specific principals to specific requesting service providers.  
195 Subject to that constraint, any access control mechanism may be supported.

196 A Shibboleth attribute authority MAY implement support for `<saml:SubjectConfirmation>` when  
197 processing queries, but is NOT REQUIRED to do so. That is, it MAY return errors when presented with  
198 queries containing unsupported confirmation methods or when asked to produce assertions containing  
199 them.

200 Finally, a Shibboleth attribute authority MUST support the attribute exchange profile described in section  
201 3.2.

## 202 **2.2.3 Single Sign-On Service**

203 A single sign-on (SSO) service is an HTTP resource controlled by the identity provider that receives and  
204 processes authentication requests sent through the browser from service providers. The SSO service  
205 initiates the authentication process, eventually redirecting the browser to the inter-site transfer service.

206 The SSO service is a Shibboleth-specific service that is not defined by SAML 1.1. It supports a normative  
207 protocol to initiate SSO by a service provider, which SAML 1.1 does not define.

208 An identity provider may expose any number of SSO service endpoints. Each endpoint SHOULD be  
209 protected by SSL/TLS [RFC 2246].

## 210 2.2.4 Inter-Site Transfer Service

211 An inter-site transfer service is an HTTP resource controlled by the identity provider that interacts with the  
212 authentication authority to issue HTTP responses to the principal's browser adhering to the SAML  
213 Browser/POST or Browser/Artifact profiles.

214 In the case of the Browser/POST profile, the HTTP response contains the form controls necessary to  
215 transmit a short-lived authentication assertion inside a digitally signed `<samlp:Response>` message to a  
216 service provider's assertion consumer service.

217 In the case of the Browser/Artifact profile, the HTTP response contains a `Location` header redirecting  
218 the browser to a service provider's assertion consumer service. The redirection URL contains one or more  
219 URL-encoded SAML artifacts.

220 The inter-site transfer service and the SSO service MAY be located at the same HTTP endpoint.

## 221 2.2.5 Artifact Resolution Service

222 An artifact resolution service is a SAML protocol endpoint controlled by the identity provider that receives  
223 requests from a service provider to resolve a SAML artifact into the corresponding assertion in  
224 accordance with the Browser/Artifact profile. The service supports the processing of SAML  
225 `<samlp:Request>` messages containing `<samlp:AssertionArtifact>` elements in a mutually  
226 authenticated fashion. Implementations typically rely on SSL/TLS [RFC 2246] or SAML message  
227 signatures to mutually authenticate the exchange.

## 228 2.3 Service Provider

229 A service provider is an entity that provides a web-based service, application, or resource subject to  
230 authorization or customization on the basis of a security context established by means of the SAML  
231 Browser/POST or Browser/Artifact profiles. It consists of one or more *assertion consumer services*,  
232 defined by the browser profiles, and may include an *attribute requester*.

233 **Note:** Previous versions of this specification referred to these components as the  
234 "SHIRE" and "SHAR", respectively.

235 Each service provider MUST be assigned a unique identifier, or *provider ID*. The identifier MUST be a URI  
236 [RFC 2396] of no more than 1024 characters. Use of an "https" URL for this purpose may be  
237 advantageous for metadata publication (see section ).

### 238 2.3.1 Assertion Consumer Service

239 An assertion consumer service is an HTTP resource controlled by the service provider that processes  
240 form submissions adhering to the SAML Browser/POST profile or HTTP GET requests adhering to the  
241 SAML Browser/Artifact profile to establish a new security context for a principal. Assuming this is  
242 successful, it eventually redirects the user agent to a resource hosted by the service provider.

243 **Note:** [SAMLBind] refers to an assertion consumer service that supports the  
244 Browser/Artifact profile as an *artifact receiver service*, but they are treated as equivalent in  
245 this specification.

246 A service provider may expose any number of assertion consumer service endpoints. Each endpoint  
247 SHOULD be protected by SSL/TLS [RFC 2246].

### 248 **2.3.2 Attribute Requester**

249 Shibboleth supplements the SAML browser profiles with an out of band attribute exchange. A service  
250 provider MAY utilize a SAML protocol binding to send SAML `<samlp:Request>` messages containing  
251 the `<samlp:AttributeQuery>` element to Attribute Authorities and process the resulting attribute  
252 assertions. Implementations typically rely on SSL/TLS [RFC 2246] or SAML message signatures to  
253 mutually authenticate the exchange.

254 Note that in some environments where privacy is not required, a well-known principal identifier might be  
255 communicated in the authentication assertion. This may be done to make the exchange of attributes  
256 optional, or to support a non-SAML mechanism such as LDAP to obtain additional information. Also, the  
257 authentication assertion MAY itself include `<saml:AttributeStatement>` elements (or be  
258 accompanied by additional assertions that do).

259 A Shibboleth attribute requester MAY implement support for `<saml:SubjectConfirmation>` when  
260 submitting queries and processing assertions, but is NOT REQUIRED to do so. That is, it MAY reject  
261 assertions containing unsupported confirmation methods.

### 262 **2.4 WAYF**

263 A WAYF, or "Where are you from?", service is an optional, centralized mechanism for interactively  
264 determining a principal's identity provider. A service provider in general has no means to determine this  
265 without asking the principal or deriving the information through some user agent interaction. The WAYF is  
266 a means for service providers to collectively delegate this step to a separate entity. Service providers are  
267 NOT REQUIRED to utilize a WAYF.

268 A WAYF service MUST support the Shibboleth Authentication Request profile defined in section 3.1.1.  
269 This is the same profile supported by an identity provider's SSO service. The WAYF acts as a proxy for a  
270 service provider and relays the authentication request from the service provider to the SSO service of the  
271 selected identity provider.

272 A WAYF service is free to interact with the principal's user agent in whatever manner it deems appropriate  
273 to determine the identity provider to which to relay the authentication request. This includes, but is not  
274 limited to, presenting lists, a search interface, heuristics based on client characteristics, etc. A WAYF  
275 service SHOULD provide some means for the user agent to cache the user's selection, perhaps using  
276 HTTP cookies, but SHOULD also provide reasonable means for the user to change the selection in the  
277 future.

---

## 278 3 Protocols and Profiles

279 This section defines the message exchanges required of Shibboleth implementations (primarily defined by  
280 SAML 1.1), and additional profiles governing the behavior of Shibboleth components.

### 281 3.1 Authentication Request and Response Profiles

282 To establish a security context at a service provider, Shibboleth combines an Authentication Request  
283 profile defined in this specification with the SAML 1.1 Browser/POST or Browser/Artifact profiles  
284 [SAMLBind]. An identity provider MAY initiate this process without an authentication request by directing  
285 the principal's user agent through unspecified means to its inter-site transfer service with sufficient  
286 information to create the proper HTTP response.

#### 287 3.1.1 Authentication Request Profile

288 A Shibboleth authentication request is a URL-encoded message sent from a service provider (or another  
289 entity on its behalf, such as a WAYF service) to an identity provider's single sign-on service endpoint using  
290 the principal's user agent. Any means of causing the user agent to access the SSO service endpoint can  
291 be used; typically an HTTP redirect is used subsequent to the user agent accessing a secured resource  
292 without a valid security context.

##### 293 3.1.1.1 Required Information

294 **Identification:** urn:mace:shibboleth:1.0:profiles:AuthnRequest

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

296 **Description:** Given below.

297 **Updates:** All earlier technical definitions of the Shibboleth authentication request format

##### 298 3.1.1.2 Message Format and Transmission

299 The HTTP request to the identity provider's SSO service endpoint MUST use the GET method and MUST  
300 contain the following URL-encoded query string parameters:

301 providerId

302 The unique identifier of the requesting service provider

303 shire

304 The assertion consumer service endpoint at the service provider to which to deliver the  
305 authentication response

306 target

307 Generally the URL of a resource accessed at the service provider, it is returned by the  
308 identity provider in the TARGET form control or query string of the authentication  
309 response

310 The query string MAY contain the following optional parameter:

311 time

312 The current time, in seconds elapsed since midnight, January 1<sup>st</sup>, 1970, as a string of up  
313 to 10 base10 digits

314 A WAYF service MUST relay the parameters that it receives from a service provider unchanged to the  
315 identity provider that is ultimately selected, except that it MUST replace the `time` parameter (if present)  
316 with a value generated at the time the user agent is redirected to the identity provider's SSO service.

### 317 3.1.1.3 Processing Rules

318 The SSO service endpoint MUST process the supplied request and either return an error response to the  
319 user agent or attempt to fulfill the request by eventually redirecting the user agent to the inter-site transfer  
320 service (assuming such a redirect is necessary). If an error occurs, the identity provider MAY return a  
321 `<samlp:Response>` in accordance with the Browser/POST profile that contains a `<samlp:Status>`  
322 element with a Value other than `samlp:Success`.

323 The `target` parameter MUST be used as the value of the `TARGET` form control or query parameter in the  
324 HTTP response returned by the inter-site transfer service, whether or not an error has occurred.

325 When using the Browser/POST profile, the `shire` parameter is used as the value of the `ACTION` attribute  
326 in the HTML form in the HTTP response returned by the inter-site transfer service, and is also the value  
327 placed in the `Recipient` attribute of the `<samlp:Response>` element encoded into the `SAMLResponse`  
328 form control.

329 When using the Browser/Artifact profile, the `shire` parameter is used as the URL prefix in the `Location`  
330 header in the HTTP redirect response returned by the inter-site transfer service.

331 The `providerId` parameter MAY be used by the identity provider to customize the processing of the  
332 request based on its knowledge of or relationship with the service provider. Such customization might  
333 include, but is not limited to, the format of the principal's identifier to be returned in the assertion(s), the  
334 credential to use while signing the `<samlp:Response>` message, and the set of attributes to include with  
335 the authentication assertion, if any.

336 Note that if the service provider's identity is used as input to processing the request (which is almost  
337 always the case), then the identity provider MUST have some means to establish that the assertion  
338 consumer service endpoint in the `shire` parameter is in fact associated with the requesting service  
339 provider. Any mechanism to establish this relationship MAY be used, but some mechanism MUST be  
340 used unless the data in the authentication response is invariant with respect to the requesting service  
341 provider. The metadata profile described in section is RECOMMENDED for this purpose.

342 Metadata MAY be used to determine the profile to use in returning the authentication response to the  
343 service provider. If an `<md:AssertionConsumerService>` element in metadata with a `Location`  
344 attribute corresponding to the `shire` parameter indicates support for only one of the response profiles  
345 (via the `Binding` attribute), then the identity provider MUST use this profile when returning the  
346 authentication response. If it cannot or will not use this profile, then the identity provider MUST return an  
347 error message to the user agent.

348 Finally, the `time` parameter MAY be used as an indicator of the freshness of the request so that replayed  
349 requests, such as might be triggered by navigation of a user agent's history list, can be detected. The  
350 parameter MUST NOT be used as part of any security measures.

### 351 3.1.1.4 Example

```
352 https://idp.example.org/SSO?shire=https%3A%2F%2Fsp.example.com%2FShibboleth.shire&  
353 target=https%3A%2F%2Fsp.example.com%2Fcgi-bin%2Flogin.cgi&time=1084819377&  
354 providerId=https%3A%2F%2Fsp.example.com%2Fshibboleth%2F
```



```

411      CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj
412      IHRYQgIv6IqaGG04eTcyVMhoeKE0b45QgvBIaOAPSZB1l3R6+KYiE7x4XAWIrCP+
413      c2MZVeXeTgV3Yz+USLg2Ylon+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE
414      pmqOIFGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq
415      hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n
416      qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX
417      8I3bsbmRAUg4UP9hH6ABVq4KQKMknxulxQxLhpRl1yLGPdiowMNTrEG8cCx3w/w==
418      </ds:X509Certificate>
419      </ds:X509Data>
420      </ds:KeyInfo>
421      </ds:Signature>
422      <samlp:Status><samlp:StatusCode Value="samlp:Success"/></samlp:Status>
423      <saml:Assertion
424      xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
425      AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
426      IssueInstant="2003-04-17T00:46:02Z"
427      Issuer="https://idp.example.org/shibboleth/">
428      <saml:Conditions
429      NotBefore="2003-04-17T00:46:02Z"
430      NotOnOrAfter="2003-04-17T00:51:02Z">
431      <saml:AudienceRestrictionCondition>
432      <saml:Audience>http://sp.example.com/shibboleth/</saml:Audience>
433      </saml:AudienceRestrictionCondition>
434      </saml:Conditions>
435      <saml:AuthenticationStatement
436      AuthenticationInstant="2003-04-17T00:46:00Z"
437      AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
438      <saml:Subject>
439      <saml:NameIdentifier
440      Format="urn:mace:shibboleth:1.0:nameIdentifier"
441      NameQualifier="https://idp.example.org/shibboleth/">
442      3f7b3dcf-1674-4ecd-92c8-1544f346baf8
443      </saml:NameIdentifier>
444      <saml:SubjectConfirmation>
445      <saml:ConfirmationMethod>
446      urn:oasis:names:tc:SAML:1.0:cm:bearer
447      </saml:ConfirmationMethod>
448      </saml:SubjectConfirmation>
449      </saml:Subject>
450      <saml:SubjectLocality IPAddress="127.0.0.1"/>
451      </saml:AuthenticationStatement>
452      </saml:Assertion>
453      </samlp:Response>

```

### 454 3.1.3 Browser/Artifact Authentication Response Profile

455 When the Browser/Artifact profile is used to respond to the service provider, one or more SAML artifacts  
456 are issued to the service provider and transmitted in the query string of an HTTP redirect response. The  
457 format of the HTTP response and the associated processing rules are defined primarily by the SAML  
458 Browser/Artifact profile in [SAMLBind]. Note that the SAML artifact values returned in the SAMLart query  
459 string parameter MUST be URL-encoded.

460 The Browser/Artifact profile permits a variety of artifact formats to be used, and two different formats are  
461 defined by [SAMLBind]. Any defined SAML artifact format MAY be used in Shibboleth.

462 An identity provider MAY send a response without having received an authentication request; in such a  
463 case, the TARGET parameter MUST contain a value expected to be understood by the service provider. In  
464 most cases, this SHOULD be the URL of a resource to be accessed at the service provider, but MAY  
465 contain other values by prior agreement.

466 Upon receiving the artifact(s), the service provider uses a SAML request/response protocol binding to  
467 resolve the artifact(s) into the corresponding SAML assertion(s), in accordance with [SAMLBind].

468 It is RECOMMENDED that service providers enforce a single-use semantic on the artifact values they  
469 receive, to prevent an attacker from interfering with the resolution of an artifact by a user agent and then  
470 resubmitting it to the service provider. If an attempt to resolve an artifact does not complete successfully,  
471 the artifact SHOULD be placed into a blocked artifact list for a period of time that exceeds a reasonable  
472 acceptance period during which the identity provider would successfully resolve the artifact. This  
473 recommendation is in addition to the existing SAML 1.1 requirement that the identity provider enforce a  
474 single-use semantic on artifact values, and anticipates a recommendation added to SAML 2.0 when using  
475 artifacts.

476 Note that the identity provider MAY supply attributes within the SAML assertions it returns in response to  
477 an artifact lookup, at its discretion (this is implicitly permitted by the Browser/Artifact profile).

478 As an additional constraint, the `Issuer` attribute of any assertions returned MUST be set to the unique  
479 identifier of the identity provider issuing the assertion.

### 480 3.1.3.1 Example

481 The example below shows a redirection URL containing a SAML artifact that might be returned when  
482 using this profile. For examples of the subsequent SOAP-based exchange to obtain the assertion, refer to  
483 [SAMLBind].

```
484 https://sp.example.com/Shibboleth.shire?SAMLart=AAEBaPzSPYZVvY%  
485 2BKh00ppTpLcDgQ7pWF5jwFhFEfTvKL3HrNthNzGv59&TARGET=https%3A%2F%2Fsp.example.com%2Fcgi-  
486 bin%2Flogin.cgi
```

## 487 3.2 Attribute Request, Response, and Syntax Profile

488 To support out of band attribute exchange from an identity provider to a service provider, Shibboleth  
489 specifies the use of the SAML request/response protocol using the `<samlp:AttributeQuery>`  
490 element, as defined in [SAMLCore], along with the additional constraints and guidelines defined in this  
491 section.

### 492 3.2.1 Required Information

493 **Identification:** urn:mace:shibboleth:1.0:profiles:attribute

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

495 **Description:** Given below.

496 **Updates:** All earlier technical definitions of the Shibboleth attribute syntax and exchange conventions

### 497 3.2.2 Attribute Requests

498 An attribute request message is a `<samlp:Request>` element containing a  
499 `<samlp:AttributeQuery>` element.

500 Additionally, the `Resource` attribute in the query MUST contain the requesting service provider's unique  
501 identifier. This is used to make up for the lack of an explicit element or attribute in SAML 1.1 to indicate  
502 the issuing service provider.

#### 503 3.2.2.1 Example

504 The example shown does not include any surrounding context from the binding, such as a SOAP  
505 envelope.

```
506 <samlp:Request
```

```

507   xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
508   IssueInstant="2004-05-25T22:46:10Z"
509   MajorVersion="1" MinorVersion="1"
510   RequestID="aaf2319617732113474afe114412ab72">
511   <samlp:AttributeQuery Resource="https://sp.example.com/shibboleth/">
512     <saml:Subject
513       xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
514       <saml:NameIdentifier
515         Format="urn:mace:shibboleth:1.0:nameIdentifier"
516         NameQualifier="http://idp.example.org/shibboleth/">
517         082dd87d-f380-4fd6-8726-694ef2bb71e9
518       </saml:NameIdentifier>
519     </saml:Subject>
520   </samlp:AttributeQuery>
521 </samlp:Request>

```

## 522 3.2.3 Attribute Responses

523 An attribute response is a <samlp:Response> element containing a <samlp:Status> element and  
524 zero or more <saml:Assertion> elements. The assertion(s), if any, SHOULD contain only attribute  
525 statements. The Issuer attribute of any assertions returned MUST be set to the unique identifier of the  
526 identity provider whose attribute authority is issuing the assertion.

527 As noted in section 2.2.2, Shibboleth Attribute Authorities MUST implement some form of access control  
528 over attribute release. They MAY support unauthenticated queries, but SHOULD limit the release of  
529 information in such a case, subject to administrative policy.

### 530 3.2.3.1 Example

531 The example shown does not include any surrounding context from the binding, such as a SOAP  
532 envelope.

```

533 <samlp:Response
534   xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
535   InResponseTo="aaf2319617732113474afe114412ab72"
536   IssueInstant="2004-05-25T22:46:10.940Z"
537   MajorVersion="1" MinorVersion="1"
538   ResponseID="b07b804c7c29ea1673004f3d6f7928ac">
539   <samlp:Status>
540     <samlp:StatusCode Value="samlp:Success"/>
541   </samlp:Status>
542   <saml:Assertion
543     xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
544     AssertionID="a144e8f3adad594a9649924517abe933"
545     IssueInstant="2004-05-25T22:46:10.939Z"
546     MajorVersion="1" MinorVersion="1"
547     Issuer="https://idp.example.org/shibboleth/">
548     <saml:Conditions
549       NotBefore="2004-05-25T22:46:10.939Z"
550       NotOnOrAfter="2004-05-25T23:16:10.939Z">
551     </saml:Conditions>
552     <saml:AttributeStatement>
553       <saml:Subject>
554         <saml:NameIdentifier
555           Format="urn:mace:shibboleth:1.0:nameIdentifier"
556           NameQualifier="https://idp.example.org/shibboleth/">
557           082dd87d-f380-4fd6-8726-694ef2bb71e9
558         </saml:NameIdentifier>
559       </saml:Subject>
560       <saml:Attribute
561         AttributeName="urn:mace:dir:attribute-def:eduPersonEntitlement"
562         AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
563       <saml:AttributeValue>

```

```
564         urn:mace:oclc.org:100277910
565     </saml:AttributeValue>
566     <saml:AttributeValue>
567         urn:mace:example.edu:exampleEntitlement
568     </saml:AttributeValue>
569     <saml:AttributeValue>
570         urn:mace:incommon:entitlement:common:1
571     </saml:AttributeValue>
572 </saml:Attribute>
573 </saml:AttributeStatement>
574 </saml:Assertion>
575 </samlp:Response>
```

## 576 3.2.4 Attribute Naming and Syntax

577 SAML does not constrain the naming of attributes or the syntax of values. It is RECOMMENDED that  
578 Shibboleth attributes be identified with a URI. In such cases, the `AttributeName` XML attribute MUST  
579 contain the URI that identifies the attribute, and the `AttributeNamespace` XML attribute SHOULD  
580 contain the value `urn:mace:shibboleth:1.0:attributeNamespace:uri`. It MAY contain a  
581 different value by prior agreement.

582 It is also RECOMMENDED that attribute values be expressed, when possible, as a single XML text node  
583 within the `<saml:AttributeValue>` element, using an XML Schema built-in datatype ([Schema2]). In  
584 such cases, the `xsi:type` XML attribute SHOULD be used to indicate the built-in datatype that describes  
585 the allowable syntax of the value.

586 If the value is not from a built-in datatype, the `xsi:type` attribute MAY be used to indicate the extension  
587 type in use, but implementers are cautioned that this may require a relying party to be aware of the  
588 extension in order to process the assertion. Omitting the `xsi:type` attribute is RECOMMENDED in such  
589 cases.

590 See the example in section 3.2.3.1.

## 591 3.3 Transient NameIdentifier Format

592 SAML identifies principals in assertions using the `<saml:NameIdentifier>` element, which contains a  
593 pair of descriptive XML attributes, `Format` and `NameQualifier`. See the example in section 3.1.2.1.

594 Shibboleth permits any legal SAML name identifier to be used, but also defines a special kind of identifier  
595 with the `Format` value of `urn:mace:shibboleth:1.0:nameIdentifier`. Identifiers of this format  
596 MUST satisfy the following criteria:

- 597 • The identifier has transient semantics and SHOULD be treated as an opaque and temporary  
598 value by the relying party.
- 599 • The identifier MUST be constructed in accordance with the rules for SAML identifiers (see  
600 section 1.2.3 of [SAMLCore]), and SHOULD NOT exceed a length of 256 characters.
- 601 • If present, the `NameQualifier` attribute MUST be set to the unique identifier of the identity  
602 provider issuing the assertion containing the element. That is, the value of the  
603 `NameQualifier` and `Issuer` attributes MUST be identical.

## 604 3.4 Metadata Profile

605 **Editor's Note:** This profile has been jointly submitted with Trustgenix, Inc. to the OASIS  
606 Security Services Technical Committee for consideration and this section has been  
607 adapted to reference and build on the draft submission by specifying only Shibboleth-  
608 specific constraints. Accordingly, this section may undergo changes until the submission  
609 has reached committee draft status.

610 SAML (and by extension Shibboleth) profiles require agreements between system entities regarding  
611 identifiers, binding/profile support and endpoints, certificates and keys, and so forth. A metadata  
612 specification is useful for describing this information in a standardized way.

613 Although SAML 1.1 did not include such a specification, SAML 2.0 includes one in [SAML2Meta].  
614 Subsequently, a profile of this specification was developed for use by SAML 1.1 deployments (see  
615 [SAML1Meta]). Shibboleth identity and service providers SHOULD describe their characteristics using this  
616 profile.

617 Role elements defined by this profile applicable to Shibboleth include `<md:IDPSSODescriptor>`,  
618 `<md:SPSSODescriptor>`, `<md:AuthnAuthorityDescriptor>`, and  
619 `<md:AttributeAuthorityDescriptor>`.

620 Multiple Shibboleth entities can be collected into groups using the `<md:EntitiesDescriptor>`  
621 element.

622 Specific use of these elements MUST adhere to the profile defined in [SAML1Meta]. Additional guidelines  
623 and processing rules pertaining to Shibboleth are specified below.

### 624 3.4.1 Element `<md:EntitiesDescriptor>`

625 The `Name` XML attribute, if present, SHOULD be a URI.

### 626 3.4.2 Element `<md:EntityDescriptor>`

627 A Shibboleth identity or service provider SHOULD be represented by exactly one  
628 `<md:EntityDescriptor>`. Its unique identifier MUST be placed in the `entityID` XML attribute.

629 If a URL is used as the unique identifier of an entity, it is RECOMMENDED that resolving this URL  
630 produce a SAML metadata document containing a single `<md:EntityDescriptor>` representing that  
631 entity.

632 Note that metadata can vary based on the relying party in question. Resolving an identifier into metadata  
633 MAY require authentication of the requester so as to produce the metadata response appropriate for that  
634 relying party.

### 635 3.4.3 Element `<md:IDPSSODescriptor>`

636 A Shibboleth identity provider MUST include this element in its metadata. The  
637 `protocolSupportEnumeration` XML attribute MUST include at least the values  
638 `urn:oasis:names:tc:SAML:1.0:protocol:v1.1` and `urn:mace:shibboleth:1.0`

639 At least one `<md:SingleSignOnService>` element MUST be present.

640 At least one of the `<md:SingleSignOnService>` elements' `Binding` XML attribute MUST contain the  
641 value `urn:mace:shibboleth:1.0:profiles:AuthnRequest`

642 The location specified in its `Location` XML attribute MUST support the Authentication Request profile  
643 defined in section 3.1.1.

#### 644 **3.4.4 Element `<md:AuthnAuthorityDescriptor>`**

645 A Shibboleth identity provider that supports an Authentication Authority service as described in section  
646 2.2.1 MUST include this element in its metadata if it supports lookup of assertions by SAML query or  
647 identifier lookup. The `protocolSupportEnumeration` XML attribute MUST include at least the value  
648 `urn:oasis:names:tc:SAML:1.0:protocol:v1.1`

#### 649 **3.4.5 Element `<md:AttributeAuthorityDescriptor>`**

650 A Shibboleth identity provider that supports an Attribute Authority service as described in section 2.2.2  
651 MUST include this element in its metadata. The `protocolSupportEnumeration` XML attribute MUST  
652 include at least the value `urn:oasis:names:tc:SAML:1.0:protocol:v1.1`

#### 653 **3.4.6 Element `<md:SPSSODescriptor>`**

654 A Shibboleth service provider MUST include this element in its metadata. The  
655 `protocolSupportEnumeration` XML attribute MUST include at least the value  
656 `urn:oasis:names:tc:SAML:1.0:protocol:v1.1`

---

## 657 4 Security and Privacy Considerations

658 As Shibboleth is principally a set of SAML profiles, the general security and privacy considerations that  
659 apply to SAML apply to Shibboleth (see [SAMLSecure]).

### 660 4.1 Additional Browser Profile Considerations

#### 661 4.1.1 Information Leakage and Impersonation

662 The SAML browser profiles contain a presumption that they are initiated by an identity provider. Assertion  
663 information (or an artifact) is therefore sent directly to service providers using locations known to be  
664 appropriate and secure.

665 The use of the Authentication Request profile defined by section 3.1.1 introduces the possibility of a  
666 malicious entity impersonating another service provider by identifying itself as one provider while indicating  
667 that the authentication response be delivered to the attacker instead. In the case of the POST profile, this  
668 can result in unintended leakage of personally identifying information within the assertion(s) to the  
669 attacker. In the case of the Artifact profile, the attacker could potentially impersonate the principal by  
670 immediately submitting the artifact(s) to the real service provider, who can subsequently authenticate to  
671 the identity provider to obtain the assertion.

672 To mitigate both attacks, it is critical for the identity provider to securely associate the assertion consumer  
673 service location to be used with the service provider to whom the assertion(s) or artifact(s) are issued. A  
674 digital signature over the authentication request would be an alternate countermeasure, but this is not  
675 supported by the current profile.

676 Another source of information leakage is the `target` parameter sent with the Authentication Request  
677 URL and returned in both Browser profiles. This parameter is informally associated with the resource URL  
678 being requested from the service provider, but it is in fact potentially opaque to the identity provider.  
679 Exposing the resource URL makes unneeded information available about the principal's activities to the  
680 identity provider and possibly various log files. It is therefore RECOMMENDED that service providers  
681 utilize some kind of obfuscation, mapping, encryption, or other mechanism to prevent the exposure of  
682 resource URLs in plaintext in this parameter.

683 Finally, when user privacy in service provider interactions is a consideration or requirement, Shibboleth  
684 provides an explicit mechanism for effective anonymity through the use of a transient identifier (see  
685 section 3.3), provided that the SAML attributes supplied in conjunction with it or subsequently are  
686 sufficiently generic so as not to inadvertently narrow down or identify the principal. It is important to avoid  
687 facilitating coordination by one or more service providers in correlating the principal's activity by insuring  
688 that a different transient identifier is used across time and space. Therefore, it is RECOMMENDED that a  
689 given transient identifier not be used more than once in assertions issued by an identity provider for a  
690 principal in different executions of an authentication response profile.

#### 691 4.1.2 Time Synchronization

692 The Browser/POST profile relies on tight synchronization of clocks between the identity and service  
693 providers to limit the usefulness of the bearer assertion. Additionally, assertions may be issued with  
694 expiration conditions that cannot be effectively honored if clock skew is excessive.

695 It is RECOMMENDED that secure time sources be used to maintain clock synchronization within the  
696 bounds usually associated with protocols like Kerberos (i.e., on the order of 5 minutes or less).

---

## 5 References

697

698 The following works are cited in the body of this specification.

### 5.1 Normative References

699

- 700       **[RFC 2109]**       S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF  
701                       RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 702       **[RFC 2119]**       S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF  
703                       RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 704       **[RFC 2246]**       T. Dierks, C. Allen. *The TLS Protocol Version 1.0*. IETF RFC 2246, January 1999.  
705                       <http://www.ietf.org/rfc/rfc2246.txt>.
- 706       **[RFC 2396]**       T. Berners-Lee et al. *Uniform Resource Identifiers (URI): Generic Syntax*. IETF  
707                       RFC 2396, August, 1998. <http://www.ietf.org/rfc/rfc2396.txt>.
- 708       **[SAMLCore]**       E. Maler et al. *Assertions and Protocols for the OASIS Security Assertion Markup  
709                       Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-core-  
710                       1.1. <http://www.oasis-open.org/committees/security/>.
- 711       **[SAMLBind]**       E. Maler et al. *Bindings and Profiles for the OASIS Security Assertion Markup  
712                       Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-  
713                       bindings-profiles-1.1. <http://www.oasis-open.org/committees/security/>.
- 714       **[SAML-XSD]**       E. Maler et al. *SAML assertion schema*. OASIS, September 2003. Document ID  
715                       oasis-sstc-saml-schema-assertion-1.1. [open.org/committees/security/](http://www.oasis-<br/>716                       open.org/committees/security/).
- 717       **[SAMPL-XSD]**       E. Maler et al. *SAML protocol schema*. OASIS, September 2003. Document ID  
718                       oasis-sstc-saml-schema-protocol-1.1. [open.org/committees/security/](http://www.oasis-<br/>719                       open.org/committees/security/).
- 720       **[SAMLSecure]**       E. Maler et al. *Security and Privacy Considerations for the OASIS Security  
721                       Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID  
722                       oasis-sstc-saml-sec-consider-1.1. [open.org/committees/security/](http://www.oasis-<br/>723                       open.org/committees/security/).
- 724       **[SAML2Prof]**       S. Cantor et al., *Profiles for the OASIS Security Assertion Markup Language  
725                       (SAML) V2.0*. OASIS SSTC, September 2004. Document ID sstc-saml-profiles-  
726                       2.0-cd-02. See <http://www.oasis-open.org/committees/security/>.
- 727       **[SAML2Meta]**       S. Cantor et al., *Metadata for the OASIS Security Assertion Markup Language  
728                       (SAML) V2.0*. OASIS SSTC, September 2004. Document ID sstc-saml-metadata-  
729                       2.0-cd-02. See <http://www.oasis-open.org/committees/security/>.
- 730       **[SAMLMeta-xsd]**    S. Cantor et al., *SAML metadata schema*. OASIS SSTC, September 2004.  
731                       Document ID sstc-saml-schema-metadata-2.0. See [http://www.oasis-  
open.org/committees/security/](http://www.oasis-<br/>732                       open.org/committees/security/).
- 733       **[SAML1Meta]**       G. Whitehead and S. Cantor, *SAML 1.x Metadata Profile*. OASIS SSTC,  
734                       November 2004. Document ID draft-saml1x-metadata-01. See [http://www.oasis-  
open.org/committees/security/](http://www.oasis-<br/>735                       open.org/committees/security/).
- 736       **[Schema2]**       P. V. Biron et al. *XML Schema Part 2: Datatypes*. World Wide Web Consortium  
737                       Recommendation, May 2001. <http://www.w3.org/TR/xmlschema-2/>.

738 **5.2 Non-Normative References**

739 **[SAML2Gloss]** J. Hodges et al., *Glossary for the OASIS Security Assertion Markup Language*  
740 *(SAML) V2.0*. OASIS SSTC, September 2004. Document ID sstc-saml-glossary-  
741 2.0-cd-02. See <http://www.oasis-open.org/committees/security/>.

742 **[LibertyBind]** J. Kemp et al., *Liberty Bindings and Profiles Specification* Version 1.2, Liberty  
743 Alliance Project, August 2004, [http://www.projectliberty.org/specs/v1\\_2/liberty-](http://www.projectliberty.org/specs/v1_2/liberty-architecture-bindings-profiles-v1.2.pdf)  
744 [architecture-bindings-profiles-v1.2.pdf](http://www.projectliberty.org/specs/v1_2/liberty-architecture-bindings-profiles-v1.2.pdf).

745 **[LibertyProt]** J. Kemp et al., *Liberty Protocols and Schema Specification* Version 1.2, Liberty  
746 Alliance Project, August 2004, [http://www.projectliberty.org/specs/v1\\_2/liberty-](http://www.projectliberty.org/specs/v1_2/liberty-architecture-protocols-schema-v1.2.pdf)  
747 [architecture-protocols-schema-v1.2.pdf](http://www.projectliberty.org/specs/v1_2/liberty-architecture-protocols-schema-v1.2.pdf).