



---

# Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1

**Draft 05, 4 May 2004**

**Document identifier:**

sstc-saml-tech-overview-1.1-draft-05

**Location:**

[http://www.oasis-open.org/committees/documents.php?wg\\_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security)

**Editors:**

John Hughes, Entegriy Solutions  
Eve Maler, Sun Microsystems

**Contributors:**

Rob Philpott, RSA Security

**Abstract:**

The Security Assertion Markup Language (SAML) standard defines a framework for exchanging security information between online business partners. It was developed by the Security Services Technical Committee (SSTC) of the standards organization OASIS (the Organization for the Advancement of Structured Information Standards). This document provides a technical description of SAML V1.1.

**Status:**

This is a non-normative document; readers should refer to the normative specification suite for precise information concerning SAML V1.1. This document is not currently on an OASIS Standard track. It has been produced by the Security Services Technical Committee. Publication of this draft does not imply TC endorsement. This working draft may be updated, replaced, or obsoleted at any time.

Committee members should submit comments to the [security-services@lists.oasis-open.org](mailto:security-services@lists.oasis-open.org) list. Others should submit comments by filling out the form at [http://www.oasis-open.org/committees/comments/form.php?wg\\_abbrev=security](http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=security). The committee will publish vetted errata on the Security Services TC web page (<http://www.oasis-open.org/committees/security/>).

For information on whether any patents have been disclosed that may be essential to implementing the SAML specification suite, and any offers of patent licensing terms, please refer to the Intellectual Property Rights web page for the Security Services TC (<http://www.oasis-open.org/committees/security/ipr.php>).

---

36 **Table of Contents**

37 1 Introduction.....3

38 2 SAML Overview.....4

39 3 SAML Architecture.....6

40 3.1 SAML Concepts.....6

41 3.2 SAML Structure and Examples.....7

42 3.3 Security of SAML.....9

43 4 Use Cases and Profiles.....10

44 4.1 Browser/Artifact Profile.....10

45 4.1.1 Detailed Processing for the Source-Site-First Scenario.....11

46 4.2 Browser/POST Profile.....12

47 4.2.1 Detailed Processing.....13

48 4.3 Destination-Site-First.....14

49 4.3.1 Detailed Processing for the Destination-Site-First Scenario.....14

50 5 Documentation Roadmap .....16

51

---

# 1 Introduction

52

53 The Security Assertion Markup Language (SAML) standard defines a framework for exchanging security  
54 information between online business partners.

55 More precisely, SAML defines a common XML framework for exchanging security assertions between  
56 entities. As stated in the SSTC charter, the purpose of the Technical Committee is:

57 *...to define, enhance, and maintain a standard XML-based framework for creating and*  
58 *exchanging authentication and authorization information.*

59 SAML is different from other security systems due to its approach of expressing assertions about a  
60 subject that other applications within a network can trust. What does this mean? To understand the  
61 answer, you need to know the following two concepts used within SAML:

## 62 **Asserting party**

63 The system, or administrative domain, that asserts information about a subject. For instance, the  
64 asserting party asserts that this user has been authenticated and has given associated attributes. For  
65 example: This user is **John Doe**, he has an email address of [john.doe@acompany.com](mailto:john.doe@acompany.com), and he  
66 was authenticated into this system using a **password** mechanism. In SAML, asserting parties are also  
67 known as SAML authorities.

## 68 **Relying party**

69 The system, or administrative domain, that relies on information supplied to it by the asserting party. It  
70 is up to the relying party as to whether it trusts the assertions provided to it. SAML defines a number  
71 of mechanisms that enable the relying party to trust the assertions provided to it. It should be noted  
72 that although a relying party can trust the assertions provided to it, local access policy defines whether  
73 the subject may access local resources. Therefore, although the relying party trusts that I'm **John**  
74 **Doe** – it doesn't mean I'm given carte blanche access to all resources.

75

## 2 SAML Overview

76

Why is SAML needed? The SSTC developed a number of use cases to drive SAML's requirements. For SAML 1.x, the most important of these use cases described a SAML-based solution to the problem of Web Single Sign-On (SSO). Web SSO allows users to gain access to website resources in multiple domains without having to re-authenticate after initially logging in to the first domain. To achieve SSO, the domains need to form a trust relationship before they can share an understanding of the user's identity that allows the necessary access. Figure 1 illustrates the high-level Web SSO use case; more details about how this is achieved are provided later in the document.

77

78

79

80

81

82

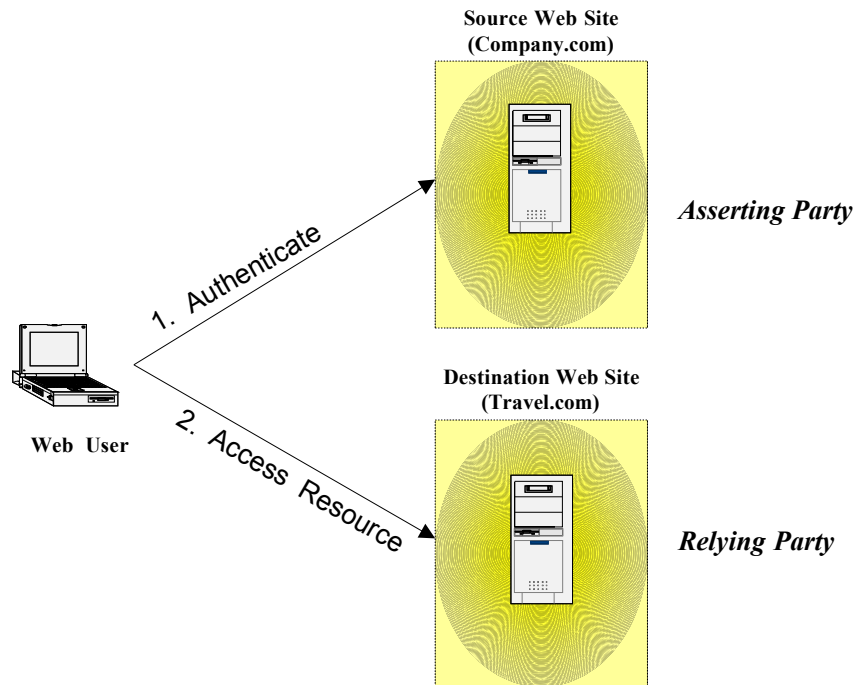


Figure 1: Web SSO High-Level Use Case

84

Following are some specific scenarios to which SAML's SSO capabilities are relevant:

85

- **Government Portal**

86

A Government department has implemented a centralized portal system. Linked to the portal system are a number of satellite systems. The central portal system maintains the authentication information for all users; however, the satellite systems use a wide range of access management products from a variety of vendors. Users should only be required to be authenticated once, and they can either go initially to the satellite system or the central portal. In this scenario the portal is the asserting party for the whole system and the satellite systems are the relying parties.

87

88

89

90

91

92

- **Travel Bookings**

93

Authenticated users of Company.com need to gain access to protected resources at Travel.com in order to make travel arrangements. The Company.com users should not need to have to re-authenticate to Travel.com. In addition, only certain privileged users (for example, above a certain job grade) may book international travel.

94

95

96

97

- **Goods Purchasing**

98

Authenticated users of Company.com use an internal purchasing system to place orders for office supplies from Supplier.com. Supplier.com needs to know the user and their shipping address.

99

100

Supplier.com also needs to know whether the user is authorized to purchase goods of that value.

101

The following technical factors drove an urgent need for SAML when it was first created:

- 102 • **Limitations of browser cookies:** Before SAML, most SSO products used browser cookies to maintain  
103 state so that re-authentication is not required. Browser cookies are not transferred between DNS  
104 domains. So, if you obtain a cookie from www.abc.com, then that cookie will not be sent in any HTTP  
105 messages to www.xyz.com. This could even apply within an organization that has separate DNS  
106 domains. Therefore, to solve the cross-domain SSO problem requires the application of a different  
107 approach.
- 108 • **SSO interoperability:** Products had implemented cross-domain SSO in completely proprietary ways,  
109 meaning that organizations that want to perform cross-domain SSO had to use the same SSO product  
110 in all the domains, whether within one organization or across trading partners.
- 111 • **Web services:** There is an increasing trend towards inter-organizational distributed computing. Many  
112 standards have emerged that facilitate this trend, in particular web services based applications.  
113 However, there has been no standard way to convey security attributes associated with inter-  
114 organizational communications.
- 115 When SAML V2.0 is released in 2004, additional use cases will be supported. To find out more about the  
116 scope and design of SAML V2.0, visit the SSTC home page at [http://www.oasis-](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security)  
117 [open.org/committees/tc\\_home.php?wg\\_abbrev=security](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security) and review the SAML V2.0 Scope/Work Items  
118 document.

## 3 SAML Architecture

119

120 The SAML technology is rooted in XML. The information passed around between asserting parties (SAML  
121 authorities) and relying parties is mostly in the form of XML, and the format of these XML messages and  
122 assertions is defined in a pair of SAML XML schemas.

### 3.1 SAML Concepts

123

124 SAML has the following key concepts:

- 125 • **Assertions:** An assertion is a package of information that supplies one or more statements made by  
126 a SAML authority. SAML defines three kinds of statements that can be carried within an assertion.  
127 *Authentication statements* say “This subject was authenticated by this means at this time.” *Attribute*  
128 *statements* provide specific details about the subject (for example, that a user holds “Gold” status).  
129 *Authorization decision statements* identify what the subject is entitled to do (for example, whether a  
130 user is permitted to buy a specified item). The XML format for assertions and their allowable  
131 extensions is defined in an XML schema.
- 132 • **Protocol:** SAML defines a request/response protocol for obtaining assertions. A SAML request can  
133 either ask for a specific known assertion or make authentication, attribute, and authorization  
134 decision queries, with the SAML response providing back the requested assertions. The XML format  
135 for protocol messages and their allowable extensions is defined in an XML schema.
- 136 • **Bindings:** A binding details exactly how the SAML protocol maps onto transport and messaging  
137 protocols. For instance, the SAML specification provides a binding of how SAML request/responses  
138 are carried within SOAP exchange messages over HTTP.
- 139 • **Profiles:** Profiles are technical descriptions of particular flows of assertions and protocol messages  
140 that define how SAML can be used for a particular purpose. They are derived from use cases. Use  
141 cases and profiles are discussed later on in the document.

142 Figure 2 shows the relationship between these components.

143

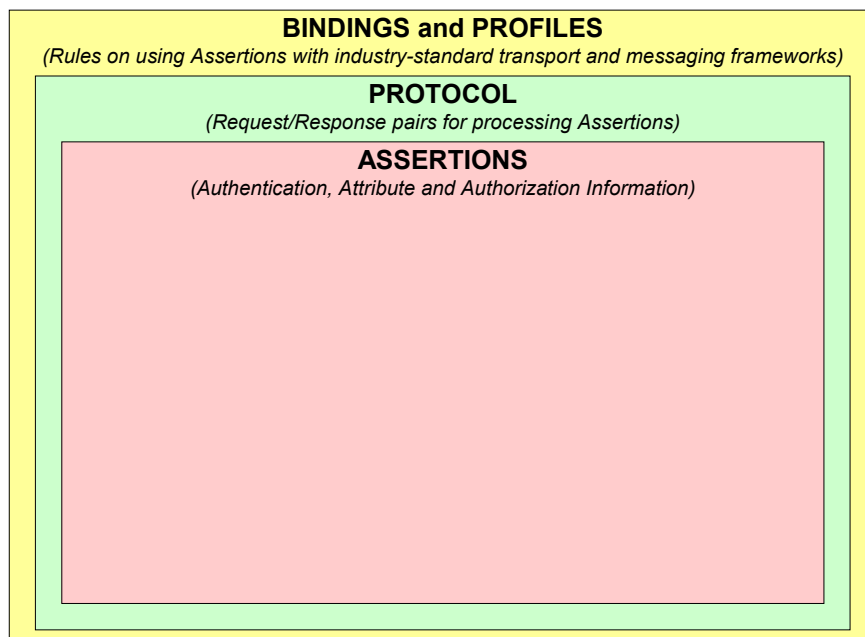


Figure 2: Relationship between SAML Components

145

146 **3.2 SAML Structure and Examples**

147 The sole binding specified in SAML V1.1 is the “SOAP-over HTTP” binding. Figure 3 illustrates the  
148 relationship between SOAP and the SAML protocol messages being transported within the SOAP body.  
149

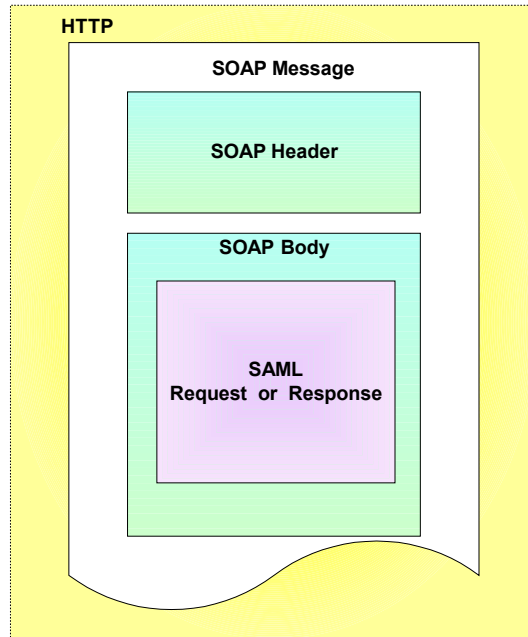


Figure 3: SOAP over HTTP Binding

150 SAML responses carry assertions that satisfy the parameters of the SAML request. Figure 4 illustrates a  
151 SAML response being transported within a SOAP body. Note the following characteristics:

- 152 • The SAML response contains SAML status information in addition to one or more assertions.
- 153 • One or more assertions can be transported, although typically only a single assertion is provided in a  
154 SAML response.
- 155 • An assertion consists of one or more statements. For SSO, typically a SAML assertion will contain a  
156 single authentication statement and possibly a single attribute statement.

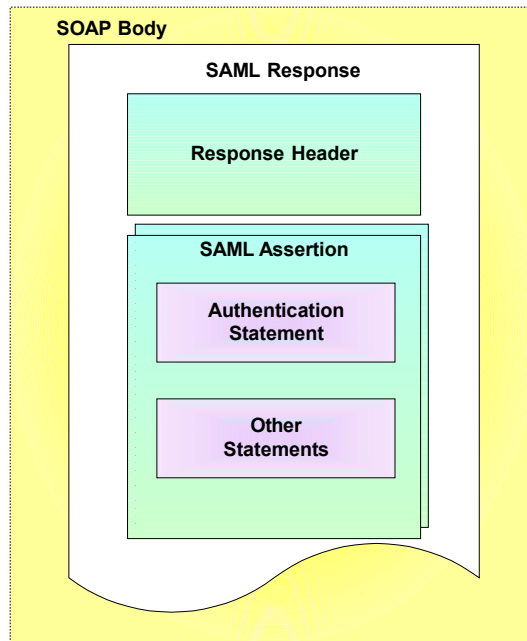


Figure 4: SAML Response Structure

157 So what does the XML look like? Figure 5 shows an example of a SAML request being transported within  
 158 a SOAP message. In this example, a SAML assertion is being requested pertaining to a supplied artifact.  
 159 The use of the artifact is explained later in the Use Case and Profiles section. The SAML request has  
 160 been highlighted.

```

161 <env:Envelope
162   xmlns:env="http://www.w3.org/2003/05/soap/envelope/"
163   <env:Body>
164     <samlp:Request
165       xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
166       xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
167       MajorVersion="1"
168       MinorVersion="1"
169       RequestID=" 192.168.16.51.1024506224022"
170       IssueInstant="2002-06-19T17:03:44.022Z">
171       <samlp:AssertionArtifact>
172         AAGZE1RNQJEFzYNCGAGPjWvtDIRSZ4
173         lWDqBphqAEYkgG/RBdHoeMsulf
174       </samlp:AssertionArtifact>
175     </samlp:Request>
176   </env:Body>
177 </env:Envelope>
  
```

Figure 5: SAML Artifact Request

178 Figure 6 shows how a SAML response is embedded within a SOAP message. The SAML response  
 179 provides details as to the version of SAML being used and what request it is responding to. The  
 180 ResponseID, InResponseTo, version numbers, IssueInstant and the status code represent the SAML  
 181 response header. Within the response is the SAML assertion and typically one or more statements. The  
 182 SAML response has been highlighted.

```

183 <env:Envelope
184   xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
185   <env:Body>
186     <samlp:Response
187       xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
188       ResponseID="huGxcDQc4cNdDyocphmi6CxEMnga"
189       InResponseTo=" 192.168.16.51.1024506224022"
190       MajorVersion="1"
191       MinorVersion="1"
192       IssueInstant="2002-06-19T17:05:37.795Z">
193       <samlp:Status>
194         <samlp:StatusCode Value="samlp:Success" />
195       </samlp:Status>
  
```

```

196
197 ..... SAML ASSERTION AND STATEMENTS
198
199 </samlp:Response>
200 </env:Body>
201 </env:Envelope>

```

Figure 6: SAML Response

Figure 7 shows an example assertion with a single authentication statement. The authentication statement has been highlighted. Note the following:

- The subject (e.g. user) that the authentication pertains to is “joe”. The format of the subject has been defined. In this case its a custom format; however, a number of predefined formats have been provided in the SAML specification, including email addresses and X.509 subject names.
- Joe was originally authenticated using a password mechanism at “2002-06-19T17:05:17.706Z”.

```

208 <saml:Assertion
209   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
210   MajorVersion="1"
211   MinorVersion="1"
212   AssertionID="buGxcG4gILg5NlocyLccDz6iXrUa"
213   Issuer="www.acompany.com"
214   IssueInstant="2002-06-19T17:05:37.795Z">
215   <saml:Conditions NotBefore="2002-06-19T17:00:37.795Z"
216     NotOnOrAfter="2002-06-19T17:10:37.795Z"/>
217   <saml:AuthenticationStatement
218     AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
219     AuthenticationInstant="2002-06-19T17:05:17.706Z">
220     <saml:Subject>
221       <saml:NameIdentifier
222         NameQualifier=http://www.acompany.com
223         Format="http://www.customformat.com/">
224         uid=joe
225       </saml:NameIdentifier>
226       <saml:SubjectConfirmation>
227         <saml:ConfirmationMethod>
228           urn:oasis:names:tc:SAML:1.0:cm:artifact-01
229         </saml:ConfirmationMethod>
230       </saml:SubjectConfirmation>
231     </saml:Subject>
232   </saml:AuthenticationStatement>
233 </saml:Assertion>

```

Figure 7: SAML Assertion

### 3.3 Security of SAML

Just providing assertions from an asserting party to a relying party may not be adequate for a secure system. How does the relying party trust what is being asserted to it? In addition, what prevents a “man-in-the-middle” attack that grabs assertions to be illicitly “replayed” at a later date? SAML defines a number of security mechanisms that prevent or detect such attacks. The primary mechanism is for the relying party and asserting party to have a pre-existing trust relationship, typically involving a Public Key Infrastructure (PKI). Whilst use of a PKI is not mandated, it is recommended. Use of particular mechanisms is described for each profile; however, an overview of what is recommended is provided below:

- Where **message integrity** and **message confidentiality** are required, then HTTP over SSL 3.0 or TLS 1.0 is recommended.
- When a relying party requests an assertion from an asserting party then **bi-lateral authentication** is required and the use of SSL 3.0 or TLS 1.0 using server *and* client authentication are recommended.
- When an assertion is “pushed” to a relying party (as with the Browser/POST profile), then it is mandated that the response message be **digitally signed** using the XML digital signature standard.

## 4 Use Cases and Profiles

249

250 Early in its business requirements analysis, the SSTC defined a number of use cases for SAML. To date,  
251 only the Web SSO use case has been profiled. With the emergence of SAML V2.0 in 2004, a number of  
252 other use cases will also be profiled.

253 SAML V1.1 has defined Web SSO two profiles. These profiles assume:

- 254 • Use of a standard commercial web browser using either HTTP or HTTPS
- 255 • The user has authenticated to the local source site
- 256 • The assertion's subject refers implicitly to the user that has been authenticated

257 The profiles are:

- 258 • **Browser/Artifact Profile:** This represents a “pull model”. A special form of reference to the  
259 authentication assertion (called an artifact) is sent to the relying party, which can use this reference to  
260 obtain (or pull) the assertion from the Asserting Party.
- 261 • **Browser/POST Profile:** This represents a “push model”. An assertion is POSTed (using the HTTP  
262 POST command) directly to the relying party.

263 We shall now go on to describe in detail each of these profiles.

### 4.1 Browser/Artifact Profile

264

265 This Browser/Artifact profile is based on a pull model. Figure 8 illustrates the overall processing.

266

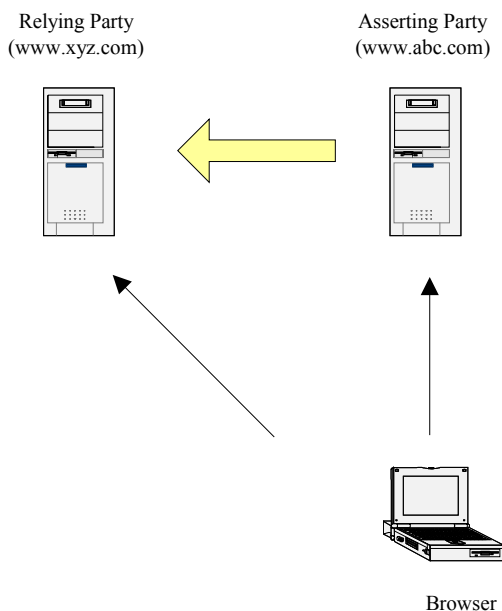


Figure 8: Browser/Artifact Profile Overview

267 In summary, the processing is as follows:

- 268 1. A user has an authenticated session on the local source site (asserting party).
- 269 2. The user wants to access a resource on the destination web site and is directed there. In the HTTP  
270 message, an HTTP query variable is passed called an *artifact*. The artifact is a base-64 encoded  
271 string. It consists of a unique identity of the source site (called the Source ID) and a unique reference  
272 to the assertion (called the AssertionHandle). The artifact therefore enables the destination web site to  
273 reference an assertion on a given web site.
- 274 3. The destination site (relying party) needs to determine the identity and entitlements of the user and

275 sends a SAML request, containing the artifact, to the local site (the asserting party) asking it what it can  
 276 assert about the user. The assertions are transferred back in a SAML response.  
 277 4. The destination site then can make whatever authentication and authorization decisions it needs to,  
 278 based on the received assertion(s).  
 279 Two scenarios are possible in this use case:  
 280 • **Source-site-first:** The user visits their local source site first and is authenticated at the source site  
 281 before using a click-through link to gain access to the destination site.  
 282 • **Destination-site-first:** The user visits the destination site first; however, they need to be authenticated  
 283 at the source site prior to being granted access to resources on the destination site. This scenario  
 284 typically represents a centralized portal architecture.  
 285 The SAML 1.1 specifications **only** define the Source-site-first use case.

#### 286 4.1.1 Detailed Processing for the Source-Site-First Scenario

287 The following figure shows the processing and message flows for the Browser/Artifact profile in the  
 288 Source-Site-First scenario. In this example, the source web site includes a component called an Inter-site  
 289 Transfer Service (ITS). This is an addressable component that provides a point of functionality for SAML  
 290 processing such as artifact and redirect generation.  
 291

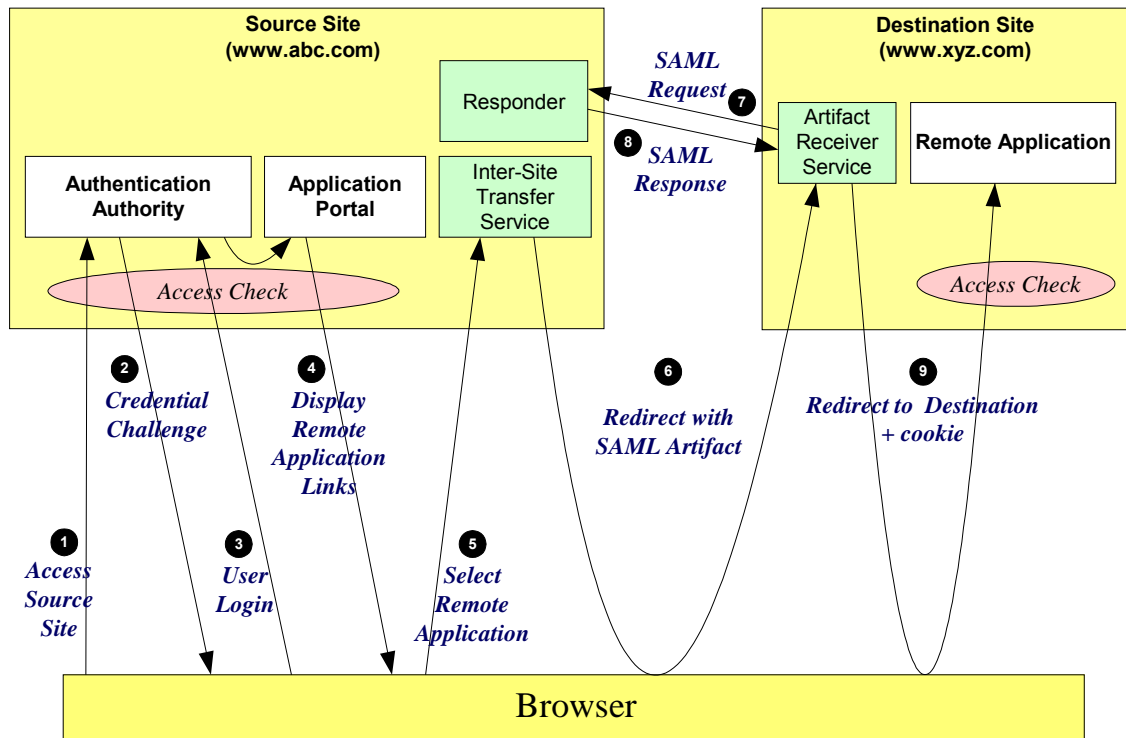


Figure 9: Browser/Artifact Profile – Local-Site-First - Detailed Processing

292 The processing is as follows:

- 293 1. The user accesses the source web site ([www.abc.com](http://www.abc.com)).
- 294 2. The source web site performs an access check and determines that the user does not have a current  
 295 session and requires the user to be authenticated. As a result, the user is challenged to authenticate.
- 296 3. The user supplies back credentials, for instance username and password.
- 297 4. If the authentication is successful, then a session is created for the user and the appropriate welcome  
 298 screen of the Portal application is displayed to the user.
- 299 5. The user selects a menu option (or function) on the displayed screen that means the user wants to  
 300 access a resource or application on a destination web site [www.xyz.com](http://www.xyz.com) (although, of course, the user

301 may not be made aware of this). This causes a HTTP request to be sent to the source site's Inter-site  
302 Transfer Service (in this example, hosted on the same web site). The request contains the URL of the  
303 resource on the destination site. This is known as the TARGET URL. For instance, the portal  
304 application will issue an HTTP GET to the Inter-site Transfer Service on the [www.abc.com](http://www.abc.com) site which  
305 is listening on port 8002. The URL would look something like the following (without the URL encoding):  
306 <https://www.abc.com:8002/InterSiteTransfer?TARGET=http://www.xyz.com/index.asp>

307 6. The Inter-site Transfer Service generates an assertion for the user while also creating an artifact (The  
308 Asserting Party). The artifact contains the source ID of the [www.abc.com](http://www.abc.com) SAML responder together  
309 with a reference to the assertion (the AssertionHandle). The Inter-site Transfer Service then sends  
310 back an HTTP redirection response to the browser, with the HTTP location header containing the URL  
311 of the Artifact Receiver service, the TARGET URL, and the artifact. On processing the redirect, the  
312 Browser will issue an HTTP GET of the form provided below, where the <artifact> is a base 64  
313 encoded number. This will be sent to the server hosting the TARGET URL.  
314 <https://www.xyz.com:7001/ArtifactConsumer?TARGET=http://www.xyz.com/index.asp&SAMLart=<artifact>>

315 7. On receiving the HTTP message, the Artifact Receiver, on the destination web site, extracts the  
316 source-ID. A mapping between source IDs and remote Responders will already have been established  
317 administratively. The Artifact Receiver will therefore know that it has to contact the [www.abc.com](http://www.abc.com)  
318 SAML responder at the prescribed URL. The [www.xyz.com](http://www.xyz.com) Artifact Receiver will send a SAML request  
319 to the [www.abc.com](http://www.abc.com) SAML responder containing the artifact supplied by the Inter-site Transfer Service  
320 of [www.abc.com](http://www.abc.com).

321 8. The [www.abc.com](http://www.abc.com) SAML responder supplies back a SAML response message containing the  
322 assertion generated during step 7. In most implementations, if a valid assertion is received back, then  
323 a session on [www.xyz.com](http://www.xyz.com) is established for the user (the relying party) at this point.

324 9. The Artifact Receiver, on the destination web site, sends a redirection message containing a cookie  
325 back to the browser. The cookie identifies the session. The browser then processes the redirect  
326 message and issues a HTTP GET to the TARGET resource on [www.xyz.com](http://www.xyz.com). The GET message  
327 contains the cookie supplied back by the Artifact Receiver. An access check is then back to  
328 established whether the user has the correct authorization to access the [www.xyz.com](http://www.xyz.com) web site and  
329 the index.asp resource.

## 330 4.2 Browser/POST Profile

331 This profile uses the push model and does not rely on an artifact. The processing, in summary, is as  
332 follows:

- 333 • A user has an authenticated session on the local source site (the asserting party).
- 334 • The user wants to access a resource on the destination web site (the relying party). An HTML form is  
335 provided back to the browser from the source site. The form contains the assertion about the user. The  
336 form will also contain a button (or other type of trigger) that causes a POST of the assertion to the  
337 destination site to occur. This could also be in the form on JavaScript "auto-submit" action so that the  
338 user doesn't have to press a button.
- 339 • The destination site then can make whatever authentication and authorization decisions it needs to,  
340 based on the received assertion contained within the POST message.

As with the Browser/Artifact Profile the SAML 1.1 specifications only define this use case when use in a source-site-first situation.

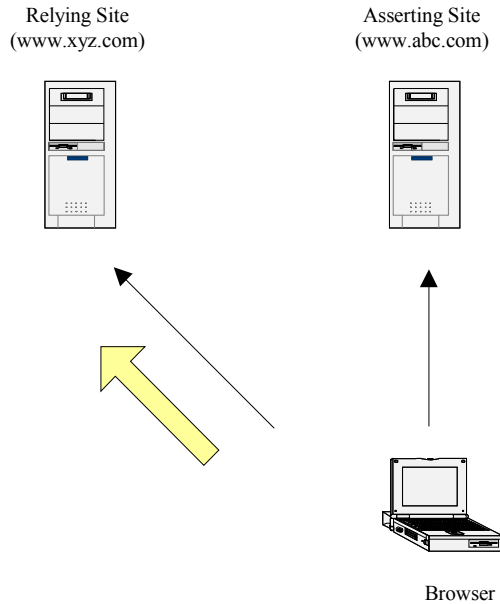


Figure 10 – Browser/POST Profile Overview

341

#### 342 4.2.1 Detailed Processing

343 Figure 11 illustrates the processing.

344

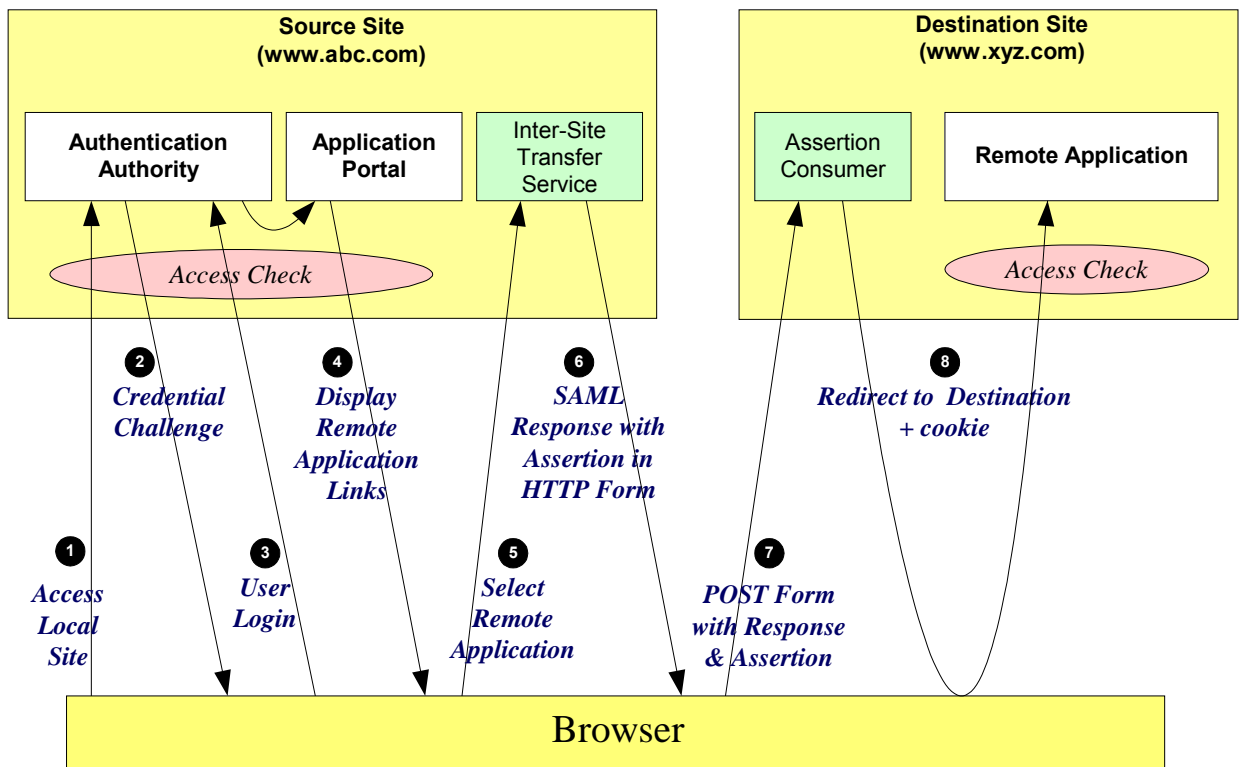


Figure 11: Browser/POST Profile – Detailed Processing

345 The processing is as follows:

346 1. The user accesses the source web site ([www.abc.com](http://www.abc.com))

- 347 2. The source web site performs an access check and determines that the user does not have a current  
348 session and requires the user to be authenticated. As a result, the user is challenged to authenticate.
- 349 3. The user supplies back credentials, for instance username and password.
- 350 4. If the authentication is successful, then a session is created for the user and the appropriate welcome  
351 screen of the Portal application is displayed to the user.
- 352 5. The user selects a menu option (or function) on the displayed screen that means the user wants to  
353 access a resource or application on a destination web site [www.xyz.com](http://www.xyz.com). The portal application then  
354 directs the request to the local Inter-site Transfer Service (in this example, hosted on the same web  
355 site). The request contains the URL of the resource on the destination site (the TARGET URL).
- 356 6. The Inter-site Transfer Service sends a HTML form back to the browser. The HTML FORM contains a  
357 SAML response, within which is a SAML assertion. The SAML specifications mandate that the  
358 response must be digitally signed. Typically the HTML FORM will contain an input or submit action that  
359 will result in a HTTP POST.
- 360 7. The browser, either due to a user action or via an "auto-submit", issues a HTTP POST containing the  
361 SAML response to be sent to the destination's (relying party) Assertion Consumer service.
- 362 8. The replying party's Assertion Consumer validates the digital signature on the SAML Response. If this  
363 validates, it sends a redirect to the browser causing it to access the TARGET resource. An access  
364 check is then made to establish whether the user has the correct authorization to access the  
365 [www.xyz.com](http://www.xyz.com) web site and the TARGET resource. The TARGET resource is the returned to the  
366 browser.

## 367 **4.3 Destination-Site-First**

368 As previously described in a number of use case scenarios the user may not initially access the asserting  
369 party. For instance, in the case of a centralized portal system, a user may first access a satellite system  
370 but is required to be authenticated centrally. This is known as "Destination-Site-First". This particular use  
371 case is not described in the Web SSO Profile, the use of TARGET from the Replying Party to the  
372 Asserting Party is just one way to process this use case. However as a number of vendors support this  
373 scenario, for completeness, the use case is described in this document.

### 374 **4.3.1 Detailed Processing for the Destination-Site-First Scenario**

375 Figure 12 illustrates the processing steps for the Browser/Artifact Profile. Processing is a variant of the  
376 previous use case.

377

378

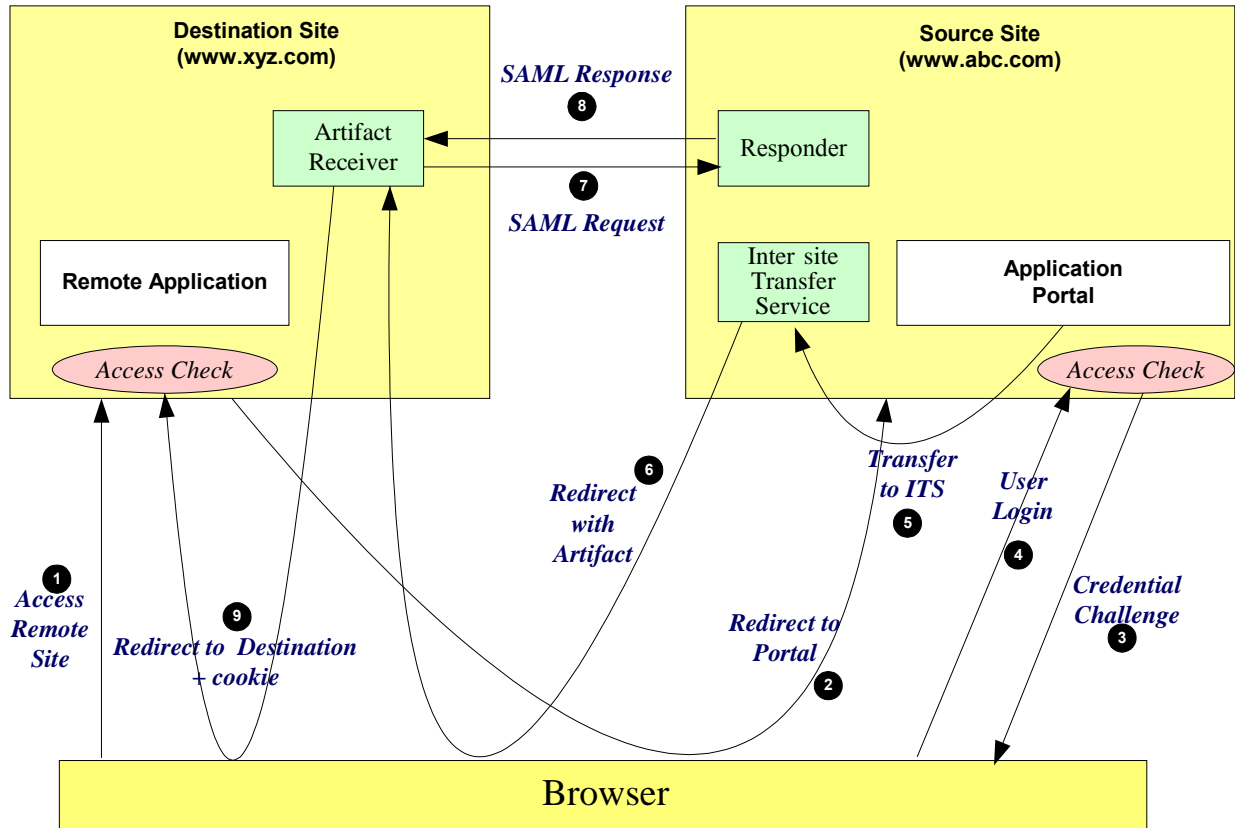


Figure 12: Browser/Artifact Profile - Destination-Site-First – Detailed Processing

379  
380

- 381 1. The user accesses the destination web site ([www.xyz.com](http://www.xyz.com)).
- 382 2. The destination web site performs an access check and determines that the user must be
- 383 authenticated by the central site (source site). A redirection is issued to the source site. Typically, this
- 384 redirection is to the central site's Inter-site Transfer Service.
- 385 3. The source site (the asserting party) challenges the user for their credentials.
- 386 4. The user supplies back credentials, for instance username and password.
- 387 5. The portal application then directs the request to the local Inter-site Transfer Service (in this example,
- 388 hosted on the same web site). The request contains the URL of the resource on the destination site
- 389 originally requested.
- 390 6. The Inter-site Transfer Service generates an assertion for the user while also creating an artifact. The
- 391 artifact contains the source ID of the [www.abc.com](http://www.abc.com) SAML responder together with a reference to the
- 392 assertion (the AssertionHandle). The Inter-site Transfer Service then sends back an HTTP redirection
- 393 response to the browser, with the HTTP location header containing the URL of the Artifact Receiver
- 394 service, the TARGET URL, and the artifact.
- 395 7. On receiving the HTTP message, the Artifact Receiver on the destination site sends a SAML request to
- 396 the [www.abc.com](http://www.abc.com) SAML responder containing the artifact supplied by the Inter-site Transfer service of
- 397 [www.abc.com](http://www.abc.com).
- 398 8. The [www.abc.com](http://www.abc.com) SAML responder supplies back a SAML response message containing the
- 399 assertion generated during step 7.
- 400 9. The Artifact Receiver, on the destination web site, sends a redirection message containing a cookie
- 401 back to the browser. The cookie identifies the session. The Browser then processes the redirect
- 402 message and issues a HTTP GET to the TARGET resource on [www.xyz.com](http://www.xyz.com) that was originally
- 403 requested in step 1.

404

## 5 Documentation Roadmap

405

406 Following is the SAML V1.1 suite of specifications, approved and published on 2 September 2003.

Short Name	Document Identifier	Description
Assertions and Protocol (also known as the "core" spec)	oasis-sstc-saml-core-1.1	Defines the syntax and semantics for XML-encoded assertions about authentication, attributes and authorization, and for the protocol that conveys this information.
Assertion schema	oasis-sstc-saml-schema-assertion-1.1	The schema document governing the formal definition of SAML's XML-form assertions.
Protocol schema	oasis-sstc-saml-schema-protocol-1.1	The schema document governing the formal definition of SAML's XML-form request and response protocol messages.
Bindings and Profiles	oasis-sstc-saml-bindings-1.1	Defines protocol bindings and profiles for the use of SAML assertions and request-response messages in communications protocols and frameworks.
Security and Privacy Considerations	oasis-sstc-saml-sec-consider-1.1	Describes and analyzes the security and privacy properties of SAML. (Note that the Bindings and Profiles specification also contains some security information pertaining to each profile.)
Conformance Program Specification	oasis-sstc-saml-conform-1.1	Describes the program and technical requirements for SAML conformance.
Glossary	oasis-sstc-saml-glossary-1.1	Defines terms used throughout the SAML specifications and related documents.

407

408 The following are other documents related to SAML V1.1.

Short Name	Document Identifier	Description
Technical Overview	sstc-saml-tech-overview-1.1	This document. It provides an overview of basic SAML goals and concepts and the flows specified in the SAML profiles.
Differences from V1.0	sstc-saml-diff-1.1-draft-01	A description of the changes made to the SAML specifications from V1.0 to V1.1.
V1.1 Errata	sstc-saml-errata-1.1-draft-16	A list of problems and resolutions kept during the public review of the SAML V1.1 Committee Specifications. Note that this is <b>not</b> a list of errata on the final SAML V1.1 specifications. <b>This is a historical document only.</b>
V1.1 Issues	sstc-saml-1.1-issues-draft-02	The list of issues from which the SSTC worked during the creation of SAML V1.1. <b>This is a historical document only.</b>

409

410 These documents can all be found at the public SAML home page:

411

[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=security](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security)

---

## A. Notices

413 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that  
414 might be claimed to pertain to the implementation or use of the technology described in this document or  
415 the extent to which any license under such rights might or might not be available; neither does it represent  
416 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to  
417 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made  
418 available for publication and any assurances of licenses to be made available, or the result of an attempt  
419 made to obtain a general license or permission for the use of such proprietary rights by implementors or  
420 users of this specification, can be obtained from the OASIS Executive Director.

421 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or  
422 other proprietary rights which may cover technology that may be required to implement this specification.  
423 Please address the information to the OASIS Executive Director.

424 **Copyright © OASIS Open 2004. All Rights Reserved.**

425 This document and translations of it may be copied and furnished to others, and derivative works that  
426 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
427 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
428 this paragraph are included on all such copies and derivative works. However, this document itself does  
429 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as  
430 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights  
431 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it  
432 into languages other than English.

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

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