



# Liberty ID-FF 1.1 Static Conformance Requirements

Version: 1.0

**Editors:**

Eric Tiffany, IEEE-ISTO

**Abstract:**

Defines the Static Conformance Requirements for the Liberty Alliance version 1.1 specifications.

**Filename:** liberty-idff-1.1-scr-v1.0.pdf

1 Notice

2 Copyright © 2003 America Online, Inc.; American Express Travel Related Services; Bank of America; Bell Canada;  
3 Cingular Wireless; Cisco Systems, Inc.; Communicator, Inc.; Deloitte & Touche LLP; Earthlink, Inc.; Electronic  
4 Data Systems, Inc.; Entrust, Inc.; Ericsson; Fidelity Investments; France Telecom; Gemplus; General Motors;  
5 Hewlett-Packard Company; i2 Technologies, Inc.; Intuit Inc.; MasterCard International; NEC Corporation; Netegrity;  
6 NeuStar; Nextel Communications; Nippon Telegraph and Telephone Corporation; Nokia Corporation; Novell, Inc.;  
7 NTT DoCoMo, Inc.; OneName Corporation; Openwave Systems Inc.; PricewaterhouseCoopers LLP; Register.com;  
8 Royal Mail; RSA Security Inc; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; SK Telecom; Sony  
9 Corporation; Sun Microsystems, Inc.; Symlabs, Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa International;  
10 Vodafone Group Plc; Wave Systems;. All rights reserved.

11 This specification document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to  
12 use the document solely for the purpose of implementing the Specification. No rights are granted to prepare  
13 derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other  
14 uses must contact the Liberty Alliance to determine whether an appropriate license for such use is available.

15 Implementation of certain elements of this Specification may require licenses under third party intellectual property  
16 rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are  
17 not, and shall not be held responsible in any manner, for identifying or failing to identify any or all such third party  
18 intellectual property rights. **This Specification is provided "AS IS", and no participant in the Liberty Alliance**  
19 **makes any warranty of any kind, express or implied, including any implied warranties of merchantability,**  
20 **non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementors  
21 of this Specification are advised to review the Liberty Alliance Project's website (<http://www.projectliberty.org/>) for  
22 information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance  
23 Management Board.

24 Liberty Alliance Project  
25 Licensing Administrator  
26 c/o IEEE-ISTO  
27 445 Hoes Lane  
28 Piscataway, NJ 08855-1331, USA  
29 info@projectliberty.org

30 **Contents**

31 [1. Overview](#) ..... 4  
32 [2. Conformance Profiles](#) ..... 5  
33 [References](#) ..... 23

## 34 1. Overview

35 Static conformance requirements (SCR) describe features that are mandatory and optional for implementations  
36 conforming to the Liberty Alliance Specifications (version 1.1). This document defines these requirements. This  
37 is a normative document with several non-normative explanatory sections.

### 38 1.1. Definitions and Motivation

39 The Liberty specifications define a large number of features and variations. Applications often do not require all  
40 the features within a specification. It is also possible that implementations may not be able to implement all the  
41 features. In these cases, it may be desirable to partition the specifications into subsets of functionality. A profile is  
42 a subset of the overall specifications that includes all of the functionality necessary to satisfy the requirements of a  
43 particular community of users. This document identifies several Liberty conformance profiles based on subsets of  
44 these specifications according to the following guidelines:

- 45 • The number of profiles should be kept small.
- 46 • The profiles should correspond to the major roles within the specifications: Identity Provider (IdP), Service  
47 Provider (SP), and Liberty Enabled Client/Proxy (LECP).
- 48 • The SCR should distinguish between software implementations and deployments. Allow deployments to option-  
49 ally configure features that are mandatory in the conformance profiles.
- 50 • The SCR should place more stringent requirements on IDPs as compared with SPs to promote interoperability.
- 51 • Implementations conforming to one (or more) profiles should be able to interoperate with another conforming  
52 implementation of a complementary profile.

53 The resulting profiles are described below by means of tables indicating mandatory and optional features, with  
54 references to the appropriate sections of the specification documents.

### 55 1.2. Notation

56 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD  
57 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in  
58 [RFC2119]: "they MUST only be used where it is actually required for interoperation or to limit behavior which has  
59 potential for causing harm (e.g., limiting retransmissions)."

60 The normative sections of this document are identified with special formatting as indicated here:

61       **EXAMPLE**     This is an example of the format of the normative requirements. Each requirement is identified  
62                       by a requirement identifier ("EXAMPLE" in this example).

## 63 2. Conformance Profiles

### 64 2.1. Profile Matrix

65 The following table summarizes the features that comprise the four profiles. The detailed specifications for each profile  
66 follow in subsequent sections

67 Table 1. Profile Matrix

Feature	IdP Profile	SP Basic	SP Complete	Com-LECP
Single Sign-On using Artifact Profile	MUST	MUST	MUST	
Single Sign-On using Browser POST Profile	MUST	MUST	MUST	
Single Sign-On using LECP Profile	MUST	MUST	MUST	MUST
Register Name Identifier - (IdP Initiated) - HTTP-Redirect	OPTIONAL	MUST	MUST	
Register Name Identifier - (IdP Initiated) - SOAP/HTTP	OPTIONAL	OPTIONAL	MUST	
Register Name Identifier - (SP Initiated) - HTTP-Redirect	MUST	MUST	MUST	
Register Name Identifier - (SP Initiated) - SOAP/HTTP	MUST	OPTIONAL	MUST	
Federation Termination Notification (IdP Initiated) – HTTP-Redirect	MUST	MUST	MUST	
Federation Termination Notification (IdP Initiated) – SOAP/HTTP	MUST	OPTIONAL	MUST	
Federation Termination Notification (SP Initiated) – HTTP-Redirect	MUST	MUST	MUST	
Federation Termination Notification (SP Initiated) – SOAP/HTTP	MUST	OPTIONAL	MUST	
Single Logout (IdP Initiated) – HTTP-Redirect	MUST	MUST	MUST	
Single Logout (IdP Initiated) – HTTP-GET	MUST	MUST	MUST	
Single Logout (IdP Initiated) – SOAP	MUST	OPTIONAL	MUST	
Single Logout (SP Initiated) – HTTP-Redirect	MUST	MUST	MUST	
Single Logout (SP Initiated) – SOAP	MUST	OPTIONAL	MUST	
Identity Provider Introduction	MUST	OPTIONAL	OPTIONAL	

### 68 2.2. Identity Provider Profile

69 This section defines the conformance requirements for an identity provider. These requirements are derived from  
70 the steps defined in the [\[LibertyBindProf1.1, Section 3\]](#), which describe the interactions between the user agent, the  
71 service provider, and the identity provider.

#### 72 2.2.1. Single Sign-On and Federation

73 This section defines how an identity provider facilitates single sign-on by processing incoming and outgoing requests.  
74 The steps indicated refer to the interaction diagram in [\[LibertyBindProf1.1, 3.2.1\]](#) which illustrates the general single  
75 sign-on framework.

##### 76 2.2.1.1. Common Interaction and Processing

77 The single sign-on requirements specified here assume that the user agent has already authenticated with the identity  
78 provider, and that a valid session exists for the user agent at the identity provider.

79 There are only two actions required of the identity provider, as noted here:

80 **IDP-SSO-001** In step 5, the identity provider MUST process the <lib:AuthnRequest> message according  
81 to the rules specified in [\[LibertyProtSchema1.1\]](#).

82     **IDP-SSO-002**   In step 6, the identity provider MUST respond to the user agent with a  
83                    <lib:AuthnResponse>, a SAML artifact, or an error. The form of this response is  
84                    contingent on the specific interaction method employed by the identity provider.

### 85   **2.2.1.2. Single Sign-on Using Browser Artifact**

86   This section describes the identity provider actions necessary to perform single sign-on using browser artifact. The  
87   requirements in this section are derived primarily from [Liberty Bindings and Profiles, Section 3.2.2](#), with additional  
88   references to Step 8 in Section 3.1.

89     **IDP-SSO-003**   Single Sign-on using Browser Artifact is a mandatory supported feature of the identity provider  
90                    conformance profile. The requirements in this section MUST be implemented according to the  
91                    the relevant sections of [\[LibertyBindProf1.1\]](#).

92   The identity provider must complete two processing steps to implement this feature: processing an authentication  
93   request, and processing a SAML assertion. The authentication interaction proceeds as follows:

94     **IDP-SSO-004**   The identity provider MUST process the <lib:AuthnRequest> message as specified in  
95                    [\[LibertyBindProf1.1, 3.2.1, Step 5\]](#).

96     **IDP-SSO-005**   In response to the <lib:AuthnRequest> the identity provider MUST perform a redirection  
97                    as specified in [\[LibertyBindProf1.1, 3.2.2.1, Step 6\]](#)

98     **IDP-SSO-006**   The identity provider MUST process the <samlp:Request> produced by the service provider  
99                    in Step 8 of the single sign-on interaction, and produce a <samlp:Response> as specified in  
100                    [\[LibertyBindProf1.1, 3.2.1, Step 9\]](#).

101     **IDP-SSO-007**   The artifact produce by the identity provider MUST be be formatted as specified in [\[Liberty-](#)  
102                    BindProf1.1, 3.2.2.2, Artifact Format].

### 103   **2.2.1.3. Single Sign-on using Liberty Browser POST**

104   This section describes the identity provider requirements for performing single sign-on using Liberty Browser POST.

105     **IDP-SSO-008**   Single Sign-on using Liberty Browser POST is a mandatory supported feature of the identity  
106                    provider conformance profile. The requirements in this section MUST be implemented  
107                    according the the relevant sections of [\[LibertyBindProf1.1\]](#).

108     **IDP-SSO-009**   The identity provider MUST process the <lib:AuthnRequest> message as specified in  
109                    [\[LibertyBindProf1.1, 3.2.1, Step 5\]](#). (Same as [\[IDP-SSO-004\]](#)).

110     **IDP-SSO-010**   The identity provider generates an HTML 200 OK response containing an authentication  
111                    request. This response MUST conform to the specification in [\[LibertyBindProf1.1, 3.2.3, Step](#)  
112                    6]

#### 113 **2.2.1.4. Single Sign-on using Liberty WML POST**

114 The Liberty WML POST is deprecated.

#### 115 **2.2.1.5. Single Sign-on using Liberty-Enabled Client and Proxy**

116 This section specifies the identity provider requirements for performing single sign-on using the Liberty-Enabled  
117 Client and Proxy (LECP) interaction.

118 **IDP-SSO-011** Single Sign-on using LECP is a mandatory supported feature of the identity provider confor-  
119 mance profile. The requirements in this section **MUST** be implemented according the the  
120 relevant sections of [[LibertyBindProf1.1](#)].

121 **IDP-SSO-012** The identity provider **MUST** process the `<lib:AuthnRequest>` in the body of the SOAP  
122 POST message from the LECP as specified in [[LibertyBindProf1.1, 3.2.1, Step 5](#)]. (See [[IDP-SSO-004](#)]).  
123

124 **IDP-SSO-013** The identity provider **MUST** respond to the `<lib:AuthnRequest>` with a HTTP 200  
125 OK response as specified in [[LibertyBindProf1.1, 3.2.5.2, Step 6](#)], with the correct MIME  
126 type (`application/vnd.liberty-response+xml`) and Liberty-Enabled HTTP header (see  
127 [[LibertyBindProf1.1, 3.2.5.1](#)])

#### 128 **2.2.2. Register Name Identifier**

129 This section specifies the required and optional features used by an identity provider to register or change a name  
130 identifier for a Principal. There are four variations of the register name identifier protocol: the name registration  
131 interaction can be initiated by either the identity provider or the service provider, and the protocol can be either HTTP-  
132 Redirect based, or SOAP/HTTP based [[LibertyBindProf1.1, 3.3](#)].

133 Note that while the name registration features are optional interactions in a *deployment*, some of these features are  
134 required for a implementation to be conformant. The following sections describe the mandatory and optional  
135 conformance requirements for an identity provider implementing the register name identifier feature.

##### 136 **2.2.2.1. Register Name Identifier Initiated at Identity Provider**

###### 137 **2.2.2.1.1. HTTP-Redirect Based**

138 This section specifies the requirements for the HTTP-Redirect-based register name identifier initiated at the identity  
139 provider. These requirements are based on the identity provider actions in the interaction described in [[Liberty-  
140 BindProf1.1, 3.3.1.1](#)]. Note that the timing and mechanism of the initiation of this interaction are not normatively  
141 specified, although the preceding specification reference offers some examples.

142 **IDP-RNI-001** The identity provider **MUST** redirect the user agent to the register name identifier service at the  
143 service provider as specified in [[LibertyBindProf1.1, 3.3.1.1, Step 2](#)].

###### 144 **2.2.2.1.2. SOAP/HTTP Based**

145 **IDP-RNI-002** The identity provider **MUST** only initiate SOAP/HTTP-based register name identifier when the  
146 service provider metadata specifies the appropriate URI identifier as specified in [[LibertyBind-  
147 Prof1.1, 3.3.1.2](#)].

148 **IDP-RNI-003** The SOAP/HTTP-based Register Name Identifier transactions must use the SOAP binding for  
149 Liberty as defined in [LibertyBindProf1.1, 2.1].

150 **IDP-RNI-004** The identity provider MUST initiate the Register Name Identifier transaction by sending a  
151 <lib:RegisterNameIdentifierRequest> message to the service provider's SOAP end-  
152 point as specified in [LibertyBindProf1.1, 3.3.1.2, Step 1].

153 **IDP-RNI-005** The identity provider MUST process the <lib:RegisterNameIdentifierResponse> from  
154 the service provider as specified in [LibertyProtSchema1.1, 3.3.3].

#### 155 **2.2.2.2. Register Name Identifier Initiated at Service Provider**

156 The following sections describe the interactions for an identity provider implementing the Register Name Identifier  
157 initiated at the service provider.

158 **IDP-RNI-006** Register Name Identifier initiated at service provider is a REQUIRED feature of the Identity  
159 Provider Conformance Profile. Both the HTTP-Redirect and the SOAP/HTTP interaction  
160 MUST be implemented.

161 Note that this section refers to [LibertyBindProf1.1] Register Name Identifier interactions initiated at the identity  
162 provider. The steps associated with the service provider in those interactions are used here as if they were associated  
163 with the identity provider. All references to to service provider and identity provider have been interchanged as  
164 indicated in [LibertyBindProf1.1, 3.3.2.1 ] and [LibertyBindProf1.1, 3.3.2.2].

##### 165 **2.2.2.2.1. HTTP-Redirect Based**

166 **IDP-RNI-007** The identity provider MUST process the <lib:RegisterNameIdentifierRequest> from  
167 the service provider as specified in [LibertyProtSchema1.1, 3.3.3]. See [LibertyBindProf1.1,  
168 3.3.1.1, Step 4].

169 **IDP-RNI-008** The identity provider MUST respond to the service provider with a redirection URL as specified  
170 in the RegisterNameIdentifierServiceReturnURL metadata element. The redirection  
171 MUST adhere to the rules specified in [LibertyBindProf1.1, 3.3.1.1, Step 5].

##### 172 **2.2.2.2.2. SOAP/HTTP Based**

173 [IDP-RNI-003] is a requirement for this interaction.

174 **IDP-RNI-009** The service provider will send a <lib:RegisterNameIdentifierRequest> proto-  
175 col message to the identity provider. The identity provider MUST record the new  
176 <lib:SPProvidedNameIdentifier>.

177 **IDP-RNI-010** After a successful registration of the <lib:SPProvidedNameIdentifier>, the identity  
178 provider MUST respond with a <lib:RegisterNameIdentifierResponse> according to  
179 the processing rules in [LibertyProtSchema1.1, 3.3.3].



## 180 **2.2.3. Identity Federation Termination Notification**

181 Liberty identity federation termination notification specifies how service providers and identity providers are notified  
182 of federation termination. There are four variations of the federation termination notification interaction: the  
183 federation termination notification interaction can be initiated by either the identity provider or the service provider,  
184 and the protocol can be based on either HTTP-Redirect or SOAP/HTTP.

185 **IDP-FTN-001** This section specifies the conformance requirements for an identity provider to support identity  
186 federation termination notification. All four interactions specified in [LibertyBindProf1.1, 3.4]  
187 MUST be implemented.

### 188 **2.2.3.1. Federation Termination Notification Initiated at the Identity Provider**

#### 189 **2.2.3.1.1. HTTP-Redirect**

190 **IDP-FTN-002** This interaction MUST NOT be used unless the service provider metadata ele-  
191 ment `FederationTerminationNotificationProtocolProfile` specifies the URI  
192 *http://projectliberty.org/profiles/fedterm-idp-http* .

193 **IDP-FTN-003** This interaction REQUIRES certain preconditions specified in [LibertyBindProf1.1, 3.4.1.1]  
194 are met.

195 **IDP-FTN-004** In response to a request to the identity provider's federation terminaiton service URL, the  
196 identity provider MUST redirect the user agent to the federation termination service at the  
197 service provider. This redirection MUST adhere to the rules specified in [LibertyBindProf1.1,  
198 3.4.1.1, Step 2].

#### 199 **2.2.3.1.2. SOAP/HTTP**

200 **IDP-FTN-005** This interaction MUST NOT be used unless the service provider metadata ele-  
201 ment `FederationTerminationNotificationProtocolProfile` specifies the URI  
202 *http://projectliberty.org/profiles/fedterm-idp-soap* .

203 **IDP-FTN-006** This interaction REQUIRES certain preconditions specified in [LibertyBindProf1.1, 3.4.1.2]  
204 are met.

205 **IDP-FTN-007** In response to a request from the user agent to the identity provider's federation termination  
206 service URL, the identity provider MUST send an asynchronous SOAP over HTTP notification  
207 message to the service provider's SOAP endpoint. The SOAP message MUST adhere to the  
208 rules specified in [LibertyBindProf1.1, 3.4.1.2, Step 2].

209 The service provider will respond to termination notification with a HTTP 204 No Content response.

210 **IDP-FTN-008** The identity provider MUST process the HTTP 204 No Content response from the service  
211 provider and send an HTTP response confirming the requested action of federation termination  
212 with the specified service provider.

## 213 **2.2.3.2. Federation Termination Initiated at the Service Provider**

214 The following sections describe the interactions for an identity provider implementing federation termination notifica-  
215 tion initiated at the service provider.

216 Note that this section refers to [LibertyBindProf1.1] federation termination notifications interactions initiated at the  
217 identity provider. The steps associated with the service provider in those interactions are used here as if they  
218 were associated with the identity provider. All references to to service provider and identity provider have been  
219 interchanged as indicated in [LibertyBindProf1.1, 3.4.2.1 ] and [LibertyBindProf1.1, 3.4.2.2].

### 220 **2.2.3.2.1. HTTP-Redirect**

221 **IDP-FTN-009** The identity provider MUST process the `<lib:FederationTerminationNotification>`  
222 received from the user agent according to the rules defined in [LibertyProtSchema1.1, 3.4.2]  
223 and in [LibertyBindProf1.1, 3.4.1.1, Step 4].

224 **IDP-FTN-010** The identity provider's federation termination service MUST respond by redirecting the user  
225 agent as specified in [LibertyBindProf1.1, 3.4.1.1, Step 5].

### 226 **2.2.3.2.2. SOAP/HTTP**

227 **IDP-FTN-011** The identity provider MUST process the `<lib:FederationTerminationNotification>`  
228 in the SOAP message received from the service provider according to the rules defined in  
229 [LibertyProtSchema1.1, 3.4.2] and in [LibertyBindProf1.1, 3.4.1.2, Step 3].

230 **IDP-FTN-012** The identity provider MUST respond to the `<lib:FederationTerminationNotification>`  
231 with a HTTP 204 OK response [LibertyBindProf1.1, 3.4.1.2, Step 4].

## 232 **2.2.4. Single Logout**

233 Liberty single logout specifies how service providers and identity providers synchronize logout across all sessions  
234 authenticated by a particular identity provider. There are five variations of the single logout interaction: the single  
235 logout can be initiated by either the identity provider or the service provider, and the protocol can be based on either  
236 HTTP-Redirect, HTTP-GET (only when initiated at the identity provider), or SOAP/HTTP.

237 **IDP-SLO-001** This section specifies the conformance requirements for an identity provider to support Single  
238 Logout. All five interactions specified in [LibertyBindProf1.1, 3.5] MUST be implemented.

### 239 **2.2.4.1. Single Logout Initiated at the Identity Provider**

240 The following sections specify the requirements for single logout when initiated by a user agent at the identity provider.

#### 241 **2.2.4.1.1. HTTP-Redirect**

242 **IDP-SLO-002** This interaction MUST NOT be used unless the service provider metadata element  
243 `SingleLogoutProtocolProfile` specifies the URI `http://projectliberty.org/profiles/slo-idp-`  
244 `http`.

245     **IDP-SLO-003**     In response to the user agent request, the identity provider **MUST** redirect the user agent to  
246     the single logout service URL at each service provider for which the identity provider has  
247     provided an authentication assertion during the Principal's current session. Each redirection  
248     **MUST** adhere to the rules specified in [[LibertyBindProf1.1, 3.5.1.1.1, Step 2](#)].

249     **IDP-SLO-004**     After receiving the request from the user agent to the `SingleLogoutServiceReturnURL` as  
250     specified in the identity provider metadata, the identity provider **MUST** process the request and  
251     send an HTTP response to the user agent confirming that the requested action of a single logout  
252     has been completed.

#### 253   **2.2.4.1.2. HTTP-GET**

254     **IDP-SLO-005**     This interaction **MUST NOT** be used unless the service provider metadata element  
255     `SingleLogoutProtocolProfile` specifies the URI *http://projectliberty.org/profiles/slo-idp-*  
256     *http*.

257     **IDP-SLO-006**     In response to the user agent request, the identity provider **MUST** respond with a standard  
258     HTTP 200 OK response containing image tags referencing the logout service URL for each of  
259     the service providers for which the identity provider has provided an authentication assertion  
260     during the Principal's current session. Each image tag **MUST** adhere to the rules specified in  
261     [[LibertyBindProf1.1, 3.5.1.1.2, Step 2](#)].

262     **IDP-SLO-007**     After receiving the request from the user agent to the `SingleLogoutServiceReturnURL` as  
263     specified in the identity provider metadata, the identity provider **MUST** process the request and  
264     send an HTTP response to the user agent confirming that the requested action of a single logout  
265     has been completed.

#### 266   **2.2.4.1.3. SOAP/HTTP**

267     **IDP-SLO-008**     This interaction **MUST NOT** be used unless the service provider metadata element  
268     `SingleLogoutProtocolProfile` specifies the URI *http://projectliberty.org/profiles/slo-idp-*  
269     *soap* .

270     **IDP-SLO-009**     In response to a SOAP 200 OK `<lib:LogoutRequest>` message from the service provider,  
271     the identity provider **MUST** send an HTTP response confirming the requested action of single  
272     logout has completed.

#### 273   **2.2.4.2. Single Logout Initiated at the Service Provider**

##### 274   **2.2.4.2.1. HTTP-Redirect**

275     **IDP-SLO-010**     The user agent will access the identity provider's single logout service URL. The identity  
276     provider **MUST** process the `<lib:LogoutRequest>` according to the rules defined in [[Lib-  
277     ertyProtSchema1.1, 3.5.1](#)]

278     **IDP-SLO-011**     The identity provider **MUST** notify each service provider for which the identity provider has  
279     provided authentication assertions of the logout request via the service provider's preferred  
280     profile [[LibertyBindProf1.1, 3.5.2.1, Step 4](#)].

281     **IDP-SLO-012**     The identity provider **MUST** terminate the Principal's current session, and no more authentica-  
282     tion assertions for the Principal are to be given to service providers.

283     **IDP-SLO-013**     The identity provider **MUST** respond and redirect the user agent back to the service provider us-  
284     ing the return URL location obtained from the `SingleLogoutServiceReturnURL` metadata  
285     element as specified in [[LibertyBindProf1.1, 3.5.2.1, Step 5](#)].

#### 286   **2.2.4.2.2. SOAP/HTTP**

287     **IDP-SLO-014**     After receiving a `<lib:LogoutRequest>` from the service provider, the identity provider  
288     **MUST** process it according to the rules in [[LibertyProtSchema1.1, 3.5.1](#)].

289     **IDP-SLO-015**     The identity provider **MUST** submit to each service provider for which the identity provider has  
290     provided authentication assertions during the Principal's current session a request to logout the  
291     Principal as specified in [[LibertyBindProf1.1, 3.5.2.2, Step 3](#)].

292     **IDP-SLO-016**     The identity provider **MUST** respond to the `<lib:LogoutRequest>` with a SOAP 200 OK  
293     `<lib:LogoutResponse>` message [[LibertyBindProf1.1, 3.5.2.2, Step 4](#)].

### 294   **2.2.5. Identity Provider Introduction**

295   This section describes the conformance requirements for an identity provider implementing the identity provider  
296   introduction feature. The identity provider introduction feature is intended to allow service providers to discover  
297   which identity providers a Principal is using.

298     **IDP-IPI-001**     The identity provider introduction feature is a **REQUIRED** element of identity provider confor-  
299     mance profile.

300   Although a deployment may choose not to enable the identity provider introduction feature, an identity provider  
301   implementation must provide the feature in order to be conformant.

#### 302   **2.2.5.1. Common Domain Cookie**

303   The identity provider introduction relies on the use of a common domain cookie.

304     **IDP-IPI-002**     The common domain cookie **MUST** be constructed as specified in [[LibertyBindProf1.1, 3.6.1](#)].

#### 305   **2.2.5.2. Setting the Common Domain Cookie**

306   Creating and updating the common domain cookie is the responsibility of the identity provider.

307     **IDP-IPI-003**     After authenticating a Principal, the identity provider **MUST** attempt to set the common domain  
308     cookie, subject to cookie-setting restrictions of the user-agent.

309 The details of this procedure are implementation-dependent, and are not normatively specified. However, one possible  
310 strategy is described in [[LibertyBindProf1.1, 3.6.2](#)].

## 311 **2.3. Liberty Enabled Client/Proxy (LECP)**

312 This section contains detailed specifications of the LECP Profile.

### 313 **2.3.1. General LECP Requirements**

314 **LCP-SSO-001** All HTTP requests made by a LECP MUST include a Liberty-Enabled indication. A Liberty-  
315 Enabled indication is either a Liberty-Enabled header or User-Agent header containing a  
316 Liberty-Enabled value as defined in [[LibertyBindProf1.1, 3.2.5.1](#)].

317 The preferred Liberty-Enabled indication is the Liberty-Enabled header.

318 **LCP-SSO-002** A LECP SHOULD add the Liberty-Enabled header to each HTTP request. This header MUST  
319 be constructed as specified in [[LibertyBindProf1.1, 3.2.5.1](#)].

320 **LCP-SSO-003** A LECP MAY add a Liberty-Enabled entry in the HTTP User-Agent request header, as specified  
321 in [[LibertyBindProf1.1, 3.2.5.1](#)]

### 322 **2.3.2. Single Sign-On**

323 The single sign-on interaction is the only LECP interaction specified. This interaction assumes that the user agent has  
324 authenticated at the identity provider and that a valid session exists for the user agent at the identity provider.

325 **LCP-SSO-004** To initiate single sign-on, the user agent MUST contain at most one Liberty-Enabled header.  
326 If a proxy receives a HTTP request that contains a Liberty-Enabled header, it MUST NOT add  
327 another Liberty-Enabled header.

328 **LCP-SSO-005** A proxy MAY replace the Liberty-Enabled header, but this replacement MUST adhere to the  
329 specifications in [[LibertyBindProf1.1, 3.2.5.2, Step 1](#)]

330 After receiving the HTTP 200 OK response (containing a `<lib:AuthnRequestEnvelope>`) from the service provider,  
331 the LECP will determine the correct identity provider to use.

332 **LCP-SSO-006** The LECP MUST issue an HTTP POST of the `<lib:AuthnRequest>` in the body  
333 of a SOAP message to the identity provider's single sign-on service URL. This  
334 MUST be the same `<lib:AuthnRequest>` as was received in the service provider's  
335 `<lib:AuthnRequestEnvelope>`. See [[LibertyBindProf1.1, 3.2.5.2, Step 4](#)].

336 **LCP-SSO-007** In case of any error, the LECP MUST return a `<lib:AuthnResponse>` to the service provider  
337 as specified in [[LibertyBindProf1.1, 3.2.5.2, Step 4](#)].

338 After receiving a HTTP response from the identity provider, the LECP will issue an HTTP POST to the service  
339 provider.

340 **LCP-SSO-008** The HTTP POST from the LECP MUST be composed as specified in [[LibertyBindProf1.1,](#)  
341 [3.2.5.2, Step 7](#)].

342 **LCP-SSO-009** In case of any error, the LECP MUST return a `<lib:AuthnResponse>` to the service provider  
343 as specified in [[LibertyBindProf1.1, 3.2.5.2, Step 7](#)].

## 344 **2.4. Service Provider Basic Profile**

345 This section defines the minimal conformance requirements for a Service Provider. These requirements are derived  
346 from the steps defined in the [[LibertyBindProf1.1, Section 3](#)], which describe the interactions between the user agent,  
347 the service provider, and the identity provider.

### 348 **2.4.1. Single Sign-On and Federation**

#### 349 **2.4.1.1. Common Interaction and Processing**

350 The single sign-on requirements specified here assume that the user agent has already authenticated with the identity  
351 provider, and that a valid session exists for the user agent at the identity provider.

#### 352 **2.4.1.2. Single Sign-on using Browser Artifact**

353 This section describes the service provider actions necessary to perform single sign-on using browser artifact.

354 **SP-SSO-001** Single sign-on using browser artifact is a mandatory feature of the service provider basic  
355 conformance profile. The requirements in this section MUST be implemented as specified  
356 in [[LibertyBindProf1.1, Section 3.2.2](#)] and [[LibertyBindProf1.1, 3.2.1, Steps 8 and 10](#)].

357 The user agent initiates the single sign-on by making an HTTP request to the service provider, as indicated in  
358 [[LibertyBindProf1.1, 3.2.1, Step 1](#)]. The service provider then obtains the address of the appropriate identity provider  
359 in an implementation-dependent way (not normatively specified).

360 **SP-SSO-002** The service provider MAY use the Liberty identity provider introduction interaction while  
361 processing the user agent request.

362 After determining the appropriate identity provider location, the service provider then responds to the user agent with  
363 an HTTP redirect.

364 **SP-SSO-003** The HTTP redirection response MUST adhere to the specifications in [[LibertyBindProf1.1,](#)  
365 [3.2.2.1, Step 3](#)].

366 After obtaining an a SAML assertion from the identity provider, the user agent accesses the assertion consumer service  
367 at the service provider.

368 **SP-SSO-004** The service provider MUST send a `<samlp:Request>` SOAP message to the identity  
369 provider's SOAP endpoint as specified in [[LibertyBindProf1.1, 3.2.2.1, Step 8](#)]

370 **SP-SSO-005** The service provider MUST process the `<saml:Assertion>` returned by the identity provider  
371 as specified in [[LibertyBindProf1.1, 3.2.2.1, Step 10](#)]

### 372 **2.4.1.3. Single Sign-on using Liberty Browser POST**

373 This section describes the service provider requirements for performing single sign-on using Liberty browser POST.

374 **SP-SSO-006** Single sign-on using Liberty browser POST is a mandatory supported feature of the service  
375 provider Basic conformance profile. The requirements in this section **MUST** be implemented  
376 according the the relevant sections of [[LibertyBindProf1.1, 3.2.3](#)].

377 The user agent initiates the single sign-on by making an HTTP request to the service provider's intersite transfer service  
378 as indicated in [[LibertyBindProf1.1, 3.2.1, Step 1](#)]. The service provider then obtains the address of the appropriate  
379 identity provider in an implementation-dependent way (not normatively specified).

380 The service provider's responds to the user agent request by redirecting the user agent to the single sign-on service  
381 URL at the identity provider.

382 **SP-SSO-007** The redirection **MUST** adhere to the rules specified in [[LibertyBindProf1.1, 3.2.3, Step 3](#)].

383 After obtaining an authentication assertion, the user agent will issue an HTTP POST request containing the  
384 `<lib:AuthnResponse>` to the service provider.

385 **SP-SSO-008** The service provider **MUST** process the `<lib:AuthnResponse>` in the HTTP POST from the  
386 user agent as specified in [[LibertyBindProf1.1, 3.2.2.1, Step 10](#)]

### 387 **2.4.1.4. Single Sign-on using Liberty WML POST**

388 The Liberty WML POST is deprecated.

### 389 **2.4.1.5. Single Sign-on using Liberty-Enabled Client and Proxy**

390 This section specifies the service provider requirements for performing single sign-on using the Liberty-Enabled Client  
391 and Proxy (LECP) interaction.

392 **SP-SSO-009** Single sign-on using LECP is a mandatory supported feature of the service provider basic  
393 conformance profile. The requirements in this section **MUST** be implemented according to  
394 [[LibertyBindProf1.1, 3.2.5](#)].

395 The user agent will submit request to the service provider which contains the requisite Liberty-Enabled indications.

396 **SP-SSO-010** The service provider **MUST NOT** obtain an identity provider address or perform identity  
397 provider introduction.

398 **SP-SSO-011** The service provider **MUST** issue an HTTP 200 OK response to the user agent. The response  
399 **MUST** adhere to the specifications of [[LibertyBindProf1.1, 3.2.5.2, Step 3](#)].

400 **SP-SSO-012** If the service provider does not support Liberty version advertised by the LECP, the service  
401 provider **MUST** return an HTTP 501 Not Implemented response to the LECP with the reason  
402 phrase "Unsupported Liberty Version."

403 **SP-SSO-013** The service provider **SHOULD** place appropriate headers in the response to ensure the response  
404 is not cached as specified in [[LibertyBindProf1.1, 3.2.5.2, Step 3](#)].

405 After obtaining an authentication assertion, the user agent will issue an HTTP POST request containing the  
406 <lib:AuthnResponse> to the service provider.

407 **SP-SSO-014** The service provider MUST process the <lib:AuthnResponse> in the HTTP POST from the  
408 user agent as specified in [LibertyBindProf1.1, 3.2.2.1, Step 10]

## 409 **2.4.2. Register Name Identifier**

### 410 **2.4.2.1. Register Name Identifier Initiated at Identity Provider**

411 The following sections describe the interactions for a service provider implementing the register name identifier  
412 initiated at the identity provider.

#### 413 **2.4.2.1.1. HTTP-Redirect Based**

414 **SP-RNI-001** The HTTP-Redirect based register name identifier (initiated at the identity provider) is a  
415 REQUIRED feature of the service provider basic profile.

416 **SP-RNI-002** The service provider MUST process the <lib:RegisterNameIdentifierRequest> from  
417 the identity provider as specified in [LibertyProtSchema1.1, 3.3.3]. See [LibertyBindProf1.1,  
418 3.3.1.1, Step 4].

419 **SP-RNI-003** The service provider MUST respond to the identity provider with a redirection URL as specified  
420 in the RegisterNameIdentifierServiceReturnURL metadata element. The redirection  
421 MUST adhere to the rules specified in [LibertyBindProf1.1, 3.3.1.1, Step 5].

#### 422 **2.4.2.1.2. SOAP/HTTP Based**

423 **SP-RNI-004** The SOAP/HTTP based register name identifier (initiated at the identity provider) is an OP-  
424 TIONAL feature of the service provider basic profile.

425 **SP-RNI-005** The SOAP/HTTP-based register name identifier transactions MUST use the SOAP binding for  
426 Liberty as defined in [LibertyBindProf1.1, 2.1].

427 **SP-RNI-006** The identity provider will send a <lib:RegisterNameIdentifierRequest> pro-  
428 tocol message to the service provider. The service provider MUST record the new  
429 <lib:IDPProvidedNameIdentifier>.

430 **SP-RNI-007** After a successful registration of the <lib:IDPProvidedNameIdentifier>, the service  
431 provider MUST respond with a <lib:RegisterNameIdentifierResponse> according to  
432 the processing rules in [LibertyProtSchema1.1, 3.3.3].



### 433 **2.4.2.2. Register Name Identifier Initiated at Service Provider**

434 Note that this section refers to [\[LibertyBindProf1.1\]](#) register name identifier interactions initiated at the identity  
435 provider. The steps associated with the service provider in those interactions are used here as if they were associated  
436 with the identity provider. All references to to service provider and identity provider have been interchanged as  
437 indicated in [\[LibertyBindProf1.1, 3.3.2.2\]](#).

#### 438 **2.4.2.2.1. HTTP-Redirect Based**

439 **SP-RNI-008** The HTTP-Redirect based register name identifier (initiated at the service provider) is a  
440 REQUIRED feature of the service provider basic profile.

441 The service provider can initiate the register name identifier interaction, though the circumstances of this initiation are  
442 not normatively specified.

443 **SP-RNI-009** The service provider MUST redirect the user agent to the register name identifier service at the  
444 identity provider as specified in [\[LibertyBindProf1.1, 3.3.1.1, Step 2\]](#).

#### 445 **2.4.2.2.2. SOAP/HTTP Based**

446 **SP-RNI-010** The SOAP/HTTP based register name identifier (initiated at the service provider) is an OP-  
447 TIONAL feature of the service provider basic profile.

448 **SP-RNI-011** The service provider MUST only initiate SOAP/HTTP-based register name identifier when the  
449 identity provider metadata specifies the appropriate URI identifier as specified in [\[LibertyBind-  
450 Prof1.1, 3.3.1.2\]](#).

451 **SP-RNI-012** The SOAP/HTTP-based register name identifier transactions MUST use the SOAP binding for  
452 Liberty as defined in [\[LibertyBindProf1.1, 2.1\]](#).

453 **SP-RNI-013** The service provider MUST initiate the register name identifier transaction by sending a  
454 `<lib:RegisterNameIdentifierRequest>` message to the identity provider's SOAP end-  
455 point as specified in [\[LibertyBindProf1.1, 3.3.1.2, Step 1\]](#).

456 **SP-RNI-014** The service provider MUST process the `<lib:RegisterNameIdentifierResponse>` from  
457 the identity provider as specified in [\[LibertyProtSchema1.1, 3.3.3\]](#).

### 458 **2.4.3. Identity Federation Termination Notification**

459 Liberty identity federation termination notification specifies how service providers and identity providers are notified  
460 of federation termination. There are four variations of the federation termination notification interaction: the  
461 federation termination notification interaction can be initiated by either the identity provider or the service provider,  
462 and the protocol can be based on either HTTP-Redirect or SOAP/HTTP.

#### 463 **2.4.3.1. Federation Termination Notification Initiated at the Identity Provider**

##### 464 **2.4.3.1.1. HTTP-Redirect**

465 **SP-FTN-001** The HTTP-Redirect based federation termination notification (initiated at the identity provider)  
466 is a REQUIRED feature of the service provider basic profile.

467 **SP-FTN-002** The service provider **MUST** process the `<lib:FederationTerminationNotification>`  
468 received from the user agent according to the rules defined in [[LibertyProtSchema1.1, 3.4.2](#)]  
469 and in [[LibertyBindProf1.1, 3.4.1.1, Step 4](#)].

470 **SP-FTN-003** The service provider's federation termination service **MUST** respond by redirecting the user  
471 agent as specified in [[LibertyBindProf1.1, 3.4.1.1, Step 5](#)].

#### 472 **2.4.3.1.2. SOAP/HTTP**

473 **SP-FTN-004** The SOAP/HTTP based federation termination notification (initiated at the identity provider) is  
474 an **OPTIONAL** feature of the service provider basic profile.

475 **SP-FTN-005** The service provider **MUST** process the `<lib:FederationTerminationNotification>`  
476 in the SOAP message received from the identity provider according to the rules defined in  
477 [[LibertyProtSchema1.1, 3.4.2](#)] and in [[LibertyBindProf1.1, 3.4.1.2, Step 3](#)].

478 **SP-FTN-006** The service provider **MUST** respond to the `<lib:FederationTerminationNotification>`  
479 with a HTTP 204 OK response [[LibertyBindProf1.1, 3.4.1.2, Step 4](#)].

#### 480 **2.4.3.2. Federation Termination Initiated at the Service Provider**

##### 481 **2.4.3.2.1. HTTP-Redirect**

482 **SP-FTN-007** The HTTP-Redirect based federation termination notification (initiated at the service provider)  
483 is a **REQUIRED** feature of the service provider basic profile.

484 **SP-FTN-008** This interaction **MUST NOT** be used unless the identity provider metadata ele-  
485 ment `FederationTerminationNotificationProtocolProfile` specifies the URI  
486 *<http://projectliberty.org/profiles/fedterm-sp-http>*.

487 **SP-FTN-009** This interaction **REQUIRES** certain preconditions specified in [[LibertyBindProf1.1, 3.4.1.1](#)]  
488 are met.

489 **SP-FTN-010** In response to a request to the service provider's federation terminaiton service URL, the service  
490 provider **MUST** redirect the user agent to the federation termination service at the identity  
491 provider. This redirection **MUST** adhere to the rules specified in [[LibertyBindProf1.1, 3.4.1.1,](#)  
492 [Step 2](#)].

##### 493 **2.4.3.2.2. SOAP/HTTP**

494 **SP-FTN-011** The SOAP/HTTP based federation termination notification (initiated at the service provider) is  
495 an **OPTIONAL** feature of the service provider basic profile.

- 496 **SP-FTN-012** This interaction **MUST NOT** be used unless the identity provider metadata ele-  
497 ment `FederationTerminationNotificationProtocolProfile` specifies the URI  
498 *http://projectliberty.org/profiles/fedterm-sp-soap* .
- 499 **SP-FTN-013** This interaction **REQUIRES** certain preconditions specified in [[LibertyBindProf1.1, 3.4.1.2](#)]  
500 are met.
- 501 **SP-FTN-014** In response to a request from the user agent to the service provider's federation termination  
502 service URL, the service provider **MUST** send an asynchronous SOAP over HTTP notification  
503 message to the identity provider's SOAP endpoint. The SOAP message **MUST** adhere to the  
504 rules specified in [[LibertyBindProf1.1, 3.4.1.2, Step 2](#)].
- 505 The identity provider will respond to termination notification with a HTTP 204 No Content response.
- 506 **SP-FTN-015** The service provider **MUST** process the HTTP 204 No Content response from the identity  
507 provider and send an HTTP response confirming the requested action of federation termination  
508 with the specified identity provider.

#### 509 **2.4.4. Single Logout**

- 510 Liberty single logout specifies how service providers and identity providers synchronize logout across all sessions  
511 authenticated by a particular identity provider. There are five variations of the single logout interaction: the single  
512 logout can be initiated by either the identity provider or the service provider, and the protocol can be based on either  
513 HTTP-Redirect, HTTP-GET (only when initiated at the identity provider), or SOAP/HTTP.
- 514 Note that Single Logout, in the general case, is an iterative process from the perspective of an identity provider since  
515 the identity provider must contact each service provider to which it has issued authentication assertions. However,  
516 for a service provider, the single logout interaction is a single event.

##### 517 **2.4.4.1. Single Logout Initiated at the Identity Provider**

###### 518 **2.4.4.1.1. HTTP-Redirect**

- 519 **SP-SLO-001** The HTTP-Redirect based single logout interaction (initiated at the identity provider) is a  
520 **REQUIRED** feature of the service provider basic profile.
- 521 **SP-SLO-002** The user agent will access the service provider's single logout service URL via a redirect  
522 from the identity provider. The service provider **MUST** process the `<lib:LogoutRequest>`  
523 according to the rules defined in [[LibertyProtSchema1.1, 3.5.1](#)]
- 524 **SP-SLO-003** The service provider **MUST** invalidate the session(s) of the Principal referred to in the name  
525 identifier received from the identity provider in the `<lib:LogoutRequest>`.
- 526 **SP-SLO-004** The service provider **MUST** respond and redirect the user agent back to the identity provider us-  
527 ing the return URL location obtained from the `SingleLogoutServiceReturnURL` metadata  
528 element as specified in [[LibertyBindProf1.1, 3.5.1.1.1, Step 5](#)].

529 **2.4.4.1.2. HTTP-GET**

530 **SP-SLO-005** The HTTP-GET based single logout interaction (initiated at the identity provider) is a RE-  
531 QUIRED feature of the service provider basic profile.

532 The user agent will access the single logout service URL of the service provide as a result of an image tag load  
533 generated by the identity provider.

534 **SP-SLO-006** The service provider MUST process the `<lib:LogoutRequest>` according to the rules  
535 defined in [[LibertyProtSchema1.1, 3.5.1](#)]

536 **SP-SLO-007** The service provider MUST invalidate the session(s) of the Principal referred to in the name  
537 identifier received from the identity provider in the `<lib:LogoutRequest>`.

538 **SP-SLO-008** The service provider MUST respond and redirect the user agent image load back to the identity  
539 provider's logout completion URL obtained from the `SingleLogoutServiceReturnURL`  
540 metadata element. The HTTP response MUST be formed as specified in [[LibertyBindProf1.1, 3.5.1.1.2, Step 5](#)].  
541

542 **2.4.4.1.3. SOAP/HTTP**

543 **SP-SLO-009** The SOAP/HTTP based single logout interaction (initiated at the identity provider) is an  
544 OPTIONAL feature of the service provider basic profile.

545 **SP-SLO-010** After receiving a `<lib:LogoutRequest>` from the identity provider, the service provider  
546 MUST process it according to the rules in [[LibertyProtSchema1.1, 3.5.1](#)].

547 **SP-SLO-011** The service provider MUST invalidate the session(s) of the Principal referred to in the name  
548 identifier received from the identity provider in the `<lib:LogoutRequest>`.

549 **SP-SLO-012** The service provider MUST respond to the `<lib:LogoutRequest>` with a SOAP 200 OK  
550 `<lib:LogoutResponse>` message [[LibertyBindProf1.1, 3.5.1.2, Step 4](#)].

551 **2.4.4.2. Single Logout Initiated at the Service Provider**

552 **2.4.4.2.1. HTTP-Redirect**

553 **SP-SLO-013** The HTTP-Redirect based single logout interaction (initiated at the service provider) is a  
554 REQUIRED feature of the service provider basic profile.

555 The user agent will access the single logout service URL at the service provider.

556 **SP-SLO-014** The service provider's single logout service responds and redirects the user agent to the single  
557 logout service at the identity provider. The HTTP redirect MUST adhere to the rules specified  
558 in [[LibertyBindProf1.1, 3.5.2.1, Step 2](#)].

559 After the identity provider has processed the single logout request and contacted the appropriate service providers, the  
560 user agent will be redirected back to the service provider contacted originally.

561 **SP-SLO-015** The service provider SHOULD send an HTTP 200 OK response to the user agent with  
562 confirmation of the logout.

#### 563 **2.4.4.2.2. SOAP/HTTP**

564 **SP-SLO-016** The SOAP/HTTP based single logout interaction (initiated at the service provider) is an  
565 OPTIONAL feature of the service provider basic profile.

566 The user agent will initiate single logout by accessing the single logout service URL at the service provider via an  
567 HTTP request.

568 **SP-SLO-017** In response to the single logout request, the service provider sends a SOAP over HTTP request  
569 to the identity provider's SOAP endpoint. The SOAP request MUST be constructed and  
570 processed as specified in [[LibertyBindProf1.1, 3.5.2.2, Step 2](#)].

571 The identity provider will contact all service providers to which it has issued assertions for the Principal to request a  
572 logout action. The identity provider may determine that one or more of the service providers do not support the SOAP  
573 single logout interaction. The identity provider will return a `<lib:LogoutResponse>` containing a status code of  
574 `<lib:UnsupportedProfile>`

575 **SP-SLO-018** If the identity provider responds to the single logout request with `<lib:UnsupportedProfile>`,  
576 the service provider MUST re-submit its `<lib:LogoutRequest>` via the HTTP interaction  
577 specified above.

578 **SP-SLO-019** If the identity provider responds to the logout request with a SOAP 200 OK  
579 `<lib:LogoutResponse>`, indicating successful single logout, the service provider SHOULD  
580 send a HTTP response to the user agent confirming the single logout.

### 581 **2.4.5. Identity Provider Introduction**

582 This section describes the conformance requirements for a service provider implementing the identity provider  
583 introduction feature. The identity provider introduction feature is intended to allow service providers to discover  
584 which identity providers a Principal is using.

585 **SP-IPI-001** The identity provider introduction feature is an OPTIONAL element of the service provider  
586 basic conformance profile.

#### 587 **2.4.5.1. Common Domain Cookie**

588 The identity provider introduction relies on the use of a common domain cookie.

589 **SP-IPI-002** The common domain cookie MUST be constructed as specified in [[LibertyBindProf1.1, 3.6.1](#)].

---

#### 590 **2.4.5.2. Obtaining the Common Domain Cookie**

591 The service provider uses the common domain cookie to determine which identity providers a Principal uses. The  
592 common domain cookie is presented to the service provider after being read by an HTTP server in the common  
593 domain; the details of this interaction are outside the scope of this document.

594 **SP-IPI-003** If the HTTP server in the common domain is operated by the service provider, the service  
595 provider MAY redirect the user agent to an identity provider for single sign-on.

596 The details of this procedure are implementation-dependent, and are not normatively specified. However, one possible  
597 strategy is described in [[LibertyBindProf1.1, 3.6.3](#)].

### 598 **2.5. Service Provider Complete Profile**

599 This section defines the conformance requirements for the SP complete Profile.

600 **SP-COMPLETE** The service provider complete profile is defined to be the service provider basic profile with  
601 all optional interactions changed to REQUIRED except for the identity provider introduction  
602 interaction.

---

## 603 **References**

### 604 **Normative**

605 [LibertyProtSchema1.1] Beatty, John, Kemp, John, eds. "Liberty Protocols and Schema Specification," Version 1.1,  
606 Liberty Alliance Project (January 2003). <http://www.projectliberty/specs>

607 [LibertyBindProf1.1] Rouault, Jason, Wason, Tom, eds. "Liberty Bindings and Profiles Specification," Version 1.1,  
608 Liberty Alliance Project (January 2003). <http://www.projectliberty/specs>

609 [RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet  
610 Engineering Task Force (March 1997). <ftp://ftp.rfc-editor.org/in-notes/rfc2119.txt>

### 611 **Informative**

612 [LibertyImplGuide1.1] Kannappan, Lena, Lachance, Matthieu, Kemp, John, eds. "Liberty Architecture Implementa-  
613 tion Guidelines," Version 1.1, Liberty Alliance Project (January 2003). <http://www.projectliberty/specs>

614 [LibertyGlossary1.1] Mauldin, H., Wason, T., eds. "Liberty Glossary," Version 1.1, Liberty Alliance Project (January  
615 2003). <http://www.projectliberty.org/specs/>