



Liberty ID-WSF Profiles for Liberty enabled User Agents and Devices

Version: v2.0

Editors:

Robert Aarts, Hewlett-Packard
Jukka Kainulainen, Nokia Corporation
John Kemp, Nokia Corporation

Abstract:

User agents or devices, i.e. personal computers, mobile terminals, etc., participate in ID-WSF transactions in various ways. This document specifies profiles for some cases where user agents or devices act as an ID-WSF entity, i.e. execute software that implements at least parts of the ID-WSF specifications.

Filename: liberty-idwsf-client-profiles-v2.0.pdf

1

Notice

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

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

15 Copyright © 2006 Adobe Systems; America Online, Inc.; American Express Company; Amsoft Systems Pvt Ltd.;
16 Avatier Corporation; Axalto; Bank of America Corporation; BIPAC; BMC Software, Inc.; Computer Associates
17 International, Inc.; DataPower Technology, Inc.; Diversinet Corp.; Enosis Group LLC; Entrust, Inc.; Epok, Inc.;
18 Ericsson; Fidelity Investments; Forum Systems, Inc.; France Télécom; French Government Agence pour le
19 développement de l'administration électronique (ADAE); Gamefederation; Gemplus; General Motors; Giesecke &
20 Devrient GmbH; GSA Office of Governmentwide Policy; Hewlett-Packard Company; IBM Corporation; Intel
21 Corporation; Intuit Inc.; Kantega; Kayak Interactive; MasterCard International; Mobile Telephone Networks (Pty)
22 Ltd; NEC Corporation; Netegrity, Inc.; NeuStar, Inc.; Nippon Telegraph and Telephone Corporation; Nokia
23 Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OpenNetwork; Oracle Corporation; Ping Identity Corporation;
24 Reactivity Inc.; Royal Mail Group plc; RSA Security Inc.; SAP AG; Senforce; Sharp Laboratories of America;
25 Sigaba; SmartTrust; Sony Corporation; Sun Microsystems, Inc.; Supremacy Financial Corporation; Symlabs, Inc.;
26 Telecom Italia S.p.A.; Telefónica Móviles, S.A.; Trusted Network Technologies; UTI; VeriSign, Inc.; Vodafone
27 Group Plc.; Wave Systems Corp. All rights reserved.

28 Liberty Alliance Project
29 Licensing Administrator
30 c/o IEEE-ISTO
31 445 Hoes Lane
32 Piscataway, NJ 08855-1331, USA
33 info@projectliberty.org

34 Contents

35	1. Notation and Conventions
36	2. Overview
37	3. LUAD-WSC Profile
38	3.1. Rules for WSPs that offer service to LUADs
39	3.2. Examples
40	4. LUAD acting as WSP
41	4.1. LUAD-WSP profile
42	5. LUAD implementations of a Discovery Service
43	5.1. LUAD-DS Profile
44	References

45 **1. Notation and Conventions**

46 This specification uses schema documents conforming to W3C XML Schema (see [\[Schema1-2\]](#)) and normative text
47 to describe the syntax and semantics of XML-encoded messages.

48 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT",
49 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

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

53 Namespaces

54 • The prefix disco: represents the namespace defined in [\[LibertyDisco\]](#).

55 • The prefix sa: represents the namespace defined in [\[LibertyAuthn\]](#).

56 • The prefix sec: represents the namespace defined in [\[LibertySecMech\]](#).

57 • S: represents the namespace defined in [\[SOAPv1.1\]](#)

58 2. Overview

59 The ID-WSF specifications define a number of protocols that enable any party to act as a *Web Service Consumer*, a
60 *Web Service Provider*, or both. When user agents or devices wish to act in these roles, some particular issues need to be
61 addressed and hence additional specifications are useful to guarantee interoperability. The Liberty Alliance specifies
62 the [ID-WSF Authentication Service](#) by which a WSC on a user agent or device may authenticate to an identity provider,
63 and [LibertyPAOS](#) to enable a user agent or device to act as a WSP. Also, whenever a WSC or WSP acts as a user agent
64 it typically represents only a very small number of users, hence there are some particular considerations regarding
65 privacy.

66 User agents and devices that send or consume protocol messages specified in the ID-WSF ([ID-FF](#) or [SAML](#))
67 specifications are called *Liberty enabled User Agents and Devices*, abbreviated as *LUAD*. The defining characteristic
68 of a LUAD is that it is closely associated with one user (or a few users, such as a family); the LUAD represents
69 that user. This is very different from a web-site that acts as WSC or WSP and may represent thousands of users. In
70 addition, a LUAD is often, but certainly not always, *not* a highly-available HTTP server, unlike web-site based WSCs
71 and WSPs.

72 To illustrate some of the issues we briefly sketch out a scenario where a LUAD acts as a WSC in a typical ID-WSF
73 setting. The following, as well as the remainder of this document, assumes familiarity with the Liberty ID-WSF
74 specifications, especially the [Discovery Service](#) and [Security Mechanisms](#).

75 Any WSC that wishes to contact an ID-WSF WSP requires a `Service Instance Endpoint Reference` and often
76 some security tokens. A WSC typically obtains these from a Liberty ID-WSF Discovery Service (discovery service).
77 However, the discovery service is a WSP too, so for the WSC to make a request to the discovery service, it needs a
78 `<disco:ServiceInstanceEPR>` and security tokens for the discovery service.

79 A WSC can get such discovery service specific information when it acts as an SP during a single-sign-on transaction
80 using [SAML](#) (or [ID-FF](#)); the identity provider can insert in a response an `<AttributeStatement>` containing the
81 necessary information to contact the discovery service. This process is informally known as "bootstrapping ID-WSF"
82 (see [\[LibertyDisco\]](#)).

83 But a LUAD-WSC is not a web-site that acts as SP. So when the LUAD-WSC needs to contact the discovery service
84 it needs somehow to contact a party that can issue the `<disco:ServiceInstanceEPR>` and tokens needed. Here
85 we recommend that the LUAD-WSC obtains this information through the Liberty ID-WSF Authentication Service
86 ([\[LibertyAuthn\]](#)) offered by an identity provider.

87 The identity provider will need to authenticate the LUAD – this is similar to the identity provider authenticating
88 Principals that use a browser. As the LUAD-WSC is not a full-blown browser, however, it may not be able to present
89 a login form.

90 The identity provider and LUAD should use a protocol for authentication. The use of [\[LibertyAuthn\]](#) is recommended
91 for this purpose.

92 Once the LUAD-WSC can make requests to the discovery service it can ask the discovery service for descriptions and
93 tokens for a particular identity service type (a WSP). If the WSP that is referred to in the discovery service response
94 requires security tokens, the discovery service will create such tokens. Normally such tokens include a `providerID`
95 for the WSC and require that the WSC can authenticate as that provider to the WSP, perhaps by signing the request
96 with a particular key. A LUAD-WSC however does not have a `providerID`, as a `providerID` could compromise the
97 privacy of the LUAD user: the LUAD-WSC would show the same `providerID` to various WSPs allowing the WSPs
98 to collude about the LUAD-WSC and hence about the user. Thus the content of security tokens should be profiled for
99 various situations.

100 In summary, this document then specifies how *LUAD* implementations should utilize the various Liberty Alliance
101 specifications in order to enable particular scenarios while ensuring a high degree of interoperability, security and
102 privacy. The following sections specify and discuss profiles for particular uses of a LUAD. Note that in each section,
103 profiles are defined for both the LUAD as well as for the providers that (wish to) interact with the LUAD.

104 3. LUAD-WSC Profile

105 A LUAD-WSC will often need to authenticate to a provider; for example when that LUAD-WSC wants to make a
106 request to a discovery service. The discovery service may have been set up to require a security token; web-site based
107 WSCs typically obtain such a token during a authentication transaction with an identity provider associated to that
108 discovery service. But with a LUAD-WSC there may not be an associated browsing session, hence no interaction with
109 an identity provider has occurred and the WSC cannot have a valid security token for the discovery service. In another
110 typical scenario the WSP is not an ID-WSF WSP, i.e. not an "identity providing" service but an "identity consuming"
111 service (here we abbreviate those non-ID-WSF Web Service Providers as *wSP* to indicate that these are a subclass
112 of SPs). A LUAD-WSC that requests service from such *wSP*s may need to obtain SAML (or ID-FF) authentication
113 assertions that will be presented as security tokens to the *wSP*.

114 As the LUAD represents at most a few users, the LUAD should not use a single authentication identity towards
115 different providers. To achieve the required level of security and privacy the LUAD and provider must carefully
116 choose the authentication mechanism and nature of credentials.

117 A LUAD-WSC implementation must adhere to the following rules:

118 1. The LUAD-WSC SHOULD avoid being traceable across providers. Hence, the LUAD SHOULD NOT authenti-
119 cate to different providers using a single credential.

120 **Note**

121 This implies that if a LUAD-WSC employs [message level confidentiality protection](#), different signing keys should
122 be used in communication with each individual provider.

123 2. If a LUAD-WSC is required to authenticate to a provider directly, because it does not have or cannot obtain
124 security tokens, the LUAD-WSC SHOULD authenticate using [\[LibertyAuthn\]](#).

125 **Note**

126 This applies to situations where the LUAD *itself* needs to assert its identity to a provider – typically only when
127 a LUAD authenticates to an identity provider. In most cases a LUAD-WSC can obtain (bearer) security tokens
128 from a Liberty ID-WSF Discovery Service and would include these tokens in the message to the WSP.

129 3. A LUAD-WSC SHOULD use the ID-WSF Authentication Service specified in [\[LibertyAuthn\]](#) to obtain security
130 tokens from an identity provider; these tokens can then be used when submitting a `<disco:Query>` to a
131 Discovery Service.

132 4. A LUAD-WSC that wishes to interact with a WSP SHOULD support at least the

133 `urn:liberty:security:2005-02:TLS:Bearer` security mechanism as specified in [\[LibertySecMech\]](#).

134 **Note**

135 Note that these rules *do* allow the LUAD to authenticate to a provider using a client certificate. However, that same
136 certificate should not be used to authenticate to another provider. For example a LUAD-WSC could use its certificate
137 to authenticate to a discovery service or an identity provider (to both if both interfaces are offered by one provider) but
138 not then to another WSP.

139 3.1. Rules for WSPs that offer service to LUADs

140 ID-WSF compliant WSPs that register with a discovery service SHOULD support at least the
141 `urn:liberty:security:2005-02:TLS:Bearer` security mechanism as specified in [\[LibertySecMech\]](#).

142 3.2. Examples

143 See [\[LibertyAuthn\]](#) for examples of interactions of a LUAD-WSC.

144 4. LUAD acting as WSP

145 A WSP that is deployed on a LUAD is again not very different from a network WSP. One issue for a client-WSP is
146 reachability: a LUAD is typically not acting as a HTTP/SOAP server, may be behind a firewall, and does not have a
147 fixed IP address.

148 A second issue is that a LUAD-WSP, by definition, offers service for only one, or a few, Principals. Hence, the LUAD-
149 WSP cannot have a *service provider* identity. Normally a WSP needs to offer a `providerID` and metadata that
150 WSCs use to construct requests. A LUAD-WSP should not have a `providerID` and hence cannot publish metadata.
151 Metadata and signing keys make the client traceable to different WSCs, compromising the privacy of the LUAD user.

152 Note

153 A [Liberty ID-WSF Discovery Service](#) hosted on a LUAD has to satisfy additional rules (see next section).

154 4.1. LUAD-WSP profile

155 A LUAD-WSP must adhere to the following rules:

156 1. It is RECOMMENDED that LUADs that are not normally reachable expose ID-WSP web services over
157 [LibertyPAOS](#)

158 Note

159 Note that future versions of the ID-WSF specifications may include SOAP bindings for alternative approaches,
160 such as SIP.

161 2. The LUAD-WSP SHOULD avoid being traceable. If the WSP uses [message level confidentiality protection](#),
162 different signing keys for communications with different WSCs SHOULD be used.

163 3. As the LUAD-WSP is not an entity different from the Principal it represents, it should not have a `providerID`.
164 A discovery service cannot issue a `Service Instance Endpoint Reference` for entities that do not have a
165 `providerID`. Hence, A LUAD-WSP SHOULD NOT register with a Discovery Service.

166 An ID-WSF WSC that requests services from a LUAD-WSP must adhere to the following rules:

167 1. If authentication of the WSP is needed it is RECOMMENDED that SP/WSCs authenticate the LUAD-WSP using
168 SAML (or ID-FF), presumably before making an ID-WSF request to the PAOS-exposed WSP.

169 See [\[LibertyPAOS\]](#) for an example of interaction with a LUAD-WSP. Another example is given in [Section 5](#).

170 **5. LUAD implementations of a Discovery Service**

171 A LUAD implementation of a [discovery service](#), i.e. a LUAD-DS, can be useful as a discovery service can inform
172 parties in its immediate proximity about identity services for the user of the LUAD. For example a LUAD-DS could
173 inform a mall entrance about a personal profile service, or inform a parking exit about a payment service. As with
174 any LUAD-WSP implementations there are some issues around traceability of the client, but in a discovery service
175 these problems are more important as a discovery service very likely must issue signed security tokens to parties that
176 subsequently will submit those tokens to a WSP.

177 **5.1. LUAD-DS Profile**

178 An ID-WSF discovery service that executes at a LUAD must adhere to the following rules:

- 179 1. The LUAD-DS implementation **SHOULD** adhere to the rules defined for [LUAD-WSP implementations](#).
- 180 2. The key that the LUAD-DS uses to sign security tokens **SHOULD** be unique for each WSP that registers with the
181 LUAD-DS. The LUAD-DS **SHOULD** inform the WSP about the key when the WSP registers with the LUAD-DS,
182 i.e. the LUAD should include the key in the `disco:ModifyResponse` as specified in [\[LibertyDisco\]](#).
183 When the LUAD-DS sends key material it **MUST** ensure [Transport Layer Channel Protection](#), and in the presence
184 of intermediaries **MUST** also ensure [Message Confidentiality Protection](#), using one of the mechanisms specified
185 in [\[LibertySecMech\]](#).

186 References

187 Normative

- 188 [LibertyDisco] Hodges, Jeff, Cahill, Conor, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0,
189 Liberty Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 190 [LibertyInteract] Aarts, Robert, Madsen, Paul, eds. "Liberty ID-WSF Interaction Service Specification," Version 2.0,
191 Liberty Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 192 [LibertyPAOS] Aarts, Robert, Kemp, John, eds. "Liberty Reverse HTTP Binding for SOAP Specification," Version
193 2.0, Liberty Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 194 [LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version
195 1.2-errata-v3.0, Liberty Alliance Project (14 December 2004). <http://www.projectliberty.org/specs>
- 196 [LibertySecMech] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version v2.0, Liberty
197 Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 198 [LibertyAuthn] Hodges, Jeff, Aarts, Robert, Madsen, Paul, Cantor, Scott, eds. "Liberty ID-WSF Authentication,
199 Single Sign-On, and Identity Mapping Services Specification," Version v2.0, Liberty Alliance Project (30
200 July, 2006). <http://www.projectliberty.org/specs>
- 201 [LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, Whitehead, Greg, Madsen, Paul, eds. "Lib-
202 erty ID-WSF SOAP Binding Specification," Version 2.0, Liberty Alliance Project (30 July, 2006).
203 <http://www.projectliberty.org/specs>
- 204 [RFC2119] S. Bradner "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet
205 Engineering Task Force (March 1997). <http://www.ietf.org/rfc/rfc2119.txt>
- 206 [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., eds. (June
207 1999). "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, The Internet Engineering Task Force
208 <http://www.ietf.org/rfc/rfc2616.txt>
- 209 [RFC3066] Alvestrand, H., eds. (January 2001). "Tags for the Identification of Languages," RFC 3066., Internet
210 Engineering Task Force <http://www.ietf.org/rfc/rfc3066.txt>
- 211 [Schema1-2] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (28 October
212 2004). "XML Schema Part 1: Structures Second Edition," Recommendation, World Wide Web Consortium
213 <http://www.w3.org/TR/xmlschema-1/>
- 214 [SOAPv1.1] "Simple Object Access Protocol (SOAP) 1.1," Box, Don, Ehnebuske, David, Kakivaya, Gopal, Layman,
215 Andrew, Mendelsohn, Noah, Nielsen, Henrik Frystyk, Winer, Dave, eds. World Wide Web Consortium W3C
216 Note (08 May 2000). <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

217 Informative

- 218 [LibertyIDPP] Kellomäki, Sampo, Lockhart, Rob, eds. "Liberty ID-SIS Personal Profile Service Specification,"
219 Version 1.1, Liberty Alliance Project (29 September, 2005). <http://www.projectliberty.org/specs>