



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

Version: v1.1

Editors:

Robert Aarts, 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-v1.1.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.** Implementors
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 © 2004-2005 ADAE; Adobe Systems; America Online, Inc.; American Express Company; Avatier
16 Corporation; Axalto; Bank of America Corporation; BIPAC; Computer Associates International, Inc.; DataPower
17 Technology, Inc.; Diversinet Corp.; Enosis Group LLC; Entrust, Inc.; Epok, Inc.; Ericsson; Fidelity Investments;
18 Forum Systems, Inc. ; France Telecom; Gamefederation; Gemplus; General Motors; Giesecke & Devrient GmbH;
19 Hewlett-Packard Company; IBM Corporation; Intel Corporation; Intuit Inc.; Kantega; Kayak Interactive; MasterCard
20 International; Mobile Telephone Networks (Pty) Ltd; NEC Corporation; Netegrity, Inc.; NeuStar, Inc.; Nippon
21 Telegraph and Telephone Corporation; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OpenNetwork; Oracle
22 Corporation; Ping Identity Corporation; Royal Mail Group plc; RSA Security Inc.; SAP AG; Senforce; Sharp
23 Laboratories of America; Sigaba; SmartTrust; Sony Corporation; Sun Microsystems, Inc.; Telefonica Moviles, S.A.;
24 Trusted Network Technologies.; Trustgenix; UTI; VeriSign, Inc.; Vodafone Group Plc. All rights reserved.

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

31 **Contents**

32 1. Notation and Conventions4
33 2. Overview5
34 3. LUAD-WSC Profile6
35 4. LUAD acting as WSP7
36 5. LUAD implementations of a Discovery Service8
37 References9

38 1. Notation and Conventions

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

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

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

46 Namespaces

- 47 • The prefix disco: represents the namespace defined in [\[LibertyDisco\]](#).
- 48 • The prefix lib: represents the namespace defined in [\[LibertyProtSchema\]](#).
- 49 • The prefix sa: represents the namespace defined in [\[LibertyAuthn\]](#).
- 50 • The prefix sec: represents the namespace defined in [\[LibertySecMech\]](#).
- 51 • S: represents the namespace defined in [\[SOAPv1.1\]](#)

52 2. Overview

53 The ID-WSF specifications define a number of protocols that enable any party to act as a *Web Service Consumer*, a
54 *Web Service Provider*, or both. When user agents or devices wish to act in these roles, some particular issues need to be
55 addressed and hence additional specifications are useful to guarantee interoperability. The Liberty Alliance specifies
56 the [ID-WSF Authentication Service](#) by which a WSC on a user agent or device may authenticate to an identity provider,
57 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
58 it typically represents only a very small number of users, hence there are some particular considerations regarding
59 privacy.

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

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

68 Any WSC that wishes to contact an ID-WSF WSP requires a `ResourceOffering` and often some security to-
69 kens. A WSC typically obtains these from a Liberty ID-WSF Discovery Service (discovery service). How-
70 ever, the discovery service is a WSP too, so for the WSC to make a request to the discovery service, it needs a
71 `<disco:ResourceOffering>` and security tokens for the discovery service.

72 A WSC can get such discovery service specific information when it acts as an SP in the Liberty Identity Federation
73 Framework during a single-sign-on transaction; the identity provider can insert the necessary information to contact
74 the discovery service in a `<lib:AuthnResponse>`. This process is informally known as "bootstrapping ID-WSF".

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

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

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

84 Once the LUAD-WSC can make requests to the discovery service it can ask the discovery service for descriptions
85 and tokens for a particular identity service type (a WSP). If the WSP that is referred to in the discovery service
86 response requires security tokens (such as a `<sec:ResourceAccessStatement>`) the discovery service will create
87 such tokens. Normally such tokens include a `providerID` for the WSC and require that the WSC can authenticate as
88 that provider to the WSP, perhaps by signing the request with a particular key. A LUAD-WSC however does not have
89 a `providerID`, as a `providerID` could compromise the privacy of the LUAD user: the LUAD-WSC would show the
90 same `providerID` to various WSPs allowing the WSPs to collude about the LUAD-WSC and hence about the user.
91 Thus the content of security tokens should be profiled for various situations.

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

96 3. LUAD-WSC Profile

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

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

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

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

112 **Note:**

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

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

117 **Note:**

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

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

124 4. A LUAD-WSC that wishes to interact with a WSP SHOULD support at least the `urn:liberty:security:
125 2005-02:TLS:Bearer` security mechanism as specified in [\[LibertySecMech\]](#).

126 **Note:**

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

131 3.1. Rules for WSPs that offer service to LUADs

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

134 3.2. Examples

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

136 4. LUAD acting as WSP

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

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

144 **Note:**

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

146 4.1. LUAD-WSP profile

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

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

150 **Note:**

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

153 2. The LUAD-WSP SHOULD avoid being traceable. If the WSP uses [message level confidentiality protection](#) a
154 different signing key for communications with different WSCs SHOULD be used.

155 3. As the LUAD-WSP is not an entity different from the Principal it represents, it should not have a `providerID`.
156 A discovery service cannot issue a `ResourceOffering` for entities that do not have a `providerID`. Hence, A
157 LUAD-WSP SHOULD NOT register with a Discovery Service.

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

159 1. If authentication of the WSP is needed it is RECOMMENDED that SP/WSCs authenticate the LUAD-WSP using
160 ID-FF, presumably before making a ID-WSF request to the PAOS-exposed WSP.

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

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

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

169 **5.1. LUAD-DS Profile**

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

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

References

178

Normative

179

- 180 [LibertyBindProf] Cantor, Scott, Kemp, John, Champagne, Darryl, eds. "Liberty ID-FF Bindings and
181 Profiles Specification," Version 1.2-errata-v2.0, Liberty Alliance Project (12 September 2004).
182 <http://www.projectliberty.org/specs>
- 183 [LibertyDisco] Sergeant, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification," Version 1.2, Liberty
184 Alliance Project (12 December 2004). <http://www.projectliberty.org/specs>
- 185 [LibertyInteract] Aarts, Robert, eds. "Liberty ID-WSF Interaction Service Specification," Version 1.1, Liberty Alliance
186 Project (14 December 2004). <http://www.projectliberty.org/specs>
- 187 [LibertyMetadata] Davis, Peter, eds. "Liberty Metadata Description and Discovery Specification," Version 1.1, Liberty
188 Alliance Project (14 December 2004). <http://www.projectliberty.org/specs>
- 189 [LibertyPAOS] Aarts, Robert, eds. "Liberty Reverse HTTP Binding for SOAP Specification," Version 1.1, Liberty
190 Alliance Project (14 December 2004). <http://www.projectliberty.org/specs>
- 191 [LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version
192 1.2-errata-v3.0, Liberty Alliance Project (14 December 2004). <http://www.projectliberty.org/specs>
- 193 [LibertySecMech] Ellison, Gary, eds. "Liberty ID-WSF Security Mechanisms," Version 1.2, Liberty Alliance Project
194 (14 December 2004). <http://www.projectliberty.org/specs>
- 195 [LibertyAuthn] Hodges, Jeff, Aarts, Robert, eds. "Liberty ID-WSF Authentication Service Specification,"
196 Version 1.1, Liberty Alliance Project (14 December 2004). <http://www.projectliberty.org/specs/>
197 [<http://www.projectliberty.org/specs/>]
- 198 [LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, eds. "Liberty ID-WSF SOAP Binding Specification
199 ," Version 1.2, Liberty Alliance Project (14 December 2004). <http://www.projectliberty.org/specs>
- 200 [RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet
201 Engineering Task Force (March 1997). <http://www.ietf.org/rfc/rfc2119.txt> [March 1997].
- 202 [RFC3066] Alvestrand, H., eds. (January 2001). "Tags for the Identification of Languages," RFC 3066., Internet
203 Engineering Task Force <http://www.ietf.org/rfc/rfc3066.txt> [January 2001].
- 204 [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., eds. (June
205 1999). "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, The Internet Engineering Task Force
206 <http://www.ietf.org/rfc/rfc2616.txt> [June 1999].
- 207 [Schema1] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (May
208 2002). "XML Schema Part 1: Structures," Recommendation, World Wide Web Consortium
209 <http://www.w3.org/TR/xmlschema-1/>
- 210 [SOAPv1.1] "Simple Object Access Protocol (SOAP) 1.1," Box, Don, Ehnebuske, David, Kakivaya, Gopal, Layman,
211 Andrew, Mendelsohn, Noah, Nielsen, Henrik Frystyk, Winer, Dave, eds. World Wide Web Consortium W3C
212 Note (08 May 2000). <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 213 **Informative**
- 214 [LibertyIDPP] Kellomaki, Sampo, eds. "Liberty Identity Personal Profile Service Specification," Version 1.0, Liberty
215 Alliance Project (12 November 2003). <http://www.projectliberty.org/specs>