LIBERTY ALLIANCE project

Liberty Architecture Implementation Guidelines

Version 1.0

11 July 2002

Document identifier: liberty-architecture-impl-guidelines-v1.0

Notice

Liberty Alliance Project
Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

66 ## Editors

67 Lena Kannappan, France Telecom

68 Matthieu Lachance, Openwave Systems Inc.

69 ## Contributors

70

71 The following Liberty Alliance Project Sponsor companies contributed to the development of this
72 specification:

73

ActivCard
American Express Travel Related Services
America Online, Inc.
Bank of America
Bell Canada
Catavault
Cingular Wireless
Cisco Systems, Inc.
Citigroup
Cyberun Corporation
Deloitte & Touche LLP
EarthLink, Inc.
Electronic Data Systems, Inc.
Entrust, Inc.
Ericsson
Fidelity Investments
France Telecom
Gemplus
General Motors
Hewlett-Packard Company
i2 Technologies, Inc.
Intuit Inc.
MasterCard International
Nextel Communications
Nippon Telegraph and Telephone Company
Nokia Corporation
Novell, Inc.
NTT DoCoMo, Inc.
OneName Corporation
Openwave Systems Inc.
PricewaterhouseCoopers LLP
Register.com
RSA Security Inc
Sabre Holdings Corporation
SAP AG
SchlumbergerSema
Sony Corporation
Sun Microsystems, Inc.
United Airlines
VeriSign, Inc.
Visa International
Vodafone Group Plc
Wave Systems

74

75

76

77

78

78 # Table of Contents

92
93

1 Introduction

This document defines the recommended implementation guidelines and checklists for the Liberty architecture focused on deployments for the service-providing entities: service providers, identity providers, and Liberty-enabled clients or proxies (LECPs). It is intended to provide recommended implementation guidelines to Liberty component developers to help them decide what they need to implement to meet their business needs. Because Liberty Version 1.0 does not provide formal compliance, this document does not contain any conformance requirements — only recommendations. A recommended profile tailored according to the high-level Liberty features is provided for different Liberty service-providing entities. Implementers facing specific needs can decide to implement what they need and claim support for each specific feature separately.

The document also provides a checklist of requirements based on the following Liberty architecture specification categories that implementers can use to advertise their supported feature set:

- Functionality in the Liberty protocols and schemas described
- Bindings and profiles defined for each Liberty protocol type (specific interactions between identity providers, service providers, and LECPs)
- The authentication request and reply context-specific information

Definitions for Liberty-specific terms can be found in [LibertyGloss]. Note: Phrases and numbers in brackets [ ] refer to other documents; details of these references can be found in Section 4 (at the end of this document).

2 Recommended Liberty Architecture Implementation Guidelines

The recommended implementation guidelines for identity providers, service providers, and LECPs are listed in the tables in 2.1 through 2.3. The guidelines refer to front-channel-based and back-channel-based mechanisms. *Front channel* is described as a communication channel where HTTP redirect-, GET-, and POST-based request and response protocol messages between the identity provider and the service provider flow through the Web browser. *Back channel* is a SOAP/HTTP-based direct communication channel between the identity provider and the service provider. A service provider with SOAP client support is considered to be a “back-channel-capable SP” whereas a “basic SP” is not back-channel-capable.

2.1 Identity Provider Implementation Guidelines

Liberty Feature	Recommendations
Single Sign-On	It is strongly recommended that identity providers support the LECP single sign-on profile to ensure forward compatibility. The LECP profile is intended for future clients of all kinds (thin and thick) as well as existing wireless thin clients (WML, HDML, etc) when used with a LEP. Identity providers that want to support existing HTML client environments should implement the browser artifact and the browser POST single sign-on profiles. To support existing WML client in environments that do not contain any LEP, identity providers should support the WML single sign-on profile.

Liberty Feature	Recommendations
Identity Federation	Identity providers that want to support permanent identity linking between service providers and identity providers (beyond the stateless single sign-on association) should support the `<DoFederate>` element of the `<AuthnRequest>` for all the supported single sign-on profiles.
Federation Termination Notification	Identity providers that support identity federation should also support the Federation Termination Notification Protocol. When supported, both service-provider-initiated and identity-provider-initiated federation termination notification should be supported. Liberty offers two federation termination notification mechanisms: • Front channel, or HTTP-redirect-based • Back channel, or SOAP-based As a minimum, identity providers should support the front-channel-based mechanism. Identity providers that want to support back-channel-capable SPs should implement both mechanisms.
Name Registration	The Name Registration Protocol allows the service provider to use its own opaque handle to identify the Principal when communicating with the identity provider (rather than using the identity provider's opaque handle), but requires back-channel-capable SPs. Identity providers that want to support back-channel-capable SPs should implement this feature.
Single Logout	The Single Logout Protocol allows logging out a Principal from all its active sessions to service providers, linked to an identity provider. Identity providers keeping trace of the Principal's service provider sessions should implement this feature. When supported, both service-provider-initiated and identity-provider-initiated single logout should be supported. Liberty offers two single logout mechanisms: • Front channel, or HTTP-redirect-based • Back channel, or SOAP-based As a minimum, identity providers supporting this feature should support the front-channel-based mechanism. Identity providers that want to support back-channel-capable SPs should implement both mechanisms.
Identity Provider Introduction	Identity providers that want to support more than a single circle of trust simultaneously should support the Identity Provider Introduction Protocol.

123

124 2.2 Service Provider Implementation Guidelines

125 In general service providers are divided in two categories: the back-channel-capable SPs and the
126 basic SPs (that are not back-channel-capable).

127

127

Liberty Feature	Recommendations
Single Sign-On	It is strongly recommended that service providers support the LECP single sign-on profile to ensure forward compatibility. The LECP profile is intended for future clients of all kinds (thin and thick) as well as existing wireless thin clients (WML, HDML, etc) when used with a LEP. Service providers that want to support existing HTML client environments should implement the browser artifact and the browser POST single sign-on profiles. To support existing WML client in environments that do not contain any LEP, service providers should support the WML single sign-on profile.
Identity Federation	Service providers that want to support permanent identity linking between service providers and identity providers (beyond the stateless single sign-on association) should support the `<DoFederate>` element of the `<AuthnRequest>` for all the supported single sign-on profiles.
Federation Termination Notification	Service providers that support identity federation should also support the Federation Termination Notification Protocol. When supported, both service-provider-initiated and identity-provider-initiated federation termination notification should be supported. Service providers should support either the front-channel or back-channel federation termination notification mechanisms depending on their respective capabilities although nothing prevents them from supporting both mechanisms if desired.
Name Registration	The Name Registration Protocol allows the service provider to use its own opaque handle to identify the Principal when communicating with the identity provider (rather than using the identity provider's opaque handle), but requires back-channel-capable SPs. Back-channel-capable SPs should implement this feature.
Single Logout	The Single Logout Protocol allows logging out a Principal from all its active sessions to service providers, linked to an identity provider. When supported, both service-provider-initiated and identity-provider-initiated single logout should be supported. Service providers should support either the front-channel or back-channel single logout mechanisms depending on their respective capabilities although nothing prevents them from supporting both mechanisms if desired.
Identity Provider Introduction	Service providers that want to support networks with more than a single circle of trust simultaneously should support the Identity Provider Introduction Protocol.

128

129 2.3 LECP Implementation Guidelines

130

Liberty Feature	Recommendations
Single Sign-On	Support for LECP single sign-on profile.

131

3 Liberty Architecture Specifications Checklist

3.1 Liberty Profiles and Bindings Requirements — Identity Provider

Req ID#	Description	Ref	Y/N
IDP-FED-1	Identity Federation	Section 3.2.1 [LibertyBindProf]	
IDP-SSO-1	Single Sign-On using Browser Artifact	Section 3.2.2 [LibertyBindProf]	
IDP-SSO-2	Single Sign-On using Browser POST	Section 3.2.3 [LibertyBindProf]	
IDP-SSO-3	Single Sign-On using WML POST	Section 3.2.4 [LibertyBindProf]	
IDP-SSO-4	Single Sign-On using LECP	Section 3.2.5 [LibertyBindProf]	
IDP-REG-1	Register Name Identifier	Section 3.3 [LibertyBindProf]	
IDP-FED-2	Identity Federation Termination — Front Channel	Section 3.4 [LibertyBindProf]	
IDP-FED-3	Identity Federation Termination — Back Channel	Section 3.4 [LibertyBindProf]	
IDP-FED-4	Federation Termination Notification (Identity Provider Initiated) — Front Channel	Section 3.4.1 [LibertyBindProf]	
IDP-FED-5	Federation Termination Notification (Identity Provider Initiated) — Back Channel	Section 3.4.1 [LibertyBindProf]	
IDP-FED-6	Federation Termination Notification (Service Provider Initiated) — Front Channel	Section 3.4.2 [LibertyBindProf]	
IDP-FED-7	Federation Termination Notification (Service Provider Initiated) — Back Channel	Section 3.4.2 [LibertyBindProf]	
IDP-SLO-1	Single Logout	Section 3.5 [LibertyBindProf]	
IDP-SLO-2	Single Logout Initiated by Identity Provider: Redirect	Section 3.5.1 [LibertyBindProf]	
IDP-SLO-3	Single Logout Initiated by Identity Provider: SOAP	Section 3.5.1 [LibertyBindProf]	
IDP-SLO-4	Single Logout Initiated by Service Provider: Redirect	Section 3.5.2 [LibertyBindProf]	
IDP-SLO-5	Single Logout Initiated by Service Provider: SOAP	Section 3.5.2 [LibertyBindProf]	
IDP-INT-1	Identity Provider Introduction	Section 3.6 [LibertyBindProf]	
IDP-COM-1	HTTP Connection over SSL3.0 or TLS1.0 [RFC2246], WTLS	[SSLv3], [RFC2246], [WTLS]	
IDP-COM-2	Support for Minimum URL length of 256 bytes	[RFC2965]	
IDP-COM-3	Support for Session Cookies	[RFC2965]	

3.2 Liberty Profiles and Bindings Requirements — Service Provider

Req ID#	Description	Ref	Y/N
SP-FED-1	Identity Federation	Section 3.2.1 [LibertyBindProf]	
SP-SSO-1	Single Sign-On using Browser Artifact	Section 3.2.2 [LibertyBindProf]	
SP-SSO-2	Single Sign-On using Browser POST	Section 3.2.3 [LibertyBindProf]	
SP-SSO-3	Single Sign-On using WML	Section 3.2.4 [LibertyBindProf]	
SP-SSO-4	Single Sign-On using LECP	Section 3.2.5 [LibertyBindProf]	
SP-REG-1	Register Name Identifier	Section 3.3 [LibertyBindProf]	
SP-FED-2	Identity Federation Termination — Front Channel	Section 3.4 [LibertyBindProf]	
SP-FED-3	Identity Federation Termination — Back Channel	Section 3.4 [LibertyBindProf]	
SP-FED-4	Federation Termination Notification (Identity Provider Initiated) — Front Channel	Section 3.4.1 [LibertyBindProf]	
SP-FED-5	Federation Termination Notification (Identity Provider Initiated) — Back Channel	Section 3.4.1 [LibertyBindProf]	
SP-FED-6	Federation Termination Notification (Service Provider Initiated) — Front Channel	Section 3.4.2 [LibertyBindProf]	
SP-FED-7	Federation Termination Notification (Service Provider Initiated) — Back Channel	Section 3.4.2 [LibertyBindProf]	
SP-SLO-1	Single Logout	Section 3.5 [LibertyBindProf]	
SP-SLO-2	Single Logout Initiated by Identity Provider: Redirect	Section 3.5.1 [LibertyBindProf]	
SP-SLO-3	Single Logout Initiated by Identity Provider: SOAP	Section 3.5.1 [LibertyBindProf]	
SP-SLO-4	Single Logout Initiated by Service Provider: Redirect	Section 3.5.2 [LibertyBindProf]	
SP-SLO-5	Single Logout Initiated by Service Provider: SOAP	Section 3.5.2 [LibertyBindProf]	
SP-INT-1	Identity Provider Introduction	Section 3.6 [LibertyBindProf]	
SP-COM-1	HTTP Connection over SSL3.0 or TLS1.0 [RFC2246], WTLS	[SSLv3], [RFC2246], [WTLS]	
SP-COM-2	Support for Minimum URL Length of 256 bytes	[RFC2965]	
SP-COM-3	Support for Session Cookies	[RFC2965]	

3.3 Liberty Profiles and Bindings Requirements — LECP

Req ID#	Description	Ref	Y/N
LECP-SSO-1	Single Sign-On using LECP	Section 3.2.5 [LibertyBindProf]	
LECP-COM-1	Support for Minimum URL Length of 256 bytes	[RFC2965]	
LECP-COM-2	Support for Session Cookies	[RFC2965]	

3.4 Authentication Context Requirements — Identity Provider

Req ID#	Description	Ref	Y/N
IDP-AUTHN-01	MobileContract	Section 5.1.1 [LibertyAuthnContext]	
IDP-AUTHN-02	MobileDigitalID	Section 5.1.2 [LibertyAuthnContext]	
IDP-AUTHN-03	MobileUnregistered	Section 5.1.3 [LibertyAuthnContext]	
IDP-AUTHN-04	Password	Section 5.1.4 [LibertyAuthnContext]	
IDP-AUTHN-05	Password-ProtectedTransport	Section 5.1.5 [LibertyAuthnContext]	
IDP-AUTHN-06	Previous-Session	Section 5.1.6 [LibertyAuthnContext]	
IDP-AUTHN-07	Smartcard	Section 5.1.7 [LibertyAuthnContext]	
IDP-AUTHN-08	Smartcard-PKI	Section 5.1.8 [LibertyAuthnContext]	
IDP-AUTHN-09	Software-PKI	Section 5.1.9 [LibertyAuthnContext]	
IDP-AUTHN-10	Time-Sync-Token	Section 5.1.10 [LibertyAuthnContext]	

3.5 Authentication Context Requirements — Service Provider

Req ID#	Description	Ref	Y/N
SP-AUTHN-01	MobileContract	Section 5.1.1 [LibertyAuthnContext]	
SP-AUTHN-02	MobileDigitalID	Section 5.1.2 [LibertyAuthnContext]	
SP-AUTHN-03	MobileUnregistered	Section 5.1.3 [LibertyAuthnContext]	
SP-AUTHN-04	Password	Section 5.1.4 [LibertyAuthnContext]	

Req ID#	Description	Ref	Y/N
SP-AUTHN-05	Password-ProtectedTransport	Section 5.1.5 [LibertyAuthnContext]	
SP-AUTHN-06	Previous-Session	Section 5.1.6 [LibertyAuthnContext]	
SP-AUTHN-07	Smartcard	Section 5.1.7 [LibertyAuthnContext]	
SP-AUTHN-08	Smartcard-PKI	Section 5.1.8 [LibertyAuthnContext]	
SP-AUTHN-09	Software-PKI	Section 5.1.9 [LibertyAuthnContext]	
SP-AUTHN-10	Time-Sync-Token	Section 5.1.10 [LibertyAuthnContext]	

147

148 3.6 Authentication Context Requirements — LECP

149

Req ID#	Description	Ref	Y/N
LECP-AUTHN-01	MobileContract	Section 5.1.1 [LibertyAuthnContext]	
LECP-AUTHN-02	MobileDigitalID	Section 5.1.2 [LibertyAuthnContext]	
LECP-AUTHN-03	MobileUnregistered	Section 5.1.3 [LibertyAuthnContext]	
LECP-AUTHN-04	Password	Section 5.1.4 [LibertyAuthnContext]	
LECP-AUTHN-05	Password-ProtectedTransport	Section 5.1.5 [LibertyAuthnContext]	
LECP-AUTHN-06	Previous-Session	Section 5.1.6 [LibertyAuthnContext]	
LECP-AUTHN-07	Smartcard	Section 5.1.7 [LibertyAuthnContext]	
LECP-AUTHN-08	Smartcard-PKI	Section 5.1.8 [LibertyAuthnContext]	
LECP-AUTHN-09	Software-PKI	Section 5.1.9 [LibertyAuthnContext]	
LECP-AUTHN-10	Time-Sync-Token	Section 5.1.10 [LibertyAuthnContext]	

150

151 4 References

152 [LibertyAuthnContext] Madson, P., “Liberty Authentication Assertion Context Specification.”
153 [LibertyBindProf] Rouault, J., “Liberty Bindings and Profiles Specification.”
154 [LibertyGloss] Ellison, G., “Liberty Glossary.”
155 [LibertyProtSchema] Beatty, J., “Liberty Protocols and Schemas Specification.”
156 [RFC2246] “The TLS Protocol Version 1.0,” http://www.ietf.org/rfc/rfc2246.txt.

[RFC2965] "HTTP State Management Mechanism," **http://www.ietf.org/rfc/rfc2965.txt**.

[SAMLBind] Mishra, Prateek, et al., "Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)," **http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-10.doc**, OASIS, 10 January 2002.

[SAMLCore] Hallam-Baker, P., et al., "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)," **http://www.oasis-open.org/committees/security/docs/draft-sstc-core-21.pdf**, OASIS, December 2001.

[SOAP1.1] Box, D., et al., "Simple Object Access Protocol (SOAP) 1.1," **http://www.w3.org/TR/SOAP**, World Wide Web Consortium Note, May 2000.

[SSLv3] "The SSL Protocol Version 3.0," **http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt**.

[WML1.3 **]** **"**Wireless Application Protocol Wireless Markup Language Specification Version 1.3," Wireless Application Protocol Forum, Ltd., **http://www.wapforum.org/,** 19 February 2000.

[WTLS] "Wireless Transport Layer Security," **http://www1.wapforum.org/tech/documents/WAP-261_102-WTLS-20011027-a.pdf.**

[XMLSig] Eastlake D., et al., "XML-Signature Syntax and Processing," **http://www.w3.org/TR/xmldsig-core/**, World Wide Web Consortium.