OMA Network Identity Federation Framework

Approved Version 1.0 – 28 Mar 2006

Open Mobile Alliance

OMA-TS-OWSER_NI_FF-V1_0-20060328-A

Contents

Figures

Tables

1. Scope

This document is part of a series of documents [OWSER NI FF], [OWSER NI AD] [OWSER NI WSF] that will be used to specify components of the OMA Web Services Network Identity Enabler (OWSER NI).

The "OMA Web Services Network Identity Enabler (OWSER NI): Architecture" document [OWSER NI AD] describes the architecture of a technical solution to the requirements in [NI RD]based on the Liberty Alliance Identity Federation Framework and Identity Web Services Framework

The "OMA Network Identity Web Services Enabler: (OWSER NI): Identity Web Services Framework " [OWER NI WSF] provides the specification of the components needed to fulfil the requirements in [NI-RD] related to accessing user-related attributes (e.g., user location, presence status etc.) in a privacy-protected manner in a Liberty-enabled Web Services environment.

This document, namely the "OMA Network Identity Web Services Enabler: (OWSER NI): Identity Federation Framework " document [OWSER NI FF] provides the specifications of the components needed to leverage Identity Federation in a Liberty-enabled Web services environment. This document is intended to provide normative guidance to designers of specific OMA Network Identity Web Services and implementers thereof.

2. References

2.1 Normative References

[3GPP-TR21.905] "3G Vocabulary," TR 21.905, 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org.

[IOPPROC] "OMA Interoperability Policy and Process, Version 1.1". Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL:http://www.openmobilealliance.org/

[Liberty-ProtSchema] "Liberty Protocols and Schema Specification: Version 1.2," URL:http://www.projectliberty.org

[Liberty-BindProf] "Liberty Bindings and Profiles Specification: Version 1.2,". URL:http://www.projectliberty.org

[Liberty-AuthnContext] "Liberty Authentication Context Specification: Version 1.2," URL:http://www.projectliberty.org

[OWSERSpec] "OMA Web Services Enabler (OWSER): Core Specifications, Version 1.0", Open Mobile Alliance™, OMA-OWSER-Core-Specification-V1_0, URL:http://www.openmobilealliance.org

[OWSER NI FF] "OMA Web Services Network Identity Enabler (OWSER NI): Identity Federation Framework", Open Mobile Alliance, URL:http://www.openmobilealliance.org/

[OWSER NI WSF] "OMA Web Services Network Identity Enabler (OWSER NI): Identity Web Services Framework", Open Mobile Alliance, URL:http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels". IETF RFC 2119, S. Bradner. March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[SAMLCore] "Assertions and Protocol for the Oasis Security and Assertions Markup Language (SAML)", OASIS Standard 5 November 2002, URL:http://www.oasis-open.org/committees/download.php/1371/oasis-sstc-saml-core-1.0.pdf

[SAMLBind] "Bindings and Profiles for the Oasis Security and Assertions Markup Language (SAML)", Oasis Standard 5 November 2002, URL:http://www.oasis-open.org/committees/download.php/1372/oasis-sstc-saml-bindings-1.0.pdf

2.2 Informative References

[Liberty-Glossary] "Liberty Architecture Glossary: Version 1.1," January 2003. URL:http://www.projectliberty.org

[Liberty-Overview] "Liberty Architecture Overview: Version 1.1," January 2003. URL:http://www.projectliberty.org

[SESWP] "OMA Service Enabler Strategy White Paper, Version 1.1", Open Mobile Alliance™, OMA-WP-SvcEnablerStrat-V1_1, URL:http://www.openmobilealliance.org/

[OWSER NI AD] "OMA Web Services Network Identity Enabler (OWSER NI): Architecture", Open Mobile Alliance, URL:http://www.openmobilealliance.org/

[OMADictionary-v1.0] "Dictionary for OMA Specifications Version 1.0", Open Mobile Alliance™, OMA-Dictionary-V1_0, URL:http://www.openmobilealliance.org

[RFC2828] "Internet Security Glossary," IETF RFC 2828, R. Shirey, May 2000, URL:http://www.ietf.org/rfc/rfc2828.txt

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendices, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

Clarifications intended to augment some text are shown as follows:

NOTE: This is an example of a note.

Such notes are always informative.

Capitalizing the first letter of a term defined in the “Definitions” section indicates the use of term according to the provided definition.

3.2 Definitions

Account	A formal business agreement for providing regular dealings and services between a Principal and Service Providers. (Source: [Liberty-Glossary])
Assertion	A statement about a Principal.
Attribute	An Attribute is a characteristic that describes a Principal.
Authentication	The process of verifying an Identity claimed by (or for) a Principal.
Authentication Assertion	An Assertion that can be sent from one Identity Provider (or an Identity Broker) to another Provider, which describes a successful Authentication of a Principal. An Authentication Assertion may also contain information such as for how long the Assertion is valid. An Authentication Assertion will also often include an Authentication Context, to notify the Provider what form of Authentication was used.
Authentication Context	The set of parameters (time, location, transaction value, etc.) within which a specific authentication event is acceptable, emphasising that a single authentication event may need to be re-established, perhaps with different mechanisms or classes of mechanisms, when some parameter changes.
Authorisation	A right or permission that is granted to a system Entity to access a system resource, or the process of granting the right or permission [RFC 2828].
Business Agreement	Business agreements are formal agreements (contracts) between parties in the Identity Management Circle of Trust, documenting binding commitments between the parties with respect to aspects such as mutual confidence (e.g. business standards, minimum requirements, certifications and audits supported), risk management (e.g. dissemination of knowledge and use of best practices), liabilities (e.g. defined liability, dispute resolution) and compliance (e.g. general compliance, privacy issues).”
Circle of Trust	One or more service providers and identity providers that have business relationships and operational agreements, and with whom users can transact business in a secure and apparently seamless environment.
De-Federation	A reversal of the process of Federation of two Accounts (belonging to the same Principal), or termination of the state of Identity Federation. De-Federation usually involves an exchange of messages among the systems which established the Identity Federation.
End User	An individual who uses services and content [OMADict]
Federation	The binding of two or more Accounts (within an Authentication Domain or a Circle of Trust, where one of the Accounts is at an IDP) for a given Principal. Federation does not imply that Identity Attributes are being shared – it is simply a joining of two or more Accounts (e.g. for Single Sign On), after which Attributes could then be shared.
Entity	Entity: 1: The information transferred as the payload of a request or response. 2: A distinct component of a service architecture [OMADict]. In this document the term Principal is regularly used as a subset of Entity, more specific to the Entities involved in an Identity Management enabler.

Identifier	A reference that uniquely maps to an Identity. One or more Identifiers are among the characteristics that define an Identity.
Identity	The characteristics by which an Entity or person is recognized or known.
Identity Federation	Associating, connecting, or binding multiple Accounts for a given Principal at various entities within a Circle of Trust. (Source: [Liberty-Glossary])
Identity Provider	A special type of Service Provider role that creates, maintains, and manages Identity information for Principals, and can provide an Authentication Assertion to other Service Providers within an Authentication Domain (or even a Circle of Trust).
Network Identity	The abstraction of the global set of attributes composed from all of a Principal's existing Accounts. (Source: [Liberty-Glossary])
Principal	An entity that has an identity, that is capable of providing consent and other data, and to which authenticated actions are done on its behalf. Examples of principals include an individual end user, a group of end users, a corporation, service enablers / applications, system entities and other legal entities. [OMADict]
Proxy	A computer process that relays a protocol between client and server computer systems, by appearing to the client to be the server and appearing to the server to be the client. (Source: [RFC 2828])
Pseudonym	An arbitrary name assigned by the Identity Provider or Service Provider to identify a Principal to a given relying party, so that the name has meaning only in the context of the relationship between the relying parties.
Service Provider	An Entity that provides services and/or goods to Principals.
Single Log Out	The ability for End Users to properly terminate all open connections, active services or relationships associated with a Single Sign On (SSO) Session, with one logout process.
Single Sign On	The ability to use an Authentication Assertion from one Provider (an Identity Provider or an Identity Broker) at another Provider, in order to ease the burden (for a Principal) of having to authenticate to each Provider separately within a single Session.
Subscriber	A Subscriber is an entity (associated with one or more users) that is engaged in a Subscription with a Service Provider. The subscriber is allowed to subscribe and unsubscribe services, to register a user or a list of users authorized to enjoy these services, and also to set the limits relative to the use that associated users make of these services. (Source: [3GPP-TR21.905])
Subscription	A subscription describes the commercial relationship between the Subscriber and the Service Provider. (Source: [3GPP-TR21.905])
Trust	The extent to which someone that relies on a system can have confidence that the system meets its specifications, i.e., that the system does what it claims to do and does not perform unwanted functions. [source:RFC2828]
WS-Security	WS-Security describes enhancements to SOAP messaging to provide *quality of protection* through message integrity, message confidentiality, and single message authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies.
User Agent	Any software or device that acts on behalf of a user, interacting with other entities and processing resources. (Source: [OMADictionary-v1.0])

3.3 Abbreviations

HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP Secure (aka HTTP over SSL)
MIME	Multipurpose Internet Mail Extensions
LECP	Liberty-enabled Client/Proxy
OASIS	Organization for the Advancement of Structured Information Standards
OMA	Open Mobile Alliance
OWSER	OMA Web Services Enabler
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure

SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol[1]
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
URI	Uniform Resource Identifier
WS	Web Services
WSDL	Web Service Description Language
WSF	Web Services Framework
WSP	Web Service Provider
WSR	Web Service Requester

[1] Note that starting from SOAP Version 1.2, SOAP will no longer be an acronym.

4. Introduction (Informative)

A mobile Subscriber may use multiple services, not all of which belong to the trust domain of its network operator. To facilitate a valuable end user experience, the concept of Identity Federation is required. Network identity is the term used to describe basic functionality that is used with a variety of network services to provide a coherent use of state or data related to an end user. An example of such a service is single sign-on. There are three roles identified in the Network Identity message exchanges defined in this specification: the Principal, the Identity Provider and the Service Provider.

This document, namely the OWSER Network Identity specification, provides normative descriptions of the components needed to provide aspects of the Network Identity related capabilities of the OWSER. This includes Identity Provider Introduction; Identity Federation and Single Sign-On; Name Registration; Authentication Context; Single Signout; and Federation Termination; Service Provider Affiliation; Dynamic Proxying of Identity Providers. Appendix A illustrates these functionalities. This document is intended to provide normative guidance to designers of specific OMA Network Identity Web Services and implementers thereof. Hence, material of a tutorial nature is kept to a minimum.

The choices for the OWSER Network Identity Enabler Release (OSWER NI) have been dictated by requirements arising from use cases that describe capabilities needed to support a natural OMA constituency, namely the mobile network operator (typically playing a role of an Identity Provider in this specification), its Subscribers and third party Service Providers. Thus, this release of the OWSER NI (and therefore this document as well) concentrates on the end user/Subscriber form of the Principal (to enable end-user services such as Single Sign On (SSO)), rather than one representing all the elements of a software system. Web services are relevant as a technology choice for interactions over some of these relationships, but it is has also been necessary to include and describe non-Web services based interactions to better illustrate the context and circumstances under which such Network Identity management capabilities are expected to be used in actual deployments.

In order to support various aspects of Network Identity functionality, profiles of the protocols have been specified and the message flows have been illustrated in Appendix B. Not all of the profiles specified support or require the use of Web services in all of the interactions between the user, the Service Provider and the Identity Provider. Some of these profiles allow some early deployments the advantages of Network Identity functionality to end-user Principals without the need for Web service infrastructure. Which profile is chosen is dictated by the capabilities of the Service Provider and the User Agent and any available network capabilities such as proxies.

5. Network Identity support functions

This section provides normative text for the support of Network Identity related capabilities in OWSER NI, based on references to Liberty Alliance specifications. In particular, the applicable specifications are:

- Liberty Protocols and Schema Specification, version [Liberty ProtSchema], where the Liberty Alliance protocols and messages, as well as associated XML schema, are defined for Identity Federation, single sign-on, name registration, federation termination and single logout.
- Liberty Bindings and Profiles Specification, version [Liberty-BindProf], where the Liberty Alliance bindings and profiles of the Liberty Alliance protocols and messages defined in [Liberty-ProtSchema] are specified.
- Liberty Authentication Context Specification, version 1.1 [Liberty-AuthnContext], which defines a syntax for the definition of authentication context statements and an initial list of Liberty Alliance authentication context classes.

The protocols and messages defined in this specification for Identity Federation, single sign-on, name registration, federation termination and single logout MUST comply with those defined in [Liberty-ProtSchema]. The protocol bindings for the Network Identity features described in this specification MUST comply with the protocol bindings defined in [Liberty-BindProf]. This specification defines the same profiles, i.e., the same combination of message content specification and message transport mechanisms for a single client type, as those defined in [Liberty-BindProf]. The different profiles are defined in the section corresponding to each feature. In particular, the rules with which all profile implementations MUST conform are defined in Section 3.1 of [Liberty-BindProf].

5.1 Identity Provider Introduction

Service Providers that support Network Identity MAY need to know the Identity Provider associated with a Principal. In [Liberty-BindProf] an introduction profile is defined based on the use of a common domain cookie. Whether Identity Providers and Service Providers implement this profile is a deployment issue outside the scope of this specification. However, if this profile is implemented, it MUST comply with the mechanism specified in Section 3.6 of [Liberty-BindProf].

Optionally, a Principal's User Agent that has the appropriate capability MAY have, or know how to obtain, knowledge about the Identity Provider that a Principal wishes to use with a Service Provider. Such a User Agent is referred to as a Liberty Enabled Client/Proxy in [Liberty-BindProf].

A Liberty Enabled Client/Proxy SHOULD convey to a Service Provider that it will be responsible for the determination of the appropriate Identity Provider, and that the Service Provider does not need to be concerned about this. To indicate this, such an entity MUST include an appropriate header in the message sent to the Service Provider. The header MUST be as specified in Section 3.2.5.1 of [Liberty-BindProf].

5.2 Identity Federation and Single Sign On

Identity Federation and single sign-on, as described in this specification, rely on a request/ response protocol by which Identity Federation and single sign-on occur. The Principal initiates the single sign-on by making an HTTP request to the Service Provider. The protocol works as follows:

1. A Service Provider issues a <lib:AuthnRequest> message to an Identity Provider, instructing the Identity Provider to provide an authentication assertion to the Service Provider. Optionally, the Service Provider MAY request that their respective local identities for the Principal be federated.
2. The Identity Provider responds with either a <lib:AuthnResponse> message containing authentication assertions to the Service Provider or an artifact that can be dereferenced into an authentication assertion. Additionally, if the Service Provider requested it, the Identity Provider SHOULD federate the Principal's local identities, respectively at the Identity Provider and the Service Provider.

The Identity Federation and single sign-on protocol MUST conform to that specified in Section 3.2 of [Liberty-ProtSchema].

An Identity Provider facilitates single sign-on and Identity Federation by suitably processing incoming requests and generating responses. The steps indicated refer to the interaction diagram in Section 3.2.1 of [Liberty-BindProf] which

describes the general single sign-on framework. The text that follows assumes that the Principal has already authenticated with the Identity Provider, and that an authenticated session exists for the Principal at the Identity Provider. The establishment of such initial Authentication is separate from the functionality provided by single sign on.

There are two actions required of the Identity Provider:

- In step 5, the Identity Provider MUST process the <lib:AuthnRequest> message according to the rules specified in [Liberty-ProtSchema]. As part of the <lib:AuthnRequest> message, the Service Provider MAY include an indication to request federation of the Principal's local identities at, respectively, the Service Provider and Identity Provider.
- In step 6, the Identity Provider MUST respond to the Principal's User Agent with a <lib:AuthnResponse>, a SAML artifact, or an error. The format of this response depends on the specific binding profile employed by the Identity Provider, as described below.

This specification describes three profiles for the Identity Federation and single sign-on protocol, to cater for the different capabilities of a Principal's User Agent and the availability of any network capabilities such as a Liberty-enabled Proxy:

- Browser artifact profile
- Browser POST profile
- Liberty-enabled Client/Proxy profile

Each of these profiles MUST follow the common interaction and processing rules specified in Section 3.2.1 of [Liberty-BindProf]. Appendix B illustrates the message exchanges for each of these profiles.

5.2.1 Browser artifact profile

The browser artifact profile relies on the use of an authentication artifact (a SAML artifact), which the Service Provider must dereference to an authentication assertion from the Identity Provider to determine whether the Principal is authenticated. This profile, which is an adaptation of the "Browser/artifact profile" for SAML as documented in [SAMLBind] [SAMLCore], MUST be implemented as specified in Section 3.2.2 of [Liberty-BindProf].

The support of this profile is mandatory in Service Providers and Identity Providers implementing Network Identity features. The requirements in this section MUST be implemented according to the relevant sections of [Liberty-BindProf] as indicated later in the text.

Figure 2 of [Liberty-BindProf] describes the browser artifact profile for single sign-on. The Principal initiates the single sign-on by making an HTTP request to the Service Provider, as indicated in Step 1 in Section 3.2.1 of [Liberty-BindProf]. The Service Provider then proceeds as follows:

- The Service Provider obtains the address of the appropriate Identity Provider (see section 5.1).
- The Service Provider MUST respond to the Principal's User Agent with an HTTP redirect, including a <lib:AuthnRequest> message, as specified in step 3 of Section 3.2.2.1 of [Liberty-BindProf].
- After obtaining a SAML artifact from the Identity Provider, the Service Provider MUST send a <samlp:Request> SOAP message to the Identity Provider's SOAP endpoint as specified in step 8 of Section 3.2.1 of [Liberty-BindProf].
- The Service Provider MUST process the <samlp:Assertion> returned by the Identity Provider as specified in step 10 of Section 3.2.1 of [Liberty-BindProf]

The Identity Provider must complete two processing steps to implement this feature: processing an authentication request, and processing a SAML request. The interaction to request the authentication assertion proceeds as follows:

- The Identity Provider MUST process the <lib:AuthnRequest> message as specified in step 5 of Section 3.2.1 of [Liberty-BindProf].

- In response to the <lib:AuthnRequest> the Identity Provider MUST perform an HTTP redirection, including the SAML artifact, as specified in step 6 of Section 3.2.2.1 of [Liberty-BindProf].
- The Identity Provider MUST process the <samlp:Request> produced by the Service Provider in Step 8 of the single sign-on interaction, and MUST produce a <samlp:Response> as specified in step 9 of Section 3.2.1 of [Liberty-BindProf].
- The artifact produced by the Identity Provider MUST be formatted as specified in Section 3.2.2.2 of [Liberty-BindProf].

5.2.2 Browser POST profile

The browser POST profile allows the presentation of authentication assertions to Service Providers without the use of an artifact. This profile, which is an adaptation of the "Browser/post profile" for SAML as documented in [SAMLBind], MUST be implemented as specified in Section 3.2.3 of [Liberty-BindProf].

The support of this profile is optional in both Service Providers and Identity Providers. While the support of this profile is not mandatory in Identity Providers, Identity Providers SHOULD support this profile in order to be interoperable with Service Providers that use this profile. The requirements in this section must be implemented according to the relevant sections of [Liberty-BindProf] as indicated later in the text.

Figure 3 of [Liberty-BindProf] describes the browser POST profile for single sign-on. The Principal initiates the single sign-on by making an HTTP request to the Service Provider, as indicated in step 1 of Section 3.2.1 of [Liberty-BindProf]. The Service Provider then proceeds as follows:

- The Service Provider obtains the address of the appropriate Identity Provider. (see Section 5.1)
- The Service Provider MUST respond to the Principal's User Agent with an HTTP redirect, including a <lib:AuthnRequest> message, as specified in step 3 of Section 3.2.3 of [Liberty-BindProf].
- After obtaining an authentication assertion, the Principal's User Agent MUST issue an HTTP POST request containing the <lib:AuthnResponse> to the Service Provider, which MUST be processed by the Service Provider as specified in step 10 of Section 3.2.1 of [Liberty-BindProf].

At the Identity Provider, the interaction to request the authentication assertion proceeds as follows:

- The Identity Provider MUST process the <lib:AuthnRequest> message as specified in step 5 of Section 3.2.1 of [Liberty-BindProf].
- The Identity Provider generates an HTTP 200 response containing an authentication response <lib:AuthnResponse>. This response MUST conform to the specification in step 6 of Section 3.2.3 of [Liberty-BindProf].

5.2.3 Liberty-enabled Client/Proxy (LECP) profile

The Liberty Enabled Client/Proxy (LECP) profile specifies interactions between Liberty enabled User Agent and/or proxies, Service Providers, and Identity Providers. A Liberty-enabled Client is a Principal's User Agent that has, or knows how to obtain, knowledge about the Identity Provider that the Principal wishes to use with a Service Provider for purposes of enabling Network Identity based services such as single sign on. In addition, a Liberty-enabled Client receives and sends messages implementing Network Identity protocols in the body of HTTP requests and responses. Therefore, Liberty-enabled Clients have no restrictions, when compared to browser-based User Agents, on the size of the protocol messages.

All Liberty-enabled Clients, in addition to meeting the common requirements for profiles in Section 3.1 of [Liberty-BindProf], MUST indicate that they are a Liberty-enabled Client either by including a 'Liberty-Enabled' HTTP header or an entry in the value of the HTTP User-Agent header for each HTTP request they make. The preferred method is the Liberty-Enabled header. Thus, a Liberty-enabled client SHOULD indicate this capability by including a "Liberty-Enabled" HTTP header. The formats of the Liberty-Enabled HTTP header and HTTP User-Agent header entry are defined in Section 3.2.5.1 of [Liberty-BindProf].

The Liberty-enabled Client/Proxy (LECP) profile in this specification MUST be implemented as specified in Section 3.2.5 of [Liberty-BindProf]. Figure 5 of [Liberty-BindProf] describes the LECP profile for single sign-on. The support of the LECP profile is mandatory in Service Providers and Identity Providers implementing Network Identity based capabilities. The requirements in this section must be implemented according to the relevant sections of [Liberty-BindProf] as indicated later in the text.

The Principal initiates the single sign-on by making an HTTP request to the Service Provider, as indicated in Step 1 in Section 3.2.5.2 of [Liberty-BindProf]. The Principal's User Agent will submit a request to the Service Provider, which contains the required Liberty-enabled indication. At the Service Provider:

- A Service Provider receiving the indication that the Principal's User Agent is Liberty enabled MUST NOT obtain an Identity Provider address or perform Identity Provider introduction.
- The Service Provider MUST issue an HTTP 200 OK response to the Principal's User Agent. The response MUST follow the specifications in step 3 of Section 3.2.5.2 of [Liberty-BindProf]. The Service Provider SHOULD place appropriate headers in the response to ensure the response is not cached as specified in step 3 of Section 3.2.5.2 of [Liberty-BindProf].

At the Identity Provider:

- The Identity Provider MUST process the <lib:AuthnRequest> in the body of the SOAP POST message from the Liberty Enabled Client/Proxy as specified in step 5 of Section 3.2.1 of [Liberty-BindProf].
- The Identity Provider MUST respond to the <lib:AuthnRequest> with a HTTP 200 OK response as specified in step 6 of Section 3.2.5.2 of [Liberty-BindProf], with the correct MIME type (application/vnd.liberty-response+xml) and Liberty-Enabled HTTP header (see Section 3.2.5.1 of [Liberty-BindProf]). This response MUST contain one <lib:AuthnResponseEnvelope> in the body of a SOAP message as specified in [Liberty-ProtSchema].

After obtaining an authentication assertion, the Principal's User Agent will issue an HTTP POST request containing the <lib:AuthnResponse> to the Service Provider, as specified in step 7 of Section 3.2.5.2 of [Liberty-BindProf].

- The Service Provider MUST process the <lib:AuthnResponse> received from the Principal's User Agent in the HTTP POST as specified in step 10 of Section 3.2.1 of [Liberty-BindProf].

5.2.4 Affiliation

An affiliation MUST be identified by a URI based identifier according to the description in Section 3.1.3 of [Liberty-ProtSchema].

When Single Sign-On and Identity Federation occurs between an Identity Provider and an affiliation in which the service provider is a member, the Single Sign-On and Federation protocol as described in Section 3.2 of [Liberty-ProtSchema] MUST be followed. In an authentication request, when the Service Provider wishes to indicate that it is acting as a member of an affiliation, it MUST include the <AffiliationID> element within the <AuthnRequest> element as described in Section 3.2.1.1 of [Liberty-ProtSchema]. The processing rules MUST follow the description in Section 3.2.2.6 of [Liberty-ProtSchema].

When the Principal terminates an identity federation between an Identity Provider and an affiliation in which the service provider is a member, the Federation Termination Notification protocol as described in Section 3.4 of [Liberty-ProtSchema] MUST be followed.

When a Principal initiates single logout by logging out at a Service Provider or at an Identity Provider, the Principal is then logged out of all sessions authenticated by this Identity Provider, including sessions with affiliations. The Single Logout protocol as described in Section 3.5 of [Liberty-ProtSchema] MUST be followed.

When a Service Provider requires a name identifier for a Principal with which it has an identity federation relationship, but which references an identity federation between the identity provider and another service provider, it can use the Name Identifier Mapping protocol to obtain such an identifier. Either of the Service Providers may belong to an affiliation. When

the Name Identifier Mapping protocol is used, the procedures described in Section 3.6 of [Liberty-ProtSchema] MUST be followed.

Below, we include an example of an <AuthnRequest> where the Service Provider (SP) with ProviderID `http://OWSERCompatibleSP.com` is acting as a member of the affiliation with affiliationID `http://OWSERCompatibleAffiliation.com`. The presence of the optional AffiliationID element indicates that the SP is acting as a member of the affiliation.

```
<lib:AuthnRequest RequestID="1pY6tWugT8Vz+L8+rURp51oFX6rt" MajorVersion="1" MinorVersion="2"
consent="urn:liberty:consent:obtained" IssueInstant="2005-03-24T21:42:4Z"
xmlns:lib="urn:liberty:iff:2003-08">
   <ds:Signature> . .. </ds:Signature>
   <lib:ProviderID>http://OWSERCompatibleSP.com</lib:ProviderID>
   <lib:AffiliationID>http://OWSERCompatibleAffiliation.com</lib:AffiliationID>
   <lib:NameIDPolicy>federate</lib:NameIDPolicy>
   <lib:ForceAuthn>false</lib:ForceAuthn>
   <lib:IsPassive>false</lib:IsPassive>
   <lib:ProtocolProfile>http://projectliberty.org/profiles/brws-post</lib:ProtocolProfile>
   <lib:RequestAuthnContext>
      <lib:AuthnContextClassRef>http://projectliberty.org/schemas/authctx/classes/Password-
   ProtectedTransport</lib:AuthnContextClassRef>
      <lib:AuthnContextComparison>exact</lib:AuthnContextComparison>
   </lib:RequestAuthnContext>
   <lib:RelayState>Yu8IODlhcgGSUitRAA8UhbMmCZtuYalPA2gh</lib:RelayState>
   <lib:Scoping>
      <lib:ProxyCount>1</lib:ProxyCount>
   </lib:Scoping>
</lib:AuthnRequest>
```

5.2.5 Dynamic Proxying of identity Providers

Dynamic proxying of Identity Providers enables an Identity Provider receiving an authentication request requesting authentication of a Principal, to proxy the authentication request to another Identity Provider that may have already authenticated the Principal. Dynamic proxying of Identity Providers MUST follow the mechanism specified in Section 3.2.2.7 of [Liberty-ProtSchema].

5.3 Name Registration

During Identity Federation, the Identity Provider generates an opaque handle that serves as the initial name identifier that both the Service Provider and the Identity Provider will use to refer to the Principal when communicating with each other to provide federated Network Identity based capabilities such as single sign on. This name identifier is termed the <lib:IdPProvidedNameIdentifier>.

At some later point following Identity Federation, either the Service Provider or the Identity Provider may initiate the name registration protocol described in this section and register a new name identifier for a Principal with each other. Subsequent to federation, the Identity Provider may choose to register a new <lib:IdPProvidedNameIdentifier>. Additionally, a Service Provider may register a different opaque handle, termed the <lib:SPProvidedNameIdentifier>, with the Identity Provider. After a Service Provider's name registration, the Identity Provider MUST use the <lib:SPProvidedNameIdentifier> to refer to the Principal when communicating with the Service Provider. Note that the fact that the SP or the IdP can initiate a name registration implies a mandatory support of its reception at, respectively, the IdP and the SP to achieve interoperability in every deployment scenario.

When Name Registration is used, the SP and IdP MUST comply with the procedures specified in Section 3.3 of [Liberty-ProtSchema]. Its implementations MUST use the <lib:RegisterNameIdentifierRequest> and

<lib:RegisterNameIdentifierResponse> messages defined in [Liberty-ProtSchema] for this purpose and conform to the rest of the mandatory statements below.

Name registration may be initiated either at the Identity Provider or at the Service Provider. This specification describes two profiles for Name Registration:

- HTTP redirect-based
- SOAP/HTTP-based

The profiles described in this specification MUST be implemented as specified in Section 3.3 of [Liberty-BindProf]. Service Providers and Identity Providers MUST support the SOAP/HTTP-based profile, and optionally support the HTTP redirect based profile. While the support of the HTTP redirect based profile is optional, Identity Providers SHOULD support this profile in order to be interoperable with Service Providers that use this profile. When Name Registration is initiated at the Identity Provider, the mechanism specified in Section 3.3.1 of [Liberty-BindProf] MUST be followed. When Name Registration is initiated at the Service Provider, the mechanism specified in Section 3.3.2 of [Liberty-BindProf] MUST be followed.

The actions and processing steps specified below are symmetric when the Name Registration is initiated by either the Identity Provider or the Service Provider. Only the actions and processing steps for Identity Provider initiated name registration are described below. The corresponding actions and processing steps for the Service Provider initiated Name Registration may be obtained by replacing “Identity Provider” by “Service Provider” and vice-versa in the description below, by replacing IdPProvidedNameIdentifier with SPProvidedNameIdentifier and by changing the references to sections 3.3.1.x to 3.3.2.x.

Appendix B illustrates the message exchanges for each of these profiles.

5.3.1 Name Registration initiated at Identity Provider

When Name Registration is initiated at the Identity Provider, the mechanism specified in Section 3.3.1 of [Liberty-BindProf] MUST be followed.

5.3.1.1 Processing at the Identity Provider

5.3.1.1.1 HTTP redirect-based profile

This section specifies the actions/processing at the Identity Provider for the HTTP redirect-based Name Registration initiated at the Identity Provider, based on the interaction described in Section 3.3.1.1 of [Liberty-BindProf]. Note that the timing and mechanism of the initiation of this interaction are not normatively specified, although [Liberty-BindProf] offers some examples. The Identity Provider MUST initiate HTTP redirect-based Name Registration only when the Service Provider metadata specifies the appropriate URI identifier as specified in Section 3.3.1.1 of [Liberty-BindProf].

- The Identity Provider MUST redirect the Principal’s User Agent to the Name Registration service at the Service Provider as specified in Step 2 of Section 3.3.1.1 of [Liberty-BindProf].

5.3.1.1.2 SOAP/HTTP-based profile

The Identity Provider MUST initiate SOAP/HTTP-based Name Registration only when the Service Provider metadata specifies the appropriate URI identifier as specified in Section 3.3.1.2 of [Liberty-BindProf].

- The SOAP/HTTP-based Name Registration transactions MUST use the SOAP Binding for Liberty as defined in Section 2.1 of [Liberty-BindProf].
- The Identity Provider MUST initiate the Name Registration transaction by sending a <lib:RegisterNameIdentifierRequest> message to the Service Provider’s SOAP end-point as specified in step 1 of Section 3.3.1.2 of [Liberty-BindProf].
- The Identity Provider MUST process the <lib:RegisterNameIdentifierResponse> from the Service Provider as specified in Section 3.3.3 of [Liberty-ProtSchema].

5.3.1.2 Processing at the Service Provider

5.3.1.2.1 HTTP redirect-based profile

- The Service Provider MUST process the <lib:RegisterNameIdentifierRequest> from the Identity Provider as specified in Section 3.3.3 of [Liberty-ProtSchema] and step 4 of Section 3.3.1.1 of [Liberty-BindProf].
- The Service Provider MUST respond to the Identity Provider with a redirection URL as specified in the RegisterNameIdentifierServiceReturnURL metadata element described in Section 4 of [Liberty-ProtSchema]. The redirection MUST comply with the rules specified in step 5 of Section 3.3.1.1 of [Liberty-BindProf].

5.3.1.2.2 SOAP/HTTP-based profile

- The SOAP/HTTP-based Name Registration interactions MUST use the SOAP Binding for Liberty as defined in Section 2.1 of [Liberty-BindProf].
- When used, the Identity Provider MUST send a <lib:RegisterNameIdentifierRequest> message to the Service Provider. The Service Provider MUST record the new <lib:IDPProvidedNameIdentifier>.
- After a successful registration of the <lib:IDPProvidedNameIdentifier>, the Service Provider MUST respond with a <lib:RegisterNameIdentifierResponse> according to the processing rules in Section 3.3.3 of [Liberty-ProtSchema].

5.4 Authentication Context

Authentication context is defined as the information additional to the authentication assertion itself that the Service Provider may require before it makes a decision regarding what services the subject of the authentication assertion should be allowed to access. Such information could include authentication mechanism, mechanisms for storing and protecting credentials, initial user identification mechanisms, etc.

In order to simplify for Service Providers the task of assessing and comparing authentication assertions [Liberty-AuthnContext] defines authentication contexts classes that are representative of current technologies and practices. For instance, a typical authentication context will be when a Principal uses a self-chosen password over a server-authenticated SSL session to authenticate to an Identity Provider.

The Identity Provider and the Service Provider MUST accept the inclusion of a <lib:AuthnContext> in, respectively, a <lib:AuthnRequest> and <lib:AuthnResponse> message. When a Service Provider wishes to request a specific authentication context from an Identity Provider, it MUST include a <lib:AuthnContext> element in the <lib:AuthnRequest> message that it sends to the Identity Provider. When an Identity Provider wishes to convey a specific authentication context to a Service Provider, it MUST include a <lib:AuthnContext> element in the <lib:AuthnResponse> message that it sends to the Service Provider.

[Liberty-AuthnContext] defines the mandatory syntax for the definition of authentication context statements and an initial list of authentication context classes.

5.5 Single Sign-Out

When the Principal invokes the single logout process at a Service Provider, the Service Provider MUST send a `<lib:LogoutRequest>` message to the Identity Provider that provided the Authentication service for the session.

When either the Principal invokes a logout at the Identity Provider or a Service Provider sends a logout request to the Identity Provider specifying that Principal, the Identity Provider MUST send a `<lib:LogoutRequest>` message to each Service Provider to which it provided authentication assertions in the current session with the Principal, with the exception of the Service Provider that sent the `<lib:LogoutRequest>` message to the Identity Provider. Upon receiving a `<lib:LogoutRequest>` message, the responding provider MUST return a `<lib:LogoutResponse>` message.

The Single Signout mechanism MUST comply with the procedures specified in Section 3.5 of [Liberty-ProtSchema]. The schema fragment for `<lib:LogoutRequest>` is specified in Section 3.5.1.1 of [Liberty-ProtSchema]. The schema fragment for `<lib:LogoutResponse>` is specified in Section 3.5.2.1 of [Liberty-ProtSchema].

Single signout may either be initiated at the Identity Provider or at the Service Provider. When single signout is initiated at the Identity Provider, three profiles are specified:

- HTTP redirect-based
- HTTP GET-based
- SOAP/HTTP-based

When single sign-out is initiated at the Service Provider, two profiles are specified:

- HTTP redirect-based
- SOAP/HTTP-based

The profiles described in this specification are as specified in Section 3.5 of [Liberty-BindProf]. Service Providers and Identity Providers MUST support the SOAP/HTTP-based profile, and optionally support the other profiles. While the support of the HTTP-redirect and HTTP GET based profiles is optional in an Identity Provider, Identity Providers SHOULD support this profile in order to be interoperable with Service Providers that use this profile. Appendix B illustrates the message exchanges for each of these profiles.

Upon initiation of single signout, the Identity Provider MUST terminate the Principal's current session, and any further authentication assertions for the Principal MUST NOT be given to Service Providers.

5.5.1 Single signout initiated at the Identity Provider

5.5.1.1 HTTP redirect-based profile

- This interaction MUST NOT be used unless the Service Provider metadata element SingleLogOutProtocolProfile specifies the URI http://projectliberty.org/profiles/slo-idp-http
- In response to the Principal's logout request, the Identity Provider MUST redirect the Principal's User Agent to the single logout service URL at each Service Provider to whom the Identity Provider has provided an authentication assertion during the Principal's current session. Each redirection MUST comply with the rules specified in Step 1 of Section 3.5.1.1.1 of [Liberty-BindProf].
- After receiving the request from the Principal's User Agent to the SingleLogoutServiceReturnURL (described in Section 4 of [Liberty-ProtSchema]) as specified in the Identity Provider metadata, the Identity Provider MUST process the request and send an HTTP response to the Principal's User Agent confirming that the requested action of a single logout has been completed.

5.5.1.2 HTTP GET-based profile

- This interaction must not be used unless the Service Provider metadata element SingleLogOutProtocolProfile specifies the URI http://projectliberty.org/profiles/slo-idp-http
- In response to the Principal's request for a logout, the Identity Provider MUST respond with a HTTP 200 response containing image tags referencing the logout service URL for each of the Service Providers to whom the Identity Provider has provided an authentication assertion during the Principal's current session. Each image tag MUST comply with the rules specified in Step 2 of Section 3.5.1.1.2 of [Liberty-BindProf].
- After receiving the request from the Principal's User Agent at the SingleLogoutServiceReturnURL (described in Section 4 of [Liberty-ProtSchema]) as specified in the Identity Provider metadata, the Identity Provider MUST process the request and send an HTTP response to the Principal's User Agent confirming that the requested action of a single logout has been completed.

5.5.1.3 SOAP/HTTP-based profile

- This interaction MUST NOT be used unless the Service Provider metadata element SingleLogOutProtocolProfile specifies the URI http://projectliberty.org/profiles/slo-idp-soap
- In response to a HTTP 200 OK with a SOAP <lib:LogoutRequest> message from the Service Provider, the Identity Provider MUST send an HTTP response confirming that the requested action of a single logout has completed.

5.5.2 Single signout initiated at the Service Provider

5.5.2.1 HTTP redirect-based profile

- When used, the Principal's User Agent MUST access the Service Provider's single logout service URL. The Service Provider's single logout service URL redirects the User Agent to the single logout service URL at the Identity Provider. The Identity Provider MUST process the <lib:LogoutRequest> according to the rules defined in Section 3.5.1 of [Liberty-ProtSchema].
- The Identity Provider MUST notify each Service Provider for which the Identity Provider has provided authentication assertions of the logout request using the Service Provider's preferred profile, as specified in Step 4 of Section 3.5.2.1 of [Liberty-BindProf].
- The Identity Provider MUST respond and redirect the Principal's User Agent back to the Service Provider using the return URL location obtained from the SingleLogoutServiceReturnURL metadata element (described in Section 4 of [Liberty-ProtSchema]) as specified in Step 5 of Section 3.5.2.1 of [Liberty-BindProf].

5.5.2.2 SOAP/HTTP-based profile

- After receiving a < lib:LogoutRequest> from the Service Provider, the Identity Provider MUST process it according to the rules in Section 3.5.1 of [Liberty-ProtSchema].
- The Identity Provider MUST submit to each Service Provider for which the Identity Provider has provided authentication assertions during the Principal's current session a request to logout the Principal as specified in Step 3 of Section 3.5.2.2 of [Liberty-BindProf].
- The Identity Provider MUST respond to the <lib:LogoutRequest> with a HTTP 200 OK containing a SOAP <lib:LogoutResponse> message as specified in Step 4 of Section 3.5.2.2 of [Liberty-BindProf].

5.6 Federation Termination Notification

The Federation Termination Notification protocol, as specified in Section 3.4 of [Liberty-ProtSchema], MUST be used when a Principal terminates an Identity Federation between a Service Provider and an Identity Provider. There are four variations of the Federation Termination Notification interaction: the Federation Termination Notification interaction can be initiated by either the Identity Provider or the Service Provider, and the protocol bindings are based on either HTTP redirect feature or SOAP/HTTP message exchanges. Service Providers and Identity Providers MUST support the SOAP/HTTP-based profile, and optionally support the HTTP redirect based profile. While the support of the HTTP redirect based profile is optional, Identity Providers SHOULD support this profile in order to be interoperable with Service Providers that use this profile. All four interactions are as specified in Section 3.4 of [Liberty-BindProf]. Appendix B illustrates the message exchanges for each of these profiles.

The actions and processing steps specified below are symmetric when the federation termination notification is initiated by either the Identity Provider or the Service Provider. Only the Identity Provider actions and processing steps are described in the text below. The corresponding actions and processing steps for the Service Provider initiated federation termination notification are obtained by replacing in the description below "Identity Provider" by "Service Provider" and vice-versa, and by changing the references from sections 3.4.1.x to 3.4.2.x.

5.6.1 Federation Termination Notification initiated at the Identity Provider

5.6.1.1 Processing at the Identity Provider

5.6.1.1.1 HTTP redirect

- This profile MUST NOT be used unless the Service Provider metadata element FederationTerminationNotificationProtocolProfile specifies the URI http://projectliberty.org/profiles/fedterm-idp-http.
- This profile requires that certain preconditions specified in Section 3.4.1.1 of [Liberty-BindProf] MUST be satisfied.
- In response to a request to the Identity Provider's federation termination service URL, the Identity Provider MUST redirect the Principal's User Agent to the federation termination service at the Service Provider. This redirection MUST comply with the rules specified in step 2 of Section 3.4.1.1 of [Liberty-BindProf].

5.6.1.1.2 SOAP/HTTP

- This profile MUST NOT be used unless the Service Provider metadata element FederationTerminationNotificationProtocolProfile specifies the URI http://projectliberty.org/profiles/fedterm-idp-soap.
- This profile requires that certain preconditions specified in Section 3.4.1.2 of [Liberty-BindProf] MUST be satisfied.
- In response to a federation termination request from the Principal's User Agent to the Identity Provider's federation termination service URL, the Identity Provider MUST send a SOAP/ HTTP notification message to the Service Provider's SOAP endpoint. The SOAP message MUST comply with the rules specified in step 2 of Section 3.4.1.2 of [Liberty-BindProf].

The Service Provider will respond to termination notification with a HTTP 204 OK response.

- The Identity Provider MUST process the HTTP 204 response from the Service Provider and send an HTTP response confirming the requested action of federation termination with the specified Service Provider.

5.6.1.2 Processing at the Service Provider

5.6.1.2.1 HTTP redirect

- The HTTP-Redirect based Federation Termination Notification (initiated at the Identity Provider) MUST be supported by the Service Provider.
- The Service Provider MUST process the <lib:FederationTerminationNotification> received from the Principal's User Agent according to the rules defined in section 3.4.2 of [Liberty-ProtSchema] and in Step 4 of Section 3.4.1.1 of [Liberty-BindProf].
- The Service Provider's federation termination service MUST respond by redirecting the Principal's User Agent as specified in Step 5 of Section 3.4.1.1 of [Liberty-BindProf].

5.6.1.2.2 SOAP/HTTP

-
- The Service Provider MUST process the <lib:FederationTerminationNotification> in the SOAP message received from the Identity Provider according to the rules defined in Section 3.4.2 of [Liberty-ProtSchema] and in step 3 of Section 3.4.1.2 of [Liberty-BindProf].
- The Service Provider MUST respond to the <lib:FederationTerminationNotification> with a HTTP 204 OK response as in step 4 of Section 3.4.1.2 of [Liberty-BindProf].

5.7 Security Considerations

The same security considerations that can be found in section 4 of [Liberty-BindProf] apply to this specification. When SSL/TLS is needed, it MUST comply with Section 7.1.2.1 of [OWSRSpec]. [Liberty-ProtSchema] and [Liberty-BindProf], specifications that this version of Network Identity specification is based on, does not specify the usage of SOAP message level security. However, when SOAP message level security is needed, it SHOULD comply with Section 7.1.2.2 of [OWSRSpec]. When XML Signatures are used, it MUST comply with Section 7.1.2.3.1 of [OWSRSpec] and Section 3.1 of [Liberty-ProtSchema]. When XML Encryption is used, it MUST comply with Section 7.1.2.3.2 of [OWSRSpec].

Appendix A. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [CREQ].

A.1 IdP

Item	Function	Reference	Status	Requirement
NI-IDP-001	Common Domain Cookie Introduction Protocol	5.1	O	
NI-IDP-002	Single Sign-on & Federation	5.2	M	NI-IDP-003 AND NI-IDP-005 AND NI-IDP-023
NI-IDP-003	Browser Artifact Profile	5.2.1	O	OWSER-All-002
NI-IDP-004	Browser POST Profile	5.2.2	O	
NI-IDP-005	LECP Profile	5.2.3	O	OWSER-All-002
NI-IDP-007	Name Registration Initiated at IdP	5.3.1	O	
NI-IDP-008	Name Registration Initiated at SP	5.3.1	M	NI-IDP-010
NI-IDP-009	HTTP redirect based Name Registration	5.3.1.1.1	O	
NI-IDP-010	SOAP based Name Registration	5.3.1.1.2	M	OWSER-All-002
NI-IDP-011	Authentication Context Classes	5.4	O	
NI-IDP-012	Single Sign-out	5.5	M	NI-IDP-013 AND NI-IDP-014
NI-IDP-013	Single Sign-out initiated at the IdP	5.5.1	O	NI-IDP-017
NI-IDP-014	Single Sign-out initiated at the SP	5.5.2	O	NI-IDP-017
NI-IDP-015	HTTP redirect based Single Sign-out	5.5.1.1 5.5.2.1	O	
NI-IDP-016	HTTP GET based Single Sign-out	5.5.1.2	O	
NI-IDP-017	SOAP based Single Sign-out	5.5.1.3 5.5.2.2	O	OWSER-All-002
NI-IDP-018	Federation Termination Notification	5.6	M	NI-IDP-019 AND NI-IDP-020
NI-IDP-019	Federation Termination Notification initiated at the IdP	5.6.1	O	NI-IDP-022
NI-IDP-020	Federation Termination Notification initiated at the SP	5.6.1	O	NI-IDP-022
NI-IDP-021	HTTP redirect based Federation Termination Notification	5.6.1.1.1 5.6.1.2.1	O	
NI-IDP-022	SOAP based Federation Termination Notification	5.6.1.1.2 5.6.1.2.2	O	OWSER-All-002
NI-IDP-023	Affiliation	5.2.4	M	
NI-IDP-024	Dynamic proxying of IdP	5.2.5	M	

Table 1: SCR for IdP

A.2 SP

Item	Function	Reference	Status	Requirement
NI-SP-001	Common Domain Cookie Introduction Protocol	5.1	O	
NI-SP-002	Single Sign-on & Federation	5.2	M	NI-SP-003 AND NI-SP-005 AND NI-SP-023
NI-SP-003	Browser Artifact Profile	5.2.1	O	OWSER-All-002
NI-SP-004	Browser POST Profile	5.2.2	O	
NI-SP-005	LECP Profile	5.2.3	O	OWSER-All-002
NI-SP-007	Name Registration Initiated at IdP	5.3.1	M	NI-SP-010
NI-SP-008	Name Registration Initiated at SP	5.3.1	O	
NI-SP-009	HTTP redirect based Name Registration	5.3.1.1.1	O	
NI-SP-010	SOAP based Name Registration	5.3.1.1.2	M	OWSER-All-002
NI-SP-011	Authentication Context Classes	5.4	O	
NI-SP-012	Single Sign-out	5.5	M	NI-SP-013 AND NI-SP-014
NI-SP-013	Single Sign-out initiated at the IdP	5.5.1	O	NI-SP-017
NI-SP-014	Single Sign-out initiated at the SP	5.5.2	O	NI-SP-017
NI-SP-015	HTTP redirect based Single Sign-out	5.5.1.1 5.5.2.1	O	
NI-SP-016	HTTP GET based Single Sign-out	5.5.1.2	O	
NI-SP-017	SOAP based Single Sign-out	5.5.1.3 5.5.2.2	O	OWSER-All-002
NI-SP-018	Federation Termination Notification	5.6	M	NI-SP-019 AND NI-SP-020
NI-SP-019	Federation Termination Notification initiated at the IdP	5.6.1	O	NI-SP-022
NI-SP-020	Federation Termination Notification initiated at the SP	5.6.1	O	NI-SP-022
NI-SP-021	HTTP redirect based Federation Termination Notification	5.6.1.1.1 5.6.1.2.1	O	
NI-SP-022	SOAP based Federation Termination Notification	5.6.1.1.2 5.6.1.2.2	O	OWSER-All-002
NI-SP-023	Affiliation	5.2.4	M	
NI-SP-024	Dynamic proxying of IdP	5.2.5	O	

Table 2: SCR for SP

A.3 LECP

Item	Function	Reference	Status	Requirement
NI-LECP-001	Single Sign-on & Federation	5.2	M	NI-LECP-002
NI-LECP-002	LECP Profile	5.2.3	O	OWSER-All-002

Table 3: SCR for LECP

Appendix B. Service Provider and Identity Provider Message Exchanges (Informative)

The Network Identity protocols and profiles described in this specification include exchanges that correspond to an exchange between a Service Provider (SP) and an Identity Provider (IdP). In this appendix, we illustrate some of these exchanges. This section is non-normative and used for illustrative purposes only.

Figure 1 below indicates a generic message exchange between a Service Provider (SP) and an Identity Provider (IdP). The SP sends a request message to the IdP (shown as message 1 in **Figure 1**), in response to which the IdP sends a response message back to the SP (shown as message 2 in **Figure 1**).

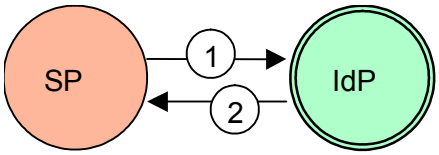

Figure 1: Generic Message Exchange between SP – IdP

In the rest of this section, we will map portions of the protocols and profiles described in this specification to messages 1 and 2.

B.1 Identity Federation

As described in Section 5.2, Identity Federation is the means by which an end user's otherwise distinct Service Provider and Identity Provider Accounts are linked. The Service Provider sends an <lib:AuthnRequest> message to the Identity Provider, in response to which the Identity Provider sends a <lib:AuthnResponse> message back to the Service Provider. In the Liberty Enabled Client/Proxy profile described in Section 5.2.3, the Liberty Enabled Client/Proxy sends an <lib:AuthnRequest> message to the Identity Provider, and the Identity Provider responds with an <lib:AuthnResponse> message. Proper instantiation of these messages enables Identity Federation.

B.2 Single Sign On

As described in Section 5.2, Single sign-on is the means by which an end user can authenticate once with an Identity Provider and then continue to access resources at a Service Provider (that has a trust relationship with the Identity Provider) without authenticating again. Single sign-on builds on Identity Federation.

In the Browser-Artifact profile described in Section 5.2.1, a <samlp:Request> message is sent by the Service Provider to the Identity Provider in order to dereference the artifact to an authentication assertion, and a corresponding <samlp:Response> message is sent by the Identity Provider back to the Service Provider.

In the Liberty Enabled Client/Proxy profile described in Section 5.2.3, the Liberty Enabled Client/Proxy sends an <lib:AuthnRequest> message to the Identity Provider, and the Identity Provider responds with an <lib:AuthnResponse> message.

B.3 Name Registration

Name registration was described in Section 5.3. At the time of federation, the Identity Provider generates an opaque handle that serves as the name identifier the Service Provider and the Identity Provider use in referring to the Principal when

communicating with each other. This name identifier is termed the IdP Provided Name Identifier. Either the Service Provider, or the Identity Provider may register a new name identifier for a Principal with each other at any time following federation. The protocol used for this purpose is the name registration protocol.

When a Service Provider wishes to register a new name identifier for a Principal, the Service Provider initiates the interaction with the Identity Provider. Similarly, when an Identity Provider wishes to register a new name identifier for a Principal, the Identity Provider initiates the interaction with the Service Provider.

B.4 Single Sign-Out

As seen in **Figure 2**, the Identity Provider (IdP) sends a <lib:LogoutRequest> to each Service Provider where the user needs to be signed out, and the Service Providers (SP1, SP2) respond with a <lib:LogoutResponse> back to the Identity Provider.

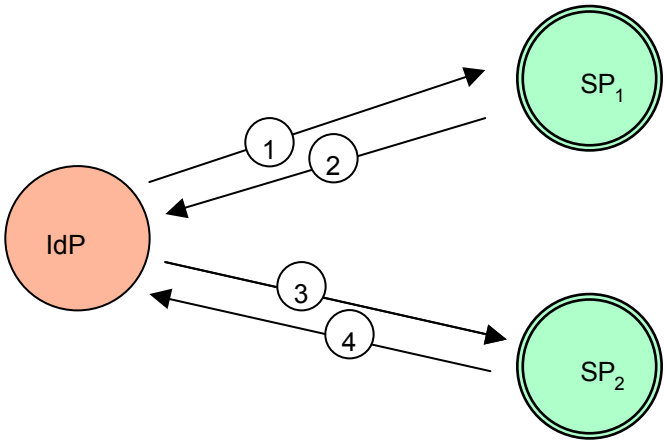

Figure 2: Single Sign-Out Message Exchange

B.5 Federation Termination

Figure 3 indicates the message exchange for federation termination initiated at the Identity Provider. As described in Section 5.6, the Federation Termination protocol is used when a Principal terminates an Identity Federation between a Service Provider and an Identity Provider.

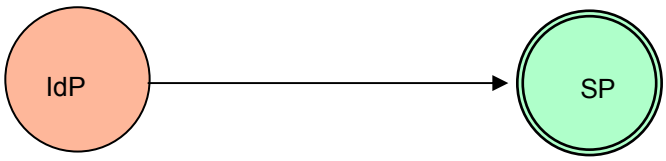

Figure 3: Federation Termination Message Exchange

Appendix C. Illustration of Profiles (Informative)

The purpose of this informative appendix is to illustrate usage of the various profiles that were specified in Section 5 of this specification.

C.1 Identity Federation and Single Sign-On Profiles

In this section, we illustrate the three profiles for Identity Federation and single signon that were specified in Section 5.2, namely:

- Browser artifact
- Browser POST
- Liberty Enabled Client/Proxy

C.1.1 Browser Artifact profile

Figure 4, which is Figure 2 of [Liberty-BindProf], has been reproduced here for illustration of the browser artifact profile. It is seen that the <samlp:Request> and <samlp:Response> messages flowing between the Service Provider and the Identity Provider (in messages 8 and 9) use SOAP over HTTP.

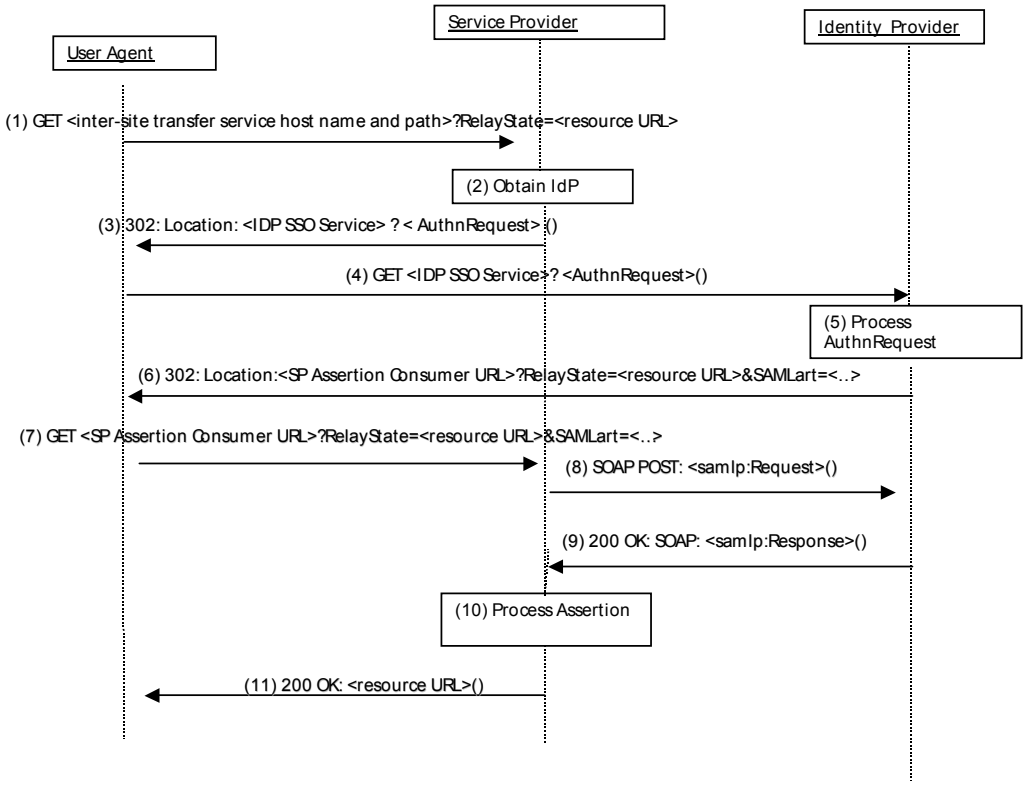

Figure 4: Browser Artifact Profile

C.1.2 Browser POST profile

Figure 5, which is Figure 3 of [Liberty-BindProf], has been reproduced here for illustration of the browser POST profile.

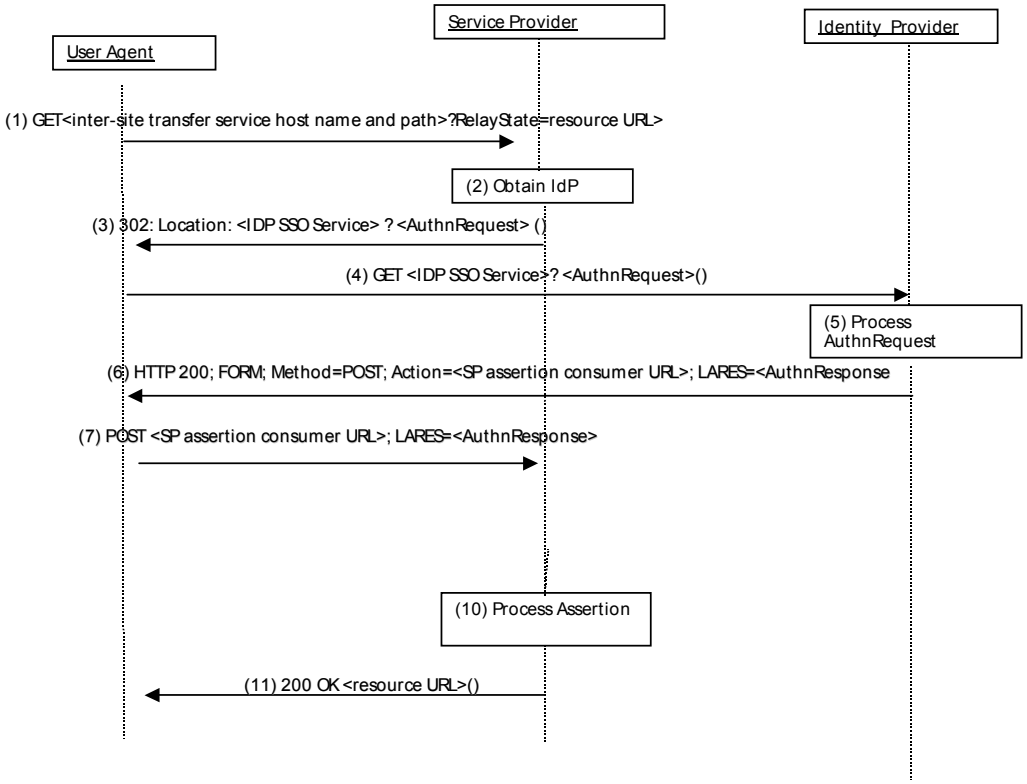

Figure 5: Browser POST Profile

C.1.3 Liberty Enabled Client/Proxy profile

A Liberty Enabled Client and Proxy device either has knowledge of, or knows how to obtain knowledge about the Identity Provider (IdP) that the user wishes to use with the Service Provider (SP). The SP needs to be aware that the client that it is communicating with is a LECP, so that the task of determining the IdP can be left to the client. Since the client uses the HTTP protocol to communicate with the SP, a suitable HTTP header is included. This header conveys to the SP that the client is a LECP, based on which the burden of determining the IdP is left to the LECP. Alternatively, while not being the preferred method, the LECP may use the User-Agent header to convey to the SP that it is a LECP. Section 3.2.5.1 of [Liberty-BindProf] provides details of the use of such headers. Message 1, in **Figure 6**, contains this header. The figure, based on Figure 5 of [Liberty-BindProf], has been reproduced here for convenience.

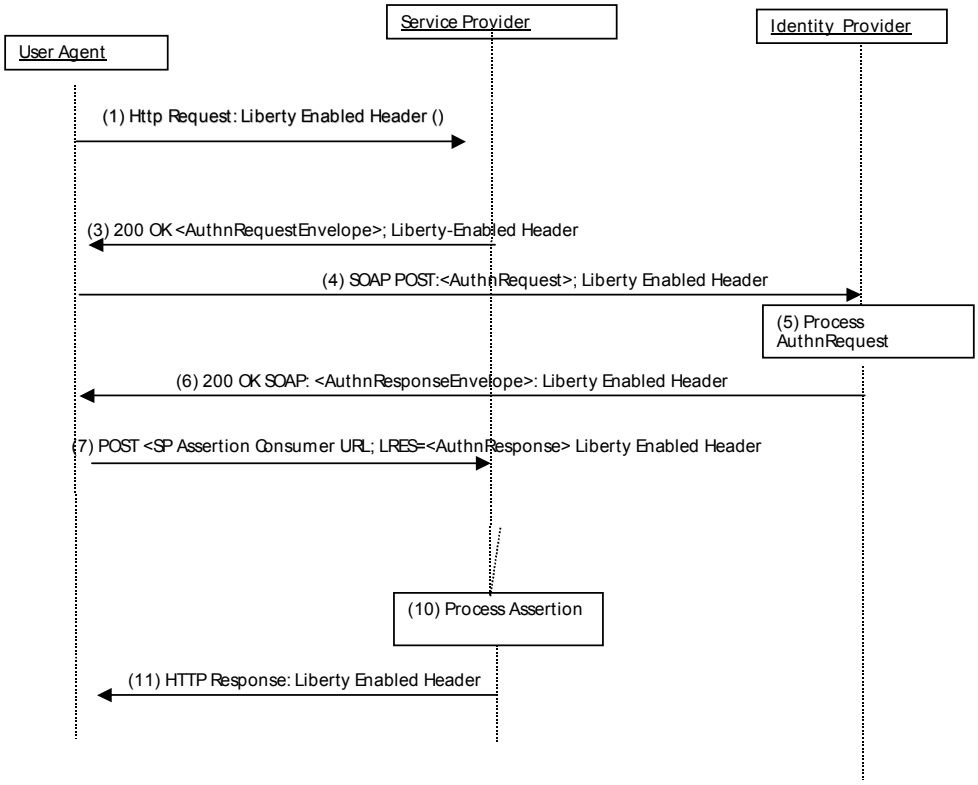

Figure 6: Liberty Enabled Client/Proxy Profile

Section 3.2.5.2 of [Liberty-BindProf] describes the message flows of **Figure 6**. It is seen that the <lib:AuthnRequest> and <lib:AuthnResponse> messages flowing between the LECP and the IdP (in messages 4 and 6) use SOAP over HTTP.

C.2 Name Registration Profiles

In this section, we illustrate the two profiles for name registration that were specified in Section 5.3, namely:

- HTTP-redirect based
- SOAP/HTTP based

Name registration may either be initiated at the Service Provider or at the Identity Provider.

C.2.1 HTTP-Redirect based Profile

Figure 7, which is Figure 6 of [Liberty-BindProf], has been reproduced here for illustration of the HTTP-redirect based profile initiated at the Identity Provider. It is seen that the <lib:RegisterNameIdentifierRequest> is initiated at the Identity Provider and sent to the Service Provider (messages 2 and 3). The Service Provider then responds with a <lib:RegisterNameIdentifierResponse> back to the Identity Provider (messages 5 and 6).

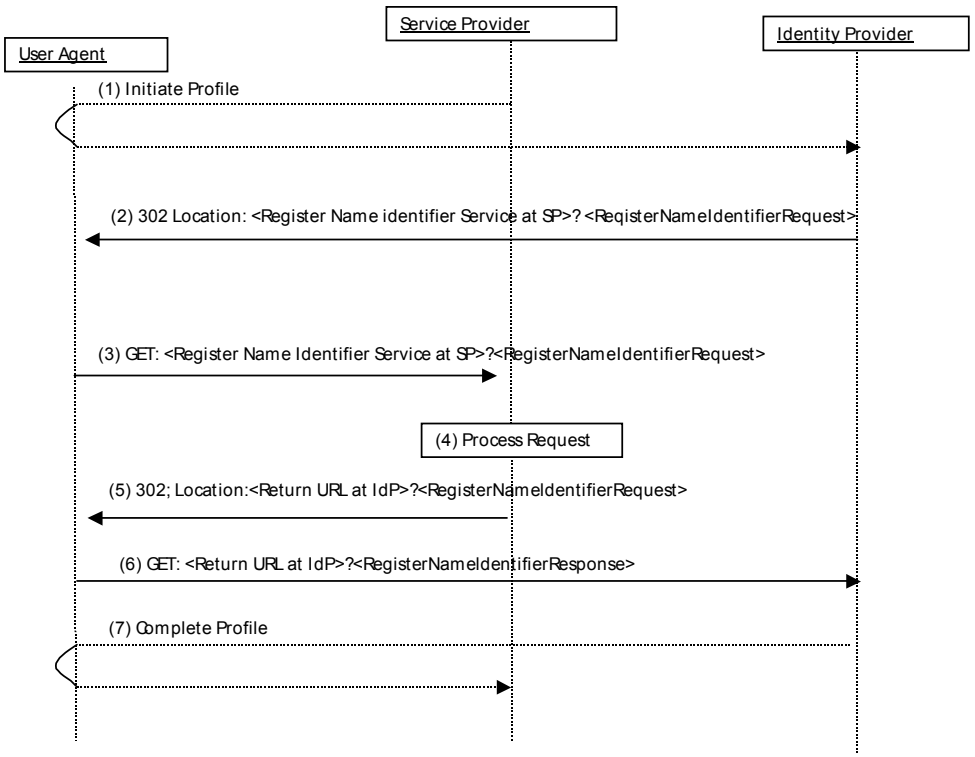

Figure 7: HTTP-Redirect Profile for Name Registration at Identity Provider

C.2.2 SOAP/HTTP based Profile

Figure 8, which is Figure 7 of [Liberty-BindProf], has been reproduced here for illustration of the SOAP-based profile for name registration initiated at the Identity Provider. It is seen that the <lib:RegisterNameIdentifierRequest> is initiated at the Identity Provider and sent to the Service Provider (message 1). The Service Provider then responds with a <lib:RegisterNameIdentifierResponse> back to the Identity Provider (message 3). Messages 1 and 3 use SOAP over HTTP.

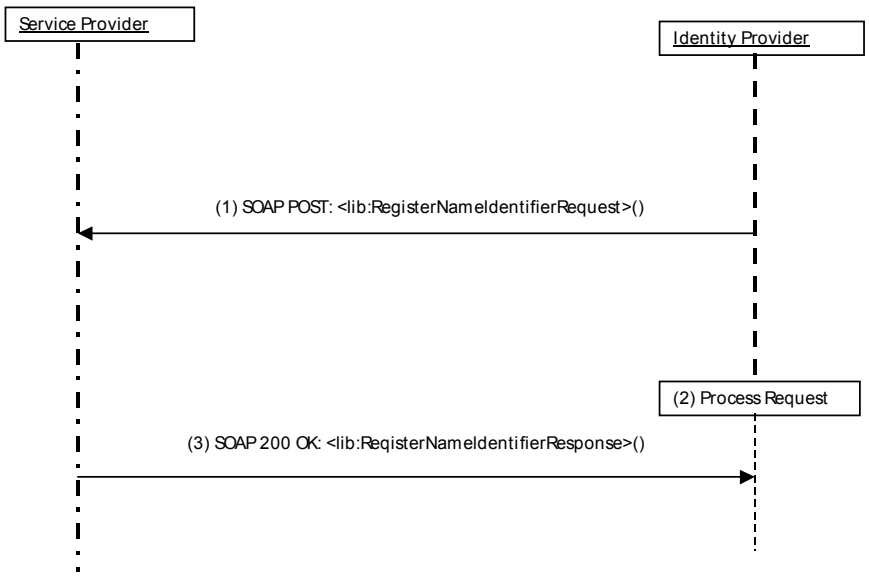

Figure 8: SOAP Profile for Name Registration at Identity Provider

C.3 Single Sign-Out Profiles

In this section, we illustrate the two profiles for single signout that were specified in Section 5.5, namely:

- HTTP-redirect based
- SOAP based

Single signout may either be initiated at the Service Provider or at the Identity Provider.

C.3.1 HTTP-Redirect based profile

Figure 9, which is Figure 10 of [Liberty-BindProf], has been reproduced here for illustration of the HTTP-Redirect based profile for single signout initiated at the Identity Provider.

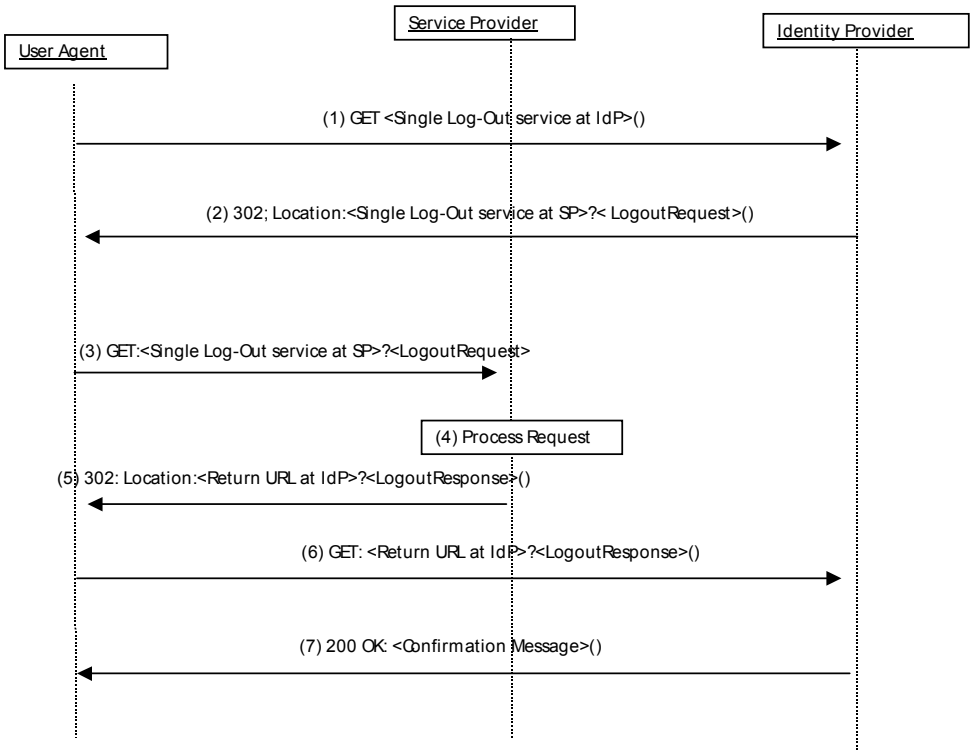

Figure 9: HTTP-Redirect Profile for Single Signout initiated at Identity Provider

Figure 10, which is Figure 13 of [Liberty-BindProf], has been reproduced here for illustration of the HTTP-Redirect based profile for single signout initiated at the Service Provider.

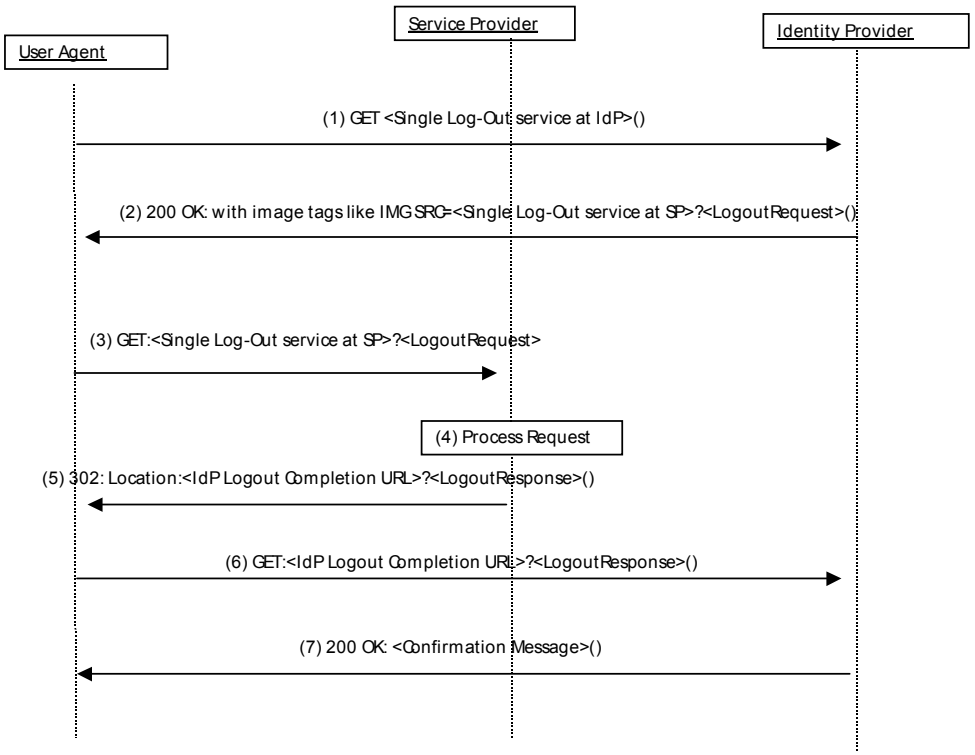

Figure 10: HTTP-Redirect Profile for Single Signout initiated at Service Provider

C.3.2 HTTP GET profile

Figure 11, which is Figure 11 of [Liberty-BindProf], has been reproduced here for illustration of the HTTP-GET based profile for single signout initiated at the Identity Provider.

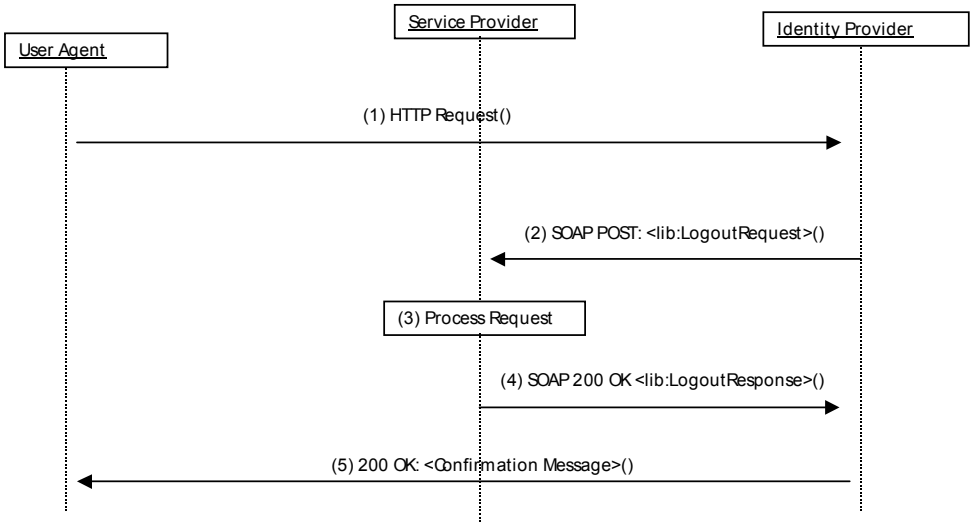

Figure 11: HTTP-GET Profile for Single Signout initiated at Identity Provider

C.3.3 SOAP/HTTP based profile

Figure 12, which is Figure 12 of [Liberty-BindProf], has been reproduced here for illustration of the SOAP/HTTP based profile for single signout initiated at the Identity Provider.

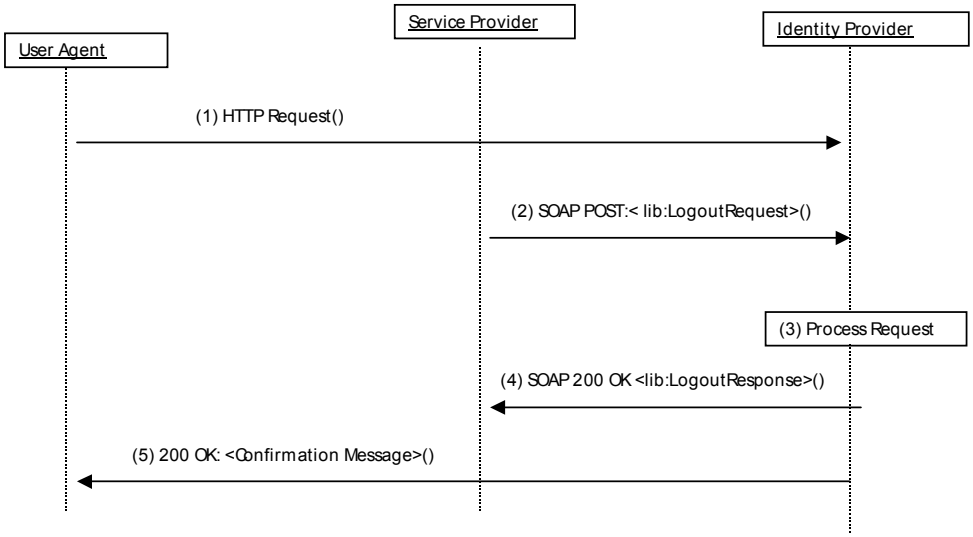

Figure 12: SOAP/HTTP based Profile for Single Signout initiated at Identity Provider

Figure 13, which is Figure 14 of [Liberty-BindProf], has been reproduced here for illustration of the SOAP/HTTP- based profile for single signout initiated at the Service Provider.

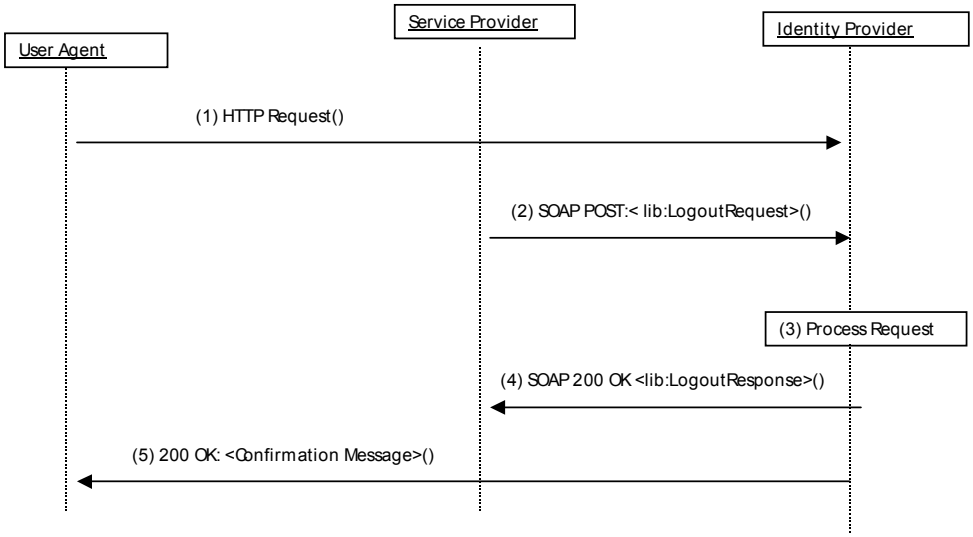

Figure 13: SOAP/HTTP based Profile for Single Signout initiated at Service Provider

C.4 Federation Termination Profiles

In this section, we illustrate the two profiles for federation termination that were specified in Section 5.6, namely:

- HTTP-redirect based
- SOAP/HTTP based

Federation termination may either be initiated at the Service Provider or at the Identity Provider.

C.4.1 HTTP-Redirect based Profile

Figure 14, which is Figure 8 of [Liberty-BindProf], has been reproduced here for illustration of the HTTP-redirect based profile for federation termination initiated at the Identity Provider.

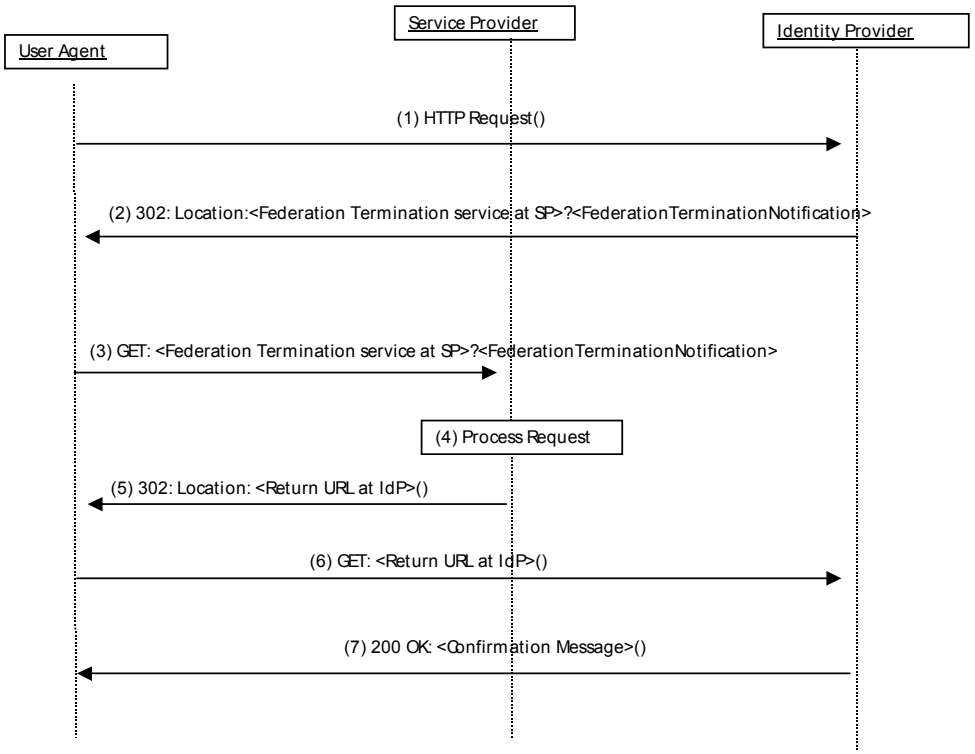

Figure 14: HTTP -Redirect based Profile for Federation Termination initiated at Identity Provider

C.4.2 SOAP/HTTP based Profile

Figure 15, which is Figure 9 of [Liberty-BindProf], has been reproduced here for illustration of the SOAP/HTTP- based profile for federation termination initiated at the Identity Provider.

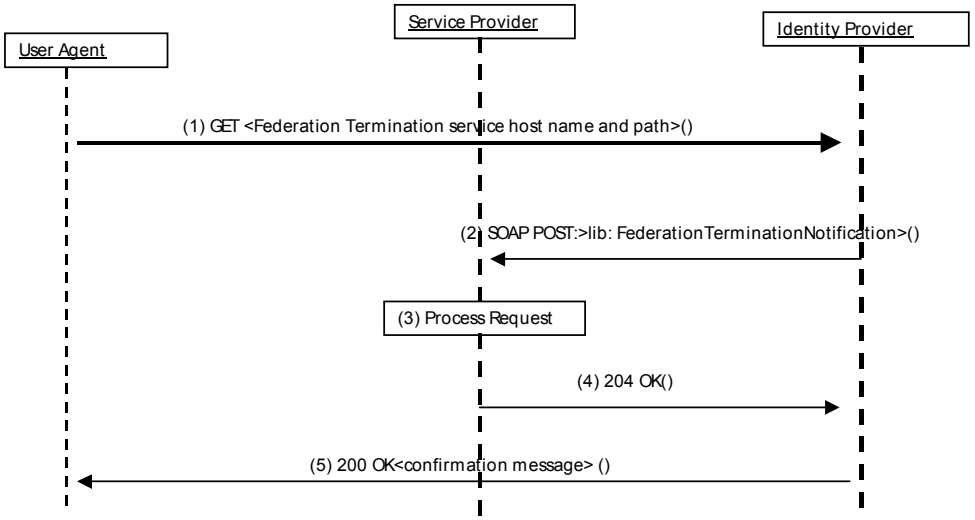

Figure 15: SOAP/HTTP based Profile for Federation Termination initiated at Identity Provider

Appendix D. Change History (Informative)

D.1 Approved Version History

Reference	Date	Description
OMA-TS-OWSER_NI_FF-V1_0-20060328-A	28 Mar 2006	Version 1.0 Approved TP ref OMA-TP-2006-0097-OWSER_NI_v1_0_for_final_approval