Liberty ID-WSF Web Services Framework Overview

Version: 1.0-errata-v1.0

Editors:
Jonathan Tourzan, Sony Corporation of America
Yuzo Koga, Nippon Telegraph and Telephone Corporation

Contributors:
John Beatty, Sun Microsystems, Inc.
Jeff Hodges, Sun Microsystems, Inc.
Gary Ellison, Sun Microsystems, Inc.
John Kemp, IEEE-ISTO
Jason Rouault, Hewlett-Packard Company
Robert Aarts, Nokia Corporation
Jukka Kainula, Nokia Corporation
Thomas Wason, IEEE-ISTO
Peter Thompson, IEEE-ISTO

Abstract:

This is a *non-normative* document intended to provide an overview of the relevant features of the Liberty ID-WSF specifications. It provides a general introduction to the Liberty ID-WSF framework, and to how it fits with the other layers of the Liberty architecture. The reader is assumed to have some familiarity with SOAP 1.1, WS-Security, SAML, XML, and basic concepts such as namespaces and URIs.

Filename: draft-liberty-idwsf-overview-1.0-errata-v1.0.pdf

Notice

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

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

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

Contents

1. Introduction

1.1. About this document

The Internet is now a prime vehicle for personal, business and community interactions. The Liberty Identity Federation Framework (ID-FF) proposed the use of federated network identity to solve the problems of network identity. The Liberty Identity Web Services Framework (ID-WSF) builds upon this foundation and provides a framework for identity-based web services in a federated network identity environment.

This document is a *non-normative* overview intended to describe principal features of the Liberty ID-WSF specifications. It provides a general introduction to the Liberty ID-WSF framework, and describes where it fits with the other layers of the Liberty architecture, as well as with other relevant technologies for authentication.

Further details of the Liberty ID-WSF may be found in the following normative technical specification documents: ID-WSF Discovery Service ([LibertyDisco]), ID-WSF SOAP Binding ([LibertySOAPBinding]), ID-WSF Security Mechanisms ([LibertySecMech]), ID-WSF Interaction Service ([LibertyInteract]), ID-WSF Client Profiles ([Liberty-ClientProfiles]), ID-WSF Static Conformance Requirements, and ID-WSF Data Services Template ([LibertyDST]). Definitions for abbreviations and acronyms not immediately defined in this document may be found in the Liberty Technical Glossary documents for Liberty ID-FF and Liberty ID-WSF ([LibertyGlossary]). As this overview is non-normative it does not use terminology "MUST", "MAY", "SHOULD" in a manner consistent with [RFC2119].

The goal of this overview is to provide sufficient information for the readers to understand the architecture defined by the ID-WSF framework and the basic usage scenarios defined for use within the framework. The overview also highlights how the ID-WSF interacts with an identity management framework (such as Liberty ID-FF).

The audience for this document is technical managers and application developers. The reader is assumed to have some familiarity with SOAP ([SOAPv1.1][SOAPv1.2]), WS-Security ([wss-sms]), SAML ([SAMLCore11]) and basic concepts such as namespaces and URIs. The ID-WSF specifications draw upon work conducted in OASIS, W3C and IETF. Standards referenced in a normative manner include SAML, WS-Security, HTTP, WSDL 1.1 ([WSDLv1.1]), XML ([XML]), SOAP 1.1, XML-Encryption ([xmlenc-core]), XML-Signature ([XMLDsig]), TLS1.0 ([RFC2246]) or SSL3.0 ([SSL]), and WAP.

1.2. What is the Liberty Alliance

The Liberty Alliance Project represents a broad spectrum of industries united to drive a new level of trust, commerce and communications on the Internet.

1.2.1. The Liberty Vision

The members of the Liberty Alliance envision a networked world across which individuals and businesses can engage in virtually any transaction without compromising the privacy and security of vital identity information.

1.2.2. The Liberty Mission

To accomplish its vision, the Liberty Alliance will establish open technical specifications that support a broad range of network identity-based interactions and provide businesses with:

- A basis for new revenue opportunities that economically leverage their relationships with consumers and business partners and
- A framework within which the businesses can provide consumers with choice, convenience, and control when using any device connected to the Internet.

1.3. What is Network Identity?

When users interact with services on the Internet, they often tailor the services in some way for their personal use. For example, a user may establish an account with a username and password and/or set some preferences for what information the user wants displayed and how the user wants it displayed. The network identity of each user is the overall global set of these attributes constituting the various accounts.

Today, users' accounts are scattered across isolated Internet sites. Thus the notion that a user could have a cohesive, tangible network identity is not realized.

1.3.1. The Liberty Objectives

The key objectives of the Liberty Alliance are to

- Enable consumers to protect the privacy and security of their network identity information
- Enable businesses to maintain and manage their customer relationships without third-party participation
- Provide an open single sign-on standard that includes decentralized authentication and authorization from multiple providers
- Create a network identity infrastructure that supports all current and emerging network access devices

These capabilities can be achieved when, first, businesses affiliate together into circles of trust based on Liberty-enabled technology and on operational agreements that define trust relationships between the businesses and, second, users federate the otherwise isolated accounts they have with these businesses (known as their local identities). In other words, a circle of trust is a federation of Service Providers and Identity Providers that have business relationships based on Liberty architecture and operational agreements. Note: Operational agreement definitions are out of the scope of the Liberty ID-FF Version 1.2 specifications. See Figure 1.

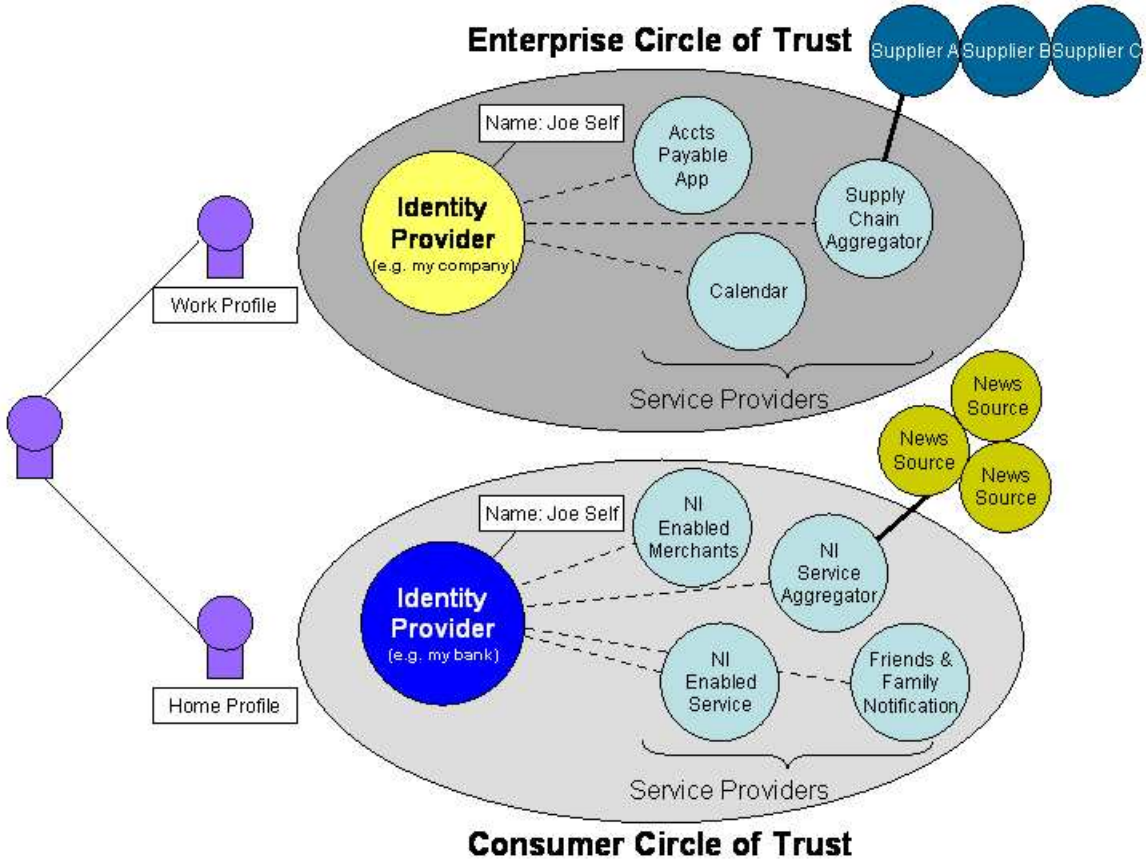


Figure 1. Federated Network Identity and Circles of Trust

From a Liberty perspective, the salient actors in Figure 1 are the user, Service Providers, and Identity Providers. Service Providers are organizations offering Web-based services to users. This broad category includes practically any organization on the Web today, for example, Internet portals, retailers, transportation providers, financial institutions, entertainment companies, not-for-profit organizations, governmental agencies, etc.

Identity Providers are Service Providers offering business incentives so that other Service Providers affiliate with them. Establishing such relationships creates the circles of trust shown in Figure 1. For example, in the enterprise circle of trust, the Identity Provider is a company leveraging employee network identities across the enterprise. Another example is the consumer circle of trust, where the user's bank has established business relationships with various other Service Providers allowing the user to wield his/her bank-based network identity with them. Note: A single organization may be both an Identity Provider and a Service Provider, either generally or for a given interaction.

Service Providers and Identity Providers enable these scenarios by deploying Liberty-enabled products in their infrastructure, but do not require users to use anything other than today's common Web browser.

1.4. What is the Identity Web Services Framework?

The Liberty Identity Web Services Framework defines a SOAP based invocation framework with a layered architecture. The framework does not specify any contents for the SOAP body, allowing the development of identity services within the context of the Liberty Identity Web Services Framework. The layering is schematically depicted in Figure 2.

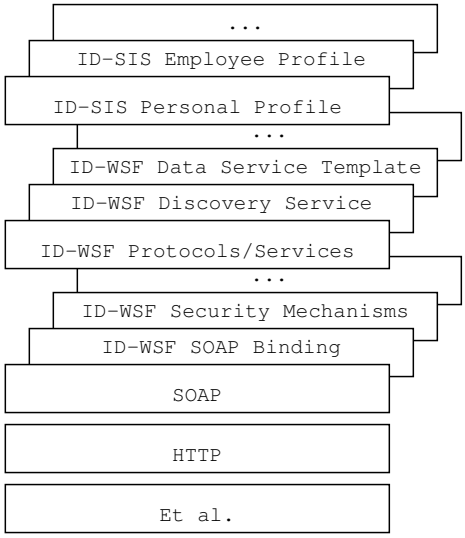


Figure 2. Liberty ID-WSF Protocol Architecture

1.5. Synopsis of Specifications

1.5.1. ID-WSF SOAP Binding (ID-WSF/Normative)

The ID-WSF SOAP Binding ([LibertySOAPBinding]) provides a SOAP-based invocation framework for identity services. It defines SOAP Header blocks and processing rules enabling the invocation of identity services via SOAP requests and responses. Additionally, a usage directive container is defined for those implementations that wish to use an existing rights expression language to specify the required service and data usage policies.

1.5.2. ID-WSF Security Mechanisms (ID-WSF/Normative)

The ID-WSF Security Mechanisms ([LibertySecMech]) describes profiles and requirements for securing the discovery and use of identity services. It includes security requirements to both protect privacy, and to ensure integrity and confidentiality of messages between Service Providers.

1.5.3. ID-WSF Discovery Service (ID-WSF/Normative)

The ID-WSF Discovery Service ([LibertyDisco]) defines a core identity service that enables various entities (e.g. Service Providers) to dynamically discover a Principal's registered identity services. Given the type of service desired (e.g. Personal Profile service ([LibertyIDPP])), the Discovery Service responds with a service description containing WSDL ([WSDLv1.1]) for the desired identity service, provided that permissions set by the Principal allow the disclosure of these resources to the relevant entity. The Discovery Service can also function as a security token service, issuing security tokens to the requester that the requester will use in the request to the discovered identity service.

1.5.4. ID-WSF Data Services Template (ID-WSF/Normative)

The ID-WSF Data Services Template ([LibertyDST]) provides the building blocks when implementing a data service (e.g. Personal Profile service ([LibertyIDPP])) on top of the Identity Web Services Framework. The specification defines how to query and modify data stored in a data service and provides some common attributes for data services.

1.5.5. ID-WSF Interaction Service (ID-WSF/Normative)

An identity service may need to obtain permission from a user (or someone who owns a resource on behalf of that user) to allow them to share data with requesting services. The ID-WSF Interaction Service ([LibertyInteract]) details protocols and profiles for interactions that allow services to carry out such actions.

1.5.6. ID-WSF Profiles for Liberty-enabled User Agents or Devices (ID-WSF/Normative)

ID-WSF Profiles for Liberty-enabled User Agents or Devices ([LibertyClientProfiles]) describes the profiles and requirements for Liberty-enabled clients interacting with the SOAP based authentication service. A user agent or device that has specific support for one or more profiles of the Liberty specifications. It should be noted that although a standard web browser can be used in many Liberty-specified scenarios, it does not provide specific support for the Liberty protocols, and thus is not a Liberty-enabled User Agent or Device (LUAD). No particular claims of specific functionality should be implied about a system entity solely based on its definition as a LUAD. Rather, a LUAD may perform one or more Liberty system entity roles as defined by the Liberty specifications it implements. For example, a LUAD-WSC is not a website that acts as a Service Provider, but a user agent or device that wants to make access to identity service, and a LUAD-DS is a user agent or device offering a Liberty ID-WSF Discovery Service ([LibertyDisco]).

1.5.7. Metadata (ID-FF/ID-WSF Independent)

The Metadata ([LibertyMetadata]) defines schema and protocols that facilitate real-time requests for metadata (previously assumed to be an out-of-band transfer). This will allow more spontaneous conversations between Liberty-compliant entities. A mechanism is defined for publishing the metadata. Several mechanisms for retrieving the metadata are defined (DNS ([RFC1034]), well known location). The metadata architecture is designed to be flexible going forward.

Functionally, there are three primary functions for this metadata:

- **entity core metadata**, which covers the metadata elements introduced in release 1 of the protocol with additional elements introduced in this release. Core metadata includes information about cryptographic keys used by entities, SOAP related information for service endpoints, as well as Identity/Service Provider specific information and other service related information.

- **entity trust metadata**, which enables entities to cast business decisions based on the characteristic trust information provided in this class. This is not defined within the Alliance, but the metadata architecture could be used to publish or retrieve this data.

- **origin and document verification** through signature use in (server authenticated) HTTPS retrieval of the instance documents, DNS signatures ([RFC1034]), and document level signatures

1.5.8. Reverse HTTP Binding (ID-FF/ID-WSF Independent)

The Reverse HTTP Binding ([LibertyPAOS]) enables a normal HTTP-based user-agent to receive SOAP requests inside an HTTP response. This allows end users to host identity services on their devices without running an HTTP server or being IP addressable from the Internet.

1.5.9. SOAP Authentication Service (ID-FF/ID-WSF Independent)

In the ID-WSF context, entities (e.g. Web Service Consumer and Web Service Provider) may need to be authenticated by exchanging SOAP messages each other. However, SOAP specifications ([SOAPv1.1][SOAPv1.2]) do not specify any particular security mechanisms. The SOAP Authentication Service ([LibertyAuthn]) defines authentication protocol between entities over SOAP, based on a profile of Simple Authentication and Security Layer framework ([RFC2222]). It also defines ID-WSF Authentication Service that an Identity Provider may offer in the ID-WSF

context, and ID-WSF Single Sign-On Service. The former enables Web Service Consumer and/or LUAD to be authenticated by Identity Providers and obtain ID-WSF security tokens. The latter enables Web Service Consumer to obtain assertions within the ID-WSF context, that can be used in the ID-FF context.

2. ID-WSF User Experience Example

This section provides a simple, plausible examples of the Liberty ID-WSF user experience, from the perspective of the user, to set the overall context for additional technical details. As such, actual technical details are hidden or simplified.

Note: The user experience examples presented in this section are non-normative and are presented for illustrative purposes only.

2.1. Usage Examples with three websites

In this section, a simple ID-WSF user experience example is described, in which Joe Self is ordering beer and pizza on the Internet. More details of this example from the implementer point of view are described in the Liberty ID-WSF Implementation Guide ([LibertyIDWSFGuide]).

2.1.1. Assumptions

These user experience examples are based upon the following set of actors:

- Joe Self: A user of Web-based online services.
- WhiteBroadBand: Internet service provider that acts as his Identity Provider. It also hosts a Discovery Service ([LibertyDisco]).
- BlueLiquor: A liquor shop website.
- YellowPizza: A pizzeria website.

This user experience example assumes two things:

- Identity federation has occurred for Joe Self's accounts at WhiteBroadBand and BlueLiquor. Joe Self registers his personal information at BlueLiquor website for delivering ordered liquors to customer's residence. BlueLiquor is also able to provide other websites with a customer's personal information if the customer has provided permission.
- Identity federation has occurred for Joe Self's accounts at WhiteBroadBand and YellowPizza. YellowPizza can discover customer's identity services by interacting with WhiteBroadBand so that it gets customer's shipping address information and delivers pizza there.

2.1.2. User Experience Scenario

One day on Sunday, Joe decides to order beers at BlueLiquor website that is his favorite liquor shop. When he tried to make access to BlueLiquor website, he is redirected to WhiteBroadBand website since he has not been authenticated. WhiteBroadBand is his Identity Provider, and he submits his credential to WhiteBroadBand. Once he is authenticated by WhiteBroadBand, he can make access to the BlueLiquor website.

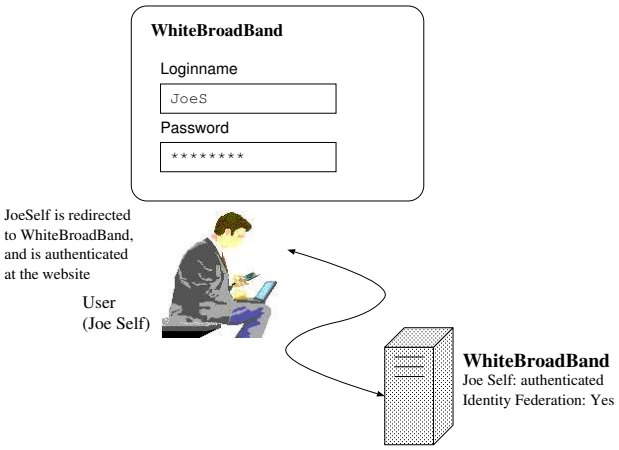


Figure 3. Joe Self is redirected to WhiteBroadBand website that is his Identity Provider

He orders two dozen beers at the BlueLiquor website, and ask to deliver them to his pre-registered shipping address. He also requests BlueLiquor to register that his shipping address information is available at this site, to the Discovery Service ([LibertyDisco]) hosted by WhiteBroadBand, so that his shipping address information attribute at the BlueLiquor website can be shared with other websites. BlueLiquor registers it to Discovery Service, and sets Joe's attribute sharing policy as it can be shared with other websites.

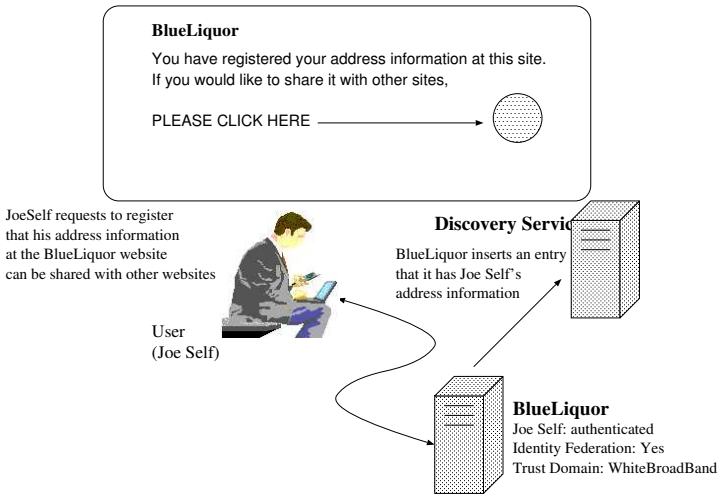


Figure 4. Joe Self requests BlueLiquor website to register that his address information can be shared

Subsequently, he tries to make access to the YellowPizza website. Since he has already been authenticated by WhiteBroadBand, he does not need to be authenticated again. He orders a mayonnaise pizza, and is asked where they should deliver it. He requests YellowPizza to get his shipping address information from other website, and they get it from the BlueLiquor website. Finally, two dozen beers and the mayonnaise pizza are delivered to his residence.

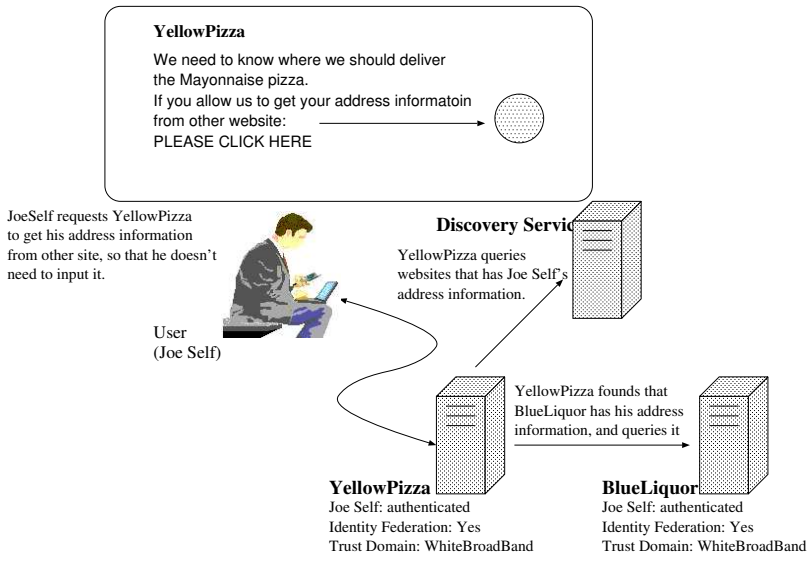


Figure 5. Joe Self requests YellowPizza website to get his address information from other website

2.2. Usage Examples with Mobile IdP

2.2.1. Assumptions

These user experience examples are based upon the following set of actors:

- Joe Self: A user of Web-based online services.
- Company XYZ: Joe self's employer. Joe Self is a Vice President for XYZ in charge of buying widgets. When Joe is in the office, Company XYZ acts as his Identity Provider.
- Company ABC: A Vendor of widgets that works closely with Company XYZ.
- Mobile IdP AntarctiCom: A Mobile Operator who acts as Identity Provider for Joe Self when not in the office.
- Airline, Inc.: One of the airline company that is able to get customer's personal information from other websites.

The Liberty ID-WSF user experience assumes three things:

- Identity federation has occurred for Joe Self's accounts at Company XYZ and Company ABC. At Company ABC there are access policies that recognize Joe Self as an Employee of Company XYZ who is authorized to purchase widgets.
- Identity federation has occurred for Joe Self's accounts between Company XYZ and AntarctiCom. Business agreements have been signed between Company XYZ and AntarctiCom such that AntarctiCom may authenticate Company XYZ's users, and that Company XYZ may chain these assertions when interacting with their own partners.
- Identity federation has occurred for Joe Self's accounts between Airline, Inc. and AntarctiCom. Business agreements have been signed between Airline, Inc. and AntarctiCom such that AntarctiCom may authenticate Airline's customers, and that Airline, Inc. can discover customer's identity services by interacting with AntarctiCom.

2.2.2. User Experience Scenario

Joe Self is on the road at a big conference. He is presenting on widgets and their importance to Company XYZ's businesses. After his big presentation, he decides to access his corporate web portal with his browser in order to check his e-mail. He turns on his Mobile Data device, say a GSM phone with GPRS capability, and the Mobile IdP, AntarctiCom, authenticates his device.

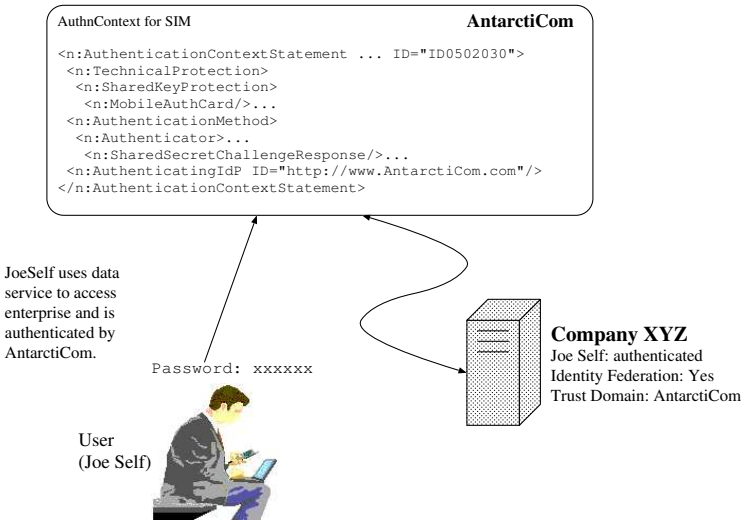


Figure 6. Joe Self Authenticated by AntarctiCom, Navigates to XYZ Portal

Joe Self finds out that XYZ has won a big order. They will need to buy widgets to make their products. Joe Self navigates to Company ABC's portal to check widget prices. Company ABC is a prime supplier to Company XYZ, so if the prices are fair Joe Self will buy from them. CompanyABC and CompanyXYZ have set up contracts and installed infrastructure in order to allow federation of accounts between their trust domains. Unfortunately Company ABC does not recognize AntarctiCom as an Identity Provider. XYZ and AntarctiCom have business agreements such that they can chain authentication though.

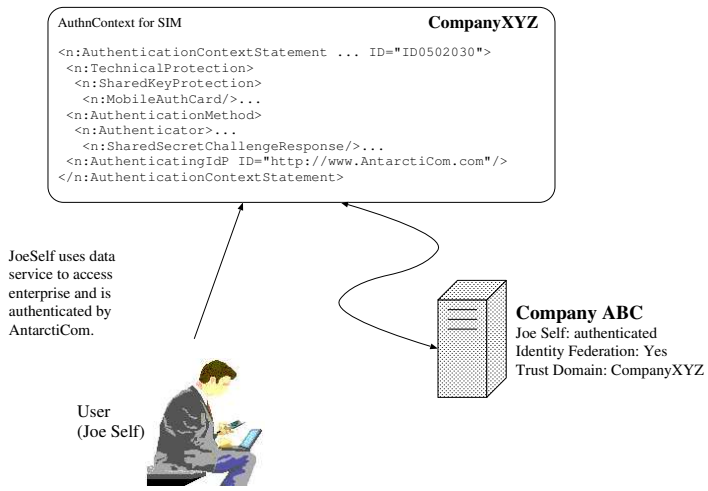


Figure 7. Joe Self Navigates to Company ABC, uses XYZ as Identity Provider

Joe checks the prices of widgets. They look good. He would like to buy. ABC has access control policies that require the use of a one time password in addition to the Identity Providers SIM based Authentication for that level of transaction. Joe provides the password and the order is processed. Joe decides that he better just change his flight home so that he can be in the office to discuss the order with his staff. Unfortunately the flight is full. Joe navigates

to another airline but notices that his personal information is not up to date. The airline was able to discover Joe's Personal Profile ([LibertyIDPP]) during his sign-on at the site. He clicks on a button on the web page to update his profile at the airline.

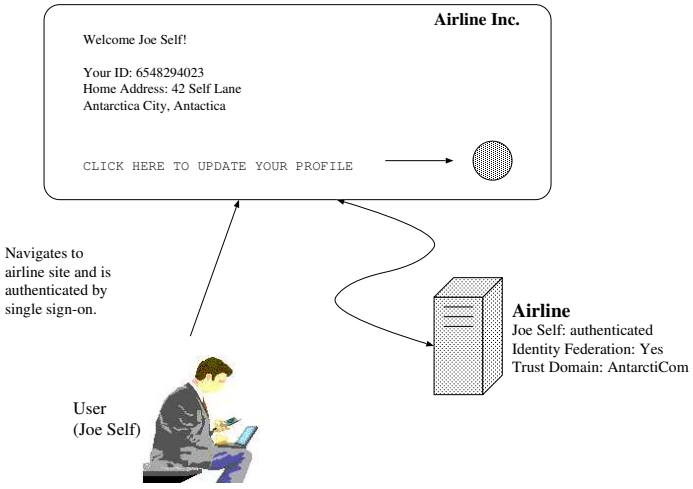


Figure 8. Joe Self Navigates to Airline site, uses AntarctiCom as Identity Provider

Joe Self has set his permissions at AntarctiCom such that he wants to be asked for permission prior to Personal Profile ([LibertyIDPP]) attributes being released to Service Providers. AntarctiCom uses the Liberty ID-WSF Interaction Service ([LibertyInteract]) to query Joe Self for permission to release certain Personal Profile attributes.

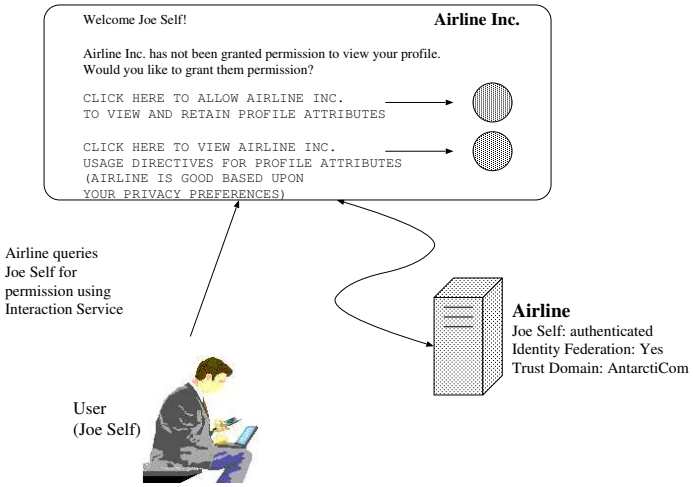


Figure 9. Airline uses Interaction Service to get permission to invoke Joe Self's Personal Profile

Joe Self is leaving Antarctica next week, and he is not sure that AntarctiCom will have data services in the visited network[1] . He decides to set up his own Personal Profile service on the mobile device that he is using. Upon arriving in the North Pole, he sets permissions on his Personal Profile service such that his Postal Code and Nationality will be available to visited Service Providers. Joe Self then receives personalized service when visiting websites. In addition, should Service Providers require additional information, they can directly query Joe Self. The ability to query is provided by the Liberty ID-WSF Interaction Service ([LibertyInteract]) defined as part of the Liberty specifications.

[1] A visited network is the network other than the home network of a mobile device, to which the mobile device is currently connected. It is usually referred as such from the mobile operator point of view.

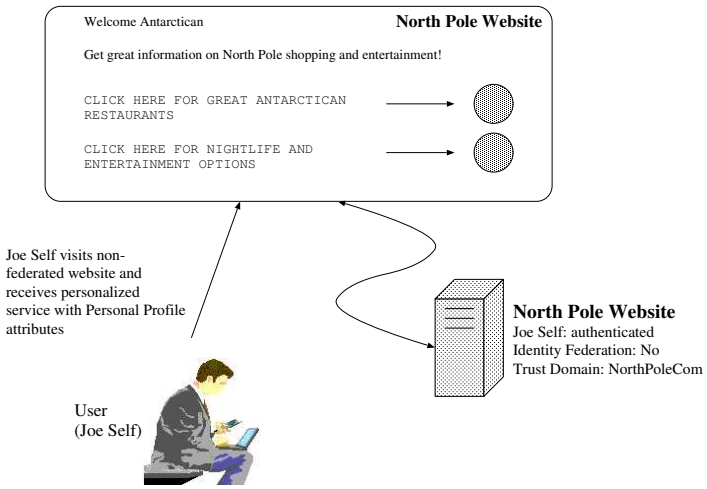


Figure 10. Joe Self visits North Pole website, privacy neutral Personal Profile attributes provided based upon set preferences for new Service Providers

The mobile device examples shows a scenario with the optimizations from the use of Reverse HTTP Binding ([LibertyPAOS]), the use of LUAD for discovery of Web Services on the mobile device ([LibertyClientProfiles]), as well as use of the SOAP Authentication Service for authentication of the LECP ([LibertyAuthn]).

3. Liberty Engineering Requirements Summary

This section summarizes the Liberty general and functional engineering requirements.

3.1. General Requirements

The Liberty-enabled systems should follow the set of general principals outlined in Section 3.1.1 and Section 3.1.2. These principles cut across categories of functionality.

3.1.1. Client Device/User Agent Interoperability

Liberty clients encompass a broad range of presently deployed Web browsers, other presently deployed Web-enabled client access devices, and newly designed Web-enabled browsers or clients with specific Liberty-enabled features.

The Liberty architecture and protocol specifications must support a basic level of functionality across the range of Liberty clients.

3.1.2. Openness Requirements

Liberty architecture and protocol specifications must provide the widest possible support for

- Operating systems
- Programming languages
- Network infrastructures

and must not impede multivendor interoperability between Liberty clients and services, including interoperability across circle of trust boundaries.

3.2. Functional Requirements

Liberty architecture and protocols must be specified so that Liberty-enabled implementations are capable of performing the following activities:

- Service discovery in identity federation environment
- Registration of services
- Support for gathering consent from a Principal
- Support for anonymous services
- Support for usage directives

3.2.1. Service Discovery

Requirements of service discovery stipulate that

- Architecture provides a mechanism for Service Providers to query the Discovery Service for the relevant providers of services or attribute classes within a service for a particular Principal.

- Support for user prompt by the Discovery Server to prompt during the registration process (e.g. to confirm the registration). Such mechanism(s) should support the ability to allow the requestor to prompt the user, asking the requestor to direct the user to the Discovery Server's site, or the Discovery Server using a LECP communications channel to ask the user directly.

3.2.2. Registration of Services

Requirements of service registration stipulate that

- Architecture provides a mechanism for Service Providers to register/deregister with the Discovery Service a list of services or attribute classes within a service that it provides for a specific Principal.

3.2.3. Support for Gathering Consent

Requirements of consent gathering stipulate that

- Mechanism for a relying Service Provider to request that the invoking Service Provider direct a Principal to the relying Service Provider to request the Principal for consent.
- Mechanism for a Service Provider to utilize a LECP communications channel for querying the Principal's consent and obtaining the Principal's response.
- Mechanism for Providers to associate Principal's consent for his/her permissions for a Service Provider for a given set of attributes, when the set of attributes are shared with the Service Provider.
- Mechanism for a relying Service Provider to partially fulfill requests for attributes if consent not given for all attributes.

3.2.4. Support for Anonymous Service

Requirements of anonymous service stipulate that

- Mechanism for a Service Provider to make anonymous attribute requests and receive anonymous attribute responses. (Ability to share attributes without disclosing the identity of the Principal to the requestor or Service Provider).
- Mechanism to prevent correlation of pseudonyms in service tokens with Principal Identifiers.

3.2.5. Support for Usage Directives

Requirements of usage directives stipulate that

- Mechanism for a Service Provider to associate intended usage with the requested attributes in an attribute request to a relying Service Provider.
- Mechanism for a Service Provider to associate the agreed upon intended usage directives with the attribute response

- Mechanism for a Service Provider to return a list of acceptable usage directives to a Service Provider, when the intended usage doesn't match the Principal's usage directives.

- Guideline for Service Providers (in the usage negotiation scenario) to always reply to an invoking Service Provider's attribute request with usage directives that are equal to or privacy-stricter than those originally stated in the Service Provider's attribute request.

4. Liberty Security Architecture

Table 1 generally summarizes the security mechanisms incorporated in the Liberty specifications, and thus in Liberty-enabled implementations, across two axes: channel security and message security. It also generally summarizes the security-oriented processing requirements placed on Liberty implementations.

Note: This section is non-normative, please refer to normative documents for detailed normative statements regarding security mechanisms ([LibertySecMech]).

Table 1. Liberty Security Mechanisms

Security Mechanism	Channel Security	Message Security (for Requests, Assertions)
Confidentiality	Required	Optional
Per-message date integrity		Required
Transaction integrity		Required
Data origin authentication		Required
Nonrepudiation		Required

Channel security addresses how communication between Identity Providers, Service Providers, and user agents is protected. Liberty implementations must use TLS1.0 ([RFC2246]) or SSL3.0 ([SSL]) for channel security, although other communication security protocols may also be employed, for example, IPsec, if their security characteristics are equivalent to TLS1.0 or SSL3.0. Note: TLS1.0, SSL3.0, and equivalent protocols provide confidentiality and integrity protection to communications between parties as well as authentication.

Critical points of channel security include the following:

- In terms of authentication, Service Providers are required to authenticate Identity Providers using Identity Provider server-side certificates. Identity Providers have the option to require authentication of Service Providers using Service Provider client-side certificates.

- Additionally, each Service Provider is required to configure a list of authorized Identity Providers, and each Identity Provider is required to be configured with a list of authorized Service Providers. Thus any Service Provider-Identity Provider pair must be mutually authorized before they will engage in Liberty interactions. Such authorization is in addition to authentication. (Note: The format of this configuration is a local matter and could, for example, be represented as lists of names or as sets of X.509 certificates ([X.509]) of other circle of trust members).

- The authenticated identity of an Identity Provider must be presented to a user before the user presents personal authentication data to that Identity Provider.

Message security addresses security mechanisms applied to the discrete Liberty protocol messages passed between Identity Providers, Service Providers, and user agents. These messages are exchanged across the communication channels whose security characteristics were just discussed.

Critical points of message security include the following:

- Liberty protocol messages and some of their components are generally required to be digitally signed and verified. Signing and verifying messages provide data integrity, data origin authentication, and a basis for non-repudiation.

- Therefore, Identity Providers and Service Providers are required to use key pairs that are distinct from the key pairs applied for TLS1.0 ([RFC2246]) or SSL3.0 ([SSL]) channel protection and that are suitable for long-term signatures.

- In transactions between Service Providers and Identity Providers, requests are required to be protected against replay, and received responses are required to be checked for correct correspondence with issued requests. Time-based assurance of freshness may be employed. These techniques provide transaction integrity.

- To become circle of trust members, providers are required to establish bilateral agreements on selecting certificate authorities, obtaining X.509 credentials ([X.509]), establishing and managing trusted public keys, and managing life cycles of corresponding credentials.

Note: Many of the security mechanisms mentioned above, for example, TLS1.0 or SSL3.0, have dependencies upon, or interact with, other network services and/or facilities such as the DNS ([RFC1034]), time services, firewalls, etc. These latter services and/or facilities have their own security considerations upon which Liberty-enabled systems are thus dependent.

5. Liberty Architecture

5.1. Concepts and Architecture

The Liberty ID-WSF defines a framework for creating, discovering, and consuming *identity services*. The Liberty ID-WSF also defines a conceptual model that provides relevant terminology for these *identity services*. Some basic identity services, such as the Discovery Service ([LibertyDisco]), are defined in a normative manner as part of the ID-WSF specifications. The following UML model describes the conceptual model presented in the Liberty specifications:

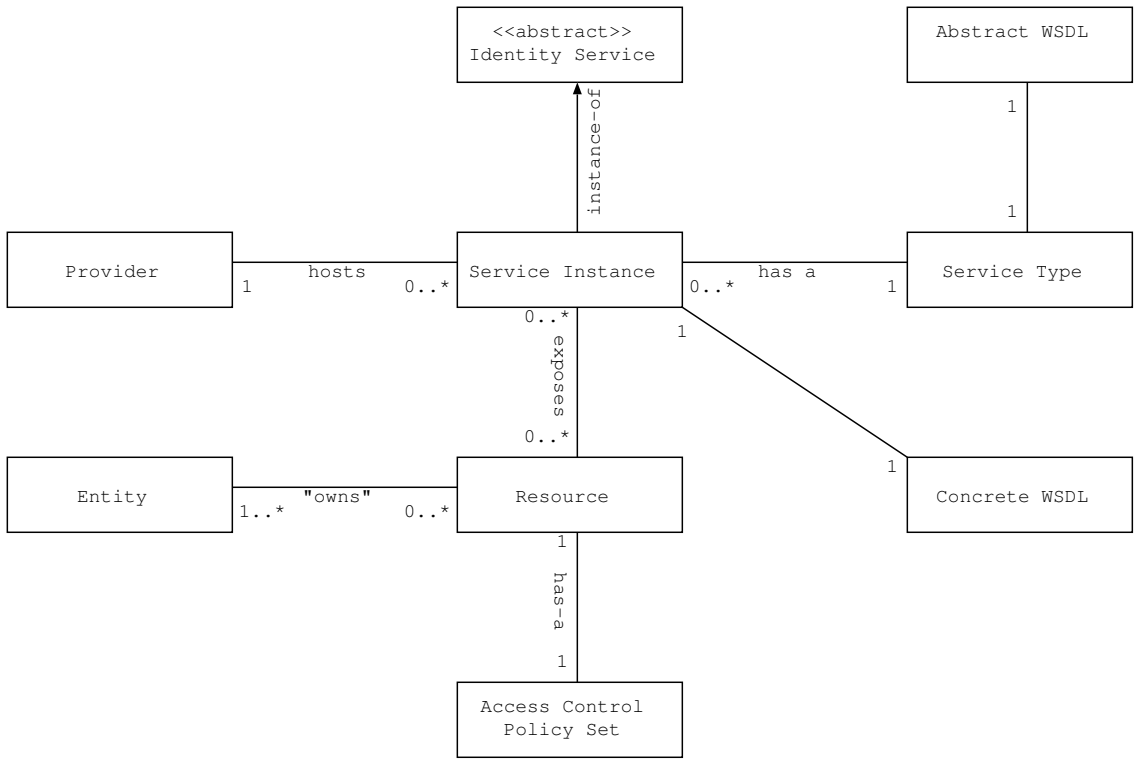


Figure 11. UML Representation of Liberty Conceptual Model

An *identity service* is an abstract notion of a web service that acts upon some resource to either retrieve information about an identity or identities, update information about an identity or identities, or perform some action for the benefit of some identity or identities.

There are different types of identity services, each of which is identified by a *service type identifier*. This service type identifier maps to exactly one *abstract WSDL* definition of a service. The definition contains only the type, message, and portType elements of a WSDL1.1 description ([WSDLv1.1]). An example of a service type is a "calendar service," which could have a service type identifier of a URI such as "urn:example:services:calendar".

A *service instance* is the instantiation of a particular type of identity service. A service instance maps to a *concrete WSDL* document (which includes the binding and service WSDL elements) that contains the *protocol endpoint* and additional information necessary for a client to communicate with the particular service instance (e.g. security policy information).

Each service instance is hosted by some *provider* that is identified by a *provider identifier*. An example of a service instance is a SOAP endpoint ([SOAPv1.1][SOAPv1.2]) offering a calendar service.

A service instance exposes a protocol interface to a set of resources. A *resource* in this specification is either data related to some identity or identities, or a service acting on behalf of some identity or group of identities. An example of a resource is a calendar containing appointments for a particular identity.

A resource commonly has *access control policies* associated with it. These access control policies are typically under the purview of the entity or entities associated with the resource (the entity or entities could be considered to "own" the resource). The access control policies on a resource must be enforced by the service instance.

5.2. Liberty Modules

The Liberty architecture consists of a multi-level layered specification set, based on open standards including SAML ([SAMLCore11]) and SOAP ([SOAPv1.1][SOAPv1.2]). There are three major components of the Liberty architecture:

- The Liberty Identity Federation Framework (ID-FF) specifies core protocols, schemata and concrete profiles that allow implementers to create a standardized, multi-vendor, identity federation network.
- The Liberty Identity Web Services Framework (ID-WSF) consists of a set of schemata, protocols and profiles for providing a basic framework of identity services, such as identity service discovery and invocation.
- Liberty Identity Service Interface Specifications (ID-SIS) utilize the ID-WSF and ID-FF to provide networked identity services, such as contacts, presence detection or wallet services that depend on networked identity.

Figure 12 below illustrates the Liberty Modules and their corresponding functional areas.

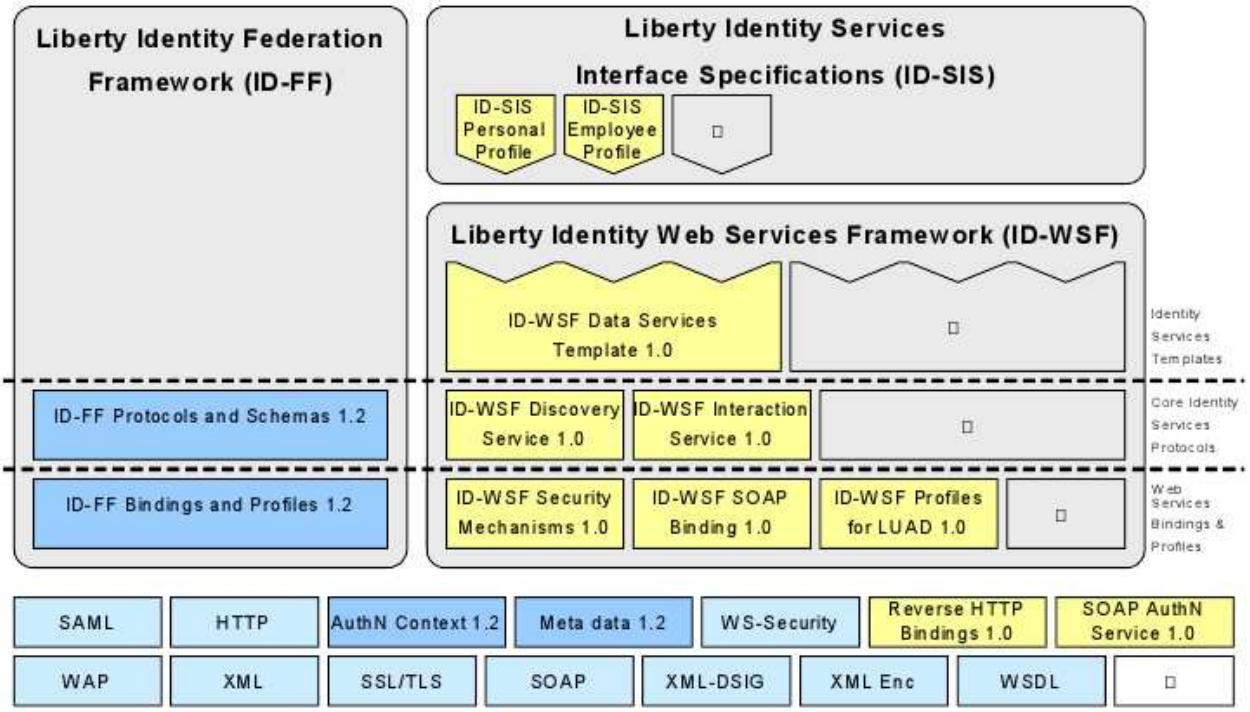


Figure 12. Liberty Modules

5.3. Summary of Functionalities

The Liberty Identity Web Services Framework defines a SOAP based invocation framework that allows identity services to be discovered and invoked. Once a service has been discovered and sufficient authorization data has been received from a trusted authority, the invoking entity (Web Service Consumer) may invoke the service at the hosting/relying entity (Web Service Provider). In order to convey the privilege of a system entity to access a resource, the framework defines extensions such that service invocation authorization data may be generated by a trusted authority and issued to the invoking system entity. The relying party or Web Service Provider can make access control decisions based upon this authorization data based upon its business practices and the preferences of the resource owner. In most cases this trusted authority is assumed to be some Identity Provider/Discovery Service ([LibertyDisco]).

The following diagram illustrates the entities involved in possible service invocation use cases.

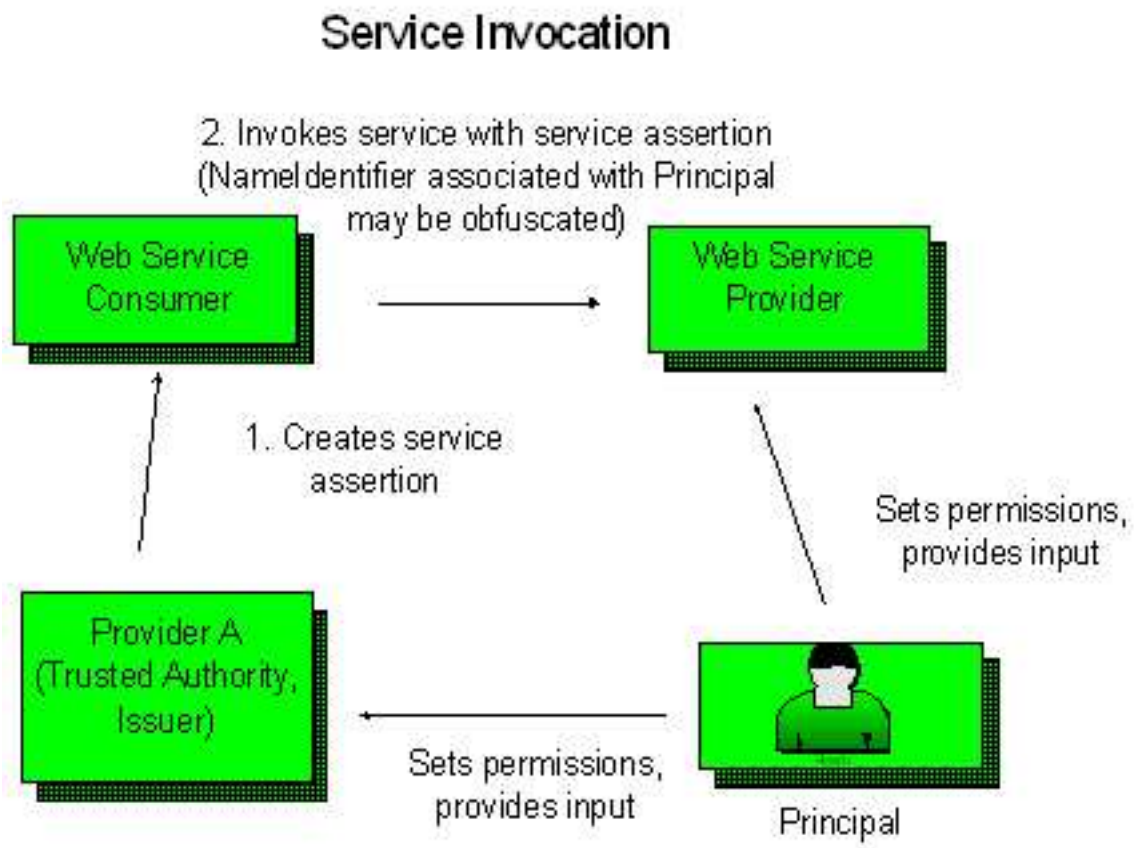


Figure 13. Service Invocation Context

5.3.1. Security Mechanisms

As in other web services contexts, access control policies must be enforced in an identity services context. The authorization decision to invoke an identity service instance offering a specific resource may be made locally (that is at the entity hosting the resource) or remotely. Regardless of whether the policy decision is distributed or not, in a permissions based context or any context with security considerations, a policy enforcement must always be implemented by the entity hosting the resource.

Identity services may rely upon a trusted third party (TTP) to make policy decisions on their behalf. In such cases, the TTP issues targeted SAML assertions ([SAMLCore11]) to those entities. These assertions have associated conditions, such as an issue instant, validity periods for each assertion. The SAML assertion also has audience restriction(s) that provide information about the intended target of the policy decision and the relying party (Web Service Provider) for the particular assertion. The SAML assertion also contains an Authorization Decision Statement which conveys the decision and information about the rights that have been granted to the resource. The Authorization Decision Statement also conveys information about the Subject and the Subject Confirmation Method by which the requesting entity will authenticate itself to the relying party.

5.3.2. Usage Directives

The Liberty ID-WSF defines extensions that allow both the invoking entity and the consuming entity to add one or more Usage Directive SOAP headers to a message ([LibertySOAPBinding]). A Usage Directive header in a request

from the invoking entity can be understood as "intended usage". It should be noted that should permissions be such that a Usage Directives level in the request cannot be met, the hosting entity must either redirect the invoking entity to the user to query for permission, or deny the service.

5.3.3. Interaction Service

The Liberty ID-WSF defines a Interaction Service protocol ([LibertyInteract]). This protocol provides schemas and profiles to enable an entity to interact with the owner of a resource that is exposed by that Web Service Provider. The ID-WSF defines three methods for a Web Service Provider to interact with a user:

1. The Web Service Provider may send a SOAP response with a RedirectRequest that instructs the Web Service Consumer to direct the user-agent to contact the Web Service Provider at a given URL.
2. The Web Service Provider may send a UserInteractionRequest to the endpoint defined in the ISService element.
3. The Web Service Provider may try to discover the Interaction Service of the resource owner to enable the Web Service Provider to send a userInteractionRequest to that service.

This interaction may be for the purposes of obtaining consent for a particular resource exposure (such as granting access to Personal Profile ([LibertyIDPP])), obtaining data from the user-agent, or some other purpose. The Interaction Service protocol is an optional part of the Liberty ID-WSF. An example of use of the Interaction Service would be to query the user for permissions in a web services context.

5.3.4. Proxy Authorization Model

The Liberty ID-WSF supports a restricted form of proxy authorization capability whereby a consumer of an identity service (the intermediate system entity or proxy) can act on behalf of another system entity (the subject) to access an identity service (the recipient) ([LibertySecMech]). To be granted the right to proxy for a subject, the intermediate system entity may need to interact with a trusted authority. Based on the authority's access control policies, the authority may generate and distribute an assertion authorizing the intermediary to act on behalf of the subject to the recipient. As an example, such the authorization decision statement might allow a proxying entity to update a calendar resource for a particular identity after a flight booking has occurred.

5.3.5. Affiliations

An affiliation allows a group of Service Providers organized to act as a single entity from the point of view of the customer (usually due to the group acting as a portal or acting as a single company such as TimeWarner and its affiliates). The Liberty Authorization Decision Statement defined allows the use of the Affiliation ID when a trusted authority is granting rights to a member of an affiliation group ([LibertyProtSchema]). An example of the use of affiliations in an application context is an Authorization Decision Statement allowing Travel Affiliation X to update a calendar after a flight booking has occurred.

5.3.6. Chaining of Services/Broker

The ID-WSF architecture provides mechanisms to allow a broker type functionality, whereby a Web Service Consumer may make a request to a Web Service Provider which acts as a broker and makes subsequent requests (as a Web Service Consumer) to other Web Service Provider(s) that have the required information. The broker subsequently aggregates the data and responds to the originating Web Service Consumer in the chain. A simple example is profile data that is stored in various places and the broker needs to query the relevant parties for the data prior to responding to a Personal Profile request.

5.3.7. Anonymous Service Requests

The trusted third party may obscure the subject's name identifier for purposes of confidentiality at the Web Service Consumer and any subsequent intermediaries. For this purpose, the ID-WSF specifies a mechanism for creating (at issuer) and consuming (at relying party) encrypted name identifiers.

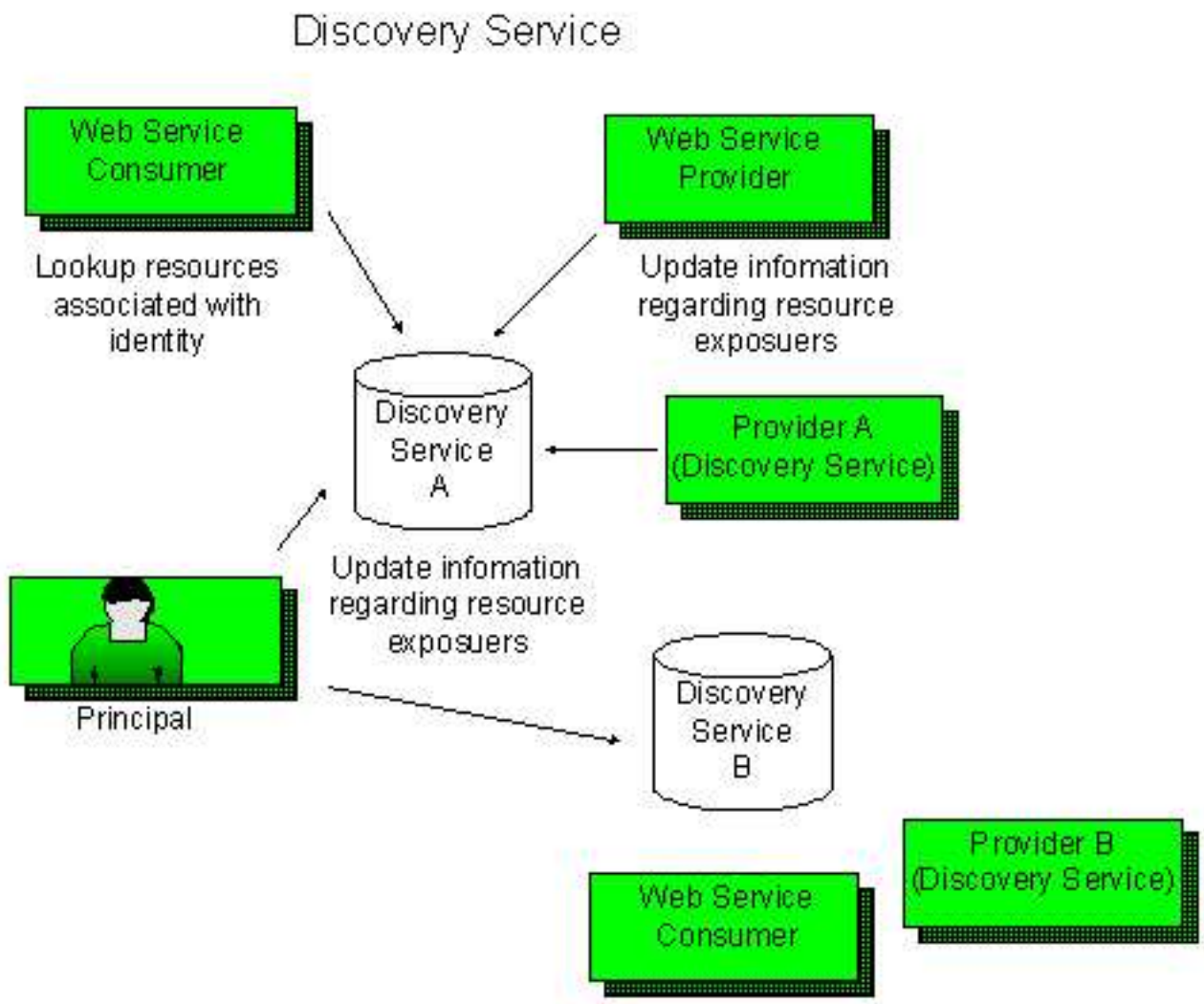
5.3.8. Discovery Service

The Discovery Service is a type of identity service that provides for the discovery of resource exposures associated with a given identity ([LibertyDisco]). An identity will typically have one or more discovery services on the network that allow other entities to discover its identity services.

The Discovery Service offers two operations, Lookup and Update. In a web services context (browsing, etc.), a Web Services Consumer may need access to a resource exposure associated with an identity (a profile or location service). The Web Service Consumer may lookup a service instance with a request that includes a service type element and extensible processing directives. The response message contains the relevant resources associated with the query, according to the access policies set by the principal/provider. The response may include tokens for service invocation.

The Update operation allows a requester to enter and remove service instances. The request allows the provider to input information about a resource exposure, and the corresponding response provides the status of the request. A Web Service Provider that hosts the resource, the host of the Directory Service, or the Principal/Resource Owner could update the resource exposure. The service registry defined by the Liberty ID-WSF has one service entry for each service type, consequently complex queries are not possible. This does not preclude having some ability to change the Lookup results based upon the access control policies of the host, and/or preferences/permissions of the resource owner. The following diagram illustrates the entities involved in possible Discovery Service use cases.

520



521

522 **Figure 14. Liberty Discovery Service**

523 5.4. Use Cases in scope for ID-WSF

524 The Liberty Alliance defines a Personal Profile service ([LibertyIDPP]) for use with the Liberty ID-WSF. The Personal
525 Profile service is designed to facilitate account creation in a web services context. The Personal Profile service allows a
526 Web Service Consumer to gather the information necessary to create an account or provide personalized services. The
527 Personal Profile specification provides a schema and API for queries of personal information. The ID-WSF provides
528 Personal Profile deployments and other ID-SIS deployments with the abilities to specify and negotiate usage directives
529 for attribute sharing, to query users for permissions using the Liberty ID-WSF Interaction Service ([LibertyInteract]),
530 as well as the ability to provide anonymous attribute requests for non-identifying Personal Profile attributes (such as
531 zip code).

532 5.5. Use Cases out of scope for ID-WSF, but relevant to later work

533 The Liberty Alliance anticipates that other services will be built on top of the Liberty ID-WSF. Some of these services
534 will be specified within the Alliance context, other services will be proprietary applications built on top of the Liberty

ID-WSF architecture. It is anticipated that services such as wallet, calendar, messaging, presence, geo-location and user groups will be useful in conjunction with the Liberty ID-WSF. These services may be formally specified by the Alliance.

References

Informative

[RFC1034] Mockapetris, P., eds. (November 1987). "DOMAIN NAMES - CONCEPTS AND FACILITIES," RFC 1510, Internet Engineering Task Force *http://www.ietf.org/rfc/rfc1034.txt [November 1987]*.

[RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet Engineering Task Force (March 1997). *http://www.ietf.org/rfc/rfc2119.txt [March 1997]*.

[RFC2222] "Simple Authentication and Security Layer (SASL)," John G. Myers (October 1997). RFC 2222, Internet Engineering Task Force *http://www.ietf.org/rfc/rfc2222.txt [October 1997]*.

[RFC2246] Dierks, T., Allen, C., eds. (January 1999). "The TLS Protocol," Version 1.0 RFC 2246, Internet Engineering Task Force *http://www.ietf.org/rfc/rfc2246.txt [January 1999]*.

[SAMLCore11] Maler, Eve, Mishra, Prateek, Philpott, Rob, eds. (27 May 2003). "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1," OASIS Committee Specification, version 1.1, Organization for the Advancement of Structured Information Standards *http://www.oasis-open.org/committees/documents.php?wg_abbrev=security*

[SOAPv1.1] "Simple Object Access Protocol (SOAP) 1.1," Box, Don, Ehnebuske, David , Kakivaya, Gopal, Layman, Andrew, Mendelsohn, Noah, Nielsen, Henrik Frystyk, Winer, Dave, eds. World Wide Web Consortium W3C Note (08 May 2000). *http://www.w3.org/TR/2000/NOTE-SOAP-20000508/*

[SOAPv1.2] "SOAP Version 1.2 Part 1: Messaging Framework," Gudgin, Martin, Hadley, Marc, Mendelsohn, Noah, Moreau, Jean-Jacques, Nielsen, Henrik Frystyk, eds. World Wide Web Consortium W3C Proposed Recommendation (07 May 2003). *http://www.w3.org/TR/2003/PR-soap12-part1-20030507/*

[SSL] Frier, A., Karlton, P., Kocher, P., eds. (November 1996). Netscape Communications Corporation "The SSL 3.0 Protocol," *http://www.netscape.com/eng/ssl3/*

[WSDLv1.1] "Web Services Description Language (WSDL) 1.1," Christensen, Erik, Curbera, Francisco, Meredith, Greg, Weerawarana, Sanjiva, eds. World Wide Web Consortium W3C Note (15 March 2001). *http://www.w3.org/TR/2001/NOTE-wsdl-20010315*

[wss-sms] Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds. (January, 2004). Organization for the Advancement of Structured Information Standards *http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf* "Web Services Security: SOAP Message Security," OASIS Standard V1.0 [OASIS 200401],

[XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C.M., Maler, Eve, eds. (Oct 2000). "Extensible Markup Language (XML) 1.0 (Second Edition)," Recommendation, World Wide Web Consortium *http://www.w3.org/TR/2000/REC-xml-20001006*

[xmlenc-core] Eastlake, Donald, Reagle, Joseph, eds. (December 2002). "XML Encryption Syntax and Processing," W3C Recommendation, World Wide Web Consortium *http://www.w3.org/TR/xmlenc-core/*

[XMLDsig] Eastlake, Donald, Reagle, Joseph, Solo, David, eds. (12 Feb 2002). "XML-Signature Syntax and Processing," Recommendation, World Wide Web Consortium *http://www.w3.org/TR/xmldsig-core*

[X.509] "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks," ITU-T (2000). ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-8:2000,

[LibertyAuthn] Hodges, Jeff, Aarts, Robert, eds. " Liberty ID-WSF Authentication Service Specification ," Version 1.0, Liberty Alliance Project (26 Feb 2004). *http://www.projectliberty.org/specs/ [http://www.projectliberty.org/specs/]*

[LibertyBindProf] Cantor, Scott, Kemp, John, Champagne, Darryl, eds. "Liberty ID-FF Bindings and Profiles Specification," Version 1.2-errata-v2.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[LibertyClientProfiles] Aarts, Robert, eds. (26 April 2004). "Liberty ID-WSF Profiles for Liberty enabled User Agents and Devices," version 1.0, Liberty Alliance Project *http://www.projectliberty.org/specs/*

[LibertyDisco] Sergent, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification," Version 1.1, Liberty Alliance Project (21 April 2004). *http://www.projectliberty.org/specs*

[LibertyDST] "Liberty ID-WSF Data Services Template Specification," Version 1.0, Liberty Alliance Project (08 November 2003). *http://www.projectliberty.org/specs* Kainulainen, Jukka, Ranganathan, Aravindan, eds.

[LibertyIDPP] Kellomaki, Sampo, eds. "Liberty Identity Personal Profile Service Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyIDWSFGuide] Weitzel, David, eds. (26 April 2004). "Liberty ID-WSF Impelmentation Guideline," Draft version 1.0-08, Liberty Alliance Project *http://www.projectliberty.org/specs/*

[LibertyInteract] Aarts, Robert, eds. "Liberty ID-WSF Interaction Service Specification," Version 1.0-errata-v2.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[LibertyGlossary] "Liberty Technical Glossary," Version 1.3-errata-v1.0, Liberty Alliance Project (12 Aug 2004). *http://www.projectliberty.org/specs* Hodges, Jeff, eds.

[LibertyMetadata] Davis, Peter, eds. "Liberty Metadata Description and Discovery Specification," Version 1.0-errata-v2.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[LibertyPAOS] Aarts, Robert, eds. "Liberty Reverse HTTP Binding for SOAP Specification," Version 1.0-errata-v1.0, Liberty Alliance Project (18 April 2004). *http://www.projectliberty.org/specs*

[LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version 1.2-errata-v2.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[LibertySecMech] Ellison, Gary, eds. "Liberty ID-WSF Security Mechanisms," Version 1.1, Liberty Alliance Project (18 April 2004). *http://www.projectliberty.org/specs*

[LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, eds. " Liberty ID-WSF SOAP Binding Specification ," Version 1.1, Liberty Alliance Project (3 May 2004). *http://www.projectliberty.org/specs*