Liberty ID-WSF Web Services Framework Overview

Version: 2.0

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

Contributors:
John Beatty, Sun Microsystems, Inc.
Carolina Canales-Valenzuela, Ericsson
Jeff Hodges, Sun Microsystems, Inc.
Gary Ellison, Sun Microsystems, Inc.
John Kemp, IEEE-ISTO
Jason Rouault, Hewlett-Packard Company
Robert Aarts, Nokia Corporation
Jukka Kainulainen, Nokia Corporation
Thomas Wason, IEEE-ISTO
Peter Thompson, IEEE-ISTO
Paul Madsen, Nippon Telegraph and Telephone Corporation
Makiko Aoyagi, Nippon Telegraph and Telephone Corporation

Abstract:

This is a *non-normative* document intended to provide an overview of the relevant features of the Liberty ID-WSF version 2.0 specifications. It provides a general introduction to the Liberty ID-WSF framework. The reader is assumed to have some familiarity with SOAP, WS-Security, WS-Addressing, SAML, XML, and basic concepts such as namespaces and URIs.

Filename: liberty-idwsf-overview-v2.0.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.** Implementers 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.

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

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 Alliance Project proposes 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 Core[LibertySecMech], ID-WSF SecMech SAML Profile[LibertySecMech20SAML], ID-WSF Interaction Service[LibertyInteract], ID-WSF Profiles for Liberty-enabled User Agents or Devices[LibertyClientProfiles], ID-WSF People Service[LibertyPeopleService], 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-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 SAML2.0[SAMLCore2]).

The audience for this document is technical managers and application developers. The reader is assumed to have some familiarity with SOAP[SOAPv1.1], WS-Addressing[WSAv1.0], WS-Security[wss-sms], SAML2.0[SAMLCore2] 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-Addressing, WS-Security, HTTP, WSDL1.1[WSDLv1.1]), XML[XML], SOAP, XML-Encryption[xmlenc-core], XML-Signature[XMLDsig], TLS[RFC4346] 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-WSF specifications. See Figure 1.

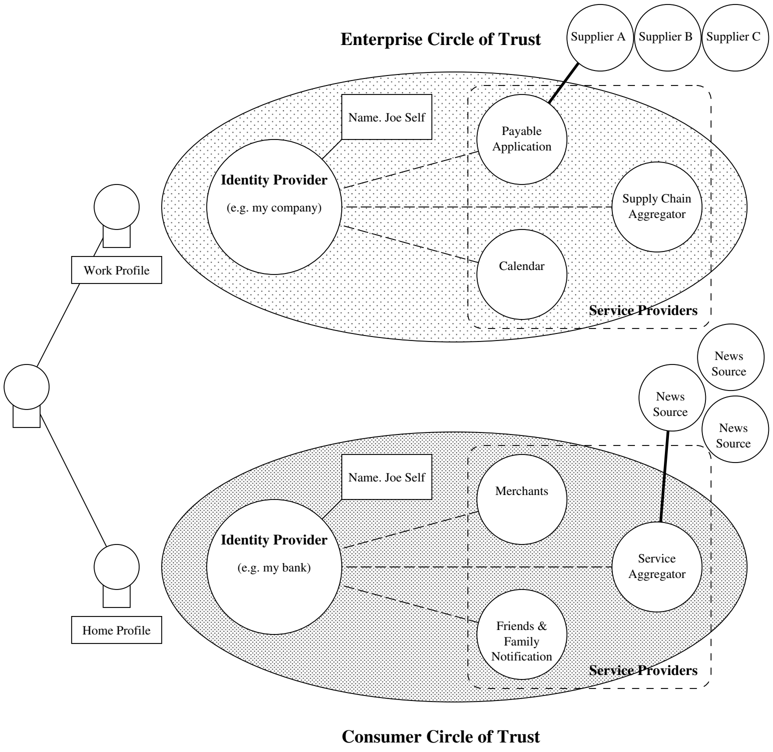


Figure 1. Federated Network Identity and Circles of Trust

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 SAML and/or Liberty-enabled products in their infrastructure, but do not require users to use anything other than today's common Web browser. Of course, Liberty solutions also allow the use of more sophisticated end-user devices if the user wishes it so, such as webservices-enabled terminals.

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.

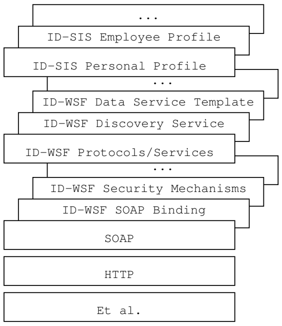


Figure 2. Liberty ID-WSF Protocol Architecture

1.5. Synopsis of Specifications

1.5.1. ID-WSF SOAP Binding

The ID-WSF SOAP Binding[LibertySOAPBinding] provides a SOAP-based invocation framework for identity services. It defines use of WS-Addressing[WSAv1.0] SOAP extensions as well as various SOAP Header blocks (such as provider declaration, processing context, consent claims, usage directives, and so on) and processing rules enabling the invocation of identity services via SOAP requests and responses.

1.5.2. ID-WSF Security Mechanisms

The ID-WSF Security Mechanisms Core[LibertySecMech] describes 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. This specification also defines an identity token that convey an identity between providers. The ID-WSF Security Mechanisms SAML Profile[LibertySecMech20SAML] describes profile of the SAML assertion and WS-Security SAML Token Profile[wss-saml11] in conjunction with the ID-WSF Security Mechanisms.

1.5.3. ID-WSF Discovery Service

The ID-WSF Discovery Service[LibertyDisco] defines a core identity service that enables various entities (e.g. Service Providers) to discover a Principal's registered identity services. Given the criteria of service desired (e.g. service type and security mechanisms), the Discovery Service responds with ID-WSF Endpoint References[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

The ID-WSF Data Services Template[LibertyDST] provides the building blocks when implementing a data service (e.g. ID-SIS Personal Profile Service[LibertyIDPP]) on top of the ID-WSF. The specification defines how to query, create, delete and modify data objects and their attributes stored in a data service, and provides some common attributes for data services. The specification facilitates identity-based web services to be defined using it as a template, but please note that the ID-WSF does not mandate to use this template to define any identity-based web services.

1.5.5. ID-WSF Subscriptions and Notifications

The ID-WSF Subscriptions and Notifications [LibertySUBS] provides a generic framework for notifications and management of such notifications, such as subscribing to receive them. The subscriptions are mechanisms through which a provider can request for notifications when specified events are happened at other providers. The notifications are mechanisms with which a provider notifies some events to providers that previously subscribed such the event to be notified. These features are not mandated to be used, but can be included in any request and response messages when designing identity-based web services on top of the ID-WSF framework.

1.5.6. ID-WSF Interaction Service

A provider of 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 providers. The ID-WSF Interaction Service[LibertyInteract] details protocols and profiles for interactions that allow providers to carry out such actions.

1.5.7. ID-WSF Profiles for Liberty-enabled User Agents or Devices

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 Liberty-enabled User Agent or Device (LUAD) is the user agent or device that has specific support for one or more profiles of the Liberty specifications[2] . 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 an ID-WSF Discovery Service.

1.5.8. Reverse HTTP Binding

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. ID-WSF Authentication, Single Sign-On, and Identity Mapping Services

---

[1] ID-WSF Endpoint Reference is the profiled Endpoint Reference of WS-Addressing[WSAv1.0].

[2] 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 LUAD.

In the ID-WSF context, Web Service Consumer (WSC) and/or Liberty-enabled User Agents or Devices (LUAD) may need to authenticate with an Identity Provider by exchanging SOAP messages. However, the SOAP specification[SOAPv1.1] does not specify any particular security mechanisms. This specification[LibertyAuthn] defines an authentication protocol between entities over SOAP, based on a profile of Simple Authentication and Security Layer framework[RFC4422]. Additionally, it defines ID-WSF Authentication Service that the Identity Provider may offer in the ID-WSF context. The ID-WSF Authentication Service enables WSC and/or LUAD to authenticate with Identity Providers based on the authentication protocol and to obtain ID-WSF security tokens. This specification also defines the Single Sign-On Service which enables WSC and/or LUAD to obtain SAML authentication assertions within the ID-WSF context, which can be used in the SAML context. In addition to these protocol and services, this specification defines the Identity Mapping Service with which WSC's obtains identity tokens for use in web service invocations and referencing principals while preserving privacy.

1.5.10. ID-WSF People Service

There exist many circumstances where a user wishes to access the identity service of another user. In such cases, it is necessary for one user to be able to obtain an identifier for another user from that user's Identity Provider, and to convey that identifier to this second user's identity services. Additionally, users will often desire to grant access rights to their browser-based resources to friends - this implies that the privileges can be assigned to a relevant identifier for that friend as known by an appropriate Identity Provider. This specification [LibertyPeopleService] describes an architecture for allowing secure, privacy-respecting access by one user to another's identity information (both browser-based and programmatic services), and normatively defines the Liberty ID-WSF People Service, one component of such an architecture. The specification also defines schema definitions and protocols for manipulating group information - this allows a principal to categorize their friends, colleagues, and family etc.

2. User Experience Examples

This section provides 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. Multiple website scenario

In this section, a simple ID-WSF user experience example is described, in which a principal Joe Self is ordering beer and pizza on the Internet. More details of this example from the implementation 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

One day on Sunday, Joe decides to order beers at BlueLiquor website that is his favorite liquor shop. When he tries to access to the 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.

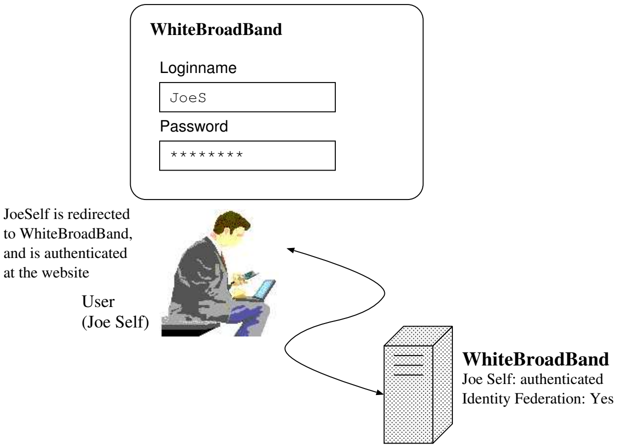


Figure 3. Joe Self is redirected to WhiteBroadBand website that is his Identity Provider

He orders a dozen beers at the BlueLiquor website, and asks 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.

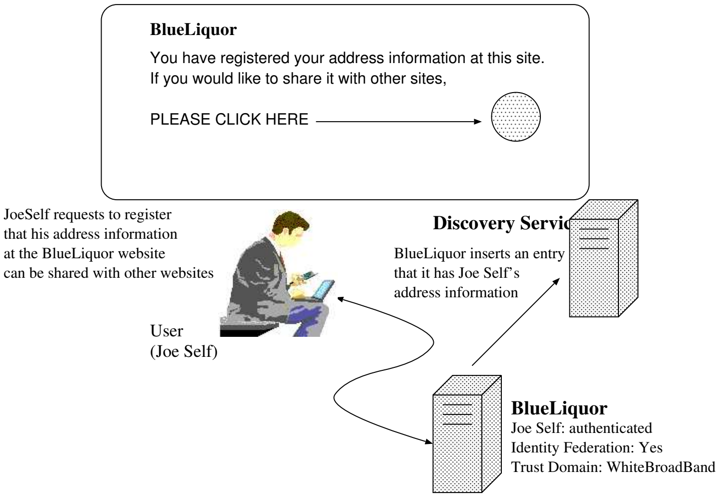

BlueLiquor
You have registered your address information at this site.
If you would like to share it with other sites,
PLEASE CLICK HERE
JoeSelf requests to register
that his address information
at the BlueLiquor website
can be shared with other websites
User
(Joe Self)
Discovery Servic
BlueLiquor inserts an entry
that it has Joe Self’s
address information
BlueLiquor
Joe Self: authenticated
Identity Federation: Yes
Trust Domain: WhiteBroadBand

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 pepperoni 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, a dozen beers and the mayonnaise pizza are delivered to his residence.

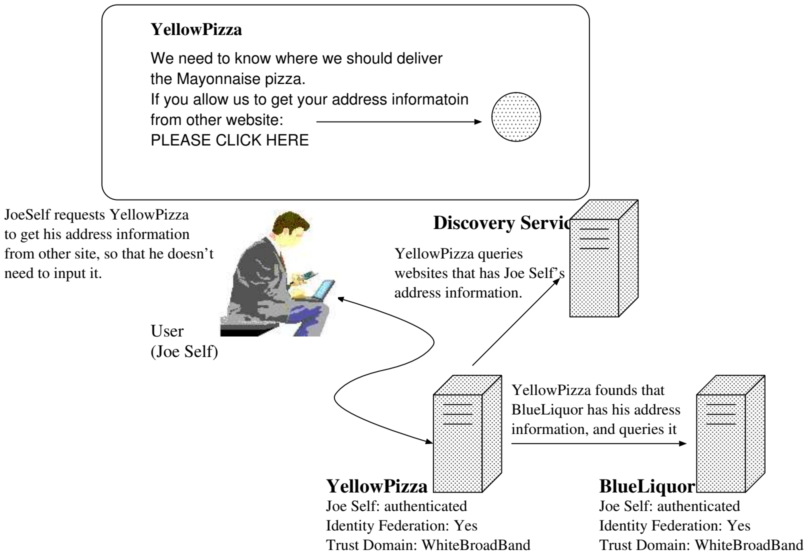

YellowPizza
We need to know where we should deliver
the Mayonnaise pizza.
If you allow us to get your address informatoin
from other website:
PLEASE CLICK HERE
JoeSelf requests YellowPizza
to get his address information
from other site, so that he doesn't
need to input it.
User
(Joe Self)
Discovery Servic
YellowPizza queries
websites that has Joe Self's
address information.
YellowPizza founds that
BlueLiquor has his address
information, and queries it
YellowPizza
Joe Self: authenticated
Identity Federation: Yes
Trust Domain: WhiteBroadBand
BlueLiquor
Joe Self: authenticated
Identity Federation: Yes
Trust Domain: WhiteBroadBand

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

2.2. Mobile IdP Scenario

This section describes a mobile scenario.

2.2.1. Actors

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 companies that is able to get customer's personal information from other websites.

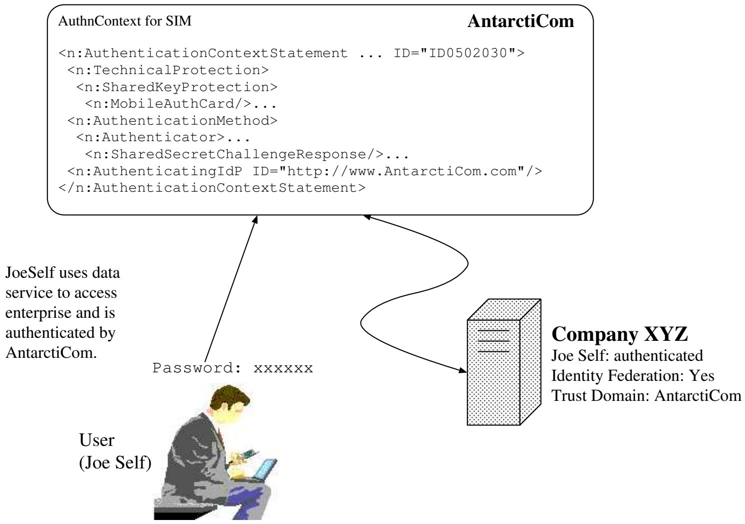
2.2.2. Assumptions

This scenario 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.

2.2.3. User Experience

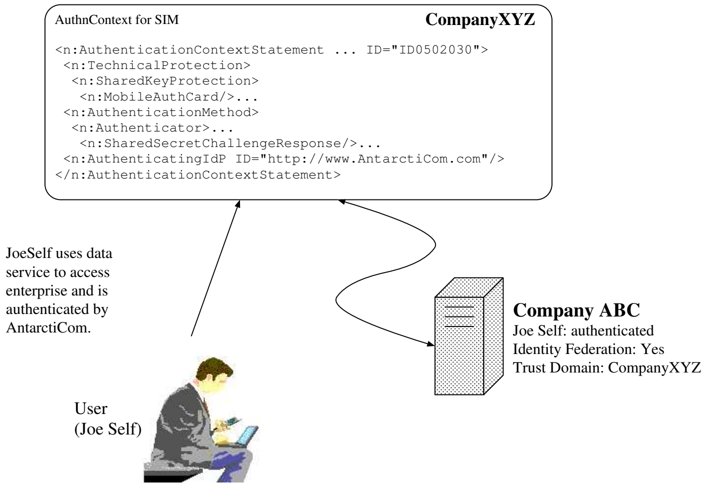
Joe Self is on the road at a big conference. He is presenting on widgets and their importance to Company XYZ's businesses. After his big presentation, he decides to access his corporate web portal with his browser in order to check his e-mail. He turns on his Mobile Data device, such as a GSM phone with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



```
<n:AuthenticationContextStatement ... ID="ID0502030">
 <n:TechnicalProtection>
  <n:SharedKeyProtection>
   <n:MobileAuthCard/>...
 <n:AuthenticationMethod>
  <n:Authenticator>...
   <n:SharedSecretChallengeResponse/>...
 <n:AuthenticatingIdP ID="http://www.AntarctiCom.com"/>
</n:AuthenticationContextStatement>
```

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.



```
<n:AuthenticationContextStatement ... ID="ID0502030">
 <n:TechnicalProtection>
  <n:SharedKeyProtection>
   <n:MobileAuthCard/>...
 <n:AuthenticationMethod>
  <n:Authenticator>...
   <n:SharedSecretChallengeResponse/>...
 <n:AuthenticatingIdP ID="http://www.AntarctiCom.com"/>
</n:AuthenticationContextStatement>
```

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.

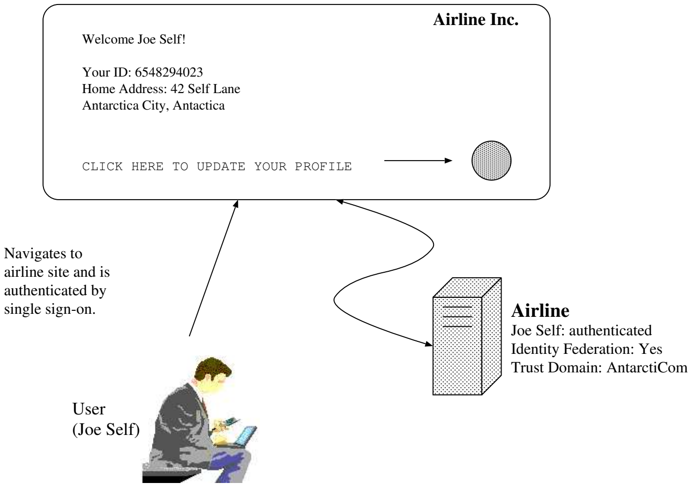


Figure 8. Joe Self Navigates to Airline site, uses AntarctiCom as Identity Provider

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.

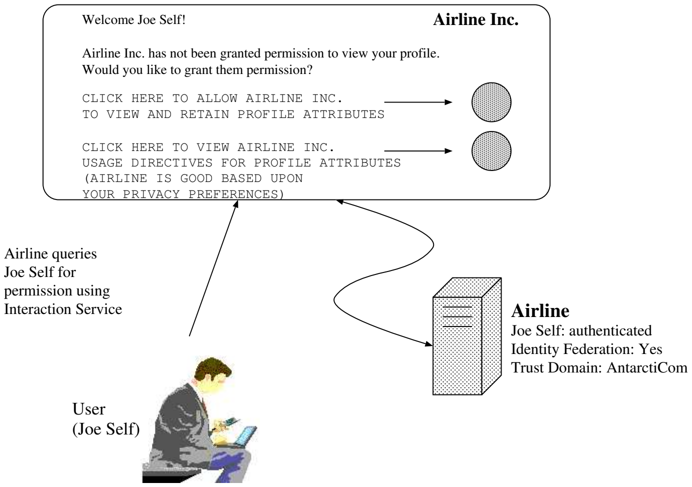

Welcome Joe Self!
Airline Inc.
Airline Inc. has not been granted permission to view your profile.
Would you like to grant them permission?
CLICK HERE TO ALLOW AIRLINE INC.
TO VIEW AND RETAIN PROFILE ATTRIBUTES
CLICK HERE TO VIEW AIRLINE INC.
USAGE DIRECTIVES FOR PROFILE ATTRIBUTES
(AIRLINE IS GOOD BASED UPON
YOUR PRIVACY PREFERENCES)
Airline queries
Joe Self for
permission using
Interaction Service
Airline
Joe Self: authenticated
Identity Federation: Yes
Trust Domain: AntarctiCom
User
(Joe Self)

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[3] . 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.

---

[3] 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.

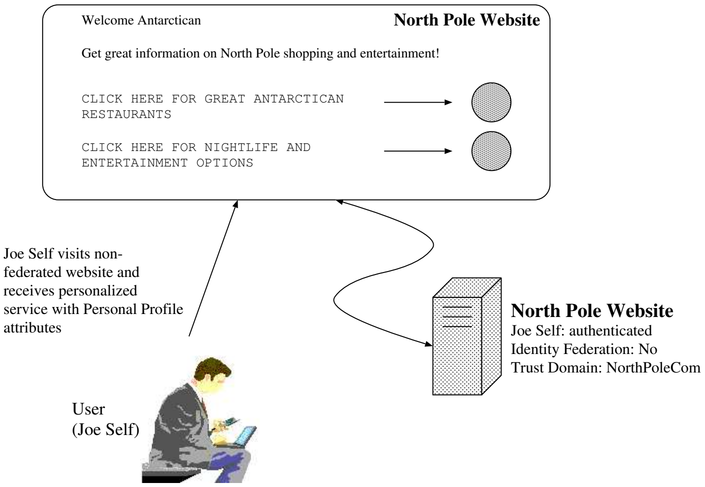

Welcome Antarctican
North Pole Website
Get great information on North Pole shopping and entertainment!
CLICK HERE FOR GREAT ANTARCTICAN RESTAURANTS
CLICK HERE FOR NIGHTLIFE AND ENTERTAINMENT OPTIONS
Joe Self visits non-federated website and receives personalized service with Personal Profile attributes
North Pole Website
Joe Self: authenticated
Identity Federation: No
Trust Domain: NorthPoleCom
User
(Joe Self)

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

This mobile device example demonstrates a scenario with 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 Authentication Service[LibertyAuthn] for authentication of the LUAD or ECP[SAMLProf2].

2.3. Cross-Principal Browser-based Resource Sharing

This example demonstrates the message flow by which a user is able to assign access rights for a particular resource to a friend

Alice maintains her photos at photos.example.com. Upon returning from a vacation in the Caribbean, she uploads her latest photos, creating a new album called ’Vacation Pics’. Alice wants to share these pictures with a friend named Bob.

2.3.1. Actors

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

- Alice: maintains vacation photos online.
- Bob: a friend of Alice with whom she wants to share photos.
- photos.example.com: Online photo site
- friends.idp.com: Alice’s People Service Provider.
- idp.com: Bob’s Identity Provider.

2.3.2. Assumptions

We make the following assumptions:

- Bob does not have (nor wish to create) an account at photos.example.com.
- Alice is federated between photos.example.com and her identity provider.

2.3.3. User Experience

Alice wants to share her photos at the photo Service Provider (photos.example.com) with her friend who she calls as Bob. The high-level steps are as follows:

(1) Alice SSOs in to photos.example.com site and indicates she wants to share a photo with a friend.

(2) photos.example.com discovers the location of Alice's PS (through standard Liberty mechanisms) and, once determined, sends a query to friends.idp.com for the members of Alice's list of friends.

(3) After determining that photos.example.com is authorized to act on Alice's behalf, friends.idp.com returns the list of members to photos.example.com which then displays the list to Alice (e.g. through an HTML form).

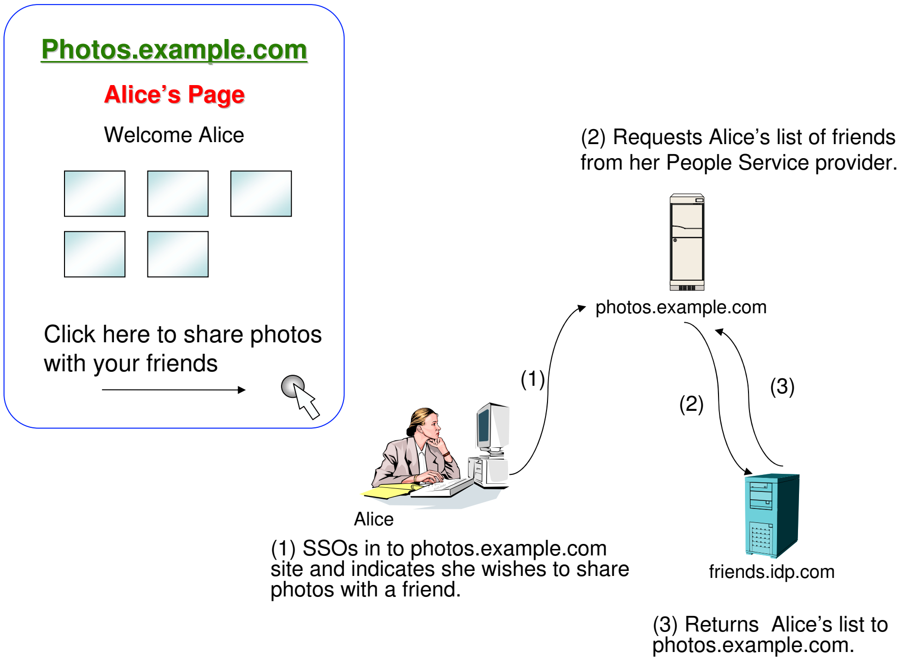
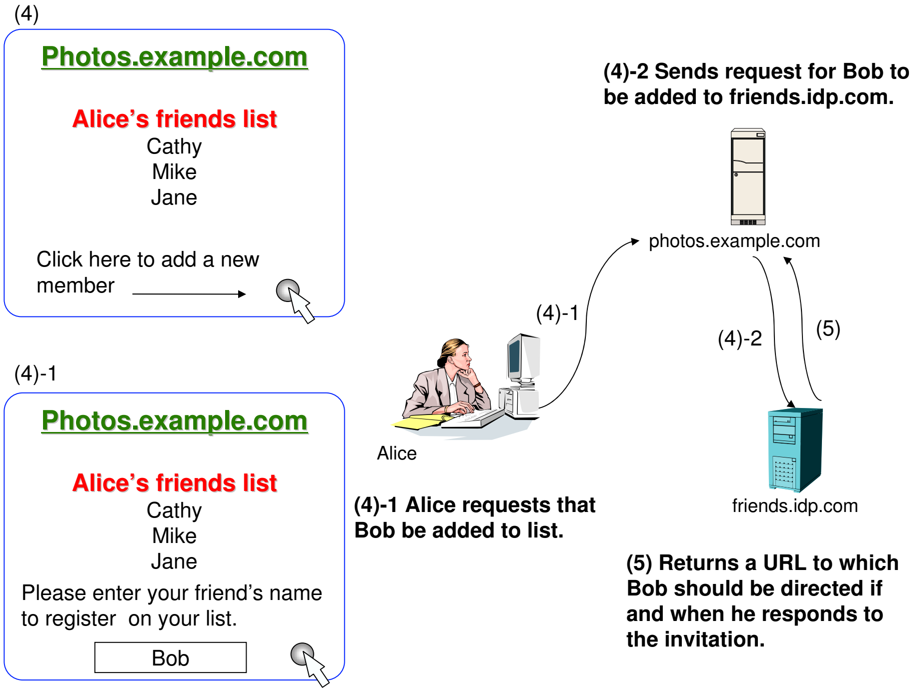

Photos.example.com
Alice's Page
Welcome Alice
Click here to share photos with your friends
(2) Requests Alice's list of friends from her People Service provider.
photos.example.com
(1)
(2)
(3)
Alice
(1) SSOs in to photos.example.com site and indicates she wishes to share photos with a friend.
friends.idp.com
(3) Returns Alice's list to photos.example.com.

Figure 11. Alice sets up to share her photos

(4) As this list is composed of individuals with whom Alice has previously established an online connection, and this is the first time she has reached out to Bob, he is not in the list. Alice asks photos.example.com to request that 'Bob' be added to her list. photos.example.com sends the appropriate request to friends.idp.com for Bob to be added.

(5) friends.idp.com returns a URL to which Bob should be directed if and when he responds to the (upcoming) invitation.



(4)-2 Sends request for Bob to be added to friends.idp.com.

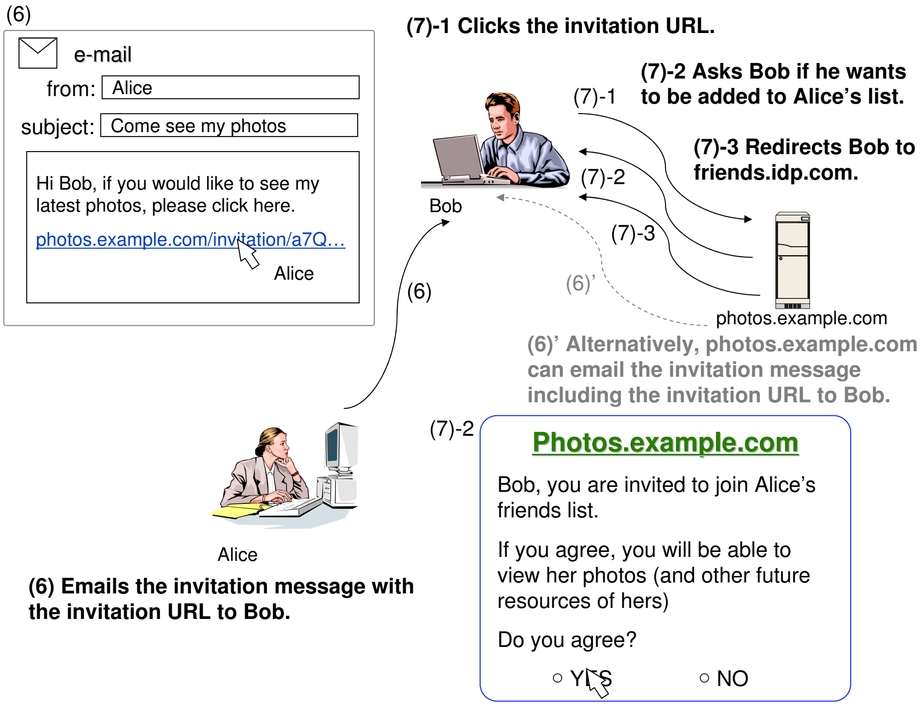
(4)-1 Alice requests that Bob be added to list.

(5) Returns a URL to which Bob should be directed if and when he responds to the invitation.

Figure 12. Alice adds Bob to her friends list

(6) photos.example.com creates an invitation for Bob indicating Alice's desire to share her photos. This invite is made available to Alice (e.g. (in an HTML page for copying) so that she can communicate it directly to Bob. Alice emails the invite message to Bob. Alternatively, photos.example.com could have sent the invited directly if Alice had provided Bob's email.

(7) Bob clicks on the URL within the invite message and is taken to photos.example.com where he can get more information about the nature of the invitation. If he consents to proceeding, and to allowing Alice to record the connection between them, he is directed to the URL friends.idp.com previously supplied.



(6)

e-mail
from: Alice
subject: Come see my photos

Hi Bob, if you would like to see my latest photos, please click here.

photos.example.com/invitation/a7Q…

Alice

(7)-1 Clicks the invitation URL.

(7)-2 Asks Bob if he wants to be added to Alice's list.

(7)-3 Redirects Bob to friends.idp.com.

(6)' Alternatively, photos.example.com can email the invitation message including the invitation URL to Bob.

(6) Emails the invitation message with the invitation URL to Bob.

(7)-2

Photos.example.com

Bob, you are invited to join Alice's friends list.

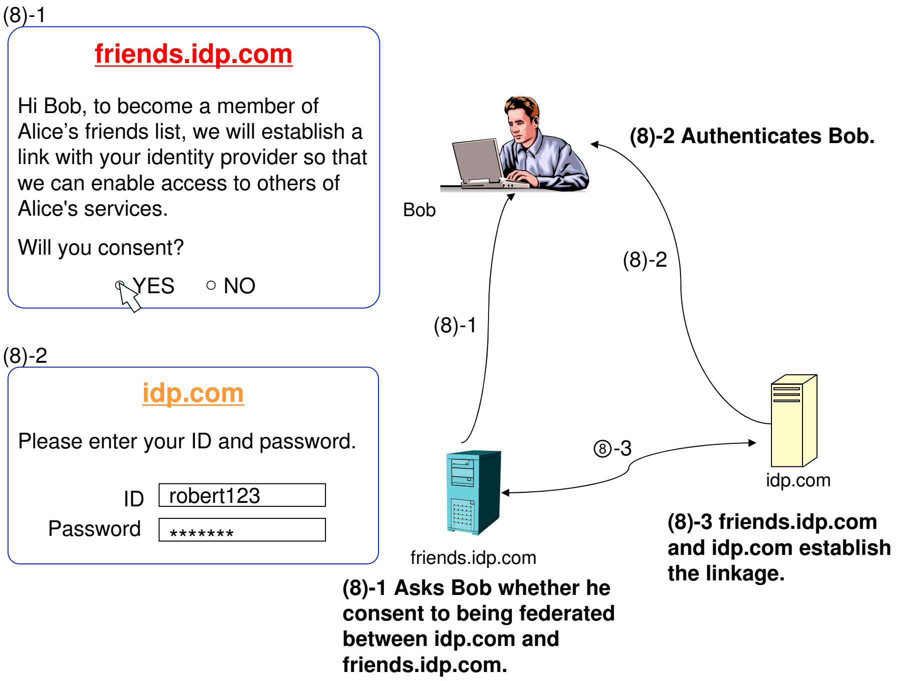
If you agree, you will be able to view her photos (and other future resources of hers)

Do you agree?

○ YES ○ NO

Figure 13. Alice sends invitation message to Bob

(8) friends.idp.com asks Bob if he would like to establish a linkage with an account he maintains at some other identity provider. If Bob consents (and assuming that the necessary business relationship exists), friends.idp.com and idp.com establish this linkage (in the lingo, they federate Bob).



(8)-1

friends.idp.com

Hi Bob, to become a member of Alice's friends list, we will establish a link with your identity provider so that we can enable access to others of Alice's services.

Will you consent?

○ YES ○ NO

(8)-2

idp.com

Please enter your ID and password.

ID robert123
Password *******

(8)-2 Authenticates Bob.

(8)-3 friends.idp.com and idp.com establish the linkage.

(8)-1 Asks Bob whether he consent to being federated between idp.com and friends.idp.com.

Figure 14. Bob federates his identities between his Identity Provider and Alice's Friends List Service Provider

(9) friends.idp.com can now ask idp.com for a identifier that photos.example.com could use for Bob. Idp.com generates such an identifier, encrypts it so that friends.idp.com can't see it, and returns it to friends.idp.com.

(10) friends.idp.com delivers this encrypted identifier for Bob to photos.example.com which, after decrypting, assigns appropriate privileges to this new identifier.

(11) The next time Bob accesses photos.example.com (by first signing on at idp.com), photos.example.com will recognize him as somebody to Alice has assigned specific permissions, and he will be able to view the relevant photos.

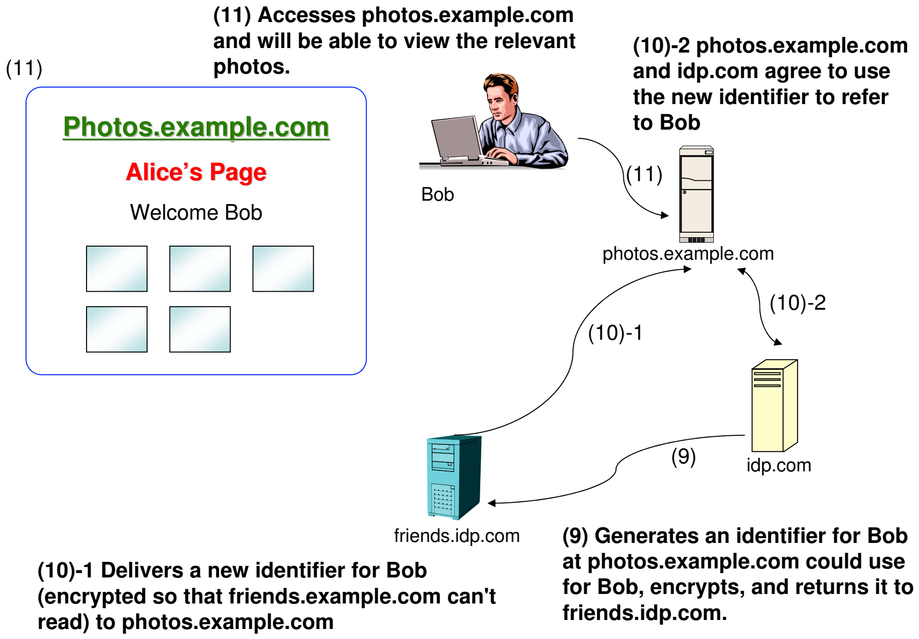

(11) Accesses photos.example.com and will be able to view the relevant photos.
(11)
Photos.example.com
Alice's Page
Welcome Bob
Bob
(10)-2 photos.example.com and idp.com agree to use the new identifier to refer to Bob
(11)
photos.example.com
(10)-2
(10)-1
(9)
idp.com
friends.idp.com
(9) Generates an identifier for Bob at photos.example.com could use for Bob, encrypts, and returns it to friends.idp.com.
(10)-1 Delivers a new identifier for Bob (encrypted so that friends.example.com can't read) to photos.example.com

Figure 15. Bob federates his identities between his Identity Provider and Alice's Photo Service Provider

2.4. Cross-principal Identity Service Access

This example demonstrates the message flow by which a WSC, acting on behalf of one principal, is able to discover and invoke an identity service of a different principal.

2.4.1. Overview

Bob has previously invited his friend Alice to access some WSF-based resource (geolocation for the sake of this exercise) of his at one of his providers, WSPb. In the process of Alice responding to the invited, in addition to Alice getting added to Bob's PS (PSb), Bob is added to Alice's PS (PSa). A necessary precondition for both these additions is that federations are established between Alice's IDP (IDPa) and PSb and also between Bob's IDP (IDPb) and PSa. Subsequently, when Alice is at one of her WSCs (WSCa), she indicates that she wishes to access the resource of Bob's (which is at WSPb but WSCa doesn't know that yet).

2.4.2. Assumptions

The assumption is that both Alice and Bob are 'known' at their respective IDPs, that their relevant services (including their PSs) are registered at their DSs. At each of their PSs, there is an entry for the other (e.g. Alice has Bob in hers and vice versa).

2.4.3. Federations

1. WSCa has federated Alice with IDPa
2. WSPb has federated Bob with IDPb
3. PSb has federated Alice with IDPa - this came out of the original invitation process. When Alice responded to the invite, she ended up at PSb, where she was given the option of federating with her IDPa.
4. PSa has federated Bob with IDPb - this came out of the reciprocal 'invitation' process. In responding to Bob's invite, Alice was given the option of 'Do you wish to add your friend to your friends list?'. Alice said yes
5. WSPb has federated Alice with IDPa - this was the result of an invitation that was sent to Alice by Bob for accessing her WSPb resource. WSPb has defined appropriate permissions for Alice's geolocation resource in terms of this federated identifier. Consequently, if Alice were to simply SSO in from IDPa to WSPb (and if WSPb provided a browser interface) she would be able to access Bob's resource (perhaps with a different set of privileges than when presented through SOAP interface)

2.4.4. Sequence

The following diagram indicates the steps involved.

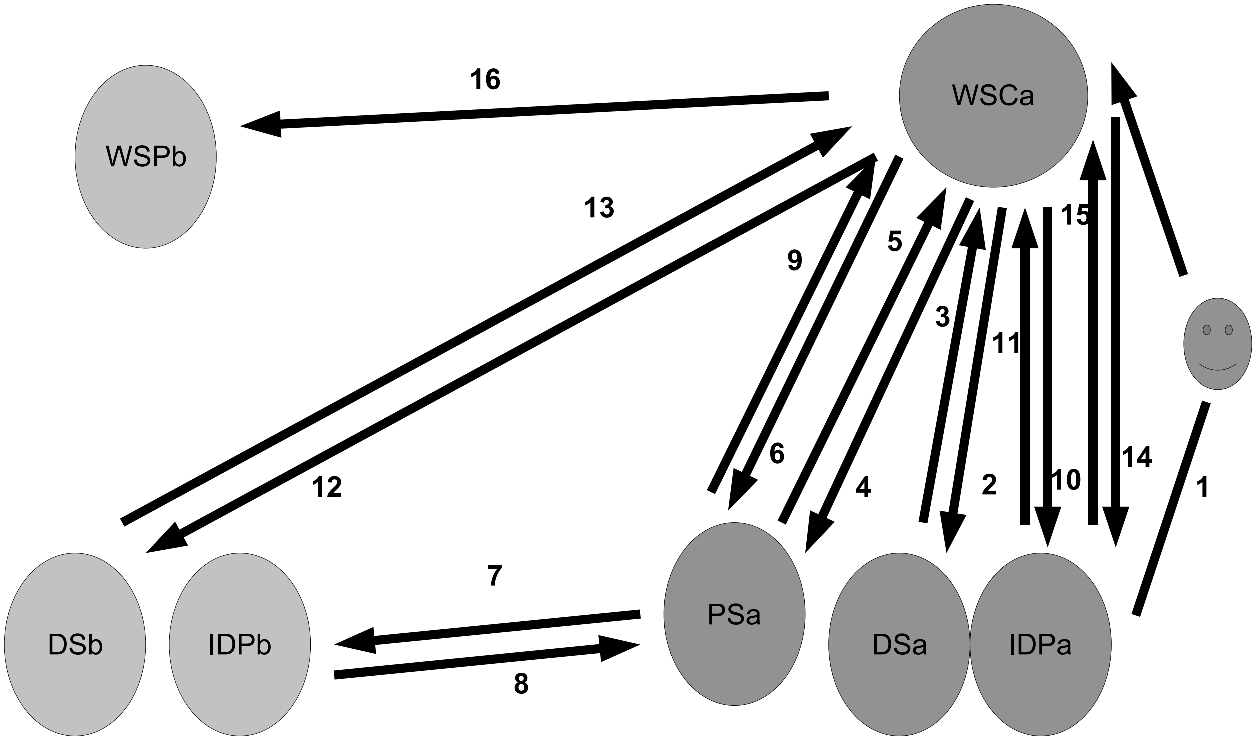


Figure 16. Sequence

The high-level steps corresponding to the numbered messages above are as follows:

1. Alice receives an SSO assertion from IDPa to be used at WSCa. This assertion (as is normal) contains a bootstrap EPR for Alice's DS (DSa).

2. The bootstrap EPR contains address of, and credentials to be used at, DSa. WSCa queries DSa for the location of Alice's PS.

3. DSa returns an EPR for Alice's PS (PSa).

4. WSCa uses the credentials in the EPR to ask PSa for the members of Alice's PS list.

5. PSa returns the appropriate list.

6. WSCa, based on Alice's indicated desire, determines which of the list members Alice is interested in (in this case Alice sees Bob in the list and says 'That guy'). Once determined, WSCa sends a `<ResolveIdentifierRequest>` message to PSa for the ObjectID associated with Bob's Object.

7. Although PSa has a long-lived identity token for Bob from his IDP, it's not targeted at WSCa. PSa gets the appropriate token for WSCb by sending a `<sa:IdentityMappingRequest>` message to IDPb, asking for a token for Bob and specifying WSCa as the target provider.

8. IDPb returns an appropriate token for Bob in the `<sa:IdentityMappingResponse>`. Included is an EPR for Bob's DS (DSb). The EPR does not have a security token for Alice

9. PSa forwards on the token just received from IDPb to WSCa in its `<ps:ResolveIdentifierResponse>`.

10. WSCa recognizes that it needs a token for Alice to use at DSb and so asks IDPa using the Token Request service, specifying DSb as the target.

11. IDPa returns a security token for Alice at DSb in its `Response>`

12. WSCa asks DSb for Bob's geolocation service using a disco query. The invocation identity (implicit) is Alice and the target identity is Bob

13. DSb returns EPR for Bob's geolocation service at WSPb. The EPR does not have a security token for Alice

14. WSCa recognizes that it needs a token for Alice to use at WSPb and so asks IDPa using the Token Request service, specifying WSPb as the target.

15. IDPa returns a security token for Alice at WSPb in its `<Response>`

16. WSCa invokes WSPb. The invocation identity (implicit) is Alice and the target identity is Bob

2.4.4.1. Detailed Messages

2.4.4.1.1. Message 1 - SSO Assertion with Bootstrap EPR - from IDPa to WSCa

```

<Response>
  <Assertion ID="firstassertion">
     <Subject>
   <NameID NameQualifier="IDPa" SPNameQualifier="WSCa" Format="persistent">Alice</NameID>
     </Subject>
    <AuthnStatement>
    </AuthnStatement>
    <AttributeStatement>
      <Attribute Name="disco-epr">
        <AttributeValue>
          <EndpointReference>
            <Address>DSa.com/disco</Address>
            <Metadata>
              <ProviderID>DSa</ProviderID>
              <ServiceType>disco</ServiceType>
              <SecurityContext>

                 <SecurityMechID>bearer</SecurityMechID>
 <Token ref="#firstassertion" usage="securitytoken"/>

 add confirmations & namespaces

              </SecurityContext>
            </Metadata>
          </EndPointReference>
        </AttributeValue>
      </Attribute>
    </AttributeStatement>
  </Assertion>
</Response>

```

2.4.4.1.2. Message 2 - Disco Query - from WSCa to DSa

```

<Envelope>
  <Header>
    <To>DSa.com</To>
    <Security>
  <Assertion ID="firstassertion">
     <Subject>
   <NameID NameQualifier="IDPa" SPNameQualifier="WSCa" Format="persistent">Alice</NameID>
     </Subject>
    <AuthnStatement>
    </AuthnStatement>
    <AttributeStatement>
      <Attribute Name="disco-epr">
        <AttributeValue>
          <EndpointReference>
            <Address>DSa.com/disco</Address>
            <Metadata>
              <ProviderID>DSa</ProviderID>
              <ServiceType>disco</ServiceType>
              <SecurityContext>

                 <SecurityMechID>bearer</SecurityMechID>
 <Token ref="#firstassertion" usage="securitytoken">
                </Token>

 add confirmations & namespaces

```

```
              </SecurityContext>
            </Metadata>
          </EndPointReference>
        </AttributeValue>
      </Attribute>
    </AttributeStatement>
  </Assertion>
    </Security>
  </Header>
  <Body>
    <Query>
      <RequestedServiceType>
        <ServiceType>peopleservice</ServiceType>
      </RequestedServiceType>
    </Query>
  </Body>
</Envelope>
```

2.4.4.1.3. Message 3 - Disco QueryResponse - from DSa to WSCa

```
<Envelope>
  <Header>
    <To>WSCa.com</To>
  </Header>
  <Body>
    <QueryResponse>
      <EndpointReference>
        <Address>PSa.com/disco</Address>
        <Metadata>
          <ProviderID>PSa</ProviderID>
          <ServiceType>peopleservice</ServiceType>
          <SecurityContext>
          <Token>
            <Assertion id="secondassertion">
            </Assertion>
           </Token>
         </SecurityContext>
       </Metadata>
     </EndPointReference>
    </QueryResponse>
  </Body>
</Envelope>
```

2.4.4.1.4. Message 4 - PS ListMembersRequest - from WSCa to PSa

```
<Envelope>
  <Header>
    <To>PSa.com</To>
    <Security>
      <Assertion ID="secondassertion">
        <Subject>
     <EncryptedID>
           <EncryptedData>
        <NameID NameQualifier="IDPa" SPNameQualifier="PSa" Format="persistent">Alice</NameID>
           </EncryptedData>
          <EncryptedKey></EncryptedKey>
          </EncryptedID>
        </Subject>
      </Assertion>
    </Security>
  </Header>
  <Body>
    <ListMembersRequest/>
  </Body>
</Envelope>
```

2.4.4.1.5. Message 5 - PS ListMembersResponse - from PSa to WSCa

```
<Envelope>
  <Header>
    <To>WSCa.com</To>
  </Header>
  <Body>
    <ListMembersResponse>
      <Object>
        <ObjectID>kkk</ObjectID>
        <DisplayName>Bob</DisplayName>
      </Object>
    </ListMembersResponse>
  </Body>
</Envelope>
```

2.4.4.1.6. Message 6 - PS ResolveIdentifierRequest - from WSCa to PSa

WSC asks for transient because it does not care

```
<Envelope>
  <Header>
    <To>PSa.com</To>
    <Security>
      <Assertion ID="secondassertion">

      </Assertion>
    </Security>
  </Header>
  <Body>
    <ResolveIdentifierRequest>
      <ResolveInput>
     <TokenPolicy wantEPR="1">
         <NameIDPolicy type="transient"/>
     </TokenPolicy>
        <TargetID>kkk</TargetID>
      </ResolveInput>
    </ResolveidentifiersRequest>
  </Body>
</Envelope>
```

2.4.4.1.7. Message 7 - AS IdentityMappingRequest - from PSa to IMSb

PSa is able to discover IMSb through the long-lived DSb EPR it has.

Alternatively, if the EPR the DSb originally constructed used TargetIdentity, then Bob’s identity could be carried through the TargetIdentity header in the call.

```
<Envelope>
  <Header>
    <To>IDPb.com</To>
    <Security>
```

```
      <Assertion ID="thirdassertion">
        <Subject>
     <NameID NameQualifier="IDPb" SPNameQualifier="PSa" Format="persistent">Bob</NameID>
        </Subject>
       SubjectConfirmation identifies PSa
      </Assertion>
    </Security>
  </Header>
  <Body>
    <IdentityMappingRequest>
      <MappingInput>
         <TokenPolicy>
            <NameIDPolicy SPNameQualifier="WSCa"/>
         </TokenPolicy>
         <Token ref="#thirdassertion" usage="securitytoken">
       </MappingInput>
    </IdentityMappingRequest>
  </Body>
</Envelope>
```

2.4.4.1.8. Message 8 - AS IdentityMappingResponse - from IMSb to PSa

```
<Envelope>
  <Header>
    <wsa:To>PSa.com</wsa:To>
  </Header>
  <Body>
    <IdentityMappingResponse>
      <Status>OK</Status>
      <MappingOutput>
      <Token>
        <EncryptedAssertion><EncryptedData>
          <Assertion ID="fourthassertion">
          <Subject>
         <NameID NameQualifier="IDPb" SPNameQualifier="WSCa" Format="saml:transient">Bob</NameID>
          </Subject>
          <AttributeStatement>
            <Attribute Name="disco-epr">
              <AttributeValue>
                <wsa:EndpointReference>
                  <wsa:Address>DSb.com</wsa:Address>
                  <wsa:Metadata>
                    <ProviderID>DSb</ProviderID>
                    <ServiceType>disco</ServiceType>
                    <SecurityContext>
                      <sec:Token usage="SecurityToken" ref=":ObtainFromIDP"/>
                      <sec:Token usage=":TargetIdentity" ref="fourthassertion"/>
                    </SecurityContext>
                  </wsa:Metadata>
                </wsa:EndpointReference>
              </AttributeValue>
            </Attribute>
          </AttributeStatement>
         </Assertion>
        </EncryptedData>
        <EncryptedKey></EncryptedKey>
        </EncryptedAssertion>
      </sec:Token>
      </sa:MappingOutput>
    </sa:IdentityMappingResponse>
  </Body>
</Envelope>
```

2.4.4.1.9. Message 9 - PS ResolveIdentifierResponse - from PSa to WSCa

```
<Envelope>
  <Header>
    <wsa:To>WSCa.com</wsa:To>
  </Header>
  <Body>
    <ps:ResolveIdentifierResponse>
      <ps:Status>OK</ps:Status>
        <ps:ResolveOutput>
        <sec:Token>
         <EncryptedAssertion><EncryptedData>
          <Assertion ID="fourthassertion">
          <Subject>
        <NameID NameQualifier="IDPb" SPNameQualifier="WSCa" Format="saml:transient">Bob</NameID>
          </Subject>
          <AttributeStatement>
            <Attribute Name="disco-epr">
              <AttributeValue>
                <wsa:EndpointReference>
                  <wsa:Address>DSb.com</wsa:Address>
                  <wsa:Metadata>
                    <ProviderID>DSb</ProviderID>
                    <ServiceType>disco</ServiceType>
                    <SecurityContext>
                      <sec:Token usage="SecurityToken" ref=":ObtainFromIDP"/>
                      <sec:Token usage=":TargetIdentity" ref="fourthassertion"/>
                    </SecurityContext>
                  </wsa:Metadata>
                </wsa:EndpointReference>
              </AttributeValue>
            </Attribute>
          </AttributeStatement>
         </Assertion>
        </EncryptedData>
        <EncryptedKey></EncryptedKey>
        </EncryptedAssertion>
      </sec:Token>
      </ps:ResolveOutput>
    </ps:ResolveIdentifierResponse>
  </Body>
</Envelope>
```

2.4.4.1.10. Message 10 - AS Token Request - from WSCa to IDPa

WSCa uses the original SSO token as the security context of the Token request. Alternatively, WSCa could discover IMSa and get credentials from DSa.

```
<Envelope>
  <Header>
    <To>IDPa.com</To>
    <Security>
     <Assertion ID="firstassertion">
     <Subject>
  <NameID NameQualifier="IDPa" SPNameQualifier="WSCa" Format="persistent">Alice</NameID>
     </Subject>
    <AuthnStatement>
    </AuthnStatement>
    <AttributeStatement>
      <Attribute Name="disco-epr">
        <AttributeValue>
          <EndpointReference>
            <Address>DSa.com/disco</Address>
            <Metadata>
              <ProviderID>DSa</ProviderID>
              <ServiceType>disco</ServiceType>
              <SecurityContext>
                <Token usage="SecurityToken" ref="#firstassertion"/>
              </SecurityContext>
            </Metadata>
          </EndPointReference>
        </AttributeValue>
      </Attribute>
    </AttributeStatement>
    </Assertion>
    </Security>
  </Header>
  <Body>
    <samlp:AuthnRequest>
      <samlp:Conditions>
        <samlp:AudienceRestriction>
          <samlp:Audience>DSb.com</saml:Audience>
        </samlp:AudienceRestriction>
      </samlp:Conditions>
      <samlp:ProtocolBinding>
      http://www.w3.org/2005/08/addressing/anony mous
      </samlp:ProtocolBinding>
    </samlp:AuthnRequest>
  </Body>
</Envelope>

```

2.4.4.1.11. Message 11 - AS Token Response - from IDPa to WSCa

Shown here is IDPa returning to WSCa a persistent identifier for Alice at DSb, this dependent on a previous federation being established for Alice between IDPa and DSb.

Alternatively, if no such federation existed for Alice, IDPa could return a transient identifier for Alice to be used by WSCa in its discovery query to DSb. In this case, if DSb enforced no access control policy over releasing Bob's EPRs, the transient identifier for Alice is sufficient. If however, DSb does enforce access control for Bob's EPRs (or just some of them), then DSb will need additional steps to determine if Alice is authorized (e.g. resolving the transient identifier through Bob's PS.

```

<Envelope>
  <Header>
    <To>WSCa.com</To>
  </Header>
  <Body>
    <samlp:Response>
      <Status>
        <StatusCode Value="urn:Success"/>
      </Status>
      <saml:Assertion ID="fifthassertion">
        <saml:Subject>
          <saml:EncryptedID><xenc:Encr yptedData>
            <saml:NameID Format="persistent" NameQualifier="IDPa" SPNameQualifier="DSb">Alice</saml:NameID>

          </xenc:EncryptedData>
          <xenc:EncryptedKey></xenc:EncryptedKey>
          </saml:EncryptedID>
        </saml:Subject>
        <saml:AuthnStatement>
        </saml:AuthnStatement>
      </saml:Assertion>
    </samlp:Response>
  </Body>
```

```
</Envelope>
```

2.4.4.1.12. Message 12 - Disco Query - from WSCa to DSb

```
<Envelope>
  <Header>
    <wsa:To>DSa.com</wsa:To>
    <Security>
      <saml:Assertion ID="fifthassertion">
        <saml:Subject>
          <saml:EncryptedID><xenc:EncryptedData>
            <saml:NameID Format="persistent" NameQualifier="IDPa" SPNameQualifier="DSb">Alice</saml:NameID>

          </xenc:EncryptedData>
          <xenc:EncryptedKey></xenc:EncryptedKey>
          </saml:EncryptedID>
        </saml:Subject>
        <saml:AuthnStatement>
        </saml:AuthnStatement>
      </saml:Assertion>
    </Security>
    <sb:TargetIdentity> <!-- uses the token previously provided by IDPb -->
      <sec:Token>
        <Assertion ID="fourthassertion">
          <Subject>
        <NameID NameQualifier="IDPb" SPNameQualifier="WSCa" Format="transient">Bob</NameID>
          </Subject>
          <AttributeStatement>
            <Attribute Name="disco-epr">
              <AttributeValue>
                <wsa:EndpointReference>
                  <wsa:Address>DSb.com</wsa:Address>
                  <wsa:Metadata>
                    <ProviderID>DSb</ProviderID>
                    <ServiceType>disco</ServiceType>
                    <SecurityContext>
                     get from earlier message
                    </SecurityContext>
                  </wsa:Metadata>
                </wsa:EndpointReference>
              </AttributeValue>
            </Attribute>
          </AttributeStatement>
    </Assertion>
      </sec:Token>
    </sb:TargetIdentity>
  </Header>
  <Body>
    <disco:Query>
      <disco:RequestedServiceType>
        <ServiceType>geolocation</ServiceType>
      </disco:RequestedServiceType>
    </disco:Query>
  </Body>
</Envelope>
```

2.4.4.1.13. Message 13 - Disco QueryResponse - from DSb to WSCa

```
<Envelope>
  <Header>
    <To>WSCa.com</To>
  </Header>
  <Body>
    <disco:QueryResponse>
      <wsa:EndpointReference>
        <wsa:Address>WSPb.com/disco</wsa:Address>
        <wsa:Metadata>
          <ProviderID>WSPb</ProviderID>
            <ServiceType>geolocation</ServiceType>
              <SecurityContext>
                <!-- no security token -->
             </SecurityContext>
             <sb:TargetIdentity>
               <sec:Token>
                 <Assertion ID="sixthassertion">
                   <Subject>
                     <saml:EncryptedID><xenc:EncryptedData>
                       <saml:NameID Format="persistent" NameQualifier="IDPb" SPNameQualifier="WSPb">Bob</saml:NameI
D>
                        </xenc:EncryptedData>
                        <xenc:EncryptedKey></xenc:EncryptedKey>
                      </saml:EncryptedID>
                   </Subject>
                 </Assertion>
               </sec:Token>
             </sb:TargetIdentity>
          </wsa:Metadata>
      </wsa:EndPointReference>
    </disco:QueryResponse>
  </Body>
</Envelope>
```

2.4.4.1.14. Message 14 - AS Token Request - from WSCa to IDPa

WSCa uses the original SSO token.

```
<Envelope>
  <Header>
    <wsa:To>IDPa.com</wsa:To>
    <Security>
     <Assertion ID="firstassertion">
     <Subject>
  <NameID NameQualifier="IDPa" SPNameQualifier="WSCa" Format="persistent">Alice</NameID>
     </Subject>
    <AuthnStatement>
    </AuthnStatement>
    <AttributeStatement>
      <Attribute Name="disco-epr">
        <AttributeValue>
          <EndpointReference>
            <Address>DSa.com/disco</Address>
            <Metadata>
              <ProviderID>DSa</ProviderID>
              <ServiceType>disco</ServiceType>
              <SecurityContext>
                <Token>

                </Token>
              </SecurityContext>
            </Metadata>
          </EndPointReference>
        </AttributeValue>
      </Attribute>
    </AttributeStatement>
    </Assertion>
    </Security>
  </Header>
  <Body>
    <samlp:AuthnRequest>
      <samlp:Conditions>
        <samlp:AudienceRestriction>
          <samlp:Audience>WSPb.com</saml:Audience>
        </samlp:AudienceRestriction>
      </samlp:Conditions>
      <samlp:ProtocolBinding>
      http://www.w3.org/2005/08/addressing/anonymous
      </samlp:ProtocolBinding>
    </samlp:AuthnRequest>
  </Body>
</Envelope>

```

2.4.4.1.15. Message 15 - AS Token Response - from IDPa to WSCa

```

<Envelope>
  <Header>
    <wsa:To>WSCa.com</wsa:To>
  </Header>
  <Body>
    <samlp:Response>
      <Status>
        <StatusCode Value="Success"/>
      </Status>
      <saml:Assertion ID="seventhassertion">
        <saml:Subject>
          <saml:EncryptedID><xenc:EncryptedData>
            <saml:NameID Format="persistent" NameQualifier="IDPa" SPNameQualifier="WSPb">Alice</saml:
NameID>
          </xenc:EncryptedData>
          <xenc:EncryptedKey></xenc:EncryptedKey>
          </saml:EncryptedID>
        </saml:Subject>
        <saml:AuthnStatement>
        </saml:AuthnStatement>
      </saml:Assertion>
    </samlp:Response>
  </Body>
</Envelope>

```

2.4.4.1.16. Message 16 - geolocation Query - from WSCa to WSPb

```

<Envelope>
  <Header>
    <wsa:To>DSa.com</wsa:To>
    <Security>
      <saml:Assertion ID="seventhassertion">
        <saml:Subject>
          <saml:EncryptedID><xenc:EncryptedData>
            <saml:NameID Format="persistent" NameQualifier="IDPa" SPNameQualifier="WSPb">Alice</saml:NameID>

          </xenc:EncryptedData>
          <xenc:EncryptedKey></xenc:EncryptedKey>
          </saml:EncryptedID>
        </saml:Subject>                                  <!--         /\              -->
        <saml:AuthnStatement>                              <!--         |              -->
        </saml:AuthnStatement>                             <!--         |              -->
      </saml:Assertion>                                  <!-- Alice is invoking, Bob is target -->
    </Security>                                         <!--              |         -->
    <sb:TargetIdentity>                                   <!--              |         -->
      <sec:Token>                                       <!--             \/         -->
         <Assertion ID="sixthassertion">
            <Subject>
              <saml:EncryptedID><xenc:EncryptedData>
                <saml:NameID Format="persistent" NameQualifier="IDPb" SPNameQualifier="WSPb">Bob</saml:
NameID>
                  </xenc:EncryptedData>
                   <xenc:EncryptedKey></xenc:EncryptedKey>
               </saml:EncryptedID>
             </Subject>
          </Assertion>
      </sec:Token>
    </sb:TargetIdentity>
  </Header>
  <Body>
    <geo:Query>
     where is Bob?
    </geo:Query>
  </Body>
</Envelope>

```

WSPb would use Bob as the target identity to determine the relevant geolocation resource, and use Alice as the invocation identity to determine if the request should be authorized.

3. Engineering Requirements Summary

This section summarizes the Liberty ID-WSF 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
- Gathering consent from a Principal
- Anonymous services
- Usage directives

3.2.1. Service Discovery

Requirements of service discovery stipulate that

- Architecture provides a mechanism for 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 Service 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 Service's site, or the Discovery Service using an ECP[SAMLProf2] communications channel to ask the user directly.

3.2.2. Registration of Services

Requirements of service registration stipulate that

- Architecture provides a mechanism for 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. Gathering Consent

Requirements of consent gathering stipulate that

- Mechanism for a relying provider to request that the invoking provider direct a Principal to the relying provider to request the Principal for consent.
- Mechanism for a provider to utilize an ECP[SAMLProf2] communications channel for querying the Principal’s consent and obtaining the Principal’s response.
- Mechanism for a provider to utilize a non-ECP 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 provider for a given set of attributes, when the set of attributes are shared with the provider.
- Mechanism for a relying provider to partially fulfill requests for attributes if consent not given for all attributes.

3.2.4. Anonymous Service

Requirements of anonymous service stipulate that

- Mechanism for a 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).
- Mechanism to prevent correlation of pseudonyms in service tokens with Principal identifiers.

3.2.5. Usage Directives

Requirements of usage directives stipulate that

- Mechanism for a provider to associate intended usage with the requested attributes in an attribute request to a relying provider.
- Mechanism for a provider to associate the agreed upon intended usage directives with the attribute response
- Mechanism for a provider to return a list of acceptable usage directives to a provider, when the intended usage doesn’t match the Principal’s usage directives.
- Guideline for providers (in the usage negotiation scenario) to always reply to an invoking provider’s attribute request with usage directives that are equal to or privacy-stricter than those originally stated in the provider’s attribute request.

4. Security Architecture

Table 1 generally summarizes the security mechanisms incorporated in the Liberty ID-WSF 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-enabled 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
Confidentiality	Yes	Yes
Per-message data integrity	No	Yes
Transaction integrity	Yes	Yes
Data origin authentication	No	Yes
Nonrepudiation	No	Yes

Channel security addresses how communications between providers and user agents are protected. Liberty implementations must use TLS[RFC4346] 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 TLS or SSL3.0. Note: TLS, 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, requesting providers are required to authenticate relying providers using relying providers' server-side certificates. The relying providers have the option to require authentication of the requesting providers using requesting providers' client-side certificates.
- Additionally, each provider is required to configure a list of authorized (other) providers. Thus, any provider-provider pair must be mutually authorized before they will engage in 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 a provider must be presented to a user before the user presents personal authentication data to that provider.

Message security addresses security mechanisms applied to the discrete Liberty ID-WSF protocol messages passed between 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 ID-WSF 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, providers are required to use key pairs that are distinct from the key pairs applied for TLS[RFC4346] or SSL3.0[SSL] channel protection and that are suitable for long-term signatures.

- In the presense of intermediaries, communicating providers must ensure that sensitive information is not disclosed to unauthorized entities. To fulfill this requirement, providers are required the confidentiality mechanisms specified in [wss-sms].

- In transactions between 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 ID-WSF 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:

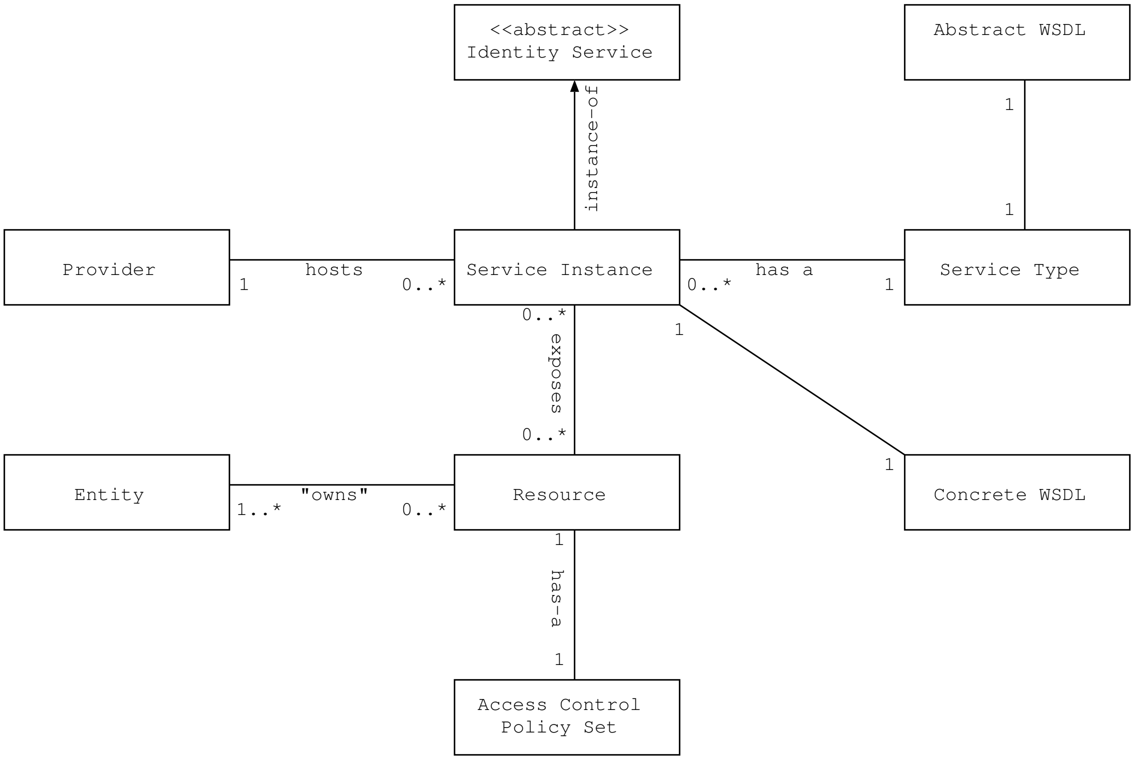


Figure 17. UML Representation of Liberty Conceptual Model

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] 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 ID-WSF Modules

The Liberty Identity Web Services Framework (ID-WSF) consists of multiple specifications in which a set of schemata, protocols and profiles for providing a basic framework of identity services are defined based on open standards including WS-Addressing[WSAv1.0], SAML2.0[SAMLCore2], WS-Security[wss-sms], and SOAP[SOAPv1.1]. On top of the ID-WSF, the Liberty Identity Service Interface Specifications (ID-SIS) are built. The ID-SIS utilize the ID-WSF to provide networked identity services, such as contacts, presence detection or wallet services that depend on networked identity.

Figure 18 below illustrates the Liberty ID-WSF modules and other related specifications.

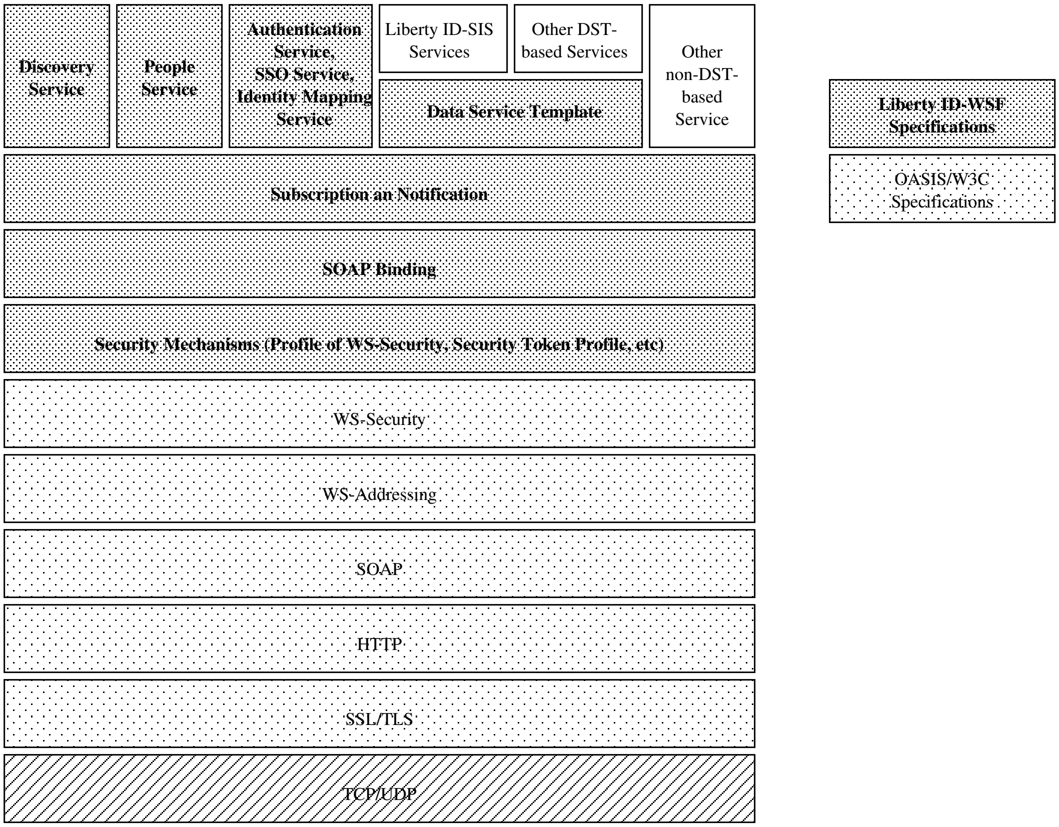

Discovery Service
People Service
Authentication Service, SSO Service, Identity Mapping Service
Liberty ID-SIS Services
Other DST-based Services
Data Service Template
Other non-DST-based Service
Liberty ID-WSF Specifications
OASIS/W3C Specifications
Subscription an Notification
SOAP Binding
Security Mechanisms (Profile of WS-Security, Security Token Profile, etc)
WS-Security
WS-Addressing
SOAP
HTTP
SSL/TLS
TCP/UDP

Figure 18. Liberty ID-WSF Modules

Services built on top of the ID-WSF framework could follow the design patterns provided by the DST. Such the type of services are typically known as "DST-based" services. On the other hand, it is also possible to build services which make use of the privacy and security features provided by the ID-WSF framework, but do not follow the DST design patterns. These services are typically known as "non-DST-based" services.

An example of DST-based service could be the ID-SIS Personal Profile[LibertyIDPP], whilst an example of non-DST-based service could be the ID-SIS CSM (Liberty Messaging Profile). Both of these are Liberty-defined services, although it is also possible that an organization defines its own identity services (of both types), by still making use of the Liberty Identity Web Services Framework (ID-WSF).

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 and/or Discovery Service[LibertyDisco][LibertyAuthn].

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

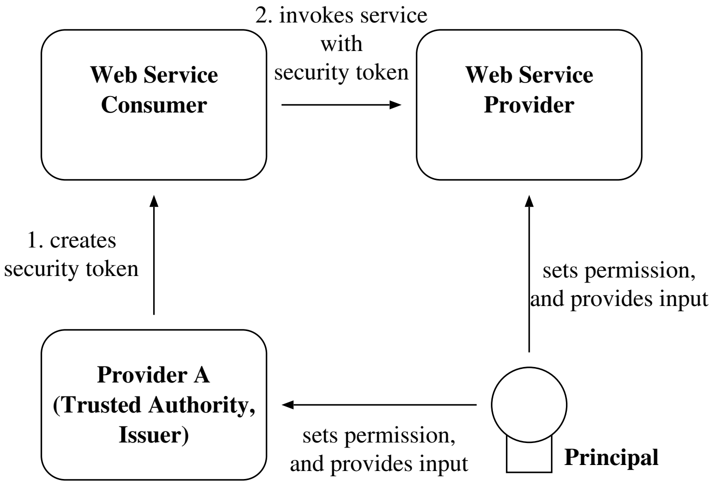

Web Service Consumer
2. invokes service with security token
Web Service Provider
1. creates security token
sets permission, and provides input
Provider A (Trusted Authority, Issuer)
sets permission, and provides input
Principal

Figure 19. 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, 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 may issue targeted assertions[SAMLCore2] as security tokens to those entities. Each of these assertions have a subject of statements in the assertion, and 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.

5.3.2. Identity Token

The Liberty ID-WSF uses an identity token, which provides a structured mechanism to refer to a principal inside the network, together with any attributes that are needed for interaction with such principal and identity-based web services that are provided on her/his behalf.

The identity token can be expressed with multiple ways, such as a SAML assertion[SAMLCore2], WS-Security Binary Security Token[wss-sms], and other XML definitions, and can be conveyed within the SOAP headers[LibertySOAPBinding].

5.3.3. Invocation/Target Identities

When an entity (Web Service Consumer) invokes an identity service, the entity does it on behalf of a particular principal. An invocation identity means an identity of this principal. The invocation identity can explicitly specify with the request SOAP message[LibertySOAPBinding], or implicitly obtained from the security context of the message[LibertySecMech].

When an identity service at an entity (Web Service Provider) is invoked, the entity responds with some resource of a particular principal. A target identity means an identity of this principal. The target identity of the request SOAP message may be same as the invocation identity, or different and explicitly specified with the message[LibertySOAPBinding].

In order to explicitly specify the invocation identity and/or target identity, an entity may use the Identity Token.

5.3.4. 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.

5.3.5. Interaction Service

The Liberty ID-WSF defines an Interaction Service[LibertyInteract]. This service 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 following methods for a Web Service Provider to interact with a user:

1. The Web Service Provider may send a SOAP response with a RedirectRequest element 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 try to discover the Interaction Service of the resource owner to enable the Web Service Provider to send an interaction request 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. When an identity service of one user is invoked by another user (a so called cross-principal interaction), the WSP may need to interact with either or both of the users. The Interaction Service 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.

5.3.6. 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.

5.3.7. Identifier Confidentiality

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. Group and Individual Management

Groups are an integral part in organizing any activities by more than one individual user. The ID-WSF specifies a protocol and schema to manage group of individual identities, so that, once a Liberty ID-WSF compliant user group has been defined, the group can be used in any tools that support the Liberty ID-WSF specifications, which facilitates the seamless integration of the tools. The group information can be manipulated by principals themselves through the providers.

5.3.9. Discovery Service

The Discovery Service is a type of identity service that provides for the discovery of identity services associated with a given identity. In ID-WSF2.0, information of an identity service is represented as an ID-WSF Endpoint Reference that is profiled based on the definition of WS-Addressing Endpoint Reference[LibertyDisco]. An identity will typically have one discovery service on the network that allow other entities to discover its identity services.

The Discovery Service offers two operations, *DiscoveryQuery* and *DiscoveryModify*. In a web services context (browsing, etc.), a Web Services Consumer may need access to a resource exposure associated with an identity (e.g., a profile or location service). The Web Service Consumer may lookup Endpoint References of service instances with a *DiscoveryQuery* request that includes criteria of service desired. The response message contains the relevant Endpoint References of identity services associated with the query, according to the access policies set by the principal/provider. The response may include security tokens and/or identity tokens for service invocation.

The *DiscoveryModify* operation allows a requester to enter and remove Endpoint References of 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 following diagram illustrates the entities involved in possible Discovery Service use cases.

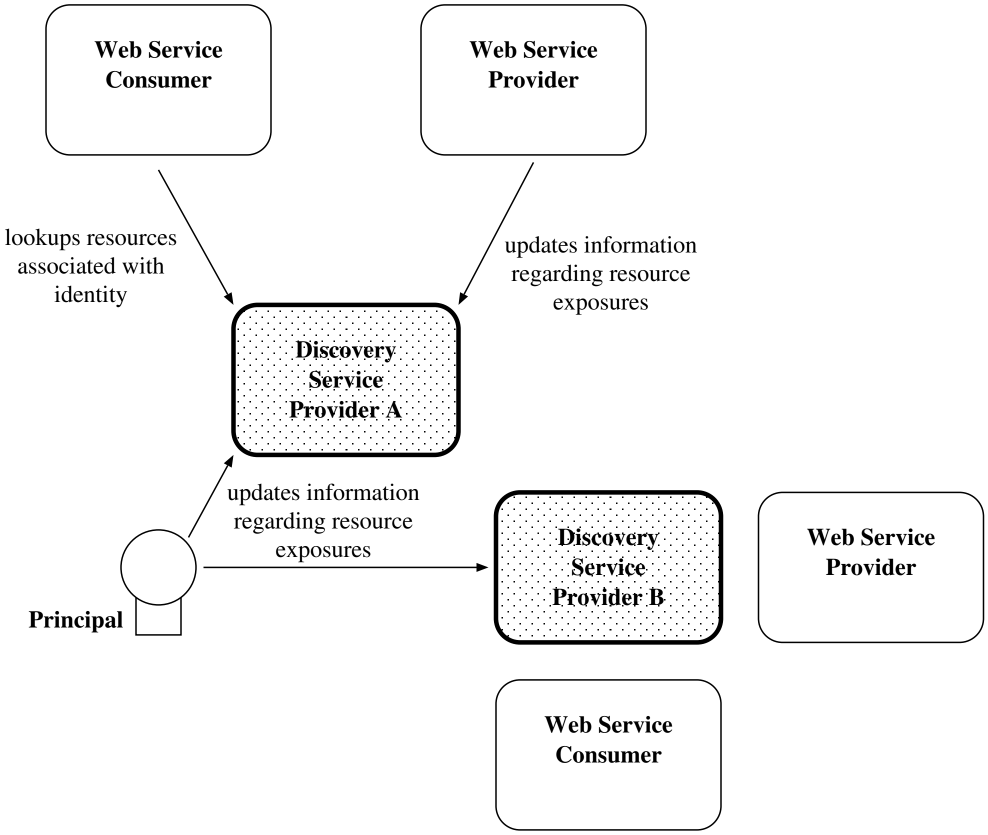

Web Service Consumer
Web Service Provider
lookups resources associated with identity
updates information regarding resource exposures
Discovery Service Provider A
updates information regarding resource exposures
Principal
Discovery Service Provider B
Web Service Provider
Web Service Consumer

Figure 20. Liberty ID-WSF Discovery Service

5.3.10. Liberty-enabled User Agents or Devices

The ID-WSF specifications define a number of protocols that enable any party to act as a Web Service Consumer (WSC), a Web Service Provider (WSP), or both. When user agents or devices wish to act in these roles, some particular issues need to be addressed and hence additional specifications are useful to guarantee interoperability. Moreover, whenever such the user-agent or device acts as WSC or WSP, it typically represents only a very small number of users. Therefore, there are some particular considerations regarding privacy, and the specifications covers those concerns.

The Liberty Alliance specifies the ID-WSF Authentication Service[LibertyAuthn] by which a WSC on a user agent or device may authenticate to an identity provider, and the Reverse HTTP Binding[LibertyPAOS] to enable a user agent or device to act as a WSP. 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 and using PAOS.

References

Informative

[RFC1034] Mockapetris, P., eds. (November 1987). "DOMAIN NAMES - CONCEPTS AND FACILITIES," RFC 1034, Internet Engineering Task Force *http://www.ietf.org/rfc/rfc1034.txt*

[RFC2119] S. Bradner "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*

[RFC4346] Dierks, T., Rescorla, E., eds. (April 2006). "The Transport Layer Security (TLS) Protocol," Version 1.1 RFC 4346, Internet Engineering Task Force *http://www.ietf.org/rfc/rfc4346.txt*

[RFC4422] "Simple Authentication and Security Layer (SASL)," Melnikov, A., Zeilenga, K., eds. (June 2006). RFC 4422, Internet Engineering Task Force *http://www.ietf.org/rfc/rfc4422.txt*

[SAMLCore2] Cantor, Scott, Kemp, John, Philpott, Rob, Maler, Eve, eds. (15 March 2005). "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0," SAML V2.0, OASIS Standard, Organization for the Advancement of Structured Information Standards *http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf*

[SAMLProf2] Hughes, John, Cantor, Scott, Hodges, Jeff, Hirsch, Frederick, Mishra, Prateek, Philpott, Rob, Maler, Eve, eds. (15 March, 2005). "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0," SAML V2.0, OASIS Standard, Organization for the Advancement of Structured Information Standards *http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf*

[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/*

[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*

[WSAv1.0] "Web Services Addressing (WS-Addressing) 1.0," Gudgin, Martin, Hadley, Marc, Rogers, Tony, eds. World Wide Web Consortium W3C Recommendation (9 May 2006). *http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/*

[wss-sms] Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds. (January, 2004). "Web Services Security: SOAP Message Security," OASIS Standard V1.0 [OASIS 200401], 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*

[wss-saml11] Monzillo, Ronald, Kaler, Chris, Nadalin, Anthony, Hallam-Baker, Phillip, eds. (June 28, 2005). Organization for the Advancement of Structured Information Standards *http://www.oasis-open.org/committees/download.php/13405/wss-v1.1-spec-pr-SAMLTokenProfile-01.pdf* "Web Services Security: SAML Token Profile 1.1," OASIS Public Review Draft 01,

[XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C. M., Maler, Eve, Yergeau, Francois, eds. (04 February 2004). "Extensible Markup Language (XML) 1.0 (Third Edition)," Recommendation, World Wide Web Consortium *http://www.w3.org/TR/2004/REC-xml-20040204* [xmlenc-core] Eastlake, Donald, Reagle, Joseph, eds. (10 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, Madsen, Paul, Cantor, Scott, eds. " Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification ," Version v2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyClientProfiles] Aarts, Robert, Kainulainen, Jukka, Kemp, John, eds. Version v2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyDisco] Hodges, Jeff, Cahill, Conor, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyDST] Kellomäki, Sampo, Kainulainen, Jukka, eds. "Liberty ID-WSF Data Services Template," Version 2.1, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyIDEP] Kellomäki, Sampo, Lockhart, Rob, eds. "Liberty ID-SIS Employee Profile Service Specification," Version 1.1, Liberty Alliance Project (29 September, 2005). *http://www.projectliberty.org/specs*

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

[LibertyIDWSFGuide] Weitzel, David, eds. "Liberty ID-WSF Implementation Guide," Version 2.0-02, Liberty Alliance Project (13 January, 2005). *http://www.projectliberty.org/specs*

[LibertyInteract] Aarts, Robert, Madsen, Paul, eds. "Liberty ID-WSF Interaction Service Specification," Version 2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyGlossary] Hodges, Jeff, eds. "Liberty Technical Glossary," Version v2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyPAOS] Aarts, Robert, Kemp, John, eds. "Liberty Reverse HTTP Binding for SOAP Specification," Version 2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyPeopleService] Koga, Yuzo, Madsen, Paul, eds. "Liberty ID-WSF People Service Specification," Version 1.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version 1.2-errata-v3.0, Liberty Alliance Project (14 December 2004). *http://www.projectliberty.org/specs*

[LibertySecMech] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version v2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertySecMech20SAML] Hirsch, Frederick, eds. "ID-WSF 2.0 SecMech SAML Profile," Version v2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, Whitehead, Greg, Madsen, Paul, eds. "Liberty ID-WSF SOAP Binding Specification," Version 2.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*

[LibertySUBS] Kellomäki, Sampo, eds. "Liberty ID-WSF Subscriptions and Notifications," Version 1.0, Liberty Alliance Project (30 July, 2006). *http://www.projectliberty.org/specs*