Liberty Alliance Web Services Framework: A Technical Overview
Version 1.0

14 Feb 2008

Contributors:

Conor Cahill, Intel
Carolina Canales, Ericsson
Hubert A. Le Van Gong, Sun Microsystems
Paul Madsen, NTT
Eve Maler, Sun Microsystems
Greg Whitehead, HP

Abstract:

This overview enumerates the major features of *Liberty Web Services,* a framework for identity-based services that provides added value for identity, security, and privacy above and beyond basic web services, and thereby makes identity data portable across domains.

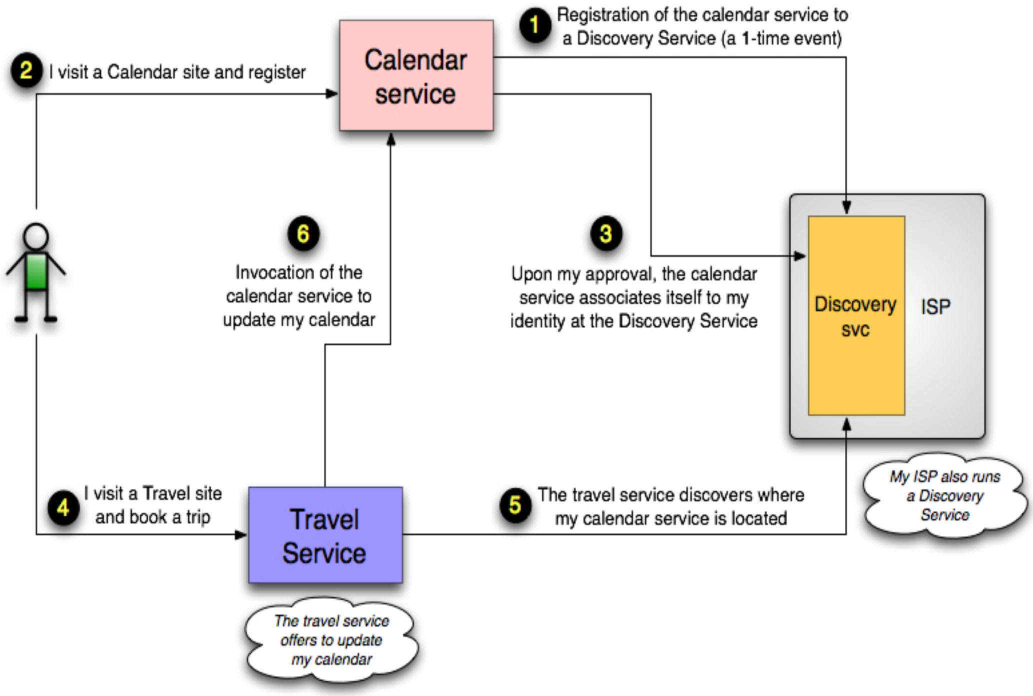
Contents

Introduction

The term **Liberty Web Services** comprises the Identity Web Services Framework (ID-WSF) and the Identity Service Interface Specifications (ID-SIS) that take advantage of that framework. Together, these two pieces enable identity-based services – web services associated with the identity attributes of individual users.

Why are identity-based services valuable? Fundamentally, because they enable a user's identity data to be portable across the many Web applications that, if able to access these attributes, can provide a more customized & meaningful experience to the user, whilst removing from that user the burden of manually repeatedly providing & managing their identity attributes at each.

The following figure shows one scenario where it's useful for a user's calendar data to be accessible to a travel service at which they are booking a business trip. If able to read both the user's work & personal calendars, the travel service could suggest a travel schedule that both got them to their first meeting, and subsequently got them home in time for their daughter's soccer game.


1 Registration of the calendar service to a Discovery Service (a 1-time event)
2 I visit a Calendar site and register
Calendar service
6 Invocation of the calendar service to update my calendar
3 Upon my approval, the calendar service associates itself to my identity at the Discovery Service
Discovery svc
ISP
4 I visit a Travel site and book a trip
Travel Service
5 The travel service discovers where my calendar service is located
My ISP also runs a Discovery Service
The travel service offers to update my calendar

Identity-enabling services

The travel & calendar services might be run entirely separately, allowing the user to choose service providers on whatever basis they like and then have them cooperate (with the user's permission) to produce a seamless result. Critically, the calendar data would not be accessible only by the travel service, but also by any other service provider (e.g., the cable provider wanting to set-up an installation appointment) that the user gave permission to.

To achieve this flexibility the travel service must be able to find the calendar service in the first place. And the whole transaction must be secure and must keep disclosure of personal information to a minimum.

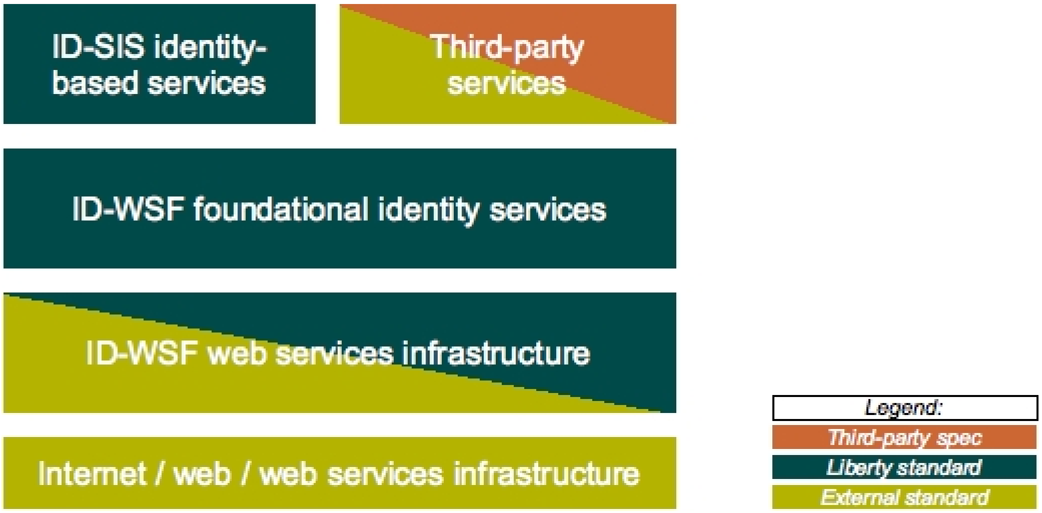
The travel service and the calendar service both need to convey, and be responsive to, the user's identity information and the parameters the user had set for its use. To interact with each other they use web services technology, some of it generic to all types of web services and some of it specially standardized by Liberty Web

Services to handle the identity dimension and its security and privacy requirements. ID-WSF builds on many existing standards for networking and distributed computing, and adds specialized capabilities for handling identity-related information and tasks and for ensuring privacy and security.

With ID-WSF providing the addressing, security & privacy plumbing – different ID-SIS specifications define the specific syntax and semantics for sharing different slices of your identity attributes. For instance, a Calendar SIS specifies how the travel service would query the user's Calendar Service for free blocks, or write an event. Other ID-SIS specification either already exist or can be defined for other aspects of your identity, e.g., The user's personal profile, geolocation, presence, or wallet.

Features and Benefits

The following diagram shows the Liberty Web Services architecture at a high level.


ID-SIS identity-based services
Third-party services
ID-WSF foundational identity services
ID-WSF web services infrastructure
Internet / web / web services infrastructure
Legend:
Third-party spec
Liberty standard
External standard

High-level view of the Liberty Web Services architecture

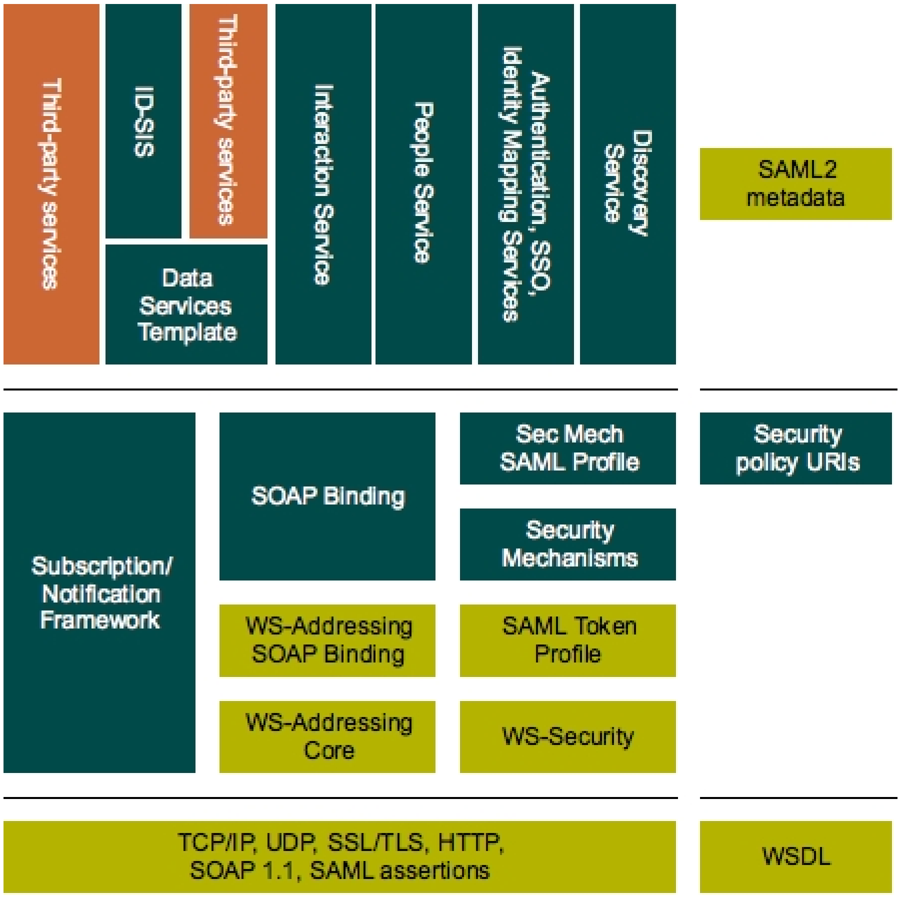
This paper will focus primarily on the framework features provided by ID-WSF for **identity-based** web services. An identity-based service is one that exposes an interface on behalf on a particular user's identity – so in our scenario, the calendar service is identity-based because it is *your* calendar service and not applicable to everyone.

The ID-WSF specifications standardize common functionality that developers can use in incorporating identity-based application features, including:

- **Authentication** – The provider of a service, such as the calendar service, might need to know who is requesting services in order to control access or provide personalized features. Thus, service requesters typically need to be authenticated, and messages sent between the parties need to be verified as coming from the claimed senders. Authentication depends on the notion of identity: Who is accessing my service, and who is this message from?
- **Message protection** – All web service endpoints, including both the providers of identity-based services and the services that use them in turn, need to know that messages they send cannot be intercepted by a malicious entity and then either modified or cached and then replayed.
- **Privacy protection** – Unless special care is taken, identifiers used to label you in web service calls can allow your actions and true identity to be inappropriately correlated and exposed.

- **Service discovery and addressing** – As noted above, your travel service needs a way to learn where your calendar service can be reached in order for them to communicate.
- **Policy** – Service providers may have particular requirements that apply to service requesters. These requirements, which can be quite varied, can be grouped in the general category of policy.
- **Data access and management** – Multiple applications might define similar operations. For example, a “query” message could equally apply to the insurance system (“Who is enrolled in plan XYZ?”) and the corporate address book system (“What is Lois’s phone number?”) within a single organization. ID-WSF offers a standard interface that can then be used and extended by application systems.
- **Social identity** – It is useful to describe and manage your relationships with other people – such as friends, family, and colleagues – through your respective online identities.
- **Transport protocols** – Web services are made available over networks, and services are frequently offered over the Internet using the HTTP protocol and carried in a standard SOAP message. ID-WSF provides a binding of application messages to SOAP that may be carried over HTTP.

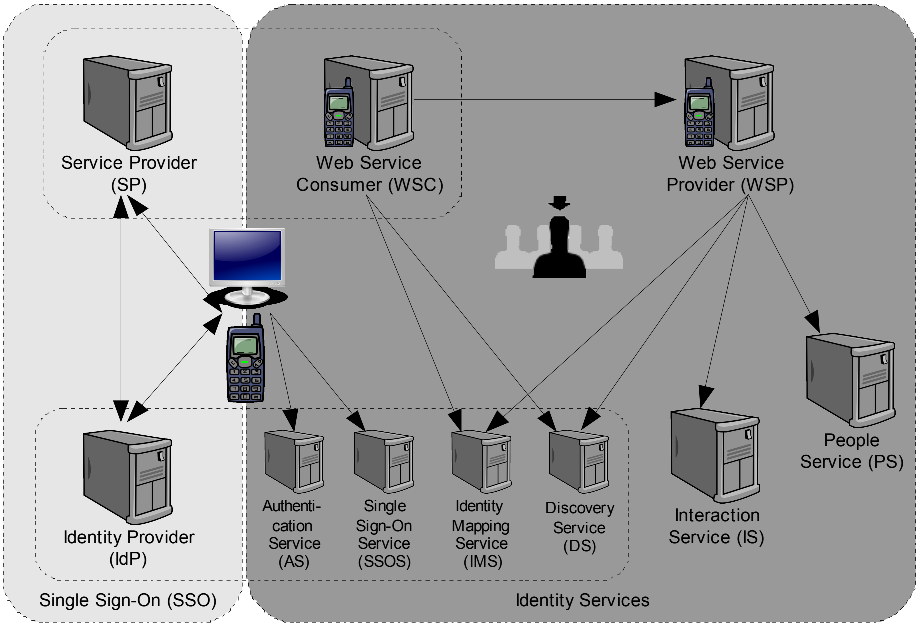
To accomplish this, ID-WSF standardizes web services “plumbing” as well as defining foundational web services that many applications are likely to need. The following diagram shows the specific components defined by ID-WSF as well as some of the other SOA standards it builds on.



Liberty Web Services components in detail

Terms and Concepts

Following are definitions of important terms and concepts used in the definition of ID-WSF, and a diagram showing the basic interaction relationships between them. These terms and their abbreviations are used in the rest of this paper, and relevant **Liberty specifications** are linked from the text. (Please refer to the **Liberty Glossary** for a full list of terms.) In the following diagram, the groupings around multiple services show likely cases where a single software component is responsible for multiple tasks, but such combining is not required.



Liberty architecture components

- **Actors/roles**
  - **User** – A system entity whose identity can be authenticated. It is often synonymous with “natural person”, but other examples include groups of individuals and organizational entities such as corporations.
  - **Identity Provider (IdP)** – A system entity that manages identity information on behalf of users and provides assertions of user authentication to other providers. You authenticate at your IdP and it may store and share information about you, such as your phone number, on your behalf.
  - **Service Provider (SP)** – A Liberty-enabled website or web application providing services or goods to users that is willing to rely on authentication operations or identity attributes performed or stored elsewhere. For this reason, SPs are often also called relying parties (RPs). You might visit a site that offers personalized service on the basis of your login process with your IdP; that site is an SP.
  - **Web Service Consumer (WSC)** – A system entity that is requesting some information or action from a web service. SPs might act as WSCs (“consumers of web services”) when they seek more information about your schedule by directly contacting your calendar web service.
  - **Web Service Provider (WSP)** – A system entity that provides a web service. Your calendar service might act as a WSP (“provider of web services”) to other entities that want to get more information about, or add events to, your schedule.

- **Services**
  - **Discovery Service (DS)** – A special web service that allows WSCs to find a relevant WSP associated with a particular user identity. This involves web service registration as a first step. Normally, as well as facilitating the WSCs to discover *where* on the network the user's different identity attributes are located, the DS enables the WSC to actually send a request for identity to that WSP endpoint by issuing to the WSC an identity token to be included in the request.
  - **Identity Mapping Service (IMS)** – A special web service for mapping user identifiers between different provider namespaces. Identity mapping may be required when providers use pairwise pseudonyms to refer to users, a technique often used for its privacy advantages.
  - **Interaction Service (IS)** – A special web service that facilitates communication between a provider and a user in order to obtain privacy consent or identity attributes. You might set up interaction rules for having your calendar service contact you by SMS in order to gather your consent for calendar-editing by others.
  - **People Service (PS)** – A special web service that helps a user manage and reuse the person-to-person federations that reflect the user's social network. You might use the People Service to set up groups of, and roles for, your friends' identities to help manage the online planning for an upcoming party, regardless of whatever IdP each friend uses.
  - **Authentication Service (AS)** – A special web service that that supports those clients or user-agents with special abilities (beyond a normal browser) for actively participating in Liberty Web Service messaging. The AS allows such Liberty-enabled clients to authenticate to an IdP in order to obtain identity tokens for use in web service calls to WSPs.
  - **Single Sign On Service (SSOS)** – A web service that supports Liberty-enabled clients to achieve SSO with an SP as if it were a plain browser.

Liberty Web Services Functionality Areas

The following sections examine each functionality area in which Liberty ID-WSF provides unique identity, privacy, and security value, and suggest sources for further reading.

Web Service Identity Model

A model is required for carrying the identity of the various parties associated with a transaction within the messages generated to invoke a web service. The parties potentially needing to be identified include:

- Sender – The party sending the message.
- Recipient – The party to whom the message is sent.
- Invoker – The party invoking the request. This can be the same as the sender, or it may be different (such as when an entity is invoking a service on behalf of a user).
- Target Identity – The party whose resource is being accessed by the recipient. This may be the invoker's resource, or a third party's resource, or the recipient's resource. For example, when you query your own mail from an email provider, it is the invoker's (your) resource being accessed. When you query the calendar of a colleague from a calendar provider, it is your colleague's resource being accessed.

Each of these parties should have a known method for identification on each request and the request should permit all the parties to be different entities.

ID-WSF V2.0 defines the following components in support:

- A profile of WS-Security and SAML to carry the Sender, Recipient, and Invoking identities, as defined in the **Security Mechanisms** specification, the **Security Mechanisms SAML Profile**, and the **Discovery Service** specification (Section 2.3.3.5).
- A new header called TargetIdentity, as defined in the **SOAP Binding** specification (Section 5.10).
- Mapping of the identity invocation context during discovery for subsequent service invocation, as defined in the **Discovery Service** specification (Section 2.3.3.5).

Identity Mapping Mechanisms

In the context of discovering and invoking the identity services of a particular user, providers must be able to refer to that user to distinguish it from others who may have their identity stored at the same provider. If privacy-respecting pairwise pseudonyms (wherein each two providers use a unique identifier to refer to a particular user) are used to inhibit correlation of that user's activities at various providers, then a provider will often need to know "what identifier should be used to refer to User X at Provider Y". Identity mapping mechanisms allow providers to pose this query to a service that can supply the answer.

ID-WSF V2.0 defines the following component in support:

- The Identity Mapping Request/Response protocol that allows a requester in possession of one or more identity tokens (whether simply a name identifier or a full security token) to translate, update, or refresh them. Conceptually, the mapping protocol is a translation or exchange of one or more inputs for corresponding outputs. Each input consists of an identity token and a policy specifying the characteristics of the identity token to be returned. The security token of the invoking identity can also be referenced as the input token. The output is the requested identity token, the exact form of which may be up to the mapping service to establish. This protocol is defined in the **Authentication, Single Sign-On, and Identity Mapping Services** specification (Section 7).

Design Patterns for Data Operations

Different services can be created that rely on the privacy and security features provided by the Liberty Web Services framework. Apart from making use of ID-WSF, some of these services share certain common characteristics in their service logic. For example, some of them need mechanisms for data creation, modification, and retrieval (the "CRUD operations"). In these cases it is very convenient to be able to share common design patterns for these operations.

ID-WSF V2.0 defines the following component in support:

- The Data Services Template, an optional component that provides a standard template to build data services accessible by means of a CRUD interface. For instance, DST includes guidelines, common XML attributes, and data types that could be reused by multiple data services. Some Liberty-defined services, such as the Personal Profile Service, use the DST, and others do not. It is defined in the **Data Services Template** specification.

Identity Provider Services

One of the basic features a web service framework must provide is the ability to establish a reasonable level of confidence about the identity of the parties involved in a transaction. Achieving such confidence relies on the ability for the peers (web services in this case) to authenticate to each other. ID-WSF defines services that enable this authentication.

ID-WSF V2.0 defines the following components in support:

- **Authentication Service (AS)** – Provides a simple yet flexible method for a WSC (as well as a Liberty-enabled user agent or device) to authenticate to an identity provider and obtain security tokens it will present to a WSP. Flexibility is guaranteed by the use of the SASL protocol, which enables run-time selection of the actual authentication mechanism that is to be used. The Authentication Service is defined in the **Authentication, Single Sign-On, and Identity Mapping Services** specification (Sections 4 and 5).
- **Single Sign-On Service (SSOS)** – Addresses more sophisticated usage patterns like authenticating a WSC so it can access a SAML V2.0-enabled service provider or cross-user invocations where a user's WSC needs to invoke a WSP owned by a different user. The SSO Service is defined in the **Authentication, Single Sign-On, and Identity Mapping Services** specification (Section 6).
- **Identity Mapping Service (IMS)** – A component needed to propagate the level of confidence about an identity to other parties in a cross-identity transaction.

Social Identity

Social identity refers to a set of mechanisms deployed by providers that allows users to share (that is, grant access rights to) their online resources and services with friends and colleagues. The key requirement for such capability is to allow a user to, in a sense, federate other people's identifiers with that user's own identity. A service called the People Service functions as a repository for a user's person-to-person federations – the set of these federations reflecting the user's social network(s).

Such federations make explicit the connections that exist between users - this essential in the deployment of secure, scalable, and privacy-aware social networks based services such as photo sharing.

ID-WSF V2.0 defines the following components in support:

- A data model to represent the relationships a user stores at his or her People Service. The data model also describes how the user has organized these relationships into groups. This is defined in the **People Service** specification (Section 2).
- An invitation model by which the necessary federations will be established before resources can be shared, as defined in Section 4.
- A SOAP interface for providers to query and manipulate information about a person's relationships, as defined in Section 3.

Service Discovery

Before a service can be invoked by other web services, it needs to be made discoverable through a registration step with the user's discovery service. A discovery service matches available registered services to a lookup request coming from a WSC. This request describes the desired services both in terms of functional criteria as well as ownership (that is, only services associated with a particular identity are considered in the discovery process).

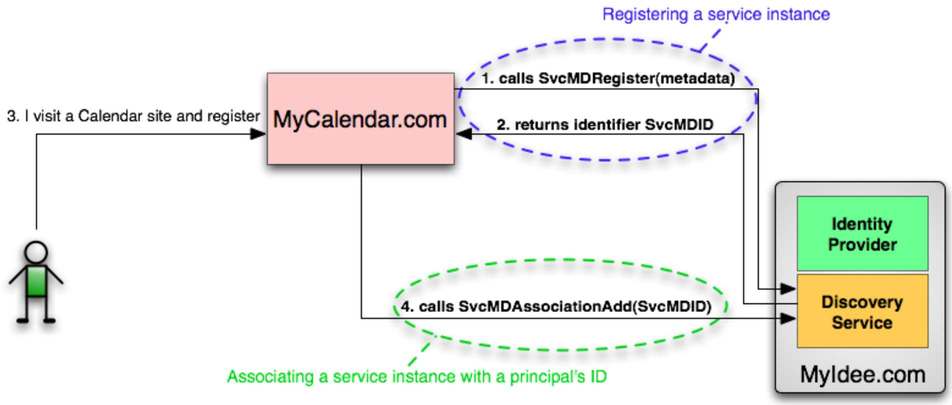
Discovery is fundamental to any web service framework, but has special requirements when that framework handles identity as well.

ID-WSF V2.0 defines the following components in support:

- A data model to represent the available web services associated with an identity (the ID-WSF endpoint reference (EPR) – as discussed in the section called Bootstrapping – and service metadata, potentially along with a security token), as defined in the **Discovery Service** specification (Sections 2.3 and 2.4).

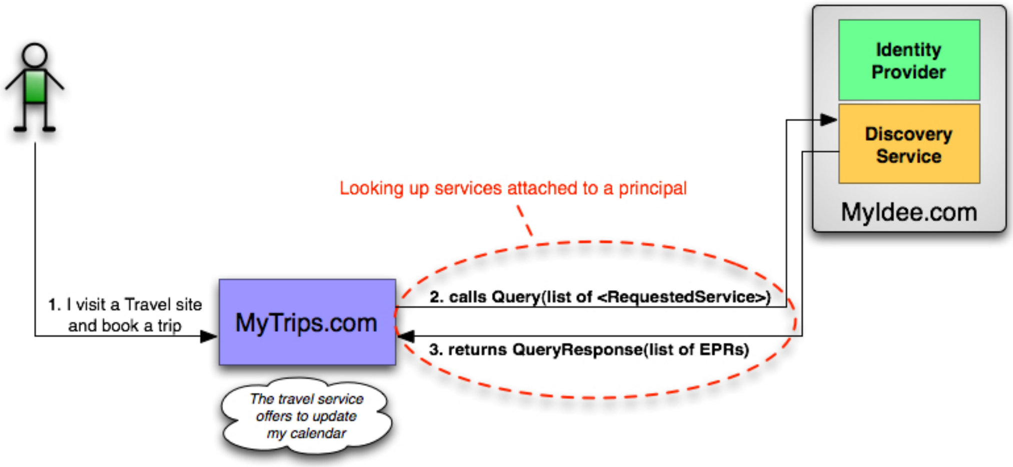
- An interface to allow WSCs to retrieve from a Discovery Service a list of web services associated with an identity based on various criteria, along with other information needed for invoking the service (such as policy data and security tokens), as defined in Section 3.3.
- An interface to allow WSPs to register and manage their resources (web services) at the Discovery Service, known as service metadata maintenance, as defined in Section 3.4 through 3.12.
- A model to bootstrap the service discovery process by including required information about the Discovery Service itself (the EPR) in security/identity tokens, as defined in Section 4.

The following diagram illustrates the one-time process in which MyCalendar.com registers that it will act as a particular user's calendar service.


Registering a service instance
1. calls SvcMDRegister(metadata)
2. returns identifier SvcMDID
3. I visit a Calendar site and register
MyCalendar.com
Identity Provider
Discovery Service
4. calls SvcMDAssociationAdd(SvcMDID)
Associating a service instance with a principal's ID
MyIdee.com

Discovery service registration and association

The following diagram illustrates the discovery process when a travel service called MyTrips.com wants to locate a particular user's calendar service in order to query or write to that calendars. MyTrips.com queries the Discovery Service hosted the relevant identity provider MyIdee.com for the location of all calendar services registered to the user in question.


Identity Provider
Discovery Service
MyIdee.com
Looking up services attached to a principal
1. I visit a Travel site and book a trip
MyTrips.com
2. calls Query(list of <RequestedService>)
3. returns QueryResponse(list of EPRs)
The travel service offers to update my calendar

Service discovery prior to invocation

Service Invocation and Message Construction

SOAP provides the definition of the XML-based information that can be used for exchanging structured and typed information between peers in a decentralized, distributed environment. SOAP is silent on the semantics of any application-specific data it conveys, as it is on issues such as the routing of SOAP messages, reliable data transfer, firewall traversal, and so on. However, SOAP provides the framework by which application-specific information may be conveyed in an extensible manner (the property known as “SOAP Extensibility”).

ID-WSF V2.0 adds an identity, security, and privacy layer around the application-specific data found in messages. It allows for mapping onto various transport or transfer protocols, SOAP being the typically chosen one. This implies that messages addressed to any Liberty infrastructural service or to any other identity-enabled service that makes use of ID-WSF will be typically conveyed in the body of a SOAP message.

ID-WSF V2.0 defines the following components in support:

- Mechanisms based on **Web Services Addressing 1.0 – Core** and **Web Services Addressing 1.0 - SOAP Binding** to provide transport-neutral mechanisms to address an identity-based web service, as defined in the **SOAP Binding** specification (Section 5).
- SOAP headers together with processing rules needed for the invocation of identity-based web services. The features provided by these headers could be classified as follows:

    1. Privacy-related (Consent, Usage Directives): Used by the requester in order to indicate the privacy context in which the service invocation takes place, or the subsequent use and distribution of the obtained invocation. These features are defined in Section 6.

    2. Processing or security context (Processing Context, Credential Context, Endpoint Update, Timeout, Application EPR, and Sender): Used by the parties to transfer extra information needed for the communication to take place (including token renewal or redirection to a different endpoint). These features are defined in Section 6.

    3. User interaction: ability to interact with the user.

    4. Identity-related information (Target Identity).

Design Pattern for Subscription and Notification

Subscription and notification refer to a messaging pattern whereby a requester can, rather than continuously requesting the state of some resource, instead subscribe to be notified if and when the resource's state does change (with defined criteria). Subscription and notification can consequently enable significant efficiencies, for example in the number of messages in a protocol flow. This pattern is intended for use with resources whose state changes only infrequently.

ID-WSF V2.0 defines the following components in support:

- A data model defining subscription and notification objects. A subscription object represents an agreement between a WSC and a WSP that the WSC be sent a notification object should the specified criteria be met for some resource. This is defined in the **Subscriptions and Notifications** specification (Sections 4 and 5).
- Rules by which subscription objects can be included in messages creating or querying some resource, as defined in Section 3.
- A request/response protocol by which a WSC can ask that a subscription object be created for a given resource, optionally specifying the desired criteria and location for a notification object to be sent, as defined in Section 4.
- A request/response protocol by which a WSP can send a notification object corresponding to a given subscription to the appropriate endpoint, as defined in Section 5.1.

Security Policy

Security policy refers to mechanisms by which actors in a web service transaction can specify their capabilities and/or expectations with respect to the security characteristics of web service messages. For instance, an actor might indicate that it expects SOAP messages to be digitally signed in addition to any transport-layer security mechanisms.

In environments of heterogeneous security technologies and models, actors must be able to determine the intersection (if any) of acceptable security processing for any particular transaction. Security policy mechanisms support this determination in such environments.

ID-WSF V2.0 defines the following components in support:

- Defined URI identifiers for particular security mechanisms, which are combinations of message- and transport-level technologies (for example, XML Signature and SSL) that together achieve an understood level of security. This is defined in the **Security Mechanisms** specification (Table 6).
- The ability for WSPs, when registering a service endpoint at a Discovery Service, to specify the particular Security Mechanisms URIs supported by the endpoint. In this context, the WSP is indicating its security policy requirements of any message sent to that service endpoint. This is defined in the **Discovery Service** specification (Sections 2.3.3.3 and 3.7).
- The ability for WSCs, when querying for a service endpoint from a Discovery Service, to specify the particular Security Mechanisms URIs as search parameters. In this context, the WSC is indicating its security policy capabilities with respect to messages it would send to any discovered endpoint. This is defined in the **Discovery Service** specification (Section 3.3.2).

Privacy

In the identity context, privacy refers to the following two major sets of mechanisms and controls.

First are the mechanisms for meeting the desires of a user and the expectations and requirements of providers as to how the user's identity information is shared, used, and processed. Privacy policy refers to mechanisms by which users and providers engaged in a web service-based identity transaction can specify their capabilities or expectations with respect to the privacy treatment of any identity attributes transmitted.

Typically, a requester will specify its commitments with respect to how it will treat any identity attributes, and a provider will specify its requirements (along with those of the user) should it release the attributes. Both actors require mechanisms and syntax by which such policy can be expressed in the web service request and response messages.

Second are the mechanisms for inhibiting inappropriate correlation (as would be possible were all providers to share a single global identifier for a user) of a user's actions at various providers. Pseudonyms that are unique to each pairing of providers offer a privacy-preserving mechanism to prevent such correlation by ensuring that providers cannot, based solely on their identifiers for a given user, collude.

ID-WSF V2.0 defines the following components in support:

- UsageDirective header block – Used by web service requesters and providers to indicate their privacy policy commitments and requirements to the other. The UsageDirective header block serves as a general-purpose container into which particular privacy policy expressions can be carried; it defines no policy syntax itself. The UsageDirective header is defined in the **SOAP Binding** specification (Section 6.6).
- Consent header block – Used by requesters and providers to assert that the user in question has consented to the identity interaction represented by the web service message, for example, a request for some identity attribute from a provider. The Consent header is defined in Section 6.2.
- Interaction Service – Provides a mechanism for interacting with a user, which can be used to collect or refine a user's own privacy policies during a transaction. See the section called User Interaction for a discussion of the Interaction Service.
- Pseudonyms and Identity Mapping Service – Provide a mechanism by which one provider, interested in engaging in a web service transaction with another, can request a pseudonymous identifier for a user targeted at that particular provider. The Identity Mapping Service provider returns an appropriate pseudonymous identifier that will be recognized by the eventual recipient but does so in such a way that the actual value for the identifier remains confidential, and so can't be inappropriately used as a correlation handle. See the section called Identity Mapping Mechanisms for a discussion of the Identity Mapping Service.

Bootstrapping

A common problem faced by a web services framework is the need to be able to jump from a single sign-on environment on a service provider into a web services environment to invoke one or more web services on behalf of the identity authenticated to the service provider via the SSO operations.

This process of crossing from SSO into web services is referred to as the “bootstrap” process, and it requires a particular set of information to be included in the tokens generated within the SSO environment such that the service provider has sufficient information to act as a WSC in the web services environment.

ID-WSF V2.0 defines the following component in support:

- A profiled version of a WS-Addressing EPR (known as an ID-WSF EPR) that points to the Discovery Service. The profile documents the particular pattern of how the ID-WSF EPR is carried within a SAML V2.0 assertion to achieve bootstrapping. This profile is defined in the **Discovery Service** specification.

User Interaction

User interaction mechanisms are systems (such as messages and prompts) by which providers engaged in an identity transaction concerning particular users can reach out and contact those users if they are not currently actively visiting that provider.

Such an interaction may be necessary in order to, for instance, obtain consent for the release of some piece of that user's identity information or to collect a piece of identity information requested by another provider.

ID-WSF V2.0 defines the following components in support:

- A SOAP Header block that allows a WSC to indicate its preferences and capabilities for interactions with requesting users, and possibly a service endpoint to which interaction requests can be sent, as defined in the **SOAP Binding** specification (Section 6.8).
- A SOAP fault and redirect request protocol by which the WSP can request that the user agent be directed to a particular address for interaction, as defined in the **SOAP Binding** specification (Section 7).
- A SOAP service to which providers can send messages requesting that the user be interacted with, as defined in the **Interaction Service** specification (Section 3).

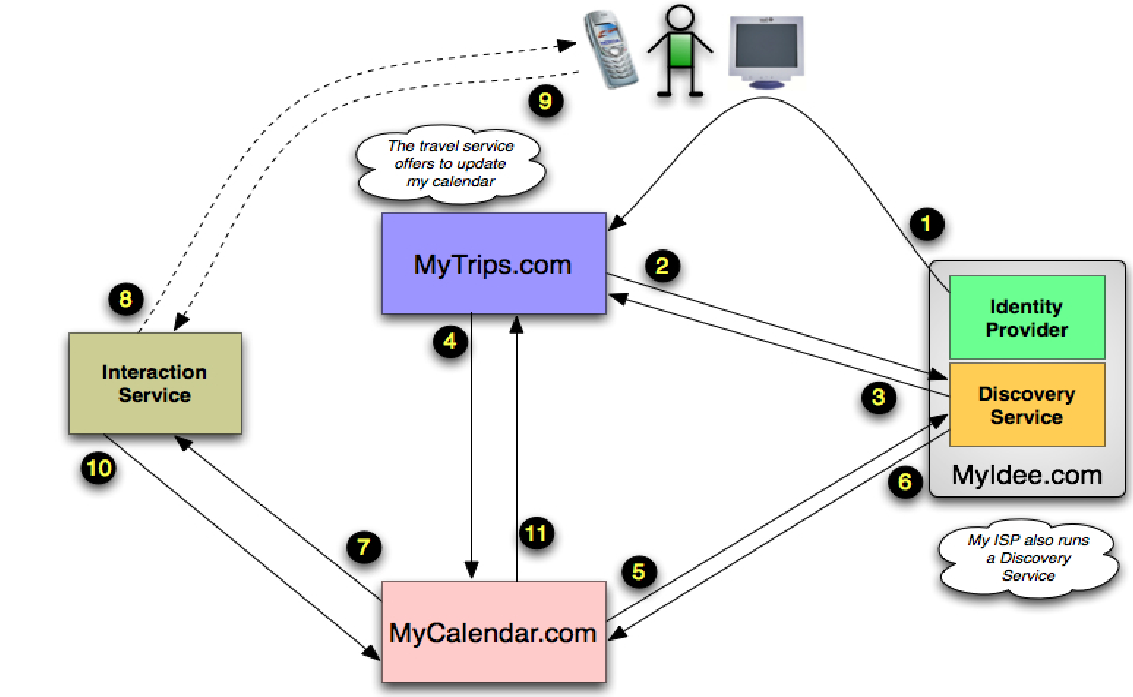
Example Use Case and Message Flow

This section examines a complete scenario – describing the full set of messages that flow between the various system entities to achieve an identity- and privacy-enabled result.

In this use case, a Travel Service provider MyTrips.com, after allowing a user to single sign-on from an Identity Provider MyIdee.com, locates and queries information about the user from a web service that holds the user's calendar information. The calendar service MyCalendar.com is, in turn, able to locate and utilize an interactive means of contacting the user in order to obtain the user's approval to release the requested calendar information.

The message flow for this use case involves bootstrapping from the user's single sign-on interaction (Step's 1, 2, & 3) into the identity services world (other Steps) – using the bootstrap mechanism in order to locate and gain access to the Discovery Service, and then using the Discovery Service to locate and gain access to MyCalendar and the Interaction Service in turn.

Note that user interaction is typically not required on a per-request basis, as the necessary approvals may be remembered from previous interactions, and/or the user's privacy policies may be managed out of band. Also, note that the role of a dynamic Discovery Service is key for providing the “nested” ability of one identity service to locate and access another.



1. The user SSOs into MyTrips.com from MyIdee.com, resulting in a SAML2 assertion being sent to MyTrips.com, that includes the bootstrap ID-WSF endpoint reference (EPR) for the user's Discovery Service (DS).

   An ID-WSF EPR extends a basic WS-Addressing EPR to allow inline specification of endpoint security policy and inclusion of required security and identity tokens. In this flow, it is assumed that the bootstrap DS EPR includes any needed security and identity tokens (perhaps referring to the enveloping Web SSO token).

   Note that if the user had authenticated directly to the MyTrips.com SP, it would have had to find another means of locating the DS.

2. MyTrips.com needs user identity information, and so calls the DS to discover the user's Calendar Service (which in this case is MyCalendar.com)

3. The DS responds with an ID-WSF EPR for MyCalendar.com.

   Again, in this flow, it is assumed that the MyCalendar.com EPR specifies the endpoint security policy and includes any needed security and identity tokens. The security token could be, for example, a SAML token with the user as the subject identity and MyTrips.com as the subject confirmation identity. This token could also include a bootstrap Discovery Service EPR, which MyCalendar.com could use to invoke other services.

4. MyTrips.com asks MyCalendar.com for the user's open calendar slots around the desired time of travel.

5. MyCalendar.com determines that it needs consent from the user before releasing the event information, and so discovers the user's Interaction Service (IS).

Note that this step assumes that the bootstrap DS EPR was included in the security token presented to MyCalendar.com or was cached and already available to it, and that the user has an IS. An alternative is to return a SOAP Fault indicating that the user needs to be redirected to the CS to provide consent.

6. The DS responds with an ID-WSF EPR for the IS.

   Again, in this flow, it is assumed that the IS EPR specifies the endpoint security policy and includes any needed security and identity tokens. The security token could be, for example, a SAML token with the user as the subject identity and the CS as the subject confirmation identity.

7. MyCalendar.com invokes the IS in order to secure consent from the user so that MyTrips can receive the requested information.
8. The IS sends an SMS to the user's cell phone (an interaction that is out-of-band with respect to Liberty Web Services; any other communications channel could be used instead).
9. The user responds with a confirmation (again, an interaction that is out-of-band with respect to Liberty Web Services).
10. The IS responds to MyCalendar.com in the affirmative.
11. Confident that the user has given their release, MyCalendar.com responds to MyTrips with the requested calendar information.

Summary

An identity-based service is a web service associated with a particular user, i.e., a web service at which a user's calendar information can be accessed. Identity-based services require functionality beyond that necessary for basic web services not associated with a given user – particularly in the areas of identity, security, and privacy This paper has outlined how *Liberty Web Services* framework meets these functional requirements.

Liberty ID-WSF specifications define the addressing, security & privacy plumbing – and different Liberty ID-SIS specifications define the specific syntax and semantics for sharing different slices of identity attributes. Together, ID-WSF & ID-SIS make identity data portable in a secure & privacy-respecting manner.