Liberty Security & Privacy Implementation Guidelines

Version 1.0-05

14 April 2003

Editors:

Susan Landau, Sun Microsystems
John Kemp, IEEE-ISTO

Contributors:

Carolina Canales Valenzuela, Ericsson
Gary Ellison, Sun Microsystems
Jeff Hodges, Sun Microsystems
Sampo Kellomäki, Symlabs
John Linn, RSA

Abstract:

This document provides an overview of the security and privacy issues in ID-WSF technology and briefly explains potential security and privacy ramifications of the technology used in ID-WSF. This is not a normative document. The intended audience for this document is implementors of the Liberty Identity Web Services Framework (ID-WSF). It is assumed that the audience is familiar with the Liberty Identity Federation Framework

Note: This document is a preliminary draft of the security and privacy guidelines for the Liberty ID-WSF architecture. This document may not be complete, and further substantive revisions to the text can be expected.

Notice

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

Revision History:

Version #	Date	Editor	Scope of changes
1.0–01		John Kemp	Original framework
1.0-02	31-Mar-2003	Susan Landau	Reorganized document
1.0-03	7-Apr-2003	Susan Landau	Added What is a Security Policy, added (from Liberty Architecture Overview) to General Security and Privacy Mechanisms section, filled in authentication and authorization sections, edited Discovery Service and Interaction Service sections.
1.0-04	14-Apr-2003	Susam Landau	Comments received from Technology Group and PPEG. In response, removed policy discussion in Security Functions section, fixed Glossary. Added Establishing Trust ,amplified Security Functions Required for Privacy, combined ID-Personal Profile Service and Data Service Template into Data Services, rewrote Usage Directives section and fixed diagram(JH), cleaned up text and formatting.
1.0-05	14-Apr-2003	Tom Wason	Adjusted legal notice, added abstract.

1 Introduction

1.1 Audience

The intended audience for this document is implementors and deployers of the Liberty Identity Web Services Framework (ID-WSF) and presents guidance for service interface specifications for identity services. It is assumed that the audience is familiar with the Liberty Identity Federation Framework [LibertyArchOverview].

1.2 Goals

This document provides an overview of the security and privacy issues in ID-WSF technology and briefly explains potential security and privacy ramifications of the technology used in ID-WSF. This is not a normative document.

1.3 Document Structure

The Liberty Alliance Project is an undertaking by a group of companies to develop a set of open, technical specifications for web services. The first step, now completed, is the Liberty Identity Federation Framework, a set of specifications enabling single sign-on using federated network identity. The Liberty Identity Federation Framework provides specifications for associating, connecting, and binding multiple accounts for a given Principal at various Liberty Alliance sites within a Circle of Trust. This document is concerned with Identity Services, which is an abstract notion of a web service that acts upon some resource to obtain information about an identity, update information about an identity, or perform some action for the benefit of an identity. The Liberty Identity Web Services Framework (ID-WSF) is a set of specifications for creating, using, and updating various aspects of identities.

Security and privacy protection in ID-WSF are enforced through several mechanisms:

1. Via general facilities provided at the application layers, and
2. Within each Liberty component, there are application-specific facilities for securing and privacy-protecting data and services.

This document first discusses general security requirements and the issues of authentication and authorization as well as a brief discussion of threat models. Then the document introduces the architectural elements comprising the ID-WSF and discusses the various mechanisms that enhance security and privacy in these components of the ID-WSF: Discovery Service, Interaction Service, and data services. Some more general security issues, including privacy, are then discussed. At a later date we expect to model specific deployment scenarios showing the security and privacy mechanisms available in “real-world” scenarios.

1.4 Definitions

Definitions for Liberty-specific terms can be found in the Liberty Glossary [LibertyGlossary]. Security is highly dependent on precise implementation of protocols and for this reason, definitions of a number of the terms used are presented.

Attribute: a distinct characteristic of a Principal. A Principal's characteristics are said to describe the Principal.

Attribute Broker: entity that serves as a relay for receiving attribute requests and sending attribute responses on behalf of multiple Attribute Providers.

Attribute Provider: entity that provides attributes to a requester.

Federate: to link or bind two entities together.

Identity: the essence of an entity and often described by its characteristics.

Identity Provider: A Liberty-enabled entity that creates, maintains, and manages identity information for Principals and provides Principal authentication to other service providers within an authentication domain.

Identity Service: a particular type of web service that acts upon some resource to retrieve information about an identity or group of identities (e.g., calendars in order to schedule a meeting), update information about an identity or group of identities, or perform some action for an identity or group of identities.

Invocation Identity: subject of an assertion, party involving a service.

Non-Transitive Proxy Capability: the ability to act for another entity based on Trusted Authority policy. The capability is not transferable.

Policy Decision Point: system entity that evaluates decision requests in light of applicable policy and renders an authorization decision.

Policy Enforcement Point: system entity that performs access control by making decision requests and enforcing authorization decisions.

Principal: a Principal is an entity that can acquire a federated identity, that is capable of making decisions, and to which authenticated actions are done on its behalf. Examples of principals include an individual user, a group of individuals, a corporation, other legal entities, or a component of the Liberty architecture.

Proxy: An entity authorized to act for another.

Recipient: an entity that receives a message which is the ultimate processor of the message.

Sender: initial SOAP sender. A sender is a proxy when its identity differs from the invocation identity.

Service: invocation responder, providing a service. Ultimate message processor.

Service instance: an instantiation of a particular type of identity service

Service Provider: an entity that provides services and/or goods to Principals.

Trusted Authority: a Trusted Third Party that issues and vouches for SAML assertions.

Web Service: a service that uses Internet protocols to provide a service designed to be used by programs.

Web Service Consumer (WSC): an entity that uses a web service to access data.

Web Service Provider (WSP): an entity that provides data via a web service.

1.5 What is a Security Policy

Security needs a clear set of rules enables the system's administrators to understand what is protected and what is not. A security policy is a set of rules and practices specifying the who, what, when, why, where, and how of access to system resources by system entities (often, but not always, involving or acting on behalf of people). Significant portions of security policies are implemented via security services, which are processing or communication services that are provided by a system to give a special type of protection to system resources [OASISGlossary].

In the Liberty context of web services in a distributed environment, two particular aspects of a security policy are worthy of special note: authentication and authorization. Authentication is the process of confirming a system's entity's asserted identity with a specified, or understood, level of confidence [OASISGlossary]. There are variety of methods for doing this. Techniques for authenticating people include account number and PIN and username and password (really two versions of the same technique), which are typically considered a weak form of authentication; challenge-response is a stronger form. The TLS/SSL “handshake protocol” is a cryptographic protocol mechanism for authenticating processing entities; it establishes server-side (and client-side) authentication at the beginning of a TLS/SSL session. In the distributed architecture of Liberty Identity Web Services, authentication is extremely important and we discuss various aspects below.

Authorization is the process of determining which types of activities an entity can perform. If access is to be limited, authorization only makes sense in the context of authenticating an entity. Depending upon the level of authentication, the entity will have authorization to perform different types of activities [OASISGlossary].

2 General Security and Privacy Mechanisms for Liberty Identity Web Services Framework

This section provides discussion and guidance related to the distributed security and privacy mechanisms in the Liberty ID-WSF protocols. It emphasizes inter-component aspects as embodied in the ID-WSF architecture; aspects oriented to individual Liberty services will be considered in the next section.

Security in the Liberty Framework is layered. Liberty protocols are themselves built with extensive security mechanisms. Furthermore they are built upon various Internet protocols that have embedded security mechanisms [LibertyInteractionService].

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

Table 1: Liberty security mechanisms

Security Mechanism	Channel Security	Message Security (for Requests, Assertions)
Confidentiality	Required	Optional
Per-message data integrity	Required	Required
Transaction integrity	—	Required
Peer-entity authentication	Identity provider — Required Service provider — Optional	—
Data origin authentication	—	Required
Nonrepudiation	—	Required

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

Critical points of channel security include the following:

- In terms of authentication, service providers are required to authenticate identity providers using identity provider server-side certificates. Identity providers have the option to require authentication of service providers using service provider client-side certificates.
- The authenticated identity of an identity provider must be presented to a user before the user presents personal authentication data to that identity provider.

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

Critical points of message security include the following:

- Liberty protocol messages and some of their components are generally required to be digitally signed and verified. Signing and verifying messages provide data integrity, data origin authentication, and a basis for nonrepudiation. Therefore, identity providers and service providers are required to use key pairs that are distinct from the key pairs applied for TLS and SSL channel protection and that are suitable for long-term signatures.
- In transactions between service providers and identity providers, requests are required to be protected against replay, and received responses are required to be checked for correct correspondence with

issued requests. Time-based assurance of freshness may be employed. These techniques provide transaction integrity.

To federate, providers are required to establish bilateral agreements on selecting certificate authorities, obtaining X.509 credentials, establishing and managing trusted public keys, and managing life cycles of corresponding credentials.

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

2.1 Establishing Trust

Web services is about sharing information. Liberty specifications aim for enabling a networked world in which individuals and businesses can engage in virtually any transaction without compromising security or privacy of vital identity information. In order for interoperating Liberty components to do so, they must establish a "trust relationship." In the Liberty Identity Federation Framework, federations, established through business/legal agreements combined with an out-of-band exchange of shared secret keys or public-key certificates, exemplified a strong and direct trust model. This model of trust does not scale well and is too limited to accomplish web services. A more flexible way of establishing trust is needed. This is done through Brokered Trust and Community Trust models. We present a brief discussion here; more detail on establishing trust among Liberty components can be found in the Liberty Trust Models Guidelines document [LibertyTrustModels].

2.2 Authentication

Authentication is the act of confirming a system entity's asserted identity with a specified, or understood, level of confidence. The simplest case occurs when a Principal presents credentials to an Identity Provider. The Identity Provider can decide whether or not to authenticate the Principal based on the credentials provided by the Principal and the Identity Provider's authentication policy. A more complex scenario occurs when a Service Provider receives an authentication from an Identity Provider. The Service Provider can decide whether to accept the Principal's authentication context as sufficient based on the Service Provider's authentication policy (note that the Service Provider will need to authenticate the Identity Provider). Both the authentication context for the Principal and the Identity Provider's authentication of itself to the Service Provider are policy enforcement points (PEPs), e.g., gateways to the resource being managed.

Brokered Trust models come into play when federation and/or authentication transactions span multiple administrative domains. They require the availability of appropriate intermediaries in order to construct a path to federate a user's relationship and/or to authenticate a particular session. For example, Brokered Trust may be applicable when a Service Provider associated with Identity Provider A receives an assertion to be processed from Identity Provider B, with which it has no prior relationship. The assertion is a piece of data produced by a SAML authority about an authentication of a subject, attributed information about a subject, or authorization permissions applying to the subject about a particular resource [SAMLGlossary].The Service Provider must decide whether to trust Identity Provider B's assertion (which, for simplicity, we will assume is an authentication assertion, though in fact it could be any of assertions mentioned). Trust is determined through a combination of business trust, based on business agreements, and authentication trust, based on cryptographic assertion infrastructure.

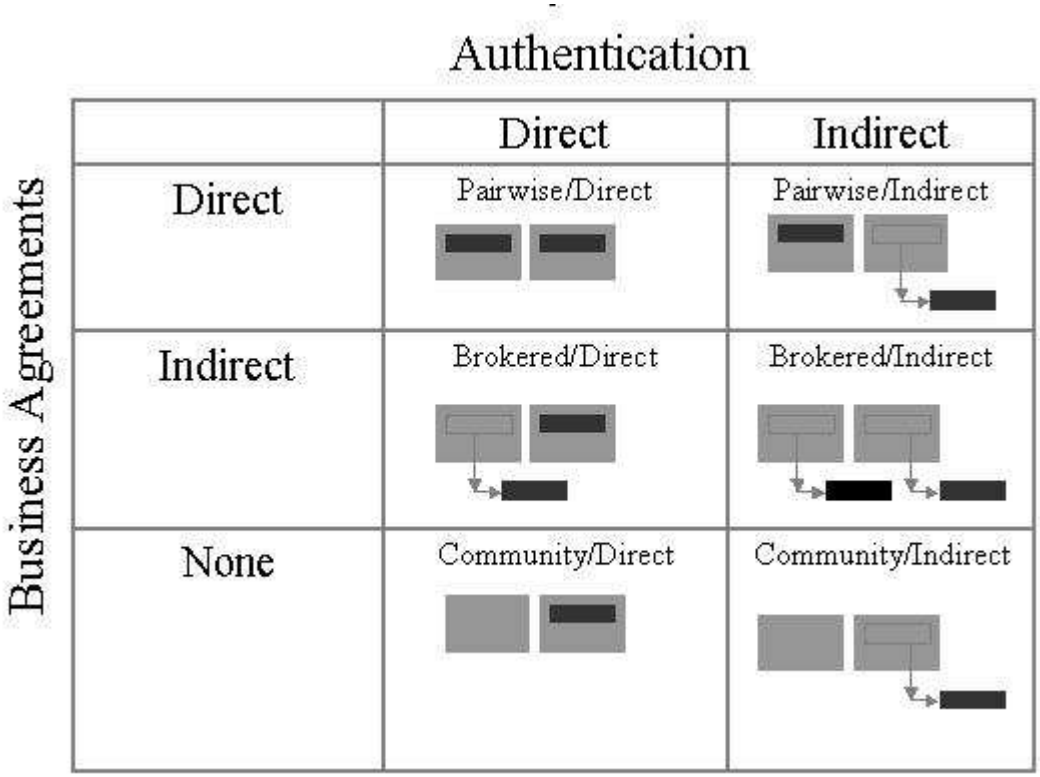


Figure 1: Trust Model Taxonomy

In Brokered Trust, there is no direct business agreement. In the case we are considering, Identity Provider B has no direct relationship with the Service Provider. There are two possible cases for Brokered Trust: either there is a business agreement between the Service Provider and an intermediary and the intermediary has direct business relationship with Identity Provider B (this can used transitively), or there is not, but the business relationship between the Service Provider and the intermediary allows the intermediary to act as an agent for the Service Provider. The latter case allows business trust to be established dynamically.

Community Trust models use membership in a community defined by a cryptographic infrastructure as a basis for enabling federation and/or authentication. Public Key Infrastructure, Kerberos realms and inter-realm relationships, and PGP webs of trust are all examples of such infrastructures.

It is also possible to develop business relationships without authentication infrastructures. That approach is out of scope in the context of Liberty.

In the physical world, authentication is established through physical tokens. Authentication in the on-line world is typically based on cryptographic mechanisms. As observed earlier, there are different mechanisms depending on whether one is authenticating Principals (human) or processing entities. In the Liberty context, Principals are authenticated by Identity Providers, which determine the means by which they choose to authenticate the Principal. Although the technique an Identity Provider uses for authenticating a Principal is not within the scope of Liberty specifications, Liberty does specify the transport mechanism for these interchanges. Communications from Principals to Liberty-enabled sites must be integrity protected and confidentiality must be ensured. Liberty-enabled sites must use SSL 3.0 or TLS 1.0 for conducting communications with Principals. Note that the security of the SSL or TLS session depends on the chosen ciphersuite; Liberty specifications recommend the use of at least a 112-bit symmetric key. More details may be found in the normative [LibertyID-WSFSecurity].

Liberty specifications require authentication of processing entities. In the absence of active intermediaries in the message path, transport-layer protection mechanisms suffice to ensure the confidentiality and integrity of the message exchange. Authentication of both sender and recipient is required SSL 3.0 or TLS 1.0 and X.509 client and server-side certificates (see [PKIX-WG] for information on the X.509 Public-Key Infrastructure) can be used for this. If

active intermediaries are present, the sender must use message layer authentication. Therefore the sender must authenticate the messaging layer either by using X.509 Certificate Message Layer Sender Authentication or SAML Assertion Layer Sender Authentication mechanism; normative specifications can be found in [LibertyID-WSFSecurity]. In both cases, the recipient receives an assertion binding the sender to the key, and the sender provides proof of possession of the key by signing elements of the message.

Under certain conditions (see Discovery Services, below), two separate identities must be authenticated for a given request: the *invocation identity* and the *sender identity*. Typically the identity of the message sender is to be treated as the invocation identity. In this instance, there is no need for a distinction between the invocation identity and the sender identity. The candidate mechanism to convey identity information is client-side X.509 certificates based authentication over a SSL/TLS connection. Generally this protocol framework may rely upon the authentication mechanism of the underlying transfer or transport protocol binding to convery the sender's identity.

For scenarios where the sender's messages are passing through one or more intermediaries, the sender must explicitly convey its identity to the recipient. This is done by using a Web Security security token; see [LibertyID-WSFSecurity].

For the cryptographic mechanisms described above to work properly, private and shared secret keys must be secured. Loss of key---private or shared secret---completely compromises the security systems based on cryptographic mechanisms. This means that sensitive processing functions must be performed within systems designed to satisfy appropriate assurance requirements and systems should be operated and managed in accordance with appropriate security practices.

Public keys need not be protected against disclosure but must be protected for integrity purposes. Effective use of a public key for signature validation requires that the key be associated with a trust anchor acceptable to the relying party. This can either be through direct knowledge of the key by the relying party or by successful validation of a correct---and timely---certification path. Secure operation of a signature-based architecture like Liberty ID-WSF requires that a relying party's set of trust anchors be correctly managed. Validation steps (including, e.g., revocation checking) should be correctly performed before accepting a signature as representing its presumed signer. Careless use of the public-key infrastructure invalidates the protections provided by the Liberty Framework security protocol specifications.

In addition to secure processing at the levels of cryptographic operations and trust validation, secure operation of the ID-WSF protocols also requires that the processing rules defined in their specifications be fully and correctly implemented. Security protocols are often fragile and a minor change to a protocol can completely invalidate its security mechanisms. Liberty ID-WSF implementers should ensure that the protocol processing modules they employ are fully conformant with the Liberty protocol specifications.

2.3 Authorization

Access to the attribute data managed by Liberty ID-WSF-based deployments is mediated according to two classes of authorization policies: policies established by Liberty processing components and policies established by the individual principals with whom attribute data is associated. Before access to protected data is granted, constraints of ALL applicable policies must be satisfied. Liberty implementers must ensure that suitable policy management interfaces are available to administrators and to principals; the type and scope of interfaces provided may vary in different operational environments.

Authorization depends on the combination of a securely managed authentication system and securely managed data describing authorization policy (e.g.,. in the form of Access Control Lists (ACLs)) for protected resources [LibertyID-WSFSecurity].

Identity services are invoked by requesters. The invocation may be direct or it may be conducted with the assistance of an active intermediary. To invoke an identity service a requester must interact with a specific service instance that exposes some resource.

Given the above, we strongly believe that access control policies must be enforced by identity services. The authorization decision to access an identity service offering a specific resource may be made locally (at the entity

hosting the resource) or remotely. Regardless of whether the policy decision point (PDP) is distributed or not, a policy enforcement point (PEP) must always be directly implemented by the entity hosting or exposing the resource.

In most cases, the service requester directly interacts with the identity service, thus the identity service may implement both the PEP and the PDP. Under these circumstances the authorization decision, at a minimum, should be based on the authenticated identity of the service requester and the resource for which access is being requested.

[1]However, an identity service may rely upon a trusted third party (TTP) to make coarse policy decision. It is also likely that the TTP will act a a Policy Information Point (PIP) such that it can convey information regarding the resource and the policy it maintains. This scenario might be deployed in the event that the Principal is unable to actively authenticate to the identity service. One such scenario is where a TTP provides a bridge function to introduce new participants to the identity service. The result of any such policy decision made by the TTP must be presented to the entity hosting the identity service. Of course this does not preclude the identity service from making additional policy decisions based on other criteria.

The Liberty ID-WSF specification enables a Trusted Authority (TA) to act as a Policy Information Point (PIP) to obtain assertions demonstrating the session context of the interacting Principal. The Liberty ID-WSF also incorporates a Resource Interaction Service (ROS) facility, which enables providers to engage in direct interactions with the principals responsible for requested attributes. Authorization policies should be specifiable in a manner that enables these facilities to be invoked as needed, either at the level of confirming that a user is currently logged on to a Liberty Identity Provider or, more strongly, obtaining explicit approval for access to designated attributes.

The Liberty ID-WSF also enables a TA to act as a PIP to determine what resources may be accessed by the request sender and authentication is needed (on its behalf or for another system entity). It is anticipated that Liberty Discovery Services will operate in this role. Under policy control, invocation identities and named resources contained within these assertions may be represented in a form that cannot be interpreted by the ID-WSF intermediaries; use of this facility limits the degree of trust that principals must place in intermediaries for privacy purposes. The TA provides a facility to register the authorization data requirements for particular identity service instances and the resources they offer. Identity services relying on authorization decision assertions provided by the TA must maintain accurate policy data at the TA, and must trust the TA to correctly reflect that data in the assertions it generates.

The importance of the distinction between invocation and sender identity lies in the service's access control policies whereby the service's decision to deny or grant access may be based on either or both identities. The degenerate case occurs when the invocation identity is identical to the sender identity, in which no distinction need be made.

Note that a browser-based user agent interacting with some service provider does not necessarily imply that the service provider will use the user identity as the invocation identity.

2.4 Threats

The Liberty Alliance specifications seek to enable individuals and businesses to engage in virtually any transaction without compromising the privacy and security of vital identity information. Liberty specifications have been designed to protect against:

- Eavesdropping: Information within the message is viewable by an unauthorized users.
- Replay attack: A message is sent in which includes portions of another message in order to gain access to otherwise unauthorized information.
- Message Insertion/Deletion/Modification: The message is altered by inserting/deleting/modifying information and is mistaken by the receiver as having been sent as is by the original sender.
- Man-in-the-Middle attack: An attack in which an intermediary poses as the other party to the real sender and receiver in order to fool both parties [WS-ISecurity].

These attacks are prevented through a combination of the authentication and authorization requirements discussed above; see also [ID-WSFSecurity]. There are also a number of security vulnerabilities and risks that are out of scope for the Liberty specifications. These include denial-of-service attacks at the network level, host penetration/access,

331 traffic analysis, timing attacks (computing the amount of time a computation takes in order to determine other
332 information).

333

3 Security Functions Required for Privacy

Considering privacy purely from a security vantage point, privacy is a security policy applied to an individual, or, in the Liberty context, a Principal. Of course, privacy is much broader than such a definition. One can easily find databases with excellent security policies which are nonetheless privacy invasive (any secured database that contains nonrelevant personal information, e.g., a research medical database that contains the patient's social security number). However, in the context of the Liberty Identity Web Services Framework, where the issue is designing technical specifications for the secure sharing of Principal attribute data, the model that "privacy is security policy applied to a Principal" is a useful model for privacy protections.

The most relevant security functions needed for privacy are:

- Authentication of the Principal and/or any other entities that could perform policy management tasks (policy definition, modification, etc.).
- Authentication of attribute requesters.
- Policy integrity in transit (at the moment of policy definition, modification or any other kind of policy management operation).
- Policy integrity in storage
- Attribute confidentiality in transit (response from the Attribute Provider to the Service Provider).
- Attribute confidentiality in storage
- Attribute integrity in storage and transit
- Policy management authorization
- Audit capability: maintenance of transaction records in secure storage.
- Avoiding collusion between Identity Provider and Service Provider.
- Data aggregation.

The ID-WSF architecture enables a broker-type functionality whereby a WSC may make a request to a WSP who acts as a broker and makes subsequent requests (as a WSC) to other WSPs who have the required information. The Broker subsequently aggregates the data and responds to the originating WSC in the chain. There are several points of concern about privacy regarding this data as it passes such a chain of providers. The first are whether each WSC in the chain is properly authenticated and is authorized to receive the data in question and whether the transmission of Principal data is done in a way that ensures confidentiality and integrity. As described previously, the Liberty specifications require authentication of entities and proper authorization for transmittal of Principal data. The specifications also require mechanisms to ensure confidentiality and integrity. The next issue concerns the WSP who is acting as an Attribute Broker for the Principal. This is both an in-scope and out-of-band issue for Liberty specifications. The in-scope aspect is provided through usage directives, which enable providers to designate permitted uses of data and enable requesters to designate the use they wish to make of requested data. However, the usage directives cannot assure that the WSC will follow that profile. That issue is out of scope for Liberty specifications and is rather a case for the legal system.

That is a general issue about the security functions described above. The Liberty specifications provide various security mechanisms that help protect the Principal's privacy. Table 1 presents an overview of these mechanisms, which are described in much greater detail in the normative document [ID-WSFSecurity]. Liberty specifications require authentication for anyone acting for a Principal and for any entity requesting or consuming attribute information. For security and privacy, the Liberty specifications encrypt Principal data during message transport. Through the appropriate use of nonces, the specifications protect the Principal against unauthorized parties accessing data about the Principal through a replay attack. Through the use of pseudonymity, the specifications protect against collusion between Identity Providers and Service Providers who may hold the Principal's attribute information. These requirements provide a high degree of security and thus privacy for the data transmission. But the Liberty specifications must be used in conjunction with business and legal agreements between entities. It is expected that entities will adhere to their business and legal agreements, including stated privacy policies. But if entities do not, the

issue is out of scope for Liberty, which is, after all, a set of technical specifications for data exchange. Instead such a situation is appropriately handled by the judicial system.

3.1 ID-WSF Architectural Elements

An Identity Service is a particular type of web service that acts upon some resource to either retrieve information about an identity, update information about an identity, or perform some action for the benefit of an identity. A resource is either data related to some identity or a service acting for the benefit of some identity [LibertyID-WSFSecurity].

In the Liberty Identity Web Services Framework (ID-WSF), we assume that the Principal has already registered with an Identity Provider. The Principal may have done so through a commercial portal or she may have been automatically enrolled through her employer. Nothing precludes the Principal from having several Identity Providers. Principals, in fact, typically have many identities: as an employee, as a <spouse, parent, child>, as a member of several distinct civic groups (e.g., membership in a political party, membership in service organizations), etc. It is expected that many people will have more than one Identity Provider, perhaps one through work and several personal ones. In an ID-WSF, the Principal uses services: ordering and arranging for a gift to be shipped (the shipping address already being known to the shipping company), scheduling a meeting with several colleagues, arranging a trip, authorizing an insurance company to view patient treatment information.

ID-WSF consists of a number of distinct elements (see [LibertyISFPrimer]) that together form a framework of web services. There are several types of system entities: Web Service Providers (WSP), which host web services such as a profile service (see below), Web Service Consumers (WSC), which, with appropriate authentication and authorization, can access a user's web services by communicating with the WSP's endpoint (the targetted entity that contains the resource), and Discovery Service (DS), which is a web service typically hosted by an Indentity Provider that enables a WSC to determine which WSP provides the needed service. Each of these elements has its own facilities for security and privacy protection.

The ID-WSF SOAP Binding provides a SOAP-based invocation framework for identity services. It defines SOAP Header blocks and processing rules enabling the invocation of identity services via SOAP requests and responses. Additionally, a usage directive container is defined for those implementations that wish to use an existing rights language to specify the required service and data usage policies. The Discovery Service defines a core identity service that enables various entities (e.g., Service Providers) to dynamically discover a user's registered identity service. The Discovery Service also functions as as security token service, issuing security tokens to the requester that the requester will use in the request to the discovered identity service.

3.2 Discovery Service

The first step in Liberty Identity Web Services is to determine where the resources needed are located: which Provider holds the Principal's credit-card information, which server stores the Principal's calendar, which Provider stores the Principal's travel preferences. The Discovery Service presents an interface for consumers of identity services to locate resource offerings. Entities place resource offerings---information describing the location of different types of information about Principals---in a discovery resource. Thus the Discovery Service is essentially a web service interface for “discovery resources,” each of which can be viewed as a registry of resource offerings.

For example, a Principal wants to make airline reservations. Through a *DiscoveryLookup* operation a WSC can determine with which resource (WSP) a Principal stores her travel preferences (e.g., client sends a *DiscoveryLookupRequest(resource(identity, airlinePrefs))* to a DS. The DS responds with a WSP that handles that resource---airlinePrefs--- for that identity). Or if a Principal wants to make a purchase over the Internet, a WSC would send a DiscoveryLookupRequest(identity, WalletServ) to discover which WSP holds the Principal's wallet data. The *DisoveryUpdate* operation enables maintenance of a discovery resource, accommodating inserts and removals of resource offerings.

The DiscoveryLookup operation enables a requester to obtain an enumeration of ResourceOffering elements. Because a provider hosting a Discovery Service may also be fulfilling other roles for an identity (such as a Policy Decision Point or an Authentication Authority), the DiscoveryLookup operation can also function as a security token service, providing the requester with an efficient means of obtaining security tokens that may be necessary to invoke service

instances returned in the DisoveryLookupResponse. A set of security tokens can be provided within the SecurityTokens element in the response. As the Discovery Service provider may have to perform significant work for each result in the response, especially if security tokens will be generated, responders should construct a DiscoveryLookupRespone to be as qualified as possible. The Discovery Service provider should provide security tokens if it knows that these tokens will be necessary and it is able to provide them based on the security token included in the request.

Previously we mentioned the notion of conveying both a *sender identity* and an *invocation identity*. In doing so the framework accommodates a restricted (non-transitive) proxy 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). 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 intermediate system may generate and distribute a token authorizing the intermediary to act on behalf of the subject to the recipient. This protocol framework can only convey authoritative information regarding the identities communicated to other system entities. Even with the involvement of an authority playing the roles of Policy Administration Point and Policy Decision Point, the recipient must still implement some degree of policy decisions and enforcement [LibertyID-WSFSecurity].

There is a second distinct type of proxy: a *proxy resource*. If there is a proxy resource registered for a service type, the Discovery Service must follow these rules in determining the subset result related to that service type:

- If the identity of the requester is not the identity of the provider of the proxy resource offering, the result set for that service type must contain only the proxy resource offering as well as all other resource offerings for the which the requester is the provider.
- If the identity of the requester is the provider of the proxy resource offering, the result set must contain all resource offerings for the specified service type, including the proxy resource offering. Additionally, the directives for all instances of the requested service type must be aggregated when formulating the security tokens, as the proxying agent will need these token to fulfill the request.

To protect users' privacy, the Liberty architecture uses pseudonymous identity. WSCs and WSPs do not have a common name for a user and the only system entity that can map between the disparate namespaces is the user's Identity Provider. For this reason it is optimal if the Discovery Service is hosted by the Indentity Provider, which provides this namespace translation.

The Identity Provider provides name translation with the Principal's name in the WSC-IDP and the Identity Provider returns a name in the WSP-IDP namespace, blinding the name through the use of encryption. This is what provides the pseudonymous identity. To prevent linkable identity information over time between the WSC and the WSP, the name's encrypted value is different each time. Furthermore, to prevent linking the Principal's actions through the long-term use of a translated name, it is best if the name translation assertion be time bound.

3.3 Interaction Service

An identity service may sometimes need to interact with the owner of the resource that it is exposing, for example, to collect attribute values or to obtain permission to share the data with a Web Service Consumer. The Interaction Service is an ID-WSF specification that defines schemas and profiles that enable a Web Service Provider to interact with the owner of the resource that is exposed by that WSP. The Interaction Service allows its clients (services) to indirectly query a resource owner for consent, authorization decisions, etc. An IS provider accepts requests to present some information and questions to a Principal. The IS provider is responsible for “rendering” a “form” to the Principal; to do so, the IS must know about the Principal's capabilities and preferences. The IS returns the answer of the principal in a response that contains the parameters and values of the request.

The Interaction Service is effectively acting to its client WSCs as a proxy for the Principal. It is therefore important that the IS can be trusted by those clients. This is especially the case when such a WSC is itself a WSP that needs to obtain consent or permissions. There is no general possibility for an IS to prove on-line that it did indeed obtain the response from the Principal. The IS can---and should---of course authenticate the Principal and then save the proof of authentication, such as an assertion. But there is little point in forwarding such assertion to the WSC as proof, as ID-FF authentication assertion will contain the NameIdentifier of the Principal as it is known to the IS, not to the WSC

(for pseudonymity purposes, this name is encrypted). An IS that is closely associated with an Identity Provider (i.e., has the same providerID as the Identity Provider) could issue an assertion that states the the Principal as known to the WSC was present.

It does not suffice to know that a Principal was present at the IS. There remains the possibility that the IS modified the Principal's response. One solution to this threat is to have the Principal sign the response with a private key for which the invoking WSC has a public key associated with that Principal. The WSC can include key in the interaction request. The WSC should have the Principal's permission to share the key with the IS.

For the Redirect Profile these considerations do not apply, as parties that need to interact with a resource owner do so themselves. It is again important that the WSP authenticates the Principal. Although the information in these redirects is not particularly valuable, it is nonetheless recommended that secure connections be used so that intruders cannot replay a request. This risk is reduced if WSPs require that all ID-WSF requests are signed and/or authenticate WSCs. All participants should protect themselves against reply attacks by checking for recently-used messageIDs, etc.

The Principal has a risk that an IS, or for that matter, any WSP, may misrepresent him. That is, of course, an out-of-band issue. Nonetheless, we observe that IS providers should make efforts to induce trust in the Principal by offering transaction logs, by employing sufficiently strong authentication methods, etc. [LibertyInteractionService].

3.4 Data Services

Web services provide data services to computers and networked devices. In the current context, a data service is a web service that supports the storage and update of specific data attributes regarding a Principal. The Liberty Personal Profile Service and the Liberty Data Service Template are two examples of data services; the Personal Profile Service provides profile information regarding a Principal while the Data Services Template provides protocols for querying and modifying data attributes while implementing a data service using ID-WSF. Although the Personal Profile Service is actually part of the Liberty Identity Services Interface Specification, for completeness, we include it here.

3.5 Personal Profile Service

The Liberty Personal Profile Service, ID Personal Profile, is a service that handles identity information for a Principal; the service provides identity attribute data structured in containers (containers are sets of related attributes, e.g., street address, town, city, postal zip, country may form the address container). Typically a Principal will have several identities that need not be linked. All of a Principal's ID Personal Profiles may, however, be registered with a Discovery Service.

The attribute data may be carefully validated (more likely if the information is from an HR database) but it may not. A Principal may list different values for an attribute in different ID Personal Profile services (e.g., different choice of personal title in work and personal ID Personal Profile services, different photo for personal and work ID Personal Profile services). Because there may be multiple hosts for a single Principal's ID Personal Profiles, data synchronization between these various hosts is infeasible. In any case, such synchronization is quite possibly not desired. It is neither expected nor necessary that all attributes of an ID Personal Profile service be populated.

There are no Liberty ID-WSF requirements on how data actually resides at an ID Personal Profile service. Thus data may be stored at the service, it may be computed on the fly, it may be kept on a backend system. Although the Liberty ID Personal Profile specification is defined in terms of XML, that does not mean that data at the ID Personal Profile service must actually be kept in XML format.

The ID Personal Profiles are queried by or updated by clients, typically a Service Provider, acting on behalf of a Principal. An ID Personal Profile is not required to report the same results to two instances of the same query unless the query is being made by the same client and no update (a modify or out-of-band update) of the data has occurred in the interim [LibertyIDPersonalProfile].

3.6 Data Service Template

The ID-WSF Data Service Template provides protocols for querying and modifying data attributes of a Principal when implementing a data service on a Liberty ID-WSF. The query must identity the Principal and the data being queried. The Data Service Template specification defines two protocols: one for querying data and one for modifying data.

The request message must state the resource it wishes to access (e.g., the Personal Profile of a certain Principal) as well as more specified information about exactly what data it wishes to access (e.g., telephone number). Both data requests and data modifys support multiple operations in a single message, but all the operations must be of the same type, e.g., all requests or all modifications. The response message includes a status element that indicates whether the processing of the request succeeded [LibertyDataService].

4 Overall Security and Privacy Guidance

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. The key objectives of the Alliance are to enable consumers to protect the privacy and security of their network identity information, to enable businesses to maintain and manage their customer relationships without third-party participation, to provide a single sign-on standard that includes decentralized authentication and authorization from multiple providers, and to create a network identity infrastructure that supports all current and emerging network technologies[LibertyArchOverview]. Below we describe some non-service-specific security and privacy guidance.

4.1.1 Individual Service Set-up/Deployment

In various jurisdictions, Service Providers may need to let the Principal exercise the first right of control over the information she chooses to share with the Attribute Provider. In this case, the Principal has to actively define the attributes that the Attribute Provider can host, and in particular, the Attribute Provider needs an explicit consent from the Principal for service creation. The Principal may select the set of attributes that each Attribute Provider holds so that certain attributes are only hosted at Attribute Providers controlled by the Principal (or which the Principal especially trusts). Because of this, a given instance of a (e.g., ID-Personal Profile Service) may not offer the complete set of user attributes.

The ID-WSF Discovery Service already supports this functionality by means of the “options” feature.

4.1.2 Identity Services Operational Policy Considerations

These are the aspects that should be considered at an operational level:

4.1.2.1 1. Policy definition

Implementations of Identity Services should provide mechanisms to enable deployments to customize the policies which control the distribution of a principals attributes. Policies cover the circumstances/conditions under which the Principal attributes are provided to a requesting Service Provider/WSC.

Although it might seem that Principals should define the policies for their personally-identifiable information (PII), in many cases the Identity Provider should also play a central role in this determination. Principals may not be prepared to define policies to control their privacy information in instances where they have not fully understood the privacy implications:

- Some attributes that are used for formal identification purposes, as the legal name, require a close control of privacy and Principal may not be aware of it.
- Some attribute values can be deduced from the combinations of other attributes value (date of birth from age and birthday) and the policies have to be defined considering it.
- In situations where the Principal has the right to expect full anonymity, their identity can often be determined from a small set of attributes (e.g., date of birth, date of hospitalization, type of medical treatment, postal code). In cases such as these, the Identity Provider needs to understand what policies are necessary in order to properly protect the Principal's anonymity.

The Attribute Provider needs to define some basic/default policies to protect Principal's privacy. These rules should be written in such a way that a Principal has to consciously choose not to use these rules (that is, the Principal has to “opt-in” for a weaker privacy policy).

There may be other reason so that the Attibute Provider or the entity managing the Attribute Provider infrastructure (e.g., telecom operator etc) defines its own policies. Besides these policies (Principal's, Attribute Provider), other policies may be needed in order to cover legal issues of the jurisdiction. Since the existence of different kind of policies may occur for the same attributes, a priority mechanism is needed for cases in whch those policies are contradictory . Thus it can be decided which policy has a higher priority and therefore which policy is applied.

The definition of policies to safeguard the Principal's privacy is not only applicable to the attributes but also to the use of the specific identity service; this is, there will be policies to decide if the Service Provider can use the identity service. Some of these policies may be based on the Principal whose information is being requested (e.g., the ID-Personal Profile service as a whole is denied to an Service Provider if this is looking for some VIP Principal).

4.1.2.2 2. Policy applicability

Defined policies may apply to a specific attribute, they can apply to a container so that the policy is applicable to all the attributes within said container or they can even apply to the whole set of attributes so that a partiuclar Service Provider cannot access any of the Principal's attributes. Moreover, the Attribute Provider's policies or legal policies may be defined in such a way so that certain Service Providers do not have access to the service. This means that there can be two kinds of policies:

- Those defined for the usage of the identity service ("**service privacy**"); this is, the resources that can be returned by the DS to the requesting Service Provider.
- Those defined for the access to Principal information ("**Principal privacy**"); this is the attributes that can be returned by the Attribute Provider to the requesting Service Provider.

It is perfectly reasonable to evaluate the policies from a higher definition level to a lower, e.g., policies at container level will first be evaluated and if that policy is satisfied, then the policies for the attributes of that container will be evaluated. For example, there may be a policy allowing access to the Address container, but with restrictions on street address, allowing only Postal Code, Locality or City, State or Provence, and Country to be sent in the answer.

4.1.2.3 3. Usage directives

The Liberty ID-WSF architecture incorporates a usage directive facility, which allows requesters to designate the use they intend for requested data, and allows providers to designate the permitted uses of released data. While it is intended that this facility can be leveraged to integrate processing of privacy policies into Liberty ID-WSF protocol exchanges, the usage directives' scope is not confined to this purpose. The architecture provides a general means for interacting parties to exchange policy statements, and is suitable for use with various policy expression languages. In order to apply the usage directive facility effectively, implementers responsible for a set of interoperating Liberty components must agree on a common set of supported policies, and on the expression language to use to represent those policies.

For example a WSC may include usage directives in a request sent to a WSP, known as request usage directives. Request usage directives may include information about the WSC, the purpose of the request, whether there is intent to share any returned information with other parties, and so forth. Request usage directives will be evaluated at the WSP against any applicable policies governing the requested information in order to determine whether the intended usage of the requested information complies. If so, then the WSP will reply to the request with the requested information, and the WSP may include usage directives of its own in the response. These response usage directives stipulate what the WSC may do with the returned information, for example whether the information may be shared with other parties.

Incorporating request usage directives as a factor in policy decisions at a WSP will influence the policy expression language used to define site-specific policies. This is by virtue of the usage directives themselves being expressed in some language. The site-specific policies do not necessarily need to be expressed in the same language as the request usage directives. But if they differ, it must be possible to create an effective mapping between the expression languages.

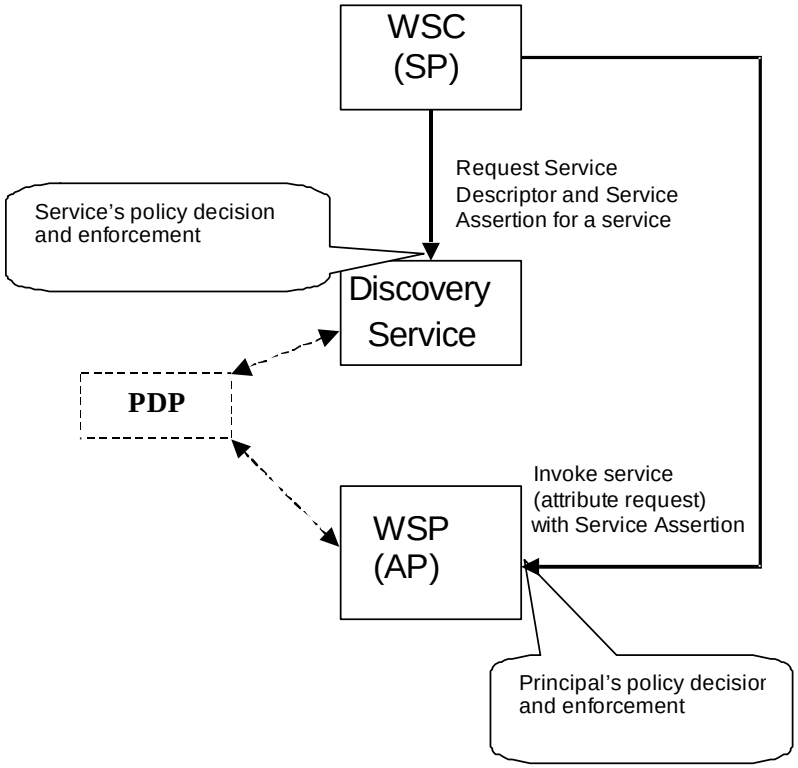
Incorporating usage directives cannot ensure the integrity of a Principal 's privacy since the requester, the WSC, might request information using an attestation of adherence to a strict privacy policy, and subsequently not adhere to it. However, this issue is out of scope for the Liberty specifications; rather, it is in-scope for the judicial system.

4.1.2.4 4. Policies decision and enforcement

The policies concerning **service privacy** have to be checked (policy decision) and executed when there is any request (DiscoveryLookup) to the Discovery Service. The policy decision and enforcement is executed before sending the information on the Attribute Provider holding the Principal attributes and therefore the Discovery Service acts as a

policy enforcement point (it could act as well as Policy Decision Point but the decision could be delegated to other entity controlling the service policies).

The policies concerning **Principal's privacy** have to be executed when there is any attribute request to the Attribute Provider. The policy decision and enforcement is executed before sending the requested information about the Principal's attributes and therefore the Attribute Provider acts as a Policy Enforcement Point (it could act as well as Policy Decision Point but the decision could be delegated to other entity controlling the Principal's policies).



When controlling the access to the whole set of attributes of certain Principals (e.g. some Service Provider doesn't have access to the Attribute Provider if the request is on a VIP Principal), the policies can be regarded as:

- Policies controlling the access to the services (for a specific Principal) and in this case the policies are enforced in the DS.
- Policies controlling the access to the attributes (the whole set) of a Principal and in this case the policies are enforced in the Attribute Provider.

5 References

[LibertyArchOverview] J. Hodges and T. Wason, eds., Copyright 2003, Liberty Alliance Project, *Liberty Architecture Overview 1.1*.

[LibertyDataService] J. Kainulainen, ed., Copyright 2003, Liberty Alliance Project, *ID-WSF Data Service Template.*

[LibertyGlossary] H. Mauldin, ed. Copyright 2003, Liberty Alliance Project, *Liberty Architecture Glossary 1.1*, liberty-arch-tech-glossary-v1.1.pdf.

[LibertyIDPersonalProfile] S. Kellomaki, ed., Copyright 2003, Liberty Alliance Project, *Liberty Identity Personal Profile Service Specification*, draft-lib-svc-id-pp-v1.0-17.

[LibertyID-WSFSecurity] G. Ellison, ed. Copyright 2003, Liberty Alliance Project, *Liberty Identity Services Framework Security Profiles*, draft-lib-arch-security-profiles-v1.0-06.

[LibertyInteractionService] R. Aarts, [ed.] Copyright 2003, Liberty Alliance Project, *ID-WSF Interaction Service*.

[LibertyISFPrimer] J. Tourzan, ed. Copyright 2003, Liberty Alliance Project, *Liberty Identity Web Services Framework Primer*.

[LibertyTrustModels] J. Linn, ed. Copyright 2003, Liberty Alliance Project, *Liberty Trust Models 1.0-12*, draft-liberty-tsp-trust-models-v1.0-13.

[OASISGlossary] J. Hodges, ed. Copyright 2003, *OASIS Security Services TC: Glossary*.

[PKIX-WG] A. Arsenault, Diversinet, and S. Turner, PKIX Working Group, I*nternet X.509 Public Key Infrastructure Roadmap*, July 2002, http://www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-09.txt

[SAMLGlossary] J. Hodges, ed. Copyright 2003, OASIS, *Glossary for the OASIS Security Assertion Markup Language (SAML)*, 31 May 2002.

[WS-ISecurity] A. Wesley, Web Services Interoperability Association, *WS-I Security Plan Framework*, 3 March 2003.