System Architecture

Open Federation

Version 1.0

Please send comments to: dev@opensso.dev.java.net

This document is subject to the following license:

Contents

1 Introduction

1.1 Document Status

Project Name	Open Federation
Document Title	System Architecture
Date of Issue	November 1, 2006
Current Version	1.0
Issuing Organization	Sun Microsystems, Inc.
Feedback E-mail	dev@opensso.dev.java.net

1.2 Revision History

Date	Version	Author	Comments
November 1, 2006	1.0	Qingwen Cheng	Initial Revision

1.3 Summary

Open Federation provides a robust framework for implementing federated identity infrastructures and for deploying web services. Multiple industry standards, such as OASIS SAML, Liberty ID-FF, ID-SIS, ID-WSF, are supported in the Open Federation system.

The Open Federation deliveries a federated identity solution which extends Web Single Sign-on across corporate boundaries. The purpose of this document is to provide the architectural details for the implementation of the Open Federation system. The structure of this document is based on the recommendations provided by IEEE Standard 1471-2000 [1]. All the important terms, acronyms, or abbreviations used in this document are defined in the Glossary section.

1.4 Scope

This document essentially describes the detailed architecture of the Open Federation framework. It is not a requirements document although it does reflect the functional requirements addressed by this architecture.

The purpose of this document is to provide a certain level of specifics about the Open Federation framework as well as to serve as an effective vehicle for facilitating any design and implementation practice pertaining to the Open Federation framework.

1.5 Context

1.5.1 Identity Federation

Consider the many times you might access service provider accounts in a single day; sending and receiving email, logging in to a news portal, checking bank balances, finalizing travel arrangements, bidding on auction items, accessing utility accounts, and shopping online are all possible services for which you would define an identity. For each of these services, you have configured a multitude of separate accounts that distinguish themselves from others. Each time you want to access one of these services, you identify yourself to the provider by logging in. Considering the number of service providers for which you can define a local identity, it can make accessing each one time-consuming and frustrating.

This disjointed identity phenomenon offers the opportunity to fashion a system for computer users to link their local identities. With the introduction of Circle of Trust and identity provider, *Identity federation* allows the user to associate, connect or bind the various local identities they have configured for multiple service providers. The linked local identities, referred to as a *federated identity,* then allow the user to log in to one identity provider site and click through to an affiliated service provider site without having to re-authenticate their identity again. The Open Federation framework is provided to endorse this notion of *single sign-on,* the implementation supports several open standard technologies, such as OASIS SAML v1/v2 and Liberty Alliance Identity Federation Framework, therefore encouraging an interoperable infrastructure among providers.

1.5.2 Identity Web Services

Many enterprises now-a-days have heterogeneous, complex IT environments that cause numerous logistical and support issues. To complicate matters, those same enterprises are now exposing and deploying Web services outside the firewall to customers and partners. Immediately question would be how could security be enforced in this deployment? Transport security protocols, such as SSL and TLS, help secure the communication among nodes, but that's not always enough, the ultimate Web-service provider only knows the identity of its immediate consumer—the intermediary—with no secure records of the ultimate consumer. Some additional questions like, how should the Web-service provider manage trust relationships with a large population of identities? how does the enterprise resolve identities across organizational boundaries? how does the Web-service consumer locate the correct Web-service provider for a particular identity?

The Open Federation provides a framework for identity and general web services based on existing standards, such as Liberty Alliance Identity Web Services Framework (ID-WSF) and WS-I Basic Security Profile. Built on those standards, Open Federation presents a flexible and extensible solution to resolve the problems of identity-enabling Web services, service discovery and invocation, security, privacy and ease-of-deployment.

1.6 Glossary

Authentication	The process by which the identity of a principal is established within the system. This process may involve explicit user interaction with the system outside the scope of any of the web applications that participate in Single Sign-on.
Authentication Web Service	A web service provider that facilitates the authentication of principals within the system.
Circle of Trust	A group of Service Providers and Identity Providers that have business relationships based on architecture and operational agreements, and with whom users can transact business in a secure and apparently seamless environment.
Federation Termination	Termination of Identity Federation.
Firewall	An entity that limits access to and from a network based on the configured security policies.
HTTP	Acronym for Hypertext Transfer Protocol. HTTP is an open standards based protocol used for exchange of information between web browsers and web servers.
ID-FF	Liberty Alliance Identity Federation Framework
ID-SIS	Liberty Alliance Identity Service Interface Specification
ID-WSF	Liberty Alliance Identity Web Services Framework
Identity Federation	Identity management protocols that enables organizations to share trusted identities across the boundaries of the corporate network.
Identity Provider	An entity that creates, maintains and manages identity information for Principals, and provides authentication of Principals for other Service Providers within the same circle of trust.
Open Federation	An open source project by Sun Microsystems Inc. to provide an extensible framework to support Identity Federation and Identity Web Services.
Open Federation Library	A subcomponent of Open Federation system that defines a pluggable infrastructure for Identity Federation and Identity Web Services.
OpenSSO	Alias for the Open Web Single Sign-On project. This project is an open source initiative of Sun Microsystems Inc., that provides the

	foundation of identity services for the web platform.
SAML	Security Assertion Markup Language
SASL	Acronym for Simple Authentication and Security Layer, a method for adding authentication support to connection-based protocols.
Service Provider	An entity that provides certain service to Principals.
Single Logout	The process to logout user authentication sessions from all Identity Providers and Service Providers in the Circle of Trust when the user logs out from one Service Provider or Identity Provider.
Single Sign-on	The ability to use proof of an existing authentication with one Identity Provider to create a new authentication session with another Identity or Service Provider.
SLO	Acronym for Single Logout.
SOAP	Acronym for Simple Object Access Protocol. SOAP is is an open standard protocol for exchanging XML messages over a transportation protocol, such as HTTP. SOAP forms the foundation layer of the Web Services stack, providing a basic messaging framework that more abstract layers can build on.
SPI	Acronym for Service Provider Interface. This is the interface normally used to extend the server functionality.
SSL	Acronym for Secure Socket Layer. SSL provides a means to encrypt communication between two entities in such a way that it becomes illegible to any other entity.
SSO	Acronym for Single Sign-on.
System	In the context of Open Federation, the System represents a complete deployment where various web applications participate in an Identity Federation and Identity Web Services environment using the identity services provided by Open Federation.
System Stakeholder	A set of people who interact with the system at various stages and in different capacities. The system stakeholders could be individuals, teams or organizations.
TLS	Acronym for Transport Level Security. TLS is designed as a successor to SSL and uses the same cryptographic methods but supports more cryptographic algorithms.
URL	Acronym for Uniform Resource Locator. A URL contains the necessary information regarding the address and access mechanism needed to

	access a resource available on the network.
User Agent	A software or device, such as browser or cellphone, which initiates requests on behalf of a Principal.
Web Application	An application hosted on either a Web Server or an Application server and is accessible via the web using a traditional browser.
WSC	Acronym for Web Service Consumer. WSC is a client to access web service provider.
WSP	Acronym for Web Service Provider. WSP is a system entity to provide certain web services to WSC.
XML	Acronym for extensible Markup Language. XML is an open standards based data markup language used for representing structured data.

1.7 References

[1] IEEE Std. 1471-2000, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE-SA Standards Board – September 2000.

[2] *System Architecture – Open Web Single Sign-On*, Version 1.0

[3] *Use Cases – Open Federation*, Version 1.0

[4] *Use Cases – Open Web Single Sign-On*, Version 1.0

[5] Liberty Alliance ID-FF, ID-WSF, ID-SIS Specifications

[6] SAML V1.0, V1.1, V2.0 Specifications

[7] WS-I Basic Security Profile 1.0

[8] WS-Federation Passive Requestor Profile

[9] Liberty Interoperable (TM) Interoperable Products

[10] Liberty Interoperable (TM) Documents

2 Objectives

2.1 Mission

The mission of the Open Federation is to provide an open and extensible framework for Identity Federated and Identity Web Services that would facilitate the deployment of interoperable customized applications.

2.2 Stakeholders

System stakeholders are the people who interact with the system during different life-cycle phases, in different capacities, and for different purposes. These could be individual users, teams and organizations that are charted with the development, adoption or execution of this system. The key stakeholders for the Open Federation are:

- **Developers**: Responsible for the overall development of the system. They may be involved various development related activities associated with the system such as designing, developing, building, testing, documenting, and troubleshooting the system.
- ***Administrators***: Privileged users who are chartered with deployment and configuration of the system in staging and production environments. Administrators can control the system behavior via the available configuration mechanisms at various levels and thus affect the way the system operates. They are expected to perform system checks to ensure its operations and take corrective actions where necessary if the system fails to perform satisfactorily.
- ***End Users***: Users who access the hosted web applications and do not have any special privileges to alter the behavior of the system for others.
- ***Application Developers***: Developers who are responsible for the creation and deployment of applications on the network. These developers may use the Open Federation Client API and public interfaces to perform application business logics and to communicate with Open Federation framework.
- ***Web Application Administrators***: Administrators who are chartered with the deployment of web applications. These administrators will configure Open Federation Client APIs as necessary to ensure that the hosted web applications can take advantage of the Open Federation Framework.
- ***System Integrators***: Developers who are chartered with the deployment of Open Federation in a given environment to integrate with existing Authentication, Session, Authorization, Auditing, Configuration and Datastore infrastructure. They may be involved in implementing public SPIs to plug-in existing infrastructure where necessary.

2.3 Architectural Concerns

Associated with the stakeholders are the architectural concerns based on their expectations and requirements from the system. While there are many concerns associated with each stakeholder's role, the architecture is defined by relatively few concerns since the rest are considered generic and apply to most software intensive systems. These generic architectural concerns are addressed by established best practices and processes, and are not explicitly included in this document. Further, the concerns that drive this architecture are derived from analysis of system requirements and are not necessarily reflective of individual or a collection of high level requirements in the form that they are specified. This section identifies the core architectural concerns that are pivotal in shaping the overall architecture of Open Federation.

2.3.1 Heterogeneity and Openness

The Open Federation framework should not place any restriction on the use of any specific underlying network technology, computer hardware, operating systems, programming languages or other hardware or software entities. This concern is based on the ability of Open Federation to provide identity services to web applications that could be developed in any programming language, be hosted on any web or application server, deployed on any operating system that utilizes any underlying hardware and network facilities.

2.3.2 Industry Standard and Interoperability

The Open Federation framework must be based on and conform to open industry standard on Identity Federation and Identity Web Services, thus to achieve interoperability among different vendors on heterogeneous systems.

2.3.3 Extensible and Customizable Architecture

The Open Federation framework must be extensible and customizable in order to allow the creation and integration of custom identity services that can be added to the system to support specific use cases. The concern is to enable system integrators and other developers to provide more functionality via Open Federation framework than what is provided by default.

2.3.4 Availability and Reliability

The Open Federation framework should provide a function of service continuation, which means the ability to honour the request even if the request reaches a different system than the original system that served an earlier request. This could happen due to the presence of network elements such as load balancers in the network.

2.3.5 Security, Confidentiality and Privacy

Information associated with an identity should be handled by Open Federation system in a secure and confidential manner, privacy of the identity must be guaranteed whenever applied to avoid leaking of identity information to unwanted parties. The concern is to based on the needs to establish trust and share vital identity information in a very secure manner among various parties which may across

corporate boundaries.

2.3.6 Performance and Scalability

Given the limitation of the capabilities of the underlying hardware and software components, the Open Federation framework should perform and scale vertically and horizontally to the necessary levels in order to meet critical throughput requirements from customer.

2.3.7 Auditing

The Open Federation framework should provide the facility to log all its activity events to keep track of identity interactions and erroneous conditions.

3 Architectural Views

This chapter presents the various architectural viewpoints and the associated views. Architectural viewpoints describe the necessary agreement between the system and its environment to ensure that the architectural concerns are addressed by the resulting system. Architectural views fill in the necessary details for these viewpoints by elaborating on how the system addresses the associated concerns. Views are depicted via the use of diagrams that are based on UML concepts but may not fully conform to UML style specifications.

3.1 System Context View

3.1.1 Viewpoint Specification

Name	System Context Viewpoint
Stakeholders	All
Concerns	System Requirements
Modeled As	Use-Case
Viewpoint Source	System Requirements

3.1.2 Detail

Please refer to the Open Federation Use Cases document [3] for details regarding System Context View.

3.2 Extensible and Customizable Infrastructure View

3.2.1 Viewpoint Specification

Name	Extensible and Customizable Infrastructure Viewpoint
Stakeholders	Developers, Administrators, System Integrators
Concerns	Extensible and Customizable Infrastructure, Industry Standard, Auditing
Modeled As	Service Provider Interface
Viewpoint Source	System Requirements

3.2.2 Detail

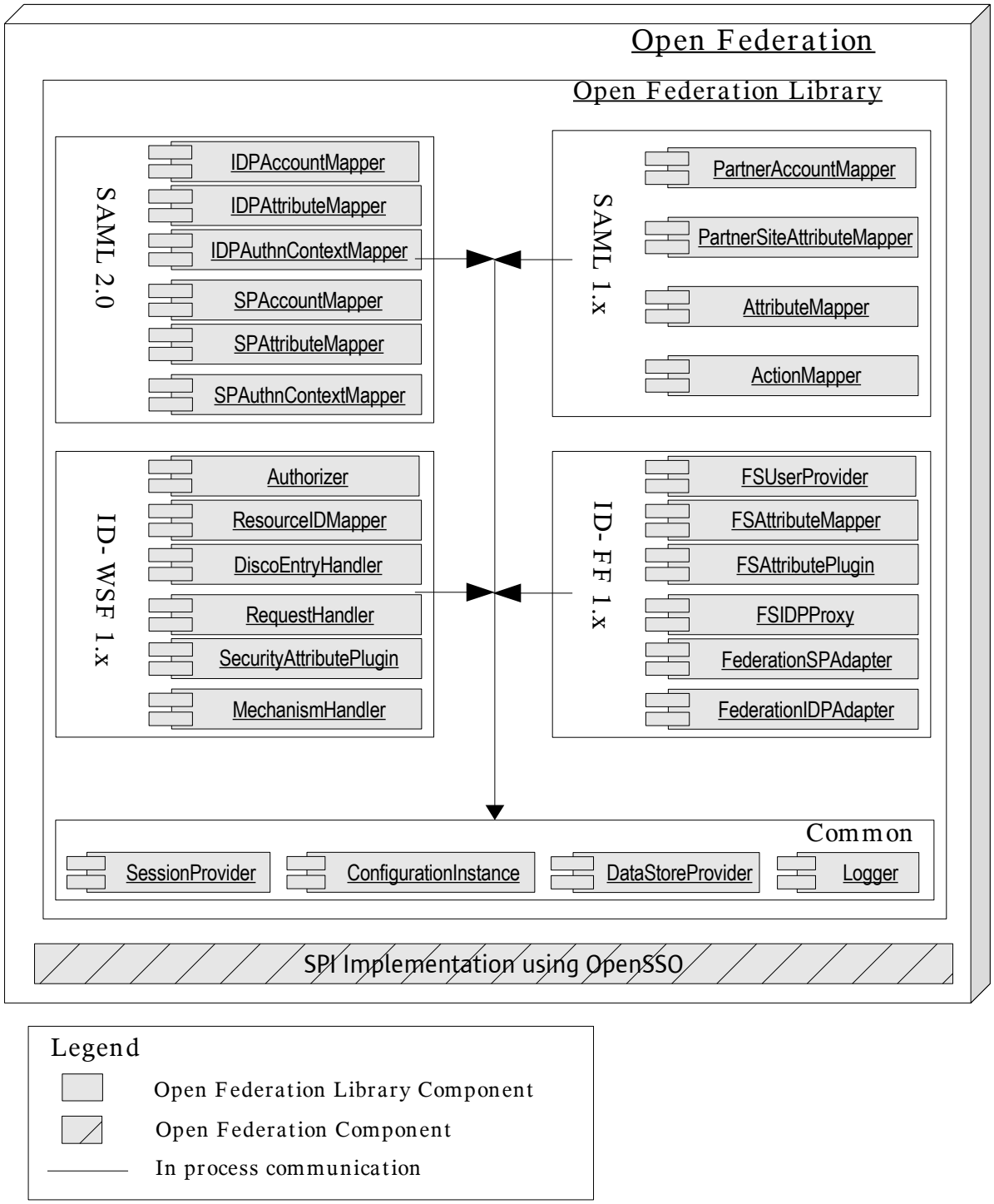

Figure 1: Extensible and Customizable Infrastructure View

IDPAccountMapper Interface Summary (SAML 2.0)
Maps local identity to the Name Identifier on IDP side.
Maps Name Identifier in *ManageNameIDRequest* to local identity on IDP side.

IDPAttributeMapper Interface Summary (SAML 2.0)
Returns list of Attributes from the local identity on IDP side. Those Attributes will be included inside the Single Sign-on Assertion to be passed down to SP side.

IDPAuthnContextMapper Interface Summary (SAML 2.0)
Maps *RequestedAuthnContext* from Service Provider to the IDP side authentication scheme.

SPAccountMapper Interface Summary (SAML 2.0)
Maps Name Identifier in Assertion sent from IDP to local identity on SP side.
Maps Name Identifier in *ManageNameIDRequest* to local identity on SP side.

SPAttributeMapper Interface Summary (SAML 2.0)
Maps Attributes inside Assertion sent from IDP to local Attributes on SP side.

SPAuthnContextMapper Interface Summary (SAML 2.0)
Maps local request to standard *RequestedAuthnContext* to be sent to IDP side for authentication.
Maps *AuthnContext* from IDP (i.e. authentication performed on IDP) to local authentication level on SP side.

PartnerAccountMapper Interface Summary (SAML 1.x)
Maps subject specified in Assertion to local identity. This is used in Single Sign-on (Artifact and POST profile), Attribute Query and Authorization Decision Query case.
Maps subject specified in *SubjectQuery* to local identity. This is used in *Attribute Query* case.

PartnerSiteAttributeMapper Interface Summary (SAML 1.x)
Returns list of Attributes from a local identity on SAML Assertion producer side. Those Attributes will be included inside the Single Sign-on Assertion to be passed down to SAML Assertion consumer side.

AttributeMapper Interface Summary (SAML 1.x)
Maps attributes inside *AttributeQuery* to local attributes for the local identity.

ActionMapper Interface Summary (SAML 1.x)
Maps *AuthorizationDecisionQuery* to Open Federation policy decision for the local identity.

FSUserProvider Interface Summary (ID-FF 1.x)
Maps name identifier in ID-FF request/response protocol message to the local identity. This is used in all ID-FF protocols, such as Single Sign-on, Federation, Single Logout, Termination, Register Name Identifier, Name Identifier Mapping.

FSAttributeMapper Interface Summary (ID-FF 1.x)
Maps Attributes inside *AttributeStatements* to local attribute of the local identity.

FSAttributePlugin Interface Summary (ID-FF 1.x)
Returns list of Attributes from the local identity on IDP side. Those Attributes will be included inside the Single Sign-on Assertion to be passed down to SP side.

FSIDPProxy Interface Summary (ID-FF 1.x)
Returns preferred IDP based on Authentication request. This is used, in the case when IDP Proxy is enabled, by the proxying IDP to find out the real IDP for authentication.

FederationSPAdapter Interface Summary (ID-FF 1.x)
Provides mechanisms on SP side to plug-in customized process logics during Single Sign-on, Federation, Single Logout, Termination and Name Identifier Registration.

FederationIDPAdapter Interface Summary (ID-FF 1.x)
Provides mechanisms on IDP side to plug-in customized process logics during Single Sign-on, Federation, Single Logout, Termination and Name Identifier Registration.

Authorizer Interface Summary (ID-WSF 1.x)
Checks if a WSC is authorized to query or modify an object.
Maps an action and data to be accessed to an authorization decision.

ResourceIDMapper Interface Summary (ID-WSF 1.x)
Maps Resource ID to local Identity and vise versa. This is used by the Discovery Service and WSP in the B2E use case to find out local identity based on Resource ID or find Resource ID based on the local identity.

DiscoEntryHandler Interface Summary (ID-WSF 1.x)
Provides interfaces to get and set discovery service entry for an identity. This is used by Discovery Service to handle discovery service Query and Modify request.

SecurityAttributePlugin Interface Summary (ID-WSF 1.x)
Provides mechanism to include SAML Attributes via the *AttributeStatement* into the *SecurityAssertion* during the Discovery Service credential generation.

RequestHandler Interface Summary (ID-WSF 1.x)
Provides hook to integrate a WSP into Open Federation system. This interface needs to be implemented by each WSP and registered with the SOAP Binding service.

MechanismHandler Interface Summary (ID-WSF 1.x)
Provides mechanism to handle *SASLRequest* and generates *SASLResponse*. This needs to be implemented on per SASL mechanism basis, and is used by Liberty Authentication Web Service to map SASL mechanisms to Open Federation authentication schemes.

SessionProvider Interface Summary
Provides mechanisms to manage principal's session object, including creating session, obtaining session, setting/getting session properties, validating session and adding session listeners.

DataStoreProvider Interface Summary
Provides generic mechanisms to manage user data store, including setting/getting user attributes, and searching for users.

ConfigurationInstance Interface Summary
Provides mechanisms to manage services (e.g. Discovery service) and metadata (e.g. ID-FF and SAML 2.0 metadata) configuration.

Logger Interface Summary
Provides mechanisms to write access and error logs.

3.2.3 Description

As shown in the View, Open Federation consists of Open Federation Library and SPI implementations using OpenSSO APIs.

Open Federation Library provides a pluggable framework for Identity Federation and Web Services. Followings are the list of industry standard supported in Open Federation Library implementation:

1. Liberty ID-FF 1.1 & 1.2, including SP/IDP extended profiles.
2. Liberty ID-WSF 1.0 & 1.1.
3. OASIS SAML 1.0 & 1.1.
4. OASIS SAML 2.0.

Features to be supported in future releases of Open Federation Library:

1. Liberty ID-WSF 2.0
2. WS-Federation Passive Requestor Profile
3. WS-I Basic Security Profile

To achieve extensibility and customizibility, a list of Service Provider Interfaces are provided in each standard implementation to satisfy different deployment use cases. A set of common Service Provider Interfaces used by all components are also defined to integrate with existing authentication, configuration, session, logging and data store infrastructure.

Open Federation system provides default implementation of those SPIs defined in the Open Federation Library using OpenSSO APIs. Those SPI implementations integrate Open Federation seamlessly with OpenSSO by utilizing its rich features in session, authentication, service management, logging and identity repository components. A customized implementation could be provided to replace the default if needed, but need to make sure other SPI implementations

are not affected by this change.

3.2.4 Extension

N/A

3.3 Simple Deployment View

3.3.1 Viewpoint Specification

Name	Simple Deployment Viewpoint
Stakeholders	All
Concerns	Heterogeneity and Openness, Interoperability
Modeled As	Deployment
Viewpoint Source	System Requirements

3.3.2 Details

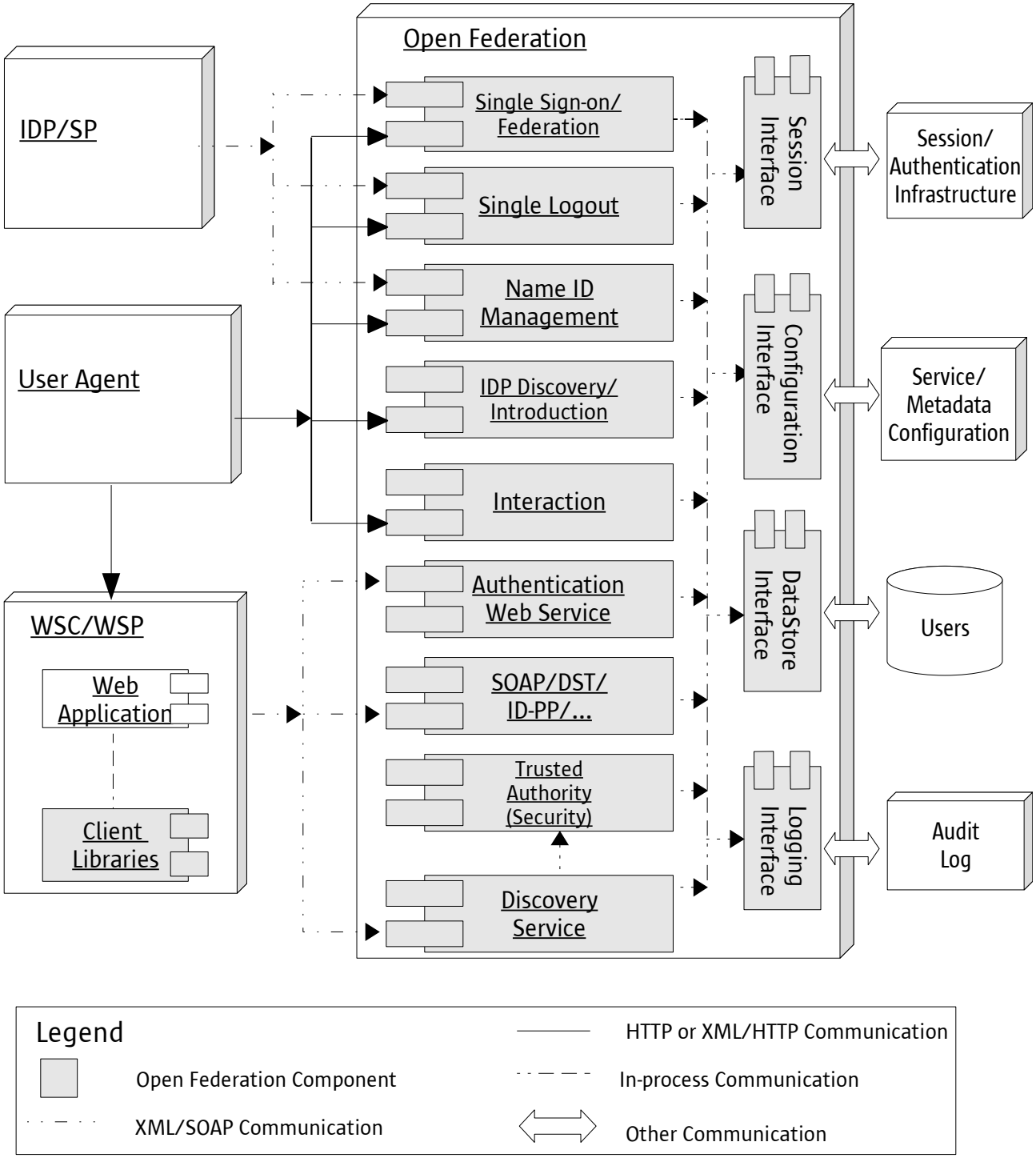

Figure 2 : Deployment View

3.3.3 Description

This view presents a simple deployment diagram for Open Federation system.

Identity Federation components shown in this view are the core federation components : Single Sign-on/Federation service, Single Logout service, Name ID Management (including Federation Termination, Name ID Registration) service, IDP Discovery/Introduction service. The remote IDP/SP and browser interacts with the Identity Federation components using standard protocols (e.g. SAML 1.x, SAML 2.0, Liberty ID-FF 1.x).

Identity Web Services components shown in this view are the main liberty ID-WSF services: Discovery Services, Trusted Authority (Security), Authentication Web Service, SOAP Binding, DST, ID-PP services, Interaction and Client Libraries. A simple scenario would be:

1. Browser accesses WSC.
2. WSC uses the Client Libraries to authenticate to Authentication Web Service, and obtains Discovery bootstrap information.
3. WSC uses the Client Libraries to accesses Discovery service to request access to ID-PP services.
4. Discovery Services returns ID-PP endpoint plus necessarily credentials to the WSC.
5. WSC uses the Client Libraries to access ID-PP service.
6. ID-PP returns attributes to the WSC.
7. WSC processes ID-PP response and presents access page to the browser.

Common Components shown in this view are the Plug-in SPIs used by both Identity Federation and Identity Web Services components : Session, Data store, Configuration and Logging interfaces. Those SPIs are part of the Open Federation Library, and default implementation are required in Open Federation system in order to communicate with external Session/Authentication infrastructure, Service/Metadata Configuration, User repository and Audit Logging system

3.3.4 Extension

1. The view could be extended to include LECP/ECP between external SP/IDP, Browser and Open Federation system.
2. The View could be extended to include Client Library to communicate with Open Federation using SOAP.

3.4 High Availability Deployment View

3.4.1 Viewpoint specifications

Name	High Availability Firewall Friendly Deployment Viewpoint
Stakeholders	Developers, Administrators, System Integrators
Concerns	Availability and Reliability, Performance and Scalability
Modeled As	Deployment
Viewpoint Source	System Requirements

3.4.2 Detail

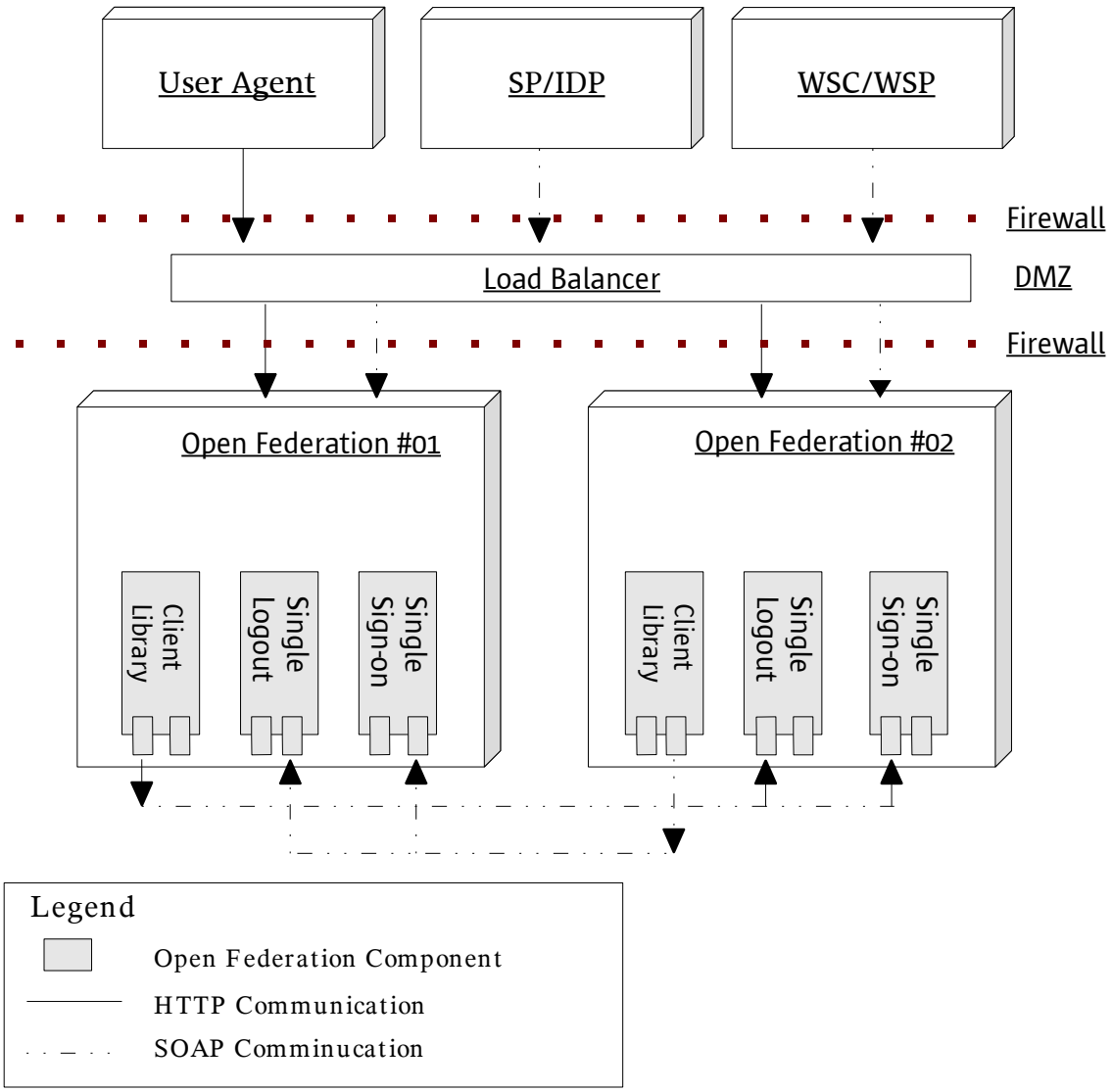

Figure 3: Load Balancer Enabled Deployment View

3.4.3 Description

In this deployment view, any number of Open Federation instances could be added behind the Load Balancer for redundancy and high availability. All instances here are presented as one single virtual IDP or SP system to the external partners. As Open Federation system stores transient information, such as Assertion corresponding to an Artifact, in the internal memory, it

is possible an request is redirected to a different Open Federation instance which does not have the transient information in order to complete the transaction. For browser based profile, this problem could be resolved using stickiness feature, such as LB cookie, supported by Load balances. But this won't work for SOAP based profile. For example, in Single Sign-on using HTTP Artifact profile, Service Provider need to send SOAP request to IDP to retrieve the Assertion. This SOAP request might land in the wrong server which does not have the in-memory Assertion information. Same problem applies to SOAP based Single Logout. The solution here is to assign a unique Server ID (e.g. 01, 02, 03, ...) to each Open Federation instance. Everytime when an Artifact or SessionIndex is created, the Server ID is included. If the request lands on the wrong instance, the Client Library retrieves the Server ID, finds the remote instance which contains the information, then sends a SOAP request to the remote instance to process the request. After getting the response from the remote instance, the Open Federation instance will perform remaining protocols accordingly.

3.4.4 Extension

1. The view could be extended to introduce an internal facing Load Balancer to serve internal application and agents.
2. The view could be extended to include Load Balancer Enabled data stores.

4 Conceptual Implementation

This chapter presents conceptual implementation of Open Federation and its subcomponents with certain levels of details and addresses viewpoint specifications detailed within this Open Federation system architecture document.

Open federation supports Identity Federation and Web Services specifications from OASIS SAML and Liberty Alliance. The framework handles protocol messages according to those standards. In addition to the pluggable interfaces in each specifications, there needs to be common interfaces to provide Session, Authentication, Data Store, Configuration and Logging infrastructure. For example, after validating Single Sign-on Assertion from IDP or Assertion Producer, both SAML (version 1.x and 2.0) and ID-FF based Service Providers need to create Session. The common interfaces identified in Open Federation system are Session, Configuration, Data Store and Logging.

4.1 Session

The main purpose to have a new Session SPI is to provide mechanism to integrate with existing Session and Authentication framework. The Session API should be used only within the Open Federation framework, not new set of API to be used in existing applications.

4.1.1 Proposed Implementation

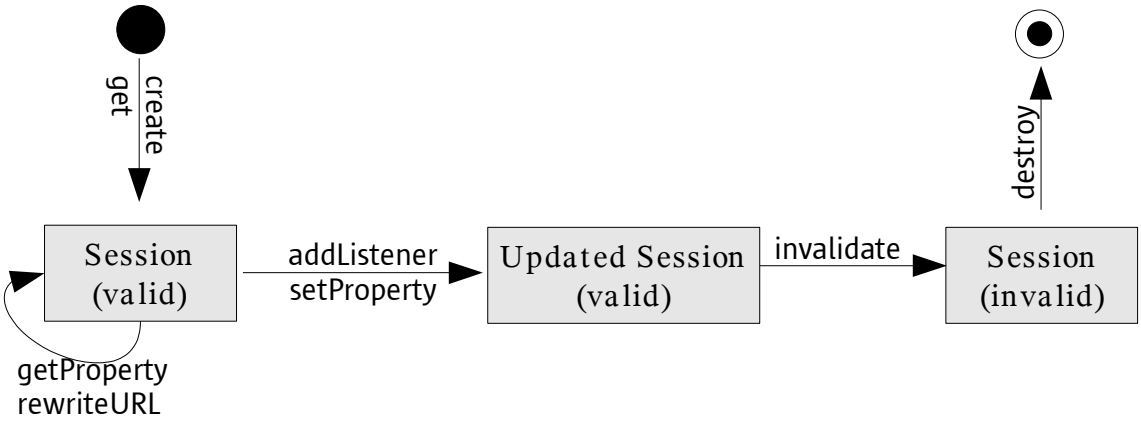

Figure 4 : Session States and Operations

SessionProvider Interface Summary
• Methods to create, retrieve and invalidate session. • Methods to get and set session properties. • Method to add session listeners . • Method to rewrite URLs.

SessionListener Interface Summary
Provides means to perform operations when session is invalidated.

SessionException Class Summary
Represents a session related exception.

Conceptually, Session SPI has two parts. The first part is a method for creating session on Service Provider side once the Response and Assertions from IDP have been validated. The second part is to get a handler to user's session object and perform certain operations (e.g set and get properties, add listener etc.).

In the proposed SPI, SessionProvider is defined as a wrapper over user's session object which is modeled as a Java Object class. This makes it easy to plug-in an existing session implementation. Not all methods in SessionProvider need to supported in the customized plug-in implementation. For example, if session listener is not supported in the application, an UnsupportedException could be thrown by the provider implementation or just do nothing in the implementation, and the Open Federation framework will continue to process other steps.

4.1.2 Session Attributes and Properties

A session must contains following two mandatory attributes:

1. Session ID : an opaque handle used to uniquely identify the session object. This session ID value MUST not change during the lifetime of the session object.
2. Session Principal : the name of the principal whose is the owner of the session object.

A session may contain other optional attributes or properties, such as Authentication Instant, Authentication Level, etc., the actual attributes contains in a session depend on the actual SPI implementation.

4.2 Configuration

The Configuration SPI is designed to manage ID-FF/SAML2 metadata plus any SAML/Liberty related service configuration.

Service Configuration needs to be used to perform following operations:

1. For SAML 2.0 and ID-FF:
    - Creates instance from standard metadata and extended configuration.
    - Imports standard metadata and extended configuration.
    - Retrieves attributes of an configuration instance.
    - Modifies standard metadata and extended configuration for an instance.
    - Deletes a configuration instance.
    - Get all configuration instances.
2. For Circle of trust (COT) in SAML 2.0 and ID-FF:
    - Creates an instance of COT.
    - Retrieves attributes of COT instance.
    - Modifies an instance of COT (e.g. add/remove member to/from a COT).
    - Deletes an instance of COT.
    - Get all COT instances.
3. ID-WSF and SAML1.x
    - Creates default configuration.
    - Retrieves attributes of the default configuration.
    - Modifies attributes of the default configuration.
    - Deletes the default configuration.

Here is the proposed Interfaces for the Configuration component:

ConfigurationInstance Interface Summary
• Methods to retrieve a configuration instance. • Methods to modify a configuration instance. • Methods to create a configuration instance. • Methods to delete a configuration instance. • Methods to retrieve all configuration instances. • Methods to add and remove configuration listeners.

ConfigurationListener Interface Summary
This represents an configuration listener interface which could be implemented to

ConfigurationListener Interface Summary
monitor configuration instance change in the system.

ConfigurationActionEvent Interface Summary
This represents an configuration change event to be used by the ConfigurationListener implementation.

ConfigurationInstance is the main interface to be implemented. ConfigurationListener and ConfigurationActionEvent interfaces are provided to support configuration listener.

4.3 Data Store

When processing protocols in Identity Federation and Identity Web Services, we need to access user's data. For example, to write name identifier to user's entry after successful persistent federation, or to retrieve user attributes in case of Liberty Personal Profile query request.

DataStoreProvider Interface Summary
• Method to retrieve user attributes from the data store. • Method to modify user attributes in the data store. • Method to search for users in the data store. • Method to check if a user exists in the data store.

The proposed DataStore interface is designed to be a general purpose SPI which make it possible to plug-in user's existing datastore. For example, existing application is reading users from LDAP, the datastore SPI could be implemented to retrieve user federation information for the LDAP server, thus to integrate seamlessly with existing application. Default implementation will be provided in Open Federation using IdRepo APIs from openSSO project.

On top of the generic DataStoreProvider SPI, each components (such as SAML 2.0, ID-FF) need to provide utilities to manage protocol and implementation specific data format.

4.4 Logging

Logging and Auditing are common requirements for all components. A common Logging mechanism is desired to provide unified logging format for all components, thus to facilitate

auditing process. Following is the proposed Logger SPI:

Logger Interface Summary
• Method to write to access log. • Method to write to error log. • Method to check if it is loggable for certain logging level.

The proposed logger SPI presents methods to distinguish between access and error logs, so they could be written to different storage if needed. The Open Federation logging service also provides a way to integrate with existing logging infrastructure, so all logging information in Open Federation components could be written to existing logging storage by providing customized Logger SPI implementation.

4.5 Implementation Considerations

4.5.1 Security, Confidentiality and Privacy

Security, Confidentiality and Privacy are the major concerns for an Identity based system. Open Federation implementation should take all measures to protect Security, Confidentiality and Privacy during protocol processing.

- Uses SSL in all protocol transactions to ensure Security and Confidentiality of the XML messages passing between participating parties.
- Follows the Countermeasures defined in OASIS SAML and Liberty Alliance security consideration specifications to prevent or mitigate attacks such as Eavesdropping, Relay, Man-in-the-Middle, Forgery, Stolen Artifact, Denial of Service.
- As the standard specifications already covered different aspects of security and privacy issues, the deployment needs to carefully choose what profiles best suite what they needs. Architecturally, Open Federation must provide all means for deployment team to choose the correct profiles best suited for their needs.
- Builds solutions to protect various services against common attacks such as Cross Site Scripting, Denial of service, etc.
- Utilizes XML digital signing wherever applicable to protect the integration of XML message.
- Utilizes XML encryption when there is an intermediary presented between communicating parties to protect Confidentiality of the XML message.
- Enforces security tokens (e.g. SAML or X509 token) in Identity Web Services transactions.
- Integrates with Policy Service of OpenSSO system when making authorization decision. For example, Authorization Query in SAML, and authorization check for WSC under

Liberty ID-WSF.

- Uses opaque handle in default Name Identifier and Resource ID implementation to ensure Confidentiality of the local identity.
- Follows security consideration for Cookie as stated in System Architecture – Open Web Single Sign-on [2] to safeguard user's session.

4.5.2 Usability

Ease of use is a deciding factor to attract developers and to facilitate deployment. Several approaches should be taken in Open Federation implementation to ensure usability of the system.

- Provides easy and simple steps to install the system. This includes integration with existing session, authentication, data store, configuration and logging infrastructure.
- Provides easy-to-deploy samples to demonstrate features supported by the system. This is to make it easier and quicker for new developers to get to know Open Federation system.
- Provides easy ways to manage the system. This may include UI and command line interface.
- Provides default SPI implementations in Open Federation system to cover as many use cases as possible out of the box. The default implementation should also provide means to adapt to variant use cases without re-programming, for example, through a change of configuration.

4.5.3 Conformance

Standard conformance is key to interoperability between different products. Open Federation must meet conformance requirements for OASIS and Liberty standards and pass the conformance test procedures. Here are current list of interoperability programs offered by Liberty Alliance (refer to [9] and [10]):

1. SAML v2
2. ID-WSD 1.1 and 1.0
3. ID-FF 1.1 and 1.0

5 Conclusion

The architecture presented in this document outlines the key concerns and provides viewpoints with sufficient details to guide the implementation of Open Federation System. The details of each architectural view are targeted toward addressing specific architectural concerns and do not address anything more than that. These details are not sufficient to directly draw a low level implementation plan and calls for an exhaustive design exercise. The conceptual implementation provided in this document may be used as a starting point for low-level design and implementation exercise.