Use Cases

Open Federation

Version 1.0

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

This document is subject to the following license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
http://www.opensource.org/licenses/cddl1.php

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

Contents

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

1 Introduction

1.1 Document Status

Project Name	Open Federation
Document Title	Use Cases
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

This document describes the high level use cases of Open Federation system. It refers to the Architecture Document [1] when writing this document.

1.4 Scope

The use cases are at the highest level. The document covers typical use cases in Identity Federation and Identity Web Services.

1.5 Context

This is a standard use case description using Unified Modeling Language.

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

1.6 Glossary

Attribute Federation	Attributes of the principal, as defined by the identity provider, are used to link to the account used at the service provider.
Circle of Trust (COT)	A group of service providers and identity providers that have business relationships based on operational agreements, and with whom principals can do business transactions in a secure and seamless environment.
DST	Data Service Template
ECP	OASIS SAMLv2 Enhanced Client or Proxy
Federation Termination	Termination of a persistent federation.
IDP	Identity Provider, this is also known as Assertion Producer.
LECP	Liberty Enabled Client or Proxy
Persistent Federation	An identity provider federates the identity provider's principal with the principal's identity at the service provider using a persistent ID.
SASL	Simple Authentication and Security Layer
SLO	Single Logout
SP	Service Provider, this is also known as Assertion Consumer.
SSO	Single Sign-on
Transient Federation	A transient ID is used to federate between the identity provider and the service provider.
UML	Unified Modeling Language
WSC	Web Service Consumer
WSP	Web Service Provider

1.7 References

[1] *System Architecture – Open Federation*, Version 1.0

[2] *Use Cases – Open Web Single Sign-on*, Version 1.0

[3] *OASIS SAML Specifications*

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

[4] Liberty Alliance Project Specifications

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

2 Use Cases

This section presents some common/typical use cases for Identity Federation and Identity Web Services, it is not intended to contain an exhaustive list of all possible use cases.

2.1 Identity Federation Use Cases

All use cases for Identity Federation are based on following two setups : one without LECP/ECP, one with LECP/ECP.

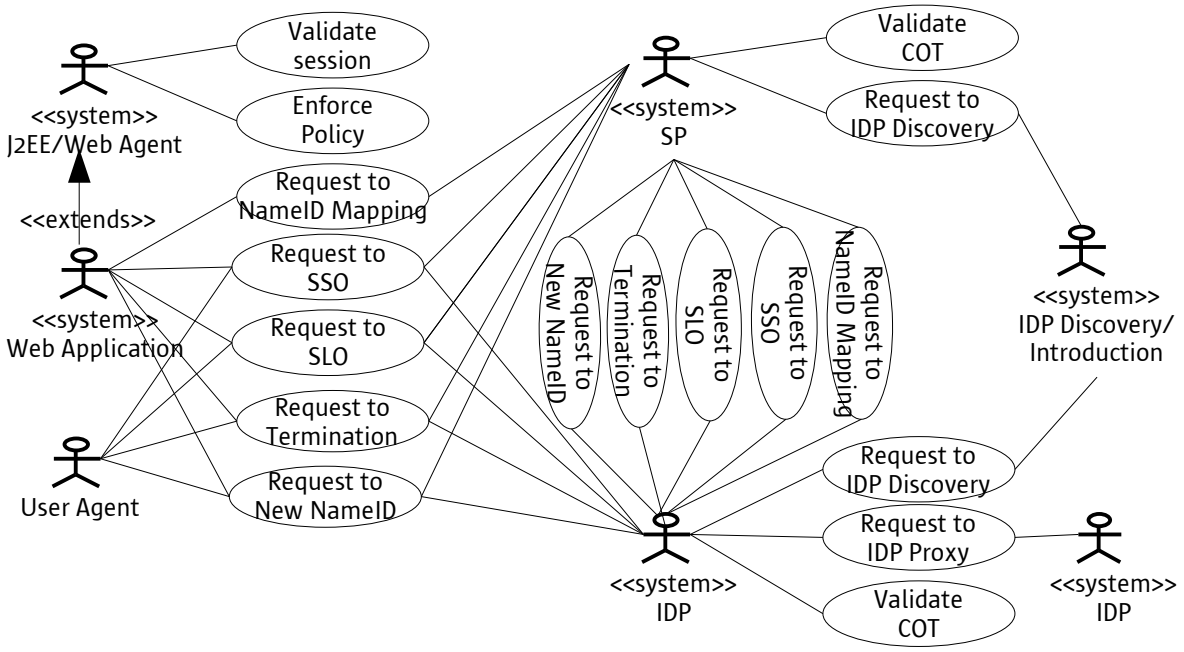

Figure 1. Identity Federation Setup without LECP/ECP

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

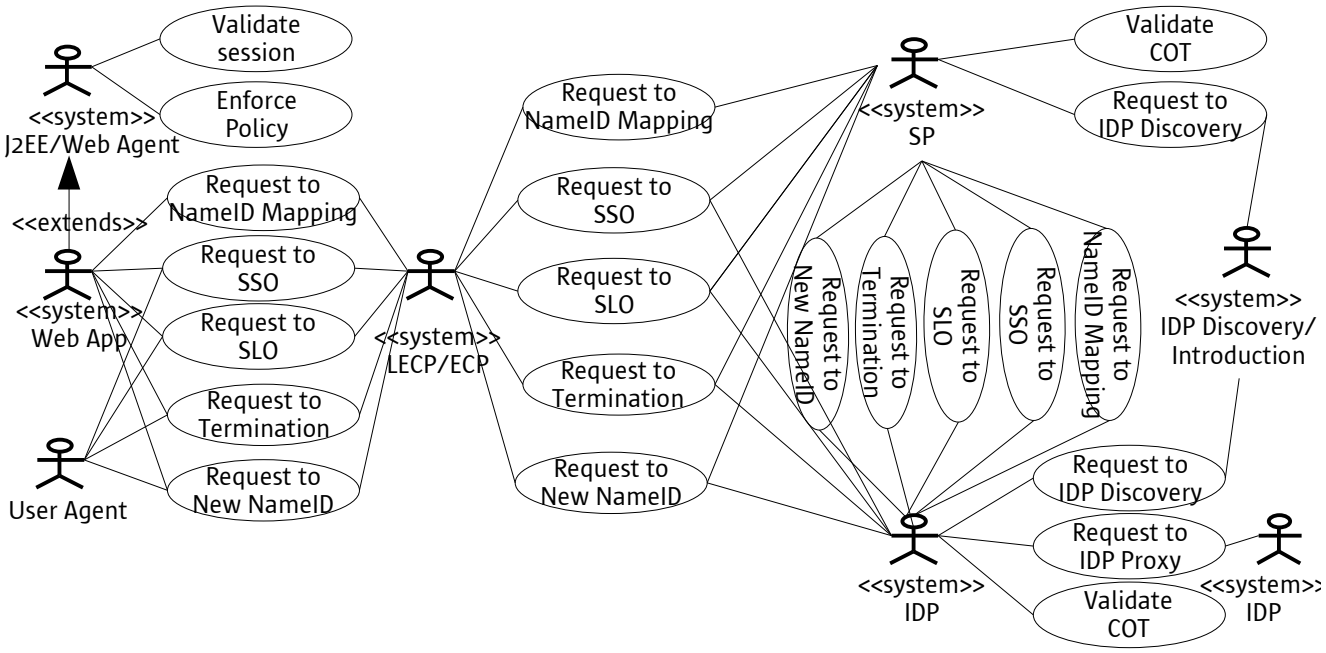

Figure 2. Identity Federation Setup with LECP/ECP

Direct Actors:

1.	User Agent	A software or device, such as browser or cell phone, which initiates requests on behalf of a principal.
2.	Web Application	A web-based application which interacts with User Agent, Service Provider or Identity Provider. It may use Client APIs provided by Open Federation system.
3.	Service Provider	A system entity that provides service to principals. This is also know as Assertion Consumer.
4.	Identity Provider	A system entity that manages identity information on behalf of principals and provides assertions of principal authentication information to other providers. This is also known as Assertion Producer.
5.	LECP/ECP	A Liberty or SAMLv2 client/proxy that has, or knows how to obtain, knowledge about the identity provider that the Principal wishes to use with the service provider.
6.	COT	Acronym for Circle of Trust, a service used to determine if providers trust each other.
7.	IDP Discovery/ Introduction	A service that is used by Service Provider to find out the preferred Identity Provider when multiple IDPs are deployed in same COT.
8.	J2EE Agent	Agent to protect resources on application and portal server based on

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	J2EE Declarative Policies. The agent could be used to enforce policy and single sign-on.
9. Web Agent	Agent to protect resources on Web and Proxy Server. The agent could be used to enforce policy and single sign-on.

2.1.1 UC001 : Persistent Federation

Summary	Federation using persistent name identifier.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent authenticates at Service Provider. 2. User Agent accesses Service Provider URL to request persistent federation with selected Identity Provider. 3. User Agent is redirected to Identity Provider with federation request. 4. Identity Provider processes the request. 5. User Agent is redirected to IDP authentication URL. 6. User Agent authenticates at IDP. 7. Identity Provider generates Single Sign-on Assertion and writes name identifier information to persistent data store. 8. User Agent is redirect back to Service Provider using POST with the Single Sign-on Assertion. 9. Service Provider validates Assertion. 10. Service Provider writes name identifier information to persistent data store.
Alternative Scenario	1. Above scenario is using POST profile, alternative scenario is Artifact profile. In step 8, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before 9: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol. 2. For SAML 2.0, Persistent Federation could be initiated from IDP, i.e. User Agent visits IDP first, then Assertion pushed to SP.

2.1.2 UC002 : Attribute Federation

Summary	Automatic federation based on attribute(s) asserted from IDP.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

Main Success Scenario	1. User Agent authenticates at Identity Provider. 2. User Agent accesses Identity Provider URL to request persistent federation with selected Service Provider. 3. Identity Provider generates Single Sign-on Assertion, the Assertion contains one or more special Attribute key/value pairs about the principal which will be used for account linking purpose on SP side. Identity Provider also writes name identifier information to persistent data store. 4. User Agent is redirected to Service Provider using POST with the Single Sign-on Assertion. 5. Service Provider validates Assertion, and uses the special Attribute(s) inside the Assertion to find the local linking account. 6. Service Provider writes name identifier information to persistent data store. 7. Service Provider generates Single Sign-on token for the local principal.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 4, an Artifact is sent to Service Provider using GET, then one more step need to be performed before step 5: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.3 UC003 : Transient Federation

Summary	Federation using transient name identifier.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent authenticates at Service Provider. 2. User Agent accesses Service Provider URL to request transient federation with selected Identity Provider. 3. User Agent is redirected to Identity Provider with federation request. 4. Identity Provider processes the request. 5. User Agent is redirected to IDP authentication URL. 6. User Agent authenticates at IDP. 7. Identity Provider generates Single Sign-on Assertion with transient name identifier. 8. User Agent is redirect back to Service Provider using POST with Single Sign-on Assertion. 9. Service Provider validates Assertion.
Alternative Scenario	1. Above scenario is using POST profile, alternative is Artifact profile. In step 7, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before Step 8: • Service Provider retrieves Assertion from IDP by presenting artifact

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	using SOAP protocol. 2. For SAML 2.0, Transient Federation could be initiated from IDP, i.e. User Agent visits IDP first, then Assertion pushed to SP.

2.1.4 UC004 : Transient Attribute Federation

Summary	Attribute Federation using transient name identifier.
Priority	Optional
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent authenticates at Identity Provider. 2. User Agent accesses Identity Provider URL to request transient federation with selected Service Provider. 3. Identity Provider generates Single Sign-on Assertion, the Assertion contains one or more special Attribute key/value pairs about the principal which will be used for account linking purpose on SP side. 4. User Agent is redirected to Service Provider using POST with the Single Sign-on Assertion. 5. Service Provider validates Assertion, and use the special Attribute inside the Assertion to find the local linking account. 6. Service Provider generates Single Sign-on token for the local principal.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 4, an Artifact is sent to Service Provider using GET, then one more step need to be performed before step 5: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.5 UC005 : Transient Attribute Federation without Individual SP Account

Summary	Transient attribute federation without individual SP account. All IDP user account will be linked to the same SP account, this is for the deployment when SP side user account is not required.
Priority	Optional
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent authenticates at Identity Provider. 2. User Agent accesses Identity Provider URL to request transient federation with selected Service Provider. 3. Identity Provider generates Single Sign-on Assertion.

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	4. User Agent is redirected to Service Provider using POST with the Single Sign-on Assertion. 5. Service Provider validates Assertion, and links the principal to one fixed local account. 6. Service Provider generates Single Sign-on token for the local principal.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 4, an Artifact is sent to Service Provider using GET, then one more step need to be performed before step 5: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.6 UC006 : Attribute Federation with Auto-creation of SP Account

Summary	Attribute Federation with auto-creation of Service Provider account. This is for the case when a user does not have an account on Service Provider side, but wants to create account automatically upon success of Single Sign-on.
Priority	Optional
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent authenticates at Identity Provider. 2. User Agent accesses Identity Provider URL to request persistent federation with selected Service Provider. 3. Identity Provider generates Single Sign-on Assertion. Identity Provider writes name identifier information to persistent data store. 4. User Agent is redirected to Service Provider using POST with the Single Sign-on Assertion. 5. Service Provider validates Assertion, and automatically registers the user by creating a local account. 6. Service Provider writes name identifier information to newly create user account. 7. Service Provider generates Single Sign-on token for the local principal.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 4, an Artifact is sent to Service Provider using GET, then one more step need to be performed before step 5: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.7 UC007 : Bulk Federation

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

Summary	Bulk Federation, i.e. federation with out-of-band account linking.
Priority	Optional
Direct Actor	Service Provider, Identity Provider.
Main Success Scenario	1. Create account linking information between IDP and SP using scripts or programming means. The IDP/SP account linking rules need to be predefined for this to work. The outcome could be account linking files, such as LDIF, containing the persistent federation information. 2. Load the account linking files to the persistent data store on SP side. 3. Load the account linking files to the persistent data store on the IDP side.
Alternative Scenario	N/A

2.1.8 UC008 : Single Sign-on Initialized from Service Provider

Summary	Single Sign-on initialized from Service Provider.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent access Service Provider. 2. Service Provider directs User Agent to Identity Provider with authentication request. 3. Identity Provider processes the authentication request. 4. User Agent is redirected to IDP authentication URL. 5. User Agent authenticates at IDP. 6. Identity Provider generates Single Sign-on Assertion with Persistent name identifier from user account. 7. User Agent is redirected back to Service Provider using POST with the Single Sign-on Assertion. 8. Service Provider validates Assertion, and finds the local account based on the persistent name identifier. 9. Service Provider creates Single Sign-on token for the local account.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 7, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 8: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

2.1.9 UC009 : Single Sign-on Initialized from IDP (Unsolicited Responses)

Summary	Single Sign-on initialized from Identity Provider.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent authenticates at Identity Provider. 2. User Agent accesses Identity Provider URL to request Single Sign-on with selected Service Provider. 3. Identity Provider generates Single Sign-on Assertion with Persistent name identifier from user account. 4. User Agent is redirected to the Service Provider using POST with the Single Sign-on Assertion. 5. Service Provider validates Assertion, and finds the local account based on the persistent name identifier. 6. Service Provider creates Single Sign-on token for the local account.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 4, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 5: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.10 UC010: Single Sign-on with Attribute Sharing from IDP

Summary	Single Sign-on with attribute sharing from IDP.
Priority	Optional
Direct Actor	User Agent, Service Provider, Identity Provider
Main Success Scenario	1. User Agent accesses Service Provider. 2. Service Provider redirects User Agent to Identity Provider with authentication request. 3. Identity Provider processes the authentication request. 4. User Agent is redirected to IDP authentication URL. 5. User Agent authenticates at IDP. 6. Identity Provider generates Single Sign-on Assertion with Persistent name identifier from user account, the Assertion also includes AttributeStatement about user some attributes on IDP side. 7. User Agent is redirected back to Service Provider using POST with the Single Sign-on Assertion. 8. Service Provider validates Assertion, and finds the local account based on the persistent name identifier.

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	9. Service Provider creates Single Sign-on token for the local account, Attributes inside the Assertion are shared by the SP side applications. For example, those attributes could be set on the Single Sign-on token as properties accessible by the SP side applications.
Alternative Scenario	1. Above scenario is using POST profile, alternative is Artifact profile. In step 7, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 8: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol. 2. For SAML 2.0, Single Sign-on could be initiated from IDP, i.e. User Agent visits IDP first, then Assertion containing attributes pushed to SP.

2.1.11 UC011: Single Sign-on with Attribute Sharing from SP

Summary	Single Sign-on with attribute sharing from SP.
Priority	Optional
Direct Actor	User Agent, Service Provider, Identity Provider
Main Success Scenario	1. User Agent accesses Service Provider. 2. Service Provider redirects User Agent to Identity Provider with authentication request. 3. Identity Provider processes the authentication request. 4. User Agent is redirected to IDP authentication URL. 5. User Agent authenticates at IDP. 6. Identity Provider generates Single Sign-on Assertion with Persistent name identifier from user account. 7. User Agent is redirected back to Service Provider using POST with the Single Sign-on Assertion. 8. Service Provider validates Assertion, and finds the local account based on the persistent name identifier. 9. Service Provider creates Single Sign-on token for the local account. 10. Service Provider may decide to share some user attributes. It could retrieve attributes from SP side data store and pass down to SP side applications by some means. For example, attributes could be set on the the Single Sign-on token as properties accessible by the SP side applications.
Alternative Scenario	1. Above scenario is using POST profile, alternative is Artifact profile. In step 7, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 8: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol. 2. For SAML 2.0, Single Sign-on could be initiated from IDP, i.e. User Agent

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	visits IDP first, then Assertion containing attributes pushed to SP.

2.1.12 UC012: Single Sign-on with Attribute Sharing from IDP side Application

Summary	Single Sign-on with attribute sharing from IDP side application.
Priority	Optional
Direct Actor	User Agent, Service Provider, Identity Provider, Web Application.
Main Success Scenario	1. User Agent authenticates at IDP. 2. User Agent accesses IDP side Web Application. It needs to navigate to selected Service Provider, and wants to share some application attributes with the Service Provider. 3. Web Application redirects User Agent to IDP Single Sign-on service URL with attributes to be shared, for example, through HTTP POST or as query string. 4. Identity Provider processes the request and generates Single Sign-on Assertion with Persistent name identifier from user account, the Assertion also includes AttributeStatement about those attributes pushed down from the Web Application. 5. User Agent is redirected back to Service Provider using POST with the Single Sign-on Assertion. 6. Service Provider validates Assertion, and finds the local account based on the persistent name identifier. 7. Service Provider creates Single Sign-on token for the local account. Attributes inside the Assertion are shared by the SP side applications. For example, those attributes could be set on the Single Sign-on token as properties accessible by the SP side applications.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 5, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 6: ● Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.13 UC013 : Single Sign-on with J2EE Declarative Policy Integration

Summary	Single Sign-on with J2EE declarative policy integration
Priority	Essential
Direct Actor	User Agent, J2EE Agent, Web Application, Service Provider, Identity Provider

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

Main Success Scenario	1. User Agent accesses Web Application protected by J2EE agent. 2. J2EE Agent redirects User Agent to Service Provider for authentication. 3. Service Provider redirects User Agent to Identity Provider with authentication request. 4. Identity Provider processes the authentication request. 5. User Agent is redirected to IDP authentication URL. 6. User Agent authenticates at IDP. 7. Identity Provider generates Single Sign-on Assertion with Persistent name identifier from user account, the Assertion also includes an attribute about user's role on IDP side. 8. User Agent is redirected back to Service Provider using POST with the Single Sign-on Assertion. 9. Service Provider validates Assertion, and finds the local account based on the persistent name identifier. 10. Service Provider creates Single Sign-on token for the local account. The role attribute inside the Assertion is set as special SSO token property. 11. Service Provider redirects User Agent back to the Web Application. 12. J2EE Agent enforces J2EE declarative policy based on the role property on user's SSO token. 13. User Agent is redirected to Web Application if access is allowed by the J2EE Agent.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 8, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 9: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.14 UC014 : Single Sign-on with Web Agent Integration

Summary	Single Sign-on with Web Agent integration.
Priority	Essential
Direct Actor	User Agent, Web Agent, Web Application, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent accesses Web Application protected by Web Agent. 2. Web Agent redirects User Agent to Service Provider for authentication. 3. Service Provider redirects User Agent to Identity Provider with authentication request. 4. Identity Provider processes the authentication request. 5. User Agent is redirected to IDP authentication URL. 6. User Agent authenticates at IDP. 7. Identity Provider generates Single Sign-on Assertion with persistent name identifier from user account. 8. User Agent is redirected back to Service Provider using POST with the

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	Single Sign-on Assertion. 9. Service Provider validates Assertion, and finds the local account based on the persistent name identifier. 10. Service Provider creates Single Sign-on token for the local account. 11. Service Provider redirects User Agent back to the Web Application. 12. Web Agent enforces policy based on user's local SSO token. 13. User Agent is redirected to Web Application if access is allowed by Web Agent.
Alternative Scenario	Above scenario is using POST profile, alternative is Artifact profile. In step 8, an Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 9: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.15 UC015 : Single Sign-on with Authentication Context Mapping

Summary	Single Sign-on with Authentication Context Mapping.
Priority	Optional
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent access Service Provider. 2. Service Provider redirects User Agent to Identity Provider with authentication request containing specific AuthnContext, e.g. "http://www.projectliberty.org/schemas/authctx/classes/PasswordProtectedTransport" as AuthnContextClassRef. 3. Identity Provider processes the authentication request. 4. The requested AuthnContext is mapped to the pre-configured authentication scheme (e.g. Role/Service/Module based authentication), and User Agent is redirected to IDP authentication URL with the mapped authentication schema. 5. User Agent authenticates at IDP. 6. Identity Provider generates Single Sign-on Assertion with persistent name identifier from user account. 7. User Agent is redirected back to Service Provider using POST with the Single Sign-on Assertion. 8. Service Provider validates Assertion, also validates that the requested AuthnContext requirement is met, and finds the local account based on the persistent name identifier. 9. Service Provider creates Single Sign-on token for the local account, a special authentication level is set on the Single Sign-on token based on the requested AuthnContext.
Alternative	Above scenario is using POST profile, alternative is Artifact profile. In step 7, an

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

Scenario	Artifact is redirected back to Service Provider using GET, then one more step need to be performed before step 8: • Service Provider retrieves Assertion from IDP by presenting artifact using SOAP protocol.

2.1.16 UC016 : Single Logout

Summary	Single Logout
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider
Main Success Scenario	1. User Agent accesses Service Provider for Single Logout 2. User Agent is redirected to IDP Single Logout URL with Single Logout request. 3. IDP process Single Logout request. It also sends out Logout Request to any other SPs which share the same IDP session. 4. Identity Provider invalidates the IDP user session. 5. Identity Provider generates Single Logout Response for the initiating SP. 6. User Agent is redirect back to Service Provider's Single Logout return URL with the Single Logout Response. 7. Service Provider validates Logout Response, and invalidates local SP side user session.
Alternative Scenario	1. Above scenario is using HTTP redirect profile, alternative is SOAP profile. Basically SP sends Single Logout Request to the SOAP endpoint of IDP side. 2. Single Logout could be initiated from IDP, i.e. User Agent visits IDP first, then Single Logout Request sent to SP.

2.1.17 UC017 : Name Identifier Registration

Summary	Name Identifier Registration, request for new name identifier.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent accesses Service Provider to request for new Name Identifier. 2. Service Provider generates new SP Name Identifier. 3. User Agent is redirected to IDP side with request to change SP Name Identifier. 4. IDP processes the request, and saves the new SP Name Identifier in persistent data store. 5. User Agent is redirected back to Service Provider with Response.

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	6. Service Provider validates Response, and writes new Name Identifier to SP side persistent data store.
Alternative Scenario	1. Above scenario is using HTTP redirect profile, alternative is SOAP profile. Basically SP sends Name Identifier Registration Request to the SOAP endpoint of IDP side. 2. Name Identifier Registration could be initiated from IDP, i.e. User Agent visits IDP first, then Request sent to SP.

2.1.18 UC018 : Federation Termination

Summary	Federation Termination.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider.
Main Success Scenario	1. User Agent accesses Service Provider URL to request Federation Termination with selected IDP. 2. Service Provider generates Federation Termination request. 3. User Agent is redirected to IDP side with the Federation Termination request. 4. IDP processes the request, and terminates the federation with the SP, persistent data store is updated. 5. User Agent is redirected back to Service Provider with Response. 6. Service Provider validates Response, and terminates federation with the IDP, SP side persistent data store is updated.
Alternative Scenario	1. Above scenario is using HTTP redirect profile, alternative is SOAP profile. Basically SP sends Federation Termination Request to the SOAP endpoint of IDP side. 2. Federation Termination could be initiated from IDP, i.e. User Agent visits IDP URL, then Request sent to SP.

2.1.19 UC019 : IDP Proxy

Summary	IDP Proxy, this is a special Single Sign-on case where one IDP is proxying authentication for another IDP.
Priority	Optional
Direct Actor	User Agent, Service Provider, two Identity Providers.
Main Success Scenario	1. User Agent accesses Service Provider. 2. Service Provider redirects User Agent to first Identity Provider with

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	authentication request. 3. The first Identity Provider processes the authentication request. 4. User Agent is redirected to second IDP for authentication, in this case, first IDP is acting as a Service Provider. 5. The second IDP authenticates user. 6. The second IDP generates Single Sign-on Assertion with the Persistent Name Identifier between first IDP (acted as SP for this transaction) and second IDP. 7. User Agent is redirected back to first IDP using POST with the Assertion. 8. The first IDP verifies the Assertion, creates local Single Sign-on session. 9. The first IDP generates Single Sign-on Assertion with persistent name identifier between SP and first IDP. 10. User Agent is redirected back to Service Provider using POST with the Single Sign-on Assertion from the first IDP. 11. Service Provider validates Assertion, and finds the local account based on the persistent name identifier. 12. Service Provider creates Single Sign-on token for the local account.
Alternative Scenario	The above scenario is using POST binding, alternative is to use Artifact binding in step 7 and 10.

2.1.20 UC020 : Name Identifier Mapping

Summary	Name Identifier Mapping. This is for the case when one Service Provider wants to obtain Name Identifier for a principal it has federated in the name space of another service provider.
Priority	Optional
Direct Actor	User Agent, Two Service Providers, Identity Provider.
Main Success Scenario	1. First Service Provider sends SOAP request to Identity Provider to query Name Identifier for the principal with Second Service Provider. The request contains Name Identifier of the principal for the first Service Provider. 2. Identity Provider processes the Request, finds the local user account, retrieves Name Identifier for the user with the second Service Provider. 3. Identity Provider generated Response containing the Name Identifier, and returns to the first Service Provider. 4. First Service Provider communicates with the second Service Provider using the retrieved Name Identifier.
Alternative Scenario	N/A

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

2.1.21 UC021 : IDP Discovery/Introduction

Summary	IDP Discovery or IDP Introduction in case of multiple IDPs in same Circle of Trust.
Priority	Essential
Direct Actor	User Agent, Service Provider, Identity Provider, IDP Discovery/Introduction,
Main Success Scenarios	Three main scenarios: • Querying preferred IDP : SP may query preferred IDP by redirecting User Agent to an IDP Discovery/Introduction service URL. IDP Discovery/Introduction service then finds out the preferred IDP using a Common Domain Cookie, and redirects User Agent back to Service Provider with the preferred IDP as query parameter value. • Setting preferred IDP : After IDP authenticates a principal, it may redirect User Agent to an IDP Discovery/Introduction service URL with its ID as query parameter. IDP Discovery/Introduction service sets a Common Domain Cookie on the User Agent, and redirects it back to the appropriate URL. • Removing preferred IDP : After Federation Termination, IDP/SP may redirects User Agent to an IDP Discovery/Introduction service URL. The IDP Discovery/Introduction service removes corresponding Common Domain Cookie, and redirects User Agent back to appropriate URL.
Alternative Scenario	N/A

2.1.22 UC022 : Establish Trust

Summary	Establish trust among entities.
Priority	Essential
Direct Actor	Service Provider, Identity Provider, COT.
Main Success Scenario	1. Creates a COT. 2. Adds Service Provider and Identity Provider to the COT. 3. Checks the COT service to see if a remote IDP/SP is trusted.
Alternative Scenario	N/A

2.1.23 UC023 : LECP/ECP

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

Summary	LECP or ECP, this is for the case when there is a proxy sits between the User Agent/Web Application and the SP/IDP.
Priority	Optional
Direct Actor	User Agent, Web Application, LECP/ECP, Service Provider, Identity Provider
Main Success Scenario	1. User Agent, Web Application sends request to LECP/ECP. 2. LECP/ECP creates wrapped request over the original request and sends to the destination SP/IDP. 3. The SP/IDP processes the Wrapped Request from LECP/ECP. 4. The SP/IDP creates wrapped response and sends to the LECP/ECP. 5. LECP/ECP processes the wrapped response and sends unwrapped response to the originating User Agent, Web Application.
Alternative Scenario	N/A

2.2 Identity Web Services Use Cases

All use cases for Identity Web Services are based on following setup:

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

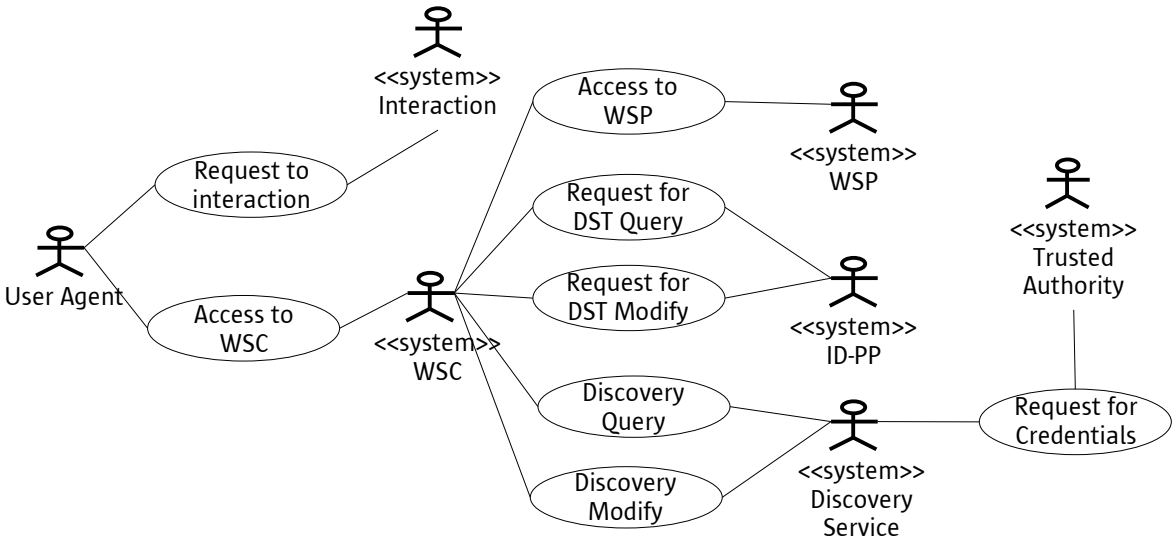

Figure 3. Identity Web Services Setup

Direct Actors:

1. User Agent	A software or device, such as browser or cellphone, which initiates requests on behalf of a Principal.
2. Web Service Consumer (WSC)	A client to access web service provider.
3 Web Service Provider (WSP)	A system entity to provide certain web services.
3. Authentication Web Service	An Identity Web Service that provides SASL based authentication service.
4. Discovery Service	An Identity Web Service to describing and discovering Identity Web Services.
5. Trusted Authority	Trusted authority to issue security credentials.
7. ID-PP	Personal Profile Service, a special type of WSP managing data service.
8. Interaction	A RedirectRequest protocol to support identity interaction.

2.2.1 UC001 : Web Service Authentication

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

Summary	Web Service Authentication.
Priority	Essential
Direct Actor	User Agent, WSC, Authentication Web Service, Discovery Service, Trusted Authority.
Main Success Scenario	1. User Agent accesses WSC. 2. WSC sends SASL request to Authentication Web Service for authentication. 3. User Agent, WSC and Authentication Web Service completes SASL authentication process. 4. Authentication Web Service creates a SASL response about the authentication done. It may contact Discovery Service to obtain bootstrapping information with optional credentials from Trusted Authority and include that as part of the SASL response. 5. WSC processes the SASL response.
Alternative Scenario	This could be authentication for the WSC itself without involvement of User Agent.

2.2.2 UC002 : Discovery Query

Summary	Discovery Query
Priority	Essential
Direct Actor	Web Service Consumer, Discovery Service, Trusted Authority.
Main Success Scenario	1. WSC needs to access certain WSP, it sends Discovery Query request to Discovery Service to find out access point for the WSP. The Query may contain credentials required to access the Discovery Service. 2. Discovery Service processes the Query. It will search for its internal registry to find the information for the WSP. It may contact Trusted Authority to generate credentials required to access the WSP. 3. Discovery Service creates Query Response containing the requested information and the credentials if any, and sends back to the WSC. 4. WSC processes the Query Response and locates the WSP information and credentials.
Alternative Scenario	N/A

2.2.3 UC003 : WSP Registry with Discovery Service using B2C model

Summary	WSP Registry with Discovery Service using B2C model, this is a normal Discovery

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	Service Modify case.
Priority	Essential
Direct Actor	WSC, WSP, Discovery Service.
Main Success Scenario	1. WSP needs to registry itself for a principal with Discovery Service. WSP constructs Discovery Modify request, containing the resource ID of the principal, and information about the WSP itself (such as endpoints, security mechanisms). 2. WSP, acting as a WSC in this case, sends the Discovery Modify request to the Discovery Service. 3. Discovery Service processes the Modify request. 4. Discovery Service sends response to the WSP/WSC.
Alternative Scenario	N/A

2.2.4 UC004 : WSP Registry with Discovery Service using B2E model

Summary	WSP registry with Discovery Service using B2E model, this is for the case where WSP have the same Resource Offering for all the principals.
Priority	Optional
Direct Actor	User Agent, WSP, Discovery Service.
Main Success Scenario	A WSP, through a User Agent, registers a template with the Discovery Service. The template will be used by Discovery Service to dynamically generate ResourceOffering for a principal when queried for the WSP.
Alternative Scenario	N/A

2.2.5 UC005 : ID-PP Query

Summary	ID-PP Query.
Priority	Essential
Direct Actor	WSC, Discovery Service, ID-PP.
Main Success Scenario	1. WSC finishes Discovery Query for ID-PP service as in UC002. 2. WSC constructs DST query request to the ID-PP service, containing the ID-PP resource ID for the principal, items to be queried and optional credentials needed to access the ID-PP service.

 Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	3. WSC sends the DST Query to ID-PP. 4. ID-PP service processes the DST Query, and retrieves data for the principal. 5. ID-PP service constructs DST Query Response with the requested data for the principal, and returns to the WSC. 6. WSC processes the DST Query Response.
Alternative Scenario	This use case is taking ID-PP as an example, it would apply to any DST type WSP, such as ID-EP.

2.2.6 UC006 : ID-PP Modify

Summary	ID-PP Modify.
Priority	Essential
Direct Actor	WSC, Discovery Service, ID-PP.
Main Success Scenario	1. WSC finishes Discovery Query for ID-PP service as in UC002. 2. WSC constructs DST Modify request to the ID-PP service, containing the ID-PP resource ID for the principal, items to be modified and credentials needed to access the service. 3. WSC sends the DST Modify request to ID-PP. 4. ID-PP service processes the DST Modify request, and modify data for the principal. 5. ID-PP service constructs DST Modify Response, and returns to the WSC. 6. WSC processes the DST Modify Response.
Alternative Scenario	This use case is taking ID-PP as an example, it would apply to any DST type WSP.

2.2.7 UC007 : RedirectRequest based Interaction

Summary	RedirectRequest protocol for Interaction.
Priority	Optional
Direct Actor	User Agent, WSC, WSP, Interaction.
Main Success Scenario	1. User Agent accesses WSC. 2. The WSC requests WSP for information. 3. The WSP returns SOAP Fault with RedirectRequest to the WSC. 4. The WSC redirects User Agent to the WSP interaction URL. 5. User Agent enters answers/consents to the interaction URL. 6. User Agent is redirected back to the WSC. 7. The WSC requests the WSP again for the same information.

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.

	8. The WSP returns response to the WSC. 9. The WSC processes response and provides service to the User Agent.
Alternative Scenario	N/A

2.2.8 UC008 : Developing and Deploying Web Service Provider

Summary	Developing and Deploying Web Service Provider.
Priority	Essential
Direct Actor	WSC, WSP, Discovery Service.
Main Success Scenario	● Open Federation provides mechanisms to make it easy to develop new Web Services Provider on top of Open Federation System. ● Open Federation provides mechanisms to make it easy to deploy new Web Service Provider on top of Open Federation System. ● Open Federation provides mechanisms to make it east to register new Web Service Provider with Discovery Service.
Alternative Scenario	N/A

Copyright © 2006 Sun Microsystems, Inc. All rights reserved.