Provisioning Middleware - Technical Design draft 04

UC IDM User Provisioning Middleware - **Technical Design**

Working Draft 04, July 08, 2011

Document identifier:

draft-ucitag-idmprovisioning-tech-design-04 (download Word document)

Technical Committee:
UC IT Architecture GroupUCTrust Work Group

Editors:
Albert Wu (albertwu@ucla.edu), University of California Los Angeles

Contributors:
Arlene Allen, University of California Santa Cruz
Dedra Chamberlin, University of California, Berkeley
Jeff McCullough, University of California Berkeley
Anthony Merriweather, University of California Los Angeles
Benn Oshrin, University of California Berkeley
Lucas Rockwell, University of California San Francisco
Dattathreya Sharma, University of California Los Angeles
David Walker, University of California, Davis
Mukesh Yadav, University of California San Francisco
Tom Zeller, University of Memphis

Abstract

This document provides a non-normative technical design overview and implementation guideline for the UC Identity Management User Provisioning Middleware.

Status of This Document

This is an incomplete ***draft*** document. Its content will change before completion, Please submit comments to Albert Wu (albertwu@ucla.edu).

Table of Contents

Introduction

This document describes an UCTrust-based middleware infrastructure that enables on-demand user provisioning among the University of California's inter-campus applications This infrastructure represents an extension to the existing Shibboleth-based UCTrust infrastructure. It complements Shibboleth's front-channel, synchronous user data provisioning capabilities with a back-channel, asynchronous and/or batch user data change notification and replication mechanism.
For the purposes of this document, user provisioning is defined to be the processes, both human and automated, that authorize (and de-authorize) people to use application systems, when those processes occur at times other than the start of an online session. This is distinguished from application systems that use a "pure" single sign-on infrastructure (*e.g.*, Shibboleth), authorizing anyone with a defined set of attributes that are provided at the start of a session.
The infrastructure described in this document supports the delivery of identity information from campus Identity and Access Management (IAM) systems to application systems. It does not account for the entire set of provisioning processes.

Background

The *UC IDM User Provisioning Middleware* is built upon the following standards:

- SAML V2.0 **[SAML2Core]**
- SAML V2.0 Change Notify Protocol V1.0 **[SSTC-SAML2-Notify-Protocol-V1.0]**

Assumptions

The middleware described in this document leverages existing UCTrust agreements, policies, processes, and technology. The document assumes the reader has detailed understanding of UCTrust technologies, policies, and operating principles.
The existing UCTrust agreements, policies, processes, and technology should be leveraged as much as possible.
This design assumes that the offices currently operating the identity management implementation within each UC campus will implement the Identity Provider Provisioning Service components. Furthermore, it assumes that the same organization operates the campus's Shibboleth Identity Provider.
This framework provides a common mechanism for application systems to obtain identity information from each campus' identity management system. Merging the results from multiple campuses, however, is left to the application.

Outline

Section 2 of this document describes the components comprising the UC IDM User Provisioning Middleware.
Section 3 introduces the message flows between components of the User Provisioning Middleware.
Section 4 provides key implementation guidance to the technical implementation team tasked to develop the UC IDM Provisioning Middleware.

UC IDM User Provisioning Middleware Components

A conforming UC IDM Provisioning Middleware implementation includes at least one Identity Provider Provisioning Agent and one Relying Party Provisioning Agent. These components interact with existing identity management components to replicate changes to a subject's identity information.

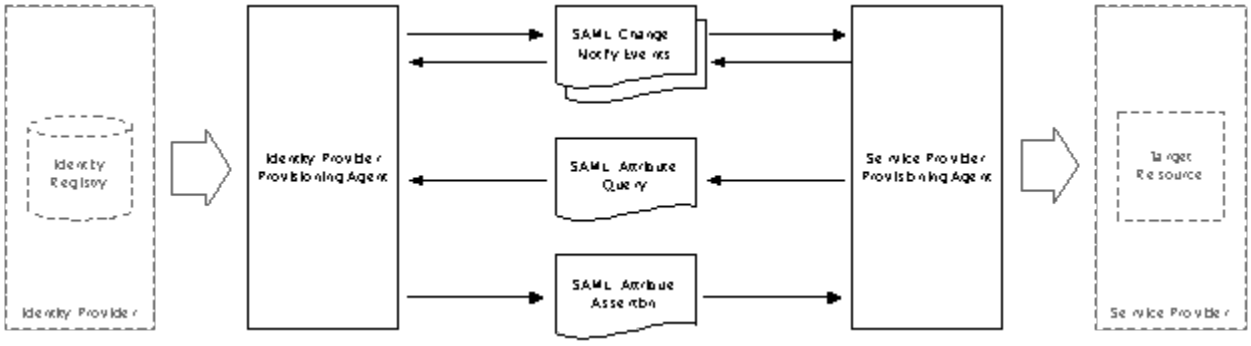


Figure 1: IDM Provisioning Middleware Components

External Components

The following identity management components interact with the provisioning components described in this document. They are not within the scope of this design.
In addition, the SAML2 Glossary defines additional identity management related terms used throughout this document. **[SAML2Gloss]**

Identity Provider

An *Identity Provider* creates, maintains, and manages identity information for *principals _and provides principal authentication to _service providers* within a *federation*.
Within this document, an Identity Provider (IdP) refers to the entire set of technical mechanisms an organization operates to manage identity information and to perform *principal* authentication. This includes, but is not limited to, the organization's Shibboleth Identity Provider.

Identity Registry

An *Identity Registry* stores identity information within an IdP. It is typically the organization's enterprise directory. Although the Identity Registry may also describe any number of programmatic interfaces an external entity uses to manipulate identity information.

Service Provider

A *Service Provider* provides services to _principals _or other system entities.
In this document, a Service Provider (SP) describes the *Target Resource* providing the service as well as the Shibboleth Service Provider components the *Target Resource* relies on to perform federated single sign-on.

Target Resource (or Application)

A *Target Resource* is the application providing services to *principals* within the *federation*. A Target Resource is likely a federated web application.

Identity Provider Provisioning Agent

An *Identity Provider Provisioning Agent* captures identity subject change events, notifies the relying party of change events, and responds to queries to retrieve identity subject data updates. Figure 2 illustrates the components within the Identity Provider Provisioning Agent (IdPPA). These sub-components are discussed in the following sub-sections.

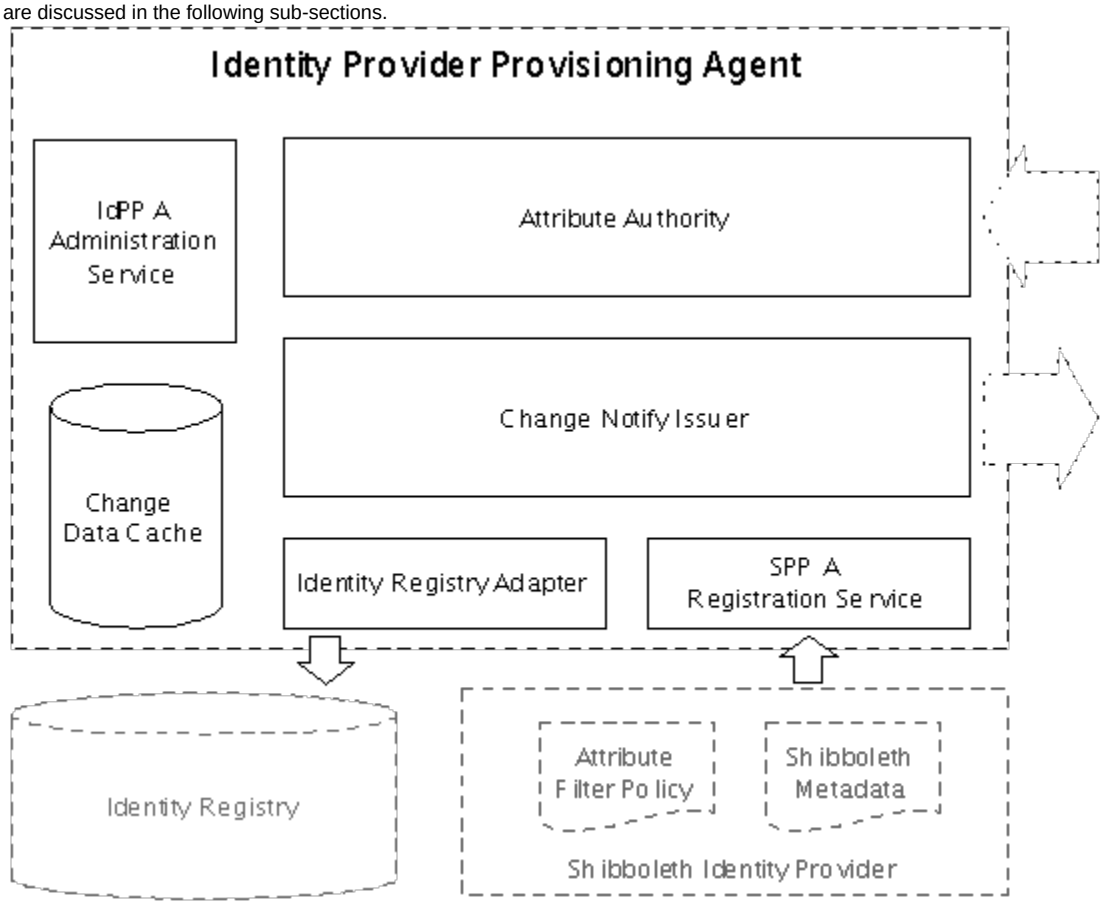


Figure 2: Identity Provider Provisioning Agent Components

Change Notify Issuer

The *Change Notify Issuer* implements the SAML 2 Change Notify Protocol. It is the IdPPA's data change detection and signaling mechanism. The Change Notify Issuer (CNI):

- Captures identity subject data changes from the Identity Registry;
- Manages pending change transactions for each registered SP Provisioning Agent;
- Composes and sends SAML Change Notify requests to registered SP Provisioning Agents;
- Process change notify responses from SP Provisioning Agents;

Identity Registry Adapter

The *Identity Registry Adapter* provides the necessary data connection and conversion processing between IdPPA and the Identity Registry. Because Identity Registries across UC differ in their design and implementation, each UC campus will likely have a different Identity Registry Adapter implementation. The Identity Registry Adapter:

- Provides the CNI with an interface to detect, query, and retrieve identity subject change events from an Identity Registry;
- Converts identity subject data from Identity Registry-specific data formats to a normalized format;

Attribute Authority

The *Attribute Authority* processes SAML attribute requests; that is, it issues SAML attribute assertions. The Attribute Authority (AA) authenticates and authorizes any requests it receives via the Service Provider Registration Service.
In implementation, the AA could be a minor variation of the Shibboleth Attribute Authority.

Service Provider Registration Service

The *Service Provider Registration Service* provides location discovery, authentication and authorization service to the CNA and the AA. The Service Provider Registration Service (SPRS):

- Accesses the Identity Provider's Shibboleth metadata and attribute filter policy to determine SP access;
- Manages the Service Provider's provisioning metadata;

Change Data Cache

The *Change Data Cache* stores all queued identity subject change events.

IdPPA Administration Service

The *IdPPA Administration Service* provides programmatic and end user facing interfaces to monitor, configure, and administer the IdPPA. The IdPPA Administration Service_'s_ key interfaces include:

- Change event monitor;
- Relying Party registration and access management;
- System startup and runtime settings;
- Change notification scheduling;
- Error and transaction log viewer;
- Identity Registry adaptor configuration;

Service Provider Provisioning Agent

A *Service Provider Provisioning Agent* listens for identity subject change notifications from registered IdPPA's, queues the change events, queries the appropriate IdPPA to retrieve the identity subject data changes, and updates the *Target Resource*. Figure 3 illustrates the components within the Service Provider Provisioning Agent (SPPA). These components are discussed in the following sub-sections.

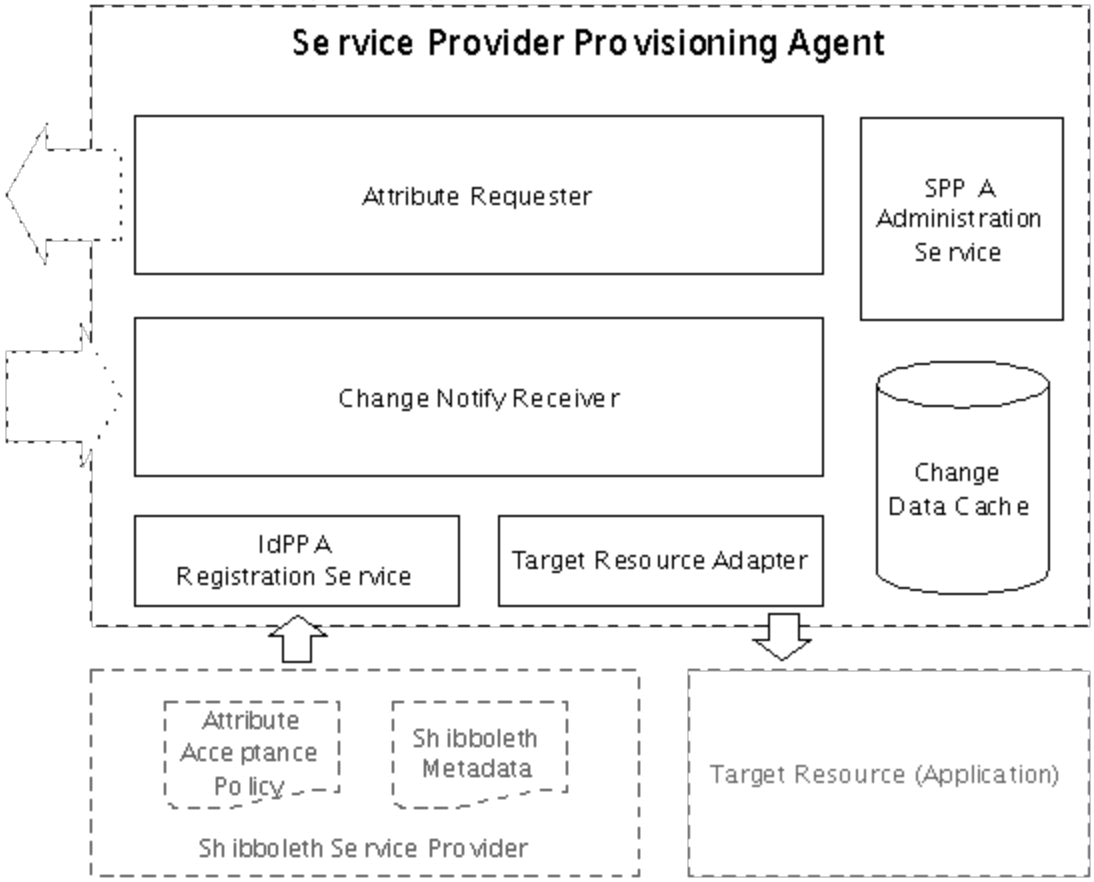


Figure 3: Service Provider Provisioning Agent Components

Change Notify Receiver

The *Change Notify Issuer* implements the SAML 2 Change Notify Protocol. It is the SPPA's data change receiving and response mechanism. The Change Notify Receiver (CNR):

- Listens for identity subject data changes from registered IdP Provisioning Agents;
- Acknowledge the receipt of identity subject change requests;
- Optionally queues the change request for batched processing;
- Signals the *Attribute Requester* to submit SAML attribute requests to update changed subject data;

Target Resource Adapter

The *Target Resource Adapter* connects to the Target Resource and pushes change data from SPPA to the Target Resource. The Target Resource Adapter:

- Converts identity subject data from the normalized format to Target Resource-specific format;
- Updates the Target Resource;

Because each Target Resource differs in its design and implementation, there will likely be multiple Target Resource Adapter implementations.

Attribute Requester

The *Attribute Requester* composes SAML attribute requests; that is, it issues SAML attribute requests. The Attribute Requester (AR) also processes the returning SAML attribute responses and queues the responses in SPPA for update to the *Target Resource*.

IdPPA Registration Service

The *IdPPA Registration Service* provides location discovery, authentication and authorization service to the CNR and the AR. The *IdPPA* Registration Service:

- Accesses the Service Provider's Shibboleth metadata to determine IdP endpoint configurations;
- Manages the IdP's provisioning configurations in addition to it's Shibboleth configuration;

SPPA Administration Service

The *SPPA Administration Service* provides programmatic and end user facing interfaces to monitor, configure, and administer the SPPA. The SPPA Administration Service_'s_ key interfaces include:

- Change event monitor;
- Identity Provider registration and access management;
- System startup and runtime settings;
- Attribute update scheduling;
- Error and transaction log viewer;
- Resource Adaptor configuration;

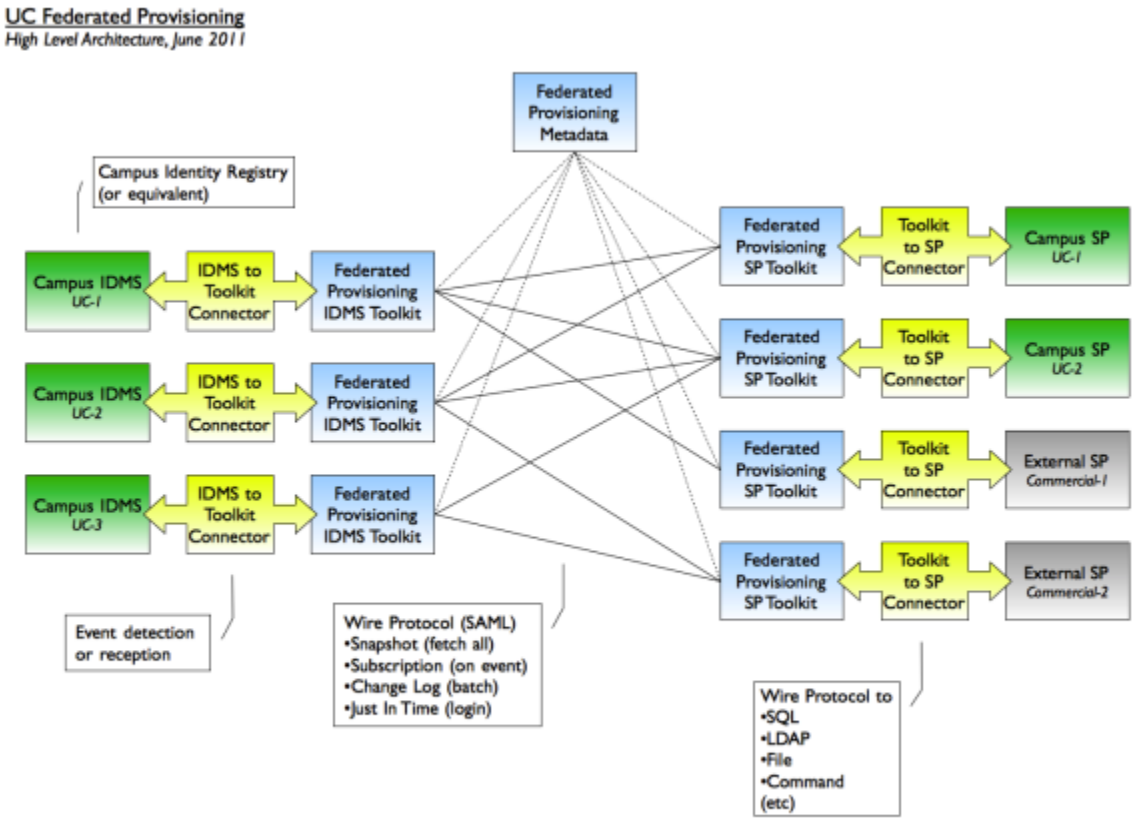
Change Data Cache

The *Change Data Cache* stores all queued identity subject change events.

Component Interactions

Operating the Middleware in a Federated Environment

The UC IDM User Provisioning Middleware operates in a federated environment, that is, there are multiple IdPPA and SPPA in a mature runtime environment. The IdPPA must have be ability to register and communicate with multiple SPPA's. Each SPPA must be able to register and communicate with multiple IdPPA's.
The IdPPA has the responsibility to track the SPPA it communicates with, including what types of data each SPPA is allowed to receive.
The SPPA has the responsibility to track each IdPPA is communicates with. It is also responsible for declaring to the IdPPA (via either registration metadata or runtime declaration) the mechanism it wishes to use to receive data.



IdP Provisioning Agent to Identity Provider Interaction

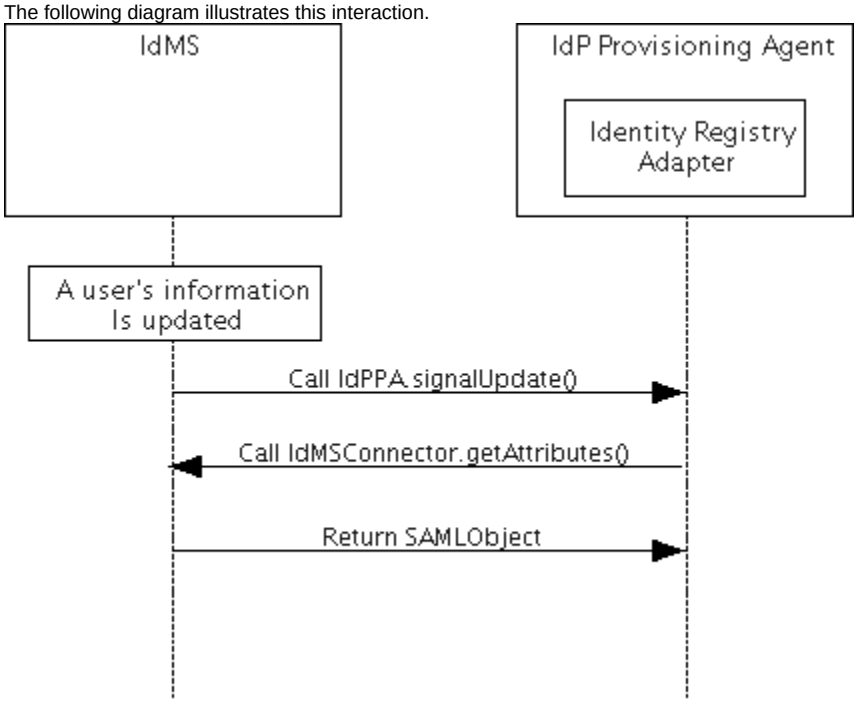
The interaction between the IdPPA and the (external) IdMS is modeled after the SAML Change Notify Protocol but does not actually require implementation of that protocol in the IdMS. The IdMS must provide implementations for the following Java interfaces:

```
// The unique identifier for individuals in the IdMS
Interface IdMSUniqueID \{
\};
\\
Interface IdMSConnector \{
\\
// Return all attributes for the individual identified by id.
// SAMLObject is provided by the Shibboleth project's OpenSAML
// distributionDoes Shib already have this?.
\\
public SAMLObject getAttributes (IdMSUniqueID id);
\\
\};
```

In addition, the IdMS must call the following IdPPA-provided method whenever user information is updated:

```
// Tell the IdPPA that the user identified by id has been updated.
\\
public void IdPPA.signalUpdate (IdMSUniqueID id) \{
\};
```

The following diagram illustrates this interaction.



SP Provisioning Agent to Target Resource Interaction

The interaction between a Target Resource and the RP Provisioning Agent depends on the specific Target Resource Adapter (TRA) selected for the Target Resource.

LDAP TRA

Interaction with the LDAP TRA is specified by version 3 of the LDAP protocol [LDAPv3]. Only read access is supported.

CSV File TRA

The CSV File TRA is configured with the location of the CSV file and the frequency with which the file is written to that location. Target Resources read the file from that location whenever they require an update to their user information.

IdP Provisioning Agent to SP Provisioning Agent Interaction

The IdP Provisioning Agent an SP Provisioning Agent bear the

Snapshot Update

The Snapshot update mechanism is intended to address the following three use cases:

- initial retrieval of user information when the SP is first put into service,
- recovery from lost transactions or other failures of the Subscription and Change Log update mechanisms, and
- on-going update of the SP's user information when the need for currency in that information is low, compared to the cost of implementing either the Subscription or Change Log update mechanisms.

In the Snapshot update mechanism, the SP initiates all transfers of user information via the Service Provider Provisioning Agent. The scheduling, therefore, is driven by the SP, but IdPs and the federation may establish service level descriptions that limit when and how often Snapshot updates can be initiated, because of the potential load that could be placed on the IdPs.

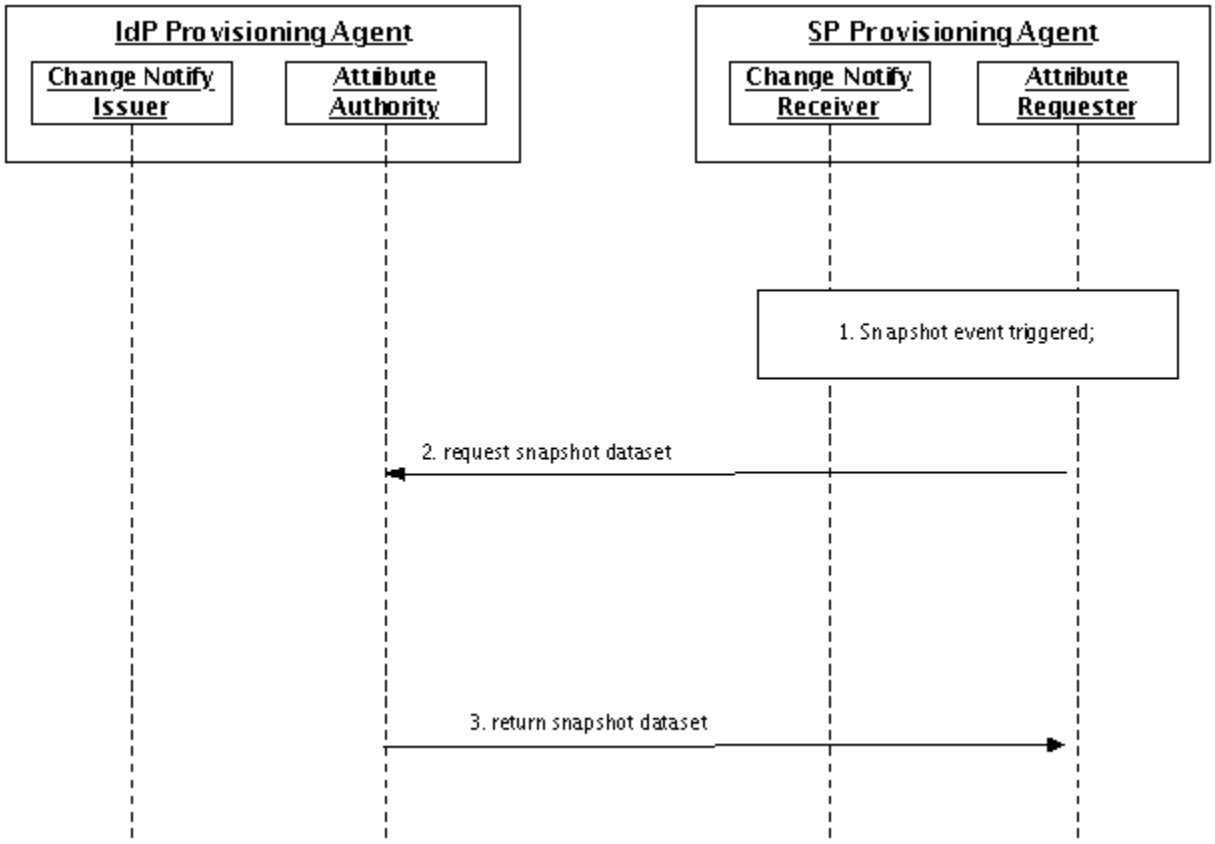


Figure 4: Snapshot Update Sequence

Subscription Update

The Subscription update mechanism is the preferred method for an SP to keep its user information current, when the need for currency is not low, compared to the cost of maintaining high availability of the SPPA (to receive the updates).
In the Subscription update mechanism, the SP initially uses the Snapshot method to initialize its user information. The SP then uses the Subscription mechanism to receive updates that occur after the Snapshot; the timing of those updates is driven by IdP events that affect user information that may be needed by the SP. Note that not all updates will be of interest to the SP; it is the SP's responsibility to apply updates according to its requirements. The federation and the IdPs will establish service level statements concerning any delays that may exists between the IdP events affecting user information and when the Change Notify Issuer signals the Change Notify Receiver.
In the event of lost transactions or other failures of the Subscription mechanism, the SP will use the Snapshot method to re-synchronize its data and then restart the Subscription mechanism.

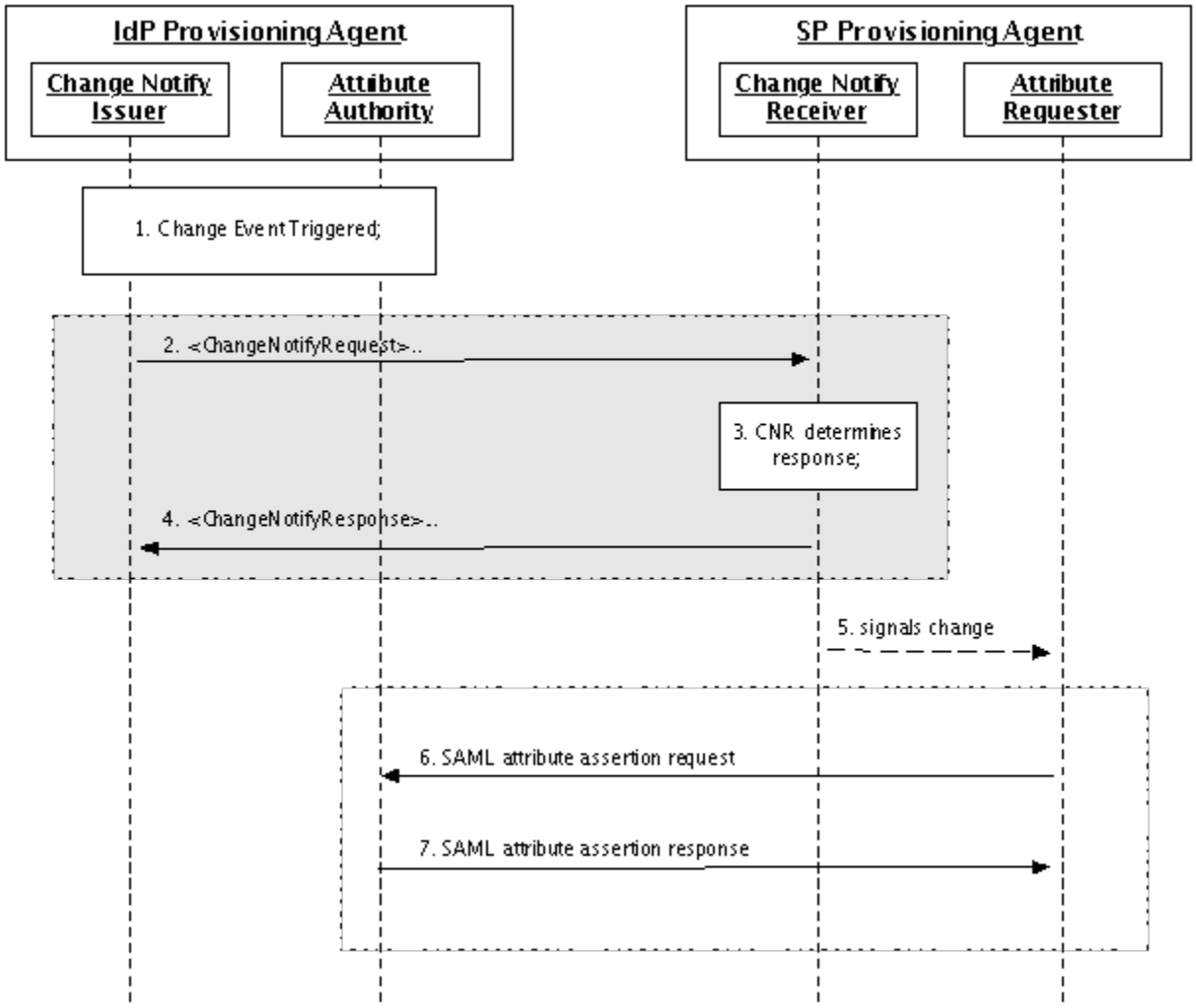


Figure 5: Subscription Update Sequence

Batched Change Update

The Batched Change (Change Log) update mechanism is the preferred method for an SP to keep its user information current, when the need for currency is not low, compared to the cost of maintaining high availability of the SPPA (to receive the updates), but there are operational or technical reasons why updates to the SP's copy of the user information should be applied in batches according to an SP-driven schedule.
In the Batched Change update mechanism, the SP initially uses the Snapshot method to initialize its user information. The SP then uses the Batched Change mechanism to receive updates that occur after the Snapshot; the timing of those updates is driven by the SP. Note that not all updates will be of interest to the SP; it is the SP's responsibility to apply updates according to its requirements. The federation and the IdPs will establish service level statements concerning any delays that may exists between the IdP events affecting user information and when the Change Notify Issuer signals the Change Notify Receiver.
In the event of lost transactions or other failures of the Batched Change mechanism, the SP will use the Snapshot method to re-synchronize its data and then restart the Batched Change mechanism.

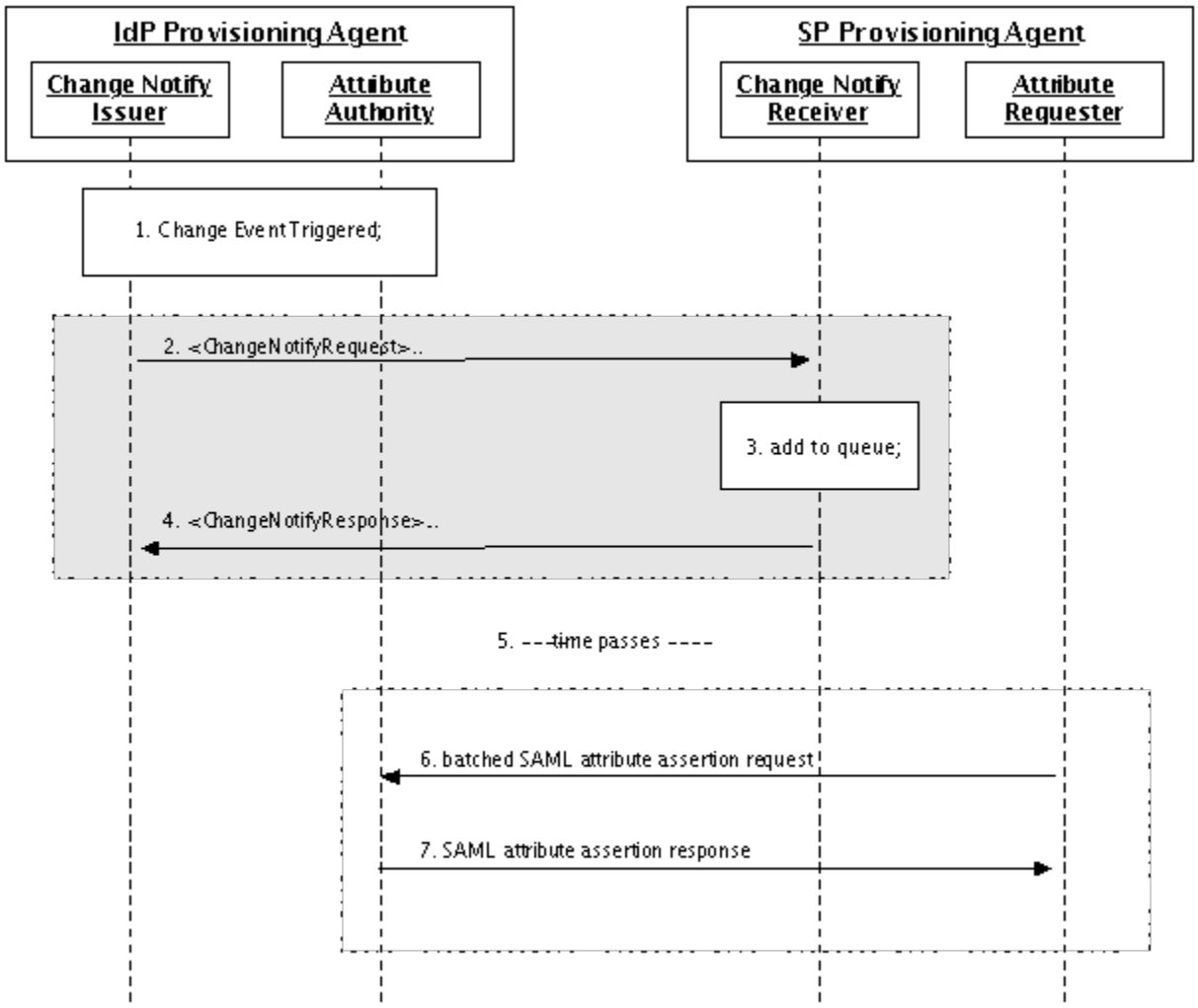


Figure 6: Batched Change(Change Log) Update Sequence

Implementation Guideline

Implementation Responsibilities

Developing, deploying, and operating the UC IDM User Provisioning Middleware is a collaborative effort. In order for the middleware to succeed, entities within UC need to work closely together. The key entities are:
UCTrust – UCTrust is <grab definition from UCTrust WG>
UC IDM User Provisioning Middleware Team – The UC IDM Provisioning Middleware Team is the implementation team (to be formed) responsible for developing the bulk of the UC IDM User Provisioning Middleware.
Identity Provider Team – The Identity Provider Team refers to each UC organization operating its Identity Management System, including Shibboleth.
Service Provider Team – The Service Provider Team refers to the UC or vendor application team responsible for the development and/or operations of an UC federated application.

UCTrust's Responsibility

The UCTrust federation is responsible for:

- The on-going development and maintenance of UC federated identity management policies and practices;

- Coordinate the deployment of the Provisioning Middleware across UC;
- Coordinating, collecting, and proposing future Provisioning Middleware feature enhancements;

The UC IDM User Provisioning Team's Responsibility

The Provisioning Middleware Team needs to:

- Design and publish detailed IdP Provisioning agent component interfaces:
  - Identity Register Adapter interface;
  - The Change Notify Issuer;
  - The Attribute Authority;
  - The Change Data Cache;
  - The SP Registration Service;
  - The IdPPA Administration Service;
- Deliver production implementation of the following IdP Provisioning Agent components:
  - The Change Notify Issuer;
  - The Attribute Authority;
  - The Change Data Cache
  - The SP Registration Service
  - The IdPPA Administration Service
- Assist each UC Identity Provider team with:
  - The development and deployment of the appropriate Identity Registry Adapter;
  - The configuration and deployment of the IdPPA at each UC location;
- Design and publish SP Provisioning agent component programmatic interfaces:
  - The Target Resource Adapter interface;
  - The Change Notify Receiver;
  - The Attribute Requester;
  - The Change Data Cache;
  - The IdP Registration Service;
  - The SPPA Administration Service;
- Deliver production implementation of the following SP Provisioning Agent components:
  - Two reference Target Resource Adapter implementations:
    - LDAPTargetResourceAdapter;
    - FileTargetResourceAdapter;
  - The Change Notify Receiver;
  - The Attribute Requester;
  - The Change Data Cache;
  - The IdP Registration Service;
  - The SPPA Administration Service;
- Demonstrate the successful deployment and operation of the UC IDM User Provisioning Middleware among at least two UC Identity Providers and three Service Providers;

The Identity Provider Team's Responsibility

Each Identity Provider Team needs to:

- Understand the Provisioning Middleware's implementation requirements;
- Work with ITLC and its CIO to develop funding and implementation plan;
- Secure the appropriate development resources to develop the IdP-specific Identity Registry Adapter;
- Work with the Provisioning Middleware team to integrate the Provisioning Middleware with its Identity Management operations;
- Work with any applicable federated Service Providers to provision user identity data via the Provisioning Middleware;

The Service Provider Team's Responsibility

Each Service Provider Team needs to:

- Understand the Provisioning Middleware's implementation requirements;
- Deploy and operate the SP Provisioning Agent for its Target Resource;
- Optionally develop custom Target Resource Adapters if neither reference adapters work for its Target Resource;
- Work with UC Identity Providers to provision user identity data via the Provisioning Middleware;

Implementation Technology

The UC IDM User Provisioning Middleware Should Be Implemented in Java.

The UC IDM User Provisioning Middleware is designed to work with SAML 2 compliant Identity Providers and Service Providers. Within UC, Shibboleth is the standard federated SAML SSO implementation.
The implementers of this document's design should leverage the same Java implemented Shibboleth/SAML code whenever appropriate.
Furthermore, a Java-written solution is easier to deploy across multiple operating systems.
Finally, it is anticipated that IdPOs will operate SPPAs as a service on behalf of SPs. The choice to implement the UC IDM User Provisioning Middleware in Java aligns with the fact that the Shibboleth IdP is also implemented in Java.

Data within and exchanged with the UC IDM User Provisioning Middleware must meet specific authentication, authorization, and encryption requirements.

The IdP Provisioning Agent must authenticate all data request to ensure that only a registered SP Provisioning Agent may connect;

We assume that authentication will be done in the same manner an Shibboleth IdP authenticates attribute requests from an Shibboleth SP.

When responding, the IDP Provisioning Agent must ensure that it only returns data a registered SPPA is authorized to access;

We assume that data authorization control will be done in the same manner an Shibboleth IdP filters attribute responses.

The SP Provisioning Agent must authenticate all change notify requests to ensure that only a registered IdP Provisioning Agent may connect;

We assume that authentication will be done in the same manner an Shibboleth IdP authenticates attribute requests from an Shibboleth SP.

All communication between an IdP Provisioning Agent and a SP Provisioning must be encrypted;

We assume that encryption between an IdPPA and a SPPA will reuse the same encryption mechanism used between a Shibboleth IDP and a Shibboleth SP.

While at rest, data managed by the Provisioning Middleware should be appropriately protected from unauthorized access.

The information managed by the Provisioning Middleware is likely sensitive and must be protected from unauthorized access. The Provisioning Middleware Team must evaluate each data store's (e.g., the change data cache, the file generated by the FileTargetResourceAdapter, etc.) technical capabilities and recommend the appropriate encryption and/or access control mechanism.

The IdP Provisioning Agent must authenticate all data change events to ensure that only the registered Identity Provider may connect;

To avoid an unauthorized agent from injecting illegal data change events, the IdPPA must authenticate data change events from its Identity Provider. The authentication mechanism used must be at least as strong as the one used between the IdPPA and the SPPA.

All user identity data exchange between an IdP Provisioning Agent its Identity Provider should be encrypted;

If the Identity Provisioning Agent will be exchanging user identity data with its Identity Provider over a network, the exchange must be encrypted. This means that the Provisioning Middleware team must implement the appropriate encryption mechanism so that the identity management operators can enable it at runtime.

All user identity data exchange between an SP Provisioning Agent its Service Provider should be encrypted;

If the SP Provisioning Agent will be exchanging user identity data with its Identity Provider over a network, the exchange must be encrypted. This means that the Provisioning Middleware team must implement the appropriate encryption mechanism so that the service provider operators can enable it at runtime.

Identity Provider Provisioning Agent

Design Directives

The Identity Provider operators are members of the development team.

{insert explanation here}

The IdP Provisioning Agent extends an organization's existing Identity Provider implementation. It does not replace it.

{insert explanation here}

The IdP Provisioning Agent SHOULD NOT consolidate identity information from multiple sources.

The Identity Provider SHOULD consolidate identity information in the Identity Registry prior to interfacing with the IdP Provisioning Agent, i.e., managing multiple identity data sources is beyond the scope of this implementation.

The IdP Provisioning Agent SHOULD be able to run on separate machines from the Identity Provider.

{insert explanation here}

The IdP Provisioning Agent MUST be able to run in a high availability configuration.

{insert explanation here}

The IdP Provisioning Agent SHOULD use the IdP's Shibboleth metadata and attribute filter policy to manage a Service Provider's access to subject identity data.

{insert explanation here}

Expect to track multiple SP's

{insert explanation here}

Target Resource Adapter Implementation

There will be two initial Target Resource Adapters (TRAs) provided as part of the UC IDM User Provisioning Middleware,

- LDAP - The LDAP TRA will implement a read-only LDAP server containing the users and attributes required by the SP.
  - The OIDs used to identify attributes will be those defined by the federation, or others mutually agreed upon by the SP and affected IdPs.
  - The user entries will be created under a top-level branch of the server's directory tree called People.
- CSV File - The CSV File TRA will provide user information in a "CSV" file format.
  - The first record in the file will contain the OIDs of the attributes provided for each user, and the user information itself will follow, one record per user. All non-numeric values will be enclosed in quotes.
  - The OIDs used to identify attributes will be those defined by the federation, or others mutually agreed upon by the SP and affected IdPs.
  - When there are multi-valued attributes, the maximum number of occurrences of those attributes will be defined by the federation, or by mutual agreement between the SP and affected IdPs.

Both TRAs can be configured for the update mechanism (Snapshot, Subscription, Batched Change) they will use in interactions with IdPs. It should be noted, however, that neither of these methods include a mechanism for signaling the SP when changes occur. They simply provide the most current information available to the SPPA.

Interacting with the SP Provisioning Agent

{TBD}

Notes

{TBD}

Service Provider Provisioning Agent

Design Directives

Service Provider operators are product consumers (end users).

{insert explanation here}

Minimize SP's need to develop custom integration code.

{insert explanation here}

Design for hassle-free installation and operation.

{insert explanation here}

Expect to track/manage multiple IdP's

{insert explanation here}

Interacting with the Target Resource

For the CSV File TRA, the files must be cryptographically signed and encrypted to ensure authenticity of the SP Provisioning Agent and the Target Resource.

Encrypting and signing data files addresses the requirement for encryption and signing of communication among and with UC IDM User Provisioning Middleware components.

Interacting with the IDP Provisioning Agent

{TBD}

Notes

{TBD}

References

[SSTC-SAML2-Notify-Protocol-V1.0] *SAML V2.0 Change Notify Protocol Version 1.0*. 22 February 2011. OASIS Committee Specification Public Review Draft 01. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd01/sstc-saml2-notify-protocol-v1.0-csprd01.odt
[SAML2Bind] OASIS Standard, _Bindings for the OASIS Security Association Markup Language (SAML) V2.0. _March 2005.http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf
[SAML2Core] OASIS Standard, _Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0._March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
[SAML2Meta] S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0.* OASIS SSTC, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf
[SAML2Prof] OASIS Standard, *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.* March 2005.http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
[SAML2Gloss] OASIS Standard, *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 2005. http://www.oasis-open.org/committees/download.php/21111/saml-glossary-2.0-os.html
[LDAPv3] *Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map (RFC 4510).* http://tools.ietf.org/html/rfc4510

Document History