User Provisioning Design

User Provisioning Design

This document provides a description of a UCTrust-based infrastructure to support user provisioning for inter-campus applications within the University of California. This infrastructure represents an extension to the existing Shibboleth-based UCTrust infrastructure to address use cases, such as those described in User Provisioning Use Cases.

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 will support the exchange of identity information from campus Identity and Access Management (IAM) systems to application systems, not the entire set of provisioning processes. The Roles and Responsibilities section below describes where those other provisioning processes should be implemented.

While UCTrust is the first intercampus use of middleware in the University of California, this project is UC's first use of middleware as an application development paradigm. The infrastructure described is specific to the exchange of identity information for user provisioning. It does, however, embody many aspects of a more general-purpose infrastructure for data interchange among arbitrary systems that should be useful in the future.

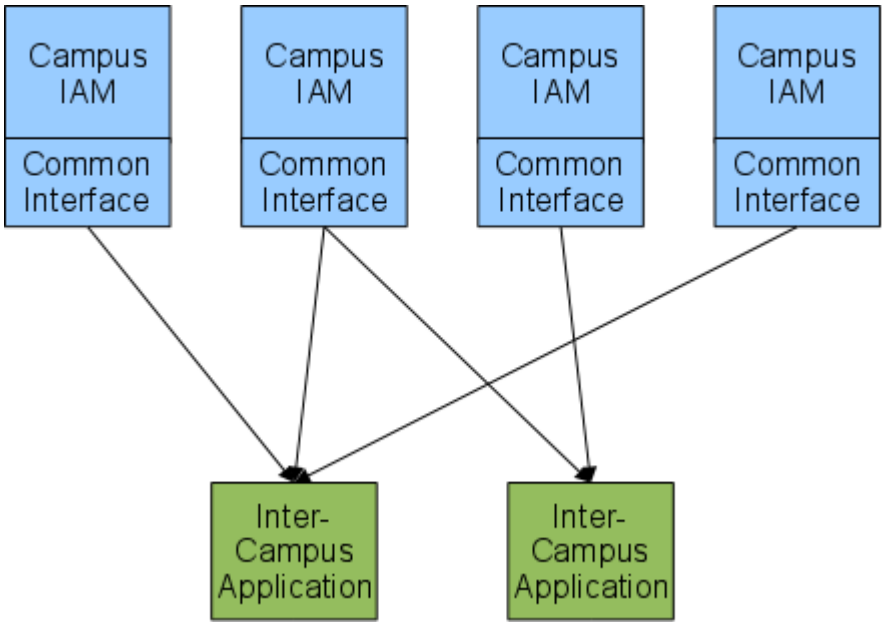
Principles and Assumptions

- Campus identity and access management systems and the organizations that operate them are authoritative for information about the members of their respective communities. The same campus organization that currently operates Shibboleth will be the organization that operates the infrastructure described in this document.
    - (Note that much of the IAM's information will likely be aggregated from other systems of record on the campus.; Nevertheless, UCTrust designates the IAM as the authoritative contact for its campus.)
- As the focus on UC-wide service provisioning grows, there will be a corresponding expansion in the number of attributes which need to be released within the UCTrust federation. This will require stronger partnerships and governance agreements between IDMS organizations and data proprietors on each campus.
- This framework provides a common mechanism for application systems to obtain identity information from campus IAM systems. Merging the results from multiple IAM systems, however, is left to the application.
- The existing UCTrust agreements, policies, processes, and technology should be leveraged as much as possible. All participating campuses have implemented UCTrust and are operating a current version of Shibboleth.
- The design and implementation must make effective use of University resources. Where possible implementations should be shared and/or reused.
- Integrations will require effort on the part of University IDMS and Service Providers. To the extent possible, the complexity of integrations should fall to the IDMS to keep the barriers to entry as low as possible for Service Providers.
- Standards for user provisioning are evolving rapidly and user provisioning design should be as flexible and adaptable as possible.
- Deployment plans should accommodate differing priorities and schedules at different campuses, allowing for inter-campus collaboration and partial implementations at each campus until the entire infrastructure is deployed.
    - This effective use of University resources extends beyond this project, in particular by being the first UC-wide deployment of common middleware that can be used by other projects in the future.

Design Diagrams

High-level design:

The following diagram illustrates the high-level design of this infrastructure for two applications that retrieve identity information from four campuses.



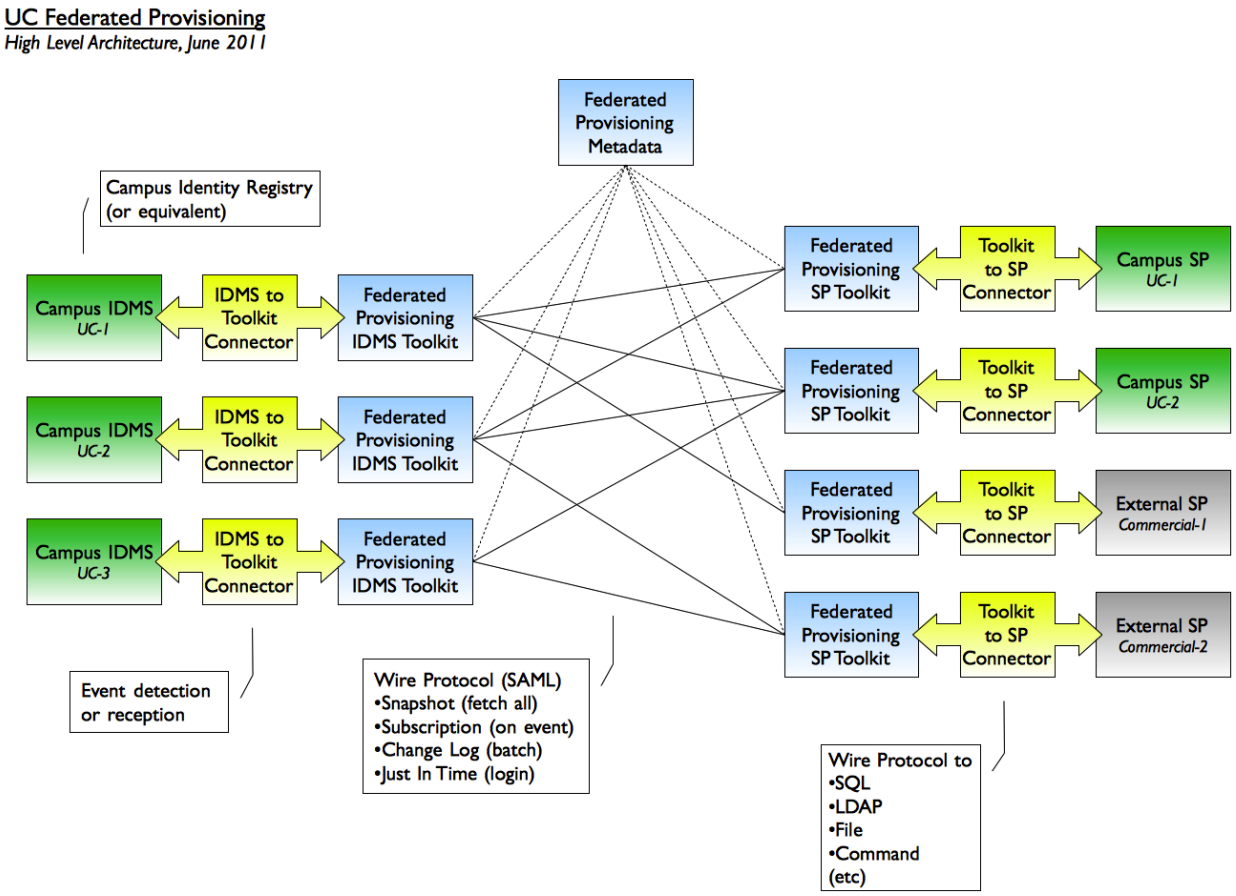
Just as with Shibboleth in UCTrust,

- Inter-campus applications obtain identity information about their users from IAMs through the use of standard network protocols and formats.
- All IAMs and inter-campus applications have unique names, called entityIDs, that are the same as those assigned for Shibboleth IdPs ("Identity Providers") and SPs ("Service Providers"), respectively.
- IAMs control the release of information to service providers through the use of *Attribute Release Policies*, which specify which identity attributes should be released to an application. In the case of user provisioning, however, the application's SP name will also determine the users for which the IAM will release those attributes.
- Software will be provided, written in Java, for integration into each campus IAM to implement the standard protocols and formats.

The following types of access will be supported. Other than SSO Event (Shibboleth), they will be supported by the Common Interface:

- **Snapshot**. All identity information allowed by the attribute release policy will be transmitted to the application.
- **Subscription.** Identity information will be transmitted to the application as add, delete, and update transactions on an event-driven basis. The transactions sent will be those that have occurred (or will occur) since the last Snapshot, Subscription, or Change Log access.
- **Change Log.** All add, delete, and update transactions that have been generated since the last Snapshot, Subscription, or Change Log access will be transmitted.
- **SSO Event.** Identity information about the current user is transmitted at the start of a session. This is the existing Shibboleth access type.

Detailed design:



Data Release and Governance

The first principle in this document is "Campus identity and access management systems and the organizations that operate them are authoritative for information about the members of their respective communities. The same campus organization that currently operates Shibboleth will be the organization that operates the infrastructure described in this document." (See Principles and Assumptions above.) In many cases, however, the organizations that operate the campus identity and access management (IAM) systems are not the ultimate proprietors of the data in their systems, so the IAM operators must represent the data release policies of those proprietors.

We also have the following principles:

- As the focus on UC-wide service provisioning grows, there will be a corresponding expansion in the number of attributes which need to be released within the UCTrust federation. This will require stronger partnerships and governance agreements between IDMS organizations and data proprietors on each campus.
- The existing UCTrust agreements, policies, processes, and technology should be leveraged as much as possible. All participating campuses have implemented UCTrust and are operating a current version of Shibboleth.

It is already the case that IAM operators aggregate data for UCTrust, but this User Provisioning project represents a significant expansion of that role. It also represents an expansion of the UCTrust Work Group's role of defining interoperable names and formats for identity attributes.

IAM operators need to ensure that the appropriate organizational relationships are in place to enable the IAM operator to aggregate data from multiple source systems, such as payroll and student information systems, so that decisions about the release of identity attributes to service providers can be made in an effective manner.

Roles and Responsibilities

IAM Responsibilities

- Accuracy and currency of identity information
- Maintenance of identity attributes to enable selection of the users to transmit to each authorized application

- Implementation of Grouper, the Internet2-sponsored open source group management system, to facilitate a common interface for specifying the users of intercampus applications throughout UC.
    - Individual campuses may propose alternatives to Grouper for implementation at their site.
- Implementation of an unchanging and unique identifier for all identity records sent to a specific application.
    - eduPersonTargetedID should be considered for this during the detailed design phase of the project.
- Deployment and operation of the Common Interface, as well as the Shibboleth interface
- Deployment and operation of the middleware that will be utilized by the Common Interface
    - Kuali Rice should be considered for the middleware during the detailed design phase of the project.
- The process for approving attribute release policies

Application Administrator Responsibilities

- Implementation of provisioning interfaces for the application
- Implementation of appropriate protections for the identity information received

UCTrust Responsibilities

- Unique naming of all IdPs (IAMs) and SPs (inter-campus applications), as is already done for Shibboleth
- Other UCTrust operational responsibilities, such as identification of support contacts, maintenance of logs, *etc.* These are described in UCTrust University of California Identity Management Federation Service Description and Policies.

Technical Implementation

For information included in the original design conversations regarding SP, IDMS and Interchange, see the the Archived User Provisioning High-Level Design Docs

- IDMS-side functions
    - Accept event signals from IDMS
    - Retrieve group membership for SPs.
    - Retrieve attributes according ARPs
    - Manage pending transactions / snapshots for each SP
    - Put transactions / snapshots onto the wire
- SP-side functions
    - Retrieve transactions / snapshots from the wire
    - Perform any necessary transformations (?)
    - Deliver transactions to provisioning engine

Related Efforts in Higher Education

- It should be noted that Internet2's COManage project is complementary to this project, as it focuses on authorizing and provisioning members of a Virtual Organization for LDAP-enabled applications. While it does include primitive exchange of user identity information via nightly LDAP queries, we believe COManage would benefit from our work on the exchange of identity information. Also, COManage provides an off-the-shelf solution for LDAP-enabled applications that can be leveraged within UC. Assuming implementation is approved for this project, potential collaboration with COManage should be pursued.

Technical Implementation Thoughts

Wire Protocols

SCIM

- Lots of momentum in industry
- Still immature
- Elegant in it's simplicity
- Wouldn't be able to deliver range if data needed without significant further development
- We could influence the course and pace of it's maturation

SPML

- Not much uptake/active development in industry, with the exception of Oracle which relies heavily on SPML

SAML

- Well-known in higher Ed
- Higher Ed reps actively participate in standards bodies
- Mature enough to deliver wide range of attributes right away
- Relatively little interest from biggest industry players

Comparison

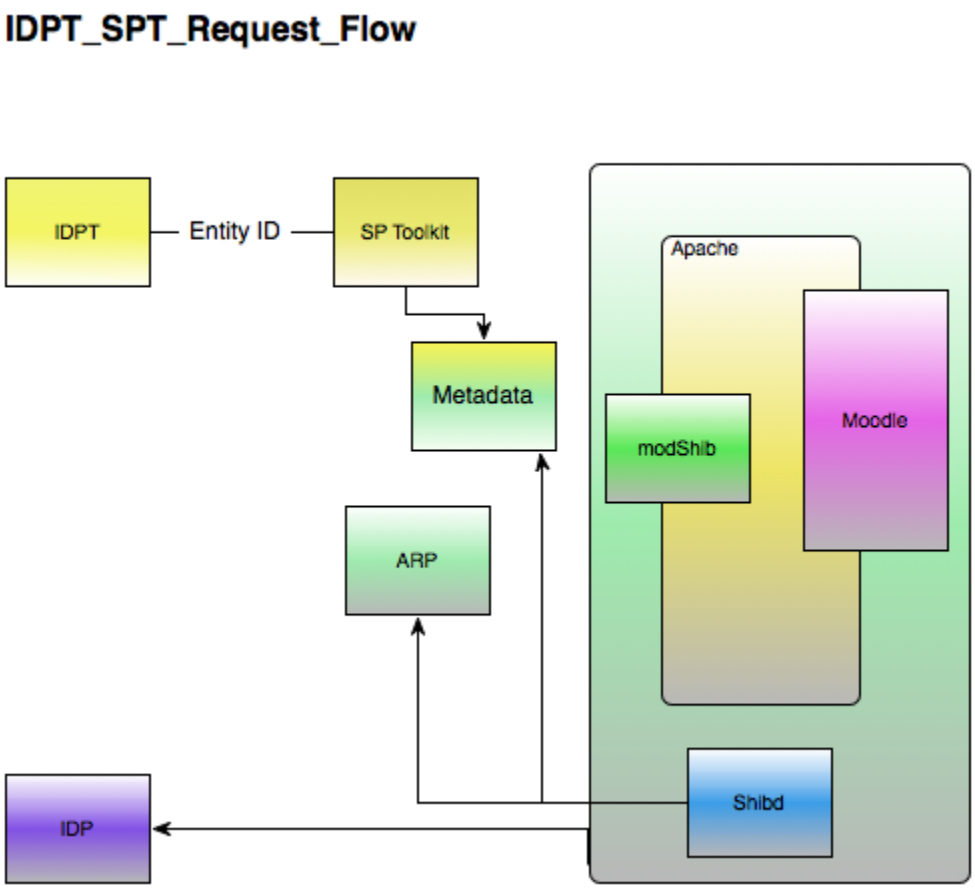
- The group compared different options on different criteria:

UCOP-Trappist-Magic-Quadrant-2.pdf

Chosen Protocol

For this project, the group has chosen SAML for the wire (the "mesh" in the Detailed Design diagram) protocol. This means that the IDMSTK and the SPTK will use SAML for communication. SAML was chosen because it is already used by Shibboleth, and with the advent of the Change Notify protocol, it was seen as the best option in terms of meshing with current infrastructure/processes.

Sample Request Flow



IDMS Toolkit

The IDMS Toolkit (IDMSTK) is a program which accepts requests from the various SPTKs (see SPTK section, below) for the purposes of account provisioning in a service provider. There is only one IDMSTK per institution, where there could be *n* SPTKs. The IDMSTK processes basic requests sent from the various SPTKs, and in turn, looks into the institution's local IDMS to fulfill the request. It is possible that not every institution's IDMS will be able to respond to all of the requests.

The IDMSTK will be able to answer the following types of requests:

- Get all of the changed IDs since the given time: `getChangedSubjects(Time t)`
- Get all changes for the given subject since the given time: `getChangesForSubjectSinceTime(SubjectID id, Time t)`
- Get current state of the given subject: `getSubject(SubjectID id)`
- Get the current state of everyone: `getAll()`

> **Note**
>
> The second bullet above is not 100% clear to me, as I don't think we can expect an IDMS to be able to relay all changes for a given person from a given point in time. So, if someone can clarify this one, that will be great. - Lucas Rockwell

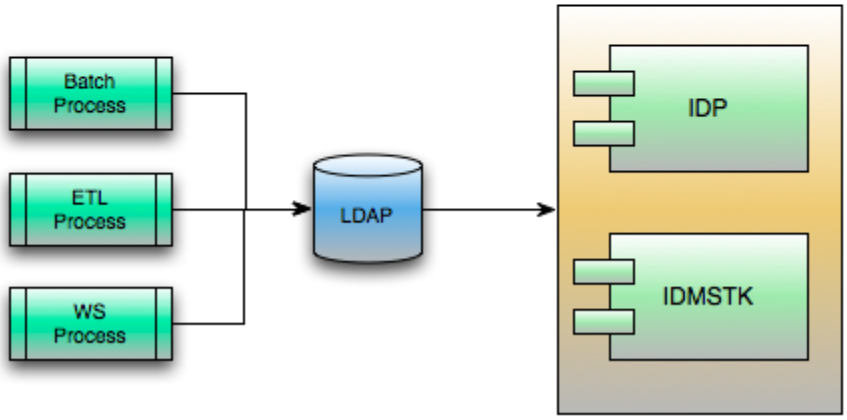
The requests outlined above will be performed over the wire using SAML (see reason for this in the Wire Protocols section, above).

The IDMSTK is comprised of the following:

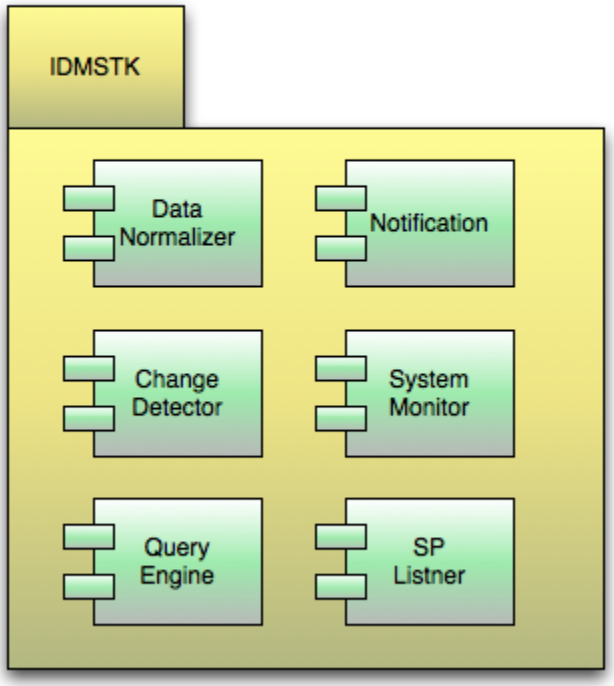
- Query Engine
- ARP
- Notifications
- Config
- Change Detector

- System Monitor
- Listener

IdP with ToolKit



IDMSTL



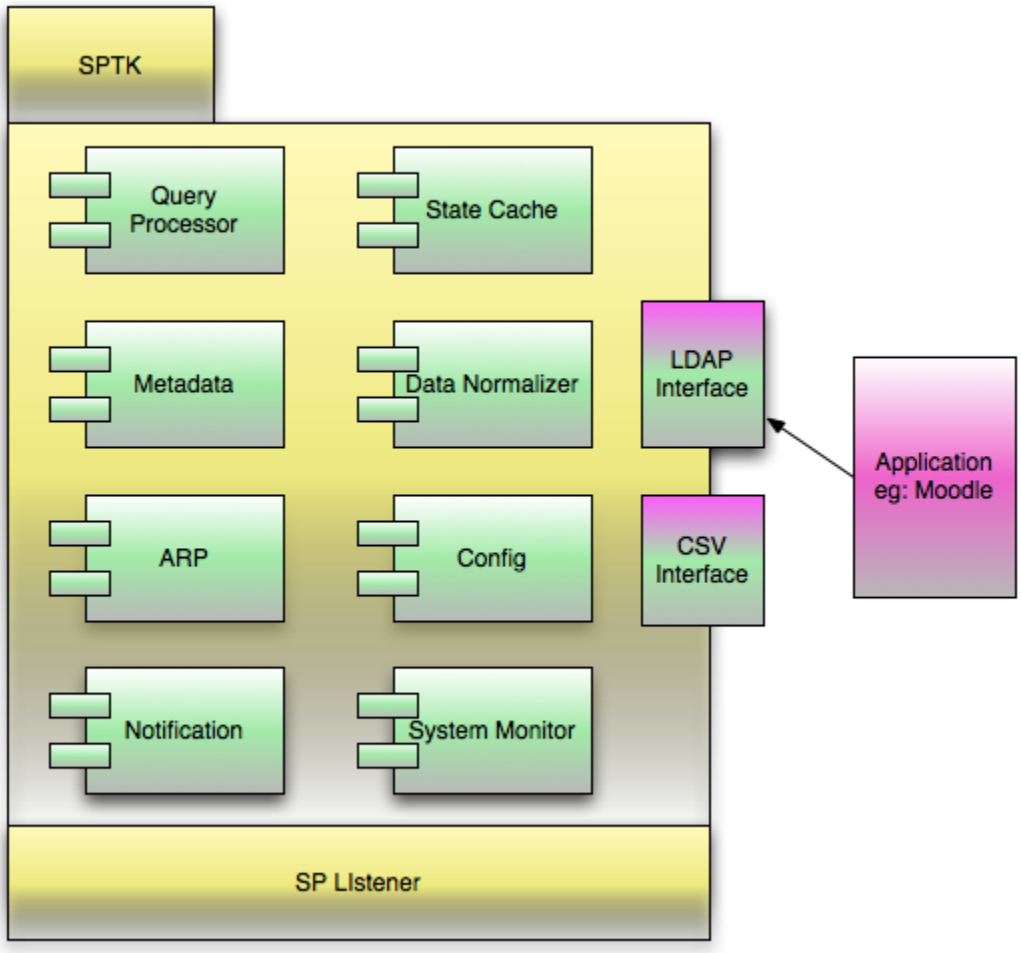
SP Toolkit

The SP Toolkit (SPTK) is a tool which will allow a local service, Moodle in the example above, to pull in data from multiple sources as if it were only talking to one source. For instance, Moodle can be configured to pull provisioning information from a single LDAP instance, so in this case, the SPTK will allow Moodle to be configured so that it pulls provisioning data from LDAP, but that LDAP is actually the SPTK, and the SPTK in turn pulls in provisioning information from each UC's IdPTK.

See the IDMSTK section above for a list of the types of request that the SPTK should be able to handle from the service provider.

The SPTK is comprised of the following:

- SP Interface – The interface used by the service which will use (read, query) the SPTK. In the case of Moodle, the Interface will be LDAP. As mentioned above, the SPTK can only handle basic query processing, so this is not a full-featured LDAP interface.
- Query Processor – Will take the native query from the service and translate it into a SAML request that the IDMSTK will be able to understand.
- Metadata – Metadata for the various institutions in the trust relationship for this SPTK. This allows the SPTK to know where the IDMSTKs are located.
- ARP --
- Notifications
- State Cache
- Data Normalizer
- Config
- System Monitor
- Listener



Related Links

- InCommon Camp
- Internet2
- InCommon