CIFER Provisioning and Integration Team Meeting, Tuesday, April 1, 4:00 pm Eastern, 1:00 pm Pacific

Dial-in numbers:
+1-734-615-7474 (Please use if you do not pay for Long Distance),
+1-866-411-0013 (toll free US/Canada Only)
Access code: 0150432#

Paired with an Adobe Connect Meeting for screen sharing:
http://wisc.adobeconnect.com/cifer-prov/ ⇐ open for presentation

Participants: KeithH, JimmyV, GaryW, VincentR, RobC, JonathanP

AGENDA

1) Discuss provisioning models to be supported in the first demos on the IAM Testbed, our platform for demonstration solutions with open source IAM components

- What other scenarios might be good targets?
    - Getting people in: Self-registration; Invitation-based; Social identity provider-driven;
    - SoR-driven case
        - RobC: OpenIDM has the flavor of connecting lots of endpoints (like the 18-armed postal clerk in MiB) could that help with getting multiple SoRs (Banner, PS,...) -to- CPR;
        - Vincent: We are using OpenIDM as a “postman” in just this sense
        - JimmyV: right now Penn State is feeding HR and student data to CPR using the same batch feeds that were used by the old IdM system; it has complete knowledge of all the identities known at PSU; work started in Jan on a new student system (PplSoft: will take 2 years); New HR implementation coming in the fall, so Penn State will need solutions to this set of problems
        - Gary: U of Az: student and HR implementations; our Identity Registry is still mainly batch driven, with some near-real-time integrations; biggest hurdle: vendors are way behind the curve: PS can expose event endpoints accessible via SOAP, but that’s not ideal
        - RobC: PS is promising a REST interface as is SAP; We’re hoping….
        - One worthy goal: come up with guidance on strategies to deal with impedance mismatch between CPR and other up-to-date identity registries and the (lagging) state of the art with commercial SoRs re event-driven messaging support
- Candidate scenario #1: Managing access of a consultant on a university IT project
    - Project Manager (PM) invites Consultant (C) to self-register for local acct.
    - C registers and gets local credentials ⇐ KeithH just tested this w CPR instance

    - PM creates a project team in Grouper
    - PM assigns permissions to a wiki space and a github repository to team members
    - Github organizational team is given read access to code repo
    - Project team membership provisioned to LDAP
    - Github organizational team membership managed via github API
    - Confluence configured to draw group memberships from LDAP

- CPR events -to- AMQ -to- custom connector code (already demo-able with OS CPR) -to- LDAP
    - Note the AMQ consumer code (DirectoryMessageProcessor.java in package edu.psu.iam.cpr.provisioner.directory) that makes the actual add and modify calls on LDAP
- Provision from Grouper hook (a la UDub) to LDAP
    - One each for AD, Radius, Google
- Grouper hook to also fire AMQ message -to- Camel-based connectors -to- github team member mgmt API
    - Camel connector to make RESTful calls to github API
- GaryW: Is there a consensus on the approaches to messaging architectures? KeithH: No, certainly not yet; As a community we're very much in the experimental phase
- JonP: Challenges with consumers: some are wedded to particular solutions; Some Us are using different solutions and may not want to change; We encrypt message body (with the change details) so our internal consumers with the decryption key can read the body contents; some consumers would just use as a notification service: You'd know something changed on a particular group; consumers can use that to look up the info if they have the right permissions in Grouper

- Do we have a sane environment for building demos?
    - Not too demanding on limited time people have to admin the pieces
    - Not too hard to recover from botched experiments (back to known good states)
    - Gary: Virtual private cloud env. in beta mode with campus with a self-service portal so campus folks can create VMs, AWS-like environments; snapshot/rollback capabilities; EC2, vmware, vsphere; Might be able to host an OpenIDM instance if someone else had the cycles to install and configure it;

2) Anyone with cycles/resources to create an instance of OpenIDM for use in demos?

- VincentR: hired for IdM project about a year ago; which product to purchase to replace their home-grown solution; spent 1 year in evaluation of commercial and open source products; there are commercial products that could solve most of their current problems, but there are questions about their evolution WRT the university needs; led them to look more seriously at open source OpenIDM (with commercial-like support option); running on total of 9 VMs; Beginning implementation process; target: summer; UQ/M has ~ 45K students, 10K staff;

    - IdReg issues: we have AD, another system with student accounts, staff accounts; creating a new LDAP where each person has one account; OpenIDM pulls from HR, Registrar, push into AD and our new “enterprise” directory, CAS and Shib will point at that instance;

- Explore feasibility of developing OpenICF-compliant connectors
    - Using off-the-shelf connectors for this phase; we have an event queuing solution; may need to develop something with the OpenICF framework; OpenIDM has REST/HTTP interfaces; plan to build a bridge from that to Business Process Management (BPM), services there (e.g. onboarding new employee workflows;
    - Software itself is not made to connect to systems like PS, etc. You have to configure for each system; it IS possible;
- Background information on provisioning to LDAP, AD, Kerberos from cifer-prov email list:
    - RobC: We're provisioning into all three (with multiple LDAPs, each getting different sets of attributes), and we're in the process of rewriting our whole provisioning engine (which is currently tightly coupled to and essentially embedded in OIM) to be driven by Grouper and triggers in our registry (which for the moment will remain OIM) and built atop a set of ActiveMQ message queues. I'm glad to provide whatever is useful about our existing processes, and whatever we can (since they're still in development, some of it may change along the way) about what we're developing.

    - GaryW: We have a central person/group LDAP directory called EDS (Enterprise Directory Service) which, functionally, serves as our "registry". We provision into it from a combination of sources:
        - An Oracle SOA ESB web service--which itself is triggered by our PeopleSoft systems (the PERSON_BASIC_SYNC message, for folks familiar with PS). This allows us to create a "skeleton" person entry in EDS as soon as a Person record is created in either of our PS SoRs (Campus Solutions and HCM), with the necessary attributes for the person to create a UA NetID.
        - Grouper PSP, used for provisioning both ad-hoc (user-defined and maintained) and "enterprise" groups, derived from SoR data (mainly course groups)
        - Batch processes, for pulling additional person attributes into EDS (e.g., position data, student career/program data, etc) and for populating Grouper with SoR data
        - We also have an enterprise AD, which pulls select attributes from EDS via a daily batch process.
        - I'm very interested in learning more about what sorts of event-driven scenarios other CIFER members have implemented (or are investigating), especially with regard to provisioning SoR data into their person registries/LDAP directories.

- MarkJ: We use NetIQ Identity Manager to pull data from five Systems of Record (Employees, Students, Residents, Physicians, and Guests) into our Person Registry (implemented on eDirectory). Selected data is then pushed to our enterprise directory and other downstream directories including AD. This is all event driven and essentially real-time.

- JonathanP: I see the Arizona design shared by Gary Windham is somewhat similar to what we have in place at the University of Washington.
  - We've utilized a home grown Person Registry for the last 15 years (now version 3) which is used for ID matching, NetID management, and account provisioning/deprovisioning (ssh calls made to do many provisioning tasks). Most data comes in from nightly data feeds from the employee/student mainframe but the payroll web application does try to write updates in a tightly coupled manner. The Person Registry publishes data out to our Person Directory via an early "event queue" mechanism running in the same space as the Person Registry DB. We also refer to our Person LDAP store as the Enterprise Directory Service (EDS) but in our environment it only contains limited person data (identifier mapping, some student and employee attributes, a uid number). We also have separate LDAP directories for Groups, email forwarding, and public whitepages data.
  - We've successfully steered most applications to use our Person Web Service which front ends the Person Directory (LDAP) data. Our groups environment has a Web Service and UI skin in front of Grouper using a listeners applications that pushes data out to the group LDAP server and Amazon SNS (was ActiveMQ until recently).
  - The four ldap service mentioned above are all utilizing OpenLDAP. Our AD environment is linked with several helper components which read in some person data from the OpenLDAP audit log and memberOf details from an Amazon SQS queue linked to the main group SNS mentioned above. In the AD environment departments are allowed to manage computer objects in a delegated OUs but person objects are in one main OU where a simple web tool provides authorized support staff the ability adjust a few AD account specific attributes.
  - Eric Kool-Brown presented on our current AD groups provisioning design at InCommon CAMP last year. For more information on that topic see linked slides under "Synchronizing Active Directory Groups" in the program schedule:
  - https://spaces.internet2.edu/display/CAMP2013/CAMP+Program

UPCOMING CALLS:

- Tuesday, 15 April 2014: David Langenberg, U Chicago, will present on their approach to identity and access lifecycle management built on a state machine engine