High-level overview of the provisioning workstream

Overview of Provisioning

- The "Job of Provisioning" (abstract technical definition): Keeping identity information state consistent across all components of the institutional IT ecosystem

- Guideline: Build provisioning models using compositions of Enterprise Integration Patterns (EIP): http://www.eaipatterns.com/
    - "Event-driven" and "messaging" are high level, abstract EIP constructs that help structure discussions of provisioning capabilities
    - They do NOT commit one to a particular technical infrastructure such as ESB, JMS, though those are among the candidate solutions
        - (RobC: Is this a good place to note (or do we even need to note) anything about the concepts of near-real-time consistency versus time-average consistency, which in turn lead us to think more in terms of "event-driven" and "messaging" than in terms of "file-transfer" and "batch"? Or to note that even with near-real-time consistency as a goal, we recognize the need to plan for bulk synchronization as a option in the event of catstrophe recovery or simple initialization in a pre-existing environment?)

- Principle: Standards at the core, customization at the edges
    - E.g., Core: Canonical data models; SCIM as the rising new protocol to serve provisioning needs
    - E.g., Edge: Connectors that speak SCIM on one side and speak app-specific APIs on the other; flow goes both into and out of app

-
- Scope: Re-label Provisioning as "App and Data Integration Services"
    - This brings into scope the feeds from source systems to person registries
    - in addition to traditional view of provisioning FROM the person registry to "consumer" systems
        - (RobC: I think this is good, also, in that it reinforces the idea that provisioning isn't necessarily just "attribute synchronization" -- at the "application integration" rather than "data integration" end of the scale, things like provisioning of access control information associated with identity (eg., FERPA privacy controls for students) or more complex entities like group memberships may (although not necessarily) fall in-scope)

- Candidate solution frameworks (existing integration stacks)
    - Kuali RICE
    - The Apache integration stack: http://servicemix.apache.org/
    - FuseSource.com, Wso2.com (packaged, supported Apache integration stack)
    - Open source projects descended from Sun IAM suite

- Deliverables:
    - [OSIdM4HEteam:Extensible] person/agent identity information schemas
        - including mappings to/from canonical data models, e.g., SCIM, LDAP, RDBMS)
    - Connector building, collecting, reposing, support for downloading
    - Detailed recommended solutions to a defined set of common provisioning tasks
        - based on compositions of EAI patterns
        - including source system of record feeds to the person registry
        - as well as classic provisioning cases
        - including optional rules engine to externalize identity business processes including identity life-cycle management
    - Reference implementations of recommended solutions in Kuali Rice AND in Apache ServiceMix
        - (RobC: Might we want to go so far as to plan for reference implementations provisioning in both the outbound (registry-to-target) and inbound (source-to-registry) direction, here? If there's interest, at least, in sharing code or strategy between data flows on the inbound and outbound sides of the registry, it might be helpful for developers looking to extend the reference tools to have reference implementations in both directions)
    - Training materials, training events

Workstream deliverables table