Interim Report Items

Interim Report Notes from the OSIdM4HE Provisioning Subcommittee

1. **Provisioning Is Critical** We remain convinced that provisioning is a critical capability of an effective IdM stack for higher ed, and that its scope extends not only to moving information between components of the stack (although that may be very important to provide for) but also to moving information between the stack and external, connected systems.
2. **Primary Goal Is Consistency** We seem to have settled, roughly, on the idea that the primary function of a provisioning facility is to produce and maintain, through the lifecycle of a provisionable resource, consistency between that resource's representation in some source facility (which we expect in this case to be a registry, although we would look to the Registry group to consider questions of whether a single registry or some form of logical meta-registry is appropriate). We have some commitment to the concept of "consistency" rather than "synchronization", largely in recognition of the possibility that a target system may represent a resource differently than the source system but may still need its representation to reflect the same semantic content as the source system. (We are working on developing a more formal definition of this notion of consistency, now).
3. **Provisioning Scope Extends Beyond Just People** We seem convinced that a successful IdM stack will scope its provisioning mechanism(s) in a way that allows for provisioning not only of traditional person identities but also of other classes of identity-related or identity-dependent resources – groups, roles, privileges, etc. As to whether actual delivery of provisioning features unique to non-person entities is critical for an initial offering or not, I don't think we're as certain, but the general timber of our discussions seems to have led to including non-person objects and their attributes in the collection of resources a good provisioning system should be capable of manipulating.
4. **Provisioning Needs to Minimize Latency** we're confident that a successful provisioning facility will ensure, to the extent that any technology can under possibly varying conditions of performance and availability, that inconsistencies between source and target systems introduced when source data (again, typically in a registry) changes will be resolved in target systems with a minimum of latency, and that under routine circumstances, target systems' representations will be kept consistent far more of the time than they are inconsistent.
5. **Provisioning Needs to Include Repair** We recognize that in pursuit of low-latency, there may be times at which "incremental" provisioning fails to meet the standard for on-average consistency above, and that at certain boundaries (such as the inception of a new provisioning facility, a new target facility, or a new connection between existing provisioning facilities and target facilities) there may be a need to instantiate consistency between sources and targets without the aid of "triggering" changes in the source facilities. As such, we note that any complete provisioning solution must surely include support for the notion of a "full resync" or "reconcile and repair" process by which inconsistencies introduced through error or architectural change may be addressed without the introduction of synthetic "events" in source systems.
6. **Provisioning Should Embrace More Than One Mode Of Operation** Items (4) and (5), I think we've determined, can be achieved in a number of different ways, each well-suited to different classes of provisioning scenario (enterprise, federated, and "cloud", for example) and sometimes to different strategic goals ("just in time" versus "just in case") and different consumer strategies ("push" versus "pull", etc.). So far, I don't think we've necessarily ruled out any of the options for implementation as either infeasible or unable to provide value under any conditions, so I don't know that we can identify any specific mix of options as "the" solution strategy. To some extent, that may be a matter for the developers to weigh in on as much as for us to consider at this point in the process. That said, I think we have a general sense that given the possible spread of target capabilities that may be encountered, it may not be possible to select exactly one collection of strategic positions (eg., to say that the provisioning facility need only support "just in case, push" provisioning). We are likely to instead need to aim for a "spanning set" of interoperable tools and processes that collectively address the needs identified in the use cases we choose (with the wider effort) to address.
7. **Provisioning Facilities Must Support Multiple Sources and Targets** A well-designed provisioning facility needs to have the capacity to support a wide range of target or connected systems. It seems reasonable that we might focus our efforts on or even limit our efforts to the OSIdM4HE provisioning facility consuming information from OS4IdM4HE registries, rather than from external "authoritative" sources. The provisioning facility, however, should employ a well-documented and well-specified interface for interacting with its data source(s)/registries, that should be well-implemented in its initial release – should other sources be deemed important in specific deployments, there should be support for developers' use of those same standard interfaces. In essence, I suspect we'd advocate pushing the responsibility for supporting multiple source systems onto the registry, standardizing on a single (preferably open-standard) consumer interface for the provisioning facility that's negotiated with the registry developers, and making support for a variety of "downstream" provisioning consumers the greater focus of the OSIdM4HE provisioning effort. (Note: to the extent to which the Registry effort may lead to the production of multiple source registries, and/or the access management effort may produce a separate source for provisionable privilege information, we would propose that all intended provisioning sources plan to write to an agreed upon standard interface that the provisioning facility can implement).
8. **Provisioning Target Flexibility Through Standardization** As much as possible, I think we'd like to try and accomplish target flexibility through implementation of standards-based interfaces for provisioning target systems. We have reviewed a number of existing and nascent provisioning standards, and would note that the SAML Change Notification work seems very promising (as evidenced by Google's stated support of the effort), as does SCIM (which has the advantage of a lively and actively open development community). Where "high impact" target systems (read: GoogleApps, Sakai, etc.) can be identified, we'd advocate trying to negotiate support from them for standard provisioning interfaces. Likewise, where our community controls high-visibility target systems (Grouper, CoManage, Kuali, etc.).
9. **Provisioning Should Exhibit the Pattern "Standards at the Core and on the Wire, Customization at the End Points"** While we recognize that there are going to be intractable target cases, some of which are likely to be of critical importance to some or all of our community, we would support the notion that customization of "adapters" should be the province of those target systems, and that their adapters should target a standards-based interface provided by the provisioning facility, rather than targeting a specific provisioning facility implementation directly. To the extent that some targets may be more easily adapted to one standard for provisioning than another, we would support the long-term goal of providing a "spanning set" of standards-based provisioning interfaces.