Pre-Conference Item 2 - Chunking and Logical Units

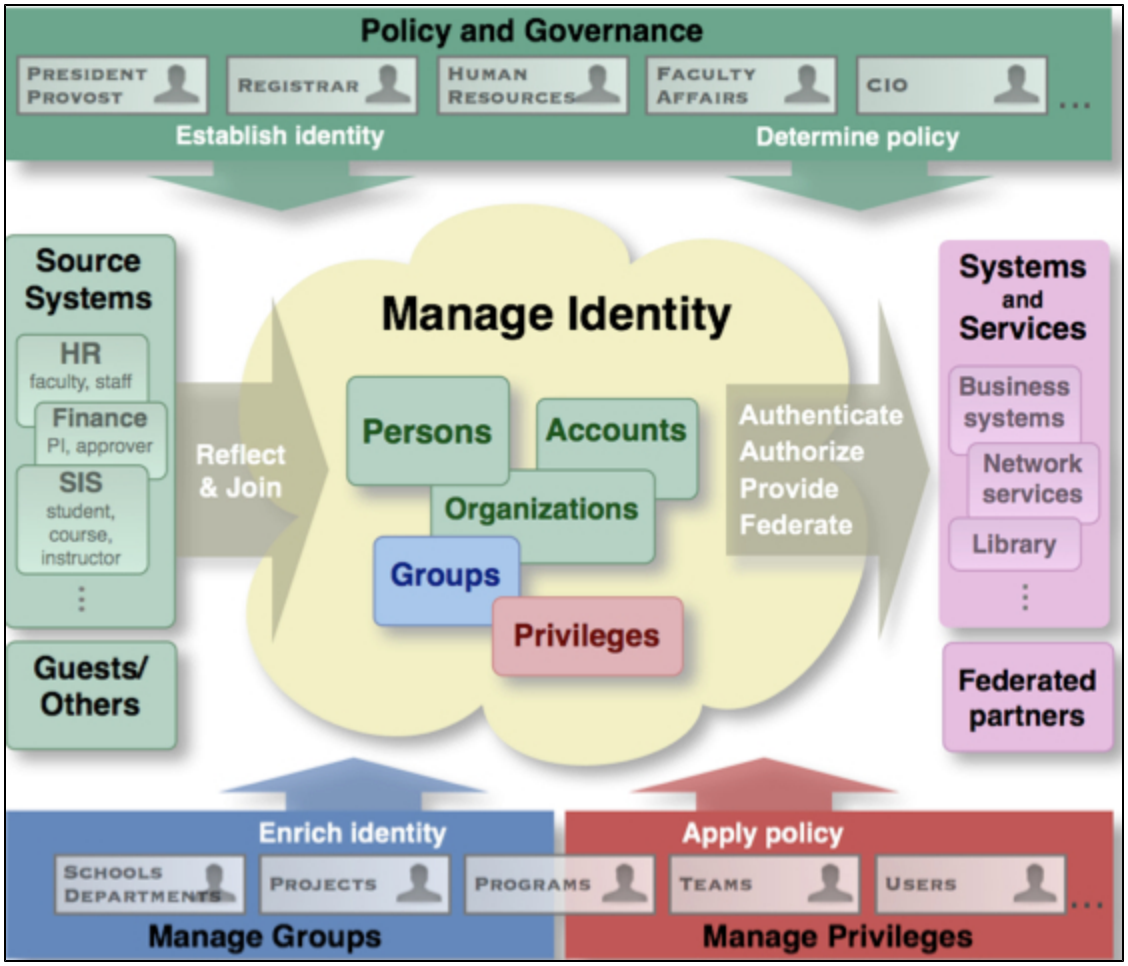
Modularization of Architecture

In early discussions with the group about how best to approach translating collected use cases for IDM services into a gap analysis and ultimately a prioritized set of design and development goals and technical requirements, a number of us expressed the concern that whatever solution or solutions arise from our collaboration be architected in a fashion that will allow for "piecewise" deployment of our solutions. Our expectation is that if we design and construct YAMIS (Yet Another Monolithic IdM Suite), most sites (who have already implemented some solutions within the IdM space) will be unable to adopt our solution, either because of political pressure to continue demonstrating benefit from prior purchase of commercial solutions or because of practical difficulties in implementing a "fork lift" replacement of the entire institutional identity infrastructure. Better, we surmise, if our solutions can be modularized, with well-defined interfaces between components that can allow individual sites to choose whether to deploy the entire OSIdM4HE suite or integrate individual components of the suite with existing solutions (potentially as a stepping-stone to replacing existing solutions with other components of the suite). The group as a whole seems committed to delivering (either through new development or through some combination of new development and integration of existing open source components) a complete IdM suite (that, itself, being a somewhat ill-defined concept for the moment) for those sites that want a top-to-bottom open source alternative, but seems agreeable to the concept of modularization.

Chunking

If we're to frame the gap analysis and requirement gathering part of our Chicago discussion in terms of logical units or chunks, then, we must answer the question of what the logical units or chunks we believe should be comprised by a comprehensive IdM suite really are. That's the point of our discussion here. Each of us (Rob, Keith, Jacob, and Steven) have expressed some interest in the effort, and also some prior art in the area – some of us have intimated that we have conceptual frameworks in some degree of maturity already in mind. We'd like to use this page in the wiki to document our ideas, and perhaps stage a conference call-cum-Adobe Connect session to flesh out our ideas in preparation for Chicago the second week in August.

Any discussion of modularizing the IAM space is probably deficient if it doesn't at least tip its hat to the Educause/I2 functional diagram:



In theory, that conceptual division of tasks spans the entire IAM space and could be used directly as the basis for our discussion. It's not clear to me, at least, though, that the logical divisions which work well for discussing the functional aspects of an IAM strategy necessarily map directly onto the kinds of modular units from which a technical architecture to deliver those functional aspects can be built – I suspect there are some concepts that may need to be added into the mix to achieve a level of granularity more appropriate to our particular discussion.

Proposals

Pre-conference discussion results

We've had some good electronic discussion, both in concalls and via email, and although we differ in the specific ways we draw the boundary between what's in-scope and what's out-of-scope for the term "IDM" and somewhat in the component parts we envision that an IDM Suite needs to comprise, we're in rough agreement about a few core concepts. At an extremely high level, we seem to agree that an IDM has at least a few high-level components:

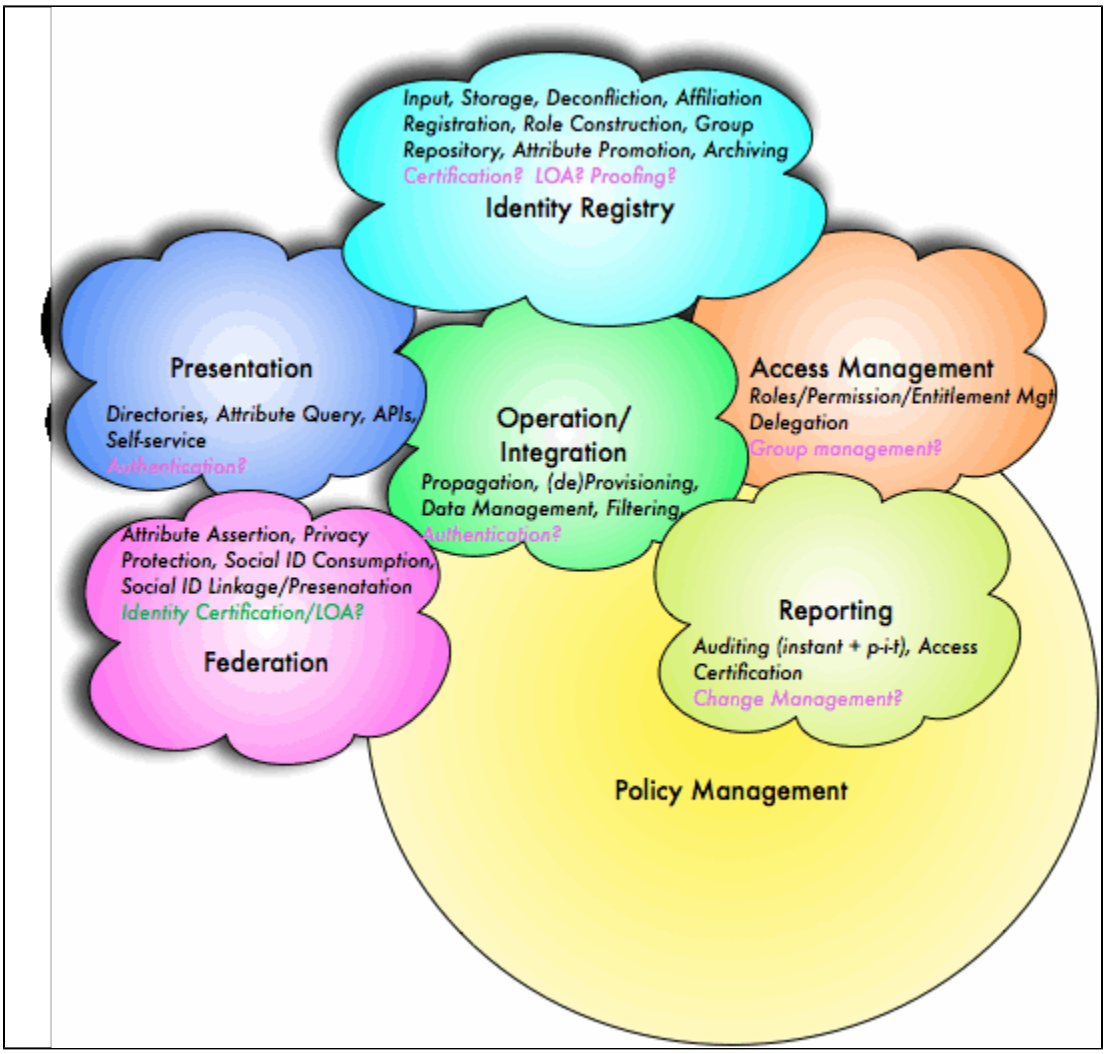
1. **Input Mechanisms** - We're all pretty certain, it appears, that any IDM suite worth the moniker needs to consume identity information from authoritative data sources external to it. We're convinced that at least some identity information must be sourced outside the IDM facility, and that that information will almost certainly be provided by *multiple* external systems.
2. **Deconfliction** - We're all pretty certain that the disparate sources of identity information will not magically deconflict against one another and construct coherent, persistent identities – that is thus one of the key services an IDM suite has to provide.
3. **Synthesis** - We're all (in slightly different ways) committed, it appears, to the idea that an IDM suite needs the ability to synthesize certain elements (attributes, roles, memberships) based on a combination of business logic and deconflicted source identity data. This may take the form of promoting attribute data from domain-specific source attributes to generic identity attributes (eg., affiliation data + HR data + student data ==> "person" data), synthesizing functional roles or group memberships based on domain-specific source attributes (eg., both Law and Med students are members of "professional students", or staff with job codes between 10XX and 20XX in the HR system have the high-level functional role "personnel manager"), or a combination of both.
4. **Data Provisioning** - We seem to all agree that any comprehensive IDM suite has to provide a means for reflecting the data it consumes and deconflicts into other consuming systems via some form of data provisioning. This presumably incorporates both traditional "account provisioning and deprovisioning" features and less traditional identity data management features (maintaining provisioned information in downstream systems).
5. **Access Provisioning** - We also seem to all agree that an IDM suite needs to provide a means for reflecting access control information into downstream systems. This provisioning (and deprovisioning), we seem to generally agree, will be dependent on both affiliation information, granular identity attributes, and synthesized functional roles. Its provisioning may entail reflecting and managing both permissions and groups or roles into downstream systems.
6. **Auditing and Reporting** - Out of necessity, we all seem agreed that an IDM suite is incomplete until it can provide management and the ever-present auditors visibility into the state of identities, permissions, roles, and other objects the IDM suite traffics in. This might be limited to an instantaneous view into the current state of all users, roles, and systems touched by the IDM, or it may include the ability to report on the history of such an object's state and/or its state at a given point in time.

There are a handful of other items that are either in my (Rob's) own personal view of the space, or seem to be implicit in our conversations to date, and which may or may not be "in-scope" as a result:

1.
   a. **Data Storage** - There's at least one common existence proof of an IDM suite (the old Sun product) that seemed to be largely devoid of a central data store, yet provided (most of) the basic features outlined above. Some would argue, though, that an IDM is only marginally complete if it does not provide some (possibly if it does not provide some *relational*) data store for persistence and to facilitate things like deconfliction and synthesis. It may be that there are no use cases which cannot be solved without an integrated identity data repository, or that there's a proof somewhere that anything an integral data store can provide can also be provided using a "virtual store" mechanism, so this may be an implementation
   detail, but there may be use cases that seem to presume the existence of some persistent and mineable store of identity information.
   b. **Access Policy** - It may be implicit in the concept of access provisioning, but I suspect it bears considering whether an IDM suite needs to provide some integrated mechanism for expressing, storing, and operating on access policies as objects in their own right. Perhaps there's a policy-free way to describe the mapping from identity attributes and synthetic functional roles into permissions, but I, at least, can't seem to envision the one without the other. This may be in-scope or out-of-scope, but if it's out-of-scope, I think we'll need to articulate how an IDM should consume policy information (even if we end up agreeing that policies should be translated manually by human agents and entered as collections of permissions mapped onto roles in an IDM suite).
   c. **Data Presentation** - A number of commercial IDM suites seem to take the approach of considering identity presentation layers outside their bailiwick, viewing facilities like directories, authentication services, and the like as "downstream" systems into which they may provision data or in some cases access rules, but not considering them an integral part of the IDM picture. Others seem to consider presentation layers (especially directories) to be so integral to what they're doing that they're provided as part of the IDM suite. Personally, I view them as in-scope and part of the IDM picture (to at least the same extent as traditional registries and/or authorization provisioning is).
   d. **Federation Tools** - Federation is perhaps the most recently deprecated buzzword in our community (being rapidly supplanted, I fear, by "cloud"), so any discussion of IDM suites probably can't be complete without some consideration of federation tooling and its relationship to the IDM. It's not clear whether we should consider things like Shibboleth (which are part authentication services and part federated attribute brokering serices) or things like SAML <-> OpenID gateway mechanisms or federated group membership mechanisms as in-scope or out-of-scope for the OSIdM4HE process, but I think it's worth at least identifying as an uncertain point. Personally, I lean toward considering federating tools (both IDP-like facilities for presenting enterprise identity across federations and RP-like facilities for consuming identities from federated partners and other trusted asserters) as part of the IDM space and something an OSIDM should at least be cognizant of. Would an IDM that's not capable of registering users with scoped identities ala those exchanged among federation partners really be applicable to the HE environment, for example?

Diagram - IDM from the consumer's perspective

To some extent, the discussion thus far (outlined above) has largely focused on the IDM suite from the perspective of the IDM itself – the discussion has been couched in terms of what services an IDM suite needs to provide or what functions it needs to fulfill. That's extremely useful, but at a very high level, I suspect there's a slightly different perspective that may be useful to consider, as well – that of an external system either consuming or integrating with the IDM suite as a data provider, provisioning endpoint, query supplicant, etc. At an extremely high level, without recourse to thinking about how the individual parts fit together, it seems as though there may be a sort of overlapping cloud diagram one might consider to represent the slightly different perspective an IDM data consumer or data provider might have on how the IDM suite functions. I'm not sure that I've got this exactly right, nor am I entirely comfortable with my own assignment of low-level features and operations to the individual clouds, but I thought this picture might be helpful, even if only as a way to prove this isn't a fruitful direction to think it...



A few notes on the diagram:

- This is based on my personal high-level chunking, in which much of the stuff we seem to largely be in agreement about falls under the rubric of an "identity registry".
- I tend to separate out the registry from the mechanisms by which data in the registry is made useful (propagation and provisioning of identities to downstream systems, handling of access management, presentation facilities like directories and such. This may be because here at Duke we've long had things roughly chunked that way in our software implementations. I don't actually *like* the way we've always done things in some senses, so I'm quite open to rethinking that.
- When I've used this diagram before, I've left the policy management circle out of the picture, usually because I'm using it to talk to the folks I want to engage about policy articulation :-). My idea putting it in the way I have in this version of the diagram is that policy management cuts across a large part (but not all) of the picture, and if the other components are represented as overlapping clouds, policy management might be viewed as what the clouds are obscuring in a sense.
- In the small type inside each cloud, the black text relates to things I'm pretty certain fit inside the category, and the pink or green text relates to things I'm not sure are relevant or I'm not sure fit well in one or another cloud within the diagram.

Other Materials

Matt Sargent from the Kuali group has an interesting alternative approach to breaking down this space available here

Questions

Scoping Questions

1. What's the overall scope of identity for our purposes? Clearly, it includes person identities, but should an IDM suite be expected to also manage identity information for systems? How about management of aggregated objects – groups? Roles?
2. Is our focus to be purely on core IDM functionality (which might be defined as functionality exposed only to other systems and to technical specialists, but not to end-users) or should our scope be broader and include end-user interfaces (things like self-service facilities for data management)?
3. Does OSIdM4HE consider the management of permissions and privileges as in-scope or out-of-scope?
4. What about management of access control policies?

5. And federation tools?
6. There are a few components we can likely all agree are needed in order to successfully implement an IDM suite – workflow support, notification mechanisms, primary authentication services – should these be considered in-scope for the IDM or should they be considered "external infrastructure" that the IDM suite relies upon and presupposes as a co-deployment, but doesn't make direct provisions for?

Chunking Questions

1. Can we agree at this point as to whether an identity repository (where aggregated identity information would be stored by the IDM, separate from their origin systems) is or isn't a requirement for the OSIdM4HE?
2. Is delegation a feature purely of the permission and privileging portion of the IDM, or should it be considered as a feature that's embedded throughout all the components of the IDM suite?
3. Does the separation of identity provisioning and access provisioning make sense, or should "provisioning" be a more general "chunk" for our discussion, and include both (or perhaps, if access control is considered out of scope, is this not a valid question)?