Minutes of the 10-28-2011 concall

10-28-2011, 4pm - 5pm ET
Attending: Rob Carter, Keith Hazelton, Lucas Rockwell, Tom Zeller

- Rob opened with a quick check of the previous call's minutes (to which there was no dissent), and noted that the bulk of the call might well be spent discussing how to arrive at a more discrete project proposal to provide to the strat-org group, but asked if there were any other items that might need attention as well.
- Keith agreed that focusing on a project proposal would be appropriate, and suggested that as things are playing out, the registries group seems to be taking something of the lead for the overall effort, in part because construction of the registry component of the project seems to be both seminal to the overall effort and important in the short term to a number of the players involved.
- Rob introduced the idea that the group might consider entertaining a collection of questions that could potentially fill in the blanks in a more specific project proposal, noting, for example, the question of whether provisioning should be constrained to outbound operations from the registry (making states in other systems consistent with the state of the registry) or whether it should be construed to include also loading data into the registry.
- Keith picked up on the question and noted that the registries group is going to have to load data, and if the provisioning group doesn't provide a mechanism for doing that, the registries group will have to develop a strategy on their own and separately, possibly leading to two very similar but distinct approaches to the same basic problem.
- Rob agreed, but noted that the other (and perhaps equally undesirable) outcome could be that provisioning would be limited to using whatever mechanism the registries group devises for loading its data.
- Keith agreed and added that in either case, it seems as though the provisioning group will have a better sense of how best to handle those issues, since that's our focus.
- Rob then noted that during the Mace-Paccman call earlier in the week, there had been a discussion of the OSIDM4HE work and particularly of the potential interaction between the Paccman effort and the provisioning effort, especially since there's some suggestion that provisioning and access management may eventually be combined within the OSIDM4HE effort. During that discussion, it was agreed that the Mace-Paccman folks would be interested in participating in the conversation at some level, and would like access to the OSIDM4HE-Prov documents in the wiki. Rob had agreed during the Paccman call to consult the group and determine if and how they would be willing to share their content with the Paccman group.
- Keith suggested simply giving the Paccman group read access to the wiki space as a first step.
- It was agreed that Rob would work with SteveO et al. to arrange for world read access to the space, and would pass along links to material in the wiki to the Paccman group. [Editor's Note: As it turns out, the space is already world-readable – there is now a link from the Paccman page back to the OSIDM4HE-Prov space].

- Rob suggested that Tom might want to provide an update on how things are progressing in the ldappc-ng world, now that Grouper 2.x is out?
- Tom explained that Grouper 2.0.1 was just released (a point-release following the Grouper 2.0 release) and that the plan is to include so-called realtime provisioning with the ldappc-ng interface in release 2.1, which is currently under development.
- Rob asked what issues, if any, Tom is running into in developing a real- or pseudo-real-time interface for provisioning group information from Grouper into an LDAP?
- Tom expressed some uncertainty as to whether we'd want to necessarily say that ldappc-ng should or would be the basis for a more general provisioning interface, especially given that for the moment, he's the sole programmer working on ldappc-ng. He explained that the primary complications he's seeing with realtime provisioning have to do with the capabilities of the two endpoint systems – the source and target – since provisioning can only be as "realtime" as those systems are able to handle. He reiterated an interest in some other efforts, particularly the ForgeRock product (an open-source branch of the Sun IDM code), and suggested that an integration-based approach (using some existing product and integrating it with the OSIDM4HE registry) might be an alternative to designing or developing something from scratch.

- Keith picked up on the concept of integration, pointing out that provisioning, in his mind, falls squarely in the middle of the integration problem space, since by its nature, provisioning is a multi-system problem. He pointed out that if we view provisioning as a facet of data integration, we then must consider which of the well-understand data integration patterns we need or wish to support in our effort, and went on to point out that there are really only a few choices that need to be considered, citing point-to-point integration, hub-and-spoke integration, and data buses with some flavor of messaging as the likely candidates. He posed the question of whether we can realistically develop a plan for provisioning in a generalized IDM suite without expanding the scope (at least of the discussion and description) out to more general data integration patterns?
- Lucas agreed that the degree of overlap between the provisioning space and the general data integration space is large enough that a discussion of wider data integration techniques is probably in order.
- Keith asked Tom if there'd been any drift toward more general data integration rather than pure "provisioining" in the development effort for ldappc-ng, and if so , how that conversation was going. He explained that his assumption was that ldappc-ng, as an LDAP provisioning tool, would be making LDAP calls to get information sourced in Grouper reflected in an LDAP directory, and asked if that is in fact the core capability he'd be interested in expanding upon with ldappc-ng and possibly with the provisioning interface for OSIDM4HE as well?
- Tom explained that with ldappc-ng, he's using the SPML library as a sort of abstraction layer to allow plugging in different types of target systems, of which LDAP is only one (although at the moment, it's the primary focus and the only one for which code's actually being provided). He noted that there's not a great deal of effort across the industry, though, in developing SPML plugins for other target systems. Comparing SPML, he explained, to SCIM, there seems to be quite a bit more effort going into development of SCIM interfaces, perhaps in part due to the RESTful approach SCIM employs. He went on to relay that the SCIM working group had recently rescheduled it periodic conference calls to accommodate Google, because Google had expressed interest in implementing a SCIM interface.
- Keith noted that Google got the "SCIM religion" rather late in the game.
- Tom agreed and noted that Google wasn't on the last SCIM concall, but pointed out that there's no equivalent involvement from the wider world in the SPML discussion. It looks, he said, like companies that sell provisioning technology are looking to write SCIM interfaces into their systems and to port those interfaces through their product lines, rather than writing SPML or other interfaces. He raised the possibility that building a SCIM interface inside Grouper might actually be sufficient for addressing the realtime provisioning problem there, and that to some extent, the provisioning problem as a whole might decompose into a matter of point-to-point data interchange via SCIM between endpoints. He asked whether it might be that writing a product to support provisioning might be less sensible than building interfaces into the other products to support SCIM, and leveraging SCIM as the basis for a provisioning strategy – essentially, build a SCIM interface into the person registry and provide guidance for how to interact with the SCIM interface. This, he suggested, might be more palatable than a whole new implementation for those groups that may already either have an existing provisioning or state-consistency mechanism or that already have a working person registry deployed and are interested in still taking advantage of other parts of the OSIDM4HE effort.
- Keith agreed that SCIM is an interesting option, and Tom suggested that simply building a data mapping service based on SCIM that could map data between two different SCIM endpoints might be sufficient to solve the provisioning problem in most if not all cases.

- Rob expressed concern that it might be a bit of a rat hole, but asked whether what we're shooting to provide in the provisioning effort is merely pipe fitting techniques and tools, or whether there's more in the scope of the provisioning effort that needs to be addressed.
- Keith indicate that he would be strongly in the camp supporting the concept that its more than just pipe-fitting, and that it borders more on data integration, but suggested a quick straw poll of the attendees on the subject.
- Rob and Keith agree that the issue can best be viewed as more akin to data integration, of which pipe fitting is only one (albeit essential) component. Perhaps it's only pipe-fitting mechanisms and some advice about how to use the tools to achieve data integration goals, but it may need to involve some specific toolkits (identified and/or developed) for handling the transformation of data, along with (potentially) some recommendations about implementing bus-like mechanisms and handling message passing (which is almost an inevitable direction these discussions head in eventually).
- Lucas indicated that his view might be heretical, but expressed the belief that downstream systems can be separated from source systems' data transfer (which term he prefers to use over "provisioning" in this case) into the registry. He pointed out that on the "loading data into the registry" side of the fence, there is a very small number of usually very well-understood and tightly controlled systems that are upstream of the registry, whereas downstream of the registry, there may be far more systems, many of which may be less well-understood, less tightly controlled, and more likely to exhibit a wide range of disparate data interfaces. On the downstream side, he suggested, something like an ESB may become appropriate (as an abstraction layer to rationalize the disparate input interfaces exposed by downstream systems) while on the upstream side, direct data exchange may be more efficient. At the least, he argued, the two operations are distinct and may admit of distinct solutions.
- Keith proposed that at the very least, that would suggest that the priority for the provisioning effort needs to be the outbound or "downstream" side of the registry, and agreed to at least that much. He suggested it's an open question to the registry team, though, how they're viewing the problem of getting data into the registry from external source systems. He pointed out that the registry group will necessarily come up with some means for getting data into their product, whether it's a collection of pipe fittings and tools, a standard interface to which source systems are expected to adapt, or simply a collection of recommendations about how to extract source data in a form useful to the registry. He expressed total agreement, however, with the concept that the center of the provisioning effort needs to be focused on the downstream side of the registry, with any upstream reuse of provisioning code or concepts being of secondary importance. He then asked if Tom had a position he'd like to express?
- Tom indicated that he's not all that familiar with tools for more general data integration – he noted that at the I2 meeting he'd run across some UK members who had positive things to say about their experience with a product from Talend for that purpose.
- Keith noted that he's had more involvement with data integration as a result of his work with the Bamboo project and that fact that Bamboo chose early in their process to go with the Apache integration stack. He noted that additionally the next big hurdle for the infrastructure at UW Madison is likely to be data integration, so he's been bumping into the space quite a bit, of late. He suggested that the provisioning team might want to check in with the registry team with a question like: "In the provisioning group, we can see a couple very different scopes for the provisioning project, from the very constrained 'pipes and fittings only' scope up to a full data integration capability scope. We believe there is a gap across the entire space that should eventually be filled. What part of that entire gap do you see the provisioning effort filling, and what, if any part of the gap do you see filling directly as part of the registry project?". Perhaps, he suggested, the registries group can offer some guidance.
- Rob agreed, but suggested that perhaps we might recast the question a bit more specifically, along the lines of "We see two gaps surrounding the registry from the outset – a gap between authoritative data sources and the registry, and a gap between the registry (or registries) and target systems. We have a good sense of the shape of both gaps, but we need to verify whether the registries group (a) sees commonality (as we seem to) in the shape of those gaps and/or (b) views addressing one of those gaps as part of its scope rather than the provisioning effort's scope." He also noted that there's a similar question regarding the positioning of business logic (and in particular, the positioning of attribute transformation (both attribute naming transforms and attribute value transforms) between the registry and the provisioning mechanism – whether the registries folks view transformation logic as part of their charge (eg., does the registry "bake" data down to the level of explicit target attributes that can simply be transported to target systems by the provisioning mechanism, or does the registry house "gold standard" attribute information that may have to be transformed in some fashion by the provisioning interface in order to be made consumable by specific target systems). The question, in essence, of what part of the data integration problem is to be solved by provisioning versus what part will be solved in the registry effort. He asked who on the registry team might be the right person to approach.
- Keith suggested that Bob Morgan is likely the right person to contact, even though he's not actually (officially) part of the registries team.
- Rob suggested that he would then take an AI to develop a short list of questions to pass to the registry folks for their input.
- Keith asked Lucas, as a participant on the registry calls, whether he thought this approach would be sensible?
- Lucas noted that he'd only just joined the group and only had one call so far, with them.
- Keith noted that the group includes Stephen Carmody, Benn Oshrin, Jeremy Rosen, and a couple folks from Kuali, with Matt Sargent probably playing lead.

- Rob suggested that a first question might be along the lines of "Does the registry group want to have its own input mechanism for loading data into the registry, or should the provisioning group plan on providing for the input of data into the registry as well as the delivery of data out of the registry?
- Lucas noted that based on his limited discussions with them, the registry folks would likely be receptive to the idea of an input mechanism, since they seem mostly focused on developing a data model for the registry at this point.
- Keith agreed that that would be a good question to pose, and forwarded a link to the registry functional model proposed by Bob Morgan.
- Rob then suggested a second question to pose to the registry group, roughly: "Does the registry expect to bake out and store states for each target system (that may have slightly different states dependent on registry data) or do they see some (or perhaps all) of the target-specific computation occurring outside the registry (either as part of the provisioning process or as target-specific data integration)?"
- Lucas noted that at his site in California, that work si largely done (more or less) inside the registry. Their provisioning interface does little if any munging of data, mapping attributes one for one. In their model, he noted, registry provisioning is more or less entirely a matter of reflecting attributes into their LDAP, from which one-to-one provisioning is done to other systems.
- Rob asked if there are other questions that should be routed to the provisioning folks?
- Keith noted to Tom in re: his point about having only one programmer (himself) largely attached to the ldappc-ng effort, that the point of the overall OSIDM4HE effort is to get some funding to provide resources (like additional programmer time) so although the hope is that Tom can lead a development effort for the provisioning interface, the expectation shouldn't be that he'll be doing it alone at all.

- Rob agreed to take AIs to provide links to OSIDM4HE-Prov content for the Paccman group, to construct straw man questions to pose to the registry group, and to construct a list of straw man questions around which to focus the next call's discussion (in an effort to tease out the different decisions we need to make in order to come to an actual project proposal and some specifications).