Reference Architecture Recommendations for Groups and Folders

From MACE-Dir call 2016-02-22:

[AI] (Bill T) will take next steps in exploring development of Grouper Reference Architecture Recommendations for Groups and Folders

Grouper is a powerful enterprise access management system. This power and flexibility however can be quite daunting especially to a new deployer. For instance the approach to group and folder design is mostly left up to the deployer. In response to that the community has provided a number of examples and suggestions: https://spaces.at.internet2.edu/display/Grouper/Group+and+folder+design+ideas

Some common themes among these approaches include:

* a root folder for the institution like “lafayette” or “lc”

* “etc” folders for configuration and admin groups

* a folder for reference groups (i.e. institutional affiliations)

* a folder for applications

* a folder for organizational hierarchies

* a folder for class rosters

* use of folders hierarchy to provide delegated management

A reasonable starting structure based on initial use cases might be:

* lc - top level folder to organize deploying institution namespace folder and groups

* lc:app - enterprise applications folders and authorization groups

* lc:org - organization hierarchy and groups

* lc:ref - reference groups

* lc:crs course rosters

* etc - top level folder for Grouper config groups, sysadmin group, loader jobs

* test - testing folder for the IAM team

Additionally there are other examples of implementing capabilities via:

* composite groups to implement allow/deny for “authorization groups"

* composite groups to supplement sources of truth (anti-pattern?) for reference groups

* nested etc folders to implement ARBAC

* rules

* various options and settings in grouper.properties, etc

Providing a reference model for these questions would make adopting and operating grouper easier, lead to more consistent practice (and value to the deployer), and set in place the possibility of higher order capabilities based on the reference model (i.e. capabilities or functions that can rely on specific folders/groups/attributes/etc to be in place). One can envision a number of different access management use cases that lead to specific grouper structure and configuration. The reference model could incorporate these as well, so for each use case or capability there would be reference model for groups /folder/attributes/rules/etc.

Rather than leaving all the options open, the suggestion is for a more opinionated "TIER model" that could help institutions get deployed and operating faster, and with quicker initial wins. Basically continuing to move from “toolkit for group management” to “enterprise access management system” with some common practice.

Making any sense?

Best,

Bill

From: **William G. Thompson, Jr.** <wgthom@gmail.com>

Date: Wed, Feb 24, 2016 at 3:07 PM

Subject: Grouper

To: Tom Barton <tbarton@uchicago.edu>

Tom,

I'm not sure if I misunderstood your comments today about wanting to keep Grouper focused solely on group management, but I'd love to chat more about that if you have a moment sometime soon.

If "account provisioning" is left to some other component it will still need all the grouper power of group delegation, group math, loader, etc. in order figure out who should be provisioned. Why not let grouper take the final step and make it so.

I suspect that just managing incoming identity data, identity life cycle, and identifier assignment, etc will be enough for any person registry. And more practically most (and possibly all) institutions will be very slow to take on such a project. So in any case that gap in functionality will remain for quite a long time.

My current plan for IAM nirvana includes driving policy about account provisioning with grouper. This might not include "primary account" such as an LDAP DN in an EDS as required for primary authentication, but it will likely include every other system that needs "some identity data" (aka "an account") to function.

My current plan goes something like this:

1) assume a source of canonical identities (person registries are too hard, and everyone has one already!)

2) point grouper subject api at 1)

3) create base "reference groups" (constituents, courses, organizations, committees, offices, etc) mostly driven by systems of record and maintained by grouper loader.

4) implement access management policy (accounts and groups) using grouper group math magic to drive effective membership for "account groups", "authorization groups", and "other groups (like mailing lists, etc)" which maintain fidelity with target systems.

5) enjoy the afterglow. :)

Is this at odds with your vision for Grouper?

Best,

Bill