UW group service (why we're so different)

It

We have historical incompatibilities

We had a group service prior to Grouper.

- We use an entirely REST API. Nothing goes in or comes out except through the API.
- The GUI also uses the API.
- Many things were incompatible with off-the-shelf Grouper.
    - Underline instead of colon for path separator
    - Lack of 'visible' stems. Our groups can have subgroups. The stem is implicit.
- Already widely used before conversion to Grouper backend.

We were impatient

We needed some features yesterday.

- The REST API. Our users couldn't live without it.
- Changes by messaging. Our group caches (AD, Google, ...) couldn't live without it.

Why Grouper

Well, it is tested, supported, and works pretty well.

Even though we mostly use our local API, if we want someday to adopt a product that is written specifically to Grouper's API, say a Kuali something, we can quite easily bring up the necessary interfaces.

Why REST

A group system fits into the RESTful model so easily and naturally that I wouldn't any longer consider anything else. In our situation the representation of groups and members as resources extends from the web to our message queue as well. In many cases group representations are retrieved from the web service and places unaltered on the message queue.

Why LDAP

LDAP directories have one redeeming feature: they are fast. Because of this our GWS satisfies the two most common requests by queries to LDAP. LDAP directories can be easily replicated, a second redeeming feature.

Historically we allowed clients to access group data directly through the LDAP interface. There is a natural, convenient RESTfulness to it, despite the awkward syntax. Nonetheless, in many ways, particularly in regard to access control and recursion, LDAP is insufficiently sophisticated and not up to the task. Most clients are better served going to the GWS. Response quickness is the same, as is reliability and redundancy. In addition, access control is more well defined and recursion is supported.

Our goal is to discourage and discontinue client access to the LDAP directories that comprise an essential component of GWS. This does not affect groups in our Active Directory (also supporting LDAP), which apparently have some utility in the Windows world. These are treated by GWS as dependent groups and populated accordingly by the messaging system.

Why a message queue

Our people, through the applications they use, access group information through a variety of interfaces: GWS directly, Google Groups, Active Directory groups, and locally maintained (by some client) caches of group memberships. We keep all of these up to date by posting change notices on an ActiveMQ message bus. Anyone can read these notices and query the GWS for the details of the change---access protections are thereby preserved.

As an efficiency improvement, we make the details of changes available on the bus through encrypted sections of the messages. Thus trusted clients are spared the overhead of querying GWS for the details.

We do not update our principal LDAP this way as it is considered more of an integral part of the GWS service.