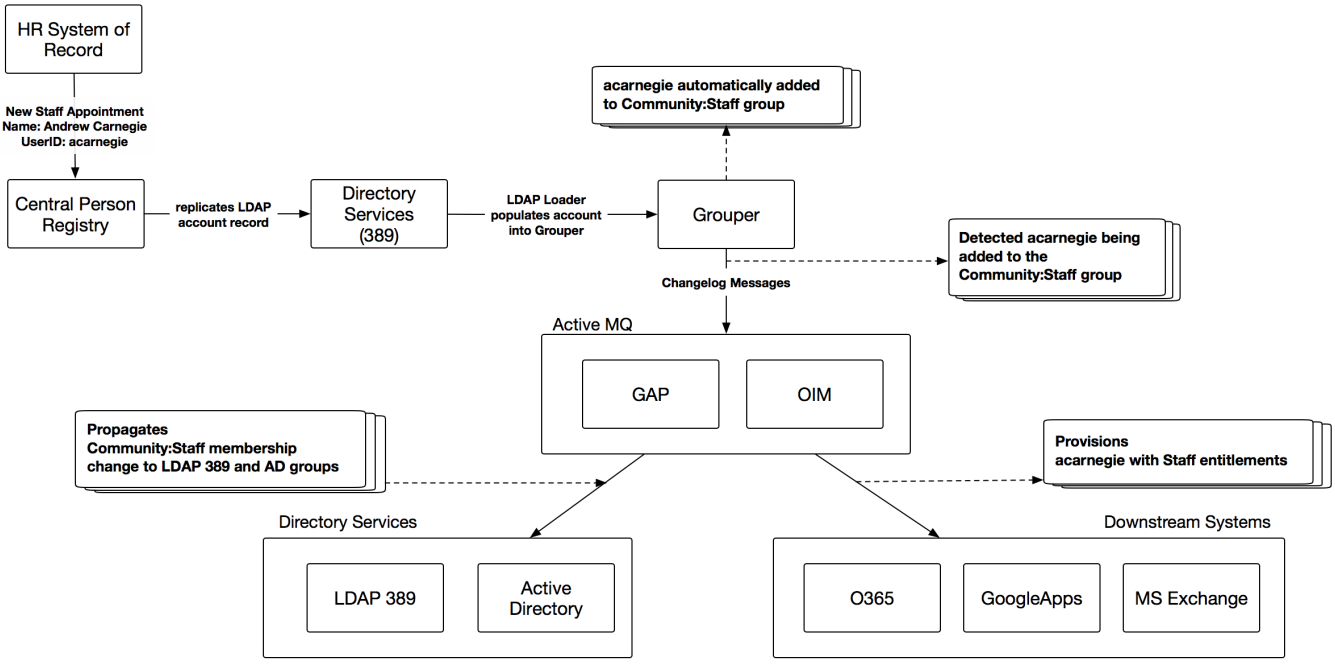
Carnegie Mellon University Grouper Project Page

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources

---Update on this service - November 2017

Carnegie Mellon have been extensively using Grouper and the Grouper Active-MQ Provisioner (GAP) framework for over 4 years now. This GAP infrastructure is our 'central hub' for distributing Grouper group change messages, via Apache ActiveMQ, to our central provisioning services. Our provisioning systems consist primarily of the GAP provisioners (custom code) and Oracle Identity Manager (OIM). This GAP infrastructure allows us to provision downstream resources, driven by institutional data through Grouper, in near real-time.

See below diagram for how an on-boarding staff member at Carnegie Mellon flows through our IAM infrastructure, and is provisioned resources thanks to Grouper and the GAP framework.

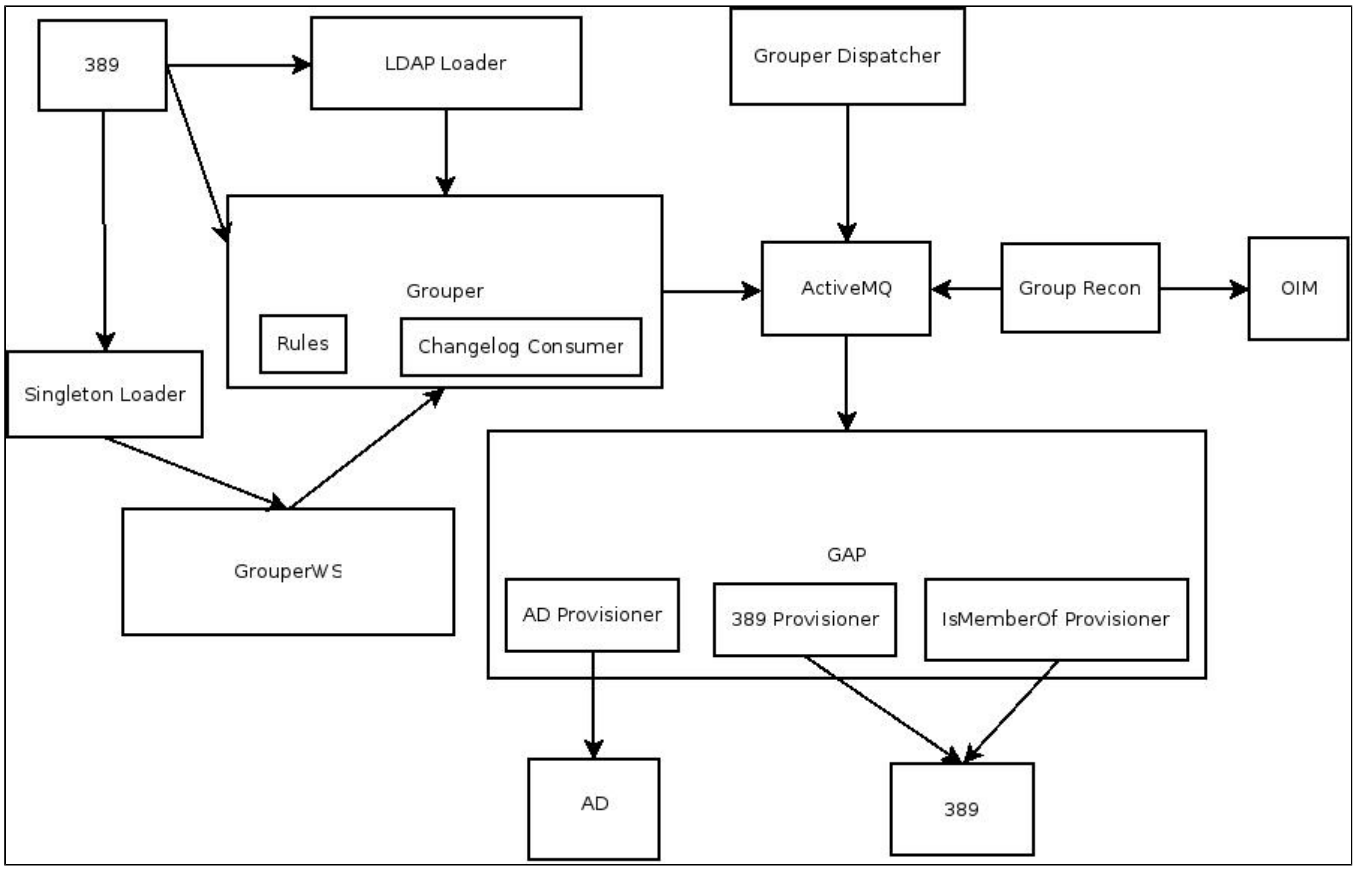


We've also found that application owners across campus, are able to leverage our ActiveMQ infrastructure, for provisioning/deprovisioning accounts into their own applications (hosted or in the cloud). For example, an application owner can come to us with the need to provision new staff into their application. We can provide them with sample ActiveMQ consumer code (which they can then adopt and host themselves) which will allow them to pick up on Grouper group change log messages and subsequently call the provisioning API for their application. This type of pattern has worked successfully for us, for services including Canvas and Slack.

Any questions/comments on the above, please drop me a line: Garrett King garrettk@cmu.edu

---

Grouper 2.1.4 was successfully deployed at Carnegie Mellon University in January 2013. Below is an architecture diagram that shows how we use Grouper in our Identity Management infrastructure.



389(port389.org**)** : It is our LDAP subject source for Grouper and also target for provisioning grouper groups.

LDAP Loader : It is a utility written in perl to load groups in Grouper based on person affiliation, campus, status etc. LDAP Loader pulls person information from 389 LDAP server. We schedule this in cron to run every 30 minutes to update grouper group memberships. Given the history of our identity management environment and as we transition to this new architecture it is necessary for us to have this mechanism. The utility will grow into a tool to import and manage data from a variety of sources.

Grouper Rules: We have setup rules in Grouper to move person group membership as their status changes from active to inactive and vice versa. We do this because we manage service provisioning and other access via Grouper and these rules handle issues like delayed status changes due to our business rules for identity management.

Singleton Loader: Utiltiy written in perl that reconciles grouper group membership for single user. It determines what groups user should be member of based on user data in 389 LDAP server and updates group memberships using grouper web services. The utility augments the LDAP Loader with near realtime group updates instead of the 30 minute delay of the LDAP Loader.

ChangeLogConsumer: This publishes any grouper group membership and privileges events to dispatcher queue on AMQ server.

Grouper Dispatcher: It is written in Java and reads messages in the dispatcher queue and puts a copy of the message into different queues based on configured rules in a properties file. Allowing us to filter groups to AD and other services. For example: course groups are not sent to AD and would only be allowed to specific AMQ queues where permission from the registrar has been given to allow the service access to course groups.

Grouper-ActiveMQ-Provisioner (GAP): GAP is written in perl. We run three instances of GAP and configure it differently to provision grouper groups in 389, AD and isMemberOf attribute in 389. GAP also supports batching changelog messages to provision large groups. All the code is publicly available on git https://github.com/cmu-ids/Grouper-ActiveMQ-Provisioner

Group Recon: This is ActiveMQ client written in Java that reads grouper changelog messages and updates group memberships within OIM. Group membership changes within OIM triggers provisioning to various target systems like Kerberos, NetReg, Cyrus etc.

More Info

You can also learn about the Grouper deployment at Carnegie Mellon University, as reported on the March 13, 2013 IAM Online webinar.

Link to the archived webinar (also featuring campus case studies from University of Montreal and University of Wisconsin-Madison)

PDF file from the IAM Online webinar

CMU Computing Services webpage on Grouper