Post PSP Provisioning

Post PSP Provisioning

Following on from the discussion on the July 3, 2013 Grouper Dev call and the July 29, 2014 and the August 27, 2014 Grouper Dev call, this page captures thoughts about the future of the PSP and directions for provisioning strategy. General consensus:

- The PSP is weighted heavily towards trying to be all things to all people.
- Due to the above, it suffers horribly in the area of performance.
- The community believes that messaging and more specifically a queue is the future direction that they all need to move in
- Building on the message-queue concept, Grouper should create a generic interface and message format that would translate changelog events or events triggered by hooks in Grouper into generic messages.
- Grouper would also create an interface to read/validate messages
- The community and project would use this message format and interface to write provisioners.
- The two messaging systems that the Grouper project will launch with support for are ActiveMQ and Amazon SQS
- Grouper will also attempt to write an implementation of the interface that does not rely on an external queue so that adopters who do not wish to run AMQ or connect up to Amazon could still run the provisioners locally.

General Requirements

General requirements moving forward for message queue readers:

- Support incremental provisioning by reading events off the message queue
- Support bulk reconciliation. Note: they may not be the same executable, but should retain and reuse code and configs where possible/practical.
  - When doing bulk-reconciliation the provisioner shall support optionally being pointed at a particular group or folder to reconcile
- Each module should be capable of being run in parallel with other modules
- All provisioning modules will follow the Grouper configuration paradigms with properties overlays and expression language.
- Modules shall provision based either on the location of a group in the tree as done today or using an attribute that will instruct the provisioner to provision the group and/or how to provision the group

Message Format

This still needs to be worked out.

Deployer	Message Format Documentation
Washington	https://wiki.cac.washington.edu/display/infra/Groups+AWS+Event+Messaging https://wiki.cac.washington.edu/display/infra/IAM+AWS+messaging
Carnegie Mellon	https://github.com/cmu-ids/Grouper-ActiveMQ-Provisioner
Grouper ESB	link to grouper ESB

- JSON will be the message format.
- Payload is JSON too
- Messages should always be signed (using JOSE JWS) – Config option to turn off signing, but leave on by default.
- Messages could be optionally encrypted (JOSE JWE)
- Messages will always carry their own metadata and consumers will not rely on the messaging system metadata but rather the grouper-produced metadata (this will keep our messages messaging-system-agnostic)
- When messages get too big, the message is chunked and a new message will be created for additional info

What's in a message

- All info about a group (name, desc, etc)
- Incremental information (adds / deletes)

Message Format Detail

First Implementations

Prospective first implementations for Grouper 2.3 are:

- LDAP
- Active Directory

Configuration

The provisioning modules will be configured via properties files using standard grouper configuration mechanisms. Modules will be activated by either calling them from GSH for batch/reconciliation functions or either a changelog consumer or Grouper hooks for incremental provisioning. Hooks will be investigated as a means of provisioning more quickly. The provisioners will support two kinds of configuration:

Global Configuration

Global configuration of the modules will be done by properties files specific to all instances of that module. Some of the envisioned properties set in these files include

- Hostnames / connection endpoints of provisioned systems
- Authentication information for provisioned systems
- Logging options
- Scheduling for running batch reconciliations
- Conflict handling
- Thresholds to prevent a provisioner from performing a change that would affect N% of users in a group
- Transformation instructions for turning group names or membership names into a format that the downstream system expects

Categorical Configuration

The idea behind categorical configuration is that rather than decorate a specific group to be provisioned to a specific target endpoint, we create an abstraction capability. The idea is that a group could receive the provisioning decoration of `standard` which would signal the downstream provisioners looking for `standard` to provision their targets accordingly. In this manner, a Group Admin, knowing that `standard` meant provision to LDAP, AD, and Google Apps, for instance, could just apply that one attribute & be done with configuring the provisioning. Categorical implementation would likely take the form of a group attribute with some metadata explaining which targets to hit.

Group-Level Configuration

This needs to be re-thought in light of the message consumers. Do we put the attributes into the message to inform the downstream consumers about it or do we require the consumers to call-back into grouper for additional data. My gut says we should do our level best to ensure a message has all the data a consumer would need to keep speed up and callbacks down.

Group level configuration would be handled by attributes placed upon groups or folders. The following standard attributes are envisioned:

- $NameProvision – when set on a group, that provisioner will provision the memberships to the target system. When set on a folder, all groups under that folder will be provisioned to the downstream system.

Specific provisioners may specify additional attributes they will use to determine how to provision that group's membership. Some examples are:

- Attribute to describe where in the LDAP DIT the group shall live. If this is set, it will override any global configuration.
- Attribute to describe the type of provisioning (members of this group shall represent Google Apps Organizations vs Google Apps Groups)
- Customized provisioning thresholds (for this group, refuse to provision if more than N% of the membership is affected)

The Grouper UIs will also be updated to facilitate managing of these attributes.