What’s new with Grouper
What’s new with Grouper - Agenda

- Rules
- Federated users
- CMU permissions users
- uPortal integration
- Demo of Penn Grouper/Rice workflow
Grouper rules
Rules architecture

Object owner (group, stem, membership, etc)

Rule owner attributes
- ID
- Name
**Rules architecture - continued**

1. Event happens in Grouper
   - e.g. membershipRemove

2. Notifies rules engine to look for rules
   
   - Rules periodically refreshed from DB
   - Rules daily validated in DB
   - Rules immediately refreshed if edited in same VM

3. Rules engine finds by hash the rules which match the “check” clause against valid rules in memory

4. Subset of matching rules have their “If” clause evaluated

- Eligible rules are run periodically always (e.g. email when about to expire)
- Eligible rules are run periodically to fix data corruptions or inconsistencies
Rules architecture - continued

Subset of matching "Check" and "If" eligible rules have their "Then" clauses executed, in the context of the rule's "ActAs" subject
Rules vs. Hooks vs. Loader

• Hooks are Java plugins
• Loader is cron based on SQL (or soon LDAP)
• Rules are like hooks
  – Rules don’t require Java
  – Rules can be configured by end users
  – Rules do not have full access to API
  – Rules can run as a daemon too
Why do we need rules?

• Quickly configure automatic actions in Grouper
• End users can control their own rules
If a subject is removed from the employee group, it will be removed from the PayrollSystemUsers group

Nightly remove subjects from PayrollSystemUsers who are not employees
Rules user case - Composite-ng intersection permissions

• If a subject is removed from the employee group, all payroll system permissions will be removed (e.g. READ Arts and Sciences payroll records)

• Nightly remove payroll permissions who are not employees
Rules user case - Composite-org intersection

• If a subject is removed from any group in a folder in part of the org structure, it will be removed from the IT admins group
• Nightly daemon
• “Composite org intersection permissions” is same for permissions
Rules user case – Disabled date activation

• If a student is removed from a group for English 110, then attach an end date on the English 110 wiki group for one week from now

• “Disabled-date permissions” is the same for permissions
Rules user case – Email notification

• Email notification on flattened membership add from stem
• Email notification on flattened membership remove
• Email notifications on disabled dates (will be removed 5 days from now)
• Email notifications permissions
• Email notifications permissions disabled dates
Rules user case – Inherited privileges

• Inherited privileges on attribute definitions
• Inherited privileges on folders
• Inherited privileges on groups
• Inherited privileges on groups with a name pattern
• E.g. all Groups under folder school:courses will have school:etc:courseReaders with READ privilege
Rules user case – Veto

- Veto if not eligible (if not in a Group)
- Veto if not eligible in org (if not in a Group in a stem)
- Veto permission if not eligible
Rules - configuration

- API
- API abbreviated (for use cases, one call sets all attributes)
- WS
- client
Grouper federated users
Federated users - description

• If your IdM supports federated users, use that
• If not, you can use Grouper federated (external) users
• Similar to comanage
• Stores some identity information in the Grouper DB (or pluggable storage, LDAP?)
Federated users - tables

```
grouper_ext_subj

Columns
- uuid, varchar(40)
- name, varchar(200), Nullable
- identifier, varchar(200), Nullable
- description, varchar(500), Nullable
- institution, varchar(200), Nullable
- email, varchar(100), Nullable
- search_string_lower, text, Nullable
- create_time, bigint(20)
- creator_member_id, varchar(40)
- modify_time, bigint(20)
- modifier_member_id, varchar(40)
- context_id, varchar(40)
- enabled, varchar(1)
- disabled_time, bigint(20), Nullable
- hibernate_version_number, bigint(20)
```

```
grouper_ext_subj_attr

Columns
- uuid, varchar(40)
- attribute_system_name, varchar(200)
- attribute_value, varchar(600), Nullable
- subject_uuid, varchar(40)
- create_time, bigint(20)
- creator_member_id, varchar(40)
- modify_time, bigint(20)
- modifier_member_id, varchar(40)
- context_id, varchar(40)
- hibernate_version_number, bigint(20)
```
Federated users - configuration

```java
externalSubjects.desc.el = ${grouperUtil.appendIfNotBlankString
    (externalSubject.name, ' - ', externalSubject.institution)}
externalSubjects.desc.manual = false

externalSubjects.calc.fields.cron = 0 0 3 * * ?

externalSubjects.name.required = true
externalSubjects.email.required = false
externalSubjects.email.enabled = true

externalSubjects.searchStringFields = name, institution, identifier,
    uuid, email, jabber

externalSubjects.institution.required = false
externalSubjects.institution.enabled = true

externalSubjects.attributes.jabber.friendlyName = Jabber ID
...
```
### Self registration for people external to this institution

**Register a new account**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Login ID</strong></td>
<td><a href="mailto:user1@school.edu">user1@school.edu</a></td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Institution</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Department and title</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Email</strong></td>
<td></td>
</tr>
</tbody>
</table>

* indicates a required field

[Image of registration form]

### Notes
- This form is for users who are not affiliated with the institution.
- Fill out the required fields to create a new account.
- Use the 'Submit' button to finalize the registration.

[Image of Internet2 logo]
Federated users – invite and provision external users

Invite external people to participate in groups

<table>
<thead>
<tr>
<th>Enter the email addresses and groups</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Email addresses of people to invite</strong></td>
</tr>
<tr>
<td><a href="mailto:someone@anotherSchool.edu">someone@anotherSchool.edu</a>,</td>
</tr>
<tr>
<td><a href="mailto:anotherPerson@someSchool.edu">anotherPerson@someSchool.edu</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Email subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invitation to the secure space collaboration application</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message to users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to the secure space collaboration application.</td>
</tr>
<tr>
<td>Regards, Steve Smith at University of Research</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Your email address</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:me@mySchool.edu">me@mySchool.edu</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Groups to assign to new users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter search text to find a group</td>
</tr>
<tr>
<td>stem: someGroup</td>
</tr>
<tr>
<td>Enter search text to find a group</td>
</tr>
<tr>
<td>stem: anotherGroup</td>
</tr>
<tr>
<td>Enter search text to find a group</td>
</tr>
<tr>
<td>Enter search text to find a group</td>
</tr>
<tr>
<td>Enter search text to find a group</td>
</tr>
<tr>
<td>Enter search text to find a group</td>
</tr>
<tr>
<td>Enter search text to find a group</td>
</tr>
</tbody>
</table>

Submit
Federated users – admin interface

• We will make a UI for admins to query external users, add, edit, delete
• Current this can be done with GSH/SQL
Federated users – logging and notification

- Grouper admins can be notified when registrations occur
- Inviters can be notified when invitees register
- All accesses are audited
Grouper / Kuali Rice integration at Penn
Kuali Rice overridable services

Rice request

Rice server

Rice server

Grouper client.jar

GrouperWS server

GrouperRegistry

Kuali DB

GrouperRice.jar

Grouper.client.properties
Rice Grouper workflow

1. Initiator starts workflow
   - On login to Rice, get subject details

2. Routes to approver group
   - One in group approves
   - Get members to route

3. Routes to approver group2
   - Add a member to a Grouper group/role and/or assign permissions

4. Final
   - Archive the document data, and workflow history
Grouper / Kuali Rice progress at Penn

- Piloting a dozen eForms in production
- Planning a rollout of more forms
- Central IT will convert paper forms to eForms as a for-fee service for schools and centers (details not final)
- Demo movie
- Will have a more detailed presentation for the Kuali Days conference
Grouper / uPortal integration
Grouper / uPortal integration

- Similar to Kuali Rice integration
- Secure portlets to Grouper groups
- University of Chicago initially developed a connector
- The Grouper team and Unicon are extending it
- In progress...
CMU billing permissions use case
CMU billing permissions use case

• CMU has a billing use case on the paccman wiki

• Involves Groups / Roles / Permissions
  – Can view own bill
  – Can delegate access to bill to someone else
  – Department admins can view bills of students in department
  – Billing admins can view all bills
• Fully implemented on the Grouper 1.6.1 demo server
• Command line client which runs all logic over web services using the grouper client
• Test data is loaded in server
• All commands to setup the permissions / roles / data are documented on the paccman wiki
CMU billing permissions use case (continued)

- One gap is centrally hosting the decision logic (decision point)
- Grouper could have a generic Java pluggable decision point to centrally manage logic over Grouper WS
What’s new with Grouper

1-Nov-2010, Fall Member Meeting
Chris Hyzer, Grouper developer

For more information, visit www.internet2.edu