

The SFU logo consists of the letters 'SFU' in white, bold, sans-serif font, set against a solid red rectangular background.

SIMON FRASER UNIVERSITY  
ENGAGING THE WORLD



# Grouper At SFU

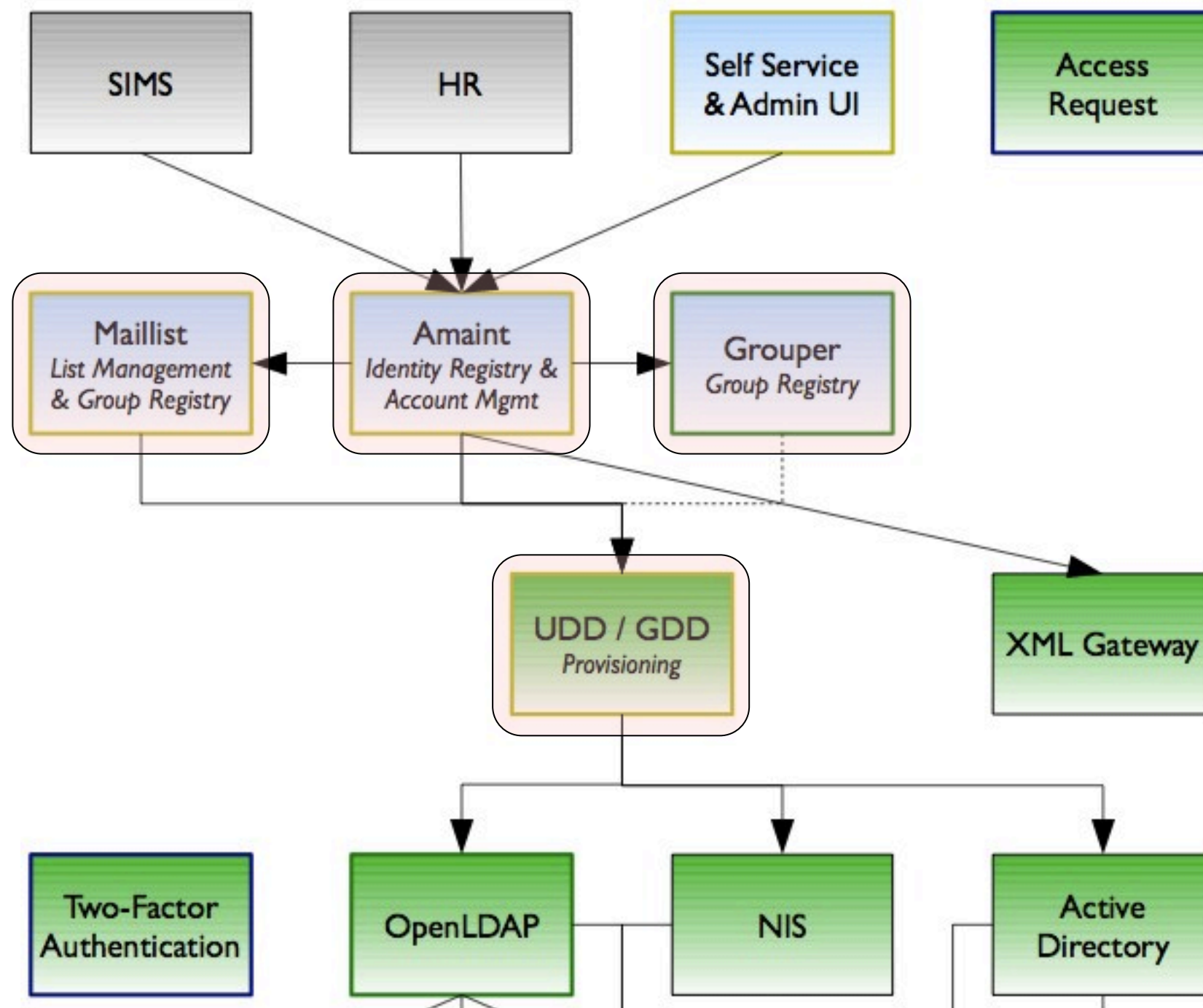
Rob Urquhart

June 2012



- Dig a little deeper into Grouper at SFU
  - How we get data into grouper
    - Bulk loading student groups
  - How we get data out of grouper
    - Provisioning to LDAP

## Overview



# SFU IdM Architecture

- Auto-create course groups
  - student groups for each course + section
    - E.g. econ103-d100-students
  - student groups for each course
    - econ103-students
  - Instructor groups for each course + section
    - econ103-d100-instructors
  - Instructor groups for each course
    - econ103-instructors

## Use Case

- Automatically sync the Grouper groups with SOR at least once a day
- Departments can manually add/delete members
  - Need access control on groups
- Create groups for next semester as soon as data is available

## Use Case

- Provision into multiple downstream systems
  - LDAP/AD
  - Maillist (our email list manager)

## Use Case

- Could create groups manually using UI
  - Too many groups
- Could write custom code to create/sync groups with SOR
  - Use Web Services api
  - Want to minimize home-grown systems
- GrouperLoader!

## Getting Data Into Grouper

- Process which can be run manually or as a daemon
- Looks for special groups of type “grouperloader”
- grouperloader groups know how and when to update their members

## GrouperLoader



- Groups can be assigned multiple types:

Type	Attributes for Type
base	members
grouperLoader	grouperLoaderQuery, etc
addIncludeExclude	

# GrouperLoader

- Simple grouperLoader group

Current location is:

Root: resource: undergrads

<u>Name</u>	undergrads																												
<u>Path</u>	resource:undergrads																												
<u>Description</u>	all undergrad students																												
<u>ID</u>	undergrads																												
<u>ID Path</u>	resource:undergrads																												
<u>Alternate ID Path</u>																													
<u>UUID</u>	9c28e1a116b04938b14d702d60f965fd																												
<u>Types</u>	<table><tr><td>base</td><td>members</td><td>List field</td></tr><tr><td rowspan="11">grouperLoader</td><td>grouperLoaderAndGroups</td><td></td></tr><tr><td>grouperLoaderDbName</td><td>grouper</td></tr><tr><td>grouperLoaderGroupQuery</td><td></td></tr><tr><td>grouperLoaderGroupsLike</td><td></td></tr><tr><td>grouperLoaderGroupTypes</td><td></td></tr><tr><td>grouperLoaderIntervalSeconds</td><td></td></tr><tr><td>grouperLoaderPriority</td><td></td></tr><tr><td>grouperLoaderQuartzCron</td><td>0 0 6 * * ?</td></tr><tr><td>grouperLoaderQuery</td><td>select amaint.grouper_view_2.subjectID as subject_id from amaint.undergrads, amaint.grouper_view_2 where amaint.undergrads.sfuid=amaint.grouper_view_2.sfuid</td></tr><tr><td>grouperLoaderScheduleType</td><td>CRON</td></tr><tr><td>grouperLoaderType</td><td>SQL_SIMPLE</td></tr></table>			base	members	List field	grouperLoader	grouperLoaderAndGroups		grouperLoaderDbName	grouper	grouperLoaderGroupQuery		grouperLoaderGroupsLike		grouperLoaderGroupTypes		grouperLoaderIntervalSeconds		grouperLoaderPriority		grouperLoaderQuartzCron	0 0 6 * * ?	grouperLoaderQuery	select amaint.grouper_view_2.subjectID as subject_id from amaint.undergrads, amaint.grouper_view_2 where amaint.undergrads.sfuid=amaint.grouper_view_2.sfuid	grouperLoaderScheduleType	CRON	grouperLoaderType	SQL_SIMPLE
base	members	List field																											
grouperLoader	grouperLoaderAndGroups																												
	grouperLoaderDbName	grouper																											
	grouperLoaderGroupQuery																												
	grouperLoaderGroupsLike																												
	grouperLoaderGroupTypes																												
	grouperLoaderIntervalSeconds																												
	grouperLoaderPriority																												
	grouperLoaderQuartzCron	0 0 6 * * ?																											
	grouperLoaderQuery	select amaint.grouper_view_2.subjectID as subject_id from amaint.undergrads, amaint.grouper_view_2 where amaint.undergrads.sfuid=amaint.grouper_view_2.sfuid																											
	grouperLoaderScheduleType	CRON																											
	grouperLoaderType	SQL_SIMPLE																											

# GrouperLoader

- Simple grouperLoader group

Types	base	members	List field
	<b><u>grouperLoader</u></b>	<b><u>grouperLoaderAndGroups</u></b>	
		<b><u>grouperLoaderDbName</u></b>	grouper
		<b><u>grouperLoaderGroupQuery</u></b>	
		<b><u>grouperLoaderGroupsLike</u></b>	
		<b><u>grouperLoaderGroupTypes</u></b>	
		<b><u>grouperLoaderIntervalSeconds</u></b>	
		<b><u>grouperLoaderPriority</u></b>	
		<b><u>grouperLoaderQuartzCron</u></b>	0 0 6 * * ?
		<b><u>grouperLoaderQuery</u></b>	select amaint.grouper_view_2.subjectID as subject_id from amaint.undergrads, amaint.grouper_view_2 where amaint.undergrads.sfuid=amaint.grouper_view_2.sfuid
		<b><u>grouperLoaderScheduleType</u></b>	CRON
		<b><u>grouperLoaderType</u></b>	SQL_SIMPLE

# GrouperLoader

- SQL\_GROUP\_LIST groups

Types	base	members	List field
	<u>grouperLoader</u>	<u>grouperLoaderAndGroups</u>	
		<u>grouperLoaderDbName</u>	grouper
		<u>grouperLoaderGroupQuery</u>	select group_name, readers, updaters from amaint.grouper_course_group_query_v where semester='1124'
		<u>grouperLoaderGroupsLike</u>	
		<u>grouperLoaderGroupTypes</u>	addIncludeExclude
		<u>grouperLoaderIntervalSeconds</u>	
		<u>grouperLoaderPriority</u>	
		<u>grouperLoaderQuartzCron</u>	0 5 4 * * ?
		<u>grouperLoaderQuery</u>	select group_name, subject_id from amaint.grouper_course_list_v where semester='1124'
		<u>grouperLoaderScheduleType</u>	CRON
		<u>grouperLoaderType</u>	SQL_GROUP_LIST

# GrouperLoader

- grouperLoaderGroupQuery fetch:
  - returns name of the group
  - list of groups that can read the group
  - list of groups that can write the group

course:1124:econ:103:d102:students\_systemOfRecord;  
course:security:readers,course:security:econ:readers;  
course:security:updaters,course:security:econ:updaters

# GrouperLoaderGroupQuery



- SQL\_GROUP\_LIST groups

Types	base	members	List field
	<u>grouperLoader</u>	<u>grouperLoaderAndGroups</u>	
		<u>grouperLoaderDbName</u>	grouper
		<u>grouperLoaderGroupQuery</u>	select group_name, readers, updaters from amaint.grouper_course_group_query_v where semester='1124'
		<u>grouperLoaderGroupsLike</u>	
		<u>grouperLoaderGroupTypes</u>	addIncludeExclude
		<u>grouperLoaderIntervalSeconds</u>	
		<u>grouperLoaderPriority</u>	
		<u>grouperLoaderQuartzCron</u>	0 5 4 * * ?
		<u>grouperLoaderQuery</u>	select group_name, subject_id from amaint.grouper_course_list_v where semester='1124'
		<u>grouperLoaderScheduleType</u>	CRON
		<u>grouperLoaderType</u>	SQL_GROUP_LIST

# GrouperLoader

- grouperLoaderQuery fetch:
  - name of the group
  - subject ids

course:1124:econ:101:d100:students\_systemOfRecord;  
jsmith

course:1124:econ:101:d100:students\_systemOfRecord;  
rjones

course:1124:econ:102:d101:students\_systemOfRecord;  
rjones

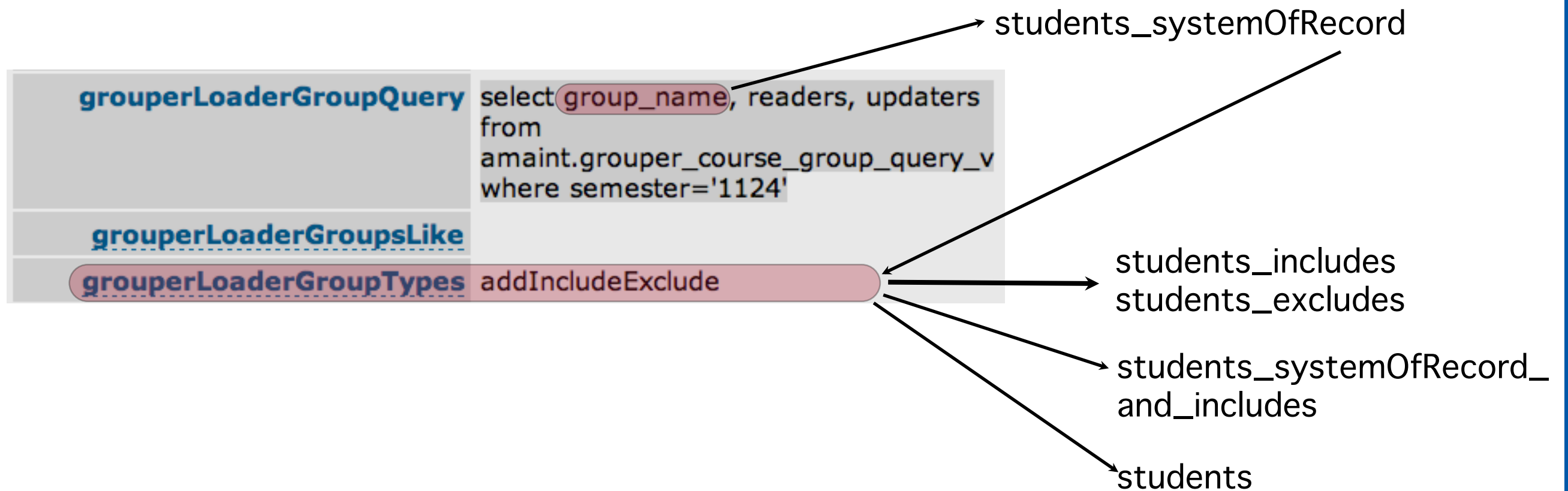
# GrouperLoaderGroupQuery

- SQL\_GROUP\_LIST groups

Types	base	members	List field
	<u>grouperLoader</u>	<u>grouperLoaderAndGroups</u>	
		<u>grouperLoaderDbName</u>	grouper
		<u>grouperLoaderGroupQuery</u>	select group_name, readers, updaters from amaint.grouper_course_group_query_v where semester='1124'
		<u>grouperLoaderGroupsLike</u>	
		<u>grouperLoaderGroupTypes</u>	addIncludeExclude
		<u>grouperLoaderIntervalSeconds</u>	
		<u>grouperLoaderPriority</u>	
		<u>grouperLoaderQuartzCron</u>	0 5 4 * * ?
		<u>grouperLoaderQuery</u>	select group_name, subject_id from amaint.grouper_course_list_v where semester='1124'
		<u>grouperLoaderScheduleType</u>	CRON
		<u>grouperLoaderType</u>	SQL_GROUP_LIST

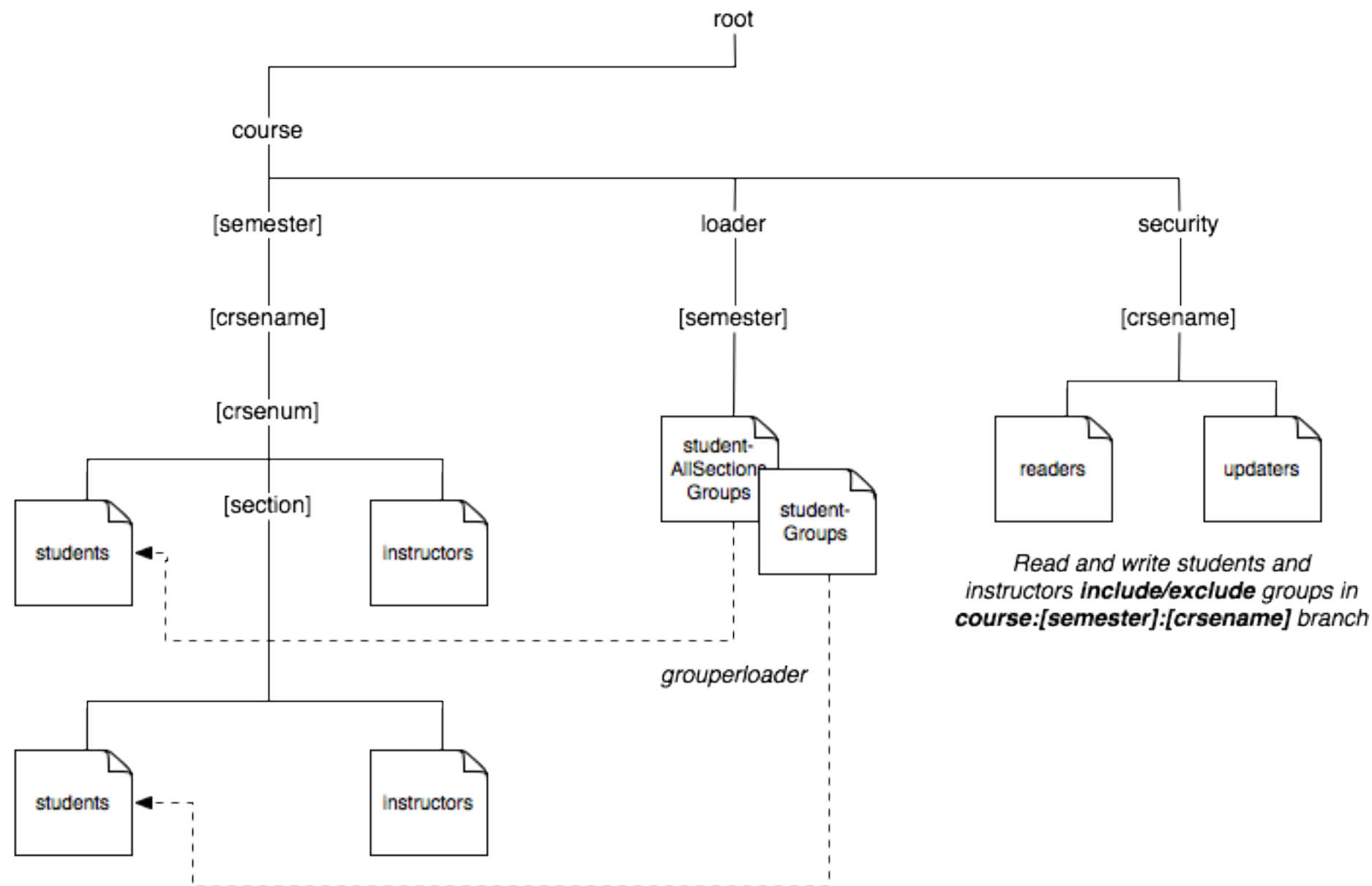
# GrouperLoader

- addIncludeExclude groups



# GrouperLoader

## Course Group Data Structure Grouper Phase I



# SFU Grouper Data Structure



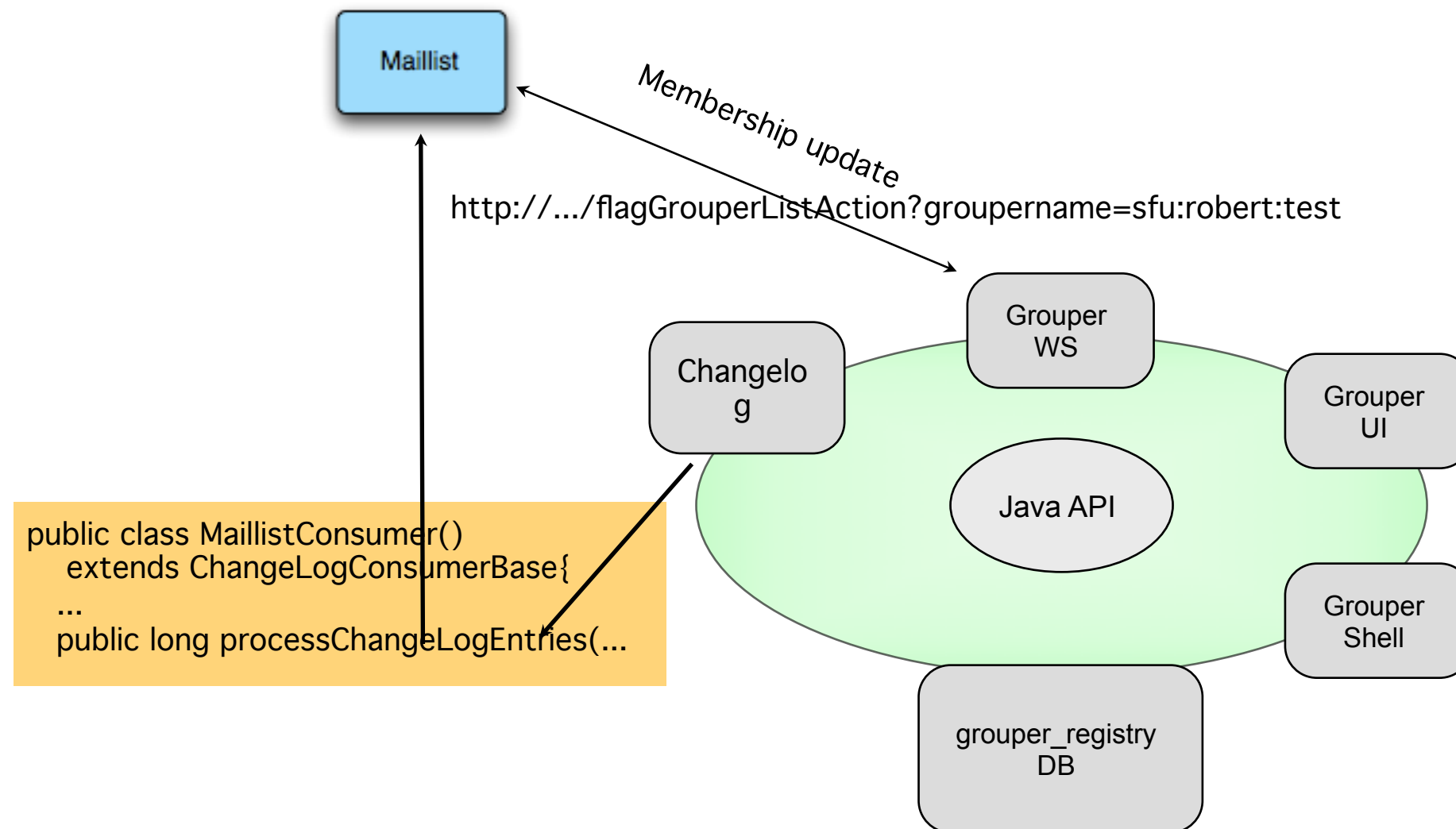
- How do we get data out of Grouper?
  - WS and REST APIs
    - Periodically poll and fetch data from grouper
    - Permissions API
  - Ldapdc-ng
    - provisioning connector for LDAP and AD
    - Version 2.0 doesn't do incremental updates
  - Custom solution using Changelog Connector
  - Custom solution using ESB Connector

## Provisioning From Grouper

- Every change in grouper is posted to the changelog
- Write a custom ChangeLog Consumer
- Every minute your consumer's processChangeLogEntries method is run
  - all the changes from the past minute
  - full control over what changes you want to handle

## The Grouper Changelog

- MaillistConsumer notifies maillist of group change



# The Grouper Changelog

- ESB ChangeLog Connector
  - provides a JSON event message for each changelog entry
  - Write custom consumer
    - Receives each event message
    - Posts each message to a JMS queue
    - Apache Camel handles routing of messages

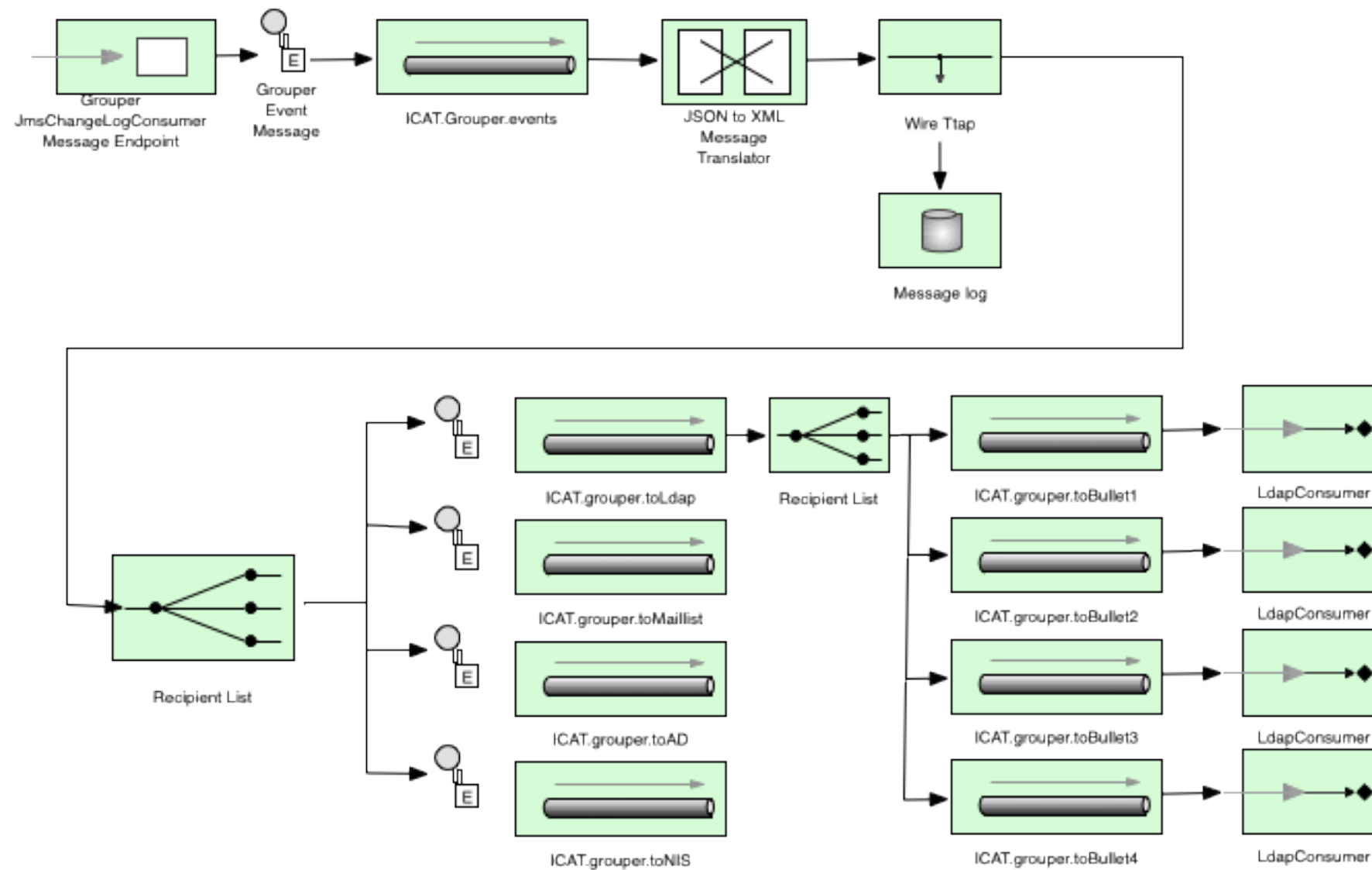
## ESB Consumer for JMS

- Camel handles message conversion
- Routes messages to multiple queues
  - LDAP
  - AD
  - NIS
- Advantages
  - Routing rules in one place - Camel config
  - Producer doesn't need to know anything about consumers

## Camel Routes



## Prod Camel Routes for Grouper LDAP Detail



# Camel Routes

- New course email lists based on Grouper groups
- Only internal use -- no end-users
- Camel provisioning to LDAP in test

## Pilot Status

- Documentation is not bad
  - but I had to go over it multiple times
- Make notes while you install and configure test installation
- Version 2.1 has a quickstart install which should make things easier
- Very stable
- Performance is good
  - dual gouper-ui and grouper-ws processes running on tomcat with shared db

## Installation Experience

Q & A