

# Univ of Washington's experiences with messaging

<a href="#">Wiki Home</a>	<a href="#">Grouper Release Announcements</a>	<a href="#">Grouper Guides</a>	<a href="#">Grouper Deployment Guide</a>	<a href="#">Community Contributions</a>	<a href="#">Internal Developer Resources</a>
---------------------------	---	--------------------------------	--	---	--

## Introduction

This describes some experiences with group event notification at the University of Washington. Our group service uses messaging for a variety of purposes:

1. Membership and other events from our Groups Web Service (GWS) to subordinate group systems: Google groups, Active Directory, local caches, and etc.
2. Course enrollment events from Student Services to the GWS. This allows students to enroll in a section and immediately use web services that rely on course group membership.
3. Multiple-system synchronization. Two or more independent installations of GWS keep in step by sending each other change notices. This is principally a way to make major Grouper upgrades without interruption of service.

In all cases the message content is essentially the same resource representation as one GETs from the web service.

We encrypt all messages.

Periodic reconciliation 'finds' any lost events.

## Early experiences with Apache ActiveMQ

For a couple of years we used an Apache ActiveMQ message broker to deliver updates to subordinate group caches. GWS published messages to a durable subscription (topic) on the broker. Clients subscribed to this subscription. This queue processed 10 to 50 thousand messages per day.

- The ActiveMQ system, because it persists unclaimed messages forever, can fill it's disks. When that happens the queue stop working and notices are lost. This happened to us several times
- We accessed ActiveMQ through openwire. There are libraries for some languages, although we had to adapt the c++ one for our c code. I found the protocol and interface difficult to program to.
- Our persistent connections would occasionally hang. We had to add heartbeats to the system.
- Somebody has to maintain an ActiveMQ system. It is not trivial.

## Experience with Amazon Web Services

### Course enrollment

For several years we have used Amazon's Simple Notification Service (SNS) to distribute course enrollment events. Our Student Web Service (SWS) gets enrollment changes directly from the source database (some trigger mechanism), accumulates these for a few minutes, and sends them to an SNS topic (5 events per SNS message). Content of these messages is similar to the resource representations that one would GET from SWS.

- Messages to an SNS topic can be delivered directly to a subscribing service, such as an http or https endpoint, or can be delivered to an SQS queue, where they can be retrieved at will by a client.
- Because notices are often (at quarter start) generated a lot faster than GWS can process them GWS uses an SQS queue to collect the messages. It retrieves them from the queue sequentially and updates the corresponding course group memberships.
- This queue processes a few hundred to 10 thousand messages per day.

### Group changes

For a couple of years we have distributed our GWS change notices via AWS messaging--switching the service from the local ActiveMQ service.

- AWS provides RESTful API access to all their services. These are well documented and easy to use. The management console is easy to use as well.
- It has been very reliable.
- AWS messages are limited to about 128K bytes. This necessitates some chunking of longer membership change notices.
- AWS messages can get out of order, and may be delivered more than once. I've not seen the latter.
  - If the messages are resources, with last update times, these issues are easily dealt with. If a message arrives with a timestamp earlier than the resource's update time it can be ignored. I've found this easy to implement.
  - Some of my colleagues instead pool messages for a time to reorder them. So the sequencing issue is a greater problem than I thought it would be.
- We process about a million messages per month.
- Azure supposedly has a message queue that appears to combine the REST API with message ordering. Investigating that.

## References

- Our xxxx: [Groups AWS Event Messaging](#)
- Generic message syntax: [IAM AWS messaging](#)
- More detail on our group service: [UW Groups Service 2.0 Architecture](#)