Grouper Box integration

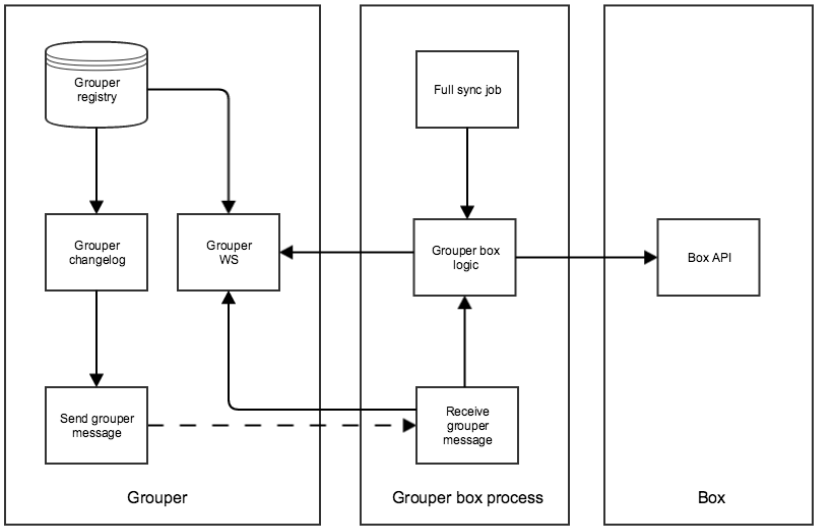
Grouper can provision a folder of Grouper groups with groups at box.com

Why use this?

You might want to share resources at box with departments or other groups in Grouper

Architecture

Box groups must be directly in a designated folder. This is one way it could work, using messaging

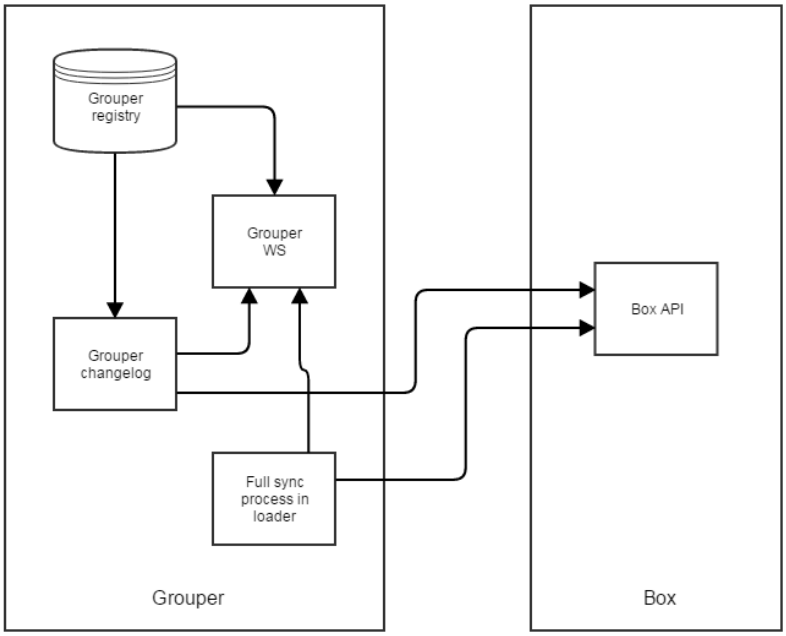


The box API WS has the power to do a lot, so your box support team might want to run that on their servers, not the central grouper machines.

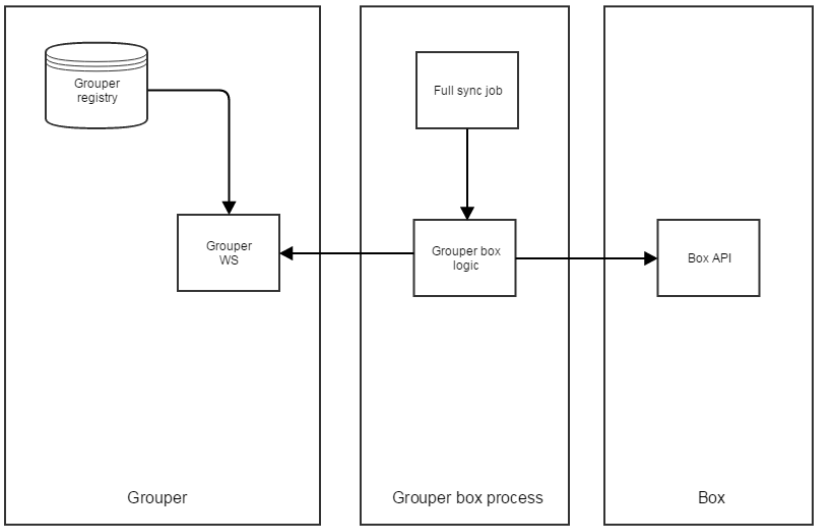
Only MEMBER roles are managed in box, not ADMIN roles (note, you dont need ADMIN roles in box)

The Grouper box process is a handful of jars including the grouper client which runs as a unix process

Here is another way it could work if you want to run it in your loader



If you have an old version of Grouper you can just run the full sync and not use messaging. Note, if you want a change log consumer we might be able to get that working in the future



Notes

If you grant a person to be a Box group admin, they can edit users too, generally this is not a good idea, but this integration protects you from that.

If you provision users just in time, you might want to run a SAML attribute assertion that puts people in groups just in time.

Setup integration (from README.txt)

Make a messaging queue, grant access to the box user (in this case keep it simple and use GrouperSystem, though you would make your own user and not user GrouperSystem

```
GrouperSession grouperSession = GrouperSession.startRootSession();
GrouperBuiltinMessagingSystem.createQueue("box_queue");
Subject subject = SubjectFinder.findById("GrouperSystem");
GrouperBuiltinMessagingSystem.allowSendToQueue("box_queue", subject);
GrouperBuiltinMessagingSystem.allowReceiveFromQueue("box_queue", subject);
```

Identify a folder in grouper which has box groups in it.

(optional) Identify a group that has all box users in it.

(Architecture #1 above) Configure a change log consumer to send messages to the box process in grouper-loader.properties. Note, if you are using messaging you dont have to add any jars or anything else to the loader. Note, you need to edit the elFilter carefully, and the subject attribute if not "email"

```
changeLog.consumer.boxEsb.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.EsbConsumer
changeLog.consumer.boxEsb.quartzCron = 0 * * * * ?
# carefully adjust this filter e.g. for sourceId and groupName
changeLog.consumer.boxEsb.elfilter = (event.sourceId == null || event.sourceId eq 'jdbc') && (event.groupName
=~ '^box\\:groups\\:.*$' || event.groupName eq 'box:boxUser' || event.name =~ '^box\\:groups\\:.*$' || event.
name eq 'box:boxUser') && (event.eventType eq 'GROUP_DELETE' || event.eventType eq 'GROUP_ADD' || event.
eventType eq 'GROUP_UPDATE' || event.eventType eq 'MEMBERSHIP_DELETE' || event.eventType eq 'MEMBERSHIP_ADD' ||
event.eventType eq 'MEMBERSHIP_UPDATE')
changeLog.consumer.boxEsb.publisher.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.
EsbMessagingPublisher
changeLog.consumer.boxEsb.publisher.messagingSystemName = grouperBuiltinMessaging
# queue or topic
changeLog.consumer.boxEsb.publisher.messageQueueType = queue
changeLog.consumer.boxEsb.publisher.queueOrTopicName = box_queue
# this is optional if not using "id" for subjectId, need to be a subject attribute in the sources.xml
changeLog.consumer.boxEsb.publisher.addSubjectAttributes = email
```

(Architecture #2 above) Configure a change log consumer to communicate with box in grouper-loader.properties. Note, if you are using this you have to add all box-grouper-connector jars to the loader and the client config. Note, you need to edit the elFilter carefully, and the subject attribute if not "email"

```
changeLog.consumer.boxEsb.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.EsbConsumer
changeLog.consumer.boxEsb.quartzCron = 0 * * * * ?
# carefully adjust this filter e.g. for sourceId and groupName
changeLog.consumer.boxEsb.elfilter = (event.sourceId == null || event.sourceId eq 'jdbc') && (event.groupName
=~ '^box\\:groups\\:.*$' || event.groupName eq 'box:boxUser' || event.name =~ '^box\\:groups\\:.*$' || event.
name eq 'box:boxUser') && (event.eventType eq 'GROUP_DELETE' || event.eventType eq 'GROUP_ADD' || event.
eventType eq 'GROUP_UPDATE' || event.eventType eq 'MEMBERSHIP_DELETE' || event.eventType eq 'MEMBERSHIP_ADD' ||
event.eventType eq 'MEMBERSHIP_UPDATE')
changeLog.consumer.boxEsb.publisher.class = edu.internet2.middleware.grouperBox.BoxEsbPublisher
# this is optional if not using "id" for subjectId, need to be a subject attribute in the sources.xml
changeLog.consumer.boxEsb.publisher.addSubjectAttributes = email
```

Test with add a member GSH script

```
GrouperSession grouperSession = GrouperSession.startRootSession();
Group group = new GroupSave(grouperSession).assignCreateParentStemsIfNotExist(true).assignName("box:groups:
someGroup").save();
Subject subject = SubjectFinder.findById("mchyzerGoogle");
group.addMember(subject, false);
GrouperLoader.runOnceByJobName(grouperSession, "CHANGE_LOG_changeLogTempToChangeLog");
GrouperLoader.runOnceByJobName(grouperSession, "CHANGE_LOG_consumer_boxEsb");
```

Test with delete a member GSH script

```
GrouperSession grouperSession = GrouperSession.startRootSession();
Group group = new GroupSave(grouperSession).assignCreateParentStemsIfNotExist(true).assignName("box:groups:
someGroup").save();
Subject subject = SubjectFinder.findById("mchyzerGoogle");
group.deleteMember(subject, false);
GrouperLoader.runOnceByJobName(grouperSession, "CHANGE_LOG_changeLogTempToChangeLog");
GrouperLoader.runOnceByJobName(grouperSession, "CHANGE_LOG_consumer_boxEsb");
```

At this point in (Architecture 1) above you should see messages in the grouper_message table.

Setup box authn token (public key web service security)

```
https://docs.box.com/docs/getting-started-box-platform
Chriss-MacBook-Air:box mchyzer$ openssl genrsa -aes256 -out private_key.pem 2048
Chriss-MacBook-Air:box mchyzer$ openssl rsa -pubout -in private_key.pem -out public_key.pem
sign up for two step authn in box if not SSO
make application in box: https://app.box.com/developers/services
1. Enterprise application
2. OAuth 2.0 with JWT (Server Authentication)
authentication type: server
2. user access: all users
3. scopes: manage users, manage groups
4. advanced features: none
5. note client_id
6. note client_secret
7. redirect uri: https://localhost
8. Save application and upload public key
9. Copy the Client ID value for the app and then paste into the Apps section of the Box admin console (to add a
new authorized app).
Note you may need to enable the app if you have set it for Enterprise-wide/All users access.
```

Configure

grouper.client.properties

```
# these are properties to add to grouperClient.properties

# put groups in here which go to box, the name in box will be the extension here
grouperBox.folder.name.withBoxGroups = box:groups


# put the comma separated list of sources to send to box
grouperBox.sourcesForSubjects = jdbc


# either have id for subject id or an attribute for the box username (e.g. netId)
grouperBox.subjectAttributeForBoxUsername = email


# if there is a require group that users must be in to be a user in box
grouperBox.requireGroup = box:boxUser


# how long to cache box users in the requireGroup in grouper
grouperBox.cacheGrouperUsersForMinutes = 60


# is grouper the true system of record, delete box groups which dont exist in grouper
# note, if you delete the box group, if it is recreated, then shares wont exist
grouperBox.deleteGroupsInBoxWhichArentInGrouper = false
```

```
# how long to cache "getAllUsers", which usually takes a minute to get tens of thousands of users
grouperBox.boxUserCacheMinutes = 10

#the quartz cron is a cron-like string.
# http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger
grouperBox.fullSync.quartzCron = 0 0 5 * * ?


# authentication settings for WS
# put pem encrypted in database, put 2k chars in each section
grouperBox.privateKeyContents_0 =
grouperBox.privateKeyContents_1 =
# if not putting pem in database, you can put it on the filesystem, list the filename
grouperBox.privateKeyFileName =
grouperBox.privateKeyPass =
grouperBox.publicKeyId =
grouperBox.enterpriseId =
grouperBox.clientId =
grouperBox.clientSecret =


# should log in the event log if no messages
grouperBox.logIfNoMessages = false


# messaging config for incremental changes, blank to use default
grouperBox.messaging.systemName = grouperBuiltinMessaging

# queueName is required for incremental provisioning
grouperBox.messaging.queueName = box_queue


# if you want to perform a full sync with each message received (note, assumes only applicable messages are
sent)
# note, will wait X 30? seconds, then mark subsequent messages as complete for those 30 seconds
grouperBox.fullSyncOnMessage = false


# note, this must be at least 5 seconds
grouperBox.fullSyncOnMessageWaitSeconds = 30


#the quartz cron is a cron-like string.
# http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger
# this defaults to every 30 seconds since the messaging long polls for 20 seconds.
grouperBox.incrementalSync.quartzCron = 0/30 * * * * ?


# if a user is not in the grouperBox.requireGroup group, then set the user's status to inactive,
cannot_delete_edit, or cannot_delete_edit_upload
# if this is blank then dont worry about it
# be careful that you dont lock out your admin account(s), whitelist below
grouperBox.statusDeprovisionedUsers =


# if a user is not in the grouperBox.requireGroup group, then set is_sync_enabled to false
grouperBox.deprovisionDisableSync = false


# if a user is in the grouperBox.requireGroup group, then set the user's status to active
# if this is blank then dont worry about it
grouperBox.statusUndeprovisionedUsers =


# if a user is in the grouperBox.requireGroup group, then set is_sync_enabled to true
grouperBox.undeprovisionEnableSync = false


# these could be administrative id's to never invalidate, comma separated
grouperBox.whitelistBoxIds = a@b.c, b@c.d
```

```
# quartz stuff
org.quartz.scheduler.instanceName = MyScheduler
org.quartz.threadPool.threadCount = 3
org.quartz.jobStore.class = org.quartz.simpl.RAMJobStore
```

Perhaps make a log4j.properties file

```
# add this to log4j.properties, adjust file path
log4j.appender.grouperBox                          = org.apache.log4j.DailyRollingFileAppender
log4j.appender.grouperBox.File                     = logs/grouperBox.log
log4j.appender.grouperBox.DatePattern              = '.'yyyy-MM-dd
log4j.appender.grouperBox.layout                   = org.apache.log4j.PatternLayout
log4j.appender.grouperBox.layout.ConversionPattern  = %d{ISO8601}: %m%n

#log4j.logger.com.box.sdk.BoxAPIResponse =
log4j.logger.edu.internet2.middleware.grouperBox.GrouperBoxLog =  DEBUG, grouperBox
log4j.additivity.edu.internet2.middleware.grouperBox.GrouperBoxLog = false
```

Install the full sync and message consumer

In the directory where your grouper.client.properties is, and grouper.client.base.properties, and log4j.properties, put all the jars from the grouper box tarball in the lib dir. Note, the Box client runs in Java7+. Run this with something like (for unix). Note, we can add scripts to make this a service.

```
nohup java -cp conf:lib/* edu.internet2.middleware.grouperBox.GrouperBoxSync > stdout.txt 2>&1 &
```

To run a full sync via command line:

```
/sbin/service grouperBox stop
java -cp conf:lib/* edu.internet2.middleware.grouperBox.GrouperBoxFullRefresh
/sbin/service grouperBox start
```

Run full sync (grouper 2.5+)

```
import edu.internet2.middleware.grouperBox.*;
import edu.internet2.middleware.grouper.app.loader.db.*;
import edu.internet2.middleware.grouper.app.loader.*;
GrouperSession grouperSession = GrouperSession.startRootSession();
BoxOtherJob boxOtherJob = new BoxOtherJob();
Hib3GrouperLoaderLog hib3GrouperLoaderLog = new Hib3GrouperLoaderLog();
OtherJobBase.OtherJobInput otherJobInput = new OtherJobBase.OtherJobInput();
otherJobInput.setGrouperSession(grouperSession);
otherJobInput.setHib3GrouperLoaderLog(hib3GrouperLoaderLog);
boxOtherJob.run(otherJobInput);
```

Call add member example in GSH

```
import edu.internet2.middleware.grouperBox.*;
java.util.Map<String, GrouperBoxGroup> allGroupsMap = GrouperBoxCommands.retrieveBoxGroups();
GrouperBoxGroup grouperBoxGroup = allGroupsMap.get("group_name_in_box");
String[] usernames = new String[1];
int i=0;
usernames[i++] = "user@school.edu";
for (String userName : usernames) {GrouperBoxUser grouperBoxUser = GrouperBoxCommands.retrieveBoxUser
(userName); grouperBoxGroup.assignUserToGroup(grouperBoxUser, false); }
```

Sample log

```
C:\temp\temp\grouper.box-2.3.0>c:\dev_inst\java7\bin\java -cp .;lib\* edu.internet2.middleware.grouperBox.
GrouperBoxSync
2016-10-26 05:32:48,579: [main] DEBUG GrouperBoxLog.boxLog(42) -  - method: grouperBoxSync, cronStringFull: 0 0
5 * * ?, scheduledFull: true, cronStringIncremental: 0/30 * * * * ?, scheduledIncremental: true, elapsedMillis:
239
2016-10-26 05:33:00,431: [MyScheduler_Worker-1] DEBUG GrouperBoxLog.boxLog(42) -  - method:
grouperReceiveMessages, messageSystemName: grouperBuiltinMessaging, messageQueueName: box_queue,
checkMessagesWsResultCode: SUCCESS, messageCount: 0, elapsedMillis: 423
2016-10-26 05:33:00,431: [MyScheduler_Worker-1] DEBUG GrouperBoxLog.boxLog(42) -  - method: incrementalSync,
successMessageCount: 0, waitMessageCount: 0, elapsedMillis: 425
2016-10-26 05:33:30,030: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method:
grouperReceiveMessages, messageSystemName: grouperBuiltinMessaging, messageQueueName: box_queue,
checkMessagesWsResultCode: SUCCESS, messageCount: 1, elapsedMillis: 28
2016-10-26 05:33:32,629: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: retrieveBoxGroups,
size: 3, elapsedMillis: 2518
2016-10-26 05:33:33,006: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: retrieveBoxUsers,
size: 5, elapsedMillis: 376
2016-10-26 05:33:33,334: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method:
assignUserToBoxGroup, userLoginId: mchyzer@gmail.com, groupName: someGroup, daemonType: incremental,
alreadyExisted: true, elapsedMillis: 326
2016-10-26 05:33:33,336: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: processMessage,
eventType: MEMBERSHIP_ADD, groupName: box:groups:someGroup, sourceId: jdbc, subjectAttributeBoxUsername: email,
username: mchyzer@gmail.com, boxUsername: mchyzer@gmail.com, boxUserExists: true, elapsedMillis: 3229
2016-10-26 05:33:33,346: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method:
grouperAcknowledgeMessages, numberOfIds: 1, acknowledgeType: mark_as_processed, messageSystemName:
grouperBuiltinMessaging, messageQueueName: box_queue, elapsedMillis: 5
2016-10-26 05:33:33,350: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: incrementalSync,
successMessageCount: 1, waitMessageCount: 0, elapsedMillis: 3347
2016-10-26 05:33:33,403: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method:
grouperReceiveMessages, messageSystemName: grouperBuiltinMessaging, messageQueueName: box_queue,
checkMessagesWsResultCode: SUCCESS, messageCount: 1, elapsedMillis: 50
2016-10-26 05:33:33,761: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: retrieveBoxGroups,
size: 3, elapsedMillis: 355
2016-10-26 05:33:34,125: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method:
assignUserToBoxGroup, userLoginId: mchyzer@gmail.com, groupName: someGroup, daemonType: incremental,
alreadyExisted: true, elapsedMillis: 362
2016-10-26 05:33:34,127: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: processMessage,
eventType: MEMBERSHIP_ADD, groupName: box:groups:someGroup, sourceId: jdbc, subjectAttributeBoxUsername: email,
username: mchyzer@gmail.com, boxUsername: mchyzer@gmail.com, boxUserExists: true, elapsedMillis: 721
2016-10-26 05:33:34,131: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method:
grouperAcknowledgeMessages, numberOfIds: 1, acknowledgeType: mark_as_processed, messageSystemName:
grouperBuiltinMessaging, messageQueueName: box_queue, elapsedMillis: 0
2016-10-26 05:33:34,134: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: incrementalSync,
successMessageCount: 1, waitMessageCount: 0, elapsedMillis: 781
2016-10-26 05:33:34,169: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method:
grouperReceiveMessages, messageSystemName: grouperBuiltinMessaging, messageQueueName: box_queue,
checkMessagesWsResultCode: SUCCESS, messageCount: 0, elapsedMillis: 31
2016-10-26 05:33:34,170: [MyScheduler_Worker-2] DEBUG GrouperBoxLog.boxLog(42) -  - method: incrementalSync,
successMessageCount: 0, waitMessageCount: 0, elapsedMillis: 32
2016-10-26 05:34:00,074: [MyScheduler_Worker-3] DEBUG GrouperBoxLog.boxLog(42) -  - method:
grouperReceiveMessages, messageSystemName: grouperBuiltinMessaging, messageQueueName: box_queue,
checkMessagesWsResultCode: SUCCESS, messageCount: 0, elapsedMillis: 69
2016-10-26 05:34:06,610: [MyScheduler_Worker-3] DEBUG GrouperBoxLog.boxLog(42) -  - method: incrementalSync,
successMessageCount: 0, waitMessageCount: 0, elapsedMillis: 6605
```