Grouper dependency custom jar strategy

When jars are hosted on maven repos, we are all good. When we have to edit a jar, this is the strategy.

1. Submit a bug/patch to the software (do this in any case)
2. Try to replace an interface or implementation in the jar

```
Suppose in Hibernate, there is this class “HelloWorldImpl” that is defective. The declaration of the
class is as such:

public class HelloWorldImpl implements HelloWorld {}

Rather than patching HelloWorldImpl directly in the JAR, do this instead:

public class GrouperHelloWorld implements HelloWorld {}

Leave the JAR. Leave the original Impl.

Then, do:

hibernateEngine.setHelloWorld(new GrouperHelloWorld());

So long as HelloWorld remains intact, you’d be fine. The trick is to find the actual setHelloWorld()
method,
and instructing hibernate how to use our impl, rather than the default. Keep the patches inside the
Grouper github repo. Update the Javadoc directly in the code.
```

3. If there is no way to replace an implementation, get the source, recompile, but change the package to a grouper specific package. e.g. from org.hibernate to edu.internet2.middleware.grouper.jarExt.org.hibernate, host on sonatype as a grouper jar for that software. Refactor grouper source to use that package
4. If it doesnt work to change the package, just compile the jar and host on sonatype