

Mike Conway - DICE ([michael\\_conway@unc.edu](mailto:michael_conway@unc.edu))

---

# Short Tutorials

Clients and API

---

---

# Obtaining Jargon and Building

---

- ❖ <https://github.com/DICE-UNC/jargon>
- ❖ `git clone https://github.com/DICE-UNC/jargon.git`
- ❖ `mvn clean install -Dmaven.test.skip=true`

# iRODS Maven Repo

❖ <https://github.com/DICE-UNC/DICE-Maven>

```
<repository>
  <id>dice.repository snaps</id>
  <name>dice.repository.snapshots</name>
  <url>https://raw.github.com/DICE-UNC/DICE-Maven/master/snapshots</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>true</enabled>
    <updatePolicy>always</updatePolicy>
    <checksumPolicy>warn</checksumPolicy>
  </snapshots>
</repository>
<repository>
  <id>dice.repository</id>
  <name>dice.repository</name>
  <url>https://raw.github.com/DICE-UNC/DICE-Maven/master/releases</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>false</enabled>
    <updatePolicy>always</updatePolicy>
    <checksumPolicy>warn</checksumPolicy>
  </snapshots>
</repository>
```

---

# iRODS Maven Artifacts

---

- ❖ Add dependencies and go!

```
<dependency>
  <groupId>org.irods.jargon</groupId>
  <artifactId>jargon-core</artifactId>
  <version>${jargon.version}</version>
</dependency>
<dependency>
  <groupId>org.irods.jargon</groupId>
  <artifactId>jargon-data-utils</artifactId>
  <version>${jargon.version}</version>
</dependency>
<dependency>
  <groupId>org.irods.jargon</groupId>
  <artifactId>jargon-conveyor</artifactId>
  <version>${jargon.version}</version>
</dependency>
<dependency>
  <groupId>org.irods.jargon</groupId>
  <artifactId>jargon-user-tagging</artifactId>
  <version>${jargon.version}</version>
</dependency>
```

---

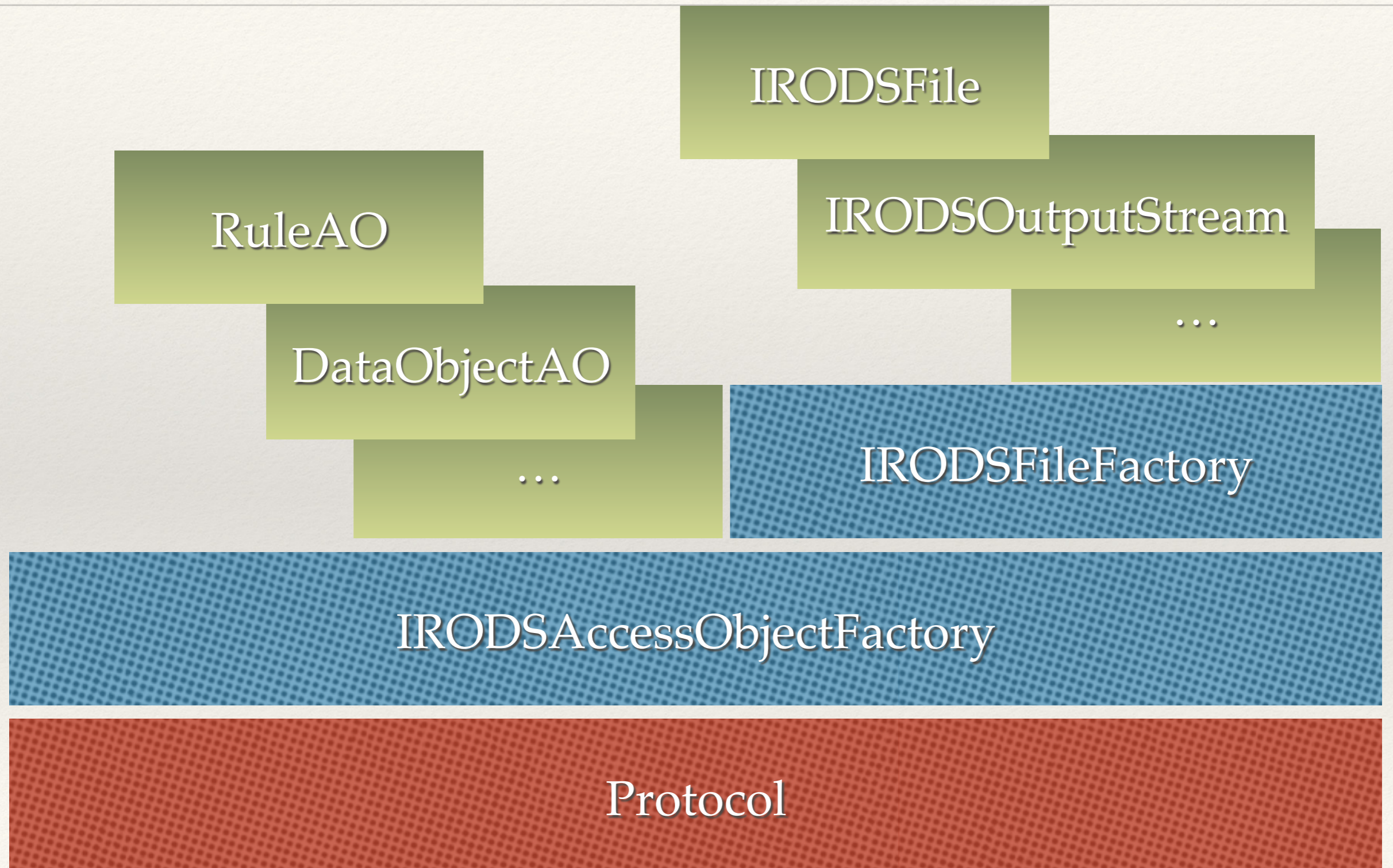
# Unit Testing

---

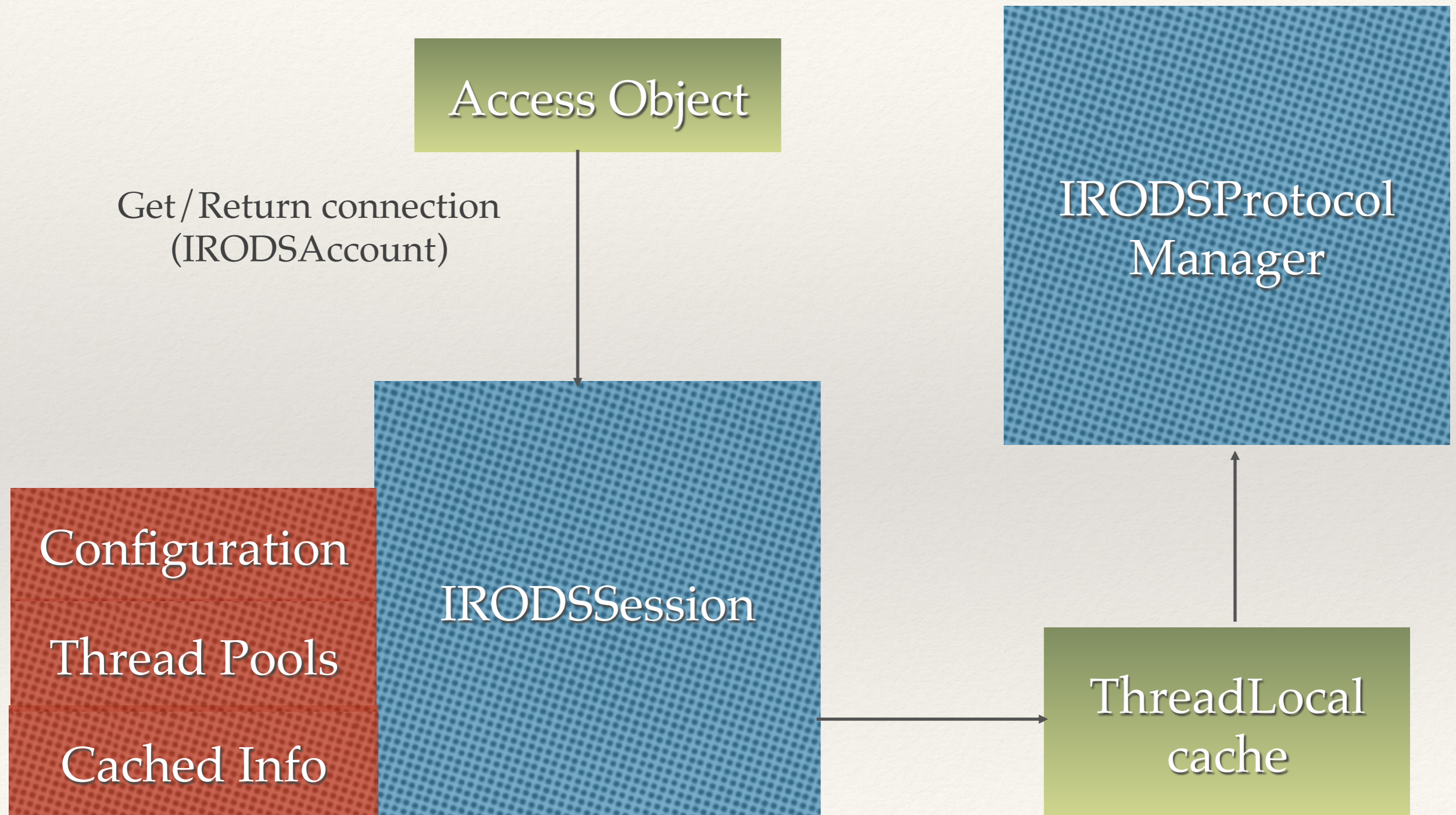
- ❖ Without the -D you will run tests, and they will fail!
- ❖ Tests rely on a pre-configured iRODS sandbox, as well as maven properties
- ❖ You can look here:  
<https://github.com/DICE-UNC/jargon/wiki/Setting-up-unit-tests>

How about some API basics?

# Jargon Layout



# Connection Handling Infrastructure





A few examples!

## Services

As we build integrations and API  
and interfaces, we find services  
to  
abstract out!

# iDROP Desktop

---

# What is it

---

- ❖ A desktop client
- ❖ A transfer manager
  - ❖ Build up 'work orders' for transfers
  - ❖ Leave it run, it keeps track and tells you what went wrong
  - ❖ It does synchronization
  - ❖ It has a rule engine (sort of)

# iDrop Demo, including flow manager

iDrop Web

---

# Updates

---

- ❖ Old version being phased out as a place-holder
- ❖ Look at branches and configurations
- ❖ New abstractions we talked about yesterday
- ❖ Preview and next steps