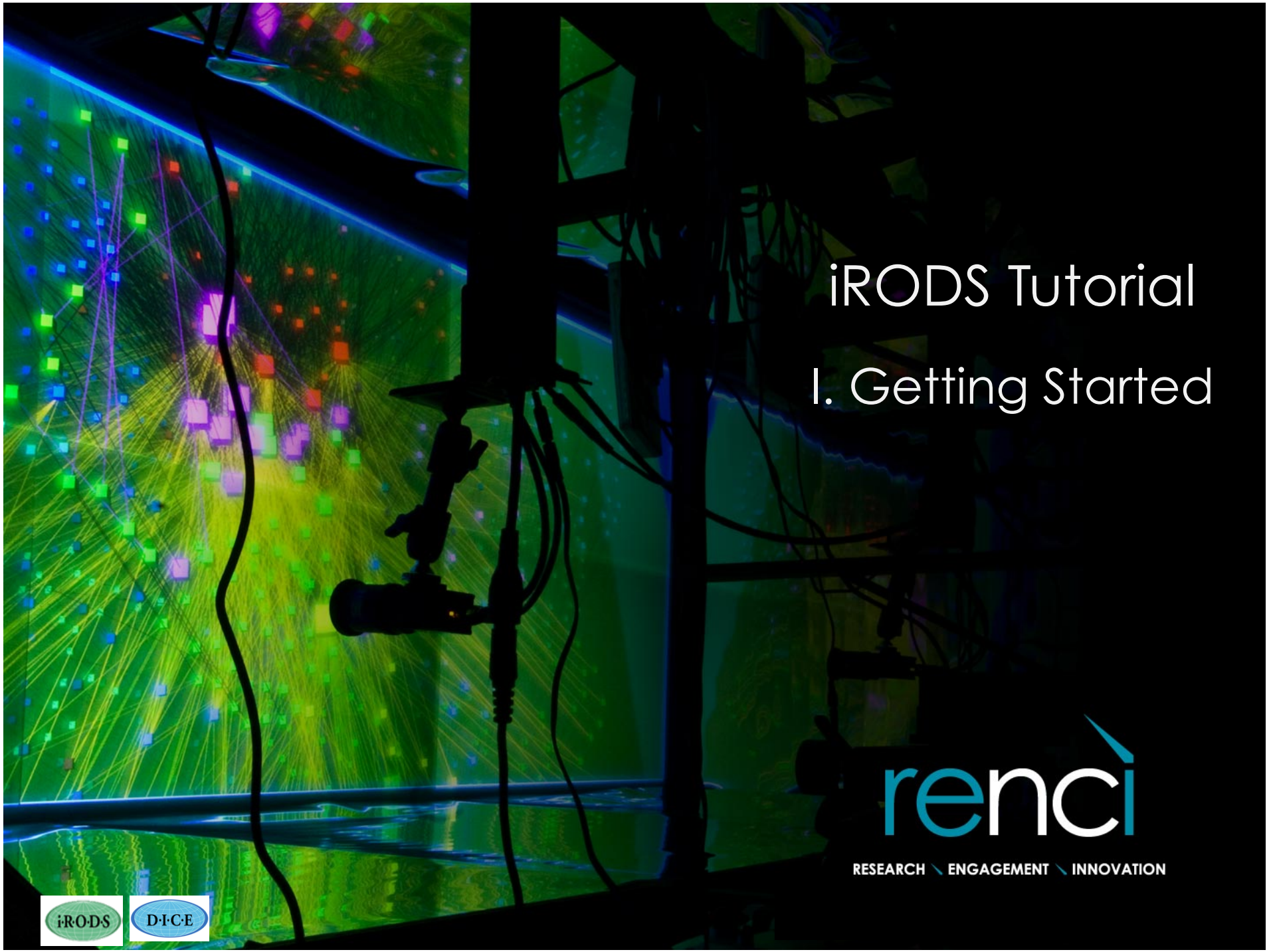

iRODS Tutorial
I. Getting Started
renci
RESEARCH ENGAGEMENT INNOVATION
i·R·O·D·S
D·I·C·E

# iRODS Tutorial Preview

I. iRODS Getting Started
   - unix client
   - usage

II. iRODS Data (Grid) Administration
   - installing server and iCAT
   - setting up users
   - adding new resources to a data grid/zone
   - federating with other grids/zones, remote users
   - microservices and rules for policy implementation and enforcement

# I. iRODS Getting Started

# iRODS Info

- Main page: http://www.irods.org
- Chat list: irods-chat@irods.org
- iRODS Documentation:
  https://www.irods.org/index.php/Documentation
- On-line tutorial:
  https://www.irods.org/index.php/Tutorial

# iRODS Books

Available from Amazon

- iRODS Primer: integrated Rule-Oriented Data System (Synthesis Lectures on Information Concepts, Retrieval, and Services)
  http://www.amazon.com/dp/1608453332

- The integrated Rule-Oriented Data System (iRODS) Micro-service Workbook
  http://www.amazon.com/dp/1466469129

# iRODS Download

- Download link from the iRODS main page:
  https://www.irods.org/download.html
- BSD license
- registration/agreement

# iRODS Download

Untar irods3.1.tgz

- cd into a directory where you want to install iRODS, eg $HOME/tutorial
- Move the tarball there: mv ~/irods3.1.tgz .
- Untar the tarball: tar –zxvf irods3.1.tgz
- cd into iRODS/

# iRODS Installation – unix client only

- Run the install script: ./irodssetup

- Can install three main components using irodssetup:
  1. an iRODS **server** (iCAT-enabled or not)
  2. the iCAT catalog metadata **database**
  3. 'icommands' – the **unix client**

- Install only the icommands for now…

# iRODS Installation - icommands

- ./irodssetup
- "No" to all prompts except last two:
  - Save configuration (irods.config) [yes]? yes
  - Start iRODS build [yes]? yes
- Set PATH to include the path to the icommands

  tcsh:
  setenv PATH $PATH:$HOME/tutorial/iRODS/clients/icommands/bin

  bash:
  export PATH=$PATH:$HOME/tutorial/iRODS/clients/icommands/bin

# Working with a Demo Data Grid

- If you have an account on an iRODS data grid, find your account name and password.
- Get your iRODS environment info from the .irodsEnv file *that goes with this data grid*.
- Make directory .irods/ in your home directory: mkdir ~/.irods
- Copy the .irodsEnv file into ~/.irods; edit if necessary to insert your user name.
- This will direct your client to the intended data grid, as the intended user.

# Sample .irodsEnv file

RENCI Demo Data Grid: compZone

```
# iRODS server host name:
  irodsHost 'ischia2.renci.org'

# iRODS server port number:
  irodsPort 1250

# Default storage resource name:
  irodsDefResource 'comp523Resc'

# Home directory in iRODS:
  irodsHome '/compZone/home/leesa'

# Current directory in iRODS:
  irodsCwd '/compZone/home/leesa'

# Account name:
  irodsUserName 'leesa'

# Zone:
  irodsZone 'compZone'

# Xmsg port:
  xmsgPort 1237
```

*The .irodsEnv file determines which data grid (zone) the icommands client connects to.*

In this example, user name is "leesa"

If you'll be using the Xmsg service

# Some iRODS Clients

- iDrop web – iDrop, iDrop-lite
  http://iren-web.renci.org:8080/idrop-web/login/login
- PHP web browser
  http://iren-web.renci.org/rodsweb
- icommands – unix client
  https://www.irods.org/index.php/icommands
- FUSE (Filesystem in Userspace) client
  https://www.irods.org/index.php/iRODS_FUSE
- Many others supplied by user communities

# Unix client: icommands

See
https://www.irods.org/index.php/icommands

Unix-like

ils
ipwd
icd
ichmod
irm
imkdir
ipasswd
irsync
ichksum
imv
icp
ienv

FTP-like

iinit
iexit
iput
iget

*(Not an exhaustive list.)*

# icommands (continued)

## Metadata

- imeta
- iquest
- idbo

## Functional

- ireg
- ibun
- irepl

## Informational

- ienv
- ilsresc
- iuserinfo
- ihelp

## Rule-oriented

- irule
- iqstat
- iqdel
- iqmod
- idbug

# icommands

```
> iinit
Enter your current iRODS password:

> ipwd
/compZone/home/leesa

> ils
/compZone/home/leesa:
  fuse-notes
  test_write.txt
  C- /compZone/home/leesa/slides

> ils -L
/compZone/home/leesa:
  leesa          0 comp523Resc              799 2012-01-08.13:59 & fuse-notes
    447a6462e578cb69ee8b0d82ade1f397    /vault2/comp523Vault/home/leesa/
     fuse-notes
  leesa          0 comp523Resc               13 2012-01-08.13:59 & test_write.txt
    59ca0efa9f5633cb0371bbc0355478d8    /vault2/comp523Vault/home/leesa/
     test_write.txt
  C- /compZone/home/leesa/slides
```

Directory naming convention:
/zone/home/user_name/collection_name

# icommands - ACLs

```
> ils -A
/compZone/home/leesa:
        ACL - leesa#compZone:own
        Inheritance - Disabled
  fuse-notes
        ACL - leesa#compZone:own
  test_write.txt
        ACL - leesa#compZone:own
  C- /compZone/home/leesa/slides

> ichmod read baretto fuse-notes
```

```
> ils -A
/compZone/home/leesa:
        ACL - leesa#compZone:own
        Inheritance - Disabled
  fuse-notes
        ACL - leesa#compZone:own
      baretto#compZone:read object
  test_write.txt
        ACL - leesa#compZone:own
  C- /compZone/home/leesa/slides
```

# ienv

```
> ienv
    NOTICE: Release Version = rods3.1beta, API Version = d
    NOTICE: irodsHost=ischia2.renci.org
    NOTICE: irodsPort=1250
    NOTICE: irodsDefResource=comp523Resc
    NOTICE: irodsHome=/compZone/home/rods
    NOTICE: irodsCwd=/compZone/home/rods
    NOTICE: irodsUserName=rods
    NOTICE: irodsZone=compZone
    NOTICE: xmsgHost=ischia2.renci.org
    NOTICE: xmsgPort=1237
```

# Group "public"

Every user in the data grid is a member of user group "public"

> ichmod -r read public slides

> ils -A slides

/compZone/home/leesa/slides:

ACL - public#compZone:read object

baretto#compZone:read object leesa#compZone:own
rods#compZone:read object mikec#compZone:read object
comp523#compZone:read object guerline#compZone:read object
holston#compZone:read object Username#compZone:read object
leesa#compZone:read object

Inheritance - Disabled

1-overview.ppt

ACL - public#compZone:read object leesa#compZone:own

slide-list.html

ACL - public#compZone:read object leesa#compZone:own

# icommands – putting & getting data

> iput -K derby.log (calculate and store checksums)
> iput notes (no checksums)
> ils -L

```
/compZone/home/leesa:
  leesa          0 comp523Resc              419 2012-01-10.11:59 & derby.log
    11adc3cf922e31db8dfd4a2806581f99
            /vault2/comp523Vault/home/leesa/derby.log
  leesa          0 comp523Resc              799 2012-01-08.13:59 & fuse-notes
    447a6462e578cb69ee8b0d82ade1f397
            /vault2/comp523Vault/home/leesa/fuse-notes
  leesa          0 comp523Resc             3645 2012-01-10.12:00 & notes
        /vault2/comp523Vault/home/leesa/notes
  leesa          0 comp523Resc               13 2012-01-08.13:59 & test_write.txt
    59ca0efa9f5633cb0371bbc0355478d8
        /vault2/comp523Vault/home/leesa/test_write.txt
  C- /compZone/home/leesa/slides
```

> iget –k notes (verify checksum without storing)

-k and –K options for checksum calculation

# icommands – replicating data objects

- > ils

  /compZone/home/leesa/rods:
  hello
  C- /compZone/home/leesa/rods/rules

Replication is not the same as copying: a replica is the same logical object as the original; a copy is a new logical object.

- > irepl -R demoResc hello

- > ils

  /compZone/home/leesa/rods:
  hello
  C- /compZone/home/leesa/rods/rules

Replicated object ("hello") appears as a single logical object

- > ils -L

  /compZone/home/leesa/rods:
  rods 0 comp523Resc 11 2011-09-19.15:42 & hello
  /vault2/comp523Vault/home/leesa/rods/hello
  rods 1 demoResc 11 2012-02-02.11:51 & hello
  /vault2/demoVault/home/leesa/rods/hello
  C- /compZone/home/leesa/rods/rules

Do the long listing (ils –L) to see all replicas of an object ("hello") and physical locations

# ireg – register data into iRODS

*Get data into iRODS without making an additional copy or moving it*

Example: directory /vault2/state-data contains state LiDAR data that we now want in an iRODS repository… without copying it

1. /vault2/state-data is mounted on the iRODS server host

2. Data admin sets up existing directory as an iRODS resource

3. User registers existing data into iRODS iCAT:

   ireg -C /vault2/state-data /compZone/home/leesa/state-data

   (-f option for picking up unregistered files)

*Register incoming files rigorously OR modify a directory **only** through iRODS once it has been registered to keep the iCAT consistent with the directory.*

# ibun – for bundling files

- Tar up and expand files for efficient iput/iget
- iput a tarball and expand it within iRODS:
  - tar -chlf tutorials.tar -C tutorials .
  - iput -Dtar tutorials.tar .
  - ibun -x tutorials.tar tutorials
- Tar up files in iRODS for iget:
  - ibun -cDtar slides.tar slides
  - iget slides.tar
  - tar –xvf slides.tar –C slides

# ilsresc

See resources available on your data grid

- compZone:
  - > ilsresc
    - msoResc2
    - demoResc
    - msoResc1
    - bundleResc
    - comp523Resc
    - stateResc
    - cpsresc
    - msoRescGroup (resource group)

# iquest – querying the iCAT

- Pre-defined queries:

> iquest "SELECT DATA_NAME, DATA_CHECKSUM WHERE
DATA_RESC_NAME like 'demo%'"

DATA_NAME = cps.test1.txt
DATA_CHECKSUM =
---
DATA_NAME = hello
DATA_CHECKSUM =
---
DATA_NAME = homewood_info.doc
DATA_CHECKSUM = 67614aedf5b41cae0487eb5fe9b0d3ae
---

(Checksums displayed only for those that have already been calculated and stored)

- > iquest attrs to see attributes that can be queried
- See https://www.irods.org/index.php/iquest for examples

# iquest – querying the iCAT

- Useful when you want to remove a resource and you discover it isn't empty:

  > iquest "SELECT DATA_NAME, USER_NAME, COLL_NAME WHERE
  DATA_RESC_NAME like 'msoResc1'"

  DATA_NAME = slide-list-html
  USER_NAME = rods
  COLL_NAME = /compZone/trash/home/rods/DataNet
  ---

- Admin can add SQL strings to be invoked by users of the data grid

- Data grid-specific queries ( added by admin)

  > iquest --sql 'pre-defined SQL string' [format] [arguments]

# imeta – add, view, modify metadata

- imeta add –d hello “Date” “2 february 2012”
- imeta ls –d hello

```
AVUs defined for dataObj hello:
attribute: Meta1
value: hello
units:
----
attribute: Date
value: 2 february 2012
units:
```

- imeta rm –d hello “Meta1” “hello”

# Realizable Objects

- Typical iRODS objects contain data
- Realizable objects:
  - symbolic links to iRODS objects
  - symbolic links to external data sources
  - workflow procedures to regenerate data
- Symbolic links implemented so far
  - instantiated through a compound resource
    - mso resource (mso: microservice object)
    - cache resource
  - symbolic links implemented for http and Z39.50

# Symbolic Links to an http Source

1. Admin user must set up the mso resource and resource group, for example:
   - mso resource: httpResc
   - mso group: httpGroup

2. User registers external data

```
> ireg  -D  mso  -R  httpResc  -G  httpGroup
        "//http://www.renci.org/~leesa/irodsEnv-files/irodsEnv-compZone"
                /compZone/home/leesa/tutorial/env-files/irodsEnv-compZone
```

# Symbolic Links to an http Source

- User puts symbolic link in his collection (registers external data)
- Data is then accessible to anyone with read authorization to the user's collection
- iget causes a replica to be made in the disk cache of the mso resource group (compound resource)
  - do an iget of a file in this directory and see

# Cloud Resources

1. Admin user must set up the cloud resource and resource group, for example:
   - S3 Resource: s3Resc
   - S3 Group: s3Group

2. This is just another iRODS resource for the users, who can manage their cloud data just as all other data:

   > iput –K –R s3Resc my_file  (put data into the cloud resource)

   > irepl –R s3Resc another_file (replicate into the cloud resource)

   > ichmod read public my_file  (give public access to cloud data)

# Database Resources

- Database Resource (DBR): a database, queried and updated via SQL (or other, for non-SQL)
- Database object (DBO): an interface to a DBR, typically a (SQL) query that returns results
- iRODS agent will open and close DB as needed; results of the query are directly returned to user
- Query results are stored to an iRODS data object, a DBO Results file (DBOR).
- iRODS access controls are applied on the DBR and DBO.

# Database Resources

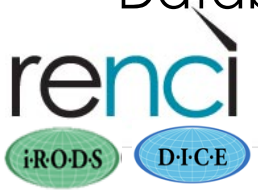
- https://www.irods.org/index.php/Database_Resources and https://www.irods.org/index.php/Database_Resource_Administration

- idbo command – to access the external DB resource

# idbo Command

- Accepts commands on the command line
- If no command is given, goes into interactive mode
- Commands:
  - open DBR (open a database resource)
  - close DBR (close a database resource)
  - exec DBR DBO [arguments] (execute a DBO on a DBR)
  - output [-f] DBOR (store 'exec' results in another data-object)
  - commit DBR (commit updates to a DBR (done via a DBO))
  - rollback DBR (rollback updates instead)
  - ls (list defined Database-Objects in the Zone)
  - help (or h) [command] (this help, or help on a command)
  - quit (or 'q', exit idbo)

Where DBR and DBO are the names of a Database Resource and Database Object.



# Access Controls for DBRs

- iRODS administrators can create DBOs, since they can give anyone (including themselves) 'write' access to the DBR.
- iRODS users with 'write' access to the DBR will also be allowed to create DBOs.
- iRODS users with 'read' access to the DBR will be allowed to execute DBOs that were created by users with 'write' access to the same DBR.

  The 'read' users, for some DBO SQL, will provide parameters to include in the SQL, which will be executed as SQL bind variables (to restrict capabilities).

  This access mode will allow more privileged users to create controlled access for additional users.

# Rules

- New rule engine with 3.0
- See https://www.irods.org/index.php/Changes_and_Improvements_to_the_Rule_Language_and_the_Rule_Engine
- Implement computer actionable policies
  - Retention, distribution, arrangement
  - Authenticity, provenance, description
  - Integrity, replication, synchronization
  - Deletion, trash cans, versioning
  - Archiving, staging, caching
  - Authentication, authorization, redaction
  - Access, approval, IRB, audit trails, report generation
  - Assessment criteria, validation
  - Derived data product generation, format parsing

# Microservices

- C code
- the unit of work within iRODS
- called by rules
- composed into workflows by rules

# Running Rules

- triggered by events/policy points
- contained in the (distributed) rule base:
  - iRODS_dir/server/config/reConfigs/core.re
  - first rule with satisfied condition is executed; others are skipped
- can be run with irule: *manual execution*
- delayed execution
  - iqstat
  - iqdel

# irule – to run a rule manually

- Example rules to tweak and run in the software distribution
  iRODS/clients/icommands/test/rules3.0

- irule -F listMS.r

- irule -F rulemsiAdmShowCoreRE.r -
  can only be run by admin users

# iDrop Client

- YouTube video overviews:

  http://www.youtube.com/user/diceresearch?feature=guide

  - iDrop – iDrop Suite Overview
  - iDrop-web: iDrop Suite Overview part 2

- Test iDrop server:

  iren-web.renci.org:8080/idrop-web

# iDrop Client

- iRODS Tree View
  - Right click on a file
  - Expand:  New folder:  Delete:  Rename
- Tag a file
  - iRODS info -> Click on a file
  - Add tag (no spaces in tag) -> Update Info
- Comment a file
  - iRODS info -> Click on a file
  - Add comment
- Search
  - By name (file)
  - By tag
  - By name and tag

# iDrop Client

- Supports
  - Drag and drop
  - Replication
  - Browsing
  - Searching
- Manages a queue of transfer requests
  - Checkmark -> Show Current and Past Activity
  - Transfer Summary -> select a transfer
  - Transfer Details
  - Purge: Delete: Resubmit: Restart: Refresh
  - View:  list subset of transfers
- Should be able to disconnect, and iDrop will continue transmissions when reconnected

# iDrop Client

- Metadata
- Replication
- Synchronizing a directory
  - Tools -> Preferences -> Synchronization
  - Pick local directory
  - Pick iRODS directory
  - Select synchronization period (1 day)
- Caveats
  - To synchronize, a list of files in the local directory is generated, which can take a long time for large directories
  - Keep the local directory small