



# iRODS Hands-On

Elena Erastova (RZG)

Based on material provided by RENCI



Rome, 28 October 2013

# Contents

## I. iRODS Getting Started

icommands

## II. iRODS Data (grid) Administration

Zone Federation

Microservices

Rules

## III. EUDAT Safe Replication

Replix-EUDAT Expamle

EPIC API

Practice

# Demo Data Grids

- 9 Virtual machines running

– *ssh -X user@IP*

VM name	IP address	hostname	user	password
<b>C:</b> eudat-0	145.100.57.50	eudat01	eudat01	25eudat01
<b>D1:</b> eudat-1	145.100.57.52	eudat-02	eudat02	25eudat02
<b>D2:</b> eudat-2	145.100.57.39	eudat01	eudat01	25eudat01
<b>C:</b> eudat-3	145.100.57.40	eudat-02	eudat02	25eudat02
<b>D1:</b> eudat-4	145.100.57.41	eudat01	eudat01	25eudat01
<b>D2:</b> eudat-5	145.100.57.42	eudat-02	eudat02	25eudat02
<b>C:</b> eudat-6	145.100.57.53	eudat01	eudat01	25eudat01
<b>D1:</b> eudat-7	145.100.57.43	eudat-02	eudat02	25eudat02
<b>D2:</b> eudat-9	145.100.57.54	eudat-02	eudat02	25eudat02

# Demo Data Grids

- iRODS installed in: “~/dice/iRODS”, iEnv files: “~/.irods/.irodsEnv”
- iRODS user accounts:

VM name	iRODS zone	iRODS admin	password
<b>C:</b> eudat-0	vzEudat0	rods0	25rods0
<b>D1:</b> eudat-1	vzEudat1	rods1	25rods1
<b>D2:</b> eudat-2	vzEudat2	rods2	25rods2
<b>C:</b> eudat-3	vzEudat3	rods3	25rods3
<b>D1:</b> eudat-4	vzEudat4	rods4	25rods4
<b>D2:</b> eudat-5	vzEudat5	rods5	25rods5
<b>C:</b> eudat-6	vzEudat6	rods6	25rods6
<b>D1:</b> eudat-7	vzEudat7	rods7	25rods7
<b>D2:</b> eudat-9	vzEudat9	rods9	25rods9



# I. iRODS Getting Started

# iRODS Control

- `./irodsctl`
  - `start` } Start/stop/restart the iRODS server and the iCAT
  - `stop`
  - `Restart`
  - `istart` } Start/stop/restart the iRODS server but not the iCAT
  - `istop`
  - `irestart`

# .irodsEnv File

```
# iRODS personal configuration file.
#
# This file was automatically created during iRODS installation.
# Created Mon Jun 18 13:13:27 2012
#
# iRODS server host name:
irodsHost 'eudat01'
# iRODS server port number:
irodsPort 1247
# Default storage resource name:
irodsDefResource 'demoResc'
# Home directory in iRODS:
irodsHome '/vzEudat0/home/rods0'
# Current directory in iRODS:
irodsCwd '/vzEudat0/home/rods0'
# Account name:
irodsUserName 'rods0'
# Zone:
irodsZone 'vzEudat0'
```

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



## iRODS i-commands

Documentation:

<https://www.irods.org/index.php/icommands>

(we don't provide a exhaustive list of i-commands)



# icommands

- icommands of the following categories will be introduced (not an exhaustive list):
  - Informational
  - Unix and FTP like
  - Functional
  - Metadata
  - Rules
- Examples of several icommands will be shown

# icommands

- Make sure the icommands can be found on the PATH
- You can find the icommands in the  
`~/dice/iRODS/clients/icommands/bin` folder

- Add the icommands to the path:

tssh:

```
setenv PATH $PATH:$HOME/dice/iRODS/clients/icommands/bin
```

bash:

```
export PATH=$PATH:$HOME/dice/iRODS/clients/icommands/bin
```

# Informational

ienv

ihelp

ilsresc

iuserinfo

imiscsvrinfo

# ihelp

Display help on any of the icommands if you need more information about the command.

Usage : `ihelp [-h] [icommand]`

```
eudat01@eudat01:$ ihelp -h
```

```
Usage : ihelp [-ah] [icommand]
```

```
Display i-commands synopsis or a particular i-command help text
```

```
Options are:
```

```
-h this help
```

```
-a print the help text for all the i-commands
```

```
Run with no options to display a synopsis of the i-commands
```

```
iRODS Version 3.2
```

```
ihelp
```

```
October 2012
```

# ienv

Lists the iRODS environment variables.

Equivalent to “iinit -l”

Usage : ienv [-h]

```
eudat01@eudat01:~$ ienv
```

```
NOTICE: Release Version = rods3.2, API Version = d
NOTICE: irodsHost=eudat01
NOTICE: irodsPort=1247
NOTICE: irodsDefResource=demoResc
NOTICE: irodsHome=/vzEudat0/home/rods0
NOTICE: irodsCwd=/vzEudat0/home/rods0
NOTICE: irodsUserName=rods0
NOTICE: irodsZone=vzEudat0
```

# iuserinfo

Show information about your iRODS user account or the entered user

Usage: `iuserinfo [-vVh] [user]`

```
eudat01@eudat01:~$ iuserinfo
```

```
name: rods0
```

```
id: 10007
```

```
type: rodsadmin
```

```
zone: vzEudat0
```

```
info:
```

```
comment:
```

```
create time: 01340018007: 2012-06-18.13:13:27
```

```
modify time: 01340018007: 2012-06-18.13:13:27
```

```
member of group: rods0
```

```
member of group: public
```

# imiscsvrinfo

Connect to the server and retrieve some basic server information. Can be used as a simple test for connecting to the server.

Usage: `imiscsvrinfo [-hvV]`

```
eudat01@eudat01:~$ imiscsvrinfo
```

```
RCAT_ENABLED  
relVersion=rods3.2  
apiVersion=d  
rodsZone=vzEudat0  
up 0 days, 23:40
```

# ilsresc

ilsresc lists iRODS resources and resource-groups

Usage: ilsresc [-lvVhA] [Name]

```
eudat01@eudat01:~$ ilsresc
```

```
demoResc
```

“demoResc” is created during a default installation



# ilsresc

Find out the details of a specific resource:

```
eudat01@eudat01:~$ ilsresc -l demoResc
```

```
resource name: demoResc
resc id: 10010
zone: vzEudat0
type: unix file system
class: archive
location: eudat01
vault: /home/eudat01/dice/iRODS/Vault
free space:
status:
info:
comment:
create time: 01340018011: 2012-06-18.13:13:31
modify time: 01340018011: 2012-06-18.13:13:31
```

# FTP and Unix-like

## Unix-like

ils

ipwd

icd

ichmod

irm

imkdir

ipasswd

irsync

ichksum

imv

icp

## FTP-like

iinit

iput

iget

iexit

# icommands - Navigating

- Print working directory:

```
eudat01@eudat01:~$ ipwd
```

```
/vzEudat0/home/rods0
```

Directory naming convention:  
/zone/home/user\_name

- List directory:

```
eudat01@eudat01:~$ ils
```

```
/vzEudat0/home/rods0:
```

- Change directory:

```
eudat01@eudat01:~$ icd /vzEudat0/home/
```

# icommands - ACLs

- List ACLs in a collection:

```
eudat01@eudat01:~$ ils -A  
/vzEudat0/home/rods0:  
    ACL - rods0#vzEudat0:own  
    Inheritance - Disabled
```

# icommands – Put and Get Data

- Put a file to iRODS (-v displays upload stats):

```
eudat01@eudat01:~$ iput -v put1.txt
```

```
put1.txt    0.000MB | 0.023sec | 0 thr | 0.000 MB/s
```

```
eudat01@eudat01:~$ ils -L
```

```
/vzEudat0/home/rods0:
```

```
rods0          0 demoResc      7 2013-10-08.17:31 & put1.txt
```

```
generic      /home/eudat01/dice/iRODS/Vault/home/rods0/put1.txt
```

```
eudat01@eudat01:~$ ichksum put1.txt
```

```
put1.txt          c16d7d3488677a3348b12eb82795d28c
```

```
Total checksum performed = 1, Failed checksum = 0
```

# iadmin – Administrative Functions

- Type “iadmin” to enter the admin console
  - h for help
  - quit to exit
- Add new users, modify passwords, add new resources, federate to remote zones, create resource groups, ...
  - mkresc/rmresc
  - mkuser/rmuser, moduser (modify passwords)
  - mkzone/rmzone, modzone (for federation)
- Information on users, resources, tokens, etc
  - lt (el-tee)
  - lu, lr, lz, ...

# Setting Up New Users

- Use iadmin
- Two steps: mkuser and moduser (for a password)

```
iadmin> mkuser user1 rodsuser
```

```
iadmin> moduser user1 password *****
```

- Use iadmin to see what user types are possible

```
iadmin> lt
```

```
zone_type  
user_type  
data_type  
resc_type  
action_type  
rulexec_type  
access_type  
object_type  
resc_class  
coll_map  
auth_scheme_type
```

```
iadmin> lt user_type
```

```
rodsgroup  
rodsadmin  
rodsuser  
domainadmin  
groupadmin  
storageadmin  
rodscurators
```

# Creating New Resources

- Use iadmin mkresc

```
eudat01@eudat01:~/dice/iRODS$ mkdir Data
```

```
iadmin mkresc dataResc "unix file system" cache eudat01 "/home/eudat01/dice/iRODS/Data"
```

- Use iadmin to list resources

```
eudat01@eudat01:~/dice/iRODS$ iadmin lr dataResc
```

- Put a file to this resource

```
eudat01@eudat01:~/dice/iRODS$ iput -R dataResc put2.txt
```

```
eudat01@eudat01:~/dice/iRODS$ ils -l
```

```
rods0          0 demoResc          7 2013-10-08.18:05 & put1.txt
```

```
rods0          0 dataResc          5 2013-10-08.18:04 & put2.txt
```

- Check the physical location of files

```
eudat01@eudat01:~$ ls ~/dice/iRODS/Vault/home/rods0
```

```
put1.txt
```

```
eudat01@eudat01:~$ ls ~/dice/iRODS/Data/home/rods0
```

```
put2.txt
```



# icommands - ACLs

- Change permissions:

```
eudat01@eudat01:~$ ichmod read user1 put1.txt
```

- List ACLs in the collection again:

```
eudat01@eudat01:~$ ils -A
```

```
/vzEudat0/home/rods0:
```

```
ACL - rods0#vzEudat0:own
```

```
Inheritance - Enabled
```

```
put1.txt
```

```
ACL - user1#vzEudat0:read object rods0#vzEudat0:own
```

```
put2.txt
```

```
ACL - rods0#vzEudat0:own
```

# Functional

ireg

ibun

irepl

# icommands – irepl

- Replicate a file to another resource:

```
eudat01@eudat01:~$ ils -l
```

```
rods0          0 demoResc          7 2013-10-08.18:05 & put1.txt
rods0          0 dataResc          5 2013-10-08.18:04 & put2.txt
```

```
eudat01@eudat01:~$ irepl -R dataResc put1.txt
```

```
eudat01@eudat01:~$ ils -l
```

```
/vzEudat0/home/rods0:
```

```
rods0          0 demoResc          7 2013-10-08.18:05 & put1.txt
rods0          1 dataResc          7 2013-10-08.18:13 & put1.txt
rods0          0 dataResc          5 2013-10-08.18:04 & put2.txt
```

```
eudat01@eudat01:~$ ls ~/dice/iRODS/Data/home/rods0
put1.txt put2.txt
```

# icommands – itrim

- Trim down the number of replicas of a file in iRODS by deleting some replicas.

```
eudat01@eudat01:~$ ils -l
```

```
/vzEudat0/home/rods0:
```

```
rods0          0 demoResc          7 2013-10-08.18:22 & put1.txt
rods0          1 dataResc          7 2013-10-08.19:03 & put1.txt
rods0          0 dataResc          7 2013-10-08.19:03 & put2.txt
```

```
eudat01@eudat01:~$ itrim -S demoResc -N 1 put1.txt
```

```
eudat01@eudat01:~$ ils -l
```

```
rods0          1 dataResc          7 2013-10-08.19:03 & put1.txt
rods0          0 dataResc          7 2013-10-08.19:03 & put2.txt
```

```
eudat01@eudat01:~$ itrim -S demoResc -N 1 put1.txt
```

```
eudat01@eudat01:~$ ils -l
```

```
rods0          1 dataResc          7 2013-10-08.19:03 & put1.txt
rods0          0 dataResc          7 2013-10-08.19:03 & put2.txt
```

# iRODS Metadata

imeta

iquest

idbo

# iquest – Querying the iCAT

```
eudat01@eudat01:~$ iquest "SELECT DATA_NAME,DATA_CHECKSUM WHERE  
DATA_RESC_NAME like 'data%'"
```

```
DATA_NAME = put1.txt
```

```
DATA_CHECKSUM = c16d7d3488677a3348b12eb82795d28c
```

```
-----  
DATA_NAME = put2.txt
```

```
DATA_CHECKSUM = f632f5eafffa5607c961e22ba40291ab7  
-----
```

`iquest attrs` to see attributes that can be queried

- See <https://www.irods.org/index.php/iquest> for examples

# imeta – Add, View, Modify User Metadata

- Add metadata to an object:

```
eudat01@eudat01:~$ imeta add -d put1.txt "Date" "08.10.2013"
```

- List metadata for an object:

```
eudat01@eudat01:~$ imeta ls -d put1.txt
```

```
AVUs defined for dataObj put1.txt:
```

```
attribute: Date
```

```
value: 08.10.2013
```

```
units
```

- Remove metadata:

```
eudat01@eudat01:~$ imeta rm -d put1.txt "Date" "08.10.2013"
```

```
eudat01@eudat01:~$ imeta ls -d put1.txt
```

```
AVUs defined for dataObj put1.txt:
```

```
None
```



## II. iRODS Data (Grid) Administration



# Federation Between Data Grids

- [https://www.irods.org/index.php/Federation\\_Administration](https://www.irods.org/index.php/Federation_Administration)
- Manage data on remote servers
- List, put, get files as if they are on a local server

```
eudat01@eudat01:~$ ils /
```

```
/:
```

```
C- /
```

```
C- /vzEudat0
```

```
C- /vzEudat1
```

```
C- /vzEudat2
```

```
eudat01@eudat01:~$ iadmin lz vzEudat0
```

```
zone_id: 9000
```

```
zone_name: vzEudat0
```

```
zone_type_name: local
```

```
zone_conn_string:
```

```
r_comment:
```

```
create_ts: 1170000000 : 2007-01-28.17:00:00
```

```
modify_ts: 1170000000 : 2007-01-28.17:00:00
```

```
eudat01@eudat01:~$ iadmin lz vzEudat1
```

```
zone_id: 10059
```

```
zone_name: vzEudat1
```

```
zone_type_name: remote
```

```
zone_conn_string: 145.100.57.52:1247
```

```
r_comment:
```

```
create_ts: 01381160376 : 2013-10-07.17:39:36
```

```
modify_ts: 01381160376 : 2013-10-07.17:39:36
```

```
eudat01@eudat01:~$ ils /vzEudat1/home/
```

```
/vzEudat1/home:
```

```
C- /vzEudat1/home/public
```

```
C- /vzEudat1/home/rods0#vzEudat0
```

```
C- /vzEudat1/home/rods1
```

```
C- /vzEudat1/home/rods2#vzEudat2
```

```
C- /vzEudat1/home/userods1
```

# Federation Between Data Grids

- [https://www.irods.org/index.php/Federation\\_Administration](https://www.irods.org/index.php/Federation_Administration)
- Zone A acknowledges Zone B: `iadmin mkzone B remote Host:Port`
- Zone B acknowledges Zone A: `iadmin mkzone A remote Host:Port`
- Zone A adds remote users: `iadmin mkuser some_user#B`
- Zone B adds remote users: `iadmin mkuser other_user#A`
- Set up ACL for the remote user
- User can see resources in remote zone A: `ilsresc -z A`

Admin users from one grid won't necessarily be admin users on the other grid

# Data Replication

- `irsync`
- Synchronize the data between a local copy (local file system) and the copy stored in iRODS or between two iRODS copies.
- The command can be in one of the three modes:
  - synchronization of data from the client's local file system to iRODS
  - from iRODS to the local file system
  - From one iRODS path to another iRODS path.

# Data Replication

```
eudat01@eudat01:~$ irsync -R demoResc i:/vzEudat0/home/rods0/put1.txt  
i:/vzEudat1/home/rods0#vzEudat0/put1.txt
```

```
eudat01@eudat01:~$ ils /vzEudat1/home/rods0#vzEudat0/
```

```
/vzEudat1/home/rods0#vzEudat0:
```

```
put1.txt
```

```
eudat01@eudat01:~$ irsync -R demoResc /home/eudat01/put2.txt  
i:/vzEudat1/home/rods0#vzEudat0/put2.txt
```

```
eudat01@eudat01:~$ ils /vzEudat1/home/rods0#vzEudat0/
```

```
/vzEudat1/home/rods0#vzEudat0:
```

```
put1.txt
```

```
put2.txt
```

```
eudat01@eudat01:~$ irsync -R demoResc  
i:/vzEudat1/home/rods0#vzEudat0/put2.txt /home/eudat01/put2.txt
```

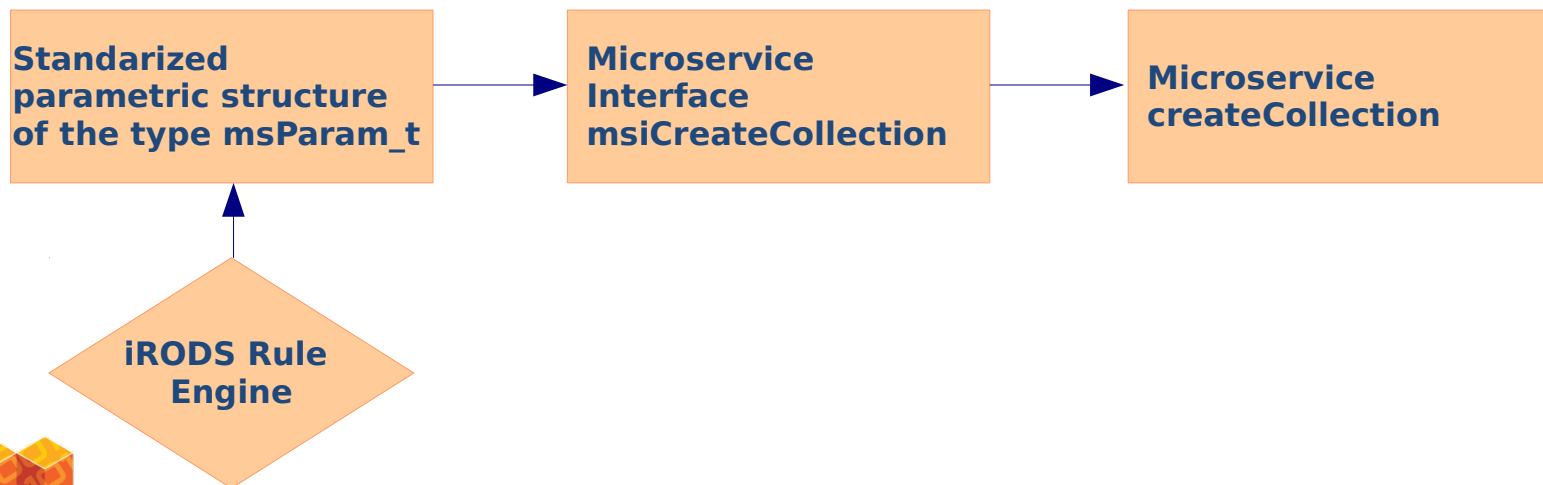
# iRODS Microservices

Micro-services are small, well-defined procedures/functions that perform a simple task

[https://www.irods.org/index.php/iRODS\\_Micro-services](https://www.irods.org/index.php/iRODS_Micro-services)

Examples: `msiCollCreate`, `msiDataObjRepl`, `msiDataObjChksum`, ...

To convert a C function `createCollection` into a microservice create an interface `msiCreateCollection`



# iRODS Rules

- A rule file replicate.r:

```
replicate{
```

```
    msiWriteRodsLog("starting replication", *status);
```

```
    msiWriteRodsLog("source = *source", *status);
```

```
    msiWriteRodsLog("collectionPath = *collectionPath", *status);
```

```
    msiDataObjRsync(*source, "IRODS_TO_IRODS", "null", *destination,  
*rsyncStatus);
```

```
}
```

INPUT

```
*source="/vzEudat0/home/rods0/put3.txt",*destination="/vzEudat1/home/rods0#v  
zEudat0/put3.txt"
```

OUTPUT ruleExecOut

# iRODS Rules

- **Execute:**

```
eudat01@eudat01:~$ irule -vF replicate.r
```

```
rcExecMyRule: replicate{  
    msiWriteRodsLog("starting replication", *status);  
    msiWriteRodsLog("source = *source", *status);  
    msiDataObjRsync(*source, "IRODS_TO_IRODS", "null", *destination, *rsyncStatus);  
}
```

```
outParamDesc: ruleExecOut
```

```
ExecMyRule completed successfully.  Output
```

```
eudat01@eudat01:~$ ils /vzEudat1/home/rods0#vzEudat0/
```

```
/vzEudat1/home/rods0#vzEudat0:
```

```
put1.txt
```

```
put2.txt
```

```
put3.txt
```

# Delayed Execution

- Example

```
myTestRule{  
    delay("<PLUSET>1m</PLUSET>"){  
        writeLine("stdout","Writing message *A with a delay.");  
    }  
}
```

INPUT \*A="1"

OUTPUT ruleExecOut

- Queue management:
  - iqstat
  - iqdel
  - iqmod



# Periodic Execution

## Example

```
myTestRule {
# Input parameters are:
#   Source collection path
#   Target collection path
#   Optional target resource
#   Optional synchronization mode:  IRODS_TO_IRODS
# Output parameter is:
#   Status of the operation
# Output from running the example is:
#   Synchronized collection 1 with collection 2
#
    delay("<PLUSET>5m</PLUSET>EF>1h</EF>"){
        msiCollRsync(*srcColl,*destColl,*Resource,"IRODS_TO_IRODS",*Status);
        writeLine("stdout","Synchronized collection *srcColl with collection
*destColl");
    }
}
INPUT *srcColl="/eu00Zone/home/rods/data",
*destColl="/eu00Zone/home/rods/data2", *Resource="demoResc"
OUTPUT ruleExecOut
```

# iRODS Log Files

- `~/dice/iRODS/server/log/rodsLog.<date>:`

...

Oct 8 19:45:42 pid:31026 NOTICE: Agent process 5775 started for puser=rods0 and cuser=rods0 from 127.0.0.1

Oct 8 19:45:42 pid:5775 NOTICE: rsAuthCheck user rods0#vzEudat0

Oct 8 19:45:42 pid:5775 NOTICE: rsAuthResponse set proxy authFlag to 5, client authFlag to 5, user:rods0#vzEudat0 proxy:rods0 client:rods0

Oct 8 19:45:42 pid:5775 NOTICE: msiWriteRodsLog message: starting replication

Oct 8 19:45:42 pid:5775 NOTICE: msiWriteRodsLog message: source = /vzEudat0/home/rods0/put3.txt

Oct 8 19:45:42 pid:31026 NOTICE: Agent process 5777 started for puser=rods1 and cuser=rods0 from 145.100.57.52

Oct 8 19:45:42 pid:5777 NOTICE: rsAuthCheck user rods0#vzEudat0

Oct 8 19:45:42 pid:5777 NOTICE: readAndProcClientMsg: received disconnect msg from client

Oct 8 19:45:42 pid:5777 NOTICE: Agent exiting with status = 0

Oct 8 19:45:42 pid:5775 NOTICE: readAndProcClientMsg: received disconnect msg from client

Oct 8 19:45:42 pid:5775 NOTICE: Agent exiting with status = 0

...

# iRODS Rule Base

- `~/dice/iRODS/server/config/reConfigs/core.re`

```
# iRODS Rule Base
```

```
# The new rule language is used to express all policies
```

```
# Recovery procedures are included for a micro-service after " ::: "
```

```
#
```

```
#Test Rules
```

```
printHello { print_hello; }
```

```
#
```

```
# These are sys admin rules for creating and deleting users and renaming
```

```
# the local zone.
```

```
acCreateUser {
```

```
    acPreProcForCreateUser;
```

```
    acCreateUserF1;
```

```
    acPostProcForCreateUser; }
```

```
acCreateUserF1 {
```

```
    ON($otherUserName == "anonymous") {
```

```
        msiCreateUser ::: msiRollback;
```

```
        MsiCommit; } }
```

```
...
```

# iRODS Rule Base

- Append to `~/dice/iRODS/server/config/reConfigs/core.re`:

```
acPostProcForPut {  
  ON($objPath like "/vzEudat0/home/rods0/data/*") {  
    msiWriteRodsLog("put a file $objPath to a collection /vzEudat0/home/rods0/data/", *status);  
    delay("<PLUSET>1m</PLUSET><EF>'REPEAT 3 TIMES'</EF>") {  
      msiWriteRodsLog("put a file $objPath to a collection /vzEudat0/home/rods0/data/",  
*status);  
    }  
  }  
}
```

- `iput put1.txt /vzEudat0/home/rods0/data`

# iRODS Rule Base

## Check log files:

### cat dice/iRODS/server/log/rodsLog.2013.10.06

```
Oct 9 14:25:26 pid:11809 NOTICE: Agent process 12011 started for puser=rods0 and cuser=rods0 from 127.0.0.1
Oct 9 14:25:26 pid:12011 NOTICE: rsAuthCheck user rods0#vzEudat0
Oct 9 14:25:26 pid:12011 NOTICE: rsAuthResponse set proxy authFlag to 5, client authFlag to 5, user:rods0#vzEudat0 proxy:rods0 client:rods0
Oct 9 14:25:26 pid:12011 NOTICE: msiWriteRodsLog message: put a file /vzEudat0/home/rods0/data/put2.txt to a collection /vzEudat0/home/rods0/data/
Oct 9 14:25:26 pid:12011 NOTICE: readAndProcClientMsg: received disconnect msg from client
Oct 9 14:25:26 pid:12011 NOTICE: Agent exiting with status = 0
```

### cat dice/iRODS/server/log/reLog.2013.10.06

```
Oct 9 14:29:14 pid:11811 NOTICE: reServerMain: checking the queue for jobs
Oct 9 14:29:14 pid:12033 NOTICE: msiWriteRodsLog message: put a file /vzEudat0/home/rods0/data/put2.txt to a collection /vzEudat0/home/rods0/data/
Oct 9 14:29:14 pid:12033 NOTICE: postProcRunRuleExec: exec of freq: 'REPEAT 3 TIMES'
Oct 9 14:29:14 pid:12033 NOTICE: modExeInfoForRepeat: rullid=10089,opStatus=0,nextRepeatStatus=3
Oct 9 14:29:15 pid:12033 NOTICE: Rule id 10089 set to run again at 1381321754 (frequency 0' REPEAT 2 TIMES. ORIGINAL TIMES=3 seconds)
Oct 9 14:29:15 pid:12035 NOTICE: msiWriteRodsLog message: put a file /vzEudat0/home/rods0/data/put2.txt to a collection /vzEudat0/home/rods0/data/
Oct 9 14:29:15 pid:12035 NOTICE: postProcRunRuleExec: exec of freq: 0' REPEAT 2 TIMES. ORIGINAL TIMES=3
Oct 9 14:29:15 pid:12035 NOTICE: modExeInfoForRepeat: rullid=10089,opStatus=0,nextRepeatStatus=3
Oct 9 14:29:15 pid:12035 NOTICE: Rule id 10089 set to run again at 1381321755 (frequency 0' REPEAT 1 TIMES. ORIGINAL TIMES=3 seconds)
Oct 9 14:29:15 pid:11811 NOTICE: reServerMain: checking the queue for jobs
Oct 9 14:29:15 pid:12037 NOTICE: msiWriteRodsLog message: put a file /vzEudat0/home/rods0/data/put2.txt to a collection /vzEudat0/home/rods0/data/
Oct 9 14:29:15 pid:12037 NOTICE: postProcRunRuleExec: exec of freq: 0' REPEAT 1 TIMES. ORIGINAL TIMES=3
Oct 9 14:29:15 pid:12037 NOTICE: modExeInfoForRepeat: rullid=10089,opStatus=0,nextRepeatStatus=2
Oct 9 14:29:45 pid:11811 NOTICE: reServerMain: checking the queue for jobs
```

# Removing / Deleting data

- `irm /zone/home/user/file1` removes file1 to `/zone/trash/user/file1`

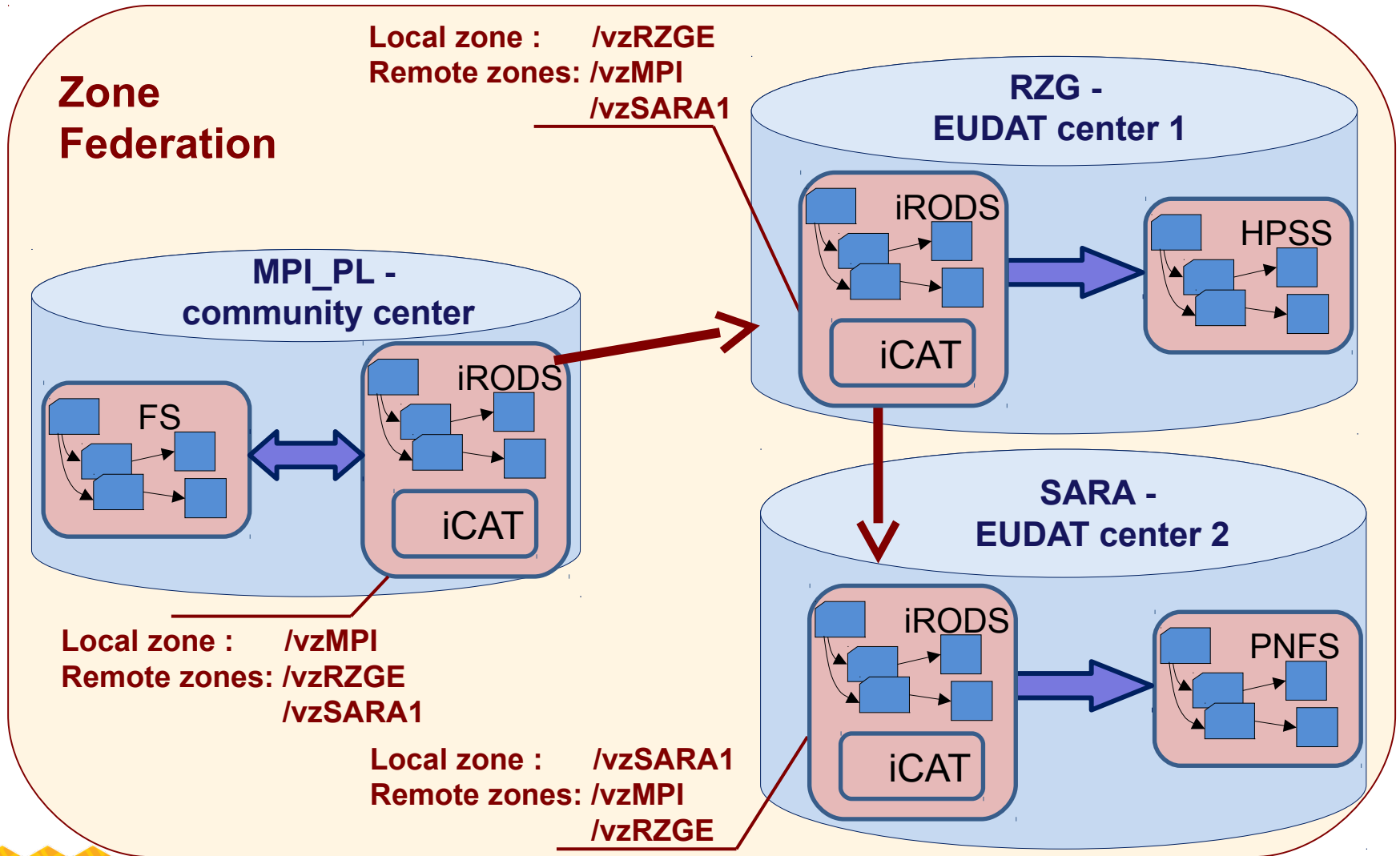
Not physically removed from disk

- `irm -f /zone/home/user/file1` physically deletes file1
- Or use `irmtrash`
- `irm -r /zone/home/user/dir` removes iRODS collection



# III. EUDAT Safe Replication

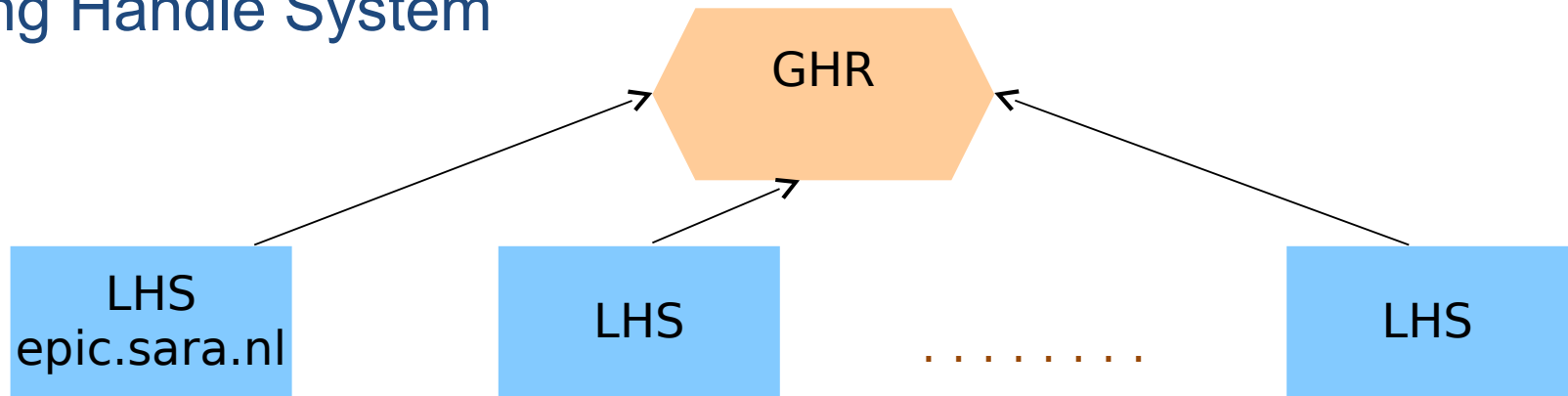
# REPLIX-EUDAT Example





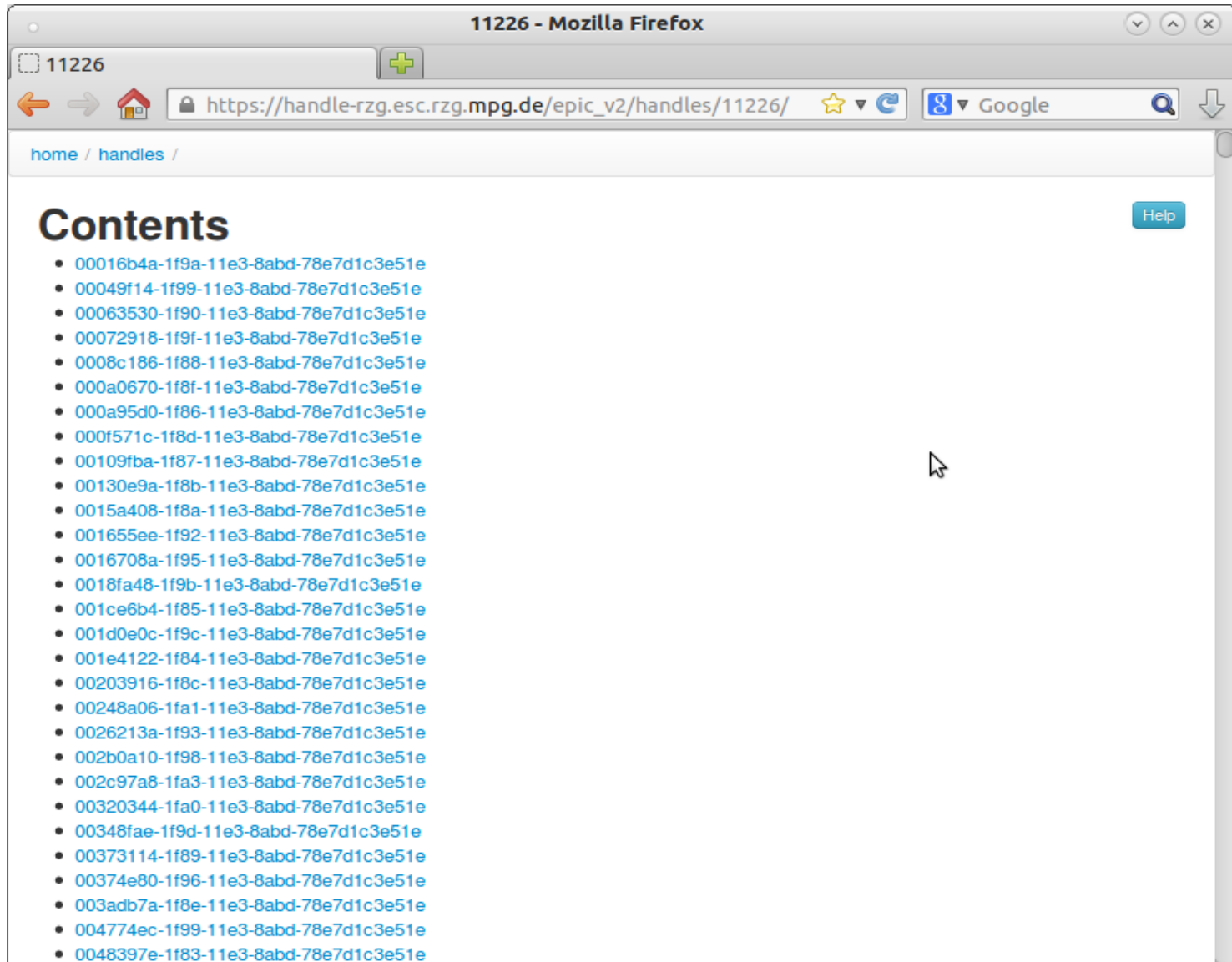
# EPIC API

- Is built on SARA's RESTful Web Service for EPIC PIDs using Handle System



- Data representations: (X)HTML, JSON, plain text
- Namespace:
  - `https://epic.sara.nl/v2_test/handles`
    - └ `<prefix>/`
      - └ `<suffix>`
- Methods: HTTP GET, HTTP PUT, HTTP POST

# EPIC API



The screenshot shows a Mozilla Firefox browser window with the title "11226 - Mozilla Firefox". The address bar contains the URL "https://handle-rzg.esc.rzg.mpg.de/epic\_v2/handles/11226/". The page content includes a breadcrumb "home / handles /", a "Contents" section with a list of 35 blue links, and a "Help" button in the top right corner.

home / handles /

## Contents

Help

- [00016b4a-1f9a-11e3-8abd-78e7d1c3e51e](#)
- [00049f14-1f99-11e3-8abd-78e7d1c3e51e](#)
- [00063530-1f90-11e3-8abd-78e7d1c3e51e](#)
- [00072918-1f9f-11e3-8abd-78e7d1c3e51e](#)
- [0008c186-1f88-11e3-8abd-78e7d1c3e51e](#)
- [000a0670-1f8f-11e3-8abd-78e7d1c3e51e](#)
- [000a95d0-1f86-11e3-8abd-78e7d1c3e51e](#)
- [000f571c-1f8d-11e3-8abd-78e7d1c3e51e](#)
- [00109fba-1f87-11e3-8abd-78e7d1c3e51e](#)
- [00130e9a-1f8b-11e3-8abd-78e7d1c3e51e](#)
- [0015a408-1f8a-11e3-8abd-78e7d1c3e51e](#)
- [001655ee-1f92-11e3-8abd-78e7d1c3e51e](#)
- [0016708a-1f95-11e3-8abd-78e7d1c3e51e](#)
- [0018fa48-1f9b-11e3-8abd-78e7d1c3e51e](#)
- [001ce6b4-1f85-11e3-8abd-78e7d1c3e51e](#)
- [001d0e0c-1f9c-11e3-8abd-78e7d1c3e51e](#)
- [001e4122-1f84-11e3-8abd-78e7d1c3e51e](#)
- [00203916-1f8c-11e3-8abd-78e7d1c3e51e](#)
- [00248a06-1fa1-11e3-8abd-78e7d1c3e51e](#)
- [0026213a-1f93-11e3-8abd-78e7d1c3e51e](#)
- [002b0a10-1f98-11e3-8abd-78e7d1c3e51e](#)
- [002c97a8-1fa3-11e3-8abd-78e7d1c3e51e](#)
- [00320344-1fa0-11e3-8abd-78e7d1c3e51e](#)
- [00348fae-1f9d-11e3-8abd-78e7d1c3e51e](#)
- [00373114-1f89-11e3-8abd-78e7d1c3e51e](#)
- [00374e80-1f96-11e3-8abd-78e7d1c3e51e](#)
- [003adb7a-1f8e-11e3-8abd-78e7d1c3e51e](#)
- [004774ec-1f99-11e3-8abd-78e7d1c3e51e](#)
- [0048397e-1f83-11e3-8abd-78e7d1c3e51e](#)

# EPIC API

1c8a0062-367c-11e3-a2a3-78e...

https://handle-rzg.esc.rzg.mpg.de/epic\_v2/handles/11226/1c8a0062-367c-11e3-a2a3-78e7d1c3e51e

home / handles / 11226 /

idx	type	parsed data	data	timestamp	ttd type	ttd	refs	privs	<a href="#">Help</a>
1	URL	ftp://ftp.rzg.mpg.de/afs/ipp-garching.mpg.de/home/f/fruwl/public/EUDAT/resolver/createPids.py	ZnRwOi8vZnRwLnJ6Zy5tcGcuZGUvYWZzL2lwcC1nYXJjaGluZy5tcGcuZGUvaG9tZS9mL2ZydXdsL3B1YmtpYy9FVURBVC9yZkNvbHJlci9jcmVhdGVQaWRzLnB5	2013-10-16T16:01:13Z	0	86400		rwr-	
2	CHECKSUM	99f9164f84339364ffb22c8eaa5e1206	OTImOTE2NGY4NDMzOTM2NGZmYjlyYzh1YWE1ZTEyMDY=	2013-10-16T16:01:13Z	0	86400		rwr-	
3	10320/LOC	<locations><location href="ftp://ftp.rzg.mpg.de/afs/ipp-garching.mpg.de/home/f/fruwl/public/EUDAT/resolver/createPids.py" id="0"/><location href="file:///afs/ipp-garching.mpg.de/home/f/fruwl/public/EUDAT/resolver/createPids.py" id="1"/></locations>	PGxvY2F0aW9ucz48bG9jYXRpb24gaHJlZj0lZnRwOi8vZnRwLnJ6Zy5tcGcuZGUvYWZzL2lwcC1nYXJjaGluZy5tcGcuZGUvaG9tZS9mL2ZydXdsL3B1YmtpYy9FVURBVC9yZkNvbHJlci9jcmVhdGVQaWRzLnB5IiBpZD0IMCivPjxsb2Nh dGlvb1BocmVmpSJMawXlOI8vL2Fmcy9pcHATZ2FyY2hpbmcutXBnLmRlL2hv bWUvZi9mcnV3bC9wdWJsaWMvRVVEQVQvcmlVzb2x2ZXIvY3JlYXRIUGlkcy5w eSlgaWQ9IjEILz48L2xvY2F0aW9ucz4=	2013-10-16T16:01:14Z	0	86400		rwr-	
100	HS_ADMIN	<b>adminid</b> 0,NA/11226 <b>adminidIndex</b> 200 <b>perms</b> <b>add_handle</b> true <b>delete_handle</b> true <b>add_naming_auth</b> false <b>delete_naming_a...</b> false <b>modify_value</b> true <b>remove_value</b> true <b>add_value</b> true <b>read_value</b> true <b>modify_admin</b> true <b>remove_admin</b> true <b>add_admin</b> true <b>list_handles</b> false	B/MAAAAKMC5OQS8xMTIyNgAAAMgAAA==	2013-10-16T16:01:13Z	0	86400		rwr-	

EPIC PID-Service version 2.3

Developed by the EPIC Consortium: CSC, DKRZ, GWDG, and SURFara

Sponsored by: CATCH+, DARIAH-DE and EUDAT

Powered by Rackful

# EPIC API

The screenshot shows a Mozilla Firefox browser window titled "Handle Proxy - Mozilla Firefox". The address bar contains "epic.rzg.mpg.de:8000". The page content includes the "Handle System" logo, the title "Handle System Proxy Server", and several paragraphs of text explaining the proxy server's functionality. Below the text is a form with a "Handle:" input field, a "Value types:" dropdown menu (showing URL, EMAIL, HS\_ADMIN, HS\_ALIAS, HS\_SITE), three checkboxes for "Authoritative Query", "Don't Redirect to URLs" (checked), and "Don't Follow Aliases", and a "Submit" button. At the bottom, there is a link to the "Handle System Web Site" and contact information for "hdladmin@cnri.reston.va.us".

Handle Proxy

epic.rzg.mpg.de:8000

Google

## Handle System<sup>®</sup>

### Handle System Proxy Server

The web form below will enable you to resolve individual handles and view their associated values. It uses a proxy server, which understands both the Handle System protocol and HTTP protocol.

If you type a handle into the text box, and that handle has a URL associated with it as one of its values, the proxy server will instruct your browser to display the location of that URL. If you select "Don't Redirect to URLs", the proxy will simply list the value. You may select other value types from the pick list, although most handles will not have all of the listed administrative values associated with them.

The Handle System uses caching to speed handle resolution. If you check "Authoritative Query", the proxy will bypass the cache, go directly to the responsible handle server, and then refresh the cache with the data for that handle.

Simply appending a handle to the URL <http://hdl.handle.net/> and giving the string to a browser as a location will also resolve that handle, but using this form will enable you to see all of the handles values.

Handle:

Value types:

- URL
- EMAIL
- HS\_ADMIN
- HS\_ALIAS
- HS\_SITE

Authoritative Query

Don't Redirect to URLs

Don't Follow Aliases

[Handle System Web Site](#)

Please contact [hdladmin@cnri.reston.va.us](mailto:hdladmin@cnri.reston.va.us) for your handle questions and comments.

# EPIC API

Handle Proxy - Mozilla Firefox

Handle Proxy

epic.rzg.mpg.de:8000

Google

## Handle System®

**Handle:** 11226/1c8a0062-367c-11e3-a2a3-78e7d1c3e51e

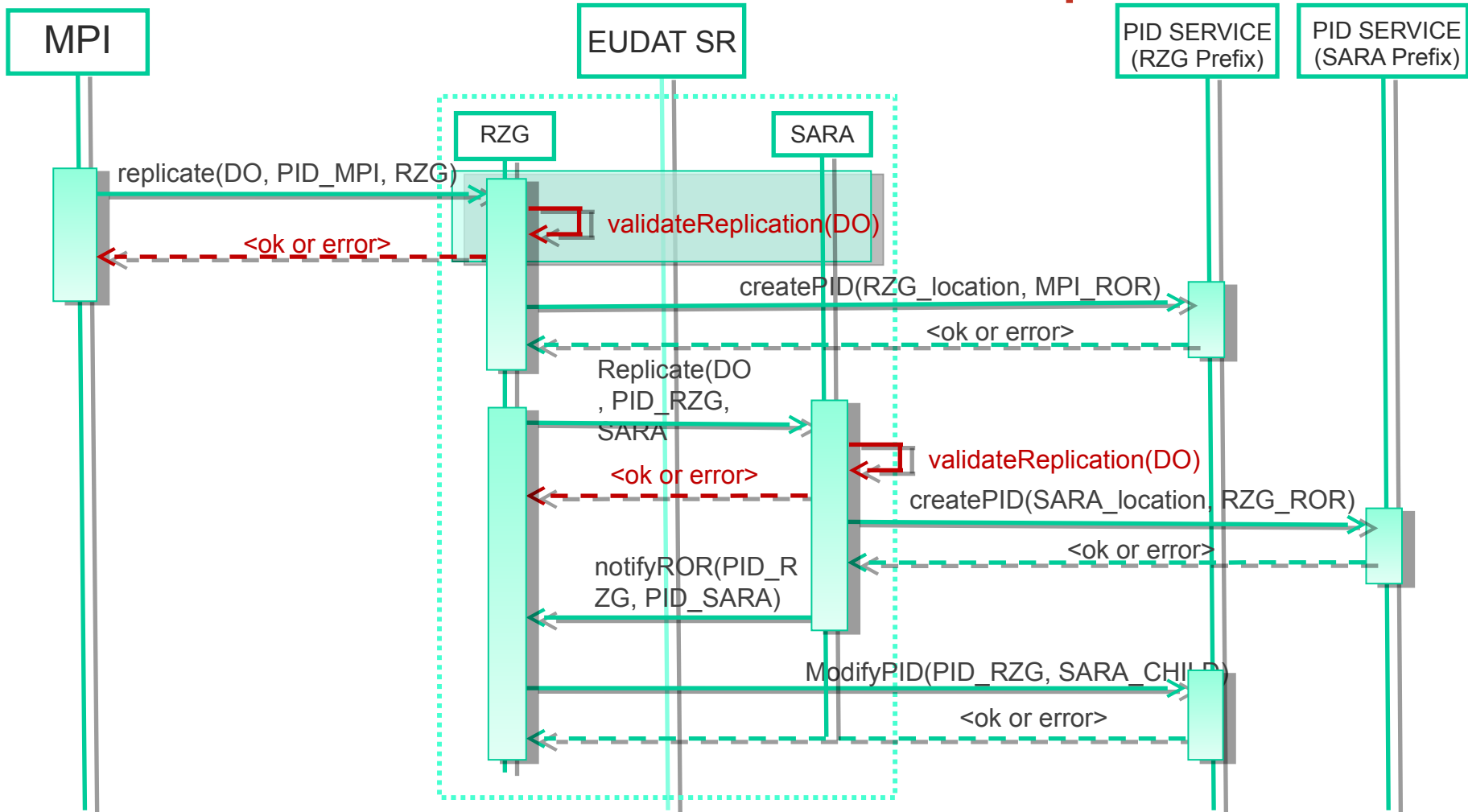
**Handle Values for:** 11226/1c8a0062-367c-11e3-a2a3-78e7d1c3e51e

Index	Type	Timestamp	Data
1	URL	Wed Oct 16 201 18:01:13 CES	<a href="ftp://ftp.rzg.mpg.de/afs/ipp-garching.mpg.de/home/f/fruwl/public/EUDAT/resolver/createPids.py">ftp://ftp.rzg.mpg.de/afs/ipp-garching.mpg.de/home/f/fruwl/public/EUDAT/resolver/createPids.py</a>
2	CHECKSUM	Wed Oct 16 201 18:01:13 CES	99f9164f84339364ffb22c8eaa5e1206
3	10320/LOC	Wed Oct 16 201 18:01:14 CES	<locations><location href="ftp://ftp.rzg.mpg.de/afs/ipp-garching.mpg.de/home/f/fruwl/public/EUDAT/resolver/createPids.py" id="0"/><location href="file:///afs/ipp-garching.mpg.de/home/f/fruwl/public/EUDAT/resolver/createPids.py" id="1"/></locations>
100	HS_ADMIN	Wed Oct 16 201 18:01:13 CES	07F3000000A302E4E412F3131323236000000C80000

[Handle System Web Site](#)

Please contact [hldadmin@cnri.reston.va.us](mailto:hldadmin@cnri.reston.va.us) for your handle questions and comments.

# REPLIX-EUDAT Example

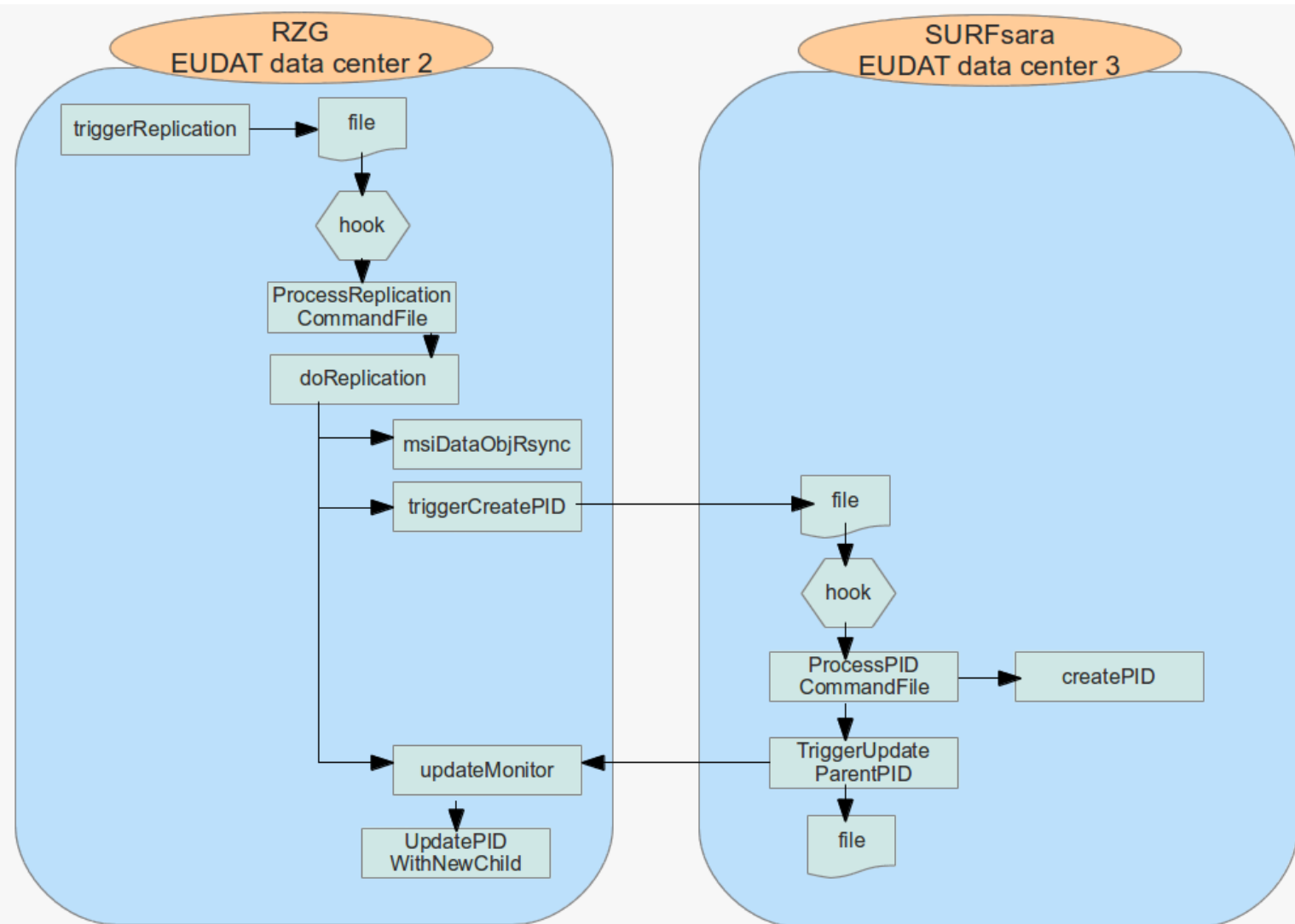




# EUDAT PID Module

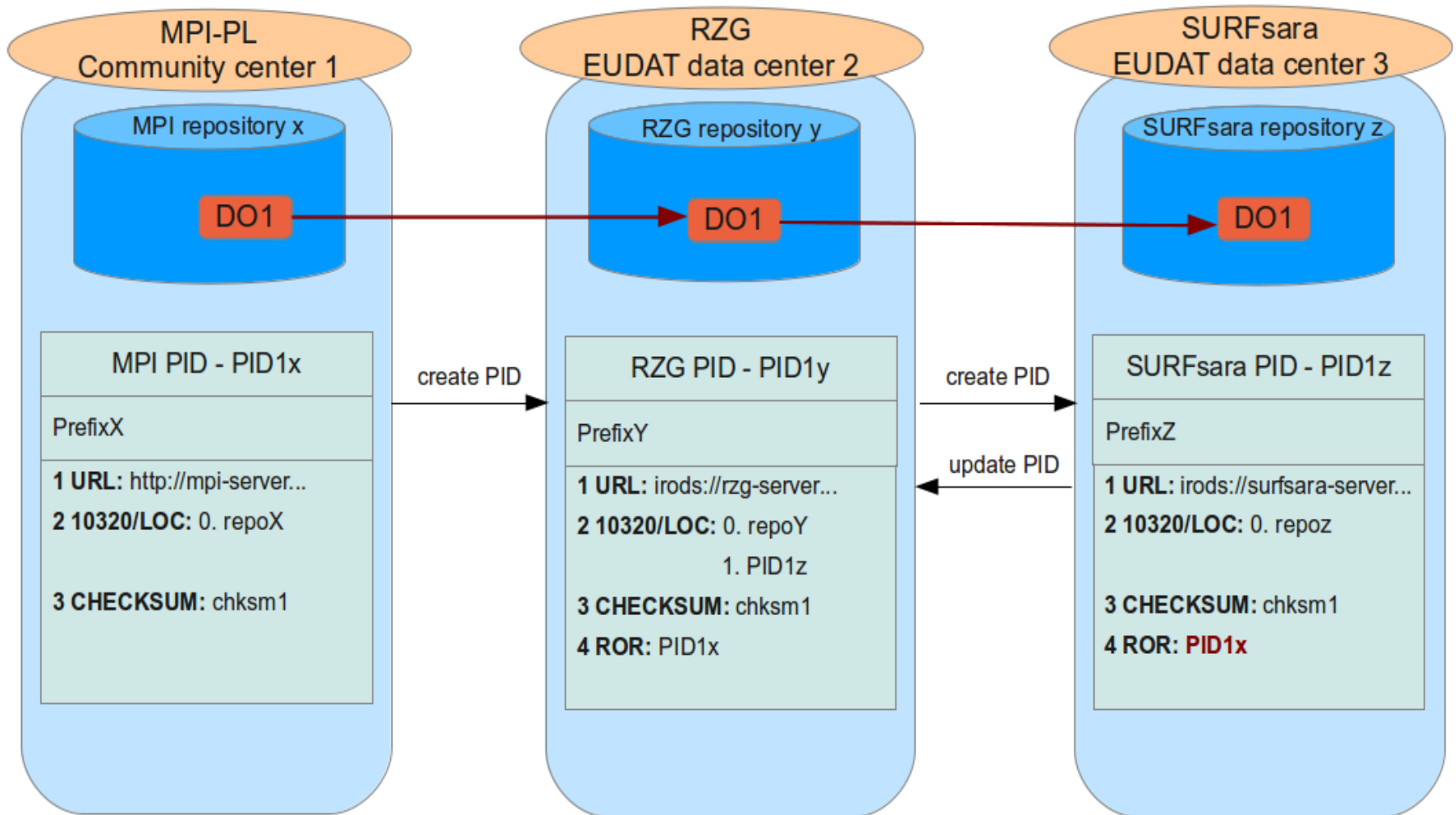
- `~/dice/iRODS/modules/EUDAT-PID`
- Python script to interact with EPIC system
- Rule base file to call python methods within iRODS  
`msiExecCmd("epicclient.py", "*credStoreType *credStorePath  
create *serverID*path", "null", "null", "null", *out);`
- Control files `*.replicate`, `*.pid.create`, and `*.pid.update`

# REPLIX-EUDAT Example



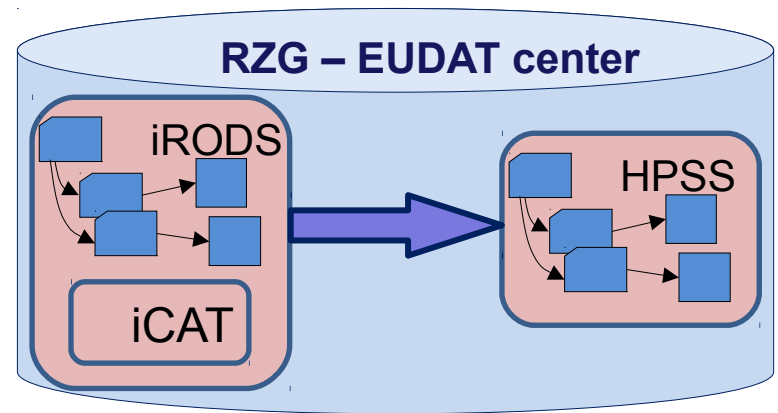


# REPLIX-EUDAT Example



# REPLIX-EUDAT Example

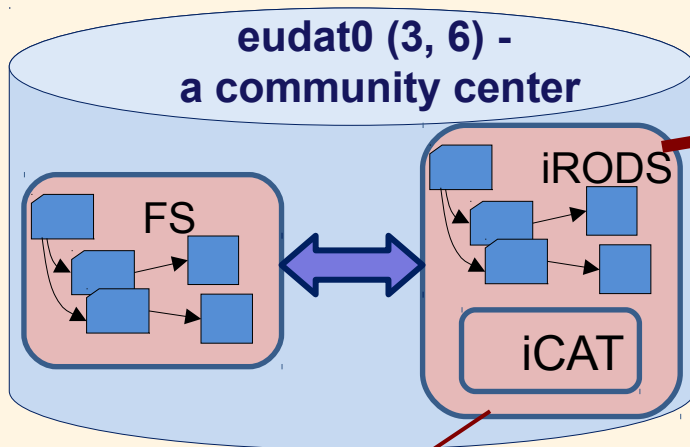
- iRODS is connected to the HPSS via the **Universal MSS Driver**
- Implemented using a **compound** iRODS resource



# Our Training Example

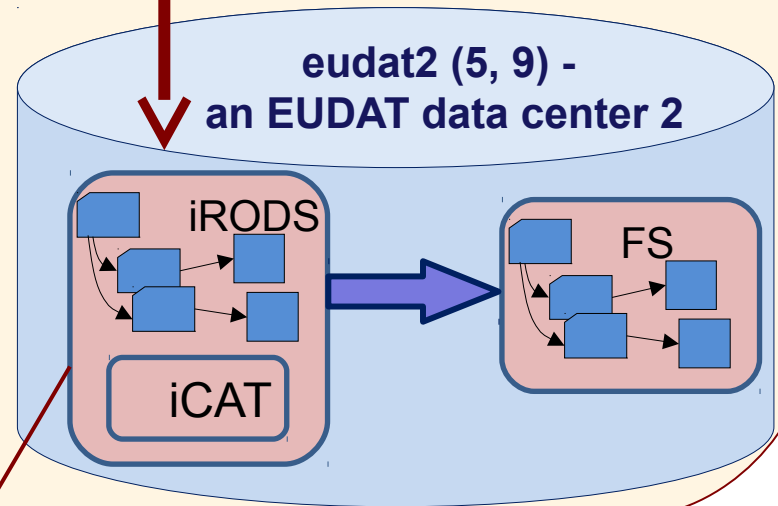
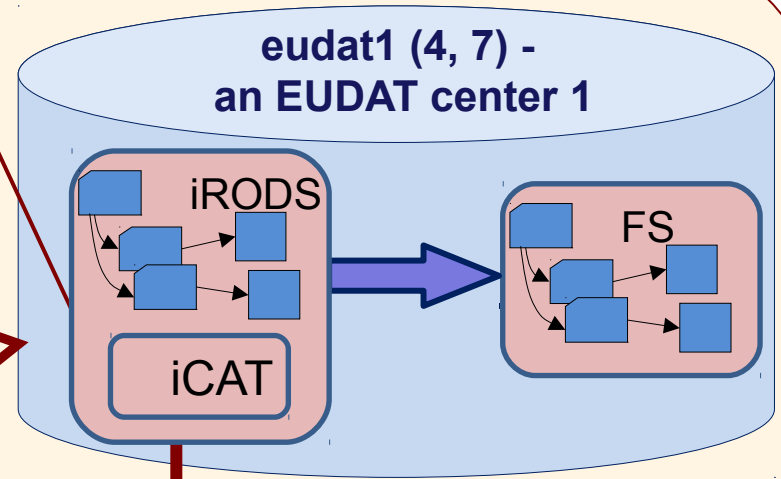
## Zone Federation

Local zone : /vzEudat1  
Remote zones: /vzEudat0  
/vzEudat2



Local zone : /vzEudat0  
Remote zones: /vzEudat1  
/vzEudat2

Local zone : /vzEudat2  
Remote zones: /vzEudat0  
/vzEudat1



# Our Training Example: Setup

- “Community” VMs: eudat0, eudat3, eudat6
- “Data center 1” VMs: eudat1, eudat4, eudat7
- “Data center 2” VMs: eudat2, eudat5, eudat9
- EPIC API URL: [https://epic.sara.nl/v2\\_test/handles/](https://epic.sara.nl/v2_test/handles/)
- EPIC API prefix: 843
- Credentials: `~/dice/iRODS/modules/EUDAT-PID/cmd/credentials`

# Our Training Example: Community: Upload Data and Create PIDs

- Machines: eudat0, eudat3, eudat6
- 
- Data in `~/comm_data` on the FS
- Put in iRODS collection `/vzEudat0/comm_data`  
`mkdir`  
`chmod`
- Create PIDs  
set up `acPostProcForPut` in `core.re`  
use `addPIDWithChecksum(*path, *newPID)` from the EUDAT rule base  
input a file
- Check PID in [https://epic.sara.nl/v2\\_test/handles/](https://epic.sara.nl/v2_test/handles/)

# Our Training Example: Community: Replicate Data to EUDAT Center 1

- **Machines:** eudat0, eudat3, eudat6
- **Trigger a replication rule with irule -F**  
use `CheckReplicas(*source, *destination, *destResc, *commandFile)` from the EUDAT rule base:
  - It compares checksums of source and destination files,  
If they are different – the replication is started:
    - Search PID of the file
    - Sync the file to the remote zone
    - Create PID for a replica
    - Update the parent PID

# Our Training Example: EUDAT Data Center 1: Receive Data, Copy to a Storage Resource and Replicate to EUDAT Center 2

- Machines: eudat1, eudat4, eudat7
- Data is received in /vzEudat1/center1\_data in iRODS
  - imkdir
  - ichmod
- Configure storage resources
  - iadmin mkresc
- Configure copying to a storage resource
  - set up acPostProcForPut in core.re
  - use repl
- Replicate data to the EUDAT data center 2
  - irule -F
  - use CheckReplicas from the EUDAT rule base
- Check PIDs in [https://epic.sara.nl/v2\\_test/handles/](https://epic.sara.nl/v2_test/handles/)

## Our Training Example: EUDAT Data Center 2: Receive Data and Copy to a Storage Resource

- Machines: eudat2, eudat5, eudat9
- Data is received in /vzEudat2/center2\_data in iRODS
  - imkdir
  - ichmod
- Configure storage resources
  - iadmin mkresc
- Configure copying to a storage resource
  - set up acPostProcForPut in core.re
  - use repl
- Check PIDs in [https://epic.sara.nl/v2\\_test/handles/](https://epic.sara.nl/v2_test/handles/)



# Practice

Community eudat-0, eudat-3, eudat-6	Data center 1 eudat-1, eudat-4, eudat-7	Data center 2 eudat-2, eudat-5, eudat-9
<pre>- create a file cd ~/comm_data/ vi repl_file_&lt;Name&gt; - put it to iRODS icd /vzEudat0/comm_data iput repl_file_&lt;Name&gt; - check log cat ~/dice/iRODS/server/log/rodsLog.&lt;Date&gt; - replicate the file cd ~/rules cp check.r check_&lt;Name&gt;.r vi check_&lt;Name&gt;.r irule -vF check_&lt;Name&gt;.r - check the PID on https://epic.sara.nl/v2_test/handles/843/ - check the file in the remote zone ils /vzEudat1/center1_data - check the PID update in 2 min cat ~/dice/iRODS/server/log/reLog.&lt;Date&gt;</pre>	<pre>- check the replication to the storage resource ils -l /vzEudat1/center1_data  - replicate to the data center2 cd ~/rules cp check.r check_&lt;Name&gt;.r vi check_&lt;Name&gt;.r irule -vF check_&lt;Name&gt;.r  - check the PID on https://epic.sara.nl/v2_test/handles/843/  - check log cat ~/dice/iRODS/server/log/rodsLog.&lt;Date&gt;  - check the file in the remote zone ils -l /vzEudat2/center2_data  - check the PID update in 2 min cat ~/dice/iRODS/server/log/reLog.&lt;Date&gt;</pre>	<pre>- check the replication to the storage resource ils -l /vzEudat2/center2_data  - check log cat ~/dice/iRODS/server/log/rodsLog.&lt;Date&gt;  - check the PID on https://epic.sara.nl/v2_test/handles/843/  ----- For fun:  - replicate data to the data center1 (to the center2_repl collection) cd ~/rules cp check.r check_&lt;Name&gt;.r vi check_&lt;Name&gt;.r irule -vF check_&lt;Name&gt;.r  - see center 1 instructions</pre>