

Further aspects of EGEE middleware components

INFN, Catania



- An introduction to certificates on Gilda
- Job Management's Command Line Interface
- Data Management's Command Line Interface

An introduction to certificates

Three steps are required to access Grid resources:

- *Authenticated: request a certificate to a Certification Authority (CA)*
- *Authorized: register it into a Virtual Organization (VO) server*
- *Using the certificate: install the certificate in the host that gives you access to the grid*

Obtaining a certificate

- Get GILDA CA Certificate: go to <http://gilda.ct.infn.it> and follow the instructions for users;
- Request a GILDA Personal Certificate : when received confirmation mail, download the certificate using the same browser that made the request and with its exporting certificate procedure, save it into a file. You will also asked to insert the password of the certificate.

Install your certificate on the UI

- Log in into the UserInterface, copy there the file you exported, and create a directory where your certificate + private key will be stored:

```
mkdir ~/.globus
```

```
openssl pkcs12 -nocerts -in mycert.p12 -out  
~/.globus/userkey.pem
```

```
openssl pkcs12 -clcerts -nokeys -in mycert.p12 -out  
~/.globus/usercert.pem
```

```
chmod 0400 ~/.globus/userkey.pem
```

```
chmod 0644 ~/.globus/usercert.pem
```

Register to the Virtual Organization

- Select the VO that you want belong to and then submit the form;
- You will receive confirmation for your request, and a notification when it will be processed.

With your .pem certificates you can use Globus Security Infrastructure.

To work on a UI user needs a valid proxy authentication, which can be retrieve with the command

```
>> grid-proxy-init
Your identity: /C=FR/O=CNRS/OU=LAL/CN=Charles
Loomis/Email=loomis@lal.in2p3.fr
Enter GRID pass phrase for this identity: *****
Creating proxy
..... Done
Your proxy is valid until Tue Aug 13 03:15:11 2002
```

Proxy default lifetime is 12 hours.

To obtain information about a generated proxy, you can use the command `grid-proxy-info`:

```
>> grid-proxy-info
subject : /C=FR/O=CNRS/OU=LAL/CN=Charles
Loomis/Email=loomis@lal.in2p3.fr/CN=proxy
issuer : /C=FR/O=CNRS/OU=LAL/CN=Charles
Loomis/Email=loomis@lal.in2p3.fr
type : full
strength : 512 bits
timeleft : 11:36:17
```

To destroy explicitly the proxy before it has expired, use the command `>>grid-proxy-destroy`

Long jobs may outlive the validity of the initial proxy: if so, the job would die. To avoid this the wms allows the proxy to be renewed automatically.

First register a proxy with the MyProxy server using the command

```
>>myproxy-init -s <server> -t <hours>
```

Retrieve a valid delegation from the Myproxy server through the command

```
>>myproxy-get-delegation -s <server>
```

Information about your stored proxy can be obtained via the command

```
>>myproxy-info -s <server>
```

and the proxy can be removed with

```
>>myproxy-destroy -s <server>
```

- The user interacts with Grid via a **Workload Management System (WMS)**
- The Goal of WMS is the **distributed scheduling and resource management in a Grid environment.**
- What does it allow Grid users to do?
 - To submit their jobs
 - To execute them on the “best resources”
 - The WMS tries to optimize the usage of resources
 - To get information about their status
 - To retrieve their output

Job submission

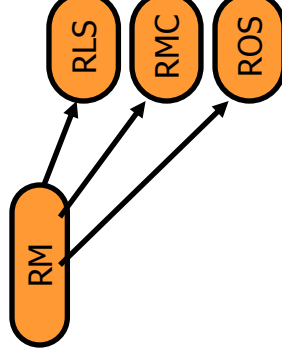
- **edg-job-submit** **[-r <res_id>] [-c <config file>] [-vo <VO>] [-o <output file>] <job.jdl>**
 - **-r** the job is submitted directly to the computing element identified by *<res_id>*
 - **-c** the configuration file *<config file>* is pointed by the UI instead of the standard configuration file
 - **-vo** the Virtual Organization (if user is not happy with the one specified in the UI configuration file)
 - **-o** the generated *edg_jobId* is written in the *<output file>*
 - Useful for other commands, e.g.:
 - **edg-job-status -i <input file> (or edg_jobId)**
 - **-i** the status information about *edg_jobId* contained in the *<input file>* are displayed

Other (most relevant) UI commands

- **edg-job-list-match**
 - Lists resources matching a job description
 - Performs the matchmaking without submitting the job
- **edg-job-cancel**
 - Cancels a given job
- **edg-job-status**
 - Displays the status of the job
- **edg-job-get-output**
 - Returns the job-output (the OutputSandbox files) to the user
- **edg-job-get-logging-info**
 - Displays logging information about submitted jobs (all the events “pushed” by the various components of the WMS)
 - Very useful for debug purposes

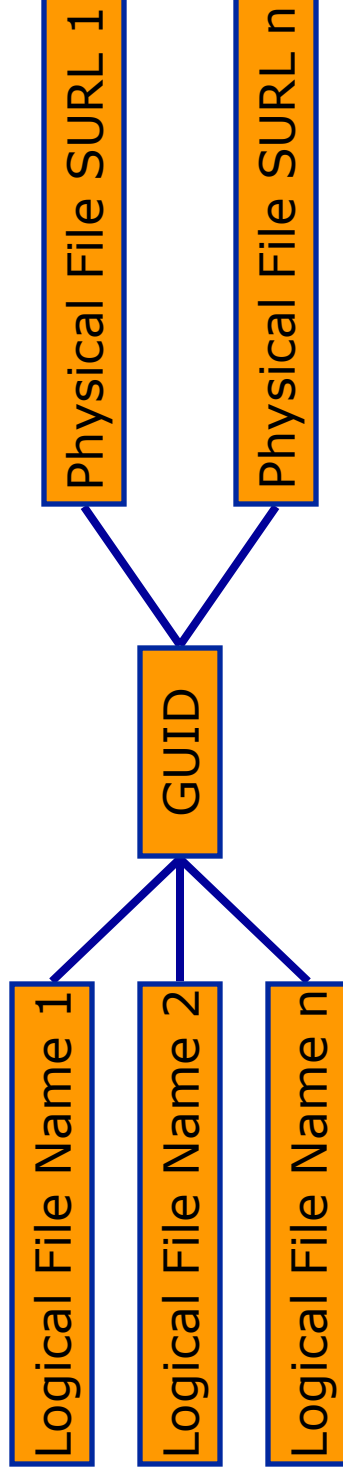
Data Management Tools

- Tools for
 - Locating data
 - Copying data
 - Managing and replicating data
 - Meta Data management
- On EDG you have
 - Replica Location Service (RLS)
 - Replica Metadata Service (RMC)
 - Replica Optimisation Service (ROS)
 - Replica Manager (RM)



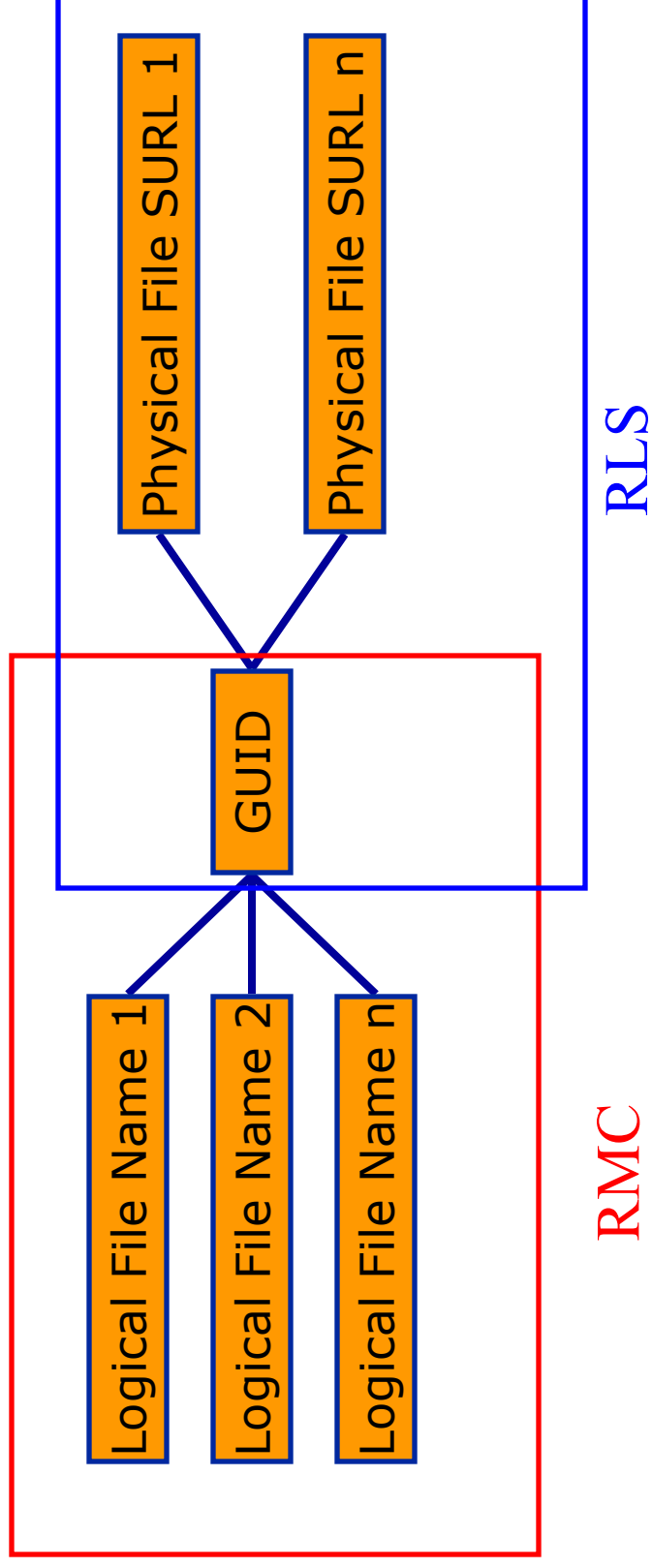
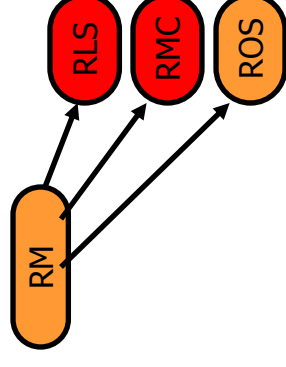
Naming Conventions

- Logical File Name (**LFN**)
 - An alias created by a user to refer to some item of data e.g.
“fn:cms/20030203/run2/track1”
- Site URL (**SURL**) (or Physical File Name (**PFN**))
 - The location of an actual piece of data on a storage system e.g.
“srm://pcrd24.cern.ch/flatfiles/cms/output10_1”
- Globally Unique Identifier (**GUID**)
 - A non-human readable unique identifier for an item of data e.g.
“guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”



Replica Metadata Catalog (RMC) vs. Replica Location Service (RLS)

- **RMC:**
 - Stores LFN-GUID mappings
- **RLS:**
 - Stores GUID-SURL mappings



The Replica Manager Interface – Management Commands

- [copyAndRegisterFile](#)

args: source, dest,

lfn, protocol, streams

- Copy a file into grid-aware storage and register the copy in the Replica Catalog as an atomic operation.

- [replicateFile](#)

args: source/lfn,

dest, protocol, streams

- Replicate a file between grid-aware stores and register the replica in the Replica Catalog as an atomic operation.

- [deleteFile](#)

args: source/seHost, all

- Delete a file from storage and unregister it.

- Example

```
edg-rm --vo=tutor copyAndRegisterFile  
file:/home/bob/analysis/data5.dat  
-d lxshare0384.cern.ch
```


The Replica Manager Interface – Catalog Commands (1)

- [registerFile](#) args: source, lfn
 - Register a file in the Replica Catalog that is already stored on a Storage Element.
- [unregisterFile](#) args: source, guid
 - Unregister a file from the Replica Catalog.
- [listReplicas](#) args: lfn/surl/guid
 - List all replicas of a file.
- [registerGUID](#) args: surl, guid
 - Register an SURL with a known GUID in the Replica Catalog.
- [listGUID](#) args: lfn/surl
 - Print the GUID associated with an LFN or SURL.

The Replica Manager Interface – *Optimization Commands*

- [listBestFile](#) args: lfn/guid, seHost
 - Return the 'best' replica for a given logical file identifier.
- [getBestFile](#) args:
lfn/guid, seHost, protocol, streams
 - Return the storage file name (SFN) of the best file in terms of network latencies.

The Replica Manager Interface – *File Transfer Commands*

- [copyFile](#) args: source, dest
 - Copy a file to a non-grid destination.
- [listDirectory](#) args: dir
 - List the directory contents on an SRM or a GridFTP server.

Let's go to the practical