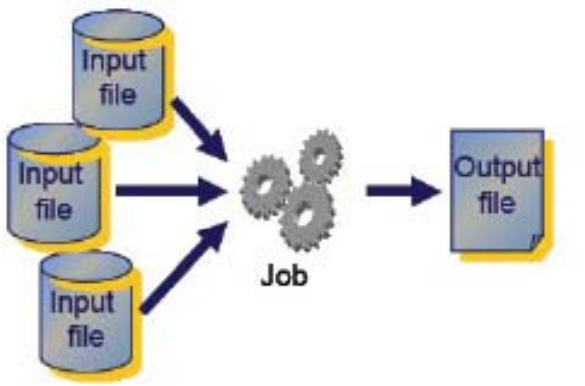
# Laboratory: Hands using EGEE Grid (gLite)

***Athanasia Asiki***
***aassiki@cslab.ece.ntua.gr***
***Computing Systems Laboratory,***
***National Technical University of Athens***

# Let's enter the Grid…

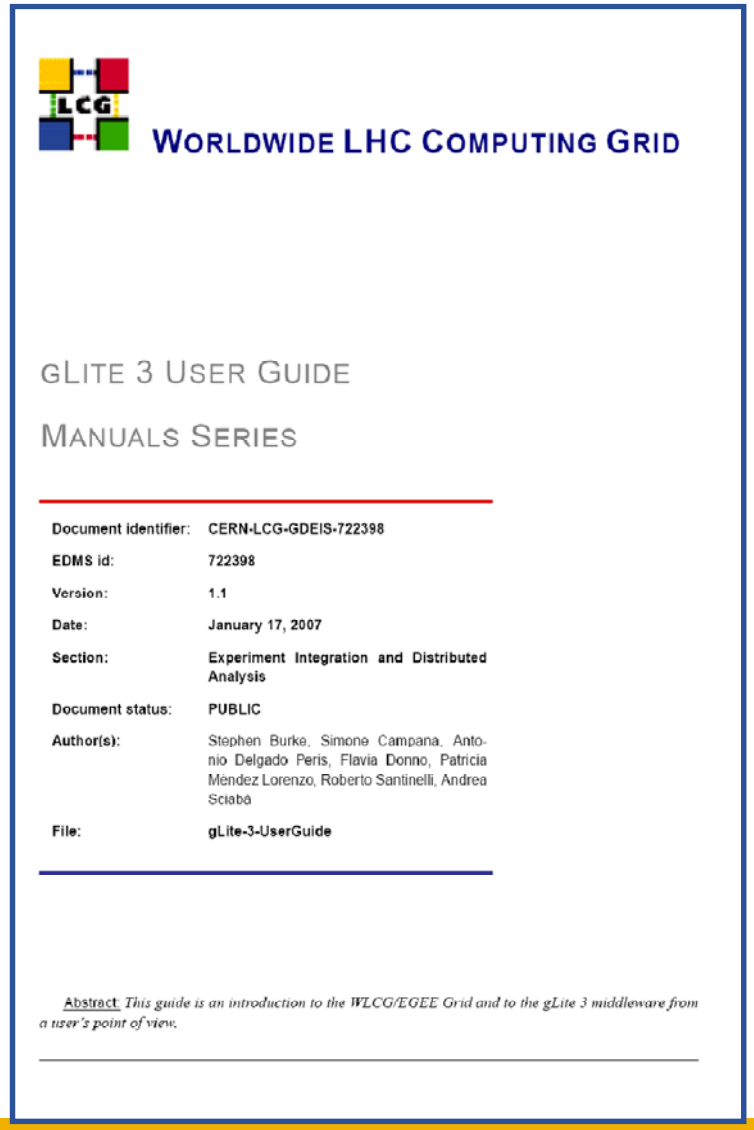
# Application's structure




- **The execution of a typical Grid application follows this scenario:**
  - The user submits its application's job to the "Grid"
  - The job is being executed
  - The job's execution may include the processing of one or more **Input Files** stored in a Storage node
  - The job may produce one or more **Output Files**
  - The **Output Files** can be stored somewhere in the Grid system (perhaps in the Storage Element or in the User Interface)
  - The User can access the **Output Files** using the corresponding Grid mechanisms

# Documentation

http://glite.web.cern.ch/glite/documentation/



WORLDWIDE LHC COMPUTING GRID

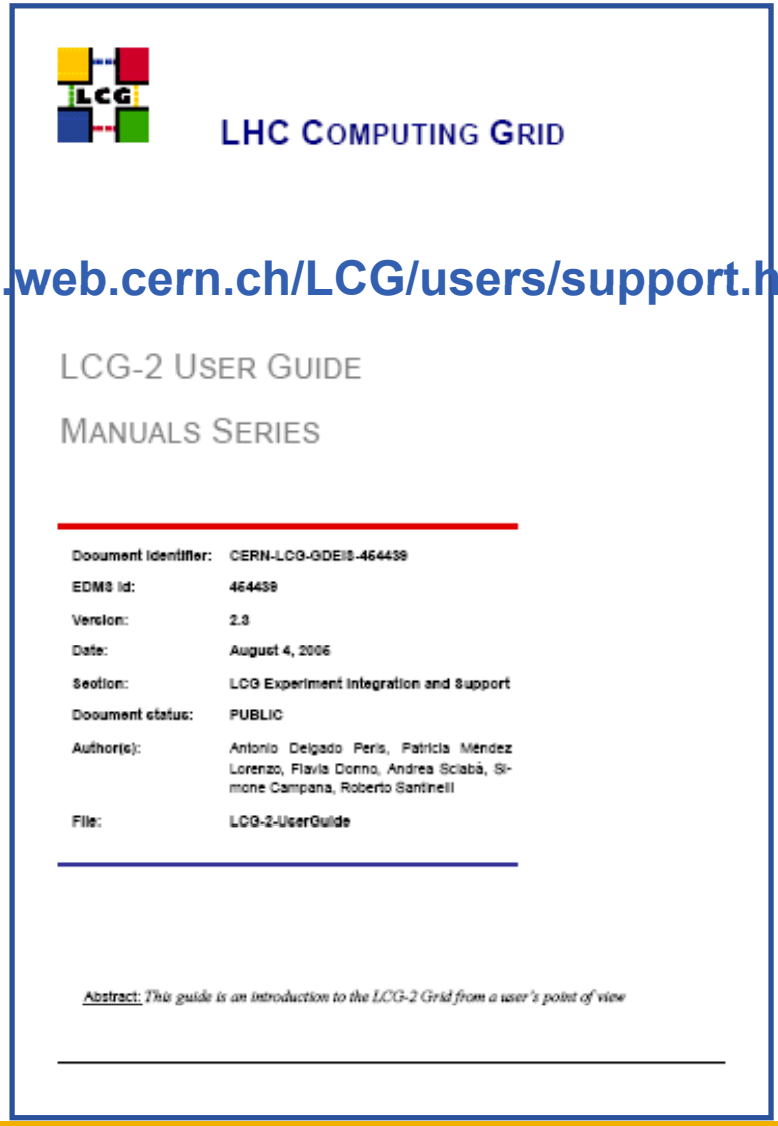
gLITE 3 USER GUIDE

MANUALS SERIES

| | |
|---|---|
| Document identifier: | CERN-LCG-GDEIS-722398 |
| EDMS id: | 722398 |
| Version: | 1.1 |
| Date: | January 17, 2007 |
| Section: | Experiment Integration and Distributed Analysis |
| Document status: | PUBLIC |
| Author(s): | Stephen Burke, Simone Campana, Antonio Delgado Peris, Flavia Donno, Patricia Méndez Lorenzo, Roberto Santinelli, Andrea Sciabà |
| File: | gLite-3-UserGuide |

Abstract: *This guide is an introduction to the WLCG/EGEE Grid and to the gLite 3 middleware from a user's point of view.*

http://lcg.web.cern.ch/LCG/users/support.html



LHC COMPUTING GRID

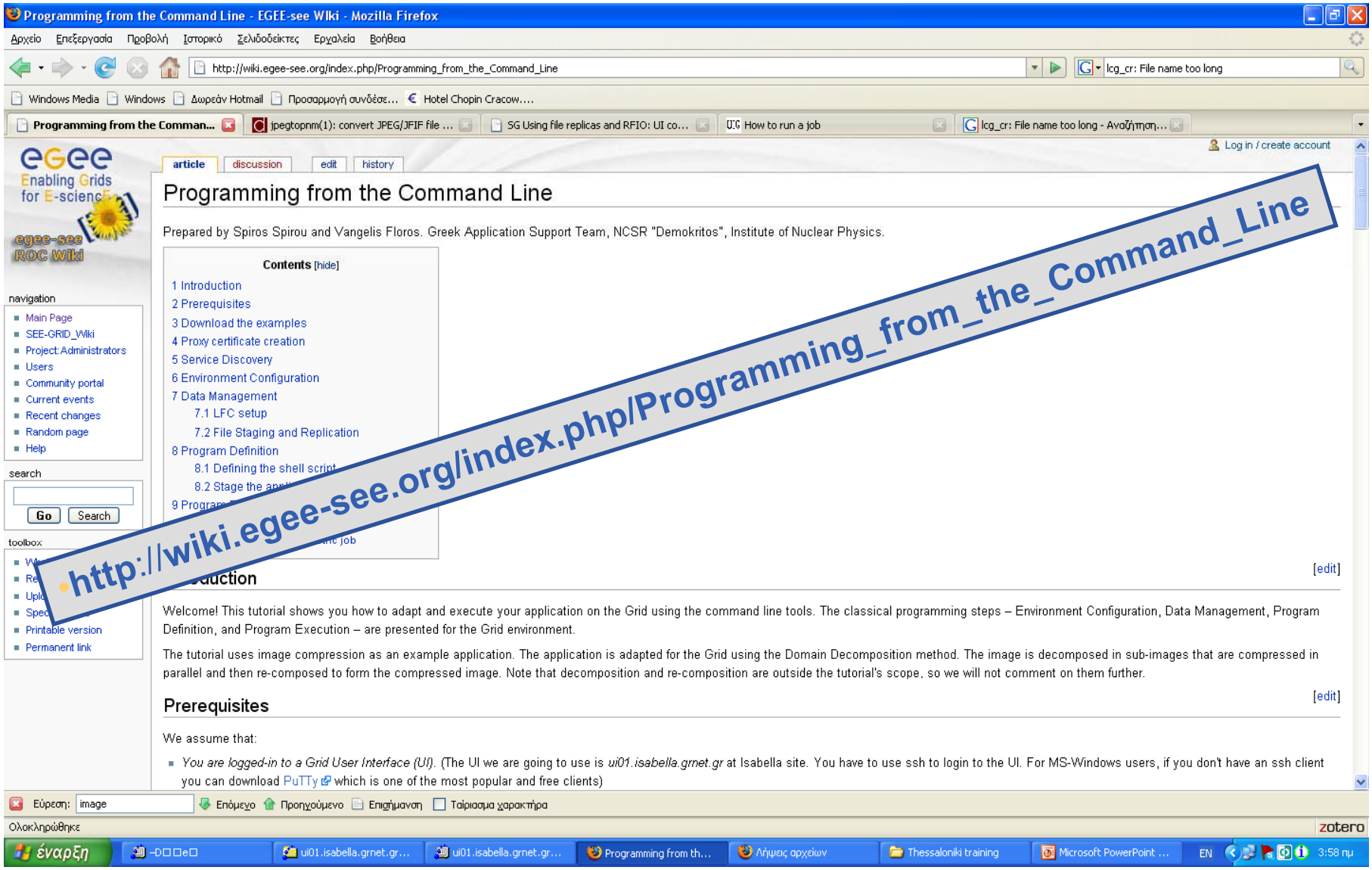
LCG-2 USER GUIDE

MANUALS SERIES

| | |
|---|---|
| Document identifier: | CERN-LCG-GDEIS-454439 |
| EDMS id: | 454439 |
| Version: | 2.3 |
| Date: | August 4, 2006 |
| Section: | LCG Experiment Integration and Support |
| Document status: | PUBLIC |
| Author(s): | Antonio Delgado Peris, Patricia Méndez Lorenzo, Flavia Donno, Andrea Sciabá, Simone Campana, Roberto Santinelli |
| File: | LCG-2-UserGuide |

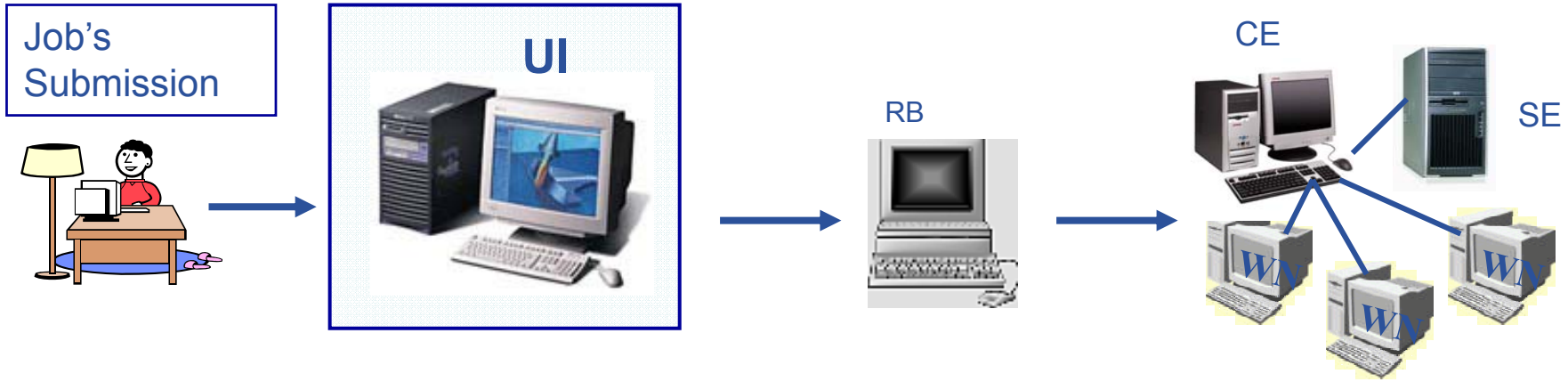
Abstract: *This guide is an introduction to the LCG-2 Grid from a user's point of view*

# Programming from the Command Line



# User Interface (1)

- Allows users to access Grid functionalities
- A machine where users have a personal account and where the user certificate is installed
- Gateway to Grid Services



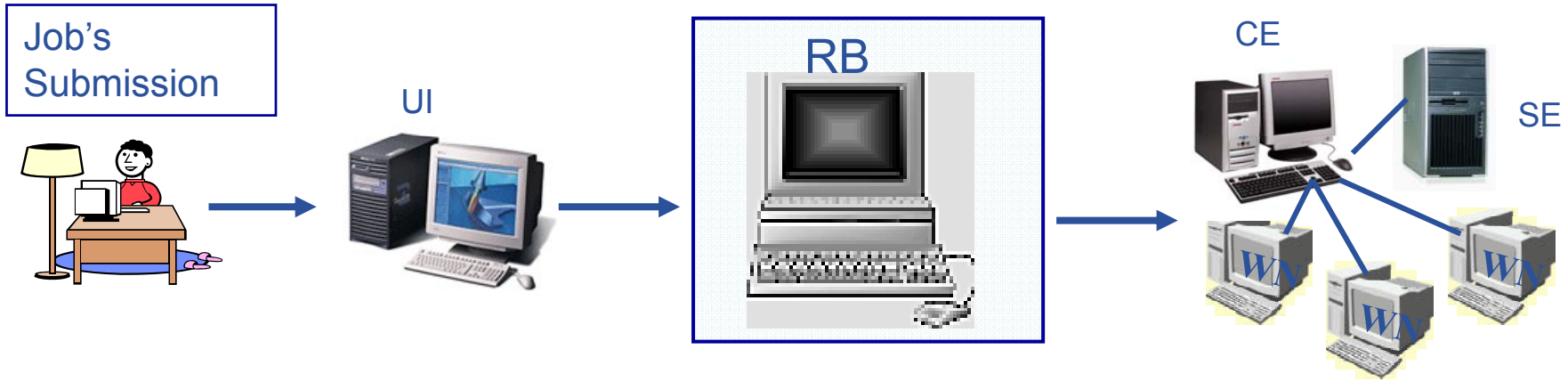
# User Interface (2)

- *It provides a Command Line Interface to perform some basic Grid operations such as:*

- List all the resources suitable to execute a given job
- Submit jobs for execution
- Show the status of submitted jobs
- Cancel one or more jobs
- Retrieve the logging and bookkeeping information of jobs
- Retrieve the output of finished jobs
- Copy, replicate and delete files from Grid

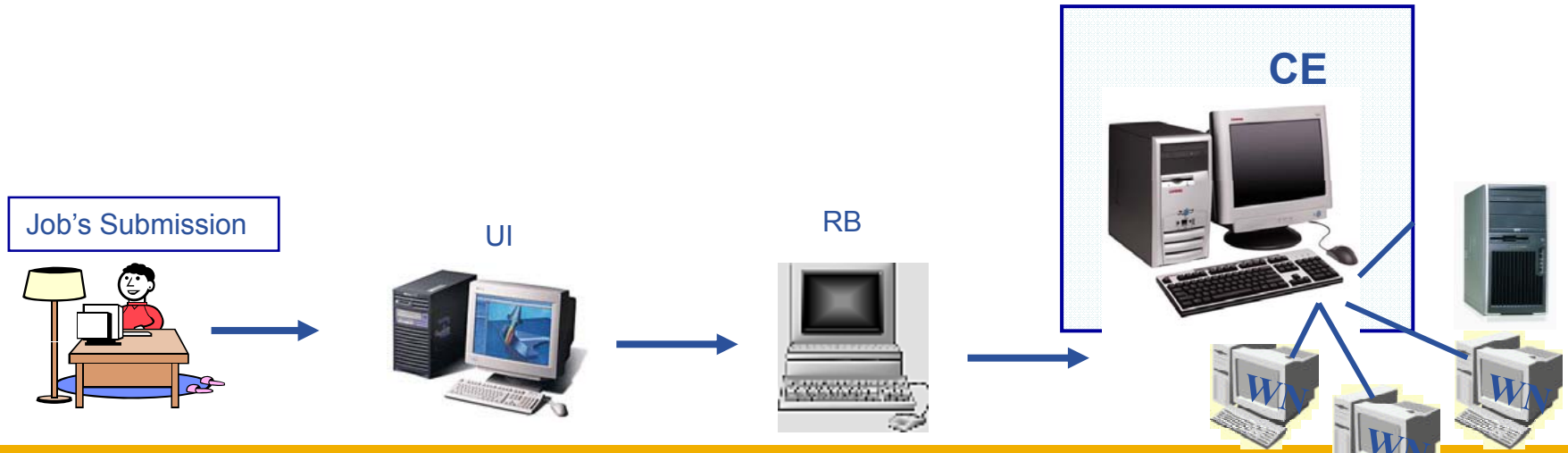
# Workload Management System

- **The resource broker is responsible for the acceptance of submitted jobs and for sending those jobs to the appropriate Computing Element**
- **Retrieves information from Information Catalogues so as to find the proper available resources depending on the job requirements**



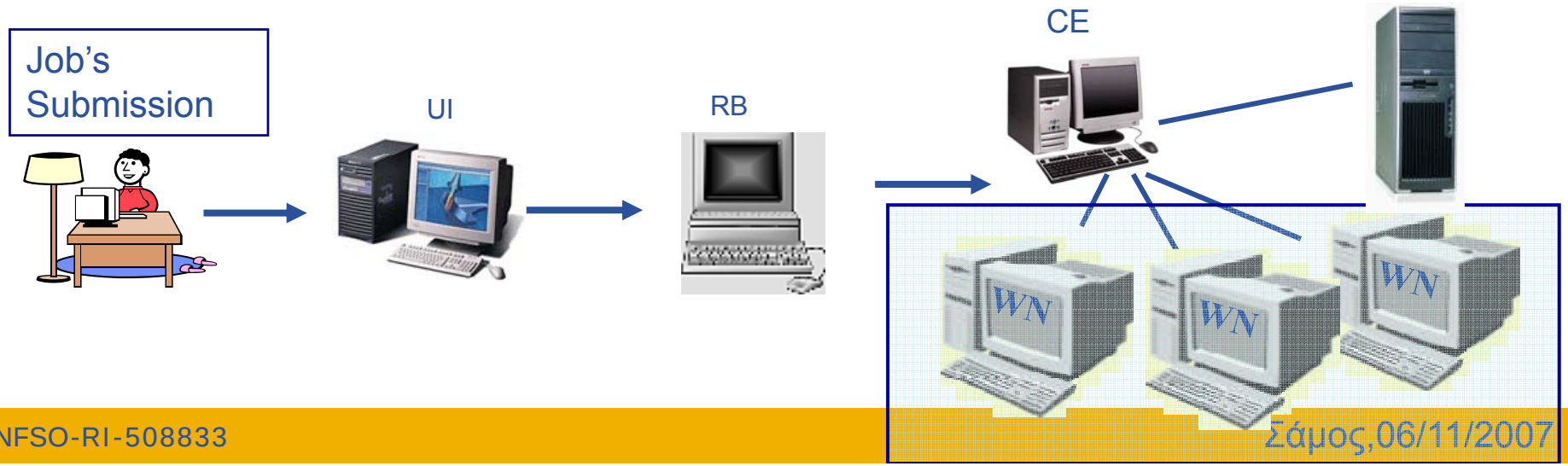
# Computing Element

- **"Grid interface"**
- **It is built on a farm of a computing nodes called Worker Nodes (WNs)**
- **Executes the basic queues functions**
- **In the Computing Element, a process is being executed that accepts jobs and dispatch them for execution to the Worker nodes (WNs)**
- **The state of an executing job is being watched by the Computing Element**



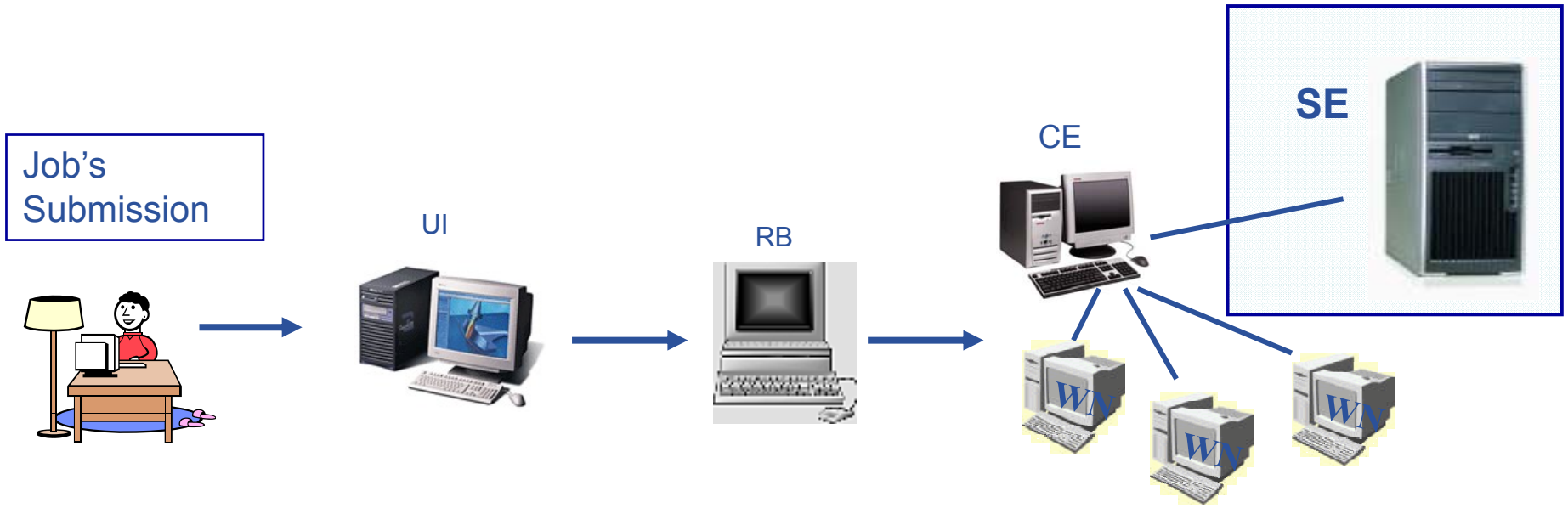
# Worker Node

- The submitted jobs are being executed in the Worker nodes
- Need only outbound connectivity
- Only basic services of middleware are required to be provided by the Worker nodes such as
  - Application libraries
  - Application Programming Interfaces (API)
  - Commands for performing actions on Grid resources and Grid data



# Storage Element

- **It provides uniform access to storage resources**
  **(it may control simple disk servers, large disk arrays or Mass Storage Systems (MSS)**
- **Each site may provide one or more SEs**



# Getting started

- **Obtaining a certificate**
- **Registering with LCG / EGEE**
- **Choosing a VO**
- **Accounts for the training events:**
  - ssh ui01.isabella.grnet.gr (Putty)
    - login as: **egee03 – egee50**

# Installing the certificate

- ✓ [egee01@ui01 egee01]$ **mkdir .globus**
  - Create directory .globus under the user home directory

- ✓ [eg
  - em
- ✓ [ege
  - em
- ✓ [eg
  - m

- **Prepare certificates for the training event only:**

- **[egee01@ui01 egee01]$ ./preparecerts.sh**

- **[egee01@ui01 egee01]$ ls -l ~/.globus**

```
total 12
-r--r--r--   1 egee01   training     5535 Sep 14 16:55 usercert.pem
-r--------   1 egee01   training      963 Sep 14 16:55 userkey.pem
```

- ✓ [egee01@ui01 egee01]$ **chmod 400 ~/.globus/userkey.pem**
  - The key must be readable only by the user

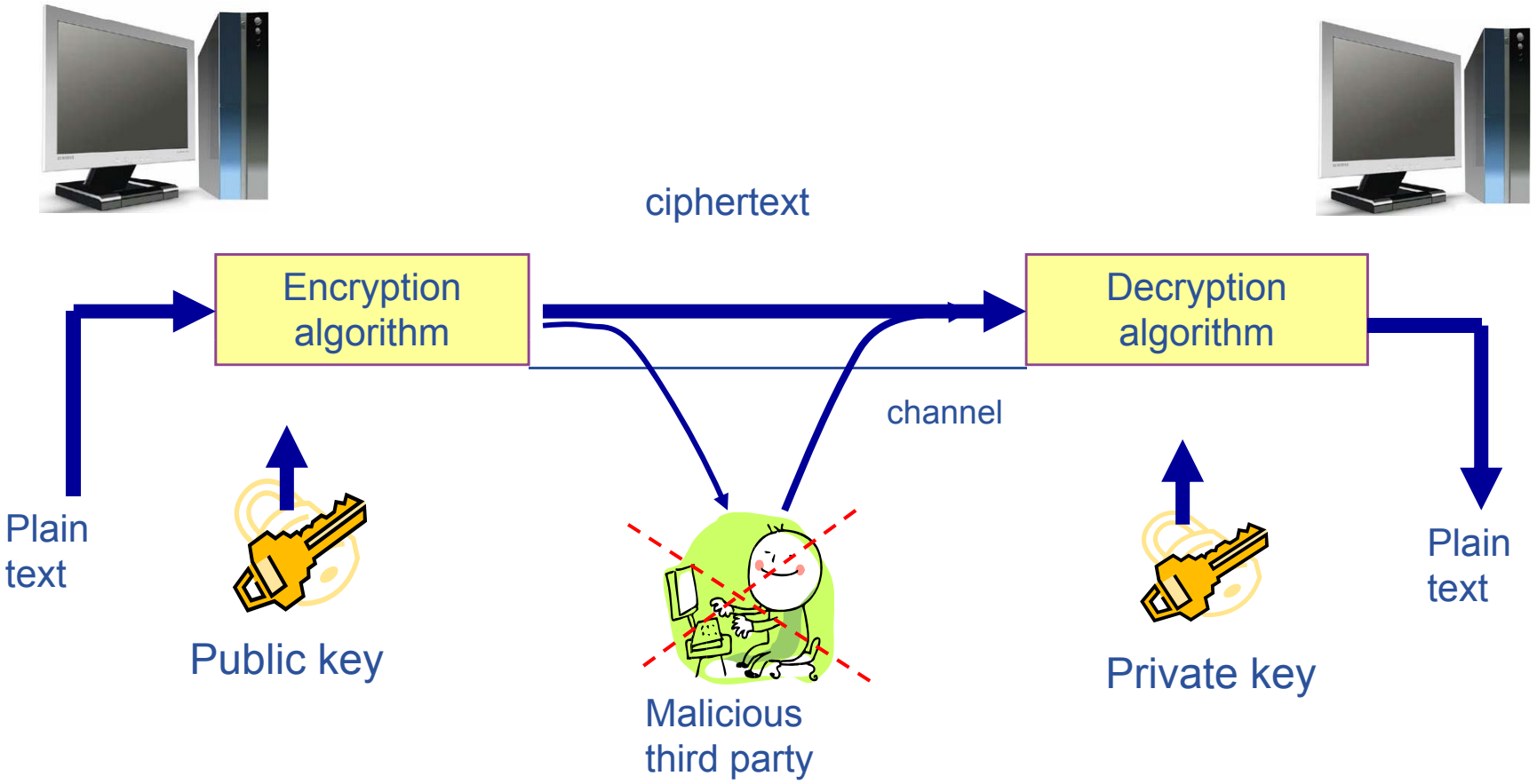
# Certificate Info

- **Retrieving information of the user certificate**

✓ **[egee01@ui01 egee01]$ grid-cert-info**

```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 1788 (0x6fc)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=GR, O=HellasGrid Demos, OU=Certification Authoritites, CN=HellasGrid
  Demo CA 2006
    Validity
        Not Before: Sep 14 11:25:01 2007 GMT
        Not After :  Sep 29 11:25:01 2007 GMT
    Subject: C=GR, O=HellasGrid Demos, OU=People, L=Thessaloniki, CN=User 1788
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
        RSA Public Key: (1024 bit)
            Modulus (1024 bit):
                00:c6:2e:31:bb:14:12:27:c3:a7:74:1a:48:3a:59:
```

# Cryptography components

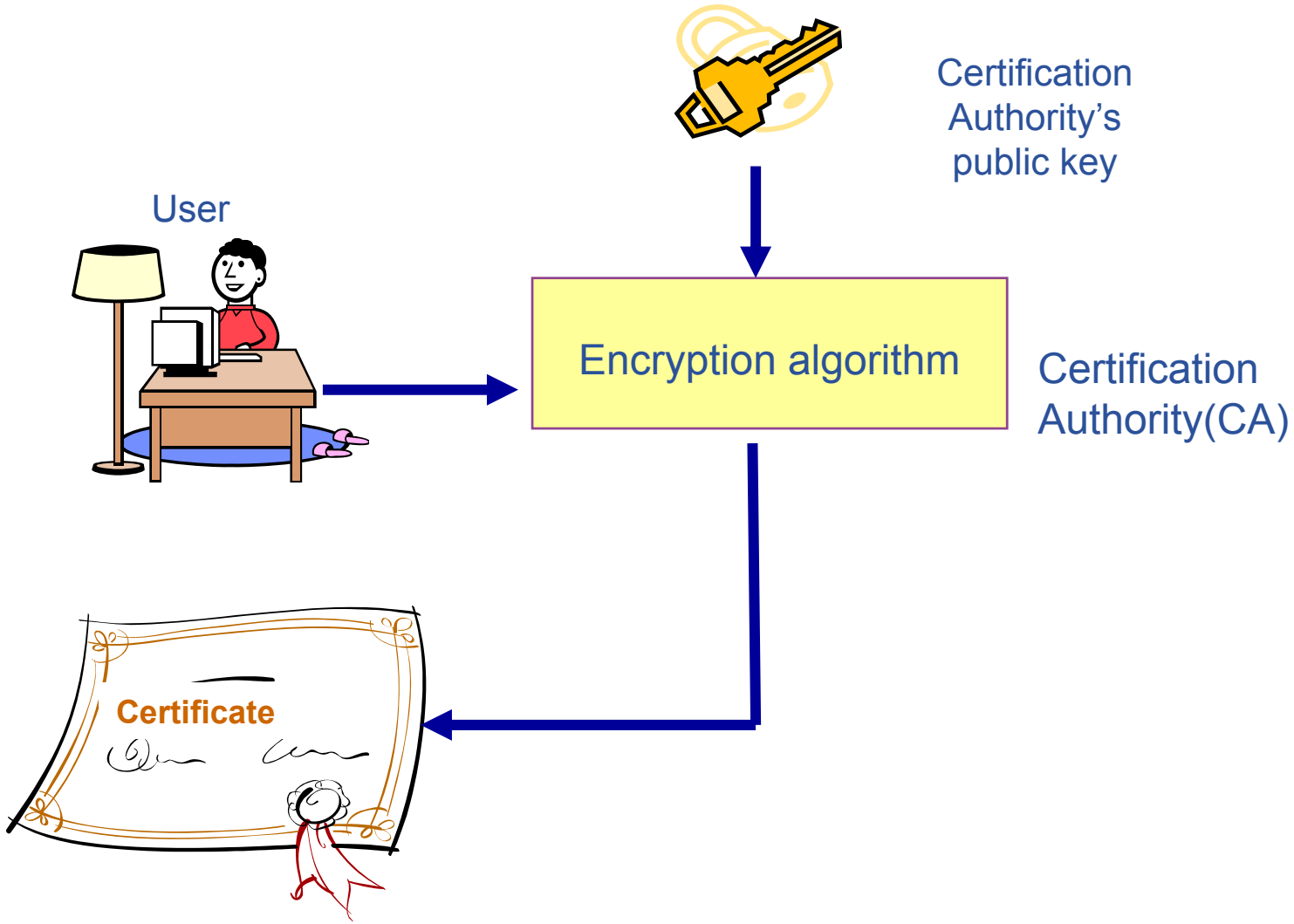


# Digital certificate X.509

- **Each entity (user, resource) must obtain a certificate**
- **The certificate includes information, such as the expiration date, the Certification Authority that signed it, the owner's public key and a DN**
- **The DN defines uniquely the owner and has the following fields:**

C = Owner's country
O = Owner's organization
OU = Owner's group
CN = Owner's name

# Certificate authority



# Proxy certificates

- A new temporal certificate created taking into account the issued certificate by the corresponding CA
  ⇨ a new key pair is created to be used during the period that the proxy is valid
- The new private key is not secured by a password
- The use of a proxy is recommended because:
  - ✓ the *proxy* has a short lifetime
  - ✓ uses a different private key from the issued certificate

# VOMS

- **Virtual Organisation Membership Service (VOMS)**
  - A system which allows a proxy to have extensions containing information:
    - About the VO
    - The groups the user belongs to in the VO
    - Any roles the user is entitled to have

- ***Group*: subset of the VO containing members who share some responsibilities or privileges in the project**
  - Hierarchically organised
  - A user can be a member of any number of groups
  - VOMS proxy contains the list of all groups the user belongs to
  - Group ⇨ privileges the user **ALWAYS** has

- ***Role*: Attribute which typically allows a user to acquire special privileges to perform specific tasks**
  - Role ⇨ privileges the user needs to have only from time to time

# Proxy certificate

## •Creating a proxy

✓ [egee01@ui01 egee01]$ **voms-proxy-init --voms=hgdemo**

Your identity: /C=GR/O=HellasGrid Demos/OU=People/L=Thessaloniki/CN=User 1978

Enter GRID pass phrase:

Creating temporary proxy ..... Done

Contacting voms.grid.auth.gr:15030

[/C=GR/O=HellasGrid/OU=auth.gr/CN=voms.grid.auth.gr] "hgdemo" Done

Creating proxy ..... Done

Your proxy is valid until Mon Sep 17 00:48:09 2007

## •Destroying a proxy

✓ [egee01@ui01 egee01]$ **voms-proxy-destroy**

# Information System (IS)

- **It provides information about the Grid resources and their status**
  **⇨ This information is essential for the operation of the whole Grid**
- **Location of availiable Computing Elements to run jobs**
- **Finding of SEs that holding replicas of Grid files and the catalogs keeping the information on these files**
- **The information is stored in databases**
- **The published information is used for**
  - ✓ monitoring purposes ⇨ for analyzing usage and performance of the Grid, detecting fault situations and any other interesting events
  - ✓ accounting purposes ⇨ for creating statistics of the applications run by the users in the resources

# MDS

- **Globus Moinitoring and Discovery service**
- **Resource Discovery and publishing the resource status**
- **OpenLDAP** which is an open source implementation of the ***Lightweight Directory Access Protocol (LDAP)***, a specialised database optimised for reading, browsing and searching information
- **Hierarchical architecture:**
  - In every resource runs a ***Grid Resource Information Server (GRIS)*** providing relevant information about the resource
  - At each site runs a ***Site Grid Information Server (GIIS)*** that collects information from the local GRISes and republishes it. The GIIS uses a ***Berkeley Database Information Index (BDII)*** to store data
  - A BDII is used to read from a group of sites, depicting a view of the overall Grid resources (on top of the hierarchy)

# A Command-line tool for querying MDS

- **lcg-infosites ⇨ obtain VO-specific information on existing Grid resources**

  **lcg-infosites --vo <vo> <option> -v <verbosity> -f <site> --is <bdii>**

  where:

  **--vo <vo>:** the name of the VO to which the information to print is related (mandatory)

  **<option>:** specifies what information has to be printed. It can take the following values:

  **ce:** the number of CPUs, running jobs, waiting jobs and CE names (global, no VO-specific information)

  **se:** the names of the SEs supporting the VO, the type of storage system and the used and available space;

  **-v 1: only the CE / SE names**

  **-v 2: the cluster names, the amount of RAM, the operating system name and version and the processor model**

  **all**: the information given by ce and se

  **closeSE**: the names of the CEs supporting the VO and their close SEs

  **tag**: the software tags published by each CE supporting the VO

  **lfc**: the hostname of the LFC catalogues available to the VO

  **lfcLocal**: the hostname of the local LFC catalogues available to the VO

  **rb**: the hostname and port of the RBs available to the VO

  **dli**: the Data Location Index servers available to the VO

  **dliLocal**: the local Data Location Index servers available to the VO

  **sitenames**: the names of all WLCG/EGEE sites;

# Examples (1)

- **Obtaining information about computing resources**
  - ✓ **[egee01@ui01 egee01]$ lcg-infosites --vo hgdemo ce**

```
valor del bdii: bdii.isabella.grnet.gr:2170
#CPU    Free    Total Jobs      Running Waiting ComputingElement
----------------------------------------------------------
 11      7       0               0       0       node001.grid.auth.gr:2119/jobmanager-pbs-hgdemo
 20      19      0               0       0       ce02.marie.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
 64      12      0               0       0       ce01.isabella.grnet.gr:2119/jobmanager-pbs-hgdemo
 118     53      0               0       0       ce01.marie.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
 118     53      0               0       0       glite-ce01.marie.hellasgrid.gr:2119/blah-pbs-hgdemo
 122     110     0               0       0       ce01.afroditi.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
[…]
```

- **Obtaining information about storage resources**
  - ✓ **[egee01@ui01 egee01]$ lcg-infosites --vo hgdemo se**

```
Avail Space(Kb) Used Space(Kb)  Type    SEs
----------------------------------------------------------
45766072        100254364       n.a     se.phy.bg.ac.yu
28651116        81503928        n.a     se001.grid.bas.bg
170660000       16280000        n.a     se02.marie.hellasgrid.gr
1780407808      3362175488      n.a     se01.isabella.grnet.gr
2730000000      200000000       n.a     se01.marie.hellasgrid.gr
2360000000      550000000       n.a     se01.afroditi.hellasgrid.gr
2500000000      190000000       n.a     se01.kallisto.hellasgrid.gr
2640000000      350000000       n.a     se01.ariagni.hellasgrid.gr
[…]
```

# Examples (2)

- **Listing the hostname of the LFC catalogues**
  - ✓ **[egee01@ui01 egee01]$ lcg-infosites --vo hgdemo lfc**

    **lfc.isabella.grnet.gr**

- **Listing the software tags published by each CE supporting the VO**
  - ✓ **[egee01@ui01 egee01]$ lcg-infosites --vo see tag**

    valor del bdii: bdii.isabella.grnet.gr:2170
    Name of the CE: g02.phy.bg.ac.yu
    VO-seegrid-vive-0.4.2
    VO-seegrid-vive-0.4.3
    […]

- **Listing all WLCG/EGEE sitenames**
  - ✓ **[egee01@ui01 egee01]$ lcg-infosites --vo hgdemo sitenames**

    […]
    HG-01-GRNET
    HG-02-IASA
    HG-03-AUTH
    HG-04-CTI-CEID
    HG-05-FORTH
    HG-06-EKT
    […]

# Job Description Language

- A high-level language based on the ***Classified Advertisement (ClassAd) language***
- JDL describes jobs and aggregates of jobs with arbitrary dependency relations
- JDL specifies the desired job characteristics and constraints, which are taken into account by the WMS to select the best resource to execute the job
- A JDL file consists of lines having the format:

  *attribute = expression;*

  - Expressions can span several lines, but only the last one must be terminated by a semicolon
  - Literals are enclosed in double quotes
  - " in strings must be escaped with a backslash ("\"Hallo")
  - The character " ' " cannot be used in the JDL
  - Comments of each line begin with # or //
  - Multi-line comments must be enclosed between "/*" and "*/"
  - **No blank characters or tabs should follow the semicolon at the end of a line**

# Attributes (1)

| | |
|---|---|
| **Executable** | ✓ **The value of this attribute is the executable filename or the command to be run by the job**<br>✓ If the command is already present on the WN, it must be expressed as a absolute path |
| **StdOutput** | ✓ The name of the files containing the standard output |
| **StdError** | ✓ The name of the files containing the standard error |
| **StdInput** | ✓ The names of the files used as Input files |
| **OutputSandbox** | ✓ The files to be transferred back to the UI after the job is finished |
| **Environment** | ✓ Modifies the shell environment of the job |
| **Virtual Organisation** | ✓ Explicitly specify the VO of the user |
| **Requirements** | ✓ Expresses constraints on the resources where the job should run<br>✓ Its value is a Boolean expression that must evaluate to true for a job to run on that specific CE<br>**(example: Requirements = other.GlueCEInfoLRMSType == "PBS" && other.GlueCEInfoTotalCPUs > 1;)** |

# Attributes (2)

| | |
|---|---|
| RetryCount<br>MaxRetryCount | ✓ Times that the WMS automatically resubmitts jobs which failed for some reason<br>(deep resubmission ⇨ when the job failed after started running in a WN) |
| ShallowRetryCount<br>MaxShalowRetryCount | ✓ Times that the WMS automatically resubmitts jobs which failed for some reason<br>(shallow resubmission – gLite) |
| MyProxyServer | ✓ The Proxy server to be used for certificate renewal |
| Rank | ✓ The CE with the highest rank is selected by the WMS to execute a job<br>✓ by default *Rank* = *other.GlueCEStateEstimatedResponseTime*<br>(but other.GlueCEStateFreeCPUs<br>other.GlueCEStateWaitingJobs) |

# A simple job

✓ [egee01@ui01 egee01]$ less testJob1.sh

```
#!/bin/bash
echo "****** Running...  date ****** "
date
echo "****** Running... hostname ******"
hostname
echo "****** Running... pwd ****** "
pwd
echo "****** Running... ls ****** "
ls -l
echo "****** Running... uptime ****** "
uptime
echo "****** Learn your process ****** "
ps aux | grep home
```

```
echo "****** Downloading... A file to
    WN****** "
wget http://www. …
echo "****** Running... ls ****** "
ls -l
echo "****** Printing Input files ****** "
echo "First file:"
cat $1 > >merge.out
echo "Second file:"
cat $2 >> merge.out
```

# A simple JDL file

- [egee01@ui01 egee01]$ **less testJob1.jdl**

```
Executable = "testJob.sh";
Arguments = "fileA fileB";
StdOutput = "std.out";
StdError = "std.err";
InputSandbox = {“./testJob.sh", “./fileA", “./fileB"};
OutputSandbox = {"std.out", "std.err",  merge.out" ,"gLite-3-UserGuide.pdf"};
```

# Job execution

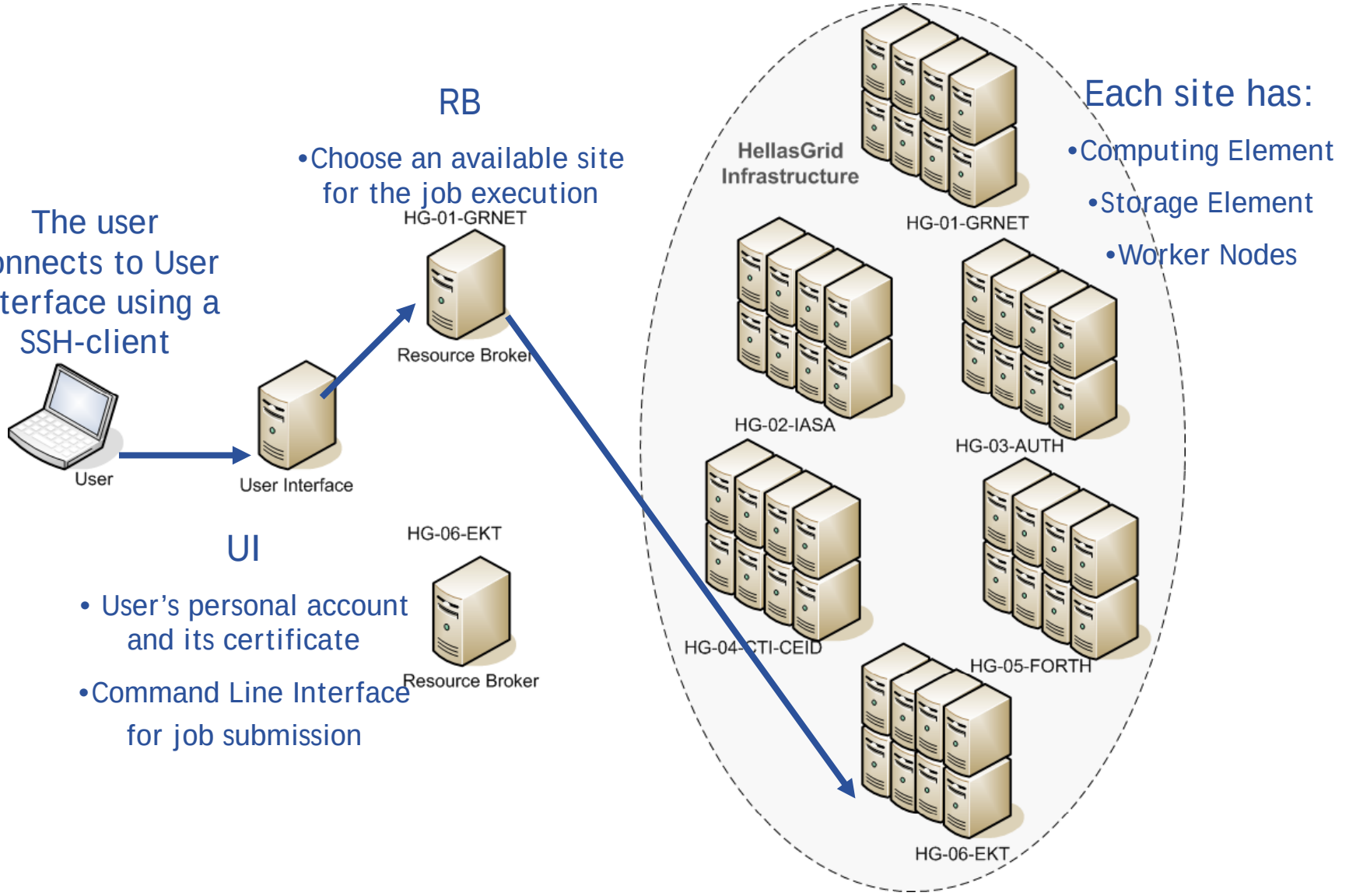
The user connects to User Interface using a SSH-client

UI

- User's personal account and its certificate
- Command Line Interface for job submission

RB

- Choose an available site for the job execution




Each site has:

- Computing Element
- Storage Element
- Worker Nodes

# Job matching to CEs

- **Listing computing elements that match a job description**
  - ✓ [egee01@ui01 egee01]$ **glite-wms-job-list-match -a testJob1.jdl**

```
Connecting to the service https://wms01.egee-see.org:7443/glite_wms_wmproxy_server

================================================================================

                    COMPUTING ELEMENT IDs LIST
 The following CE(s) matching your job requirements have been found:

         *CEId*
 - ce01.afroditi.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
 - ce01.ariagni.hellasgrid.gr:2119/jobmanager-lcgpbs-hgdemo
 - ce01.athena.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
 - ce01.isabella.grnet.gr:2119/jobmanager-pbs-hgdemo
 - ce01.kallisto.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
 - ce01.marie.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
 - ce02.athena.hellasgrid.gr:2119/blah-pbs-hgdemo
 - ce02.marie.hellasgrid.gr:2119/jobmanager-pbs-hgdemo
 - glite-ce01.marie.hellasgrid.gr:2119/blah-pbs-hgdemo
 - node001.grid.auth.gr:2119/jobmanager-pbs-hgdemo

================================================================================
```

# Job submission

- **Single Job submission**
  - ✓ **[egee01@ui01 egee01]$ glite-wms-job-submit -o jobId -a testJob1.jdl**

- **Connecting to the service https://wms01.egee-see.org:7443/glite_wms_wmproxy_server**

**========== glite-wms-job-submit Success ==========**

**The job has been successfully submitted to the WMProxy**
**Your job identifier is:**

**https://wms01.egee-see.org:9000/6INrYSPP4XfkgTYHuqHuww**

**The job identifier has been saved in the following file:**
**/home/training/egee01/jobId**

**==========**

✓ **glite-wms-job-submit –o jobId –r ce01.isabella.grnet.gr:2119/jobmanager-pbs-hgdemo -a testJob.jdl**
  - -r : sends the job directly to the specified CE

# Job status' retrieval

- **Retrieving the status of a job**
  - ✓ [egee01@ui01 egee01]$ **glite-wms-job-status -i jobId**

**********

**BOOKKEEPING INFORMATION:**

**Status info for the Job : https://wms01.egee-see.org:9000/u-Slc372Ny_DQ04reimrHw**
**Current Status: Scheduled**
**Status Reason: Job successfully submitted to Globus**
**Destination: ce01.ariagni.hellasgrid.gr:2119/jobmanager-lcgpbs-hgdemo**
**Submitted: Tue Sep 18 03:16:47 2007 EEST**

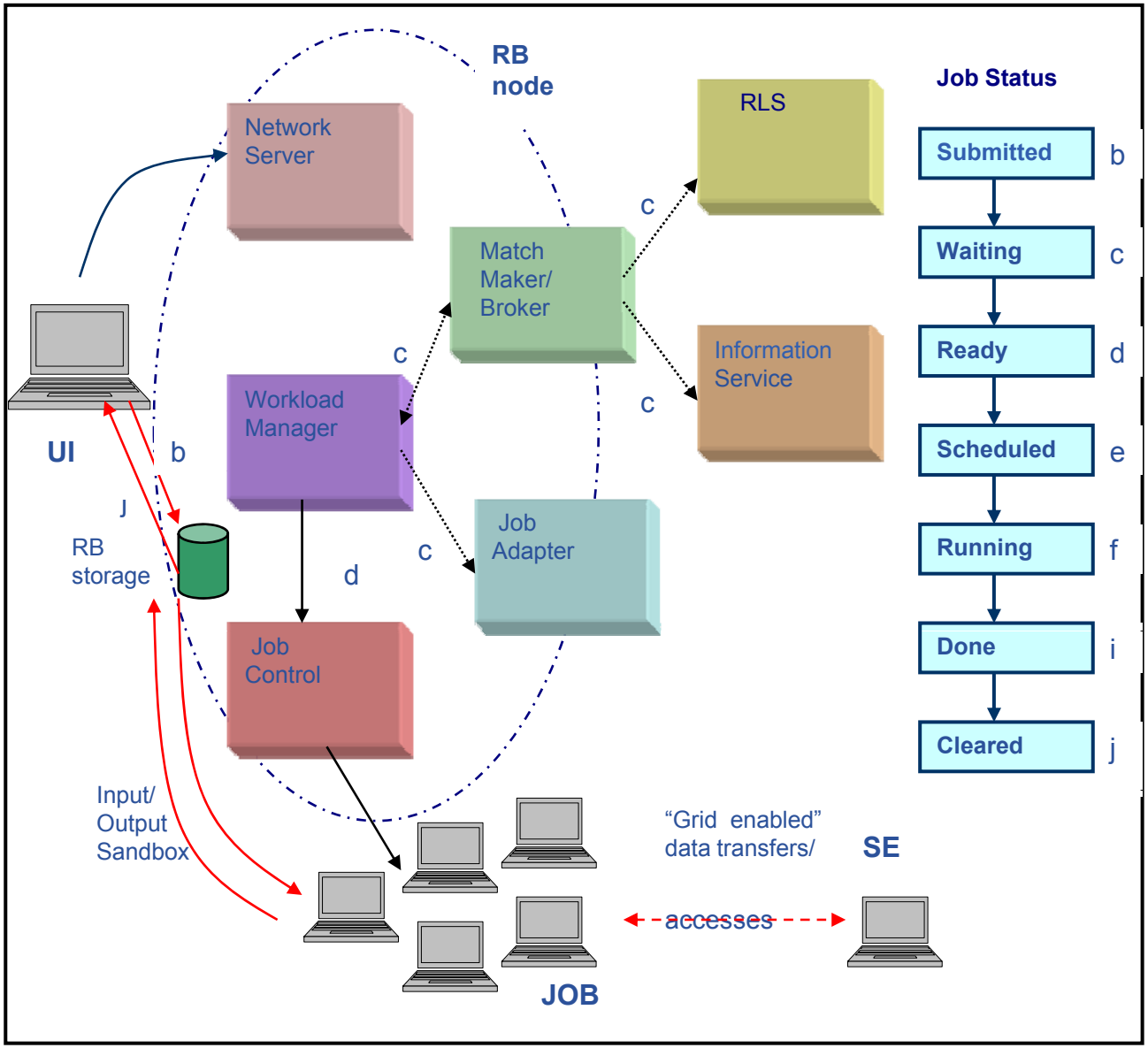
**********

:

  - ✓ [egee01@ui01 egee01]$ **watch "glite-job-status –i jobId"**

**(To exit ctrl + C)**

# Job flow (1)


RB node
Network Server
Match Maker/ Broker
Workload Manager
Job Adapter
Job Control
RLS
Information Service
UI
RB storage
Input/ Output Sandbox
JOB
"Grid enabled" data transfers/
SE
accesses
Job Status
Submitted
Waiting
Ready
Scheduled
Running
Done
Cleared
b
c
d
e
f
i
j

# Job flow (2)

- **Job submission**
  - The user logs in the UI and submits the job to a Resource broker.
  - If one or more files need to be copied from the UI to the WN, this is specified in the job description and the files are initially copied to the RB. This set of files is called the **Input Sandbox**
  - **Job status ⇨ SUBMITTED**
- **Finding the proper CE**
  - The **WMS** interrogates the **Information Supermarket (ISM)** (an internal cache of information read from the BDII) , to determine the status of computational and storage resources.
  - The WMS interogates the File Catalogue to find the location of any required input files
  - **Job status ⇨ WAITING**
- **Job submission from the RB to the selected CE**
  - The RB prepares a wrapper script that will be passed together with other parameters, to the selected CE.
  - **Job status ⇨ READY**
- **Job arrival to the CE**
  - **The CE receives the request and sends the job for execution to the local LRMS.**
  - **Job status ⇨SCHEDULED**

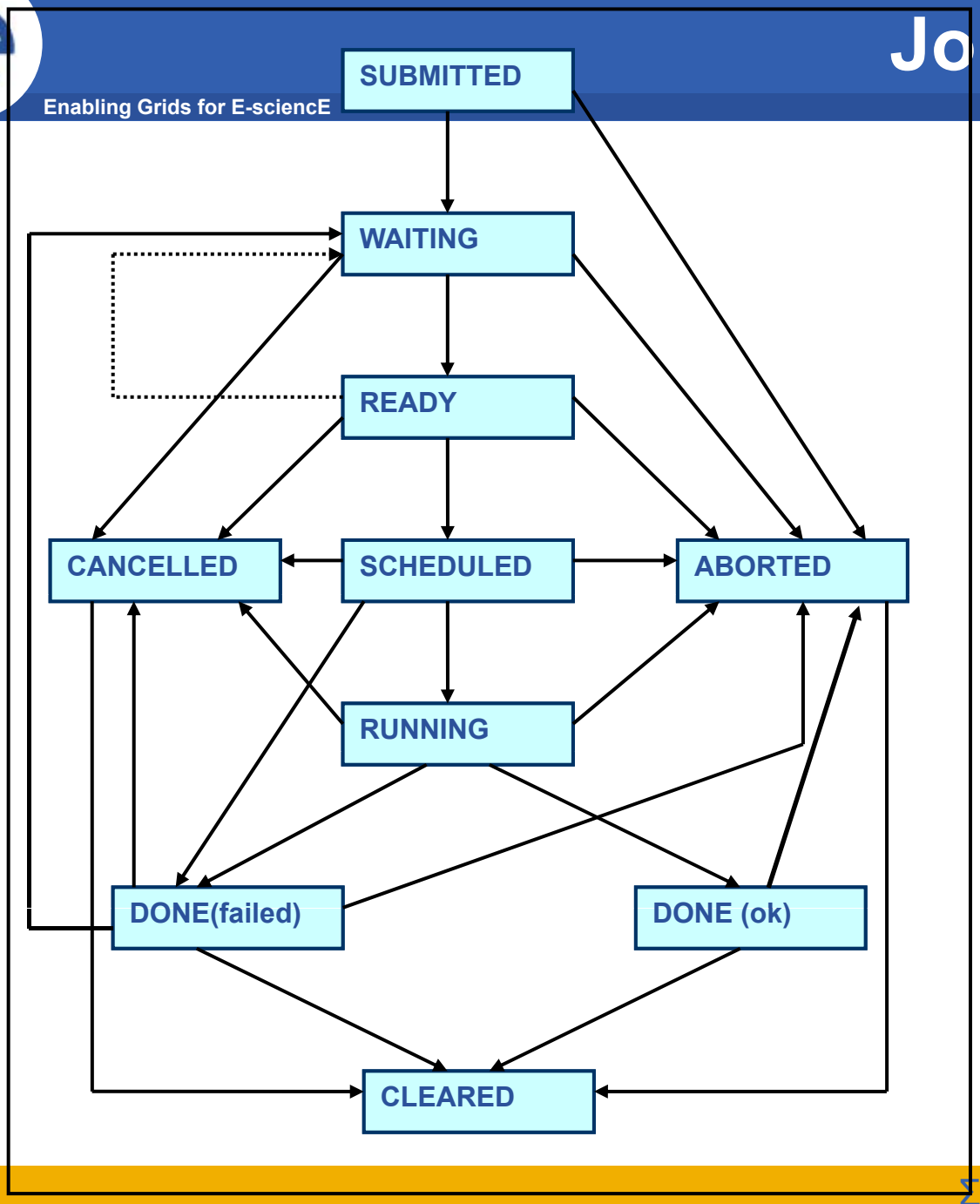
# Job flow (3)

- **Job submission to the Worker node**
  - The LRMS handles the execution of jobs on the local Worker Nodes.
  - The Input Sandbox files are copied from the RB to an available WN where the job is executed.
  - While the job runs, Grid files can be directly accessed from an SE using either the RFIO or gsidcap protocol
  - Any new produced output files which can be uploaded to the Grid and made available for other Grid users to use. This can be achieved using the Data Management tools described later. Uploading a file to the Grid means copying it to a Storage Element and registering it in a file catalogue.
  - **Job status ⇨ RUNNING**

- **Job finished**
  - If the job ends without errors, the small output files specified by the user in the ***Output Sandbox*** are transferred back to the RB node.
  - **Job status ⇨ DONE**

- **Output retrieval**
  - The user can retrieve the output files to the UI
  - **Job status ⇨ Cleared**

# Job status


SUBMITTED
WAITING
READY
CANCELLED
SCHEDULED
ABORTED
RUNNING
DONE(failed)
DONE (ok)
CLEARED

# Job cancelling

- ## Cancelling a job

  ✓ **[egee01@ui01 egee01]$ glite-wms-job-cancel -i jobId**

  Are you sure you want to remove specified job(s) [y/n]y : y

  Connecting to the service
  https://195.251.53.233:7443/glite_wms_wmproxy_server

  ================ glite-wms-job-cancel Success ==============

  The cancellation request has been successfully submitted for the following job(s):

  - https://wms01.egee-see.org:9000/E_Ykk3oGFkXTQcDi_XZs5w

  ============================================================

# Output retrieval

- If the job's status is DONE, then its output can be copied to the UI with the commands:
  - ✓ [egee01@ui01 egee01]$ **glite-wms-job-output -i jobId**

Connecting to the service
https://195.251.53.233:7443/glite_wms_wmproxy_server

==========

JOB GET OUTPUT OUTCOME

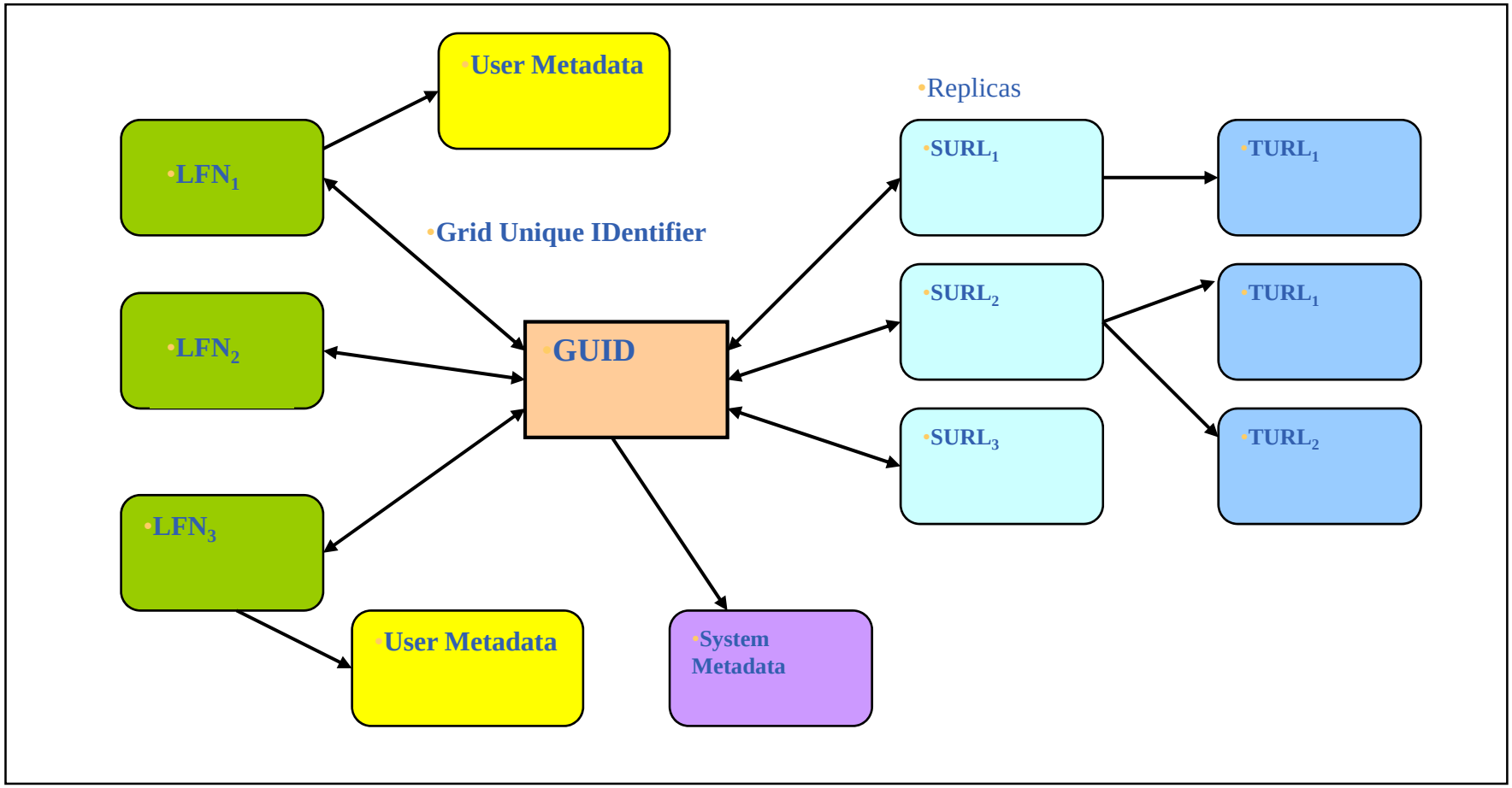
Output sandbox files for the job:
https://wms01.egee-see.org:9000/6INrYSPP4XfkgTYHuqHuww
have been successfully retrieved and stored in the directory:
/tmp/glite/glite-ui/egee01_6INrYSPP4XfkgTYHuqHuww

==========

# Filenames(1)

- **Grid Unique Identifier (GUID)**
  - Identifies a file uniquely
  - *Example*: guid:ab993b98-8bc9-4984-901e-91290276090c
- **Logical File Name (LFN) (User Alias)**
  - Refers to a file instead of a GUID
  - lfn:<any_string>
  - LFC catalogue: lfn:/grid/<MyVO>/<MyDirs>/<MyFile>
  - *Example*: lfn:/grid/hgdemo/test_egee01/test_file
- **Storage URL (SURL) (Physical File Name-PFN)**
  - Identifies a replica in a SE
  - <sfn|srm>://<SE_hostname>/<some_string>
  - *Example*: sfn://se01.isabella.grnet.gr/storage/hgdemo/generated/2007-04-20/filec4087974-dbaa-4890-91e2-3c105fa0a3df
- **Transport URL (TURL)**
  - A valid URI with the necessary information to access a file in a SE
  - <protocol>://<some_string>
  - *Example*:gsiftp://se01.isabella.grnet.gr/storage/hgdemo/generated/2007-04-20/file1a08d327-d7dc-4d89-bb01-2c86f59eae37

# Filenames(2)


User Metadata
LFN1
Grid Unique IDentifier
LFN2
GUID
LFN3
User Metadata
System Metadata
Replicas
SURL1
TURL1
SURL2
TURL1
SURL3
TURL2

# File Catalogue

- **File Catalogue (gLite) or LCG File Catalogue**
  - Maintains mappings between LFNs, GUID, SURLs
  - Local File Catalogue, holding only replicas stored at a given site
  - Global File Catalogue, containing information about all files in the Grid
  - Consists of a unique catalogue, where the LFN is the main key
  - System metadata are supported

- **Grid file**
  - Physically present in a SE
  - Registered in the file catalogue

- **High Level Tools (lcg_util) ⇨ Consistency between files in the SEs and entries in the file catalogue**

- **Low level Data Management tools ⇨ Inconsistency between SEs physical files and catalogue entries**

# LFC commands

- lfc-ls: List file / directory entries in a directory
- lfc-mkdir: Create directory
- lfc-chmod: Change access mode of a LFC file / directory
- lfc-chown: Change owner and group of a LFC file / directory
- lfc-ln: Make a symbolic link to a file / directory
- lfc-rename: Rename a file / directory
- lfc-rm: Remove a file / directory
- lfc-setcomment: Add / replace a comment
- lfc-delcomment: Delete the comment associated with a file / directory

# More advanced LFC commands

- lfc-getacl: Get file / directory access control lists
- lfc-setacl: Set file / directory access control lists
- lfc-entergrpmap: Defines a new group entry in the Virtual ID table
- lfc-enterusrmap: Defines a new user entry in Virtual ID table
- lfc-modifygrpmap: Modifies a group entry corresponding to a given virtual gid
- lfc-modifyusrmap: Modifies a user entry corresponding to a given virtual uid
- lfc-rmgrpmap: Suppresses group entry corresponding to a given virtual gid or group name
- lfc-rmusrmap: Suppresses user entry corresponding to a given virtual uid or user name.

# LCG Data Management Client Tools

- ***lcg_util* tools**
  - Allow users to copy files between UI, CE, WN and a SE
  - Allow users to register entries in the file catalogue and replicate files between SEs

- **Replica Management**
- **lcg-cp:** Copies a Grid file to a local destination (*download*)

- **lcg-cr**: Copies a file to a SE and registers the file in the catalogue (*upload*)

- **lcg-del:** Deletes one file (either one replica or all replicas)

- **lcg-rep:** Copies a file from one SE to another SE and registers it in the catalogue (replicate)

- **lcg-gt:** Gets the TURL for a given SURL and transfer protocol

# LCG Data Management Client Tools

- ## File Catalogue Interaction

**lcg-aa:** Adds an alias in the catalogue for a given GUID

**lcg-ra:** Removes an alias in the catalogue for a given GUID

**lcg-rf:** Registers in the catalogue a file residing on an SE

**lcg-uf:** Unregisters in the the catalogue a file residing on an SE

**lcg-la:** Lists the aliases for a given LFN, GUID or SURL

**lcg-lg:** Gets the GUID for a given LFN or SURL

**lcg-lr:** Lists the replicas for a given LFN, GUID or SURL

# Configuration to use the LFC

- The LFC_HOST variable must contain the hostname of the machine providing the LFC service

✓ [egee01@ui01 egee01]$ export LFC_HOST=`lcg-infosites --vo hgdemo lfc`

✓ [egee01@ui01 egee01]$ export LCG_CATALOG_TYPE=lfc

- The LCG_GFAL_VO variable must contain the name of the user's VO

✓ [egee01@ui01 egee01]$ export LCG_GFAL_VO=hgdemo

# Create a directory

- **Creating a directory in the LFN namespace**
  - ✓ [egee01@ui01 egee01]$ lfc-mkdir /grid/hgdemo/test_($USER)

  where $USER -> the username of each participant

- **Listing the entries of a LFC directory**
  - ✓ [egee01@ui01 egee01]$ lfc-ls -l /grid/hgdemo/

    drwxrwxr-x 0 26158 32000 0 Apr 20 12:45 test_egee01

- Defining LFC_HOME variable to point to the created directory
- ✓ [egee01@ui01 egee01]$ export LFC_HOME=/grid/hgdemo/test_($USER)

- **Removing LFNs from the LFC**
  - ✓ [egee01@ui01 egee01]$ lfc-rm -r lfc:/grid/hgdemo/test_($USER)

# Upload and replicate a file

- **Uploading a file to the grid with specific LFN and in a specific storage element**

✓ [egee01@ui01 egee01]$ **lcg-cr file:$PWD/image1.jpg -l lfn:image1 -d se01.isabella.grnet.gr**

guid:e20d1fa9-e35f-426d-aa72-cb99b18c2791

✓ [egee01@ui01 egee01]$ **lfc-ls /grid/hgdemo/test_$USER/**

image1

- **Replicating a file**

✓ **[egee01@ui01 egee01]$ lcg-rep –v lfn:image1 -d se01.kallisto.hellasgrid.gr**

Using grid catalog type: lfc
Using grid catalog : lfc.isabella.grnet.gr
Source URL: lfn:/grid/hgdemo/egee01_test/image1
File size: 33985
VO name: hgdemo
Destination specified: se01.kallisto.hellasgrid.gr
Source URL for copy: gsiftp://se01.isabella.grnet.gr/storage/hgdemo/generated/2007-09-18/filee8e4acdf-dd25-49e4-8ced-3d46aba13000
Destination URL for copy: gsiftp://se01.kallisto.hellasgrid.gr/se01.kallisto.hellasgrid.gr:/data02/hgdemo/2007-09-18/file5625a538-ecf7-4e87-b22b-8ef861e4b30d.172458.0

- **Copying a file out of the Grid**

✓ [egee01@ui01 egee01]$ **lcg-cp -t 100 lfn:image1 file:$PWD/copy_image1**

# File management operations (1)

- **Listing replicas**

✓ **[egee01@ui01 egee01]$ lcg-lr --vo hgdemo lfn:image1**

sfn://se01.isabella.grnet.gr/storage/hgdemo/generated/2007-09-18/filee8e4acdf-dd25-49e4-8ced-3d46aba13000

srm://se01.kallisto.hellasgrid.gr/dpm/kallisto.hellasgrid.gr/home/hgdemo/generated/2007-09-18/file5625a538-ecf7-4e87-b22b-8ef861e4b30d

- **Listing guids given the lfn**

✓ **[egee01@ui01 egee01]$ lcg-lg lfn:image1**

guid:e20d1fa9-e35f-426d-aa72-cb99b18c2791

- **Adding metadata information to LFC entries**

✓ [egee01@ui01 egee01]$ **lfc-setcomment /grid/hgdemo/test_($USER) /image1 "Created for the training"**

- **View metadata of a specific file**

✓ **[egee01@ui01 egee01]$ lfc-ls --comment /grid/hgdemo/test_($USER)/image1**

image1 Created for the training

- **Removing metadata information to LFC entries**

✓ **[egee01@ui01 egee01]$ lfc-delcomment /grid/hgdemo/test_$USER/image1**

# A simple job with data management

✓ [egee01@ui01 egee01]$ less testJob2.sh

```
#!/bin/sh
echo Running at `hostname`
echo Date `date`
echo Fetching application
echo Enabling the executable flag for the application
chmod 755 $PWD/compressjpeg
echo Fetching data
lcg-cp --vo hgdemo $1 file:$PWD/local_copy
echo Compressing
$PWD/compressjpeg local_copy > local_compressed_copy
echo Convert to jpeg
pnmtojpeg local_copy > local_copy_jpeg
echo Convert to greyscale
jpegtran -grayscale local_copy_jpeg >grey_local_copy_jpeg
ls -al
echo Registering output
lcg-cr --vo hgdemo -l $2 file:$PWD/local_compressed_copy
lcg-cr --vo hgdemo -l $3 file:$PWD/grey_local_copy_jpeg
echo Done!
```

# A simple JDL file

- [egee01@ui01 egee01]$ less testJob2.jdl

```
[
        Type = "job";
        JobType = "normal";
        RetryCount = 0;
        ShallowRetryCount = 3;
        Executable = "testJob2.sh";
        Arguments = "lfn:/grid/hgdemo/test_(username)/satimg lfn:/grid/hgdemo/te
st_(username)/satimgjpg lfn:/grid/hgdemo/test_(username)/greysatimg";
        InputSandbox =  {"file:///home/training/(username)/testJob2.sh", "file:/
//home/training/(username)/compressjpeg"};
        StdOutput = "std.out";
        StdError = "std.err";
        OutputSandbox = {"std.out", "std.err", "local_copy_jpeg"};
        DataRequirements = {
                [InputData = {"lfn:/grid/hgdemo/test_(username)/satimg"};

                 DataCatalogType = "DLI";]
        };
        DataAccessProtocol = {"gsiftp", "https"};
 ]
```

# Upload a file again!

- **Uploading a file to the grid with specific LFN and in a specific storage element**

✓ **[egee01@ui01 egee01]$ lcg-cr file:$PWD/satimg.ppm -l lfn:satimg guid:594cf6b5-e3b7-49b5-a916-0d5e3054af17**

✓ **[egee01@ui01 egee01]$ lfc-ls /grid/hgdemo/test_($USER)/**

image1

satimg

# Job execution and data retrieval (1)

- **Submit job**

  glite-wms-job-submit -o jobId2 –a testJob2.jdl

- **Watch the job status**

  watch "glite-wms-job-status -i  jobId2"

- **Retrieve the job output**

  glite-wms-job-output -i jobId2

- **Listing the entries of a LFC directory**

  [egee01@ui01 egee01]$ lfc-ls -l /grid/hgdemo/test_($USER)

```
-rw-rw-r--   1 26259    32000         1438745 Nov 04 22:36 greysatimg
-rw-rw-r--   1 26259    32000           33985 Nov 04 21:08 image1
-rw-rw-r--   1 26259    32000        14110553 Nov 04 22:32 satimg
-rw-rw-r--   1 26259    32000         1521142 Nov 04 22:36 satimgjpg
```

# Job execution and data retrieval (2)

- **Listing the entries of a LFC directory**

```
[egee01@ui01 egee01]$ lcg-lr --vo hgdemo lfn:satimgjpg
[egee01@ui01 egee01]$ lcg-lr --vo hgdemo lfn:greysatimg
```

- Download output files

```
[egee01@ui01 egee01]$ lcg-cp -t 100 lfn:satimgjpg file:$PWD/local_satimgjpg
[egee01@ui01 egee01]$ lcg-cp -t 100 lfn:greysatimg file:$PWD/local_greysatimg
```

# JDL file for parametric job

```
[
        Type = "job";
        JobType = "Parametric";
        Parameters = N;
        ParameterStart = 1;
        ParameterStep =1;
       RetryCount = 0;
        ShallowRetryCount = 3;
        Executable = "testJob2.sh";
        Arguments = "lfn:/grid/hgdemo/test_(username)/satimg_PARAM_ lfn:/grid/hg
    demo/test_(username)/satimgjpg_PARAM_ lfn:/grid/hgdemo/test_(username)/greysatim
    g_PARAM_";
        InputSandbox =  {"file:///home/training/(username)/testJob2.sh", "file:/
    //home/training/(username)/compressjpeg"};
        StdOutput = "std.out";
        StdError = "std.err";
        OutputSandbox = {"std.out", "std.err", "local_copy_jpeg"};
          DataRequirements = {
         [InputData = {"lfn:/grid/hgdemo/test_(username)/satimg_PARAM_",
    "lfn:/grid/hgdemo/test_(username)/compressjpeg"};
                DataCatalogType = "DLI";]
        };
  DataAccessProtocol = {"gsiftp", "https"};
    ]
```

# Upload files again!

- **Uploading file to the grid with specific LFN and in a specific storage element**

✓ **[egee01@ui01 egee01]$ lcg-cr file:$PWD/satimg1.ppm -l lfn:satimg1 -d se01.isabella.grnet.gr**

**guid:594cf6b5-e3b7-49b5-a916-0d5e3054af17**

**…**

☞ **All three subimages must be uploaded**

✓ **[egee01@ui01 egee01]$ lfc-ls /grid/hgdemo/test_ ($USER) /**

image1
satimg
satimg1
satimg2
satimg3

# Job execution

- **Submit job**

  glite-wms-job-submit -a -o parametric-ids.txt parametrictestJob.jdl

- **Watch the job status**

  watch "glite-wms-job-status -i parametric-ids.txt"

- **Retrieve the job output**

  glite-wms-job-output -i parametric-ids.txt

# Delete Grid files and LFC entries

- **Deleting all replicas with the specific lfn**
  - ✓ **[egee01@ui01 egee01]$ lcg-del –a lfn:image1**
  - ✓ **[egee01@ui01 egee01]$ lcg-lr --vo hgdemo lfn:image1**
    lcg_lr: No such file or directory

- **Delete LFC directory**
  - ✓ **[egee01@ui01 egee01]$ lfc-ls /grid/hgdemo/test_($USER)**
  - ✓ **[egee01@ui01 egee01]$ lfc-rm -r /grid/hgdemo/test _($USER)**
  - ✓ **[egee01@ui01 egee01]$ lfc-ls /grid/hgdemo/**

# Q&A


**Thank you !**