Enabling Grids for E-science in Europe

Information Society Technologies

General components of Grid middleware and User Interfaces

Roberto Barbera

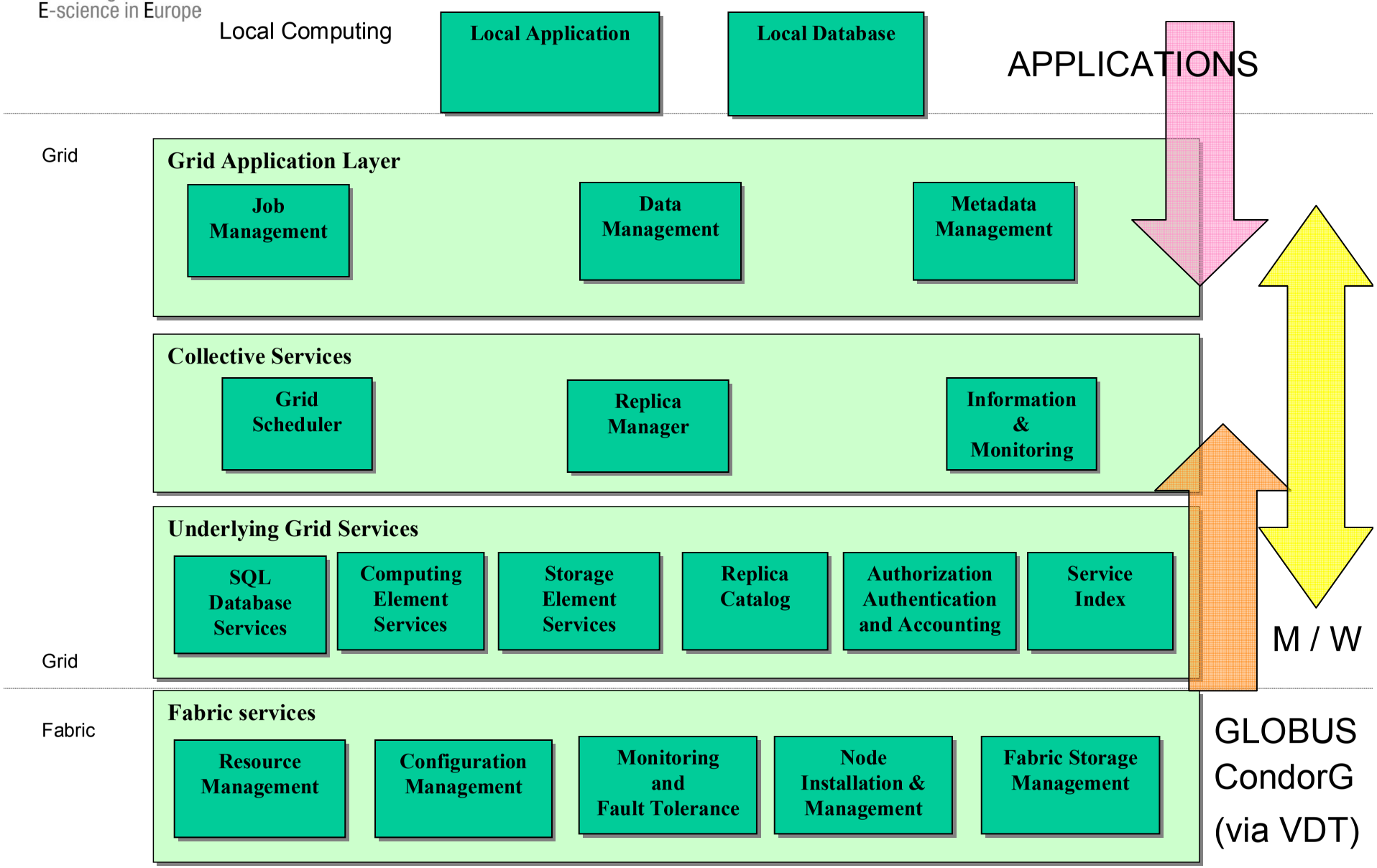
EGEE is funded by the European Union under contract IST-2003-508833

Enabling Grids for E-science in Europe

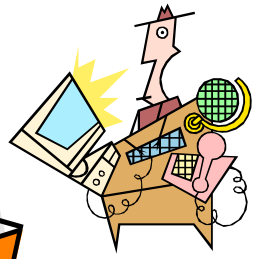
Overview

- The multi-layer model of middleware services
- The Command Line user interface
- The GENIUS Portal
- Summary and conclusions

Middleware architecture



Middleware interfaces

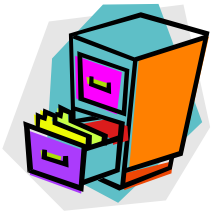


System Managers

Application Developers

Scientists

Certificate Authorities

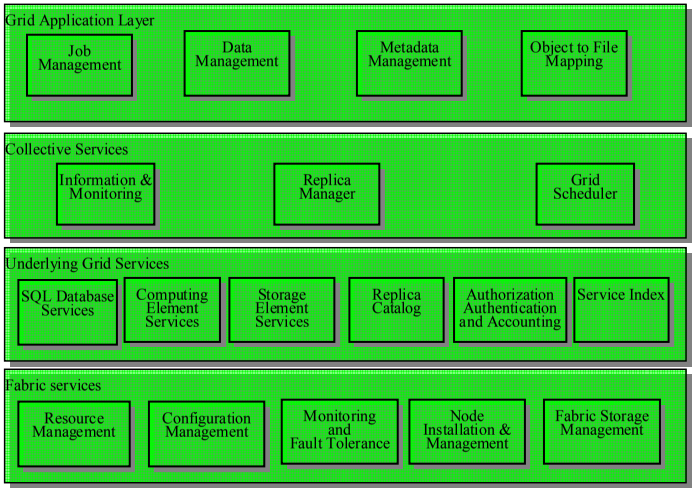


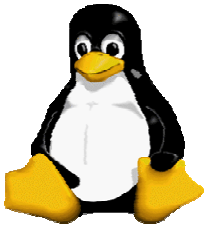
File Systems

User Accounts

Local Application

Local Database

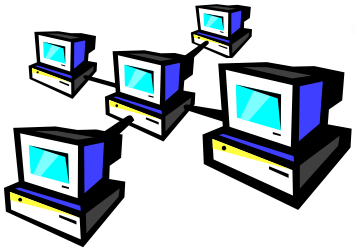




Operating Systems

Mass Storage Systems
HPSS, Castor

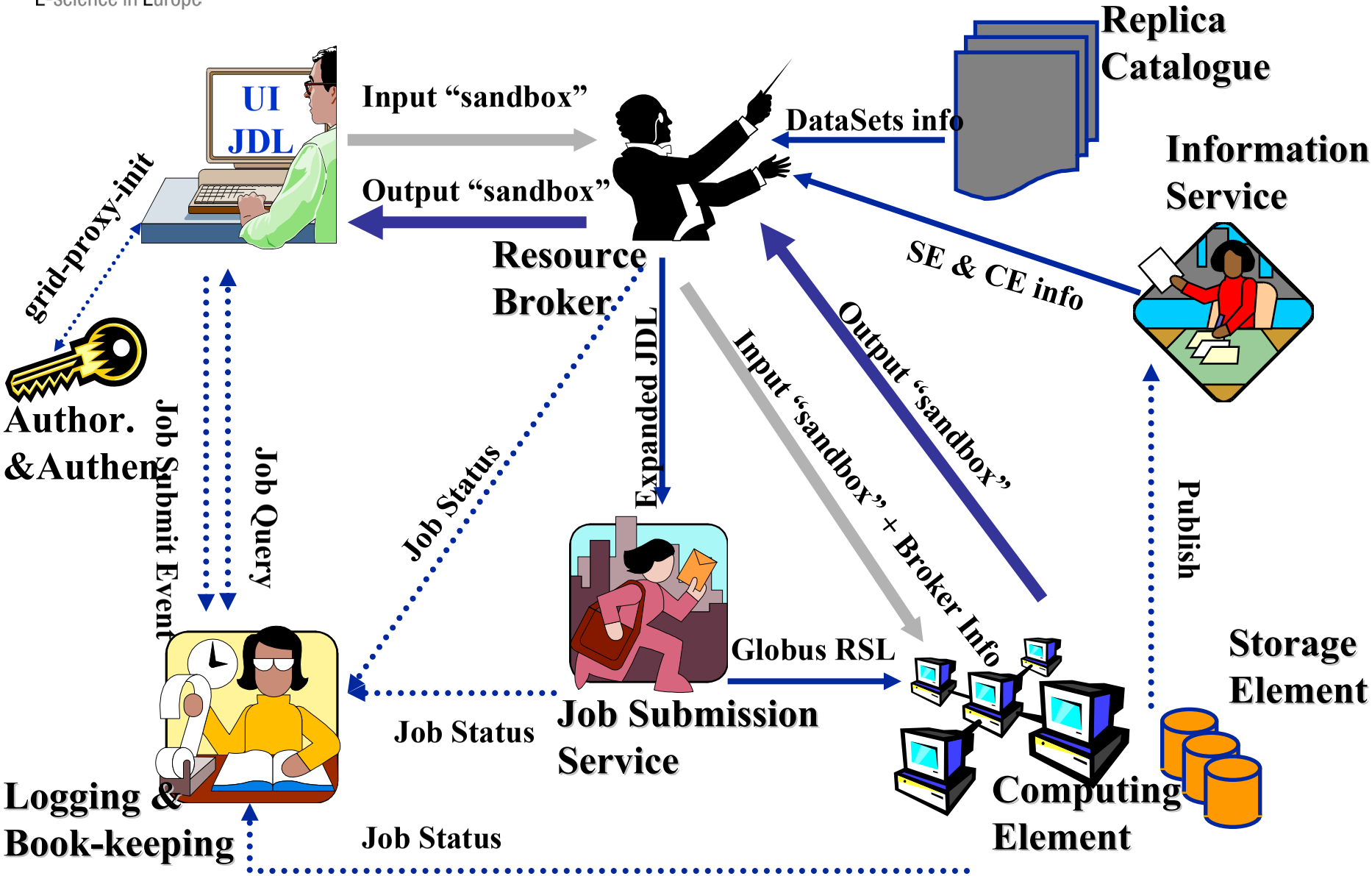
Storage Elements



Computing Elements

Batch Systems
PBS, LSF, etc.

Job workflow



The Command Line user interface

- 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 Description Language (JDL)

- The supported attributes are grouped in two categories:
  - Job Attributes
    - Define the job itself
  - Resources
    - Taken into account by the RB for carrying out the matchmaking algorithm (to choose the “best” resource where to submit the job)
    - *Computing Resource*
      - Used to build expressions of Requirements and/or Rank attributes by the user
      - Have to be prefixed with “other.”
    - *Data and Storage resources*
      - Input data to process, SE where to store output data, protocols spoken by application when accessing SEs

JDL: relevant attributes

- **JobType**
  - *Normal* (simple, sequential job), *Interactive*, *MPICH*, *Checkpointable*
  - Or combination of them
- **Executable** (mandatory)
  - The command name
- **Arguments** (optional)
  - Job command line arguments
- **StdInput**, **StdOutput**, **StdError** (optional)
  - Standard input/output/error of the job
- **Environment**
  - List of environment settings
- **InputSandbox** (optional)
  - List of files on the UI local disk needed by the job for running
  - The listed files will automatically staged to the remote resource
- **OutputSandbox** (optional)
  - List of files, generated by the job, which have to be retrieved

JDL: relevant attributes

- **Requirements**
  - Job requirements on computing resources
  - Specified using attributes of resources published in the Information Service
  - If not specified, default value defined in UI configuration file is considered
    - Default: *other.GlueCEStateStatus == "Production"* (the resource has to be able to accept jobs and dispatch them on WNs)
- **Rank**
  - Expresses preference (how to rank resources that have already met the Requirements expression)
  - Specified using attributes of resources published in the Information Service
  - If not specified, default value defined in the UI configuration file is considered
    - Default: - *other.GlueCEStateEstimatedResponseTime* (the lowest estimated traversal time)
    - Default: *other.GlueCEStateFreeCPUs* (the highest number of free CPUs) for parallel jobs (see later)

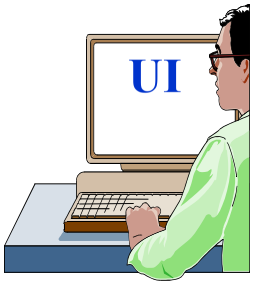
JDL: relevant attributes

- **InputData**
  - Refers to data used as input by the job: these data are published in the Replica Location Service (RLS) and stored in the SEs)
  - LFNs and/or GUIDs
- **DataAccessProtocol** (mandatory if InputData has been specified)
  - The protocol or the list of protocols which the application is able to speak with for accessing *InputData* on a given SE
- **OutputSE**
  - The Uniform Resource Identifier of the output SE
  - RB uses it to choose a CE that is compatible with the job and is close to SE

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



RB node

Network Server

Workload Manager

Job Contr. - CondorG

Replica Location Server

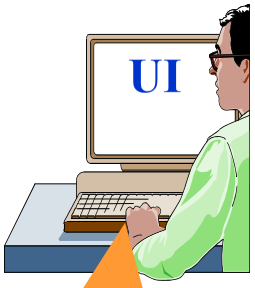
Inform. Service

CE characts & status

SE characts & status

Computing Element

Storage Element



UI: allows users to access the functionalities of the WMS
(via command line, GUI, C++ and Java APIs)

RB node

Network Server

Workload Manager

Job Contr. - CondorG

Replica Location Server

Inform. Service

Job Statu s

submitted

CE characts & status

SE characts & status

Computing Element

Storage Element

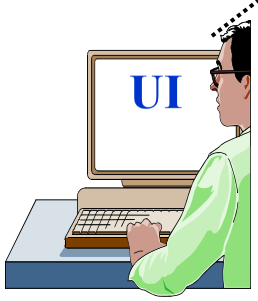
edg-job-submit myjob.jdl

Myjob.jdl

```
JobType = "Normal";
Executable     = "$(CMS)/exe/sum.exe";
InputSandbox = {"/home/user/WP1testC","/home/file*", "/home/user/DATA/*"};
OutputSandbox = {"sim.err", "test.out", "sim.log"};
Requirements   = other. GlueHostOperatingSystemName == "linux" &&
other. GlueHostOperatingSystemRelease == "Red Hat 6.2" &&
other.GlueCEPolicyMaxWallClockTime > 10000;
Rank  = other.GlueCEStateFreeCPUs;
```

Job Statu s

submitted



Workload Manager

Inform. Service

Job Contr. - CondorG

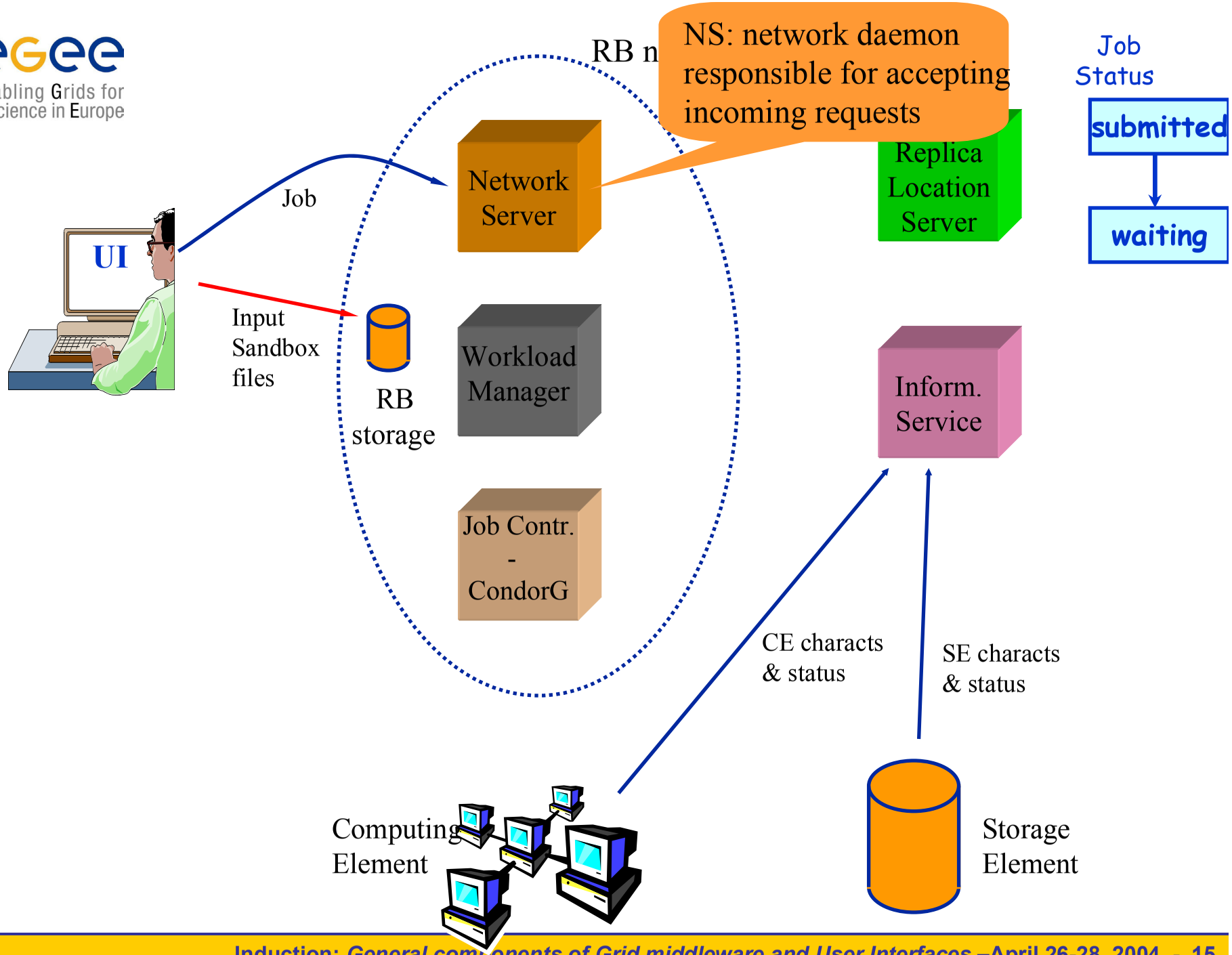
Job Description Languag (JDL) to specify job characteristics and requirements

CE characts & status

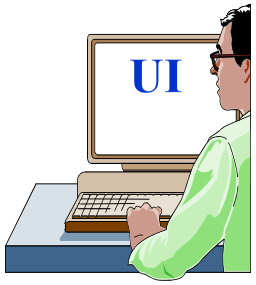
SE characts & status

Computing Element

Storage Element


RB n
NS: network daemon responsible for accepting incoming requests
Job Status
submitted
waiting
Job
Network Server
Replica Location Server
UI
Input Sandbox files
RB storage
Workload Manager
Inform. Service
Job Contr. - CondorG
CE characts & status
SE characts & status
Computing Element
Storage Element

Job submission



RB node

Network Server

Job

Workload Manager

RB storage

Job Contr. - CondorG

WM: responsible to take the appropriate actions to satisfy the request

Replica Location Server

Inform. Service

Job Status

submitted

waiting

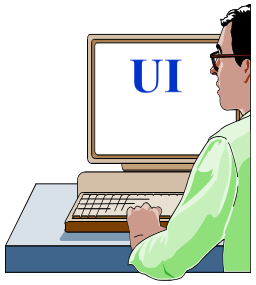
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

Match-Maker/ Broker

Replica Location Server

RB storage

Workload Manager

Where must this job be executed ?

Inform. Service

Job Contr. - CondorG

Job Status

submitted

waiting

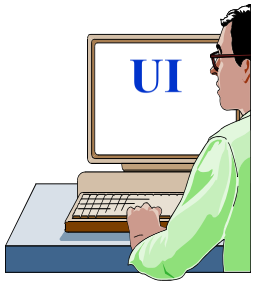
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Matchmaker: responsible to find the "best" CE where to submit a job

Match-Maker/ Broker

Workload Manager

RB storage

Job Contr. - CondorG

Replica Location Server

Inform. Service

Job Status

submitted

waiting

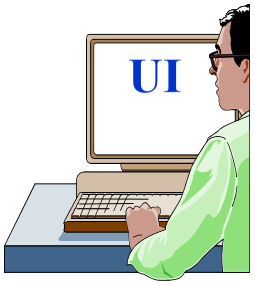
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Match-Maker/ Broker

Where are (which SEs) the needed data ?

Replica Location Server

Inform. Service

What is the status of the Grid ?

CE characts & status

SE characts & status

Computing Element

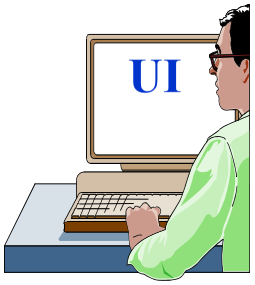
Storage Element

Job Status

submitted

waiting

Job submission



RB node

Network Server

Match-Maker/ Broker

Replica Location Server

Job Status

submitted

waiting

RB storage

Workload Manager

CE choice

Inform. Service

Job Contr. - CondorG

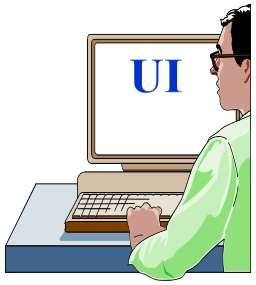
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

RB storage

Workload Manager

Job Contr. - CondorG

Job Adapter

Replica Location Server

Inform. Service

Job Status

submitted

waiting

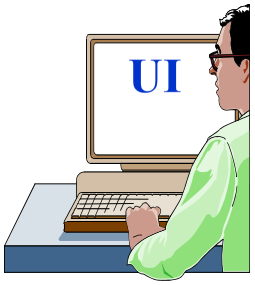
JA: responsible for the final "touches" to the job before performing submission (e.g. creation of wrapper script, etc.)

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

RB storage

Workload Manager

Job

Job Contr. - CondorG

Replica Location Server

Inform. Service

Job Status

submitted

waiting

ready

JC: responsible for the actual job management operations (done via CondorG)

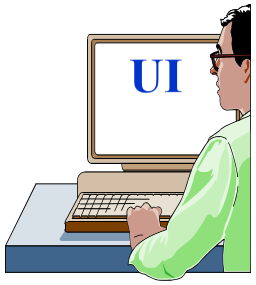
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Replica Location Server

Inform. Service

Job Status

submitted

waiting

ready

scheduled

Input Sandbox files

Job

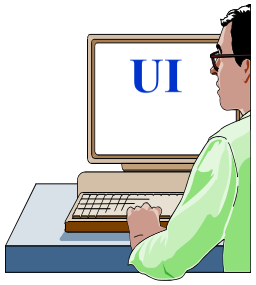
CE characts & status

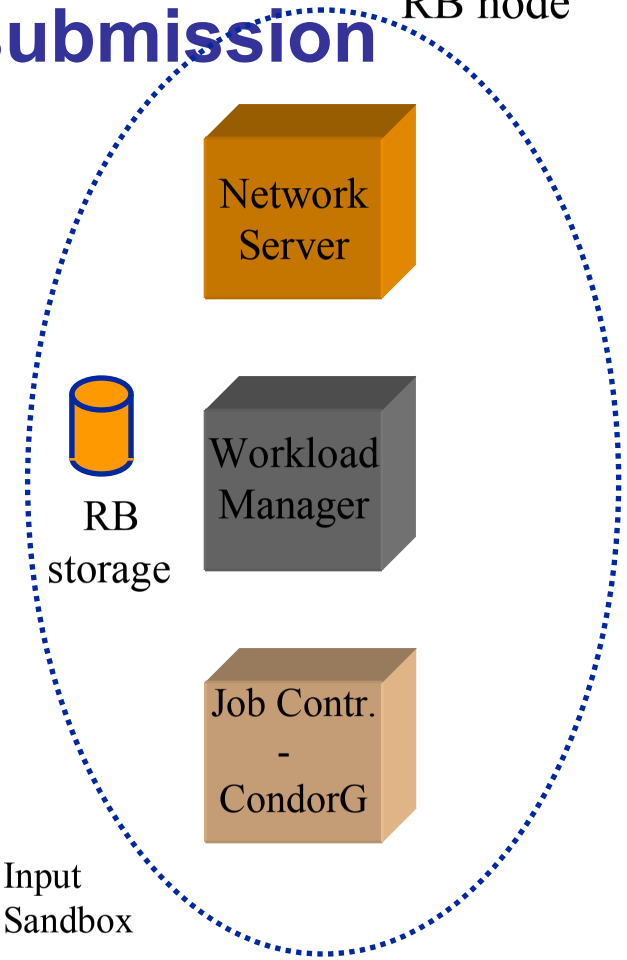
SE characts & status

Computing Element

Storage Element

Job submission

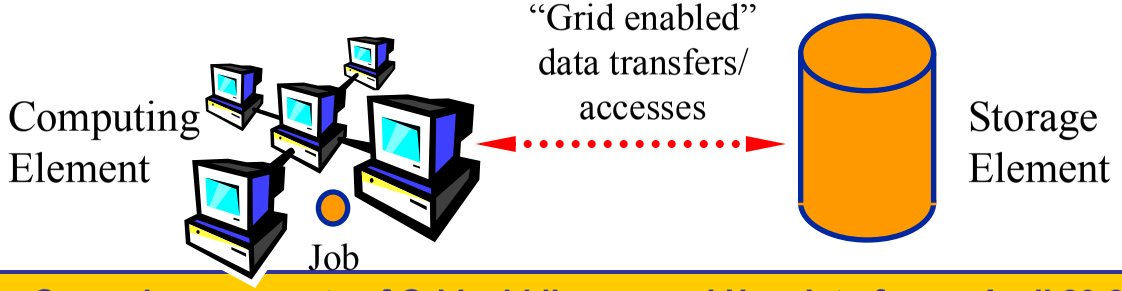




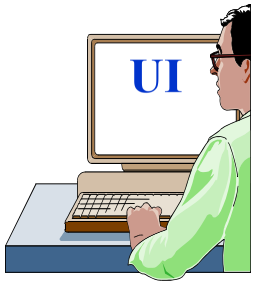
Replica Location Server

Inform. Service

Job Status



Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Output Sandbox files

Computing Element

Replica Location Server

Inform. Service

Storage Element

Job Status

submitted

waiting

ready

scheduled

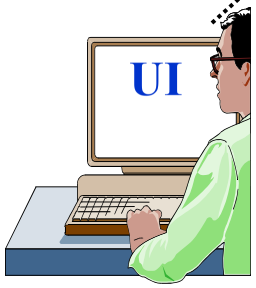
running

done

Enabling Grids for E-science in Europe

Jo

edg-job-get-output <dg-job-id>



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Output Sandbox

Replica Location Server

Inform. Service

Job Status

submitted

waiting

ready

scheduled

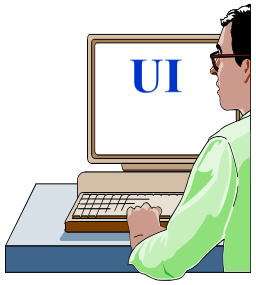
running

done

Computing Element

Storage Element

Job submission



Output Sandbox files

RB node

RB storage

Network Server

Workload Manager

Job Contr. - CondorG

Replica Location Server

Inform. Service

Job Status

submitted

waiting

ready

scheduled

running

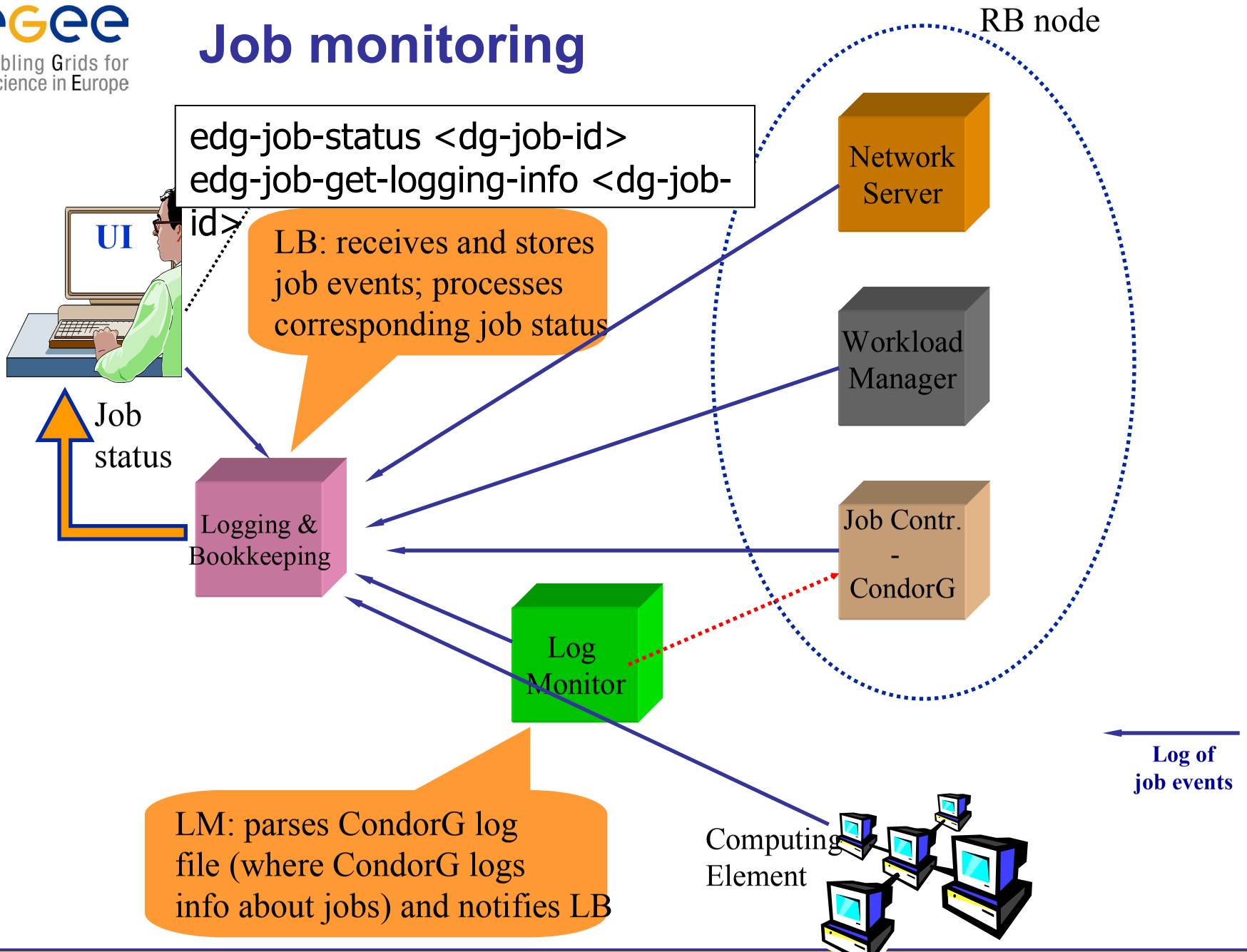
done

cleared

Computing Element

Storage Element

Job monitoring

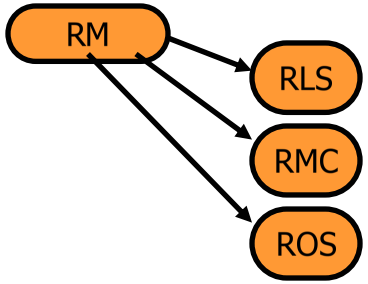


Other (most relevant) UI commands

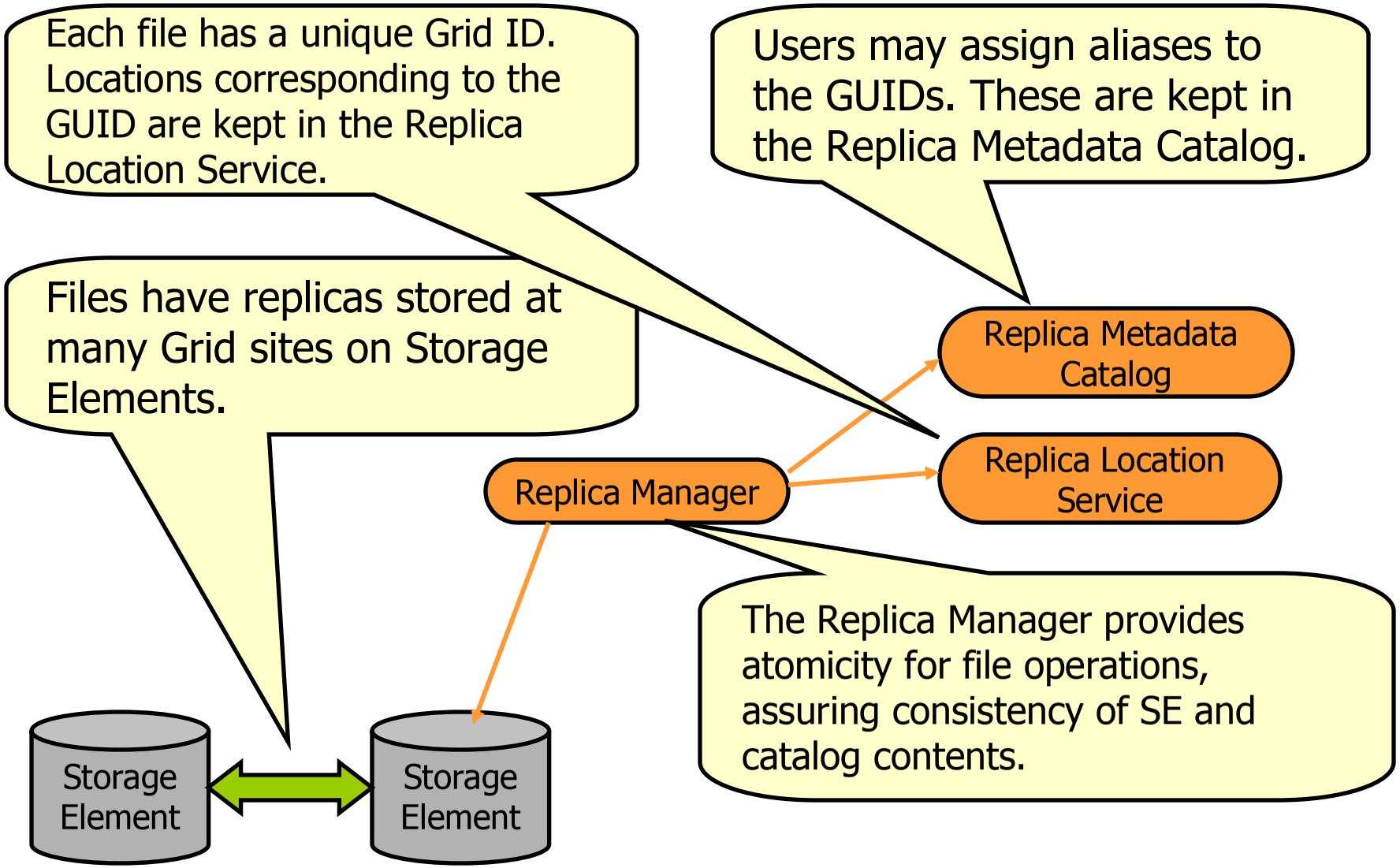
- **`edg-job-list-match`**
  - Lists resources matching a job description
  - Performs the matchmaking without submitting the job
- **`edg-job-cancel`**
  - Cancels a given job
- **`edg-job-status`**
  - Displays the status of the job
- **`edg-job-get-output`**
  - Returns the job-output (the OutputSandbox 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

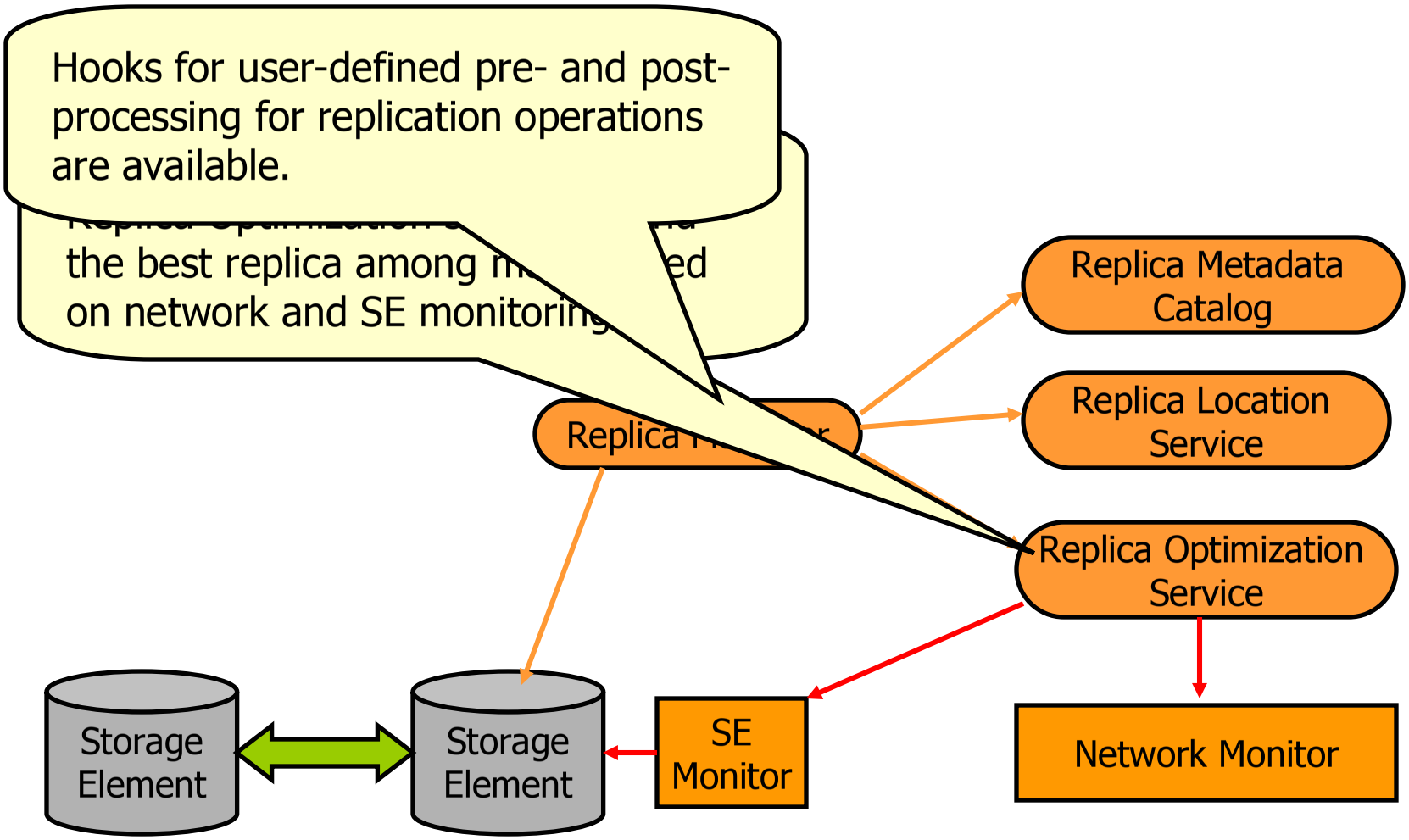
- Tools for
  - Locating data
  - Copying data
  - Managing and replicating data
  - Meta Data management
- On EDG you have
  - Replica Location Service (RLS)
  - Replica Metadata Service (RMC)
  - Replica Optimisation Service (ROS)
  - Replica Manager (RM)



Replication Services: Basic Functionality

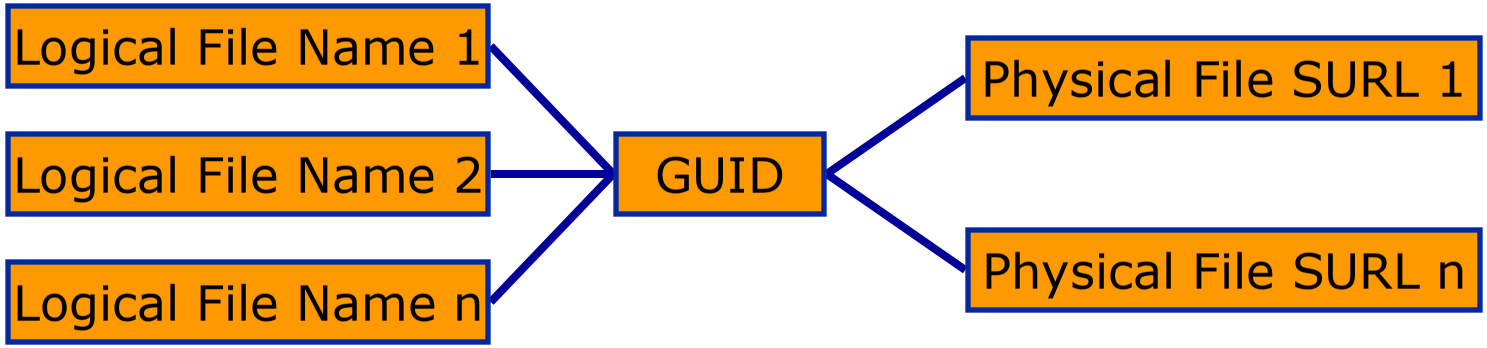


Higher Level Replication Services



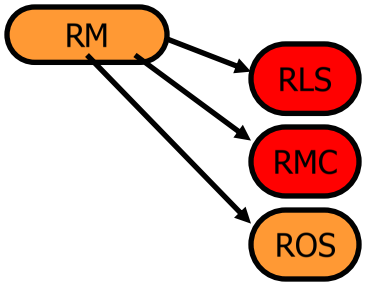
Naming Conventions

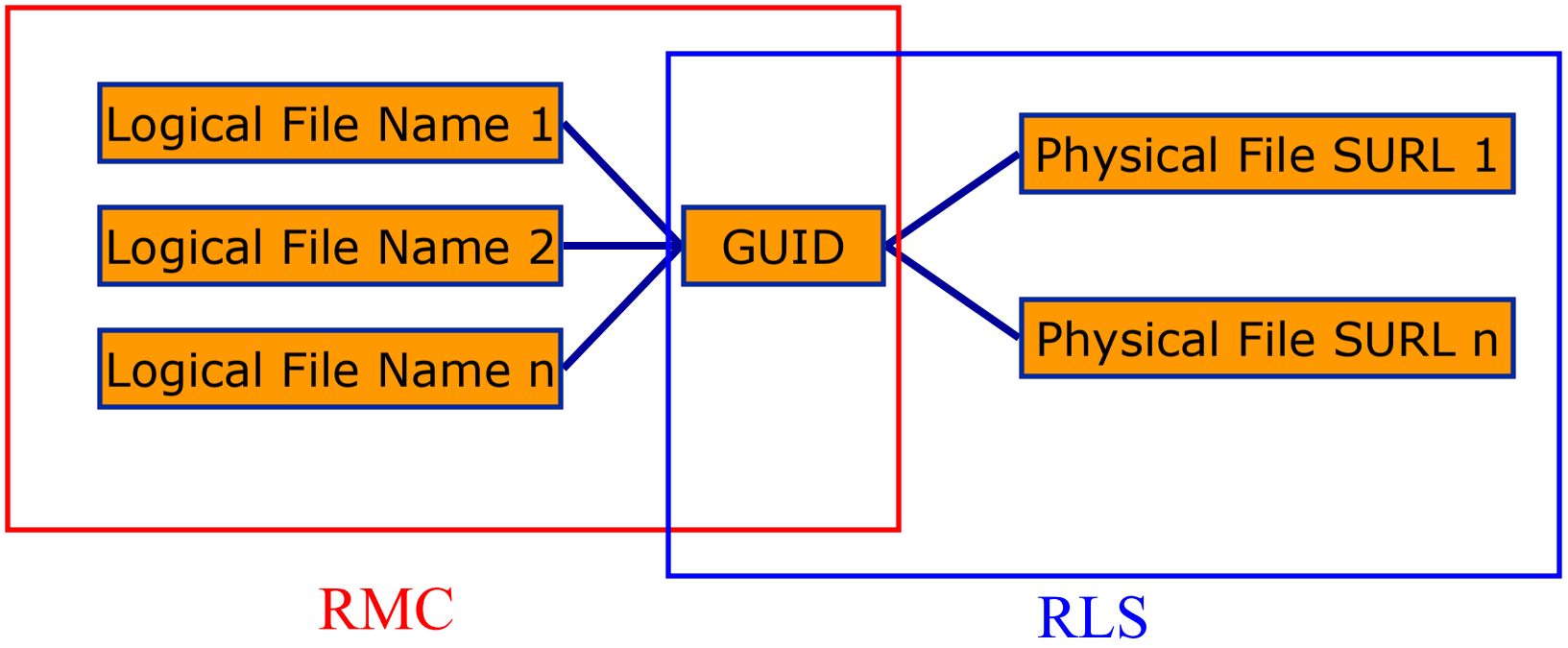
- Logical File Name (LFN)
  - An alias created by a user to refer to some item of data e.g. “lfn:cms/20030203/run2/track1”
- Site URL (SURL) (or Physical File Name (PFN))
  - The location of an actual piece of data on a storage system e.g. “srm://pcrd24.cern.ch/flatfiles/cms/output10_1”
- Globally Unique 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)

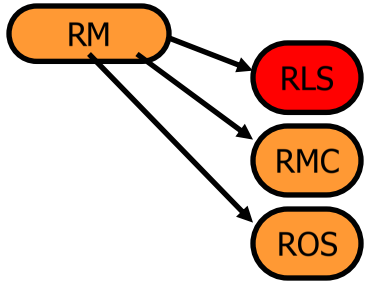
- RMC:
  - Stores LFN-GUID mappings
- RLS:
  - Stores GUID-SURL mappings





Replica Location Service (RLS)

- The Replica Location Service is a system that maintains and provides access to information about the physical location of copies of data files.
- It is a distributed service that stores mappings between globally unique identifiers of datafiles and the physical identifiers of all existing replicas of these datafiles.
- Design is a joint collaboration between Globus and EDG-WP2



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 – *Catalog Commands (2)*

- addAlias args: guid, lfn
  - Add a new alias to GUID mapping
- removeAlias args: guid, lfn
  - Remove an alias LFN from a known GUID.
- printInfo()
  - Print the information needed by the Replica Manager to screen or to a file.
- getVersion()
  - Get the versions of the replica manager client.

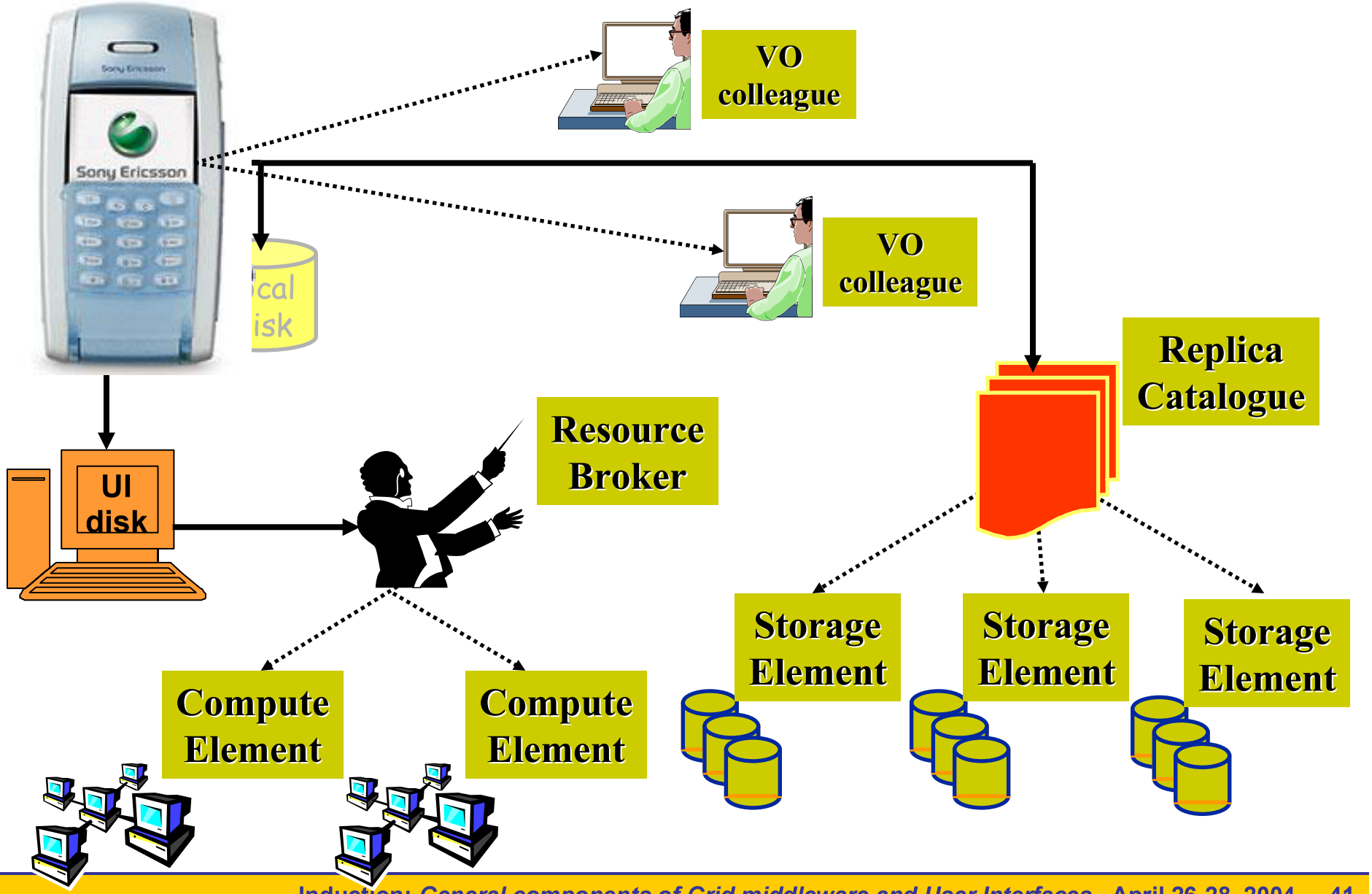
The Replica Manager Interface – *Optimization Commands*

- 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.
- getAccessCost args: lfn/guid[], ce[], protocol[]
  - Calculates the expected cost of accessing all the files specified by logicalName from each Computing Element host specified by ceHosts.

The Replica Manager Interface – *File Transfer Commands*

- copyFile args: soure, dest
  - Copy a file to a non-grid destination.
- listDirectory args: dir
  - List the directory contents on an SRM or a GridFTP server.

The “transparent” grid access



A web portal: why and how ?

- It can be accessed from everywhere and by “everything” (desktop, laptop, PDA, phone).
- It can keep the same user interface independently of the underlying middleware.
- It must be redundantly “secure” at all levels: **1)** secure for web transactions, **2)** secure for user credentials, **3)** secure for user authentication, **4)** secure at VO level.
- All available grid services must be incorporated in a logic way, just “one mouse click away”.
- Its layout must be easily understandable and user friendly.

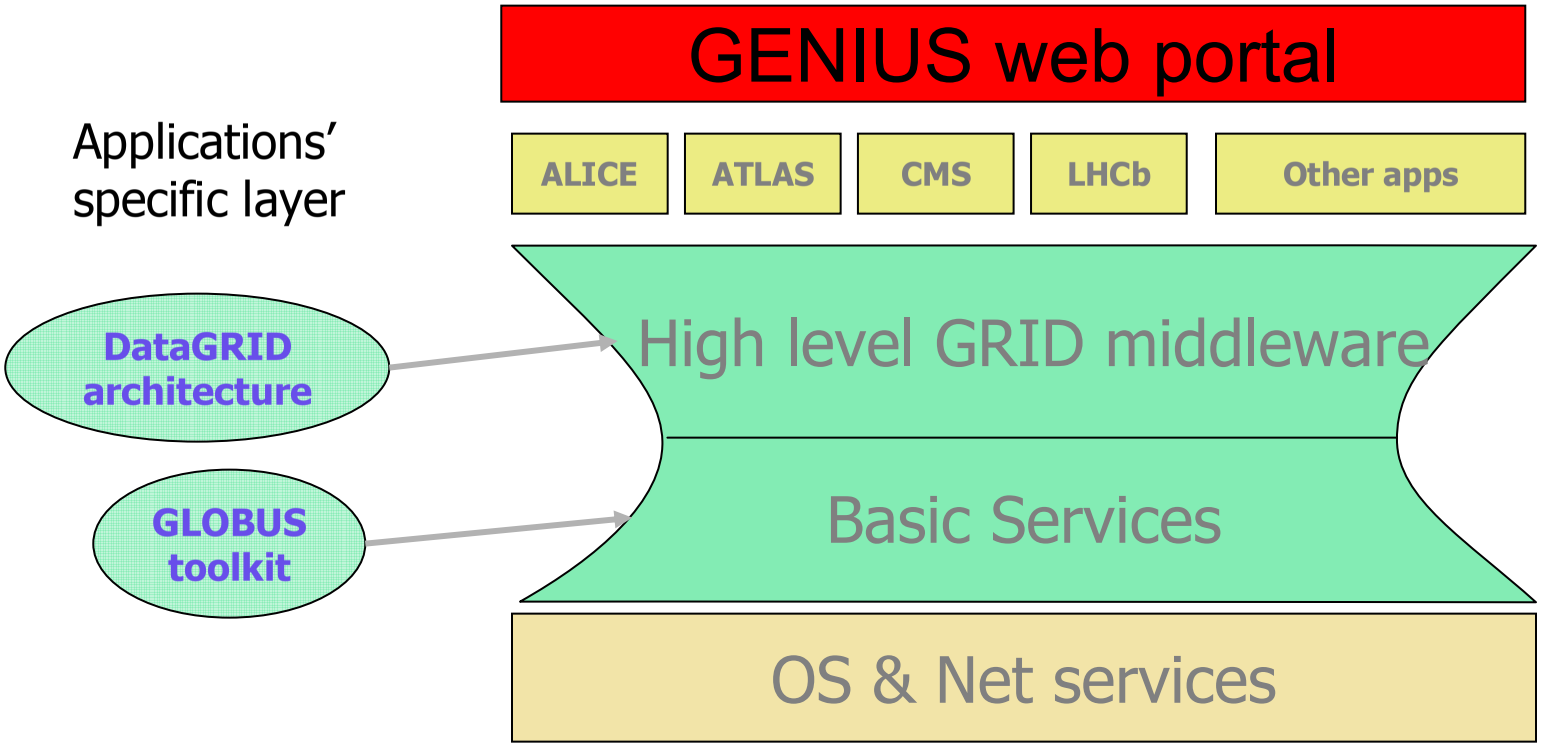
Title 1

GENIUS®

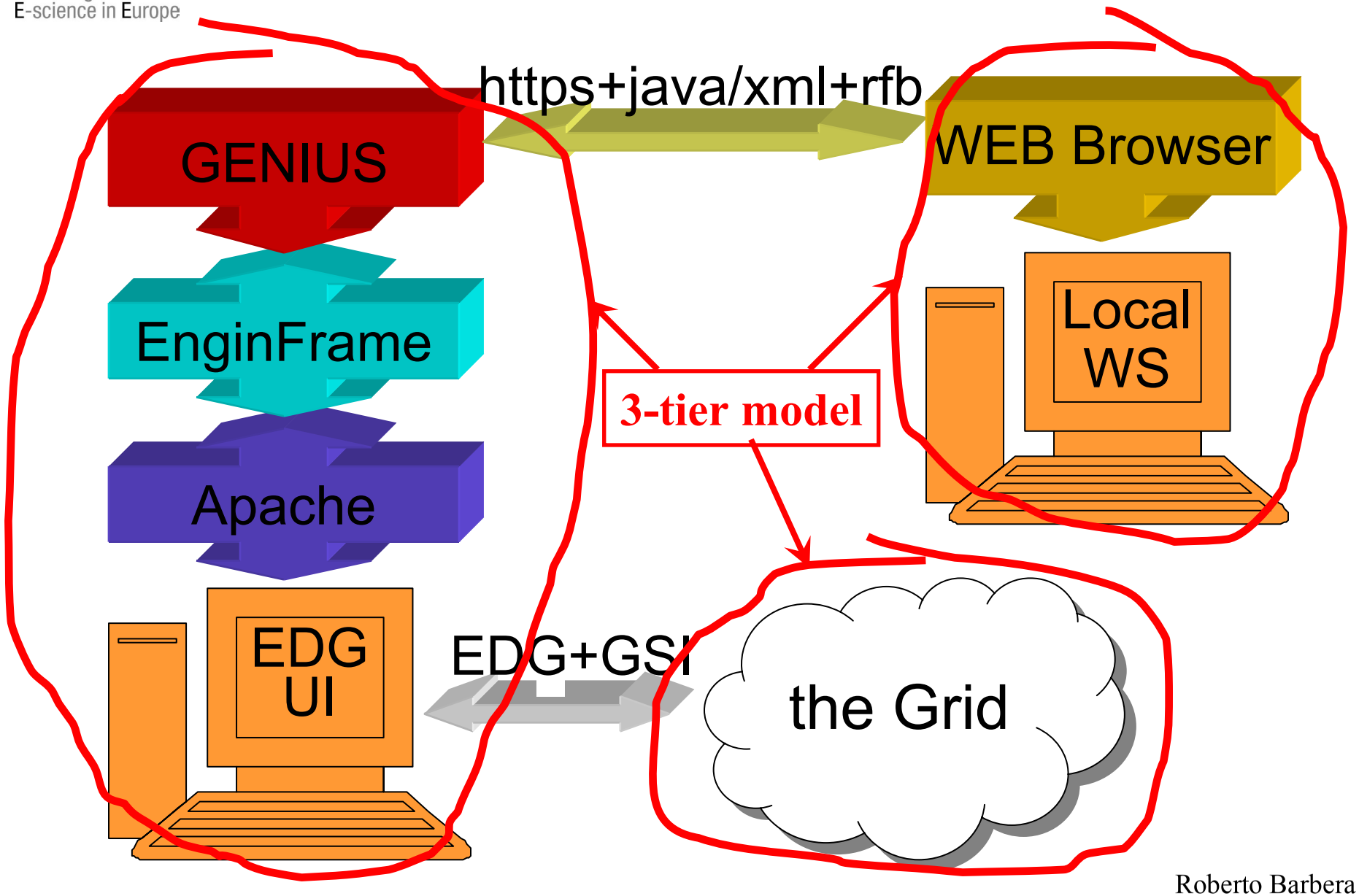
(**G**rid **E**nabled web e**N**vironment for site **I**ndependent **U**ser job **S**ubmission)

[https://genius.ct.infn.it]

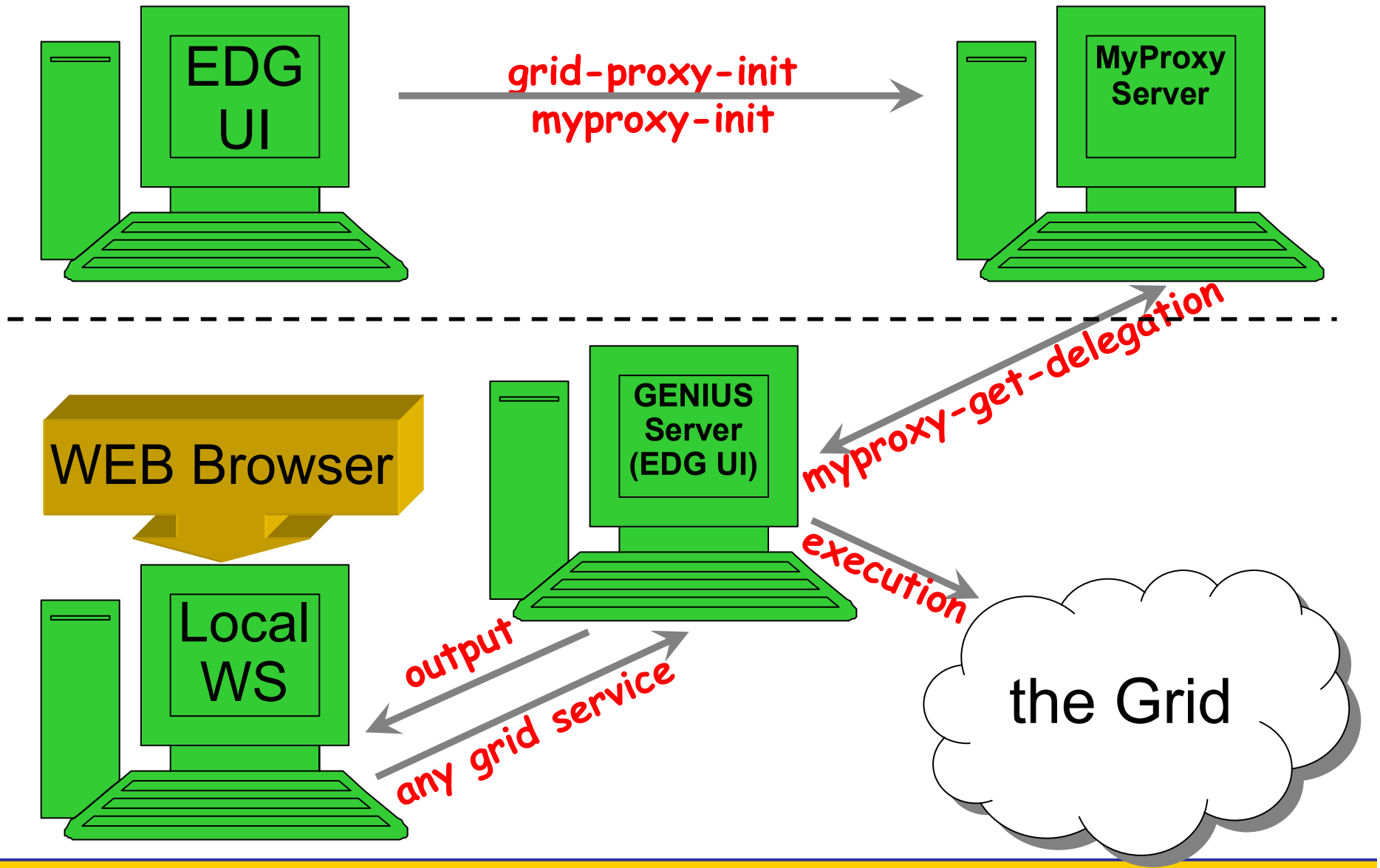
INFN/NICE collaboration



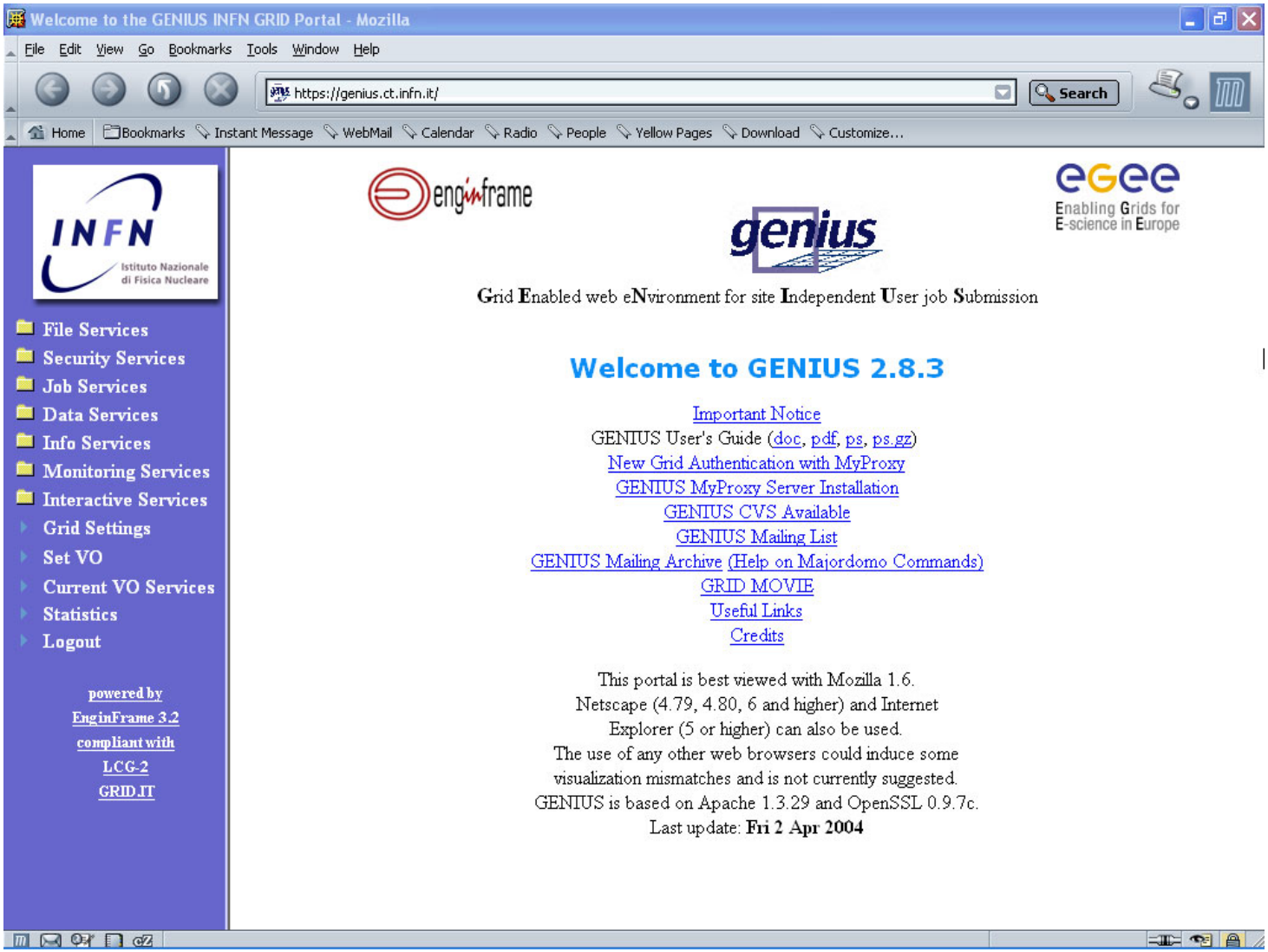
GENIUS: how it works



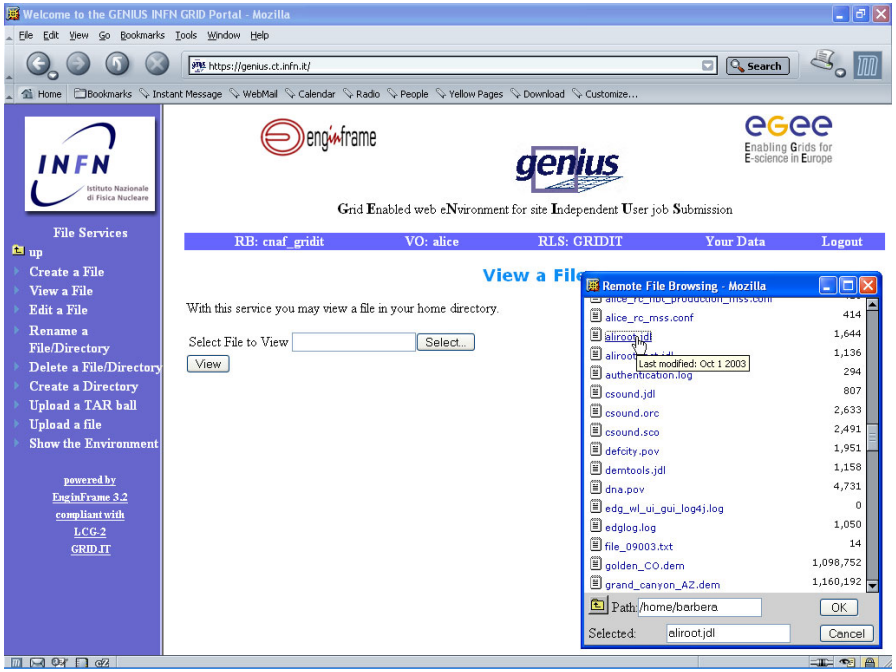
Grid authentication with MyProxy

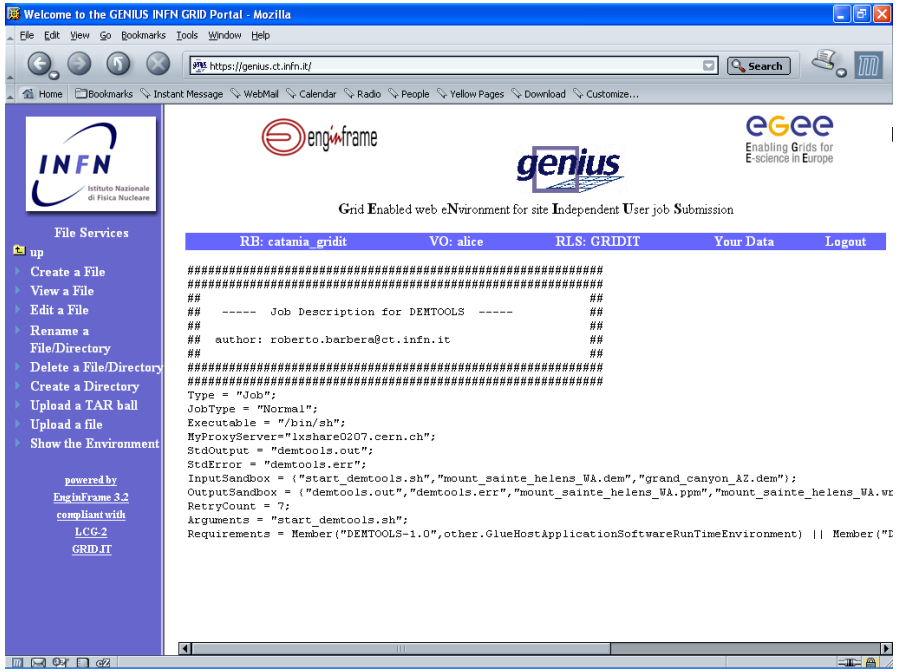


GENIUS home page

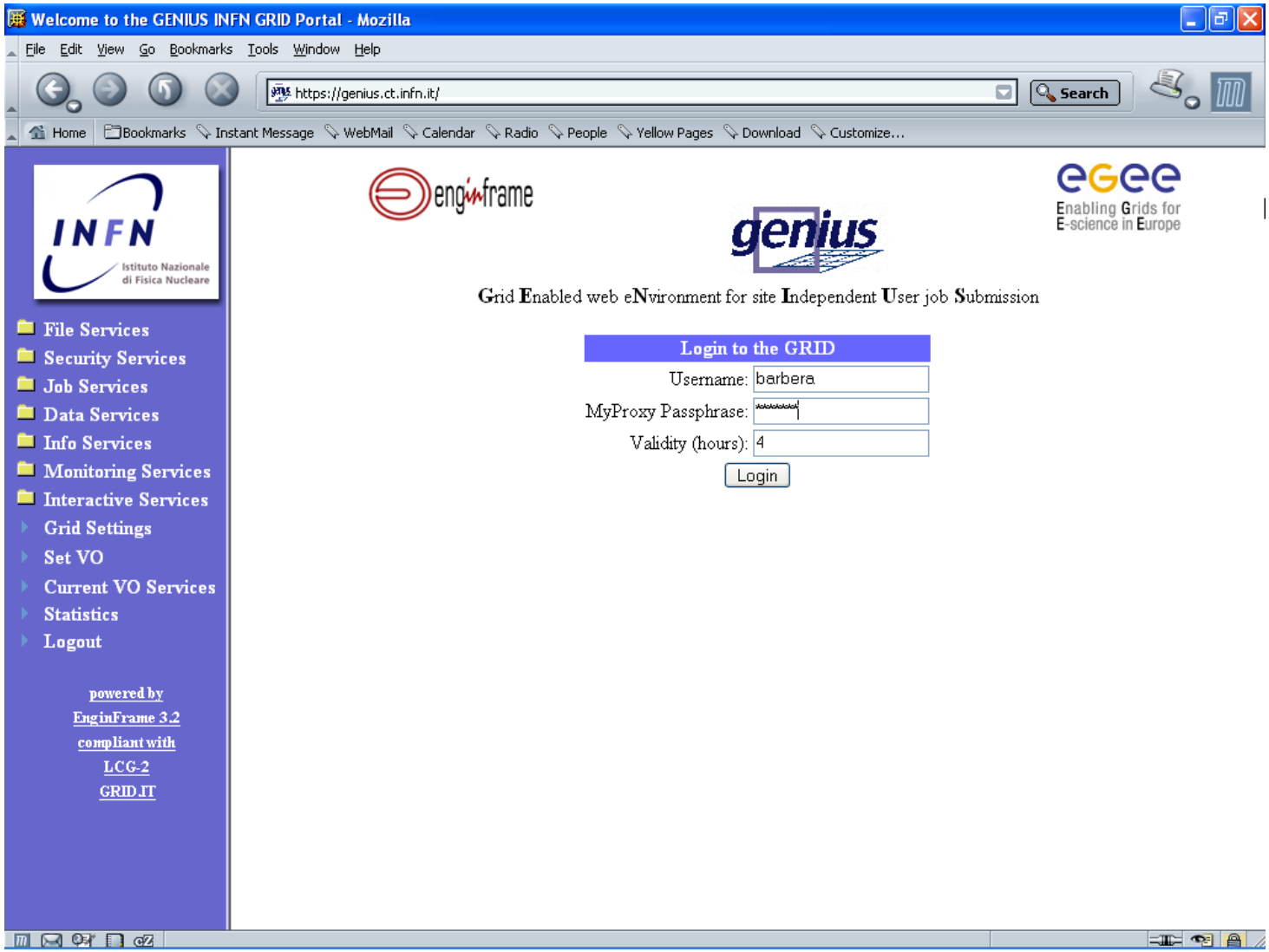


GENIUS file services

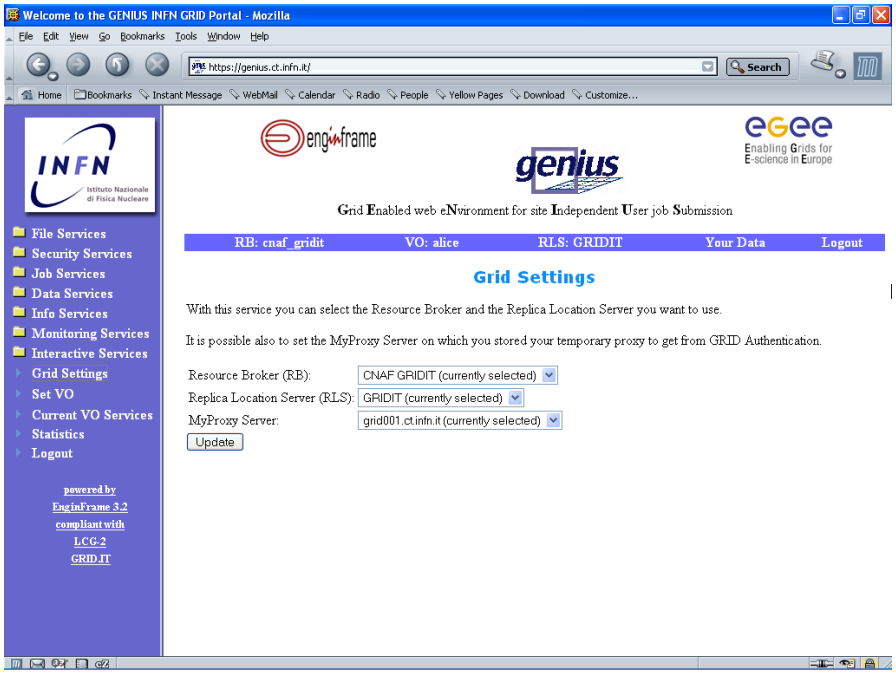


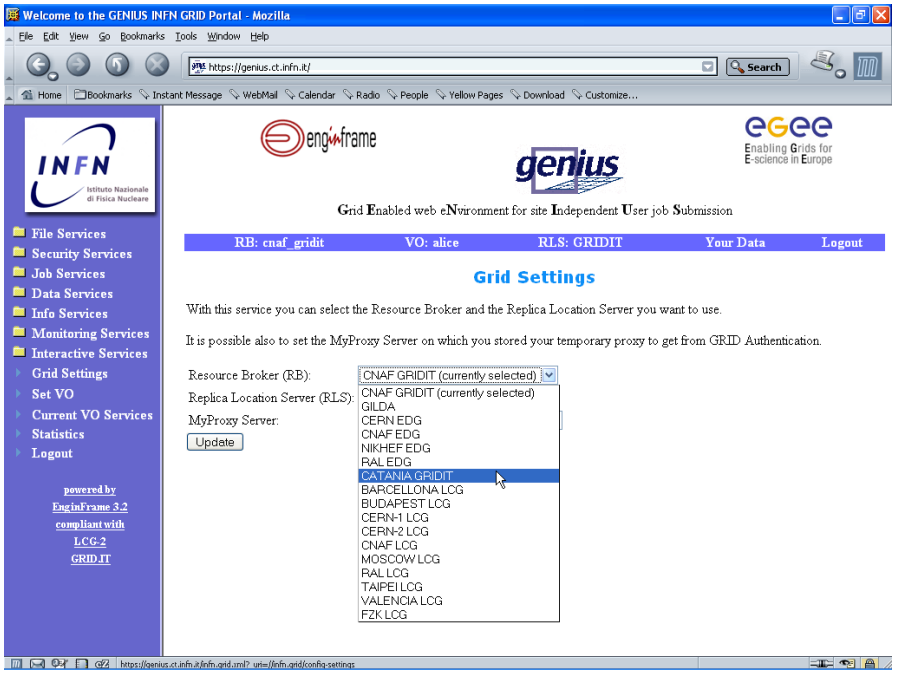


GENIUS authentication with MyProxy

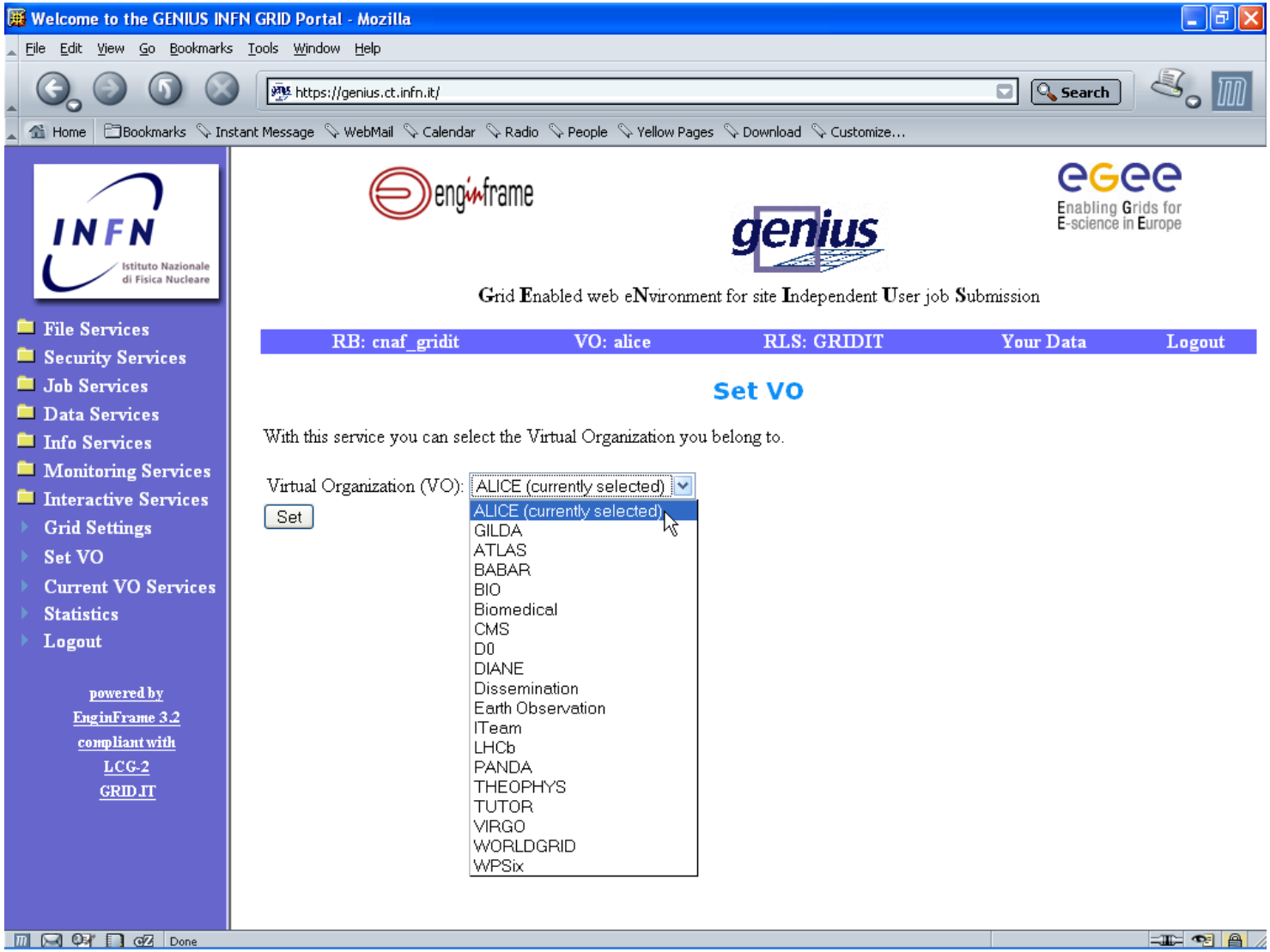


GENIUS grid settings

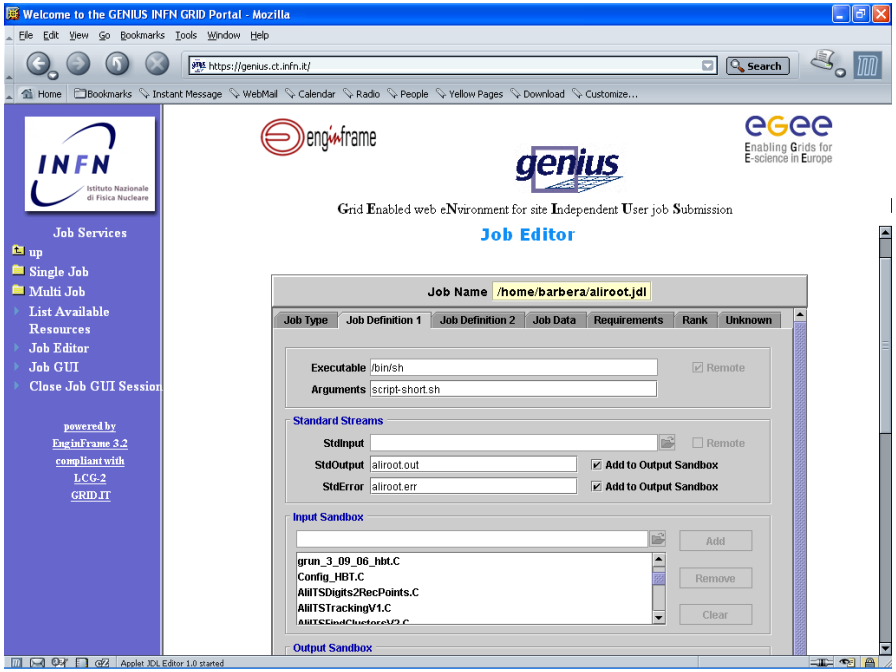


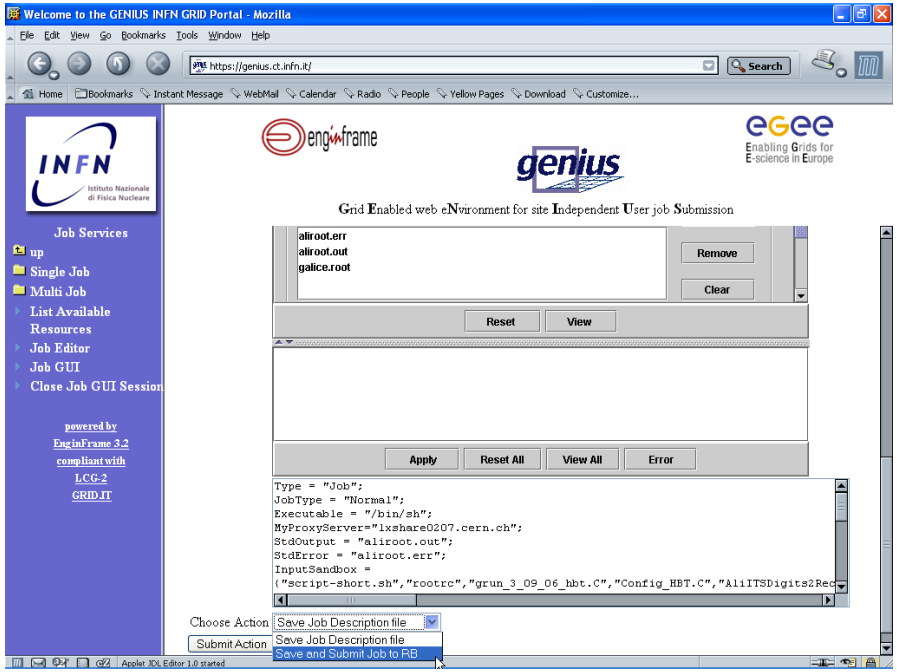


GENIUS VO selection

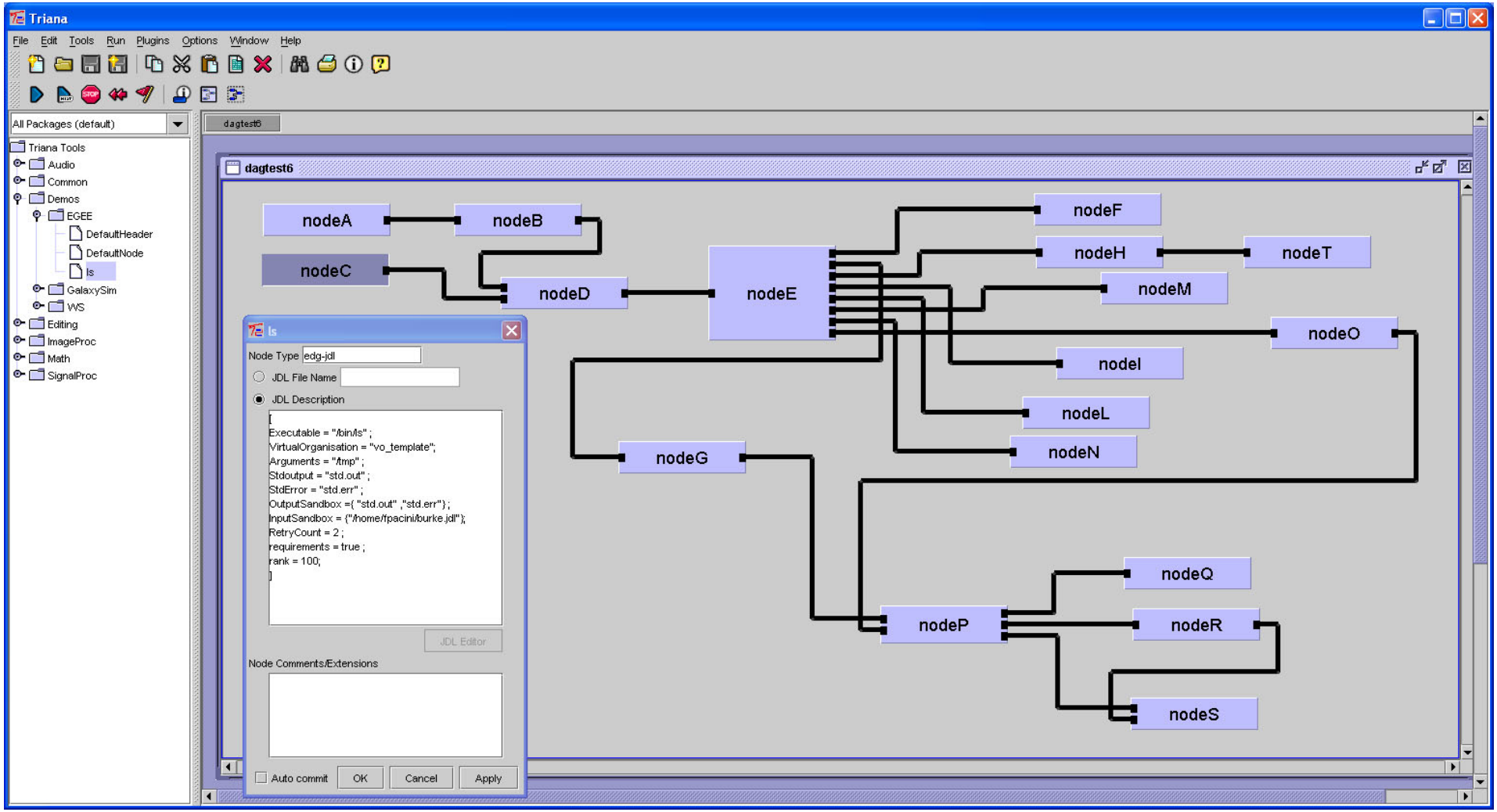


GENIUS graphic job editor (work in collaboration with DATAMAT)

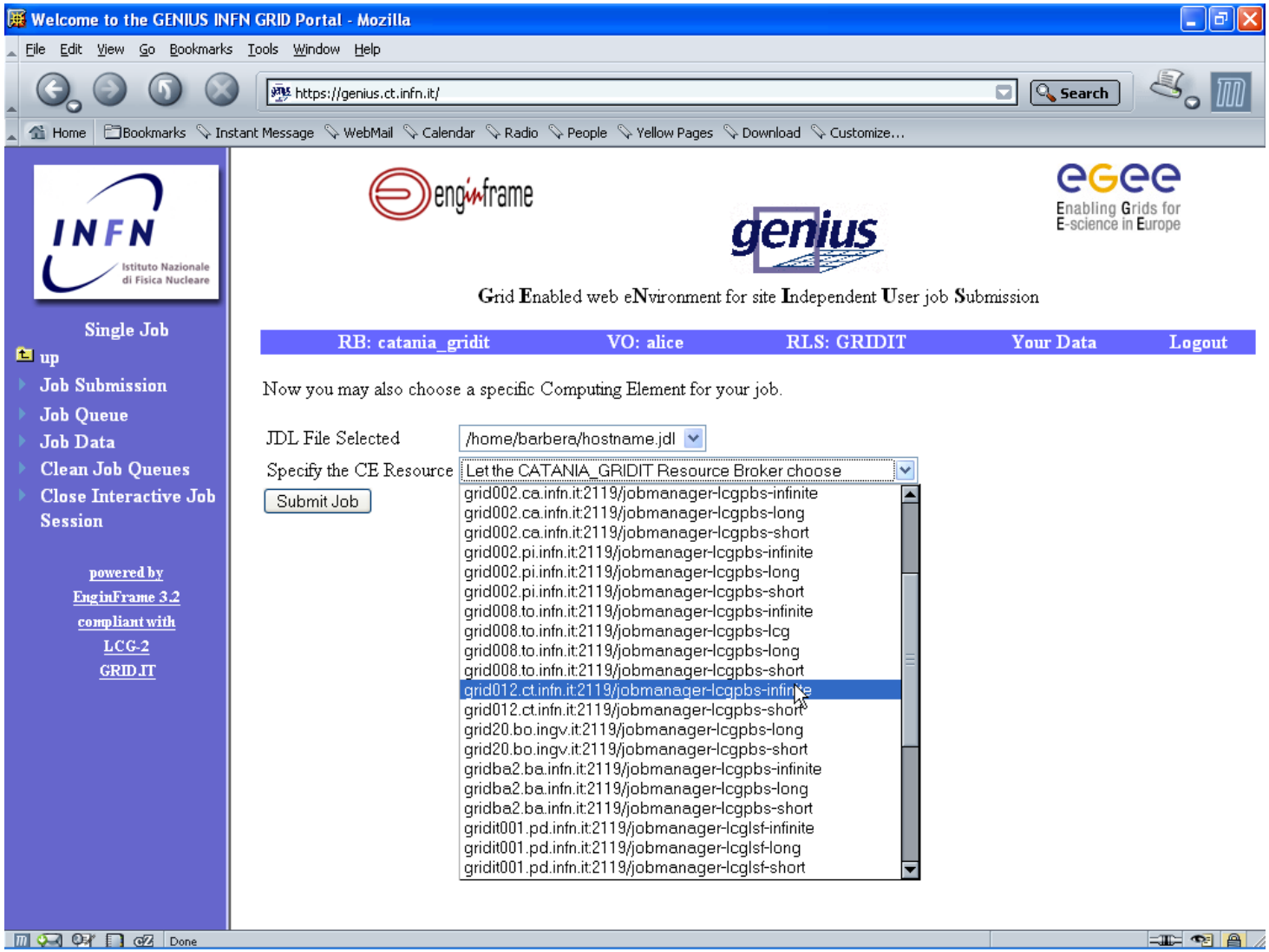




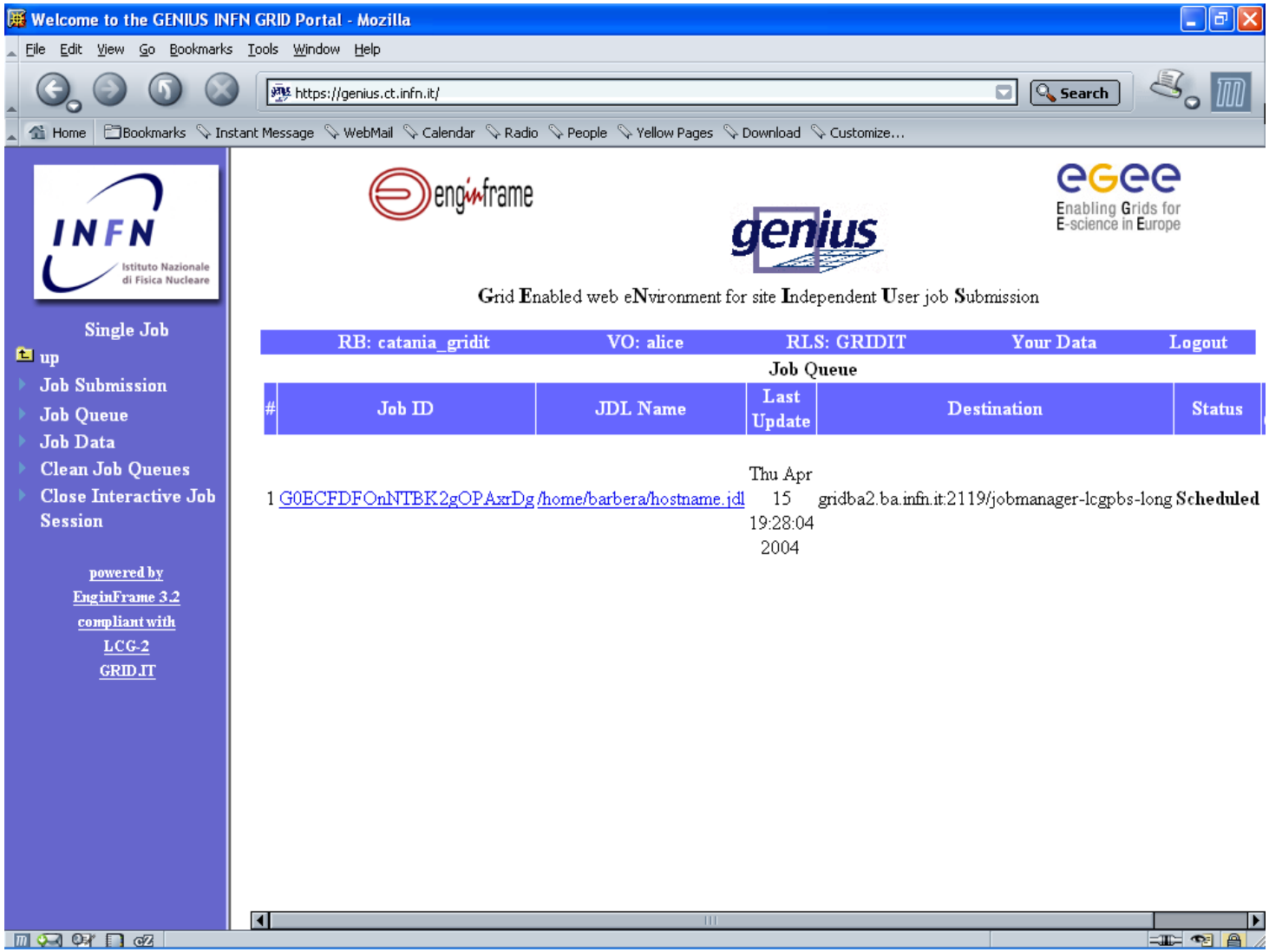
GENIUS graphic workflow editor (integration with TRIANA from GridLab)



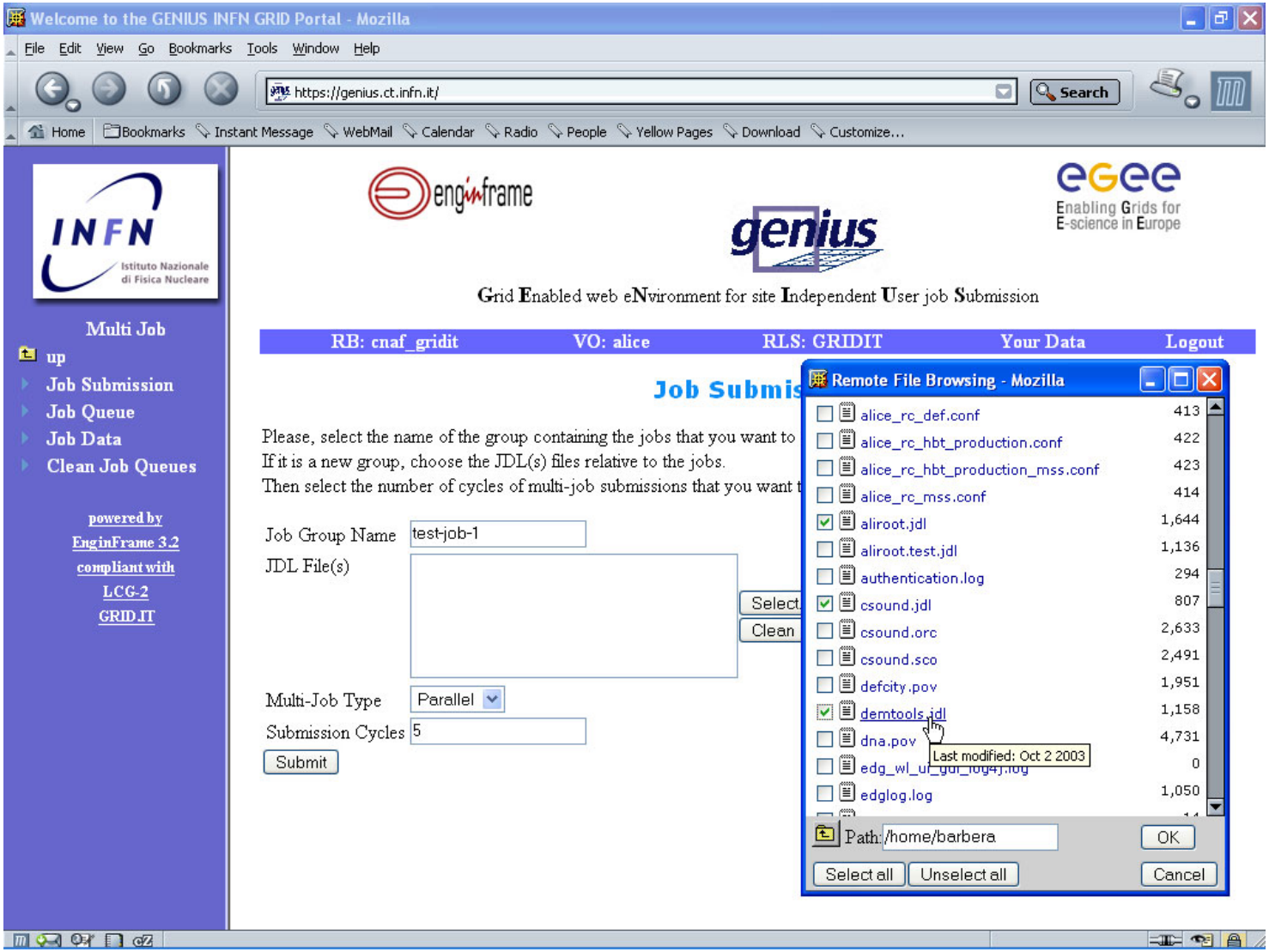
GENIUS single job submission



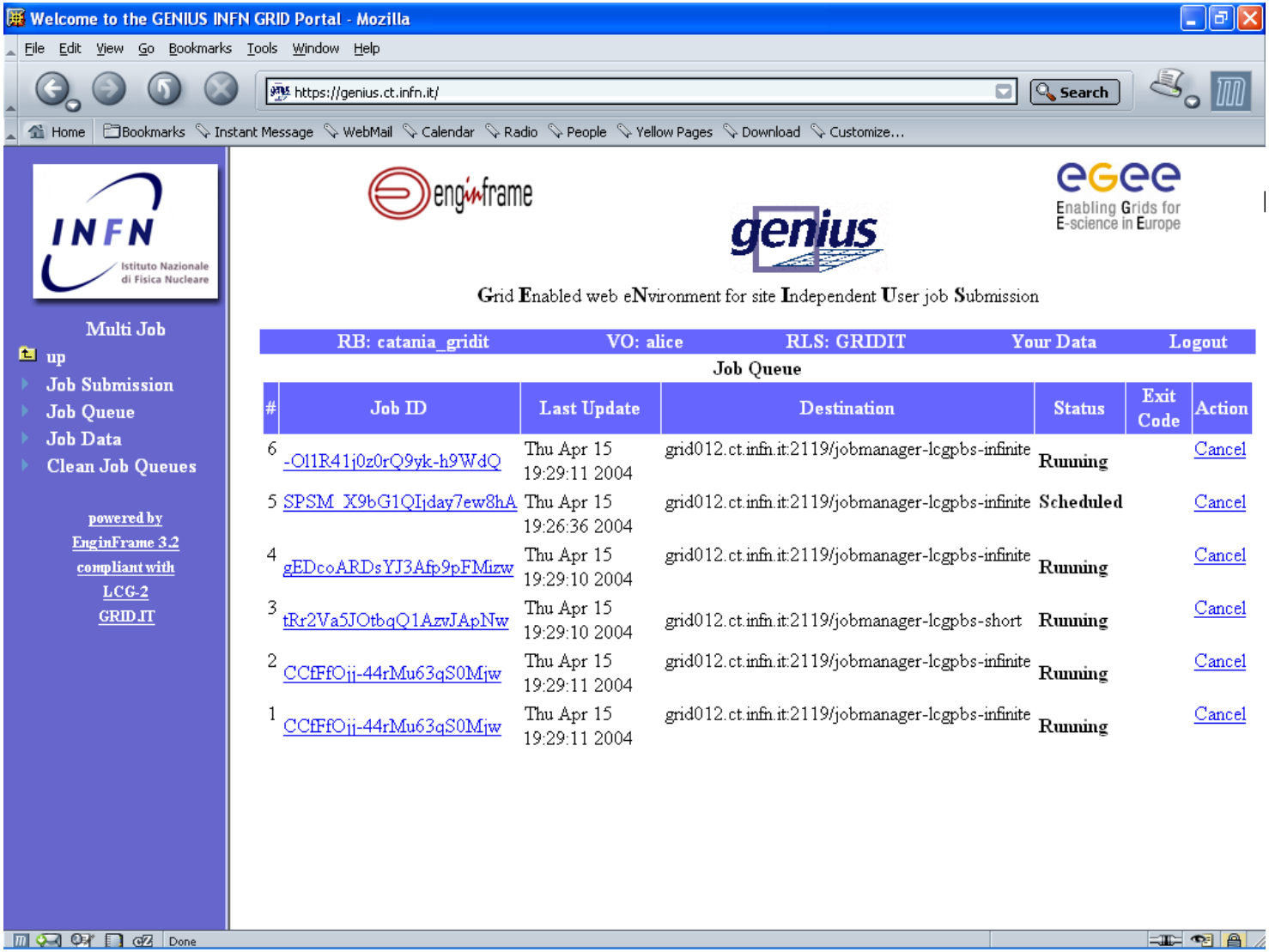
GENIUS single job queue



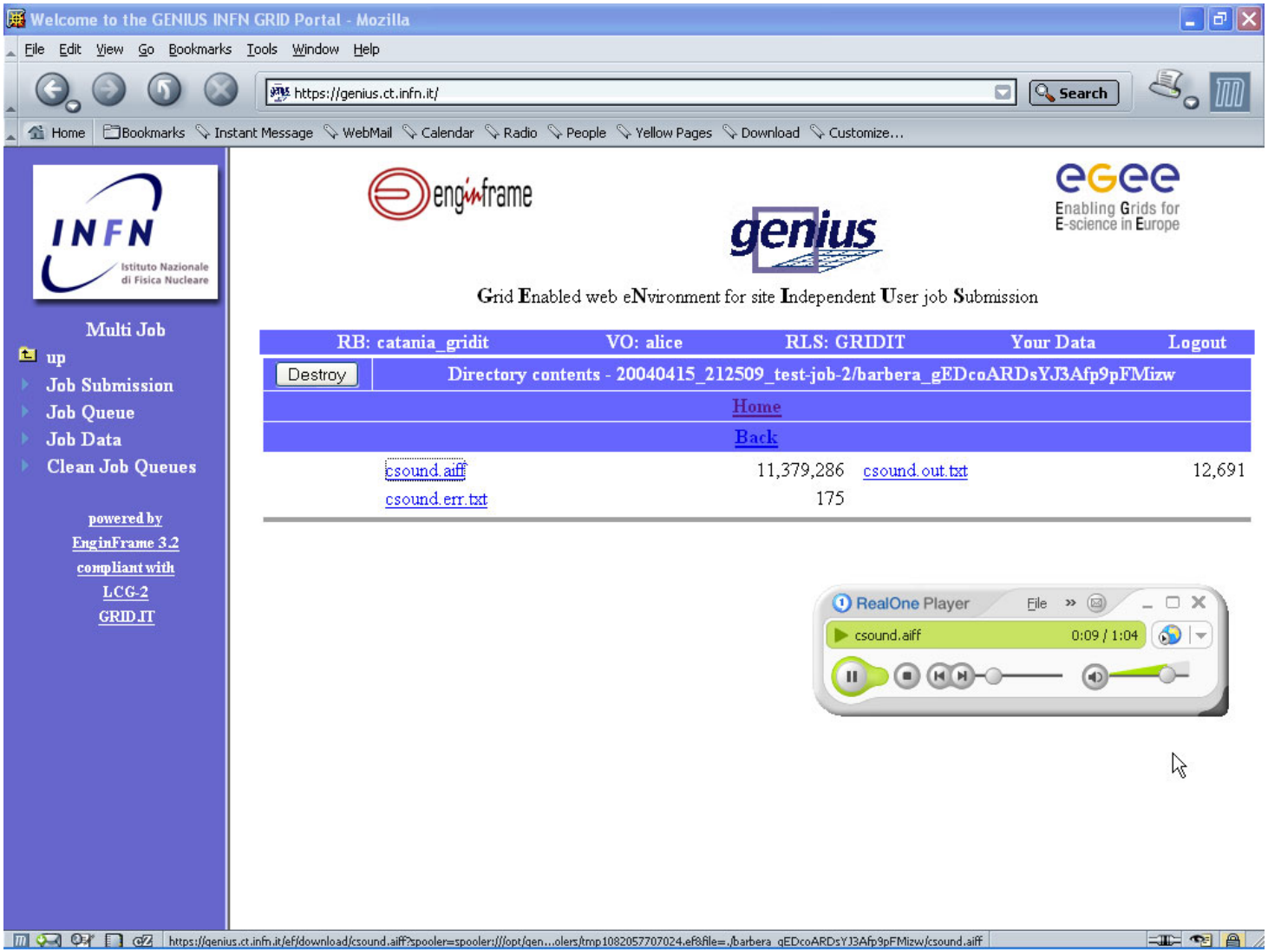
GENIUS multi job submission



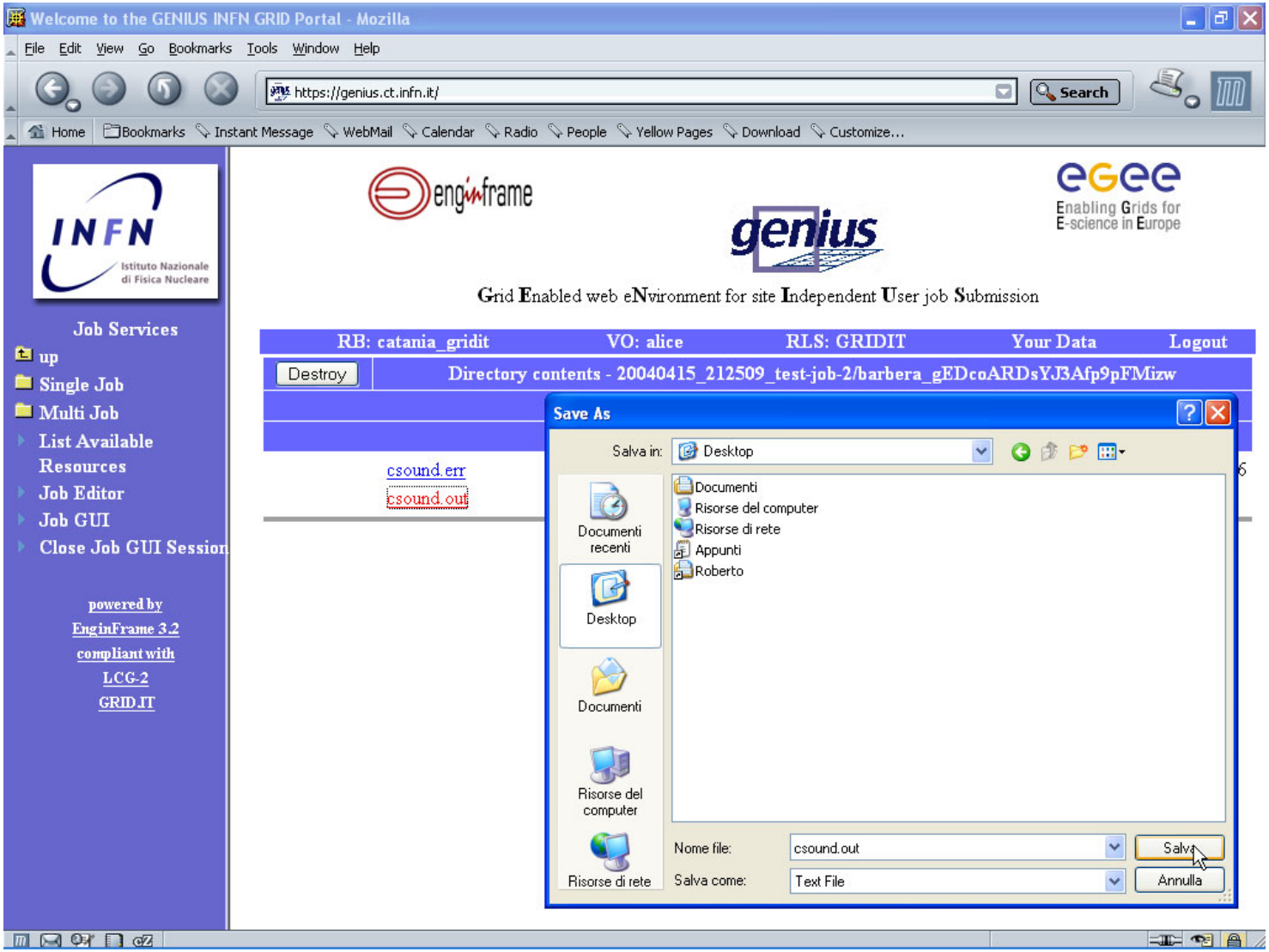
GENIUS multi job queue



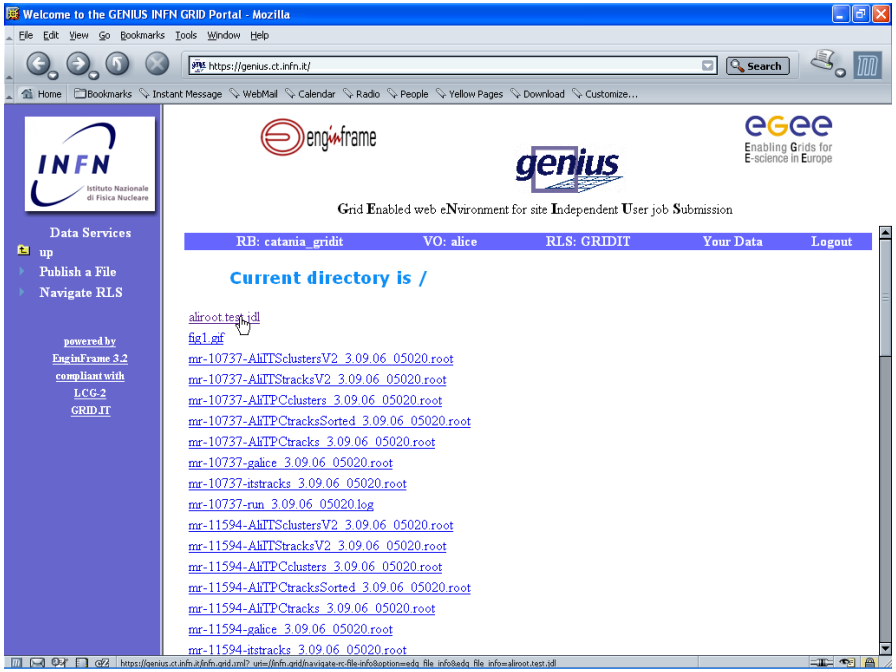
GENIUS data spooler (1/2)

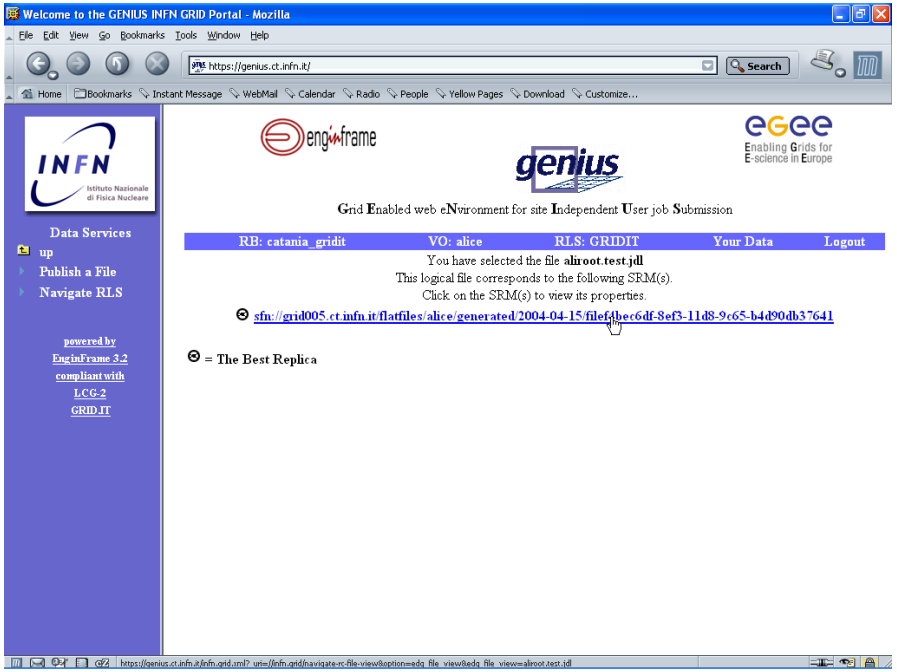


GENIUS data spooler (2/2)

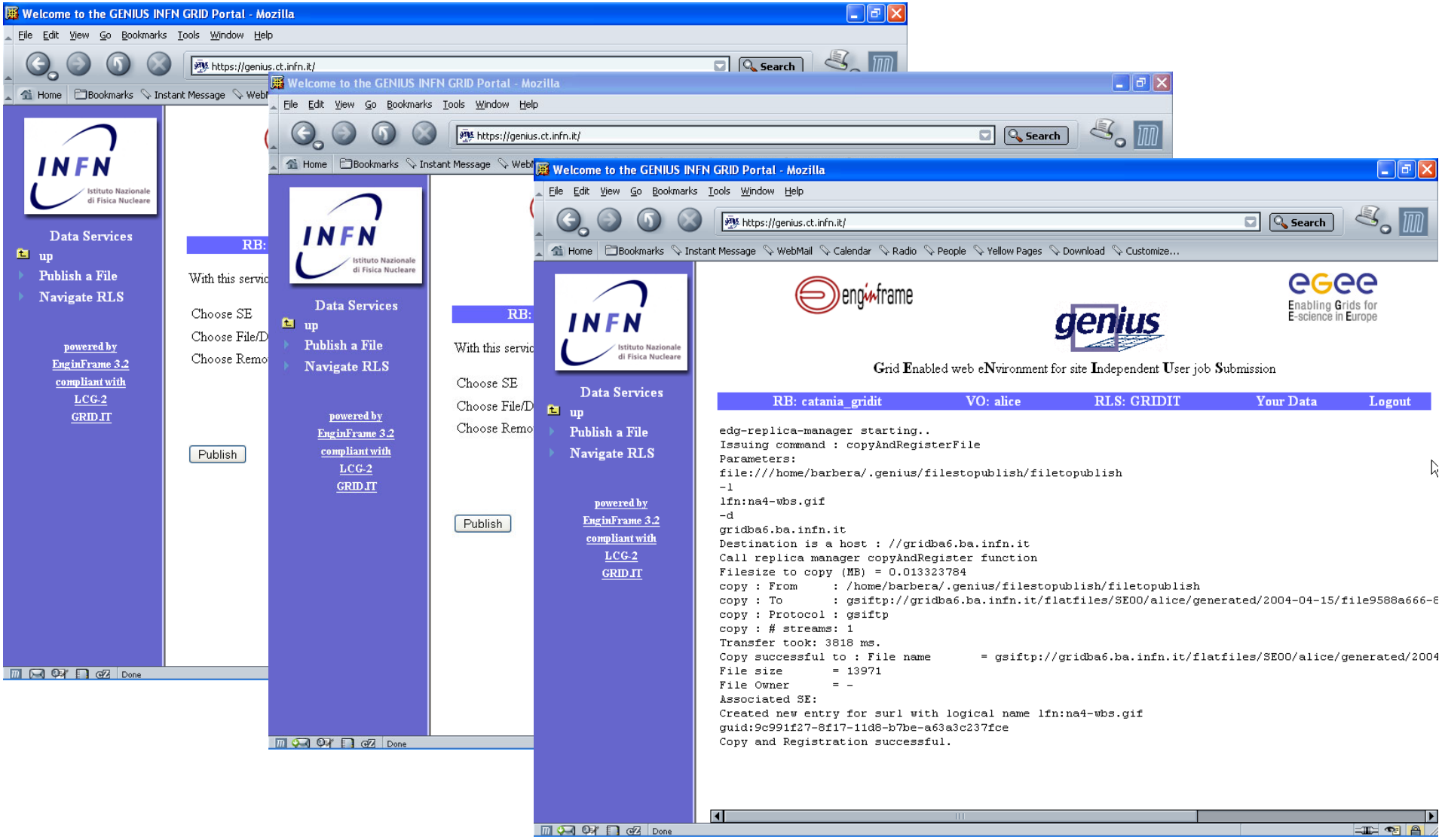


GENIUS RLS browsing

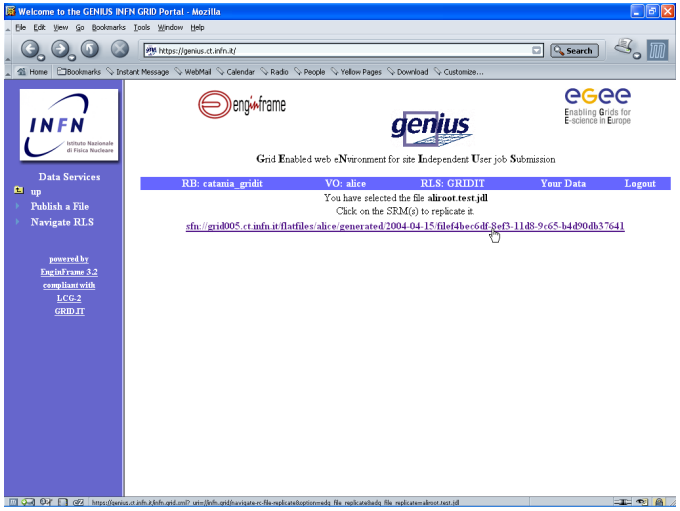


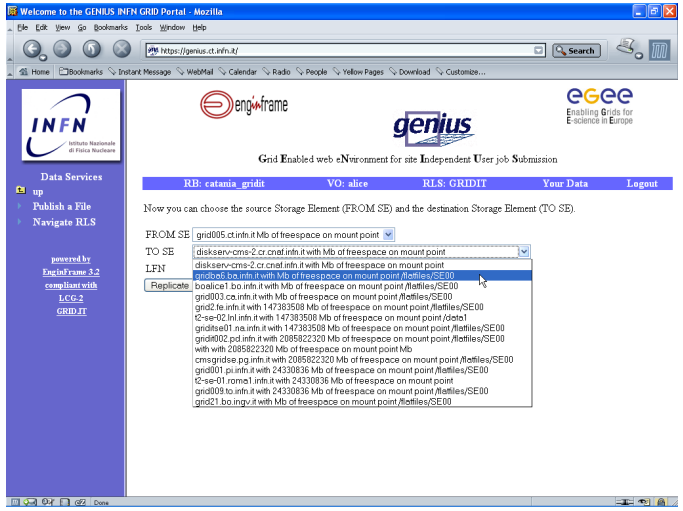


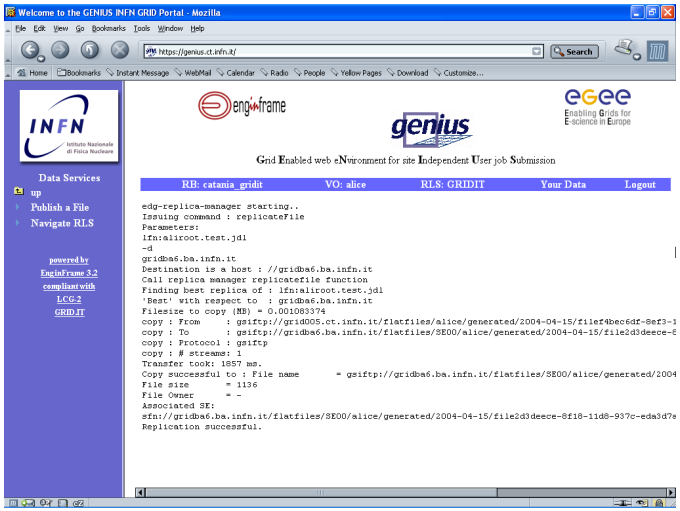
GENIUS data publication

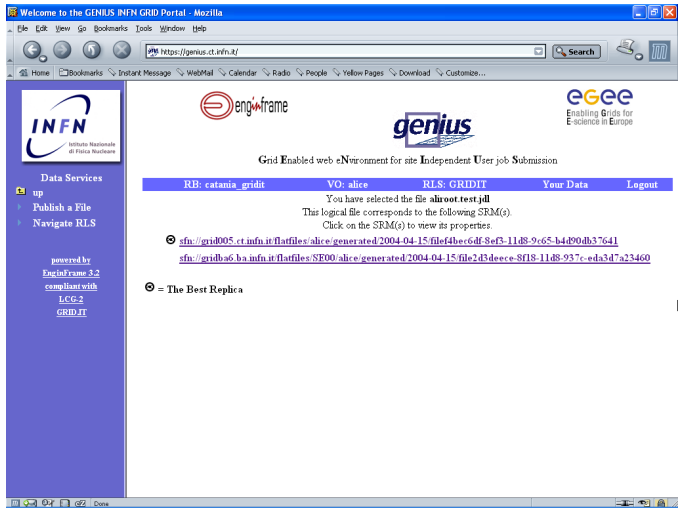


GENIUS data replication

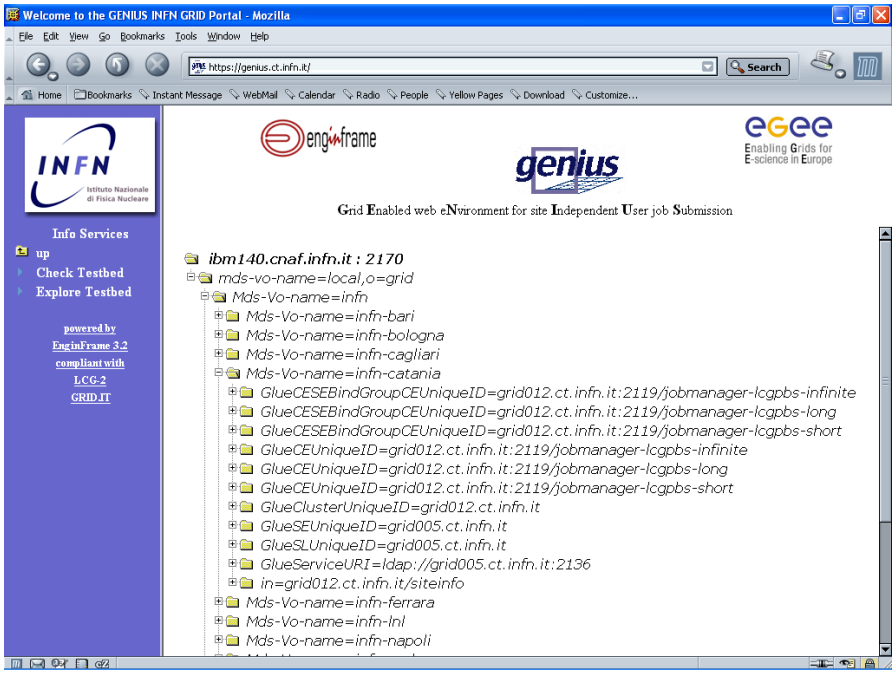


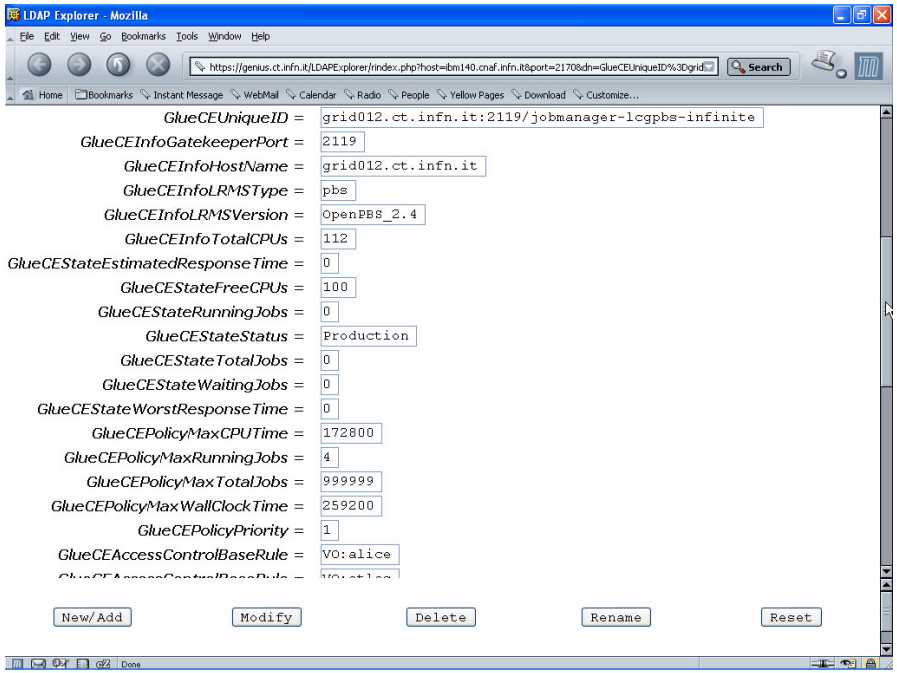




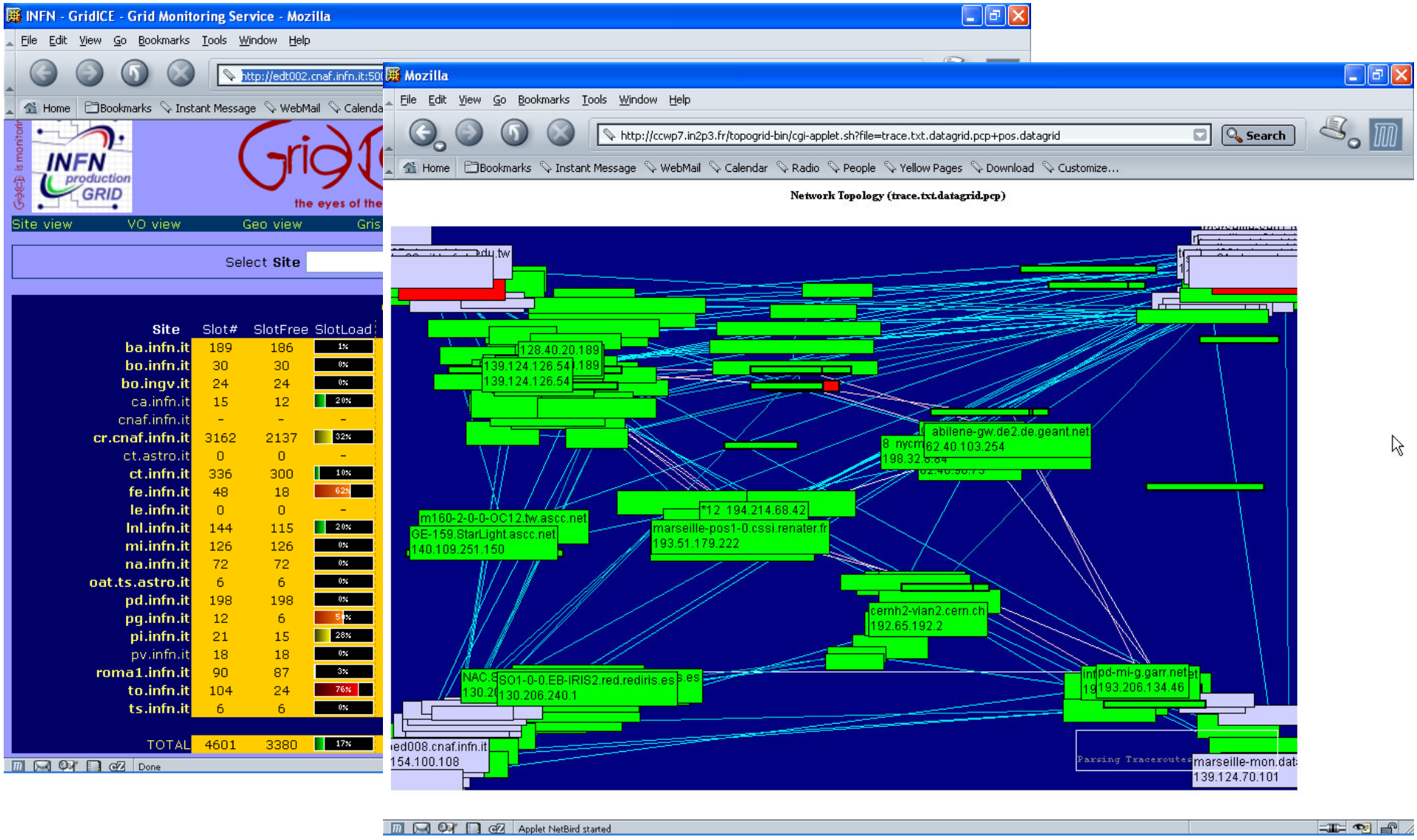


GENIUS information services

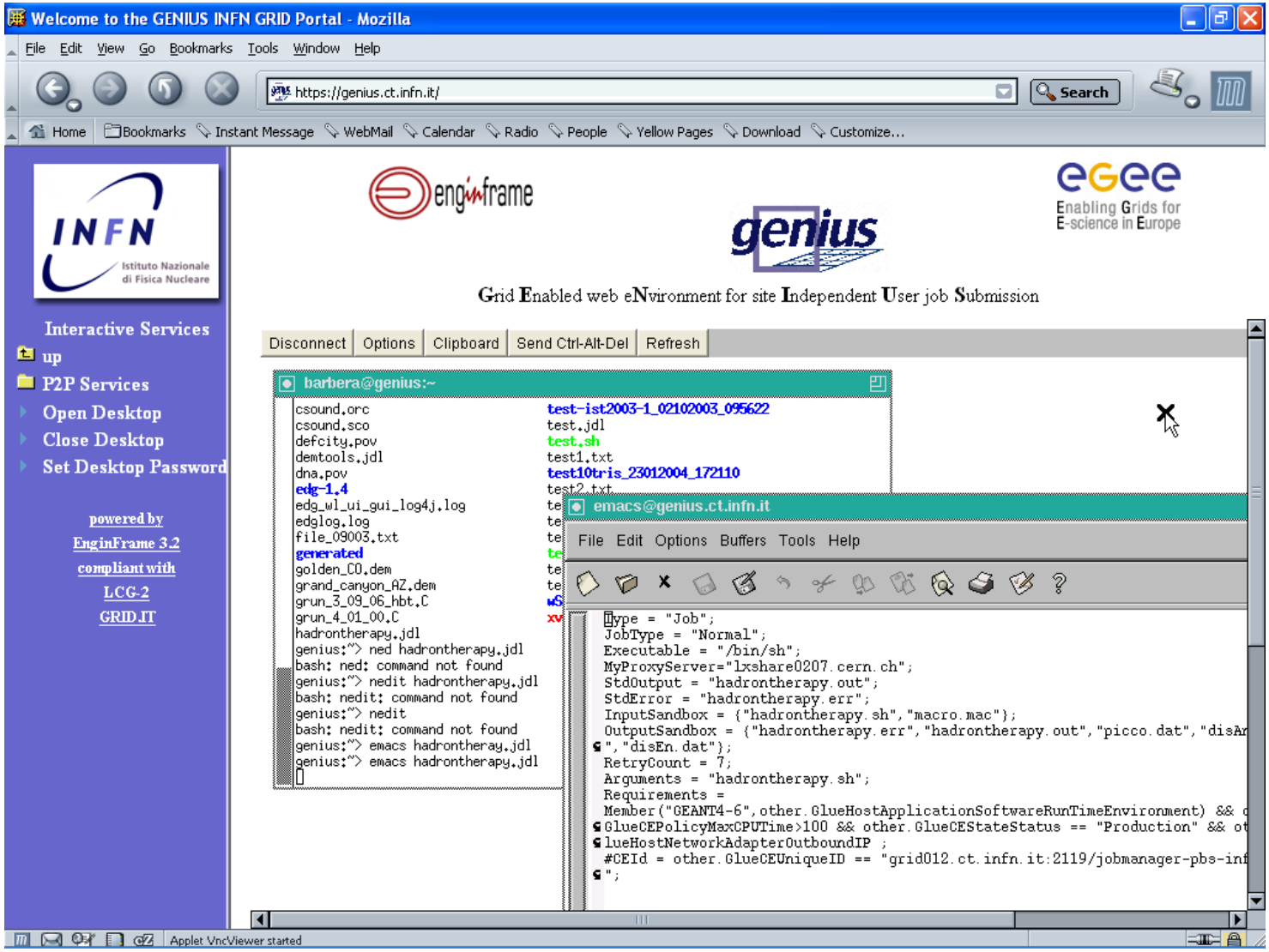




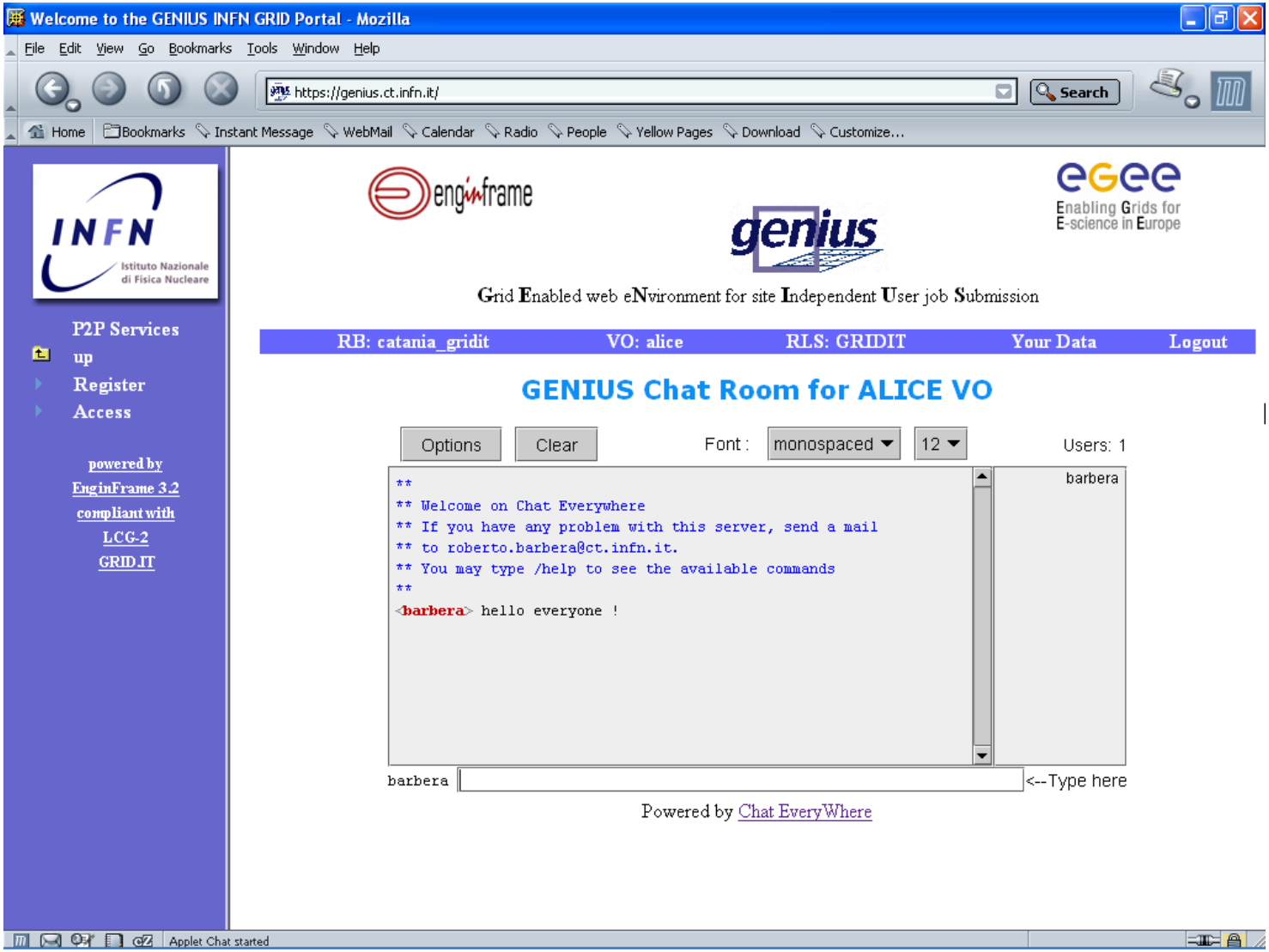
GENIUS monitoring services



GENIUS interactive services



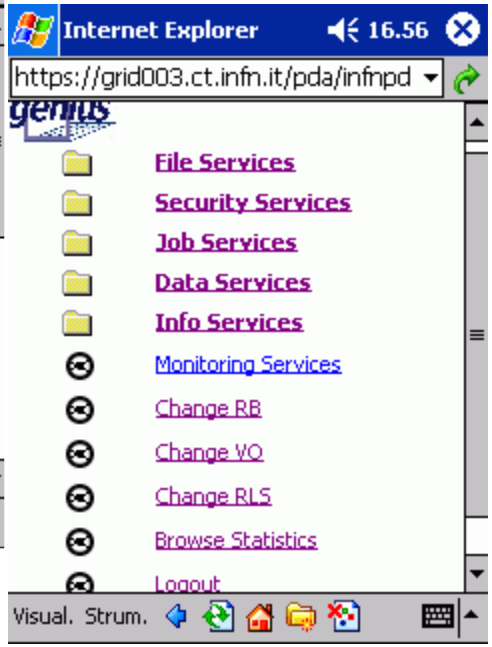
GENIUS P2P services



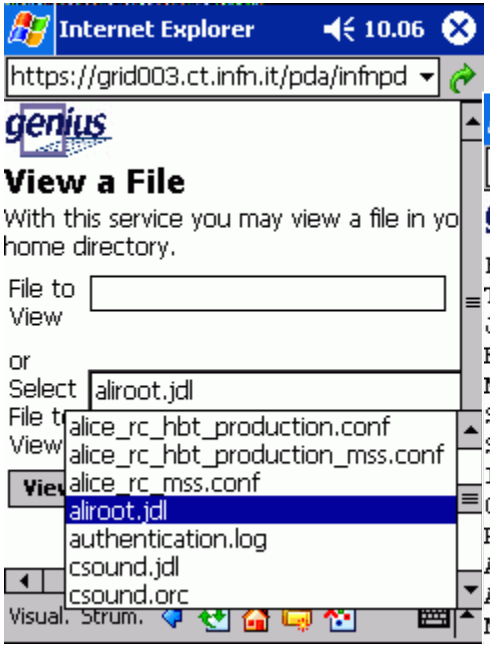
GENIUS for PDA (1/2)

Home Page

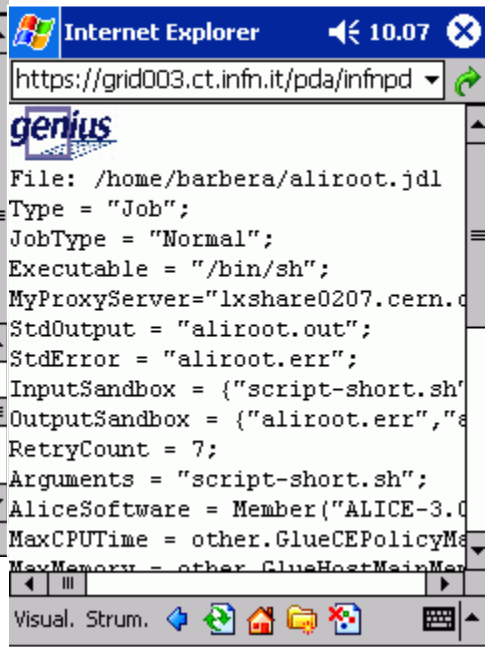
Services



Remote file browsing

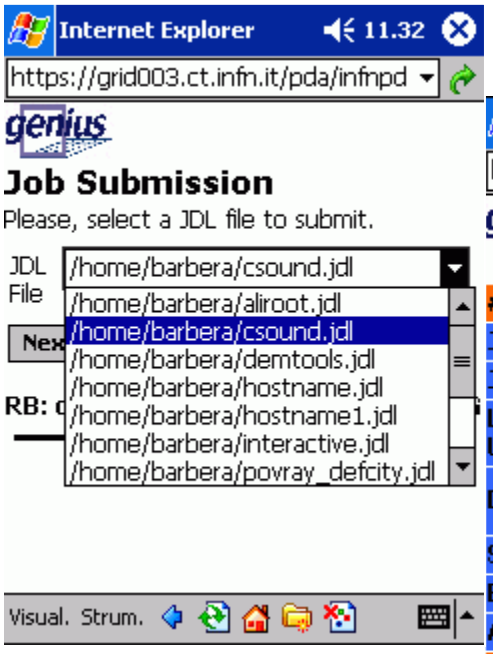


Remote file inspection

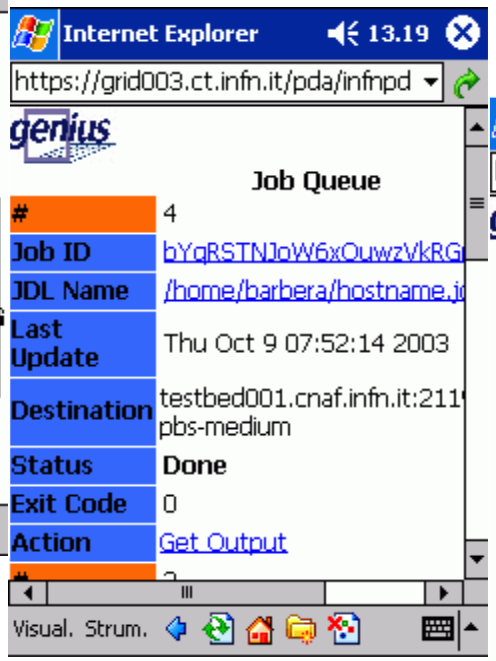


GENIUS for PDA (2/2)

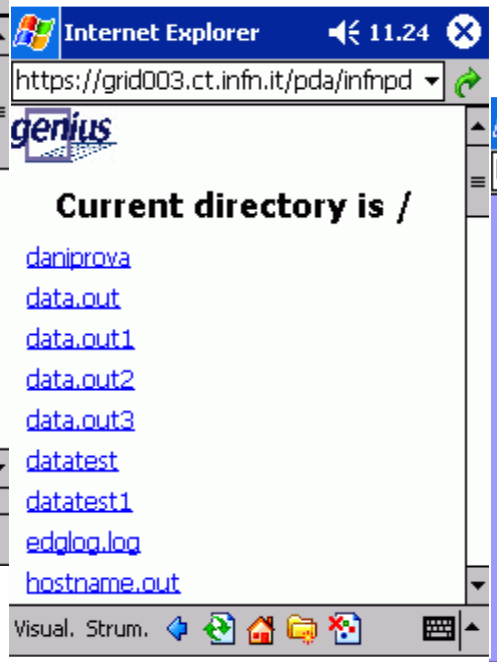
Job submission



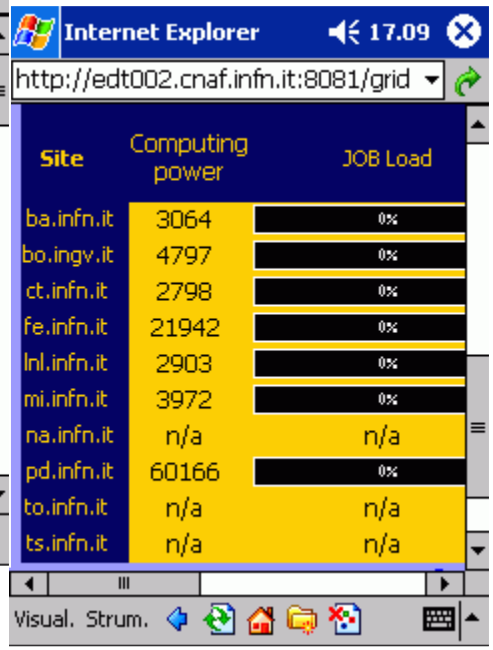
Job monitoring



RC browsing



GridICE integration



GENIUS for cell phone

Summary and conclusions

Let’s go to the practical