# Advances in Interoperability between ENEA-GRID and gLitebased Infrastructures


## Summary

The GRID approach has allowed to integrate in a single unified system, namely ENEA-GRID, all the high performance computational resources available inside ENEA. The main software components of the ENEA-GRID infrastructure are the distributed file system OpenAFS and the multi-site resource manager LSF Multicluster, which constitute the production ready ENEA-GRID middleware.

In the participation of ENEA in national and European GRID projects, solutions had to be found to enable the interoperability between ENEA-GRID and other GRID middleware. Among these gLite is one of the leading solutions, originated in the EGEE/EGEE-II projects and adopted in other contexts.

The poster presents the ongoing activity to make ENEA-GRID interoperable with the EGEE grid infrastructure. As the two grids employ different mechanisms for authentication, scheduling and data sharing, their integration is not straightforward.

However, the ability for ENEA-GRID to receive jobs from gLite infrastructures has already been established in previous works. The ENEA EGEE site is in production, using the so called SPAGO technology (Shared Proxy Approach for Grid Objects).

Now we have extended the ENEA-GRID/gLite interoperability mechanism described above, by adding the ability for ENEA-GRID users to submit jobs to gLite-based grids in a transparent way.

The same solutions have also been adopted in implementing the interoperability with another gLite based infrastructure, namely the GRID connecting the computational resources of four GRID projects in Southern Italy (ENEA CRESCO, SCOPE, CYBERSAR and PI2S2). CRESCO computation resource are fully integrated into ENEA-GRID and the solutions found for EGEE have been straightforwardly applied, in a context where each of the 4 projects retains the capability to run all the main GRID services.


Two EGEE technical notes have been prepared to document the gateway implementation:


EGEE Technical Note EGEE-TR-2007-001
**"The gateway approach providing EGEE/gLite access to non-standard architectures"** Bracco, G; Migliori, S; Sciò, C. ; Santoro , A.;
http://doc.cern.ch//archive/electronic/egee/tr/egee-tr-2007-001.pdf

EGEE Technical Note EGEE-TR-2006-006
**"AFS Pool Account Users - GSSKLOG and LCMAPS extension to support AFS users as EGEE pool account users"**
Bracco, G; Giammarino, L; Migliori, S; Sciò, C.;
http://doc.cern.ch//archive/electronic/egee/tr/egee-tr-2006-006.pdf


## ENEA

[Italian National Agency for New Technologies, Energy and Environment]

12 Research sites and a Central Computer and Network Service (ENEA-INFO) with 6 computer centres managing multi-platform resources for serial & parallel computation and graphical post processing.


Computational resources:

- Hardware / OS:
  - IBM SP - AIX;
  - ia64/x86/x86_64 Linux;
  - SGI Altix & Onyx;
  - Apple cluster;
  - Intel Windows servers.
- software:
  - commercial codes fluent, ansys, abaqus..;
  - elaboration environments Matlab, IDL..;

## ENEA GRID architecture

ENEA GRID mission [started 1999]:

- provide a unified user environment and an homogeneous access method for all ENEA researchers, irrespective of their location.
- optimize the utilization of the available etherogeneous resources

GRID functionalities (unique authentication, authorization, resource access and resource discovery) are provided using "mature", multi-platform components:

Distributed File System: OpenAFS

Resource Manager: LSF Multicluster [www.platform.com]

Unified user interface: Java & Citrix Technologies

These components constitute the ENEA-GRID Middleware.

OpenAFS

- user homes, software and data distribution
- integration with LSF
- user authentication/authorization, Kerberos V

## Interoperability gLite vs ENEA-GRID

The gLite software is the GRID middleware developed in the context of the EGEE project. Its authentication scheme is based on X509 certificates as an authentication mechanism, its data sharing is enabled through the existence of Storage Elements visible to the whole GRID, and the scheduling is carried out by a software element known as Workload Management System (WMS). Conversely, ENEA-GRID employs a middleware based on the geographically distributed filesystem OpenAFS for data sharing, the resource manager LSF Multicluster for scheduling, and a combination of Kerberos 5 and AFS tokens as authentication mechanism. Interoperability between such different systems consists of two different parts. Submitting jobs from gLite to ENEA-GRID and submitting jobs from ENEA-GRID to gLite.

### gLite vs ENEA-GRID

**Job submission in gLite infrastructure:**

- edg-job-submit <file.jdl>
  - ➔**<file.jdl>** is an ASCII format file containing the information about the executable file and the files that must be transferred on the WN to allow a proper job execution.

**Job submission to ENEA-GRID:**

- bsub -q <queuename> <file.exe>
  - ➔**<queuename>** is the name of the specific queue that must execute the job
  - ➔**<file.exe>** is the executable file. Note that all the other required files are already exported to each machines in ENEA-GRID by AFS, thus they not need to be notified to bsub.

**Note:** gLite users need to notify into the jdl all the input/output files required by the job; conversely ENEA-GRID users have no such requirement, which might lead to inconsistent communication between the two middlewares. See the issue below: "Transparent File sharing".

### gLite to ENEA-GRID: The SPAGO approach

The basic design principle of the SPAGO approach, that allows ENEA-GRID to process jobs submitted to gLite, is outlined in Figure 1 and it exploits the presence of AFS shared file system. When the CE receives a job from the WMS, the gLite software on the CE employs LSF to schedule jobs for the various Worker Nodes, as in the standard gLite architecture.

However the worker node is not capable to run the gLite software that recovers the InputSandbox. To solve this problem the LSF configuration has been modified so that **any attempt to execute gLite software on a Worker Node actually executes the command on a specific machine, labeled Proxy Worker Node** which is able to run standard gLite.

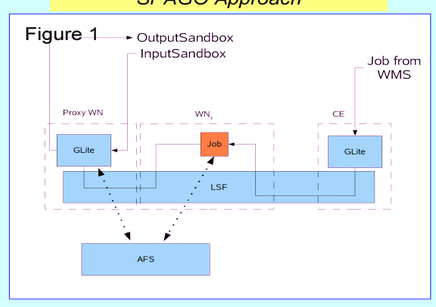
By redirecting the gLite command to the Proxy WN, the command is executed, and the InputSandbox is downloaded into the working directory of the Proxy WN.

The working directory of each grid user is maintained into AFS, and is shared among all the Worker Nodes and the Proxy WN, thus downloading a file into the working directory of the Proxy WN makes it available to all the other Worker Nodes as well. Now the job on the $WN_1$ can run since its InputSandbox has been correctly downloaded into its working directory. When the job generates output files the OutputSandbox is sent back to the WMS storage by using the same method.

In the above architecture, the Proxy WN may become a bottleneck since its task is to perform requests coming from many Worker Nodes. In that case **a pool of Proxy WN** can be allocated to distribute the load equally among them.

### gLite to ENEA-GRID interface
*SPAGO Approach*

Figure 1

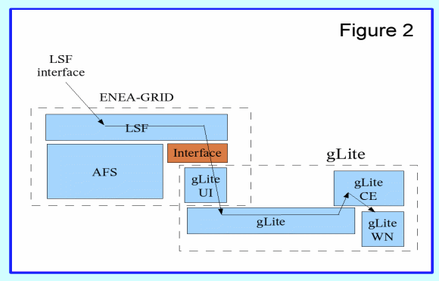



### ENEA-GRID to gLite

Job submission from gLite to ENEA-GRID took advantage of the fact that gLite CE employs LSF Multiclaster as one of its resource managers. Therefore slight modifications in the configuration of LSF allows seamless interfacing between the gLite CE and the underlying ENEA-GRID infrastructure.
On the other hand LSF multicluster does not have embedded supports to interface with gLite middleware, which leads to a more complex approach. The overall design approach, shown in Figure 2, is the following: an ENEA-GRID user who wants to submit a job to gLite, submits its request to LSF (e.g. using "bsub" command), as he would do for any ENEA-GRID job, but specifies a specific "gLite-interface" queue for the job. In its turn the LSF queue redirects the job towards a special software module that generates a proper jdl file and forwards the job and its jdl to a gLite User Interface. From there it is responsibility of the gLite middleware to send the job to the appropriate Computing Element and report to the interface software module when the job is completed.

### ENEA-GRID to gLite interface

Figure 2




### Issues under investigation

The ENEA-GRID to gLite interface presents still two issues to be solved in order to have a full ENEA-GRID/gLite interoperability:

- **Transparent Authentication:** In order to be able to use the gLite infrastructure the user is expected to have a personal X509 certificate released by the proper certification authority. This is a requirement of the EGEE project, and unavoidable. However, once the user has installed correctly his personal certificate on his machine he should be able to access the whole gLite infrastructure through the ENEA-GRID interface described above. Currently this is not the case, since the user must issue a command "setup-egee-proxy.sh" (analogous to the voms-proxy-init in gLite) to generate a proxy certificate on the gLite User Interface. Since ENEA-GRID employs its own credentials, we are studying a mechanism that may automatically translate the kerberos tickets used by ENEA-GRID into a X509 proxy certificate, thus providing transparent authentication.

- **Transparent File Sharing:** In the ENEA-GRID infrastructure all the machines share the same filesystem due to the use of OpenAFS. This means that there is no need to specify the files required by the jobs to run correctly, since the files are shared among all the machines that might run the job. On the other hand the gLite infrastructure requires to identify into the jdl file all the files required by the job. Our current approach to this problem consists in asking the user to identify such files into the bsub invocation, which would be intercepted by our interface software and a proper jdl file containing the files would be generated. However, the fact that the user must be aware of the files needed by the submitted job means that the job submission process from ENEA-GRID to gLite is not completely transparent. We are currently investigating a more transparent submission mechanisms that would allow even jobs on gLite WN to transparently import the needed files from AFS.

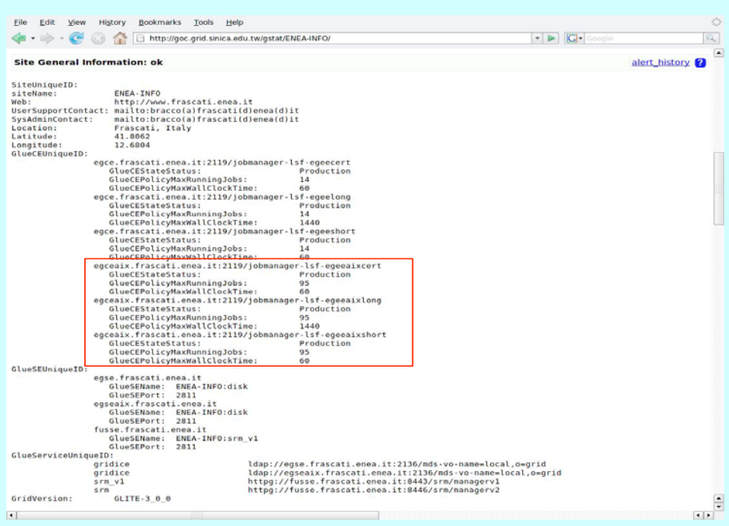
### Interoperabilty of GRID projects

In the context of the PON1575 project, there has been a strong focus on integrating the four GRID projects in Southern Italy (ENEA CRESCO, SCOPE, CYBERSAR and PI2S2) into a single, interoperable computational GRID. The platform of choice is the gLite middleware.

CRESCO computation resources are fully integrated into ENEA-GRID and therefore we have been able to apply the solution described above to make CRESCO project interoperable with the other three italian projects. Moreover, each of the 4 projects retains the capability to run all the main GRID services autonomously, so that a failure on one project will not disable the operations on others.

### GOC/GSTAT page with ENEA-GRID WN information

The ENEA-INFO site has been certified for the production grid service providing access both to linux and AIX Worker Nodes.




### CRESCO HPC Centre
**www.cresco.enea.it**

**CRESCO** (Computational Research Center for Complex Systems) is an ENEA Project, co-funded by the Italian Ministry of University and Research (MUR). The project will be functionally built around a HPC platform and 3 scientific thematic laboratories:

- the Computing Science Laboratory, hosting activities on HW and SW design, GRID technology and HPC platform management
- **The HPC system (installation 1Q 2008) will consist of a ~2500 cores (x86_64) resource (~25 Tflops peak), InfiniBand connected with a 120 TB storage area. The resource, part of ENEA-GRID, will be made available to EGEE GRID using gLite middle-ware through the gateway approach**


**A. Santoro, C. Scio*, A. Rocchi, G. Bracco, S. Migliori, A. Quintiliani, S. Podda**
**ENEA-FIM, ENEA C.R. Frascati, 00044 Frascati (Roma) Italy, (*) Esse3Esse**