The GENIUS Grid Portal

R. Barbera[*]
Dipartimento di Fisica e Astronomia, INFN, and ALICE Collaboration, via S. Sofia 64, I-95123 Catania, ITALY
A. Falzone, A. Rodolico
NICE srl, via Serra 33, I-14020 Camerano Casasco (AT), ITALY

The architecture and the current implementation of the GENIUS (Grid Enabled web eNvironment for site Independent User job Submission) Grid portal, jointly developed by INFN and NICE srl within the context of INFN Grid and DataGrid projects, is presented and discussed. Particular care is devoted to the description of job creation and submission and the services for transparent access to user's data and applications. The results of the integration of the portal with the Grid tools of some Large Hadron Collider Experiments is also shown.

1. INTRODUCTION

Grid computing vision brings the fascinating and revolutionary idea that enormous and heterogeneous computing resources and mass storage systems, distributed world wide and "linked" by large bandwidth networks, can be connected by an intelligent series of services, the "Grid middle-ware", and behave as a "single" machine "disintegrated" into a series of special purpose appliances (seen as virtual services) like computing elements, storage elements, meta-data catalogues, resource brokers, information indexes, etc.

Within this paradigm, an easy access to large geographically distributed computing and data management resources is therefore provided to large mission oriented communities, often referred to as Virtual Organizations (VO) [1]. Computational and data Grids are, hence, considered as a possible implementation of the concept of *co(l)laboratory* [2] where scientists and researchers work together to solve complex multi/inter-disciplinary problems, despite their geographic and organizational boundaries.

In order to succeed as the tomorrow's way to do world-wide computing, however, Grids must provide a secure, uniform (collaborative), pervasive and ubiquitous access to distributed, high-end computational resources, services and/or applications. Scientific domain knowledge and tools must be presented to the (non-expert) users in terms of the application science and not in terms of complex computing protocols. They should just *see* the Grid as a seamless extension of their own workstation for what concerns both job execution/monitoring and data access/management.

As a example of solution of the above problems, we describe here the architecture and the current implementation of the GENIUS Grid portal [3], a problem solving environment, based on the European Union DataGrid Project (EDG) [4] middle-ware, that allows scientists to access, execute and monitor their own applications that exploit Grid resources only using a conventional web browser.

The paper is organized as follows. An overview of the current DataGrid middle-ware services, through a typical job work-flow, is given in Section 2. Section 3 is dedicated to the description of the architecture and functionality of the portal, while Section 4 describes the integration into GENIUS of the Grid tools of some Large Hadron Collider Experiments and other applications. Summary and conclusions are drawn in Section 5.

2. THE DATAGRID SERVICES

The present version of the Grid middle-ware released within the DataGrid Project contains all the services needed to accomplish the following tasks:

1. user authentication and authorization
2. job submission and logging
3. data management (file publication, replication and deletion)
4. Grid information building and monitoring
5. computer fabric installation and management
6. interaction with hierarchical Mass Storage Systems (MSS)
7. network monitoring

The representation of the DataGrid middle-ware, in the usual multi-layered "hourglass" schema, is shown in fig. 1. The DataGrid services rely on the middleware provided by the Globus Toolkit [5], a *de facto* standard in Grid computing, but extend its functionalities making easier the interface with the application software.

The coordinated but complex interplay among the various services summarized above can be seen looking at a typical job work-flow (see fig. 2). The user

---

[*] Speaker and corresponding author. Email: roberto.barbera@ct.infn.it.

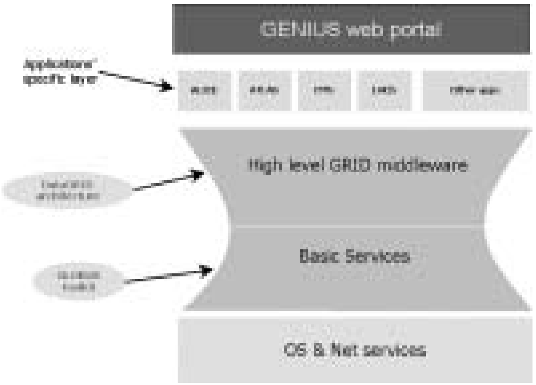

Figure 1: Multi-layered structure of the DataGrid middle-ware including the GENIUS portal.

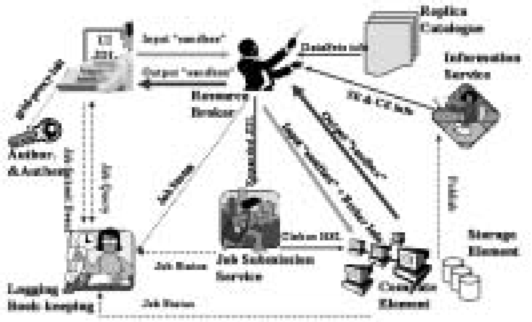

Figure 2: Job work-flow using the DataGrid services.

logs into the Grid from an User Interface (UI) machine using his digital personal certificate (upper left part of the figure). Then, he can submit a job to a Resource Broker (upper central part of the figure), specifying the executable to be used, the ensemble of input files (the InputSandbox), and the requirements for the job. The Resource Broker queries the Grid Information System (right part of the figure), to get the actual status of the resources, and the Replica Catalog (upper right part of the figure), to determine the physical location(s) of any external file(s) needed by the job and not included into the InputSandbox, and performs the best matching identifying the "best" Computing Element/Storage Element (lower right part of the figure) where the job can run. The job is then shipped to that Computing Element (lower central part of the figure) with all the relevant brokering information and executed on one of the Worker Nodes attached to that Computing Element. The status of the job is continuously updated by the Logging & Bookkeeping System (lower left part of the figure) that can be queried at all times by the user. When the jobs ends, the ensemble of output files specified by the user in the OutputSandbox are transferred back to the Resource Broker to be served, on demand, to the user while those permanently saved on the Storage Element are registered on the Replica Catalog.

At all times, both before and after the job execution, files can also be replicated between two Storage Elements, updating the information in the Replica Catalog.

3. THE PORTAL

All the services provided by the DataGrid middleware are currently exposed to the users through rather complex Command Line Interfaces (CLI). There are tens of different commands with many options and rigid sequences and the description of jobs has even a dedicated language (the Job Description Language or JDL) to be learned. All this has the negative effect of discouraging many potential users from learning how to profit from the Grid advantage.

It is just to overcome these drawbacks, exploiting the successful experiences made since the year 2000 within the Alice Experiment at the CERN Hadron Collider [6], that, at the beginning of 2002, the Italian INFN Grid Project started the GENIUS Portal Project in conjunction with the Italian web company NICE srl.

3.1. Architecture and implementation

The present implementation of GENIUS (see fig. 1) is carried out on top of the middle-ware services. The layout of the portal can be described by a three-tier model (see fig. 3):

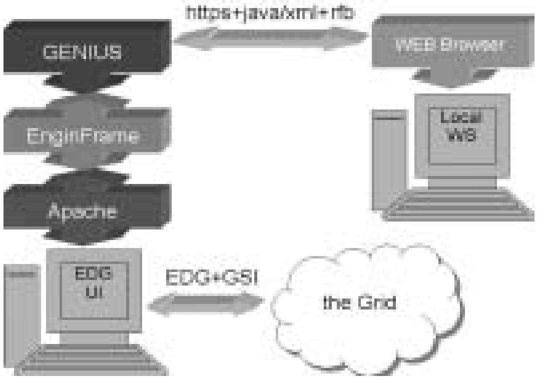

Figure 3: Three tier architecture of the GENIUS portal.

1. the client (top right in the figure): the user's workstation running a web browser;

2. the server (left side of the figure): an EDG User Interface machine (equipped with the DataGrid middle-ware services able to submit jobs and manage data on the Grid) which is running the

Apache Web Server, the Java/XML framework EnginFrame [7], developed by NICE srl, and GENIUS itself;

3. the remote resources (bottom right in the figure): the Grid.

Using the EnginFrame services the user can interact with files on the UI and, from there, the user can send jobs to the Grid and manage the data belonging to the given Virtual Organization. The use of the web interface eliminates any problem connected to the need of a particular Operating System and/or middle-ware running on the client, and to the locations themselves of the client and the server: the user can interact with the Grid from everywhere and with "everything" (workstation, PDA, WAP phone, etc.).

3.2. Security services

In order to guarantee a secure access to the Grid, GENIUS has been implemented with a multi-layered security infrastructure.

All web transactions are executed under the Secure Socket Layer (SSL) via HTTPS.

The user has to have an account on the User Interface machine where GENIUS is running: either it is his personal workstation or the system manager of a trusted Grid testbed site has to know him;

When the user wants to interact with the file-system of the UI, performing operations with his files, he gets prompted for the username and password of the account on that machine.

Grid authentication in GENIUS is currently based on the concept of user credentials delegation implemented with MyProxy [8] (see fig. 4). From a "trusted" machine (his own machine or an EDG UI installed at his laboratory/institute; upper left part of the figure), the user can, using the private key of his X.509 personal certificate, create a temporary certificate issuer (a *myproxy*) and store it, for a defined amount of time, on a MyProxy Server (upper right part of the fig. 4). When the user (lower left part of the fig. 4) asks the GENIUS portal to invoke a Grid service, the portal (lower central part of the fig. 4) contacts the MyProxy Server and tries to get a temporary proxy certificate on the user's behalf from the certificate issuer. If the life time of the *myproxy* is not expired, temporary user credentials are retrieved, the action is performed and the output is rendered back into the user's web browser. So, the three-tier architecture of MyProxy's credentials delegation avoids the private key of the user's certificate to be sent over the network, avoiding a security hole.

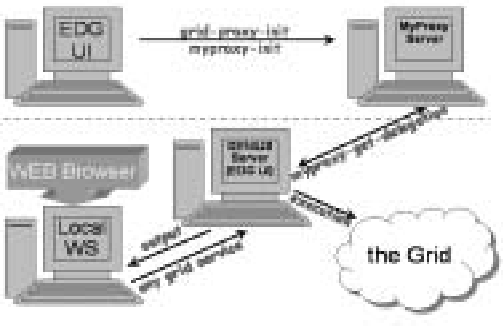

Figure 4: GENIUS Grid authentication schema with MyProxy.

When the user wants to use specific GENIUS services relative to a given Virtual Organization, the subject of his X.509 personal digital certificate is verified against the one stored in the Users' Server of the corresponding VO and only if the check is successful the user can invoke the specific VO services.

In order to increase the security as much as possible, usernames and passwords typed on the browser by the user are not saved anywhere. They are streamed under the `HTTPS` protocol and then destroyed. Furthermore, GENIUS logs automatically out all clients which have been left unattended (no action on the web browser) for more than 30 minutes in order to avoid unwanted accesses from non-authorized users who could temporary "steal" somebody else's work station.

Once the user gets authenticated and authorized by GENIUS, he can use all services of the portal implemented so far (more than one hundred). For obvious space reasons we are not going to describe all of them here (all GENIUS services are described in the User's Manual that can be browsed/downloaded from the official site [3]) but we will concentrate only on job creation and submission and on the transparent access to user's data.

3.3. Job submission services

Using GENIUS job submission services, the user can seamlessly create a job and run it on the Grid. In particular, it is possible to:

1. list the available computing elements (sites/queues) where a given job can run on;
2. create a job and submit it to a Resource Broker (which can be selected by the user) or to a particular computing element;
3. monitor the current status of a job;
4. inspect on the fly the output of a job and save it both on the User Interface machine and on the local workstation.

Figure 5 shows the graphic job creator, developed by DATAMAT [9] and embedded into the portal as an applet in collaboration with NICE srl. Using this tool, users do not have to learn the Job Description Language to create their jobs, having at "a mouse click" distance all the powerful elements of the language. Filling the appropriate fields of the applet, they can describe their jobs in terms of the applications (executables) which have to be run on the Grid and of the requirements on these applications (type, version number, dependencies, etc.) and on the underlying Grid services (number of available CPU's, aggregated CPU power, network bandwidth, etc.).Jobs can directly be submitted to the Resource Broker from within the applet or can be saved on the UI machine to run afterward.

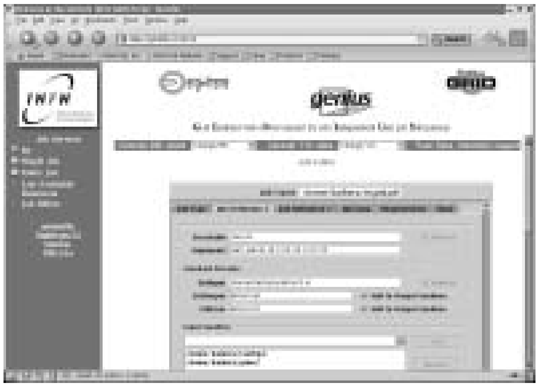

Figure 5: The DATAMAT job editor applet embedded in GENIUS.

Both single and composite-jobs can be launched on the Grid. The composite-job submission service of GENIUS, which is not available in the relative Command Line Interface, is shown in fig. 6.

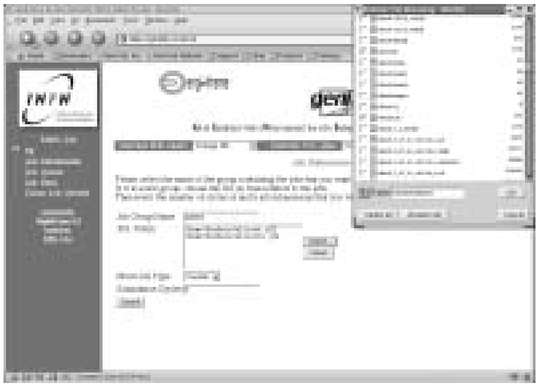

Figure 6: Creation of a composite-job with GENIUS.

We define here the concept of job group which, in the most general case, is made of N jobs (each described by a different JDL file). A job group can then go through M cycles of parallel submissions. Parallel in this respect means that all jobs are submitted at the same time to the Resource Broker and there is no dependence among them. Sequential submission of the jobs of a group is foreseen for a future version of the portal. Files containing the descriptions of the various jobs belonging to a given group can be graphically chosen browsing the user's directory on the User Interface machine (upper right part of the fig. 6).

At the time of submission, the user can decide either to let the Resource Broker choose the best computing element or select a specific one (see fig. 7).

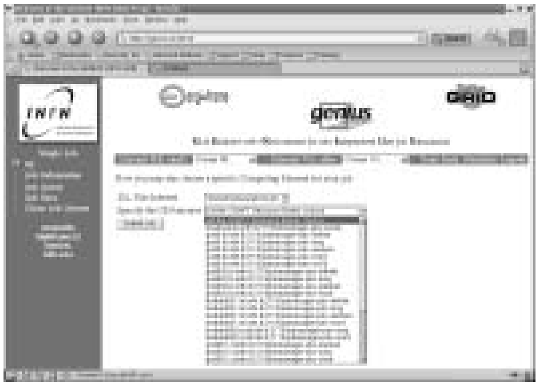

Figure 7: Selection of a particular computing element where to submit a job.

The status of both single and composite-jobs can be followed using the web browser and the output files can be both browsed in real time from within the browser or saved on the user's work station (see fig. 8).

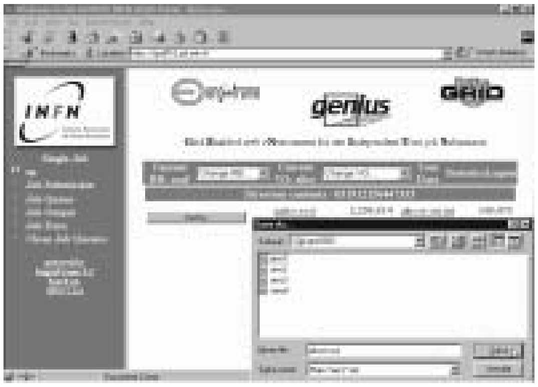

Figure 8: Browsing/downloading of the user's job output files in GENIUS.

3.4. Data management services

The EDG data management services aim at handling large amounts of data across the distributed environment of the Grid. The current paradigm in DataGrid is that only read-only data files are managed. Files may be updated but its replicas will not be automatically updated yet, such operations being available in a future release. The main components of the DataGrid Data Management System, currently provided and used by GENIUS are:

- Replica Manager: manage the creation, replication and deletion of data files in the Grid. Files can be imported from external sources, copied across Grid Storage Elements and removed from the Grid through the Replica Manager interface.
- Replica Catalog: all files managed by the Replica Manager are registered in the Replica Catalog. The Logical File Names provided by the Grid users are resolved into actual physical locations on the Grid. The Globus Replica Catalog package is deployed in EDG.
- GDMP: the GRID Data Mirroring Package is used to automatically mirror file replicas from one Storage Element to a set of other subscribed sites. It is also currently used as a prototype of the general Replica Manager service. GDMP also provides an interface to the Replica Catalog that eases its use and is accessed through GENIUS.

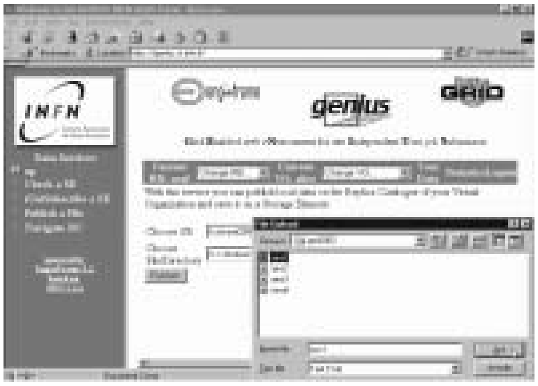

Figure 9: Uploading of a user's file onto a storage element and registration in the Replica Catalog.

The EDG Replica Manager allows users and running jobs to access and maintain consistent copies (replicas) of files across many Grid sites. The user will only see atomic operations, but the Replica Manager will need to update the Replica Catalog and invoke the file transfer, dealing with errors and failures in a robust manner.

The following scenarios will be described below:

1. the user may have files on his workstation (the client) that he needs to use as input data for the jobs or he just wants to share with the colleagues belonging to the same Virtual Organization;
2. the user wants to access from/download onto his workstation the data, shared within the Virtual Organization he belongs to, which are published on the Replica Catalog;

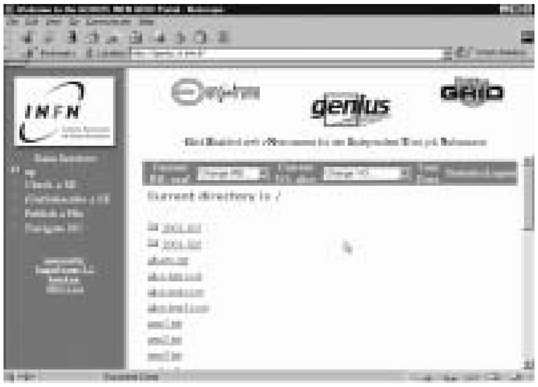

Figure 10: Browsing of a Replica Catalog.

3. the user wants to access from/download onto his workstation the output of the jobs which have ran on the Grid and have been permanently saved on the user's account (on the server);
4. the user wants to copy (replicate) a file registered on the Replica Catalog between two storage elements.

Figures 9-10 show how cases 1 and 2 are handled by GENIUS.

The user can choose the Virtual Organization he belongs to and upload the file(s) (sited on the local workstation) on a Grid storage element and publish the corresponding logical file name(s) into the VO Replica Catalog. The catalog can then be navigated from the web browser as it was a local file system. Files can be inspected, replicated from one storage element to another, removed, and even downloaded onto the user workstation. Figure 8 shows the handling of case 3. The user can inspect the output of a given job from the web browser (simply clicking on the names of the output files) and even copy them locally on the local workstation. The user need relative to case 4 is satisfied by the GENIUS file replication service, shown in fig. 11.

It is worth noting that the list of possible "destination" storage elements is dynamically retrieved on demand from the Grid information service the currently used Resource Broker points to.

3.5. Interactive services

Transparent job and data management on the Grid are, however, just a part of the day-by-day work of a scientist in a collaboratory. In fact, researchers also want to interactively analyze with their own applications the output of their jobs which have been running on the Grid and have been permanently saved on their

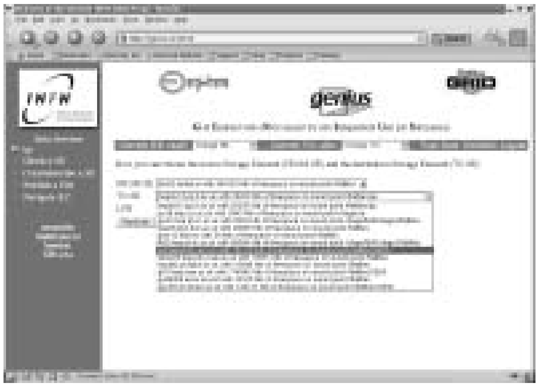

Figure 11: The GENIUS file replication service.

accounts (on the UI machine). On this purpose, the TightVNC tool [10] has been integrated into GENIUS so that, using its Remote Frame Buffer protocol and an SSL tunneled HTTP access, users can open their graphic desktops on the User Interface machine and do interactive analysis (of data produced by their jobs which have run on the Grid) from the web browser. An example is shown in fig. 12.

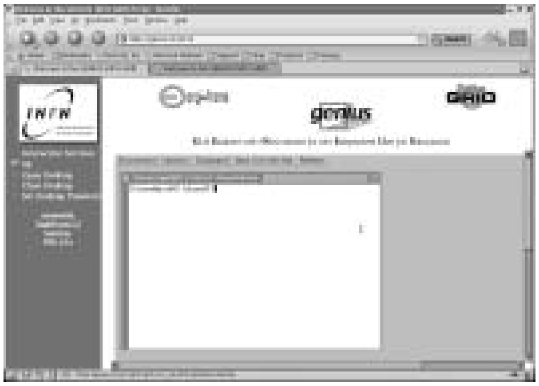

Figure 12: An example of a virtual desktop embedded into GENIUS and securely accessible through the portal interactive services.

4. THE APPLICATIONS

Up to now, several installations of GENIUS have been made in Grid testbed sites in Europe and worldwide and much work has been made to start integrating the Grid environments of some applications into the portal. In this section, a short overview of the use of the portal made by some Virtual Organizations is carried out.

4.1. ALICE

The ideas which led to the design of GENIUS and to its implementation over the EnginFrame Grid portal framework originated few years ago inside the ALICE Experiment [11]. GENIUS has been extensively used to run ALICE simulation jobs on the DataGrid testbed in order to test the middle-ware provided by the Project. As an example, fig. 13 shows an interactive analysis of a small ALICE simulated event produced on the Grid.

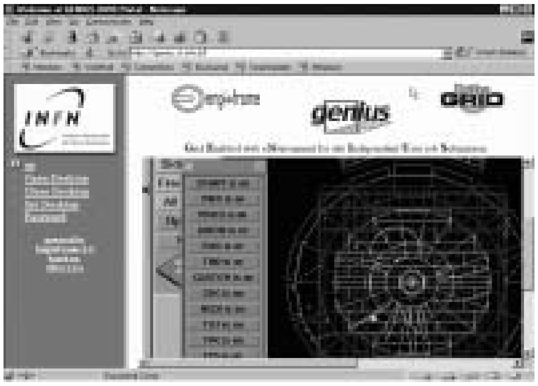

Figure 13: Visualization of a simulated heavy-ion collision at ultra-relativistic energies as ”seen” by the ALICE detector.

4.2. ATLAS

The ATLAS [12] and CMS [13] (see next subsection) Experiments are doing a lot of work with GENIUS in the context of the EU DataTAG [14] and US WorldGrid [15] Projects which aim at uniting European and US Grid sites under the common ”umbrella” of the GLUE schema [16]. Figure 14 shows the status of ATLAS jobs submitted to both EU and US Grid computing resources using GENIUS and a Resource Broker compliant with the GLUE schema which is available in the portal. Figure 15 shows the interactive analysis, performed with GENIUS, of some ATLAS simulations executed on the WorldGrid/DataTAG testbed.

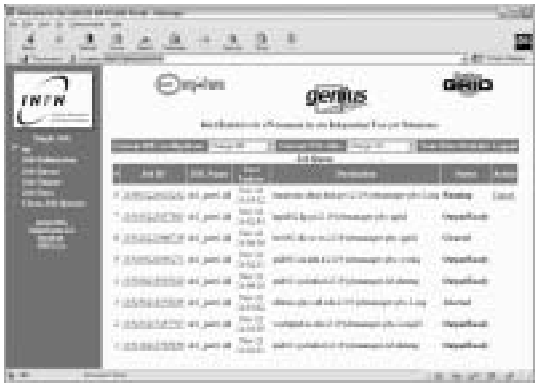

Figure 14: Monitoring of ATLAS jobs submitted to both EU and US Grid computing resources.

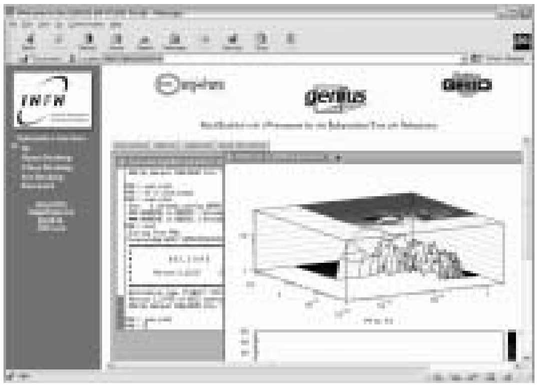

Figure 15: Interactive analysis in GENIUS of ATLAS events simulated on the Grid.

4.3. CMS

The CMS Experiment has by far the most integrated Grid environment into GENIUS, as a result of almost one year of work.

Figure 16 shows the integration of the CMS tool IMPALA into GENIUS, to create and submit jobs to the Grid. while fig. 17 shows the integration of the CMS tool BOSS into GENIUS, to monitor the status of the jobs launched on the Grid.

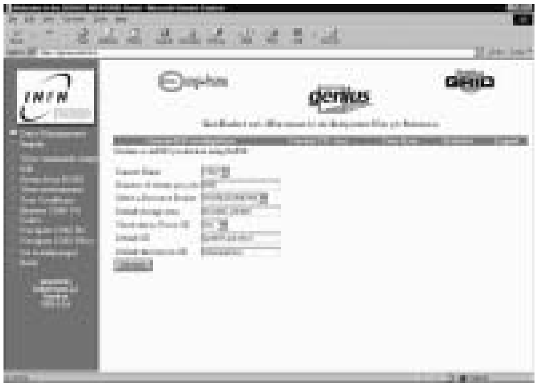

Figure 16: A view of the CMS IMPALA tool integrated into GENIUS.

4.4. Other applications

The GENIUS Grid portal is not only used by the CERN Large Hadron Collider Experiments for their work on the Grid. Figures 18 and 19 show, as examples, interactive analysis, performed with GENIUS, of a geo-physical Digital Elevation Models and a 3D-rendered human DNA.

Acting as a simple and intuitive "gate" to access the Grid, the GENIUS portal brings with itself a tremendous dissemination power. In fact, GENIUS is the official portal of the GriDis Project [17], a complete Grid dissemination testbed set up and managed by the Italian National Research Council in the context of the DataGrid Project.

At the time of writing, also the BaBar Experiment [18] has started to integrate its Grid tools into the portal.

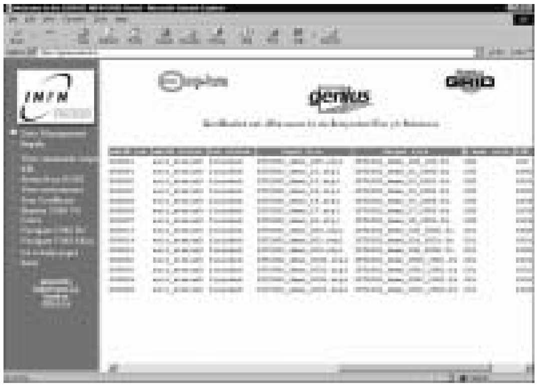

Figure 17: A view of the CMS BOSS tool integrated into GENIUS.

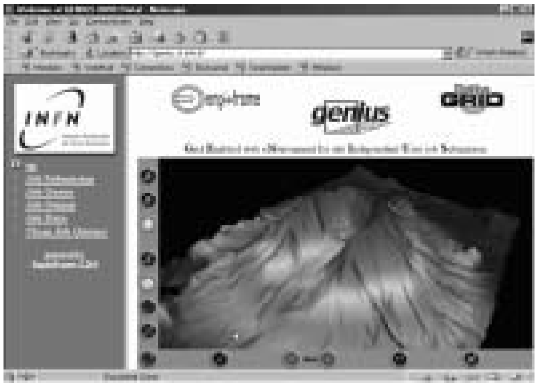

Figure 18: Interactive analysis in GENIUS: virtual reality navigation in a geo-physical Digital Elevation Model (DEM).

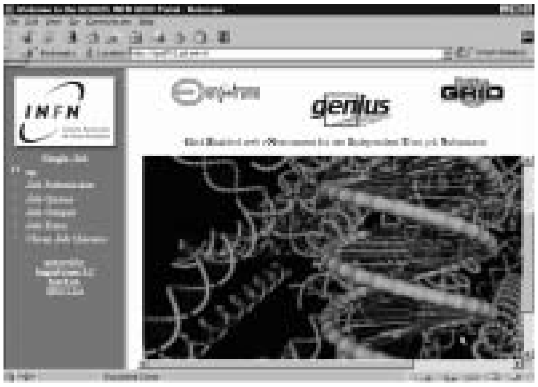

Figure 19: Interactive analysis in GENIUS: 3D rendering of the human DNA.

5. SUMMARY AND CONCLUSIONS

The current implementation of the GENIUS web portal is based on the middle-ware released within the context of the European DataGrid Project. With GENIUS the user can securely access the Grid, submit the single and composite-jobs he has prepared with a graphic job creator to the distributed computing resources and take back the output and can interact, in a seamless way, both with the remote files located on the User Interface and with those belonging to a given Virtual Organization as they all were local ones. Compared to other Grid portals available "on the market", GENIUS offers the following advantages:

- it is a complete production-ready environment which combines the concepts of "user portal" and "science portal";
- absolutely no client software needs to be installed on the user's workstation apart from the web browser with its usual plug-ins;
- it includes support for both single and composite jobs;
- interactive analysis and web access to personal spooling areas are possible.

Furthermore, due to the modularity and flexibility of EnginFrame, which acts as a general-purpose framework underneath GENIUS, the portal can be easily customized/adapted to interact with other Grids or new Virtual Organization using any other kind of middle-wares.

Finally, thanks to the specific grant provided by the INFN Grid Project and to agreement between INFN and NICE srl, GENIUS code is open source and the EnginFrame license is free of charges for the academic and research world.

Acknowledgments

We would like to warmly thank the management of the INFN Grid Project for the grant under which the GENIUS Project is developing and all peoples involved in the INFN Grid and DataGrid Projects for the stimulating discussions and continuous and fast feedback and help. We also would like to thank L. Vaccarossa and M. Verlato, from ATLAS and CMS Experiments, respectively, for providing figures concerning the integration of their Experiment's Grid tools in the GENIUS Grid portal.

References

[1] I. Foster and C. Kesselman (eds.), The GRID: blueprint for a new computing infrastructure, Morgan Kaufmann Publishers, 1999.
[2] R. T. Kouzes, J. D. Myers, and W. A. Wulf, IEEE Computer 29 (1996) 40.
[3] Visit the GENIUS web site `https://genius.ct.infn.it`.
[4] Visit the DataGrid web site `http://www.eu-datagrid.org`.
[5] Visit the Globus Toolkit web site `http://www.globus.org`.
[6] G. Andronico *et al.*, ALICE Internal Note ALICE/OFF 2001-27.
[7] Visit the EnginFrame web site `http://www.enginframe.com`.
[8] Visit the page `http://www.ncsa.uiuc.edu/Divisions/ACES/MyProxy/`.
[9] Visit the page `http://www.datamat.it`.
[10] Visit the TightVNC web site `http://www.tightvnc.com/`.
[11] Visit the Alice web site `http://www.cern.ch/Alice`.
[12] Visit the Atlas web site `http://atlasexperiment.org/`.
[13] Visit the CMS web site `http://cmsinfo.cern.ch/Welcome.html/`.
[14] Visit the DataTAG web site `http://www.eu-datatag.org`.
[15] Visit the WorldGrid web site `http://www.ivdgl.org/worldgrid/index.php`.
[16] Visit the Glue schema web site `http://www.cnaf.infn.it/~sergio/datatag/glue/index.htm`.
[17] Visit the GriDis web site `http://web.datagrid.cnr.it/GriDis/jsp/index.jsp`.
[18] Visit the BaBar web site `http://www.slac.stanford.edu/BFROOT/`.