# DELIVERABLE D.3.1
# PORTALS AND ROAMING ACCESS
# TASK 3.1
# SRS

**WP3,Task 3.1 New Grid Services and Tools**

| | |
|---|---|
| Document Filename: | **CG-3.1-SRS-0017.doc** |
| Work package: | **WP3 New Grid Services and Tools** |
| Partner(s): | **PSNC** |
| Lead Partner: | **PSNC** |
| Config ID: | **CG-3.1-SRS-0017** |
| Document classification: | **PUBLIC** |

Abstract: This document provides a description of the initial specification and the requirements of the Roaming Access, Migrating Desktop and Application portal, which will be developed within the framework of the CrossGrid project.

## Delivery Slip

| | Name | Partner | Date | Signature |
|---|---|---|---|---|
| **From** | | | | |
| **Verified by** | | | | |
| **Approved by** | | | | |

## Document Log

| Version | Date | Summary of changes | Author(s) |
|---|---|---|---|
| 1-0-DRAFT-A | 12Mar 2002 | Draft version | Maciej Kruczkowski, Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer |
| 1-1-DRAFT-A | 9April 2002 | Draft version | Maciej Kruczkowski, Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer |
| 1-2-DRAFT-A | 21April 2002 | Draft version | Maciej Kruczkowski, Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer |
| 1-3-DRAFT-A | 27April 2002 | Draft version | Maciej Kruczkowski, Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer, Stefano Beco, Yannis Perros |
| 1-4-DRAFT-A | 6 May 2002 | Draft version | Maciej Kruczkowski, Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer, Stefano Beco, Yannis Perros |
| 1-5 | 9 May 2002 | Public version | Maciej Kruczkowski, Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer, Stefano Beco, Yannis Perros |
| 1-6 | 13 May 2002 | Public version | Maciej Kruczkowski, Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer, Stefano Beco, Yannis Perros |

| | | | |
|---|---|---|---|
| 1-7 | 31 May 2002 | Public version | Bartosz Palak, Marcin Płóciennik, Paweł Wolniewicz, Mirosław Kupczyk, Norbert Meyer, Stefano Beco, Yannis Perros |

# CONTENTS

# 1 INTRODUCTION

## 1.1 PURPOSE

The purpose of this document is to define the requirements for the Roaming Access, Migrating Desktop and user application portals. In detail, this document will provide a description of requirements for the Migrating Desktop and Application portal, which will allow the user to access his environment from remote computers, like i.e. laptops. It will provide some requirements and problems for other CG WPs.

## 1.2 SCOPE

The application portal should give a unified and consistent window to the CrossGrid environment. The Grid tools will be integrated into the application portal providing easier access and use of the Grid by applications. The CrossGrid application portals will securely authenticate and authorise users to remote resources and help them make better decisions for scheduling jobs by allowing them to view pertinent resource information obtained and stored in a remote database.

The Migrating Desktop should produce a transparent user work environment, independently of the system version and hardware. The Migrating Desktop will allow the user to access grid resources and his local resources from remote computers, like i.e. laptops. It will allow to run applications, manage data files, store personal settings (configuration definitions that characterise e.g. links to the user data files, links to applications, access to application portals and HPC infrastructure, as well as windows settings), independently of the localisation or the terminal type.

This tool will extend the functionality of the application portal by providing:

- a specialised advanced graphical user interface,
- a mechanism for designing and managing the workflow;

The Roaming Access and Migrating Desktop will not support „remote users“. It means that no mechanisms to access Grid resources via mobile phones, PDAs (such as palmtops, organisers, etc.) will be designed and implemented within the confines of task WP3.1.

## 1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

| | |
|---|---|
| DAP | Data Access Package |
| DTD | Document Type Definition |
| LDAP | Lightweight Directory Access Protocol |
| LRU | Last Recently Used |
| MD | Migrating Desktop |
| PDA | Personal Digital Assistant |
| RA | Roaming Access |
| SOAP | Single Object Access Protocol |
| SSL | Secure Sockets Layer |
| TBD | To Be Defined |
| XML | Extended Mark-up Language |
| HEP | High Energy Physics |
| SM | Session Manager |

| | |
|---|---|
| SOM | Self Organising Map |
| VO | Virtual Organisation |
| HTTP | HyperText Transport Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| ISM | Interactive Session Manager |
| SVG | Scalable Vector Graphic |

### 1.4 REFERENCES

CrossGrid CrossGrid Project Technical Annex

### 1.5 OVERVIEW

This document provides the software requirements for MD.

Section 2 provides the overall description.

Section 3 provides requirements and problems for other WPs.

# 2 OVERALL DESCRIPTION

## 2.1 PRODUCT PERSPECTIVE

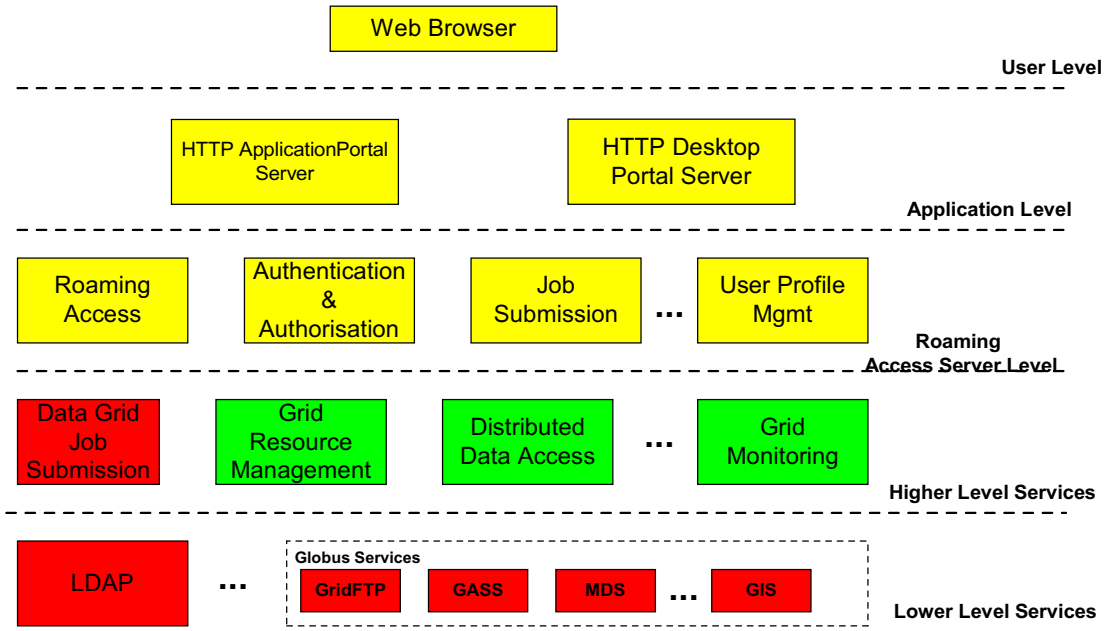


Fig. 1 **Layered view of Task 3.1 architecture**

The main components of the proposed architecture of Task 3.1 are (see Fig. 1, 2):

- Web Browser – a simple web browser displaying pages generated by HTTP Server.
- Command Line Interface – a tool that gives the user text interface to the Roaming Access Server (basic services only);
- Application Portal Server – a service that provides information for HTTP Server needed to create web pages. It keeps information about user sessions and provides first parameter verification.
- Desktop Portal Server – a service that extends the functionality of the Application Portal Server by providing a specialised advanced graphical user interface and a sharing mechanism that allows the user to make files stored on his machine available from other locations/machines;
- Roaming Access Server – a network server responsible for managing user profile, user authorisation and authentication, job submission, file transfer and grid monitoring
- LDAP Database – a network database used for storing user profiles;
- Replica Manager – a module responsible for handling file replicas;
- Resource Broker – a module used while performing job submission;
- Grid Monitoring – WP3.3 module invoked and configured via portal
- Grid Metrics and Benchmarks – WP2.3 module invoked and configured via portal;

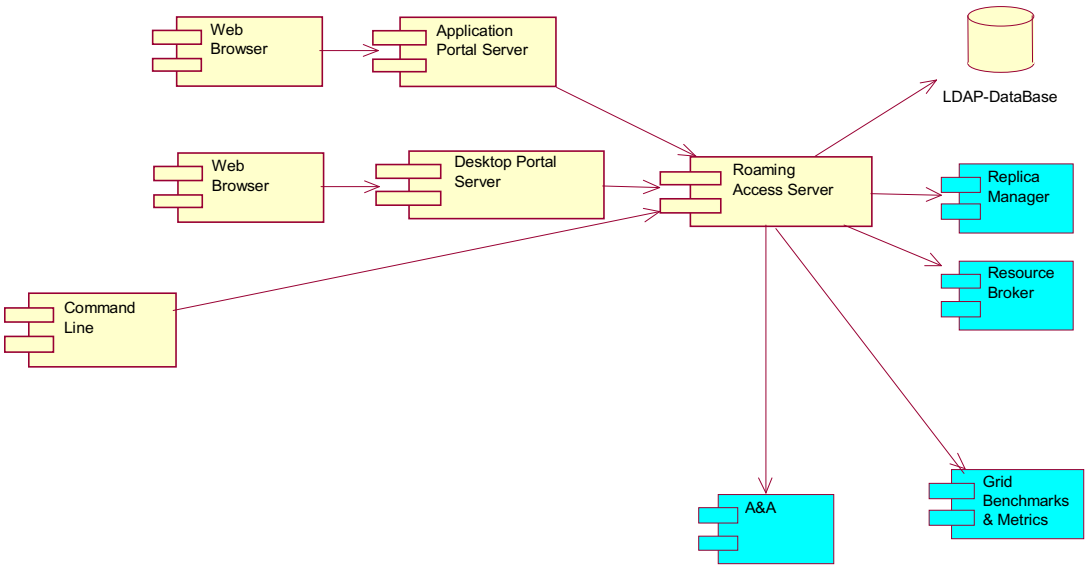


**Fig. 2** Component view of Task 3.1

Task WP3.1 will have connection with task WP3.4 using Replica Manager (see WP3.4 SRS), and task WP3.2 using Resource Broker.

There is no direct connection between task WP3.1 and task WP3.3.

Task WP3.1 will have connection with task WP2.3 (Benchmark and Metrics)

There will be no other direct connection with other task of WP2.

The diagrams below show the workflow while performing file uploading/downloading and job submission operations. Some mechanisms developed in DataGrid will be used:

- Replica Manager (DataGrid WP2), (Crossgrid WP3.4);
- Replica Catalogue (DataGrid WP2);
- File Copier (DataGrid WP2);
- Storage Element (DataGrid WP5);
- Resource Broker (DataGrid WP1), (Crossgrid WP3.2);
- Job Submission Service (DataGrid WP1);

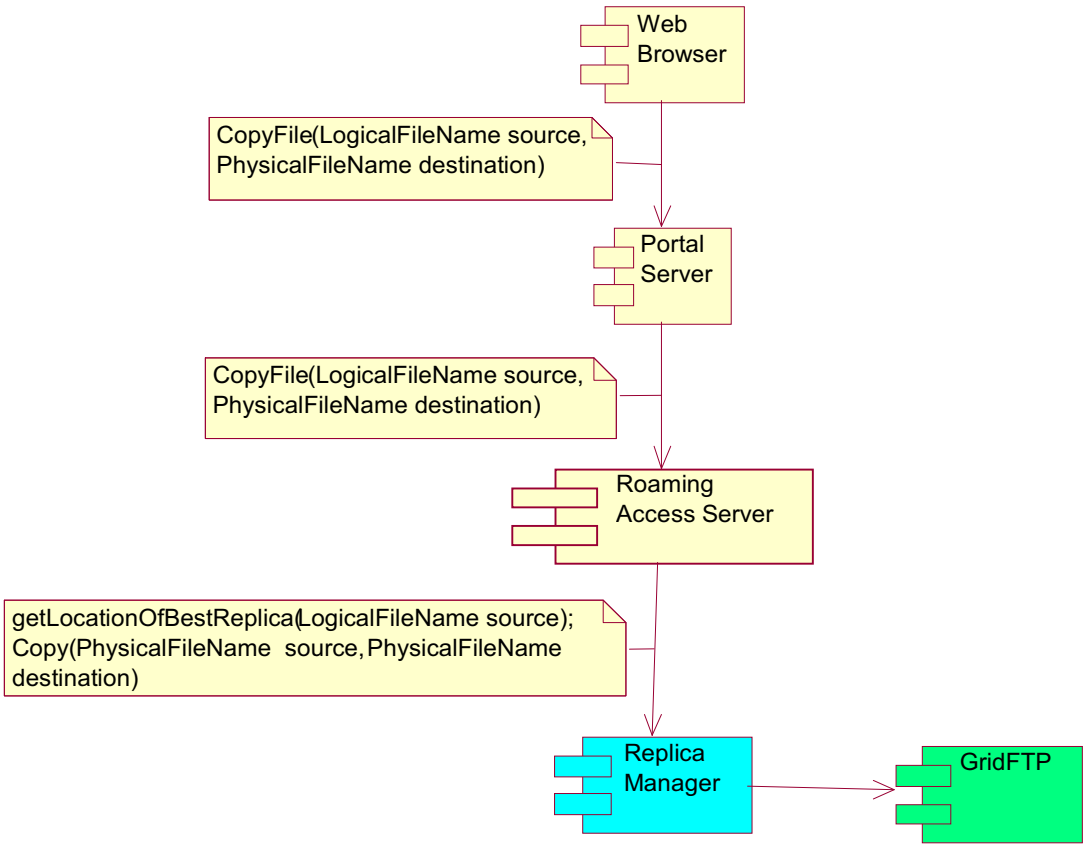


**Fig. 3** Workflow diagram of file downloading.

The diagram depicted on Fig. 3 shows the communication between modules engaged in downloading the file from any Grid Storage Element to a local workstation.

Performed steps:

- The client (Web Browser) orders downloading a file (from some Grid localisation to a local workstation) to HTTP Portal Server, and this server forwards it to the Roaming Access Server.
- The Roaming Access Server asks the Replica Manager about the location of the best replica.
- The Roaming Access Server (based on the answer from the Replica Manager) orders the Replica Manager to copy the file operation from some Storage Element to the local host – the Roaming Access acts only as the mediator in the transmission of files (the dashed line shows this step).

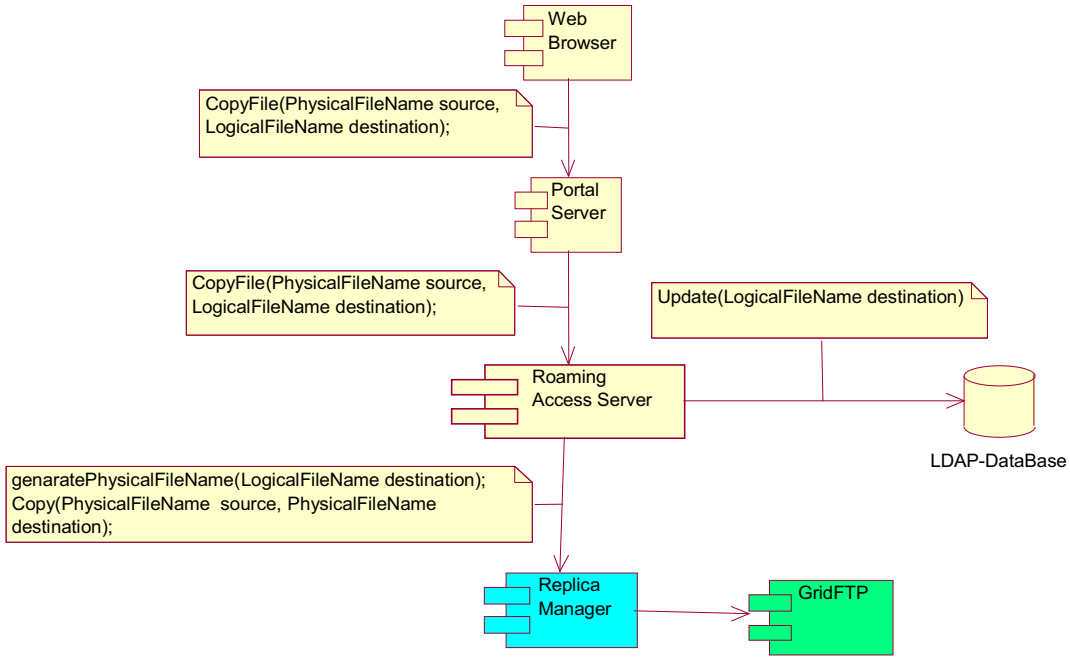


**Fig. 4** Workflow diagram of file uploading.

The Uploading Procedure uses similar mechanisms to the downloading one (see Fig. 4). Performed steps:

- The client (Web Browser) orders uploading a file (from the local workstation to some Grid localisation) to HTTP Portal Server, and this server forwards it to the Roaming Access Server.
- The Roaming Access Server asks the Replica Manager about the best localisation where the given file can be put.
- The Roaming Access Server orders the Replica Manager to copy the file operation from the local host to some Storage Element – *similarly* to *the* downloading file the Roaming Access acts only as the mediator in the transmission of files (see the dashed line).
- An entry *describing* logical file name is added to LDAP database;

Because of some security restrictions (some machines can refuse the demand of uploading files), the third party transfer may be impossible to perform, both in file download and upload operation. In that case the client shall put/get file to/from the given localisation itself.

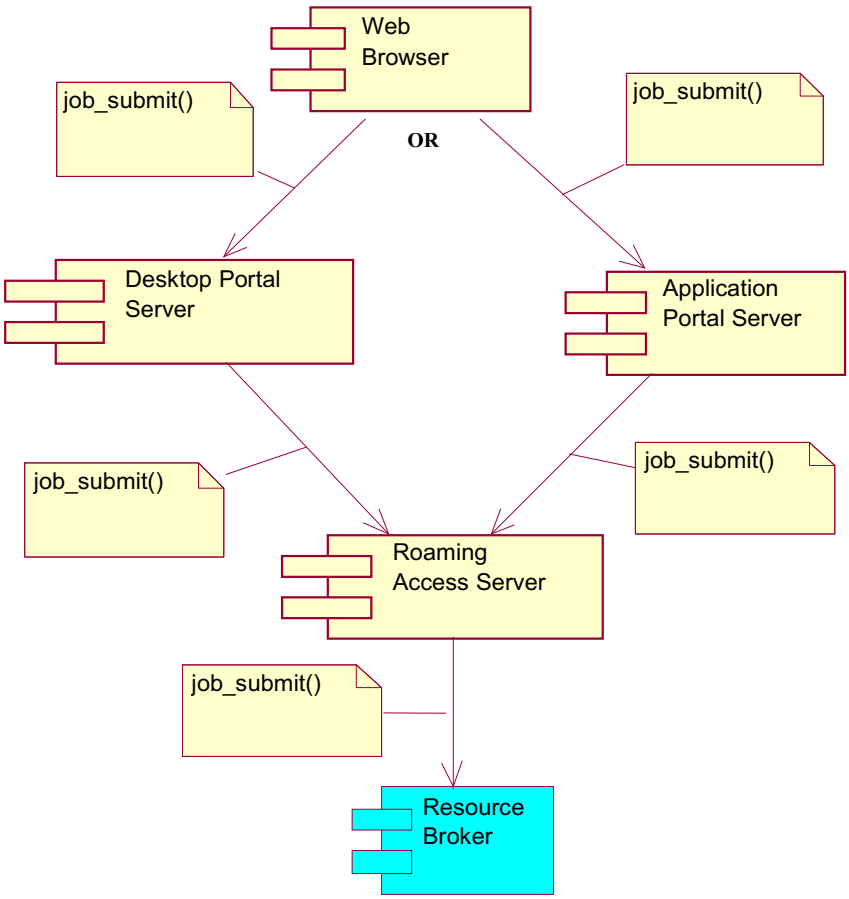


**Fig. 5** Workflow diagram of application submission

Fig. 5 shows the flow of message passing between modules while submitting a Grid application. All information needed to run a job will be passed to the Resource Broker using the Job Description Language that describes e.g. job name, input arguments, job requirements as well as input and output data files localisation.

The user can run an application with the input data file(s) that are normally stored on the local workstation or on any of the Grid Storage Elements. The Resource Broker will transfer files (according to the description) to the remote Computing Element. When an input file resides on the submitting machine, there is a danger that during computations they will not be accessible (the local computer could be switched off every time), so the required files should be copied to a safer place – e.g. on any of the Grid Storage Element.

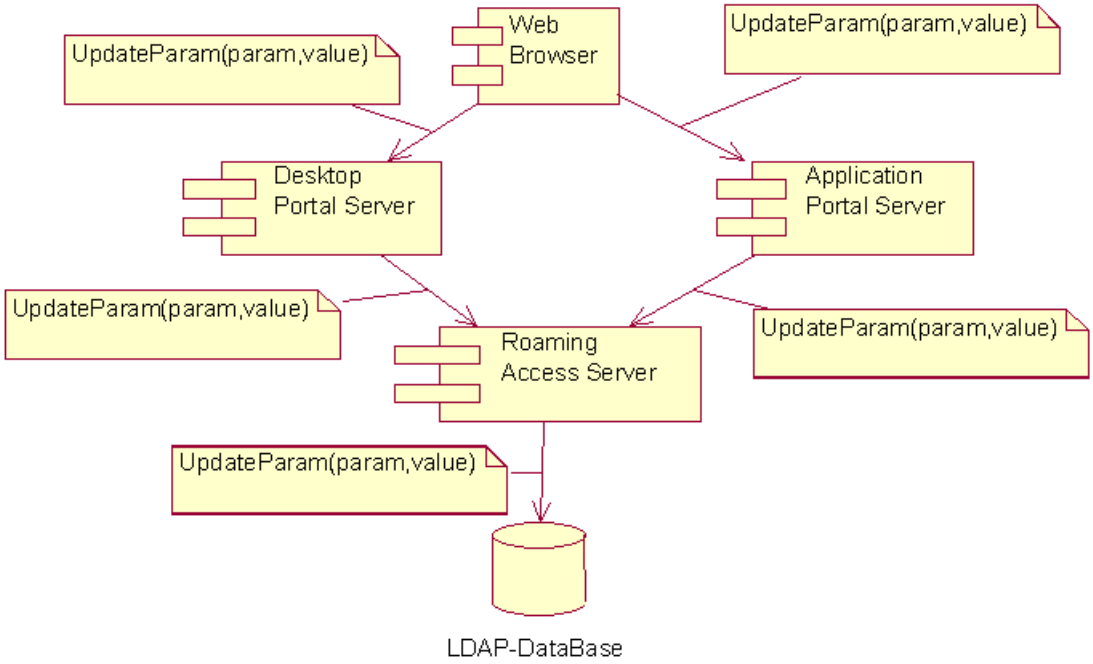


**Fig. 6** Workflow diagram of update of settings

Designing the user environment is probably the most important aspect of Task 3.1. Certainly, intermediate modules and middleware service are also important, but they are not visible for the end users. The transparency of usage and hidden engines are the aim of the Grid related projects. That is why we have to create a system being as user-friendly as possible. 'Settings' stand for a set of possibilities which can be changed, e.g. icon destination, desktop layout, popular applications, special monitoring tools, etc., and colours and background as well. The future popularity of the CrossGrid applications and tools depends on the functionality of the front-end.

The proposed protocols for that part of the product are SOAP, Globus GSI and LDAP, see Fig. 6. Whenever something has changed in the user personal environment, the Client sends a request to update the database. The RA module mediates between the Client and the LDAP database. These updates can be achievable from anywhere with access to the grid.

### 2.1.A Interoperability with Task 1.1

This section describes the User Portal that provides access to the interactive services for Biomedical Application applications (task 1.1). The Biomedical Application applications enable the user to perform simulation using the results of the 3D modelling and visualisation that shows representation of the arteries.

**Groups of potential users:**

Users of the Portal services for the Biomedical Application (see Fig. 10) can be considered on two levels, that is at the institutional and at the individual level.

At the Institutional Level the users shall be:

- Hospitals
- Academic Institutions
- Research Institutions
- …;

At the Individual Level the users are classified as:

- *Domain experts*;
  - Radiologists
  - Vascular surgeons
  - …;
- *Medical doctors*
- *Postgraduate Students*
- *Researchers*
- ....

Another group of Beta users that shall provide most of the requirements of the Portal services have to be defined. This group can initially include people suggested by UvA and LUMC.So for the Biomedical Application addresses as users the experts of the specific medical domain. However, the CrossGrid portal services can address to the needs and requirements of other interested groups of users such as researchers, postgraduate students and trainees as well as medical doctors with *relevant* interests.

Access to the portal services must be organised according to the user groups. According to the above-mentioned matter, we can consider three major groups of users of the Biomedical Grid Portal:

- Domain experts
- Researchers
- General public

The domain expert can access the portal through their web-browser.

The portal visitor must be identified and authenticated. Specific permissions will be attributed to the user according to the user group they belong to and according to their individual permissions.

The domain expert shall access the patients' (distributed) database; make queries using appropriate forms and retrieve specific records that will be fed afterwards to the 3D modelling application. Thus the user will be presented with the 3D representation of the arteries according to the retrieved data.

For the researcher group of users the procedure can be the same, though the databases will be formed with suitable datasets created properly for the research and educational purpose.

For the general public 3D modelling will be limited to a demonstration of the module including a look up table with Input Selection (Patient or Specific Case) and Result Visualisation/Presentation (3D images and/or Table) without processing.

Concerning simulation the domain expert can have the possibility (possible **actions**)

a. to perform the simulation using the results of the 3D modelling and visualisation application
b. to input the application (simulation) related parameters
    a. by means of form input and/or
    b. interactive image
c. to define the Grid network requirements (speed, throughput etc)
d. to interact during the execution for changing the simulation parameters
e. to interact during the execution for changing the grid network requirements
f. to visualise the results of the simulation
g. to visualise intermediate results if applicable
h. to store the results of the simulation

Thus the following needs of the above-mentioned group of users shall be served through the portal **services**:

General:

- Users must be registered with the service (application)
- Different permissions must be given to different groups of users
- The portal should authenticate the user and check for permissions
- The user can personalise the Biomedical Application environment
- User profile management

Domain Experts.

- Have access to the distributed medical data
- Have access to stored records of a 3D model of the arteries generated by the medical imaging application for the purpose of estimation and diagnosis
- Define the parameters of the calculations of the blood flow simulation using a graphical interface or tabular data
- Start/Restart, pause terminate simulation
- See intermediate results of the simulation (if applicable)
- Interact with the application for simulating vascular interventions
- Visualise the results of the applications

- Store results data and/or scenarios

The other groups of potential users (students, researchers, trainees, medical doctors etc) have similar requirements, except that they will be given a limited access permission to the resources. Specifically they shall use

- historical archives of 3D models of arteries instead of real data
- recorded scenarios of test cases etc.

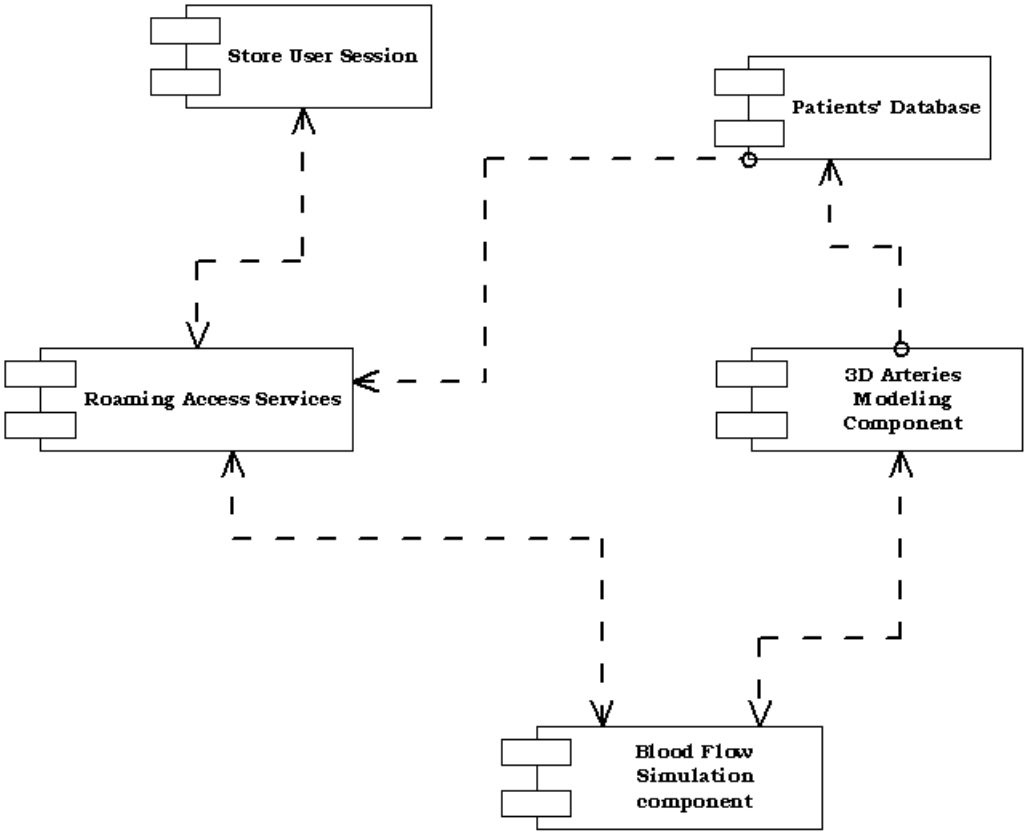


**Fig. 7** Blood Flow Simulation using 3D model user interface

### 2.1.B Interoperability with Task 1.2

This section describes the User Portal that provides access to the interactive services for flooding crisis team support applications (task 1.2). The flooding crisis team support applications enable the user to perform visualisation that describes hydrological and meteorological situation.

**Groups of potential users:**

- End users:
- River authorities;

- City councils;
- Local governments;
- Ordinary citizens, public;
- Scientists: meteorologist, hydrologist, hydraulic engineers, etc.
- Crisis teams members;
- Media: press, TV, etc.
- Insurance companies.
- There are two general group of users:
- Experts – they will have a possibility of defining: the workflow between applications, applications parameters, simulations scenarios, etc...
- Ordinary users – they will only be able to see the information that *is given by the portal* (e.g. results of simulations, some historical data, etc) or run predefined simulations.
- SAV will prepare a more detailed specification of groups of portal users.

The User Portal provides **services**:

- to define jobs to analyse data;
- to request the job output in graphical and/or alphanumerical format;
- user profile management;
- to create script that would enable the next interactions of applications, a possibility to use the results in the next application (modelling of the data flow);
- to save results (remotely and/or locally);
- to make a “reservation” of the computing power for “emergency simulations”;

List of possible **actions**:

- taking the input data for applications from data sources (e.g. measurement equipment as meteorological radar, etc) or from data storage (historical data);
- making a “reservation” of the computing power for “emergency simulations”;
- choosing the results of one simulation as input for the next simulation step in the interactive manner;
- running applications on 1 cluster, or on parallel machines;
- choosing the parameters for the given simulations (e.g. type of model) by “experts” (only); For different model types there will be different input parameters.
- the incoming results can be input data in the next iteration;
- browsing the database of simulation results;
- informing the chosen group of users about the particular events by emails (information about precipitation, state of calculation, etc)

**Application remarks**:

- Results of the simulations should be presented to the user in near real-time manner (e.g. after every 15 minutes in case of flush flood crisis);

- Application in workflow will communicate using files (via GridFTP).
- There will be no interaction with the end user while the application is running (application should finish its computation);
- Amount of data – e.g. for hydraulic simulation 27 MB of input data generates 280 MB of the output data;
- Communication between applications: every simulation is an independent entity and does not communicate with other applications while performing the computations.
- SAV IU in co-operation with SAV Meteorological Institute will prepare a more exact specification of the way of controlling simulations.

Information available via portal**.**

- *Current* data about the general hydrological and meteorological situation from the data sources (e.g. radar – information about precipitation – about 100 kB per every 15 min);
- Detailed information about the situation for the given city (e.g. for the city council, local government, local crisis teams, etc.);
- Information should be propagated in a form of sheets, tables, charts, etc.
- Historical data – stored on a data storage system;
- SAV IU will prepare a more exact specification of the sort of information which should be accessible via portal, in co-operation with the SAV Meteorological Institute.

Visualisation

- All graphical information will be generated in the form of standard formats of graphical files (e.g. jpeg, bmp, etc)
- It will be nice to give the user a possibility of choosing what kind of pictures will be presented via portal;
- The user will specify the parameters for visualisation tools.
- Visualisation (preparing graphic files) will be made “inside the Grid”. The computing of visualisation is also possible on local machines but requires a large amount of data so the computation will be *longer*
- All visualisation tools will be prepared by SAV. SAV will make the existing visualisation tools available or (in case the mechanism cannot be used by the portal) will prepare other ones.

### 2.1.C Interoperability with Task 1.3

This section describes the User Portal that provides access to the interactive services for HEP applications (task 1.3). The HEP applications enable the user to perform analyses on data generated in HEP experiments and stored in large distributed databases. These data are processed by data-mining techniques such as Neural Networks or clustering techniques like SOM.

The User Portal provides **services**:

- to give access to large distributed databases where HEP data are stored;
- to define jobs to analyse data;

- to request the job output in graphical and/or alphanumerical format;
- to save the whole user working sessions;
- to reload and to rerun the saved sessions.

The data in storage elements are structured in DATASETS. A METADATA catalogue provides global information about different DATASETS. This description and the information itself will be based on an XML format. The user opens an interactive session with grid resources: they can run their application in the grid environment with their VO DATASETS and see the output data on their user interface.

This is a **list of actions** that a HEP user can perform in a general scenario:

1. A user starts with the login operation authenticating themselves using their certificates to be accepted as a member of the HEP VO.
2. Once accepted, the user sees a list of DATASETS available for their HEP VO on the client.
3. Within this list, they select the relevant DATASET for their purposes to the system via the client and submits the choice.
4. Then the user selects their HEP job among the available list of predefined HEP jobs to be executed in the computational grid, or they can define one for their specific needs.
5. The user defines the input data for the selected HEP job, e.g. subsamples obtained by filtering operations over their DATASETS. The user submits the selected HEP job with the defined input data.
6. While the user waits for the complete output, they can see intermediate outputs either in a graphical and alphanumeric format, if the running HEP job allows this.
7. If the user understands from intermediate outputs that the running job does not converge, they can send a request to stop it (like CTRL-C command).
8. At the end of the session, the user can save the whole session trace in XML format either on their local storage or on the HEP VO repository within the system. It will be possible to edit and reuse it later on all the saved sessions.

### 2.1.D Interoperability with Task 1.4

This section describes the User Portal that provides access to the interactive services for advanced browser of meteorological data applications (task 1.4). The advanced browser of meteorological data applications enables the user to perform analyses on collected meteorological data.

The User Portal **provides services**:

- to give access to large distributed databases where meteorological data are stored;
- to define jobs to analyse data;
- to request the job output in graphical format;
- to store the whole user working sessions;
- to create script that would enable the next interactions of applications, a possibility to use the results in the next application (modelling of the data flow);

- to save results (remotely and/or locally);
- to inform about events connected with calculations and/or date transmission by emails. (Because most of the GSM operators offer email to SMS gateways, also SMS can be sent as a notification about events.)

List of possible **actions**:

- Requesting a given analysis (a possibility of a preliminary request, prompt of system ref. number of iteration).
- After the GUI-side first verification of the parameters correctness.
- The user should have a possibility to browse the database of quickly accessible analyses. If there is no such database – there should be a possibility to work with data on approximate areas similar to that requested by the user. The user does not have to be conscious about the state of data (they can already exist in a finished form, or data can be received from the archive and to execute conversion).
- After finishing the application the user should be informed about it in session (interactive work off-line).
- To use results in the application that run on local workstations the user can save them locally or can be informed about the address of the localisation of results in the Grid. The user may have to build a kind of wrapper to get access to data that are available in the Grid.
- The user has a possibility to choose a visualising application located in the Grid. As a result, in such situation a small picture is sent to applet. If the visualising application is on the client workstation –a link to the result application file should be given. The user will decide whether to copy this result file to the local disc.
- The incoming results can be input data in the next iteration of the same application (or a completely different application, eg. Wave modelling with the use of weather prognoses, see Fig. 8). The module of data flow modelling.
- In one session it should be possible to start simultaneously two different applications on the same input data.
- The user should finish the session.

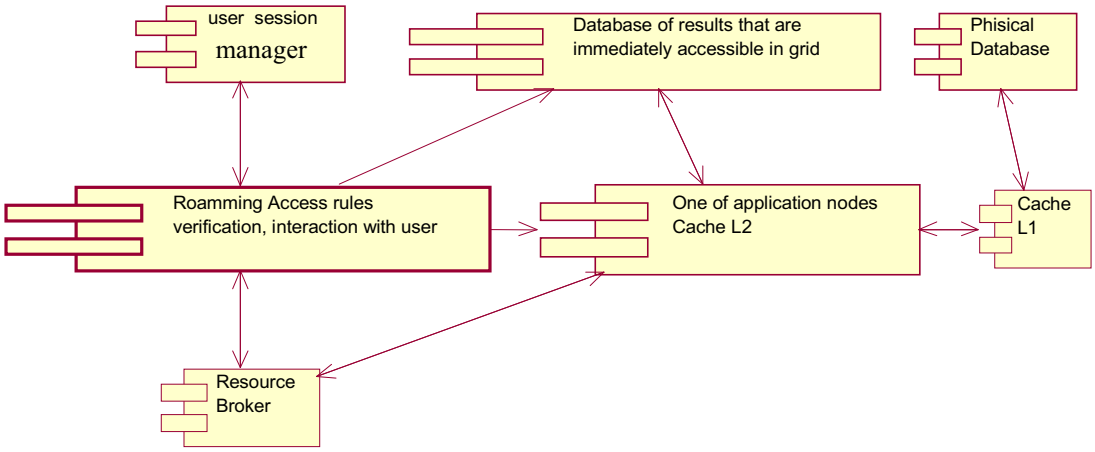


**Fig. 8** Advanced browser of meteorological data

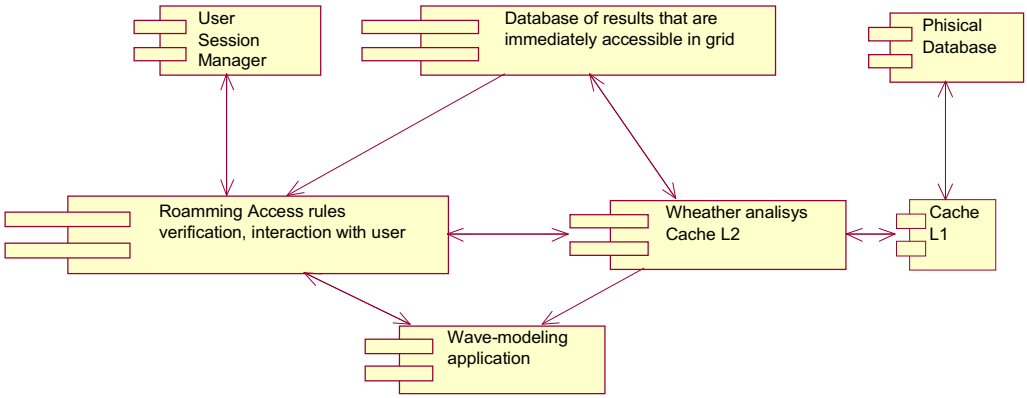


**Fig. 9** Wave modelling with the use of weather prognoses

**Application remarks**:

The user should be informed about the presumable time estimation of his operations. In particular the message about the state of transmission should be generated from „belt devices" (assembly devices).

**Additional remarks**:

Module of modelling task flow (PROPOSAL):

The user has a possibility of modelling the dependence between assignments. Generally we can assume choosing as a model of connections *between* a directed graph, e.g. :

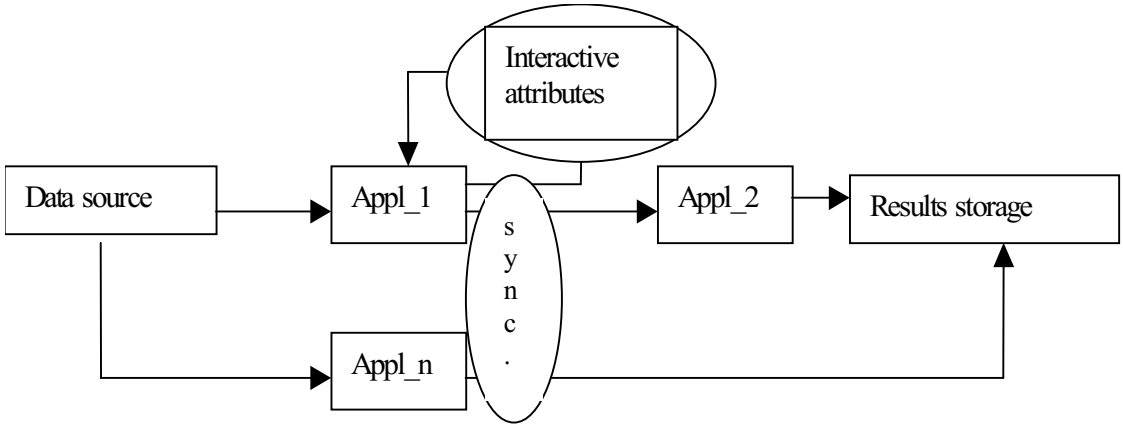


**Fig. 10** Module of modelling task flow (example)

We have introduced an example of dependence between the user tasks. The user composes his model of task flow using graphic components with a description of task and actions. The rectangles contain a description of the assignment Fig. 10 (RSL and edge data for task), the ellipses symbolise steering loops in high level language. The "Interactive Attributes" symbolise a collection of edge limitations for tasks in the next iterations. The user can be asked to confirm or give new edge conditions after the iteration is finished. The Synchronisation Point makes possible the management flow, settlement of readiness of all given streams to the next iteration / of application.

### 2.1.E Summary of interoperability with task 1.1- 1.4

**Common points**:

Different user type – generally access to the portal services must be organised according to the user groups.

The User Portal provides **services**:

- to define jobs to analyse data;
- to request the job output in graphical and/or alphanumerical format;
- user profile management;
- to create script that would enable the next interactions of applications, a possibility to use the results in the next application (modelling of the data flow);
- to save results (remotely and/or locally);
- authenticate the user and check for permissions;

Common actions type:

- input the application (simulation) related parameters
  - by means of form input and/or
  - interactive image
- defining the Grid network requirements (speed, throughput etc)

- interacting during the execution for changing the application parameters (not for all application)
- visualising the results of the application
- visualising intermediate results if applicable
- storing the results of the application/simulation

### 2.1.1 System interfaces

The Roaming Access Server offers one interface to all kinds of clients. This interface consists of the functions that allow a client application, such as the Thin Web Client, Command Line Interface, Advanced Client to submit or cancel a job, manage the user files, user profiles and user authorisation and authentication.

### 2.1.2 User interfaces

Graphical interface between user and Desktop Portal Server application will be in the form of the user desktop – similar to Windows or Linux systems desktops. New desktop will be independent on system desktop. It means that it will be no direct interaction between desktops, like copying files using drag and drop mechanism. The main goal of the extra desktop is to make it possible for the grid user to work with local and grid applications without difficulty.

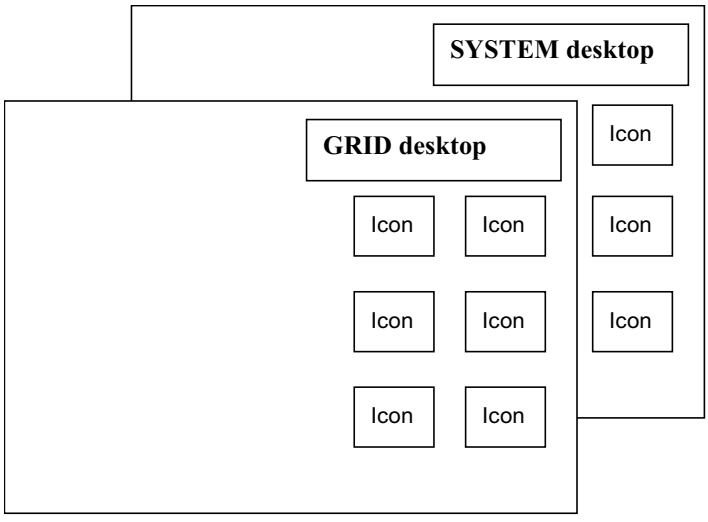


**Fig. 11** The Grid Desktop

The Grid Desktop (see Fig. 11) contains objects called icons that symbolise links to files (local or stored “inside the Grid” on any Storage Element) and applications – local or Grid (applications that are dedicated to execute in a distributed environment). There will be set of defined available actions and properties related to every icon will have its own popup menu with. The user can invoke actions specific to given icon in few ways (e.g. double clicking on icon, from the pop-up menu or a keyboard operation). Icons can be freely moved – their

positions will be remembered as part of the user profile. Adding icons to the desktop will initialise the procedure of storing their properties in the LDAP based server. It may additionally cause a physical move of file into the grid (into Storage Element or other storage). The user can define, similar to Windows or Linux family systems desktops, wallpaper as the background. The new desktop is not supposed to be split into two parts: local and grid part.

### 2.1.3 Hardware interfaces

No special interfaces with hardware components are considered at this moment.

TBD

### 2.1.4 Software interfaces

Proposed technologies:

- XML 1.0
- DTD
- Java
- HTML
- Servlet
- GPDK, CoG
- Globus

### 2.1.5 Communications interfaces

Proposed protocols:

- SOAP
- SSL
- SHTTP
- LDAP
- Globus based

### 2.1.6 Memory constraints

TBD

### 2.1.7 Operations

TBD

### 2.1.8 Site adaptation requirements

TBD

## 2.2 PRODUCT FUNCTIONS

The Grid Desktop provides the following main features to the end users:

- a specialised advanced graphical user interface:
  - managing icons that symbolise actions, data objects or applications

- a mechanism for designing and managing workflow;
- a sharing mechanism that allows the user to access files stored on his machine from other locations/machines (a kind of “mobile briefcase”);
- secure authentication and authorisation;
- a single point of entry;
- storing/retrieving users profiles independently of their location and used terminal;
- “Choose & Go” service allowing users to select applications to be run;
- “Check Status” service allowing users to check the status of their own jobs;
- “Retrieve” service allowing users to retrieve the job output;

The application portal shall provide the following main features to end users:

- Input manager, interaction with user
- Graphic output presentation (can serve as an input to next iteration)
- Benchmark and Metrics presentation
- “Check Status” service allowing users to check the status of their own jobs - Job Monitoring presentation
- Workflow /scripts tool
- Alphabetic output presentation
- File management: downloading/uploading
- Parameters pre-verification
- System documentation
- Additional application portal enhancements – chat, etc.

The user profile shall contain the following data as a minimum:

- Username
- PassPhrase
- List of pending jobs
- *Any A&A info*

### 2.2.1 General functionality of the application portal.

According to the applications summary in §2.1, the next figure Fig. 12 shows a high level view of the applications portal:

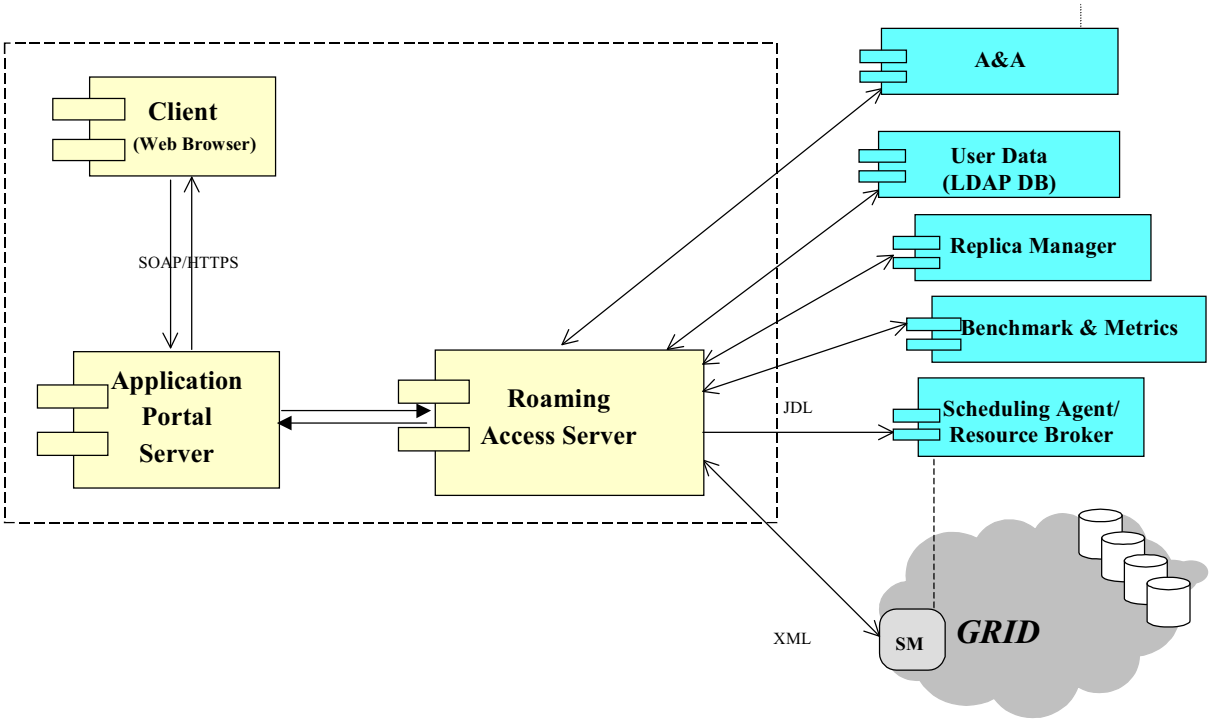


Fig. 12 **High level view of the application portal**

Referring to Fig. 12, three main blocks can be identified: the Client, the Application Portal Server and the Roaming Access Server.

**Client:** Fig. 13 shows a possible layout for the user interface composed of seven panels.

On the left side in Fig. 12, all input panels are presented on Fig. 13, while on the right side all the output panels are depicted.

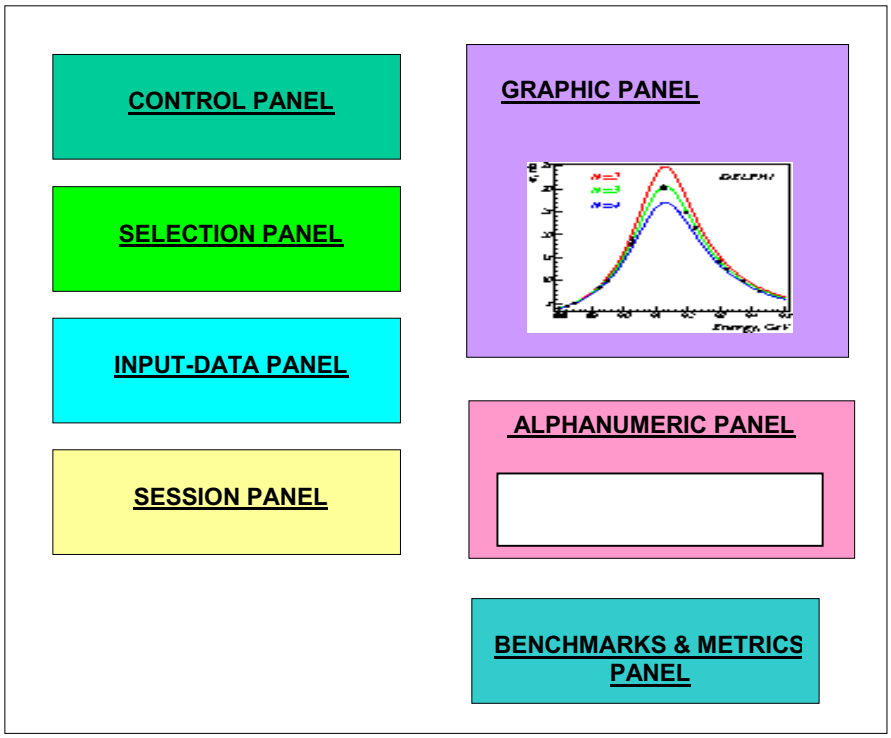


**Fig. 13** All panels of applications

- *"Control panel"* allows the user to perform login operations to submit the selected job; to stop the running job; and finally, to close the current session with the possibility to save the session trace. In this panel the user can also see the status of running jobs.
- *"Selection Panel"* allows the user to select datasets; to select the job to submit; to select the input data from the chosen dataset for the application job.
- *"Input-Data Panel"* allows the user to define new scripts or input data for the job .
- *"Session Panel"* allows the user to look at the current session trace;.(if possible session can be restored)
- *"Graphic Panel"* shows the output data in a graphical format. In some application it can have additionally functions similar to “Selection Panel” (Data that can be used as input to next iteration).
- *"The Alphanumeric Panel"* shows the output data in an alphanumerical format (if possible).
- *"The Benchmark & Metrics Panel"* shows benchmarks and metrics. It allows user to run benchmarks and metrics and to see the results;

**Application Portal Server:** figure Fig. 14 shows services of the Application Portal Server that will be used for applications:

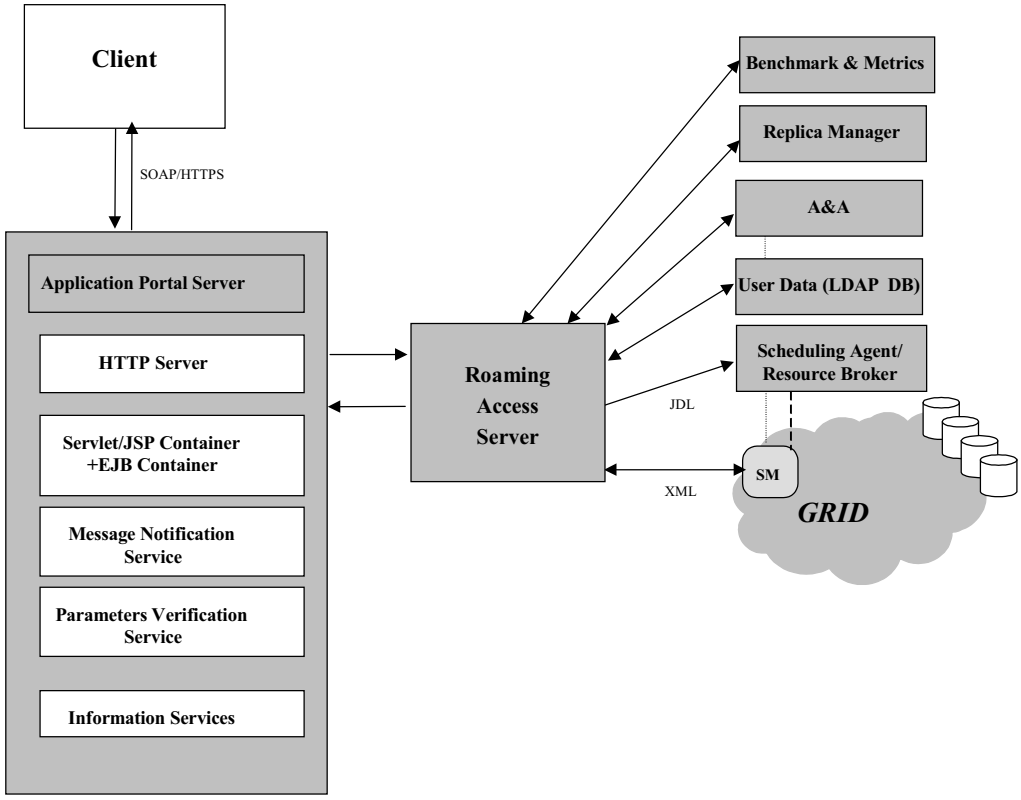


**Fig. 14** Main services of the Application Portal Server for applications

- *"HTTP Server "* handles HTTP connection with Client.
- *"Servlet/JSP Container "* + *"EJB Container"* prepare the pages for the Client; invokes the Roaming Access Service services to send requests and receive data. It is designed to support the development and deployment of applications. The role of this applications is to prepare the presentation of UI, perform operations connected with file transfer, support UI of grid tools (e.g. applets for application graphical output, benchmark, monitoring), workflow management.
- The Message Notification Service is a service that enables notifying the user about particular application states (mail, SMS) of other events
- The Parameters Verification Service – responsible for pre–verification of the user interaction with the application portal. Grid applications will define rules that should be checked.
- Information services – will be responsible for system documentation, additional application portal enhancements – chat, etc.

**Roaming Access Server:** figure Fig. 15 shows the three main services of the Roaming Access Server that will be used for applications:

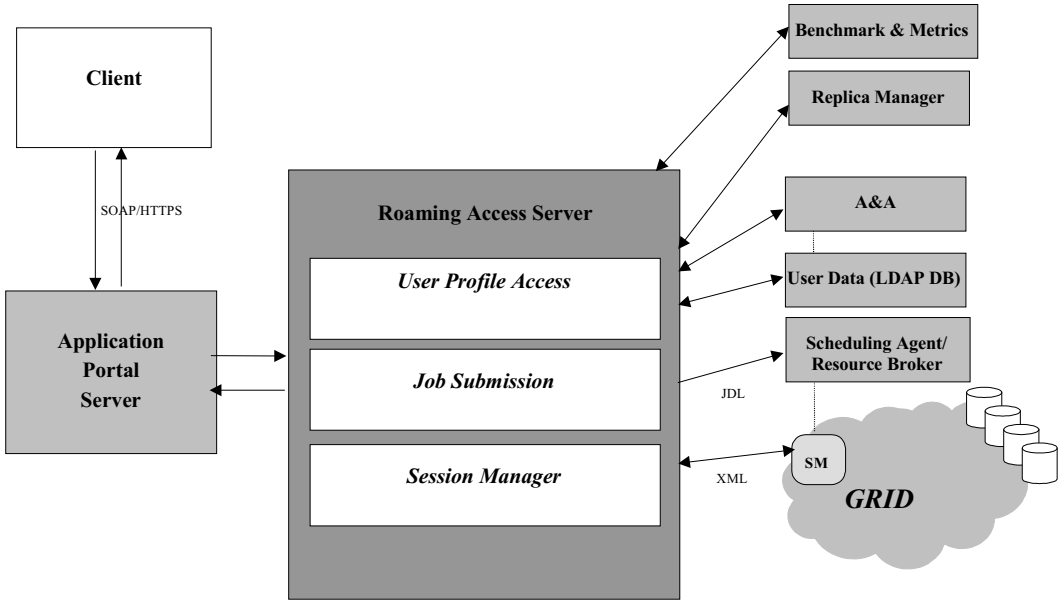


**Fig. 15** Main services of the Roaming Access Server for applications

- *"User Profile Access "* service handles the user login operations to the grid environment and retrieves the User Profile according to their certificate and privileges; retrieves the list of the datasets available for the application users; retrieves the list of the predefined application jobs available for the application users.
- *"The Job Submission"* service sends the request for grid resources to work on the selected dataset to the Scheduling Agent using a high level description language (JDL script).

*Session Manager* receives the information about the application job selected to be executed; requests/receives the outputs of the executed application job in XML format;, sends a CANCEL-CURRENT-PROCESS command to SM to stop the running application job; when requested by the client, sends a BYE command to SM to close the current session; when requested by the client, saves the whole session trace in an XML format; handles the client request about storing/retrieving the saved session.

#### 2.2.1.1 *Example of functionality of the application portal – HEP application portal.*

Product functions based on HEP application example (we will provide general mechanism similar for all application) are presented below.

According to the HEP application scenario in §2.1, the figure Fig. 16 shows a high level view of the application portal for HEP applications and related data flows:

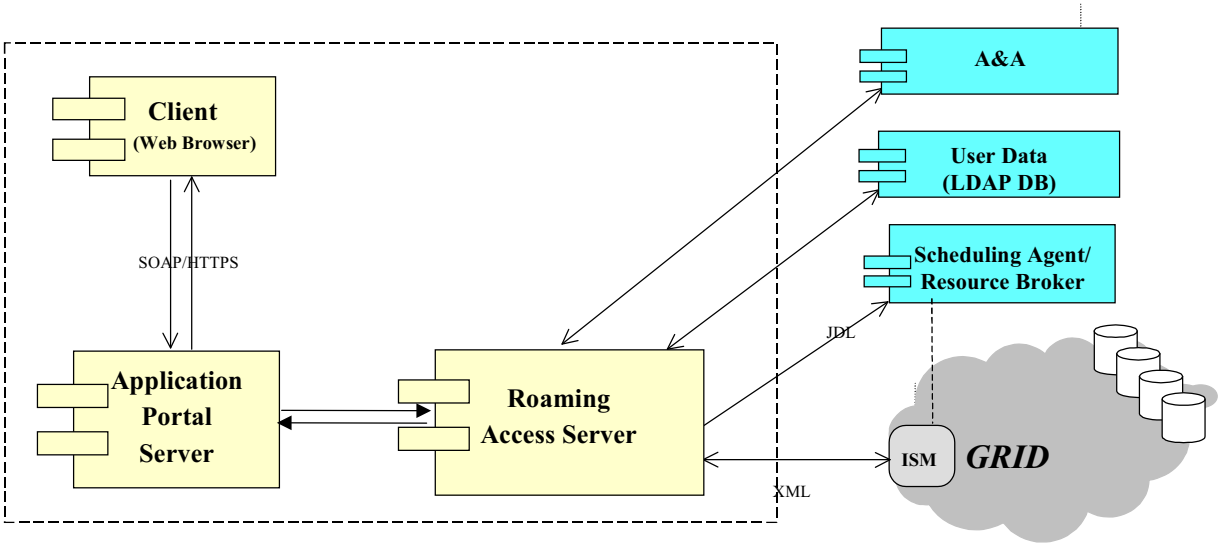


**Fig. 16** High level view of the application portal for HEP

Referring to Fig.16, three main blocks can be identified: the Client, the Application Portal Server and the Roaming Access Server.

**Client:** Fig. 17 shows a possible layout for the user interface composed of six panels.

On the left side, all input panels are represented Fig. 17, while on the right side all the output panels are depicted.

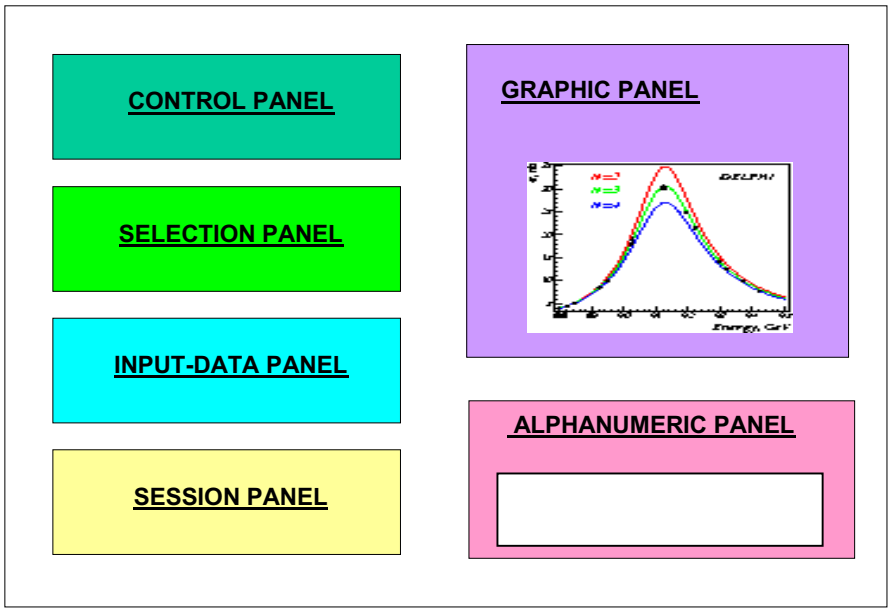


**Fig. 17** All input panels of HEP application

- *"Control panel"* allows the user to perform login operations (step 1 in the HEP application scenario); to submit the selected HEP job (step 6); to stop the running HEP job (step 8); finally, to close the current session with the possibility to save the session trace (step 9). In this panel the user can also see the status of running HEP jobs.
- *"Selection Panel"* allows the user to select the DATASET (step 3); to select the HEP job to submit (step 4); to select the input data from the chosen DATASET for the HEP job (step 5).
- *"Input-Data Panel"* allows the user to define new scripts (step 4) or input data (step 5) for the HEP job .
- *"Session Panel"* allows the user to save the current session trace; to reload the previously saved sessions, to edit them and eventually to run it again as a whole or only some steps of them, applying small changes (step 9).
- *"Graphic Panel"* shows the output data in a SVG graphical format (step 7).
- *"Alphanumeric Panel"* shows the output data in an alphanumerical format (step 7).

**Application Portal Server:** figure Fig. 18 shows the two main services of the Application Portal Server that will be used for HEP applications:

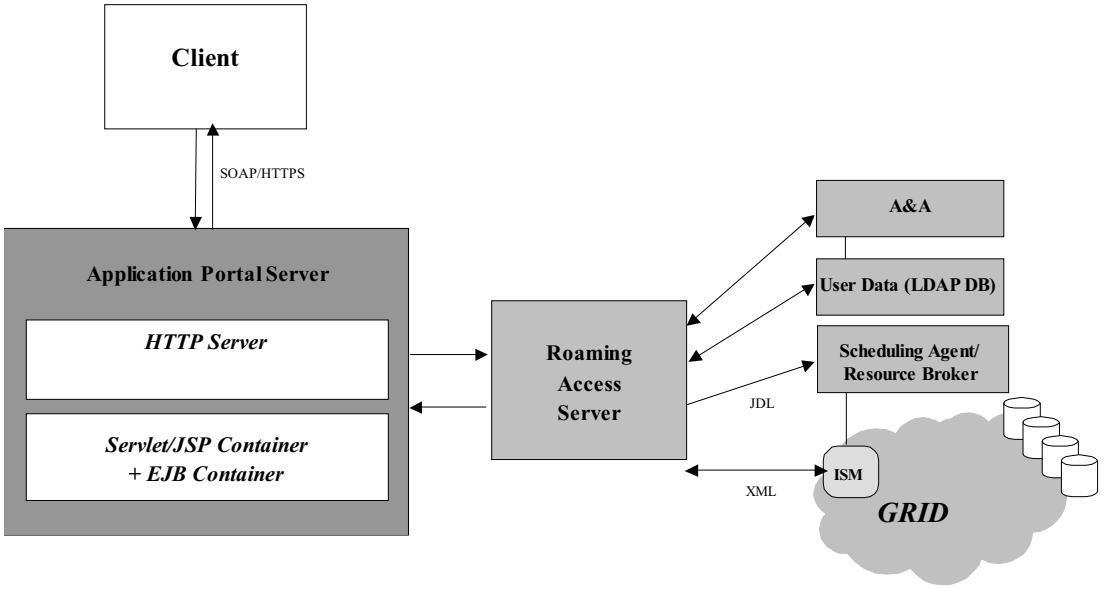


**Fig. 18** Main services of the Application Portal Server for HEP applications

- *"HTTP Server "* handles connection with Client side by HTTP protocol.
- *"Servlet/JSP Container "* + *"EJB Container"* prepare the pages for the Client; invokes the Roaming Access Service services to send requests and receive data.

**Roaming Access Server:** figure Fig. 19 shows the three main services of the Roaming Access Server that will be used for HEP applications:

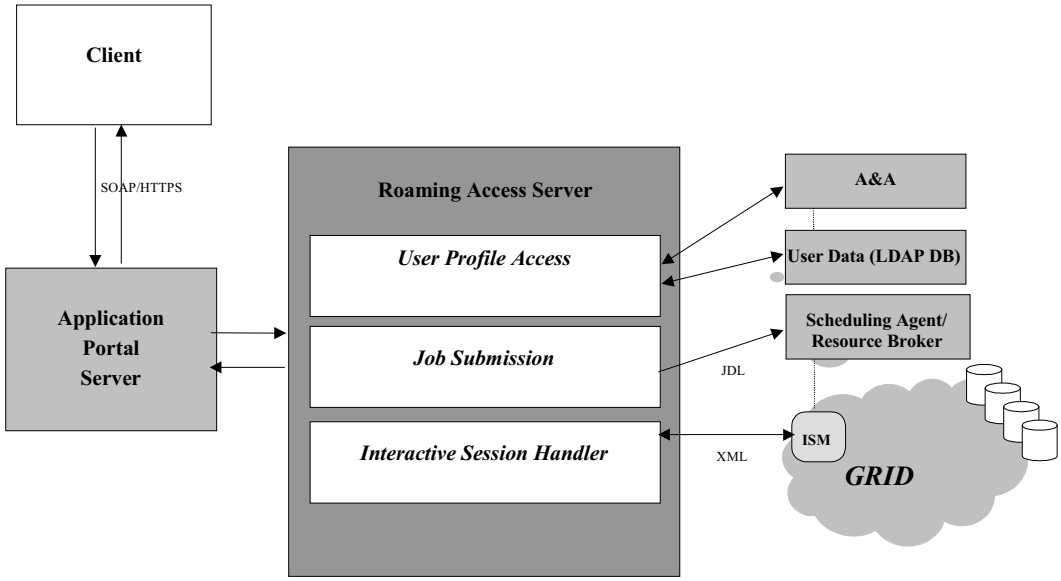


**Fig. 19** Main services of the Roaming Access Server for HEP applications

- *"User Profile Access "* service handles the user login operations to the grid environment and retrieve the User Profile according to their certificate and privileges (step 1); retrieve the list of the DATASETS available for the HEP users (step 2); retrieve the list of the predefined HEP jobs available for the HEP users (step 4).
- *"The Job Submission"* service sends the request for grid resources to work on the selected DATASET to the Scheduling Agent using a high level description language (JDL script) (step 3).
- *"The Interactive Session Handler"* receives the information about the HEP job selected to be executed and related the user input data from the client and submits them in an XML format to the assigned grid resources, represented by a so-called Interactive Session Manager (ISM) (step 5); requests/receives the outputs of the executed HEP job in XML format from the ISM (step 6); maintains the session alive between the Remote Access Server and ISM sending KEEP-ALIVE messages; when requested by the client, sends a CANCEL-CURRENT-PROCESS command to ISM to stop the running HEP job (step 7); when requested by the client, sends a BYE command to ISM to close the current session (step 8); when requested by the client, saves the whole session trace in an XML format (step 8); handles the client request about storing/retrieving the saved session (step 8).

### 2.2.2 Product functions – Migrating Desktop.

Next figure Fig. 20 shows a high level view of the Migrating Desktop

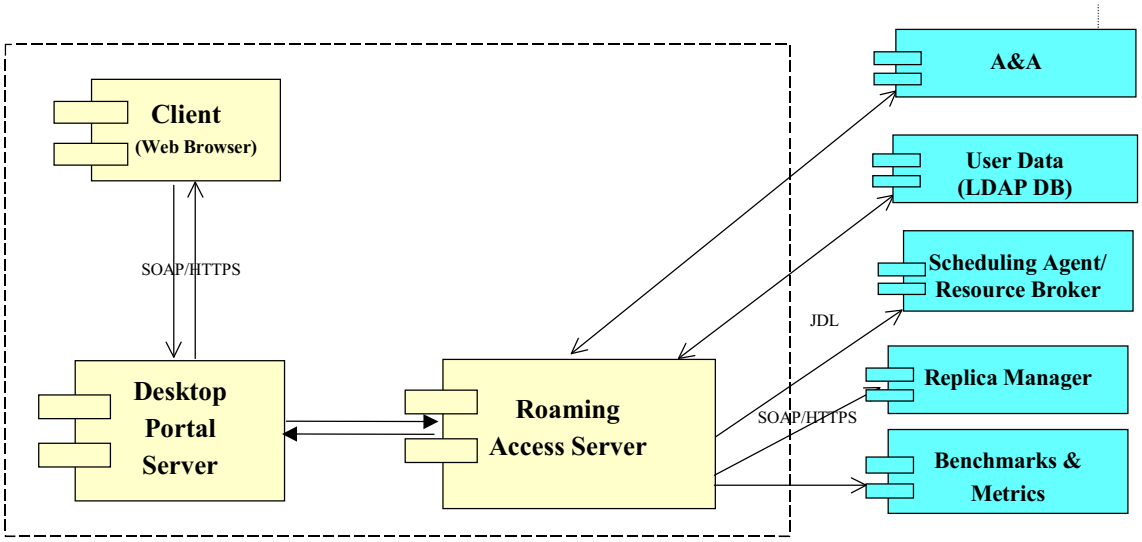


**Fig. 20** High level view of the Migrating Desktop

Referring to Fig. 20, three main blocks can be identified: the Client, the Desktop Portal Server and the Roaming Access Server.

**Client:** Fig. 21 shows a possible layout for the user interface.

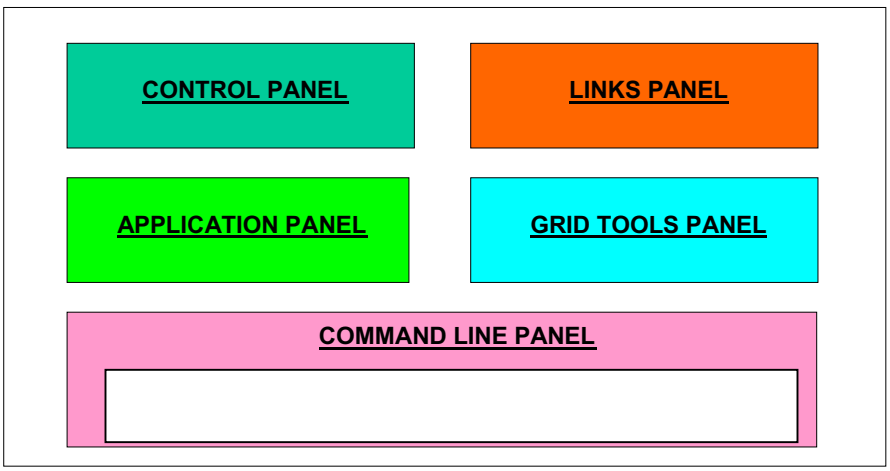


**Fig. 21** General layout of Migrating Desktop.

- *"Control panel"* allows the user to submit the selected jobs; to stop the running jobs; to monitor job status; to manage user profile;
- *"Application Panel"* contains links to chosen by user Grid applications, allows the user to select the job to submit (from links placed on desktop); allows the user to add new links to applications;
- *"Links Panel"* contains links to files, web locations, etc.;
- *"Grid Tools Panel"* contains links to Grid tools (like e.g. GridFTP, Grid Benchmarks, etc.);
- *"Command Line Panel"* gives the user text interface to basic services of Roaming Access Server;

**Application Portal Server:** figure Fig. 22 shows the two main services of the Migrating Desktop

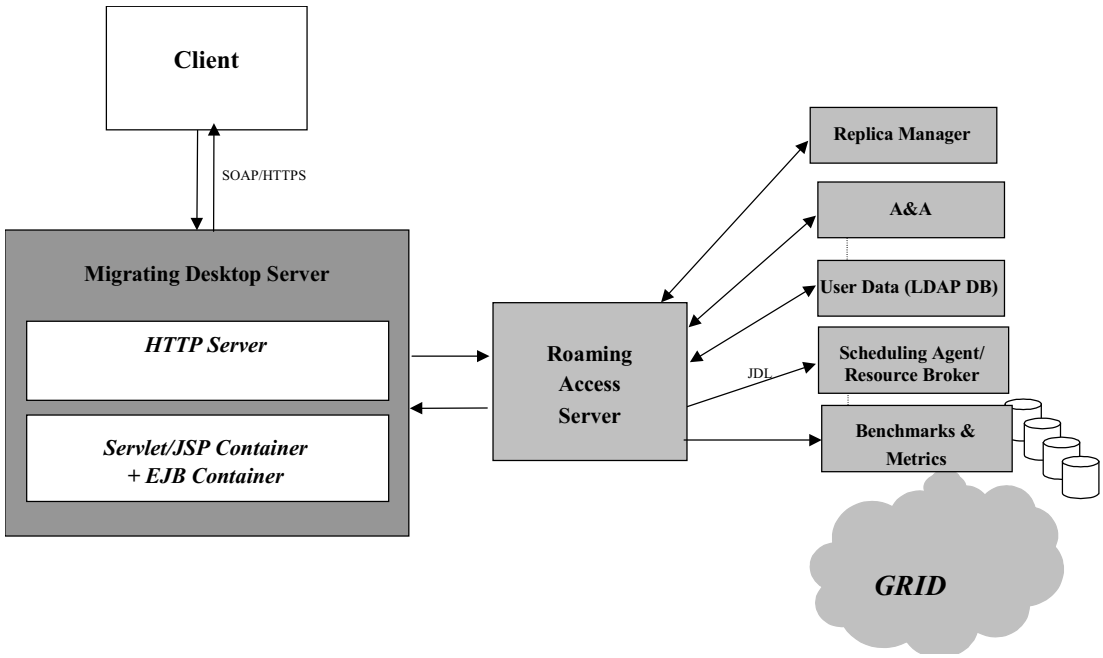


**Fig. 22** Main services of the Migrating Desktop

- *"HTTP Server "* handles connection with Client side by HTTP protocol.
- *"Servlet/JSP Container "* + *"EJB Container"* prepare the pages for the Client; invokes the Roaming Access Service services to send requests and receive data.

**Roaming Access Server:** next figure Fig. 23 shows the three main services of the Roaming Access Server that will be used for Migrating Desktop:

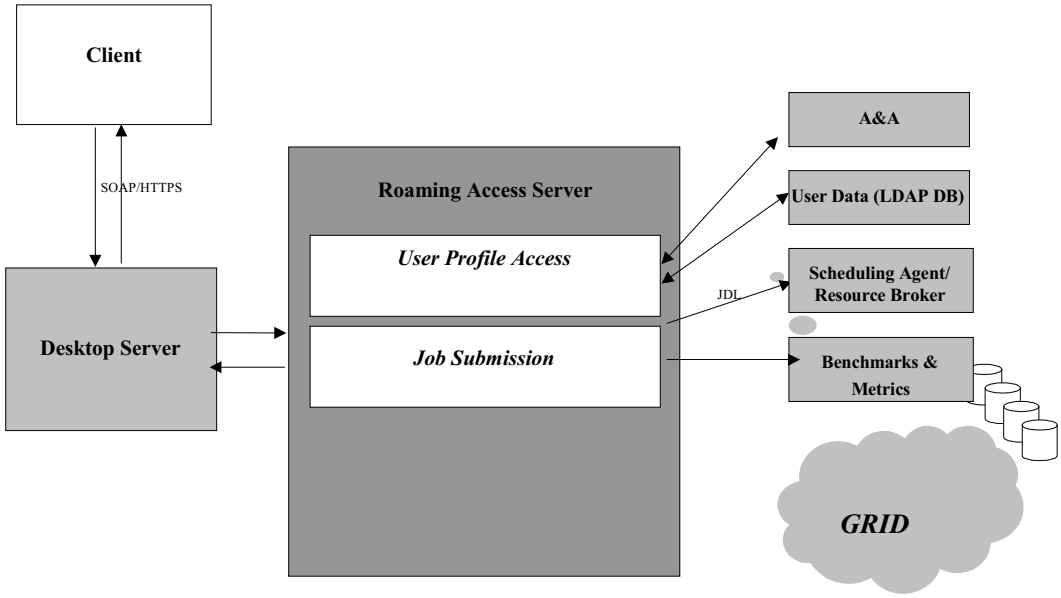


**Fig. 23** Main services of the Migrating Desktop

- *"User Profile Access "* service handles the user login operations to the grid environment and retrieve the User Profile according to their certificate and privileges (step 1); retrieve the user environment;.
- *"The Job Submission"* service sends the request for grid resources to the Scheduling Agent using a high-level description language (JDL script).

## 2.3 USER CHARACTERISTICS

The Grid Desktop is a tool for people - itinerants that use the Grid applications in their everyday work, i. e. scientists physicists, meteorologists, etc. It is not designed for people with special knowledge of computer techniques or technologies, so it should be characterised by user-friendliness.

## 2.4 CONSTRAINTS

TBD

## 2.5 ASSUMPTIONS AND DEPENDENCIES

Application shall be as platform independent as possible. In case of problems that can be solved in a generic way, the client application will be supported at least for platforms:

- Solaris;
- Linux;
- Windows NT/2000;

## 2.6 APPORTIONING OF REQUIREMENTS

TBD

# 3 SPECIFIC REQUIREMENTS

## 3.1 EXTERNAL INTERFACES

In the following, requirements from IF-001 to IF-200 are related to interfaces originating from the client part of the application portal, while requirements from IF-201 to IF-400 are related to interfaces originating from the server part.

| ID | SOURCE | DESCRIPTION |
|---|---|---|
| IF-001 | task 1.3 | In the application portal for HEP applications, the client part shall use XML language over SOAP/HTTPS protocols for the communications to the server part. |
| | | |
| IF-201 | task 1.3 | In the application portal for HEP applications, the server part shall use XML language over SOAP/HTTPS protocols for the communications to the client part. |
| IF-202 | task 1.3 | In the application portal for HEP applications, the server part shall use JDL to submit grid jobs to the Scheduling Agents. |
| IF-203 | task 1.3 | The server part of the application portal for HEP applications shall submit HEP jobs to the assigned computing resources using XML. |
| IF-204 | task 1.3 | The server part of the application portal for HEP applications shall send input data to HEP jobs running on the assigned computing resources using XML. |
| IF-205 | task 1.3 | The server part of the application portal for HEP applications shall send three different control messages to the assigned computing resources: KEEP-ALIVE, CANCEL-CURRENT-PROCESS and BYE. |
| IF-206 | task 1.3 | The server part of the application portal for HEP applications shall send control messages to the assigned computing resources using XML. |
| IF-207 | task 1.3 | The server part of the application portal for HEP applications shall receive output data from HEP jobs running on the assigned computing resources using XML. |
| IF-208 | task 1.3 | The server part of the application portal for HEP applications shall have access to user profile data using LDAP. |
| | | |
| IF-051 | task 1.2 | In the application portal for Flood Crisis Team Support applications, the client part shall use the XML language over SOAP/HTTPS protocols for the communications to the server part. |
| Server part of the application portal shall: | | |
| IF-251 | task 1.2 | Use XML language over SOAP/HTTPS protocols for the |

| | | |
|---|---|---|
| | | communications to the client part. |
| IF-252 | task 1.2 | Use JDL to submit grid jobs to the Scheduling Agents. |
| IF-253 | task 1.2 | Submit jobs to the assigned computing resources using XML. |
| IF-254 | task 1.2 | Receive control messages from the assigned computing resources. |
| IF-255 | task 1.2 | Send control messages to the assigned computing resources |
| IF-256 | task 1.2 | Receive output data from jobs running on the assigned computing resources. |
| IF-257 | task 1.2 | Have access to user profile data using LDAP (?????). |
| IF-258 | task 1.2 | Define visualisation parameters. |
| IF-259 | task 1.2 | To define input data for the selected jobs. |
| | | |
| IF-101 | task 1.4 | In the application portal for 1.4 applications, the client part shall use the XML language over SOAP/HTTPS protocols for the communications to the server part. |
| | | |
| IF-301 | task 1.4 | In the application portal for 1.4 applications, the server part shall use XML language over SOAP/HTTPS protocols for the communications to the client part. |
| IF-302 | task 1.4 | In the application portal for 1.4 applications, the server part shall use JDL to submit grid jobs to the Scheduling Agents. |
| IF-303 | task 1.4 | The server part of the application portal for 1.4 applications shall submit HEP jobs to the assigned computing resources using XML. |
| IF-304 | task 1.4 | The server part of the application portal for 1.4 applications shall send input data to 1.4 jobs running on the assigned computing resources using XML. |
| IF-305 | task 1.4 | The server part of the application portal for 1.4 applications shall receive output data from 1.4 jobs running on the assigned computing resources using XML. |
| IF-306 | task 1.4 | The server part of the application portal for 1.4 applications shall have access to user profile data using LDAP |

TBD

## 3.2 FUNCTIONS

The application portal shall have the capability to retrieve the stored user profile to support the roaming access.

The application portal shall have the capability to have an efficient access to the stored user profile.

In the following, requirements from FUN-001 to FUN-200 are related to the client part of the application portal, while requirements from FUN-201 to FUN-400 are related to the server part.

| ID | SOURCE | DESCRIPTION |
|---|---|---|
| The application portal for HEP applications shall allow the user: | | |
| FUN-001 | task 1.3 | to initiate a secure session logging in using their certificates; |
| FUN-002 | task 1.3 | to submit their HEP jobs; |
| FUN-003 | task 1.3 | to monitor the status of submitted HEP jobs; |
| FUN-004 | task 1.3 | to stop the running HEP jobs; |
| FUN-005 | task 1.3 | to close the current session; |
| FUN-006 | task 1.3 | to save the current session either on a local storage or in the HEP V.O. repository; |
| FUN-007 | task 1.3 | to reload the saved session either from a local storage or from the HEP V.O. repository; |
| FUN-008 | task 1.3 | to edit the reloaded saved session; |
| FUN-009 | task 1.3 | to rerun the saved session; |
| FUN-010 | task 1.3 | to rerun the saved session with changes on the input data; |
| FUN-011 | task 1.3 | to select a HEP job to submit within the list; |
| FUN-012 | task 1.3 | to select the needed data within the selected DATASET to be used as input for the selected HEP jobs; |
| FUN-013 | task 1.3 | to edit the selected HEP jobs; |
| FUN-014 | task 1.3 | to define a HEP job from scratch; |
| FUN-015 | task 1.3 | to define input data for the selected HEP jobs; |
| FUN-016 | task 1.3 | to define data filters for the selected HEP jobs; |
| FUN-017 | task 1.3 | to visualise output data in an alphanumerical format; |
| FUN-018 | task 1.3 | to visualise output data in a graphical format (SVG); |
| FUN-019 | task 1.3 | to send a request for partial output data from HEP jobs running on the assigned computing resources; |
| FUN-020 | task 1.3 | to visualise partial output data in an alphanumerical format, while the submitted job is running; |
| FUN-021 | task 1.3 | to visualise partial output data in a graphical format (SVG), while the submitted job is running; |
| FUN-022 | task 1.3 | to visualise partial output data in alphanumerical format, while the submitted job is running; |
| | | |

| | | |
|---|---|---|
| The application portal for HEP applications shall provide a service: | | |
| FUN-201 | task 1.3 | to authenticate and authorise access to HEP users; |
| FUN-202 | task 1.3 | to retrieve the HEP users stored profile; |
| FUN-203 | task 1.3 | to retrieve the set of authorised DATASETS; |
| FUN-204 | task 1.3 | to retrieve the set of authorised HEP jobs; |
| FUN-205 | task 1.3 | to submit a grid job via the Scheduling Agents using the JDL language; |
| FUN-206 | task 1.3 | to submit a HEP job directly to the assigned computing resources; |
| FUN-207 | task 1.3 | to stop a running HEP job (CANCEL-CURRENT-PROCESS command); |
| FUN-208 | task 1.3 | to establish an interactive session; |
| FUN-209 | task 1.3 | to convert received histograms in XML format in to graphical format (SVG). |
| The application portal for HEP applications shall: | | |
| FUN-210 | task 1.3 | be able to maintain an interactive session alive (KEEP-ALIVE message); |
| FUN-211 | task 1.3 | keep trace of all exchanged messages of an interactive session; |
| FUN-212 | task 1.3 | provide a service to close an interactive session (BYE command); |
| FUN-213 | task 1.3 | be able to save/reload a session trace; |
| FUN-214 | task 1.3 | send a request for output data. |
| The application portal for flood crisis team support applications shall allow the user | | |
| FUN-051 | task 1.2 | To initiate a secure session logging in using certificates. |
| FUN-052 | task 1.2 | To submit jobs. |
| FUN-053 | task 1.2 | To monitor the status of submitted jobs. |
| FUN-054 | task 1.2 | To stop the running jobs. |
| FUN-055 | task 1.2 | To close the current session. |
| FUN-056 | task 1.2 | To select the needed predefined input file among a list. |
| FUN-057 | task 1.2 | To select a job to submit within a list. |
| FUN-058 | task 1.2 | To select the data from repository to be used as input for the selected jobs. |
| FUN-059 | task 1.2 | To define input data for the selected jobs. |
| FUN-060 | task 1.2 | To define visualisation parameters. |
| FUN-061 | task 1.2 | To visualise output data in a graphical format. |
| FUN-062 | task 1.2 | To define data source. |
| The application portal for flood crisis team support applications shall: | | |
| FUN-251 | task 1.2 | Provide a service to authenticate and authorise access. |
| FUN-252 | task 1.2 | Provide a service to retrieve users stored profile. |

| | | |
|---|---|---|
| FUN-253 | task 1.2 | Provide a service to retrieve the set of available historical input files from repository. |
| FUN-254 | task 1.2 | Provide a service to retrieve the set of authorised jobs. |
| FUN-255 | task 1.2 | Provide a service to submit a grid job via the Scheduling Agents using the JDL language. |
| FUN-256 | task 1.2 | Provide a service to submit a job directly to the assigned computing resources. |
| FUN-257 | task 1.2 | Provide a service to stop a running job. |
| FUN-258 | task 1.2 | Provide a service to establish an interactive session. |
| FUN-259 | task 1.2 | Be able to maintain an interactive session alive |
| FUN-260 | task 1.2 | Provide a service to close an interactive session |
| FUN-261 | task 1.2 | Provide a service to make a reservation of the computational power |
| FUN-262 | task 1.2 | Provide a service to inform the user about the state of their jobs by email. |
| The application portal for weather forecast and air pollution modelling applications shall allow the user | | |
| FUN-101 | task 1.4 | To initiate a secure session logging in using their certificates. |
| FUN-102 | task 1.4 | To submit their jobs. |
| FUN-103 | task 1.4 | To monitor the status of submitted jobs. |
| FUN-104 | task 1.4 | To be informed about the presumable time estimation of their operations. |
| FUN-105 | task 1.4 | To close the current session. |
| FUN-106 | task 1.4 | To be informed about events connected with calculations and/or date transmission by emails. |
| FUN-107 | task 1.4 | To select the needed repository of quickly available data among a list. |
| FUN-108 | task 1.4 | To select the needed data within a selected repository of quickly available data to be used as input for the selected jobs. |
| FUN-109 | task 1.4 | To define input data for the selected jobs. |
| FUN-110 | task 1.4 | To define data filters for the selected jobs. |
| FUN-111 | task 1.4 | To visualise output data in a graphical format. |
| FUN-112 | task 1.4 | To save output data on local storage. |
| FUN-113 | task 1.4 | To simultaneously start two different applications in one session and on the same input data. |
| The application portal for weather forecast and air pollution modelling applications shall provide a service | | |
| FUN-301 | task 1.4 | To authenticate and authorise access to users. |
| FUN-302 | task 1.4 | To retrieve users stored profile. |

| | | |
|---|---|---|
| FUN-303 | task 1.4 | To retrieve the set of repository of quickly available data. |
| FUN-304 | task 1.4 | To retrieve the set of authorised jobs. |
| FUN-305 | task 1.4 | To submit a grid job via the Scheduling Agents using the JDL language. |
| FUN-306 | task 1.4 | To submit a job directly to the assigned computing resources. |
| FUN-307 | task 1.4 | To verification of input parameters |
| FUN-308 | task 1.4 | To inform user about the state of their jobs by email. |

### 3.3 PERFORMANCE REQUIREMENTS

TBD

### 3.4 LOGICAL DATABASE REQUIREMENTS

The User Profile information will be stored in the LDAP database. It will be read every time the user logs into the Grid using any of the clients. Writing the graphical part of the user profile will be done when logging off from the Grid. Writing the rest of the user profile will take place after changing the user configuration. The LDAP server should be accessible from every point of the Grid.

TBD.

### 3.5 DESIGN CONSTRAINTS

The application portal shall be integrated and tested with WP3 software.

### 3.6 STANDARDS COMPLIANCE

TBD

### 3.7 SOFTWARE SYSTEM ATTRIBUTES

TBD

# 4 APPENDIXES

## 4.1 STATE OF THE ART – FILE: *CG-3.1-SRS-0020-STATEOFTHEART.DOC*

Abstract: This document gives a brief survey of projects and works related to the problems and describes the current state of the art in the scope of giving the end user easy and simple access to resources via a graphical interface. The main goal of this additional software layer positioned between the end user and middleware and applications is to decrease the knowledge required to access and use resources and data in a heterogeneous distributed environment. Portal and vortals can be a simple example of this layer.

## 5 INDEX

| | | |
|---|---|---|
| WP3.1, 5 | | **X** |
| | XML, 5 | |