



# DELIVERABLE D5.2.2 CROSSGRID ARCHITECTURE REQUIREMENTS AND FIRST DEFINITION OF ARCHITECTURE

## WP5, Task 5.2 Architecture Team

---

Document Filename:	<b>D5.2.2-CGArchitecture-FINAL</b>
Work package:	<b>WP5</b>
Partner(s):	<b>CYFRONET</b>
Lead Partner:	<b>CYFRONET</b>
Config ID:	<b>CG-5-D5.2.2-0009-1-0-FINAL</b>
Document classification:	<b>PUBLIC</b>

---

Abstract: This document presents an overview of the general architecture of the CrossGrid Project with respect to its individual Work Packages (WPs).

the following papers are integral parts of this deliverable:

- *Overview of Common Component and Grid Services Architectures*
- *Review of Technologies for Interactive and Distributed Simulation on the Grid*
- *Software Engineering Tools*
- The Architecture Team's SRS review



### Delivery Slip

	Name	Partner	Date	Signature
<b>From</b>				
<b>Verified by</b>				
<b>Approved by</b>				

### Document Log

Version	Date	Summary of changes	Author
1-0-DRAFT-A	24/5/2002	Draft version	CrossGrid Architecture Team (Marian Bubak, Katarzyna Zając, Maciej Malawski, Marek Garbacz, Piotr Nowakowski)
1-0-FINAL	05/06/2002	Final Version	CrossGrid Architecture Team (Marian Bubak, Katarzyna Zając, Maciej Malawski, Marek Garbacz, Piotr Nowakowski)

---

## CONTENTS

<b>1</b>	<b>LIST OF ACRONYMS USED.....</b>	<b>5</b>
<b>2</b>	<b>INTRODUCTION.....</b>	<b>6</b>
<b>3</b>	<b>MAIN OBJECTIVES OF THE CROSSGRID PROJECT.....</b>	<b>7</b>
<b>4</b>	<b>KEY APPLICATION REQUIREMENTS.....</b>	<b>8</b>
4.1	BIOMEDICAL APPLICATION .....	8
4.2	FLOODING CRISIS TEAM SUPPORT APPLICATION .....	8
4.3	DISTRIBUTED DATA ANALYSIS IN HEP .....	9
4.4	WEATHER FORECASTING AND AIR POLLUTION MODELLING .....	10
<b>5</b>	<b>ARCHITECTURE OVERVIEW.....</b>	<b>11</b>
<b>6</b>	<b>APPLICATION-SPECIFIC GRID SERVICES.....</b>	<b>13</b>
6.1	USER INTERACTION SERVICES .....	13
6.2	INTERACTIVE SESSION SERVICES .....	14
6.3	GRID VISUALIZATION KERNEL .....	14
<b>7</b>	<b>GENERIC CROSSGRID SERVICES.....</b>	<b>15</b>
7.1	SCHEDULER .....	15
7.2	OPTIMIZATION OF DATA ACCESS.....	15
7.3	GRID MONITORING .....	16
7.4	ROAMING ACCESS .....	17
<b>8</b>	<b>FABRIC COMPONENTS.....</b>	<b>18</b>
8.1	OPTIMIZATION OF LOCAL DATA ACCESS .....	18
8.2	RESOURCE MANAGER FOR INSTRUMENTS .....	18
<b>9</b>	<b>TOOLS .....</b>	<b>19</b>
9.1	TOOLS FOR APPLICATION DEVELOPMENT.....	19
9.2	PORTAL AND MIGRATING DESKTOP.....	19
<b>10</b>	<b>GRID ORGANIZATION FOR SPECIFIC APPLICATIONS.....</b>	<b>20</b>
10.1	BIOMEDICAL APPLICATION .....	20
10.2	FLOODING CRISIS TEAM SUPPORT APPLICATION .....	21
10.3	DISTRIBUTED DATA ANALYSIS IN HEP .....	22
10.4	WEATHER FORECASTING AND AIR POLLUTION MODELING .....	23
<b>11</b>	<b>GENERAL RULES FOR CROSSGRID SOFTWARE DEVELOPMENT.....</b>	<b>24</b>
<b>12</b>	<b>CONCLUDING REMARKS.....</b>	<b>25</b>
<b>13</b>	<b>REFERENCES.....</b>	<b>26</b>



**CROSSGRID ARCHITECTURE REQUIREMENTS AND  
FIRST DEFINITION OF ARCHITECTURE  
Working Draft**

---

## 1 LIST OF ACRONYMS USED

API	Application Programmer's Interface
CES	Component Expert System
DBMS	Database Management System
EDG	European DataGrid
GIS	Globus Information System
GRAM	Globus Resource Allocation Manager
GSI	Grid Security Infrastructure
HEP	High-Energy Physics
HLA	High-Level Architecture
HPC	High-Performance Computing
HSM	Hierarchical Storage Management
LHC	Large Hadron Collider
MDS	Metacomputing Directory Service
MPI	Message Passing Interface
NN	Neural Network
PAW	Physics Analysis Workstation
PIII	Pentium III
SA	Scheduling Agents
TRLFM	Tape Resident Large Files Middleware
WP	Work Package

## 2 INTRODUCTION

This document begins with a synthesis of all the requirements stated by all applications of the Project. Basing on these requirements we propose the first definition of the CrossGrid architecture. This definition consists of descriptions of functionality of all components and indicates where interfaces should be defined. It ought to be perceived as a starting point for a more detailed description of interfaces and technology used for their implementation. The components of the CrossGrid architecture are modular and they are organized in the following six layers (similar to the approach presented by I. Foster and C.Kesselmann [GridNCI]):

- Applications,
- Supporting Tools,
- Application Support,
- Application Specific Grid Services,
- Generic Services,
- Fabric.

### 3 MAIN OBJECTIVES OF THE CROSSGRID PROJECT

According to the Technical Annex, the applications of the CrossGrid Project are characterized by interaction with a person in a processing loop. Each requires a response from the Grid to an action by that person in different time scales; from real time through intermediate delays to long waiting periods, and they are simultaneously compute- and data-intensive. The following applications are being developed:

- interactive simulation and visualization for surgical procedures,
- flooding crisis team decision support systems,
- distributed data analysis in high-energy physics,
- air pollution combined with weather forecasting.

Development of this category of applications for the Grid requires new tools for verification of parallel source code, performance prediction, performance evaluation and monitoring.

Next, both applications and tools need new Grid services for application performance monitoring, efficient distributed data access, and specific resource management. Users should be able to run their applications on the Grid in an easy and transparent way, without needing to know any details of the Grid structure and operation. CrossGrid will develop user-friendly portals and roaming personalized environments.

The CrossGrid development exploits all the available achievements of Globus, DataGrid and other related projects in a way which enables their interoperability. The elaborated methodology, generic application architecture, programming environment, and new Grid services will be validated and tested thoroughly on the testbeds.

## 4 KEY APPLICATION REQUIREMENTS

Below we present key requirement formulated by the participants of Work Package 1. More detailed information about basic features of these applications, the required computer power, the character of computation (High Throughput Computing, High Performance Computing) main memory requirements and secondary storage capacity requirements are given in the respective Software Requirements Specification documents, which were prepared according to the template adapted by the Technical Architecture team.

### 4.1 BIOMEDICAL APPLICATION

The application will be a distributed near real-time simulation environment where a user interacts in Virtual Reality (VR) and other interactive display environments [WP1.1]. A 3D model of arteries will be the input to a simulation for blood flow and the results will be presented in a VR environment. The user will change the structure of the arteries, thus mimicking an interventional or surgical procedure. The effects will be analysed in near real-time and the results will be presented to the user in VR. The software will consist of several components, which may be executed at different locations on different hardware platforms:

- Medical scanners (data acquisition systems – from outside CG),
- Segmentation software for medical 3D images (outside CG),
- Database with scanned medical images and metadata (interfaces with CG),
- Blood flow simulator with interaction capability (developed mostly outside CG; to be integrated),
- History database,
- Visualization software, suitable for several interactive 3D visualization platforms,
- Interactive measurement facilities for the analysis and quantification of simulation results with an interactive display environment,
- Interaction module, allowing the user to adjust simulation and display parameters,
- Interface modules, to facilitate coupling of visualization, interaction and simulation, and to regulate data traffic between the various modules.

### 4.2 FLOODING CRISIS TEAM SUPPORT APPLICATION

This application will be a support system for a Virtual Organization for flood forecasting [WP1.2]. Grid technology will be applied to seamlessly connect together experts, data and computing resources required for quick and correct flood management decisions. The main component of the system will be a highly automated early warning system based on hydro-meteorological (snowmelt) rainfall-runoff simulations. The experts will be able to run the simulations with changed parameters and analyze the impact (what-if analysis).

Flood forecasting requires quantitative precipitation forecasts based on meteorological simulations of different resolutions, from mesoscale to storm-scale. Next, the hydrological models are used to determine the discharge from the affected area, and with this information hydraulic models simulate flow through various river structures to predict the impact of the flood. The results of simulations can be interactively reviewed, some of them accepted and forwarded to the next step, some rejected or re-run with modified inputs.

The system will consist of

- data sources (radar and satellite images, topographical data, permanent and temporary storage management (operational database, archive database),
- meteorological simulation models,
- hydrological simulation models,

- hydraulic simulation models,
- visualization tools (2D and also 3D).

The meteorological models are generally computationally intensive and they usually run on supercomputer systems. The output of mesoscale models is used by meteorologists as a set of boundary conditions for regional models. This requires efficient management and transfers of large (tens to hundreds of megabytes) datasets.

In hydrological models, component representation requires a set of parameters to specify the particular characteristic of the component and mathematical relations to describe the physical process. The result of the modelling process is a set of stream-flow hydrographs at desired locations in the river basin. Due to the next step in the simulation cascade being hydraulic simulation, in which more tributaries are taken into account, several independent hydrological simulations are performed, i.e. high throughput computing is applicable.

Hydraulic models simulate the propagation of flood waves through various river structures. They can be one-dimensional or two-dimensional. Two-dimensional models are generally more accurate than one-dimensional models, however they are computationally intensive. The size of input/output data for two-dimensional models depends on the density of meshes and the number of time steps and can occasionally exceed a hundred megabytes. The size of input/output data for one-dimensional models is usually several dozen kilobytes.

The models used in flood forecasting require a huge amount of temporary storage as they produce massive output sets. There is also a need for storing several (possibly hundreds of) pre-simulated situations, in order to reduce the response time in critical situations. All of this data will be well-described by an elaborate metadata structure.

The results obtained from the meteorological or hydraulic models are quite useless unless well visualized by an elaborate tool. This application requires the visualization tool to display meteorological forecasts, hydrological status of target areas, and estimated water flows, along with endangered areas.

### 4.3 DISTRIBUTED DATA ANALYSIS IN HEP

This application [WP1.3] will address

- access to large distributed databases in the Grid environment,
- development of distributed data-mining techniques suited to the HEP field.

In the development of this HEP Interactive Data Analysis application two approaches will be considered: one based on PAW and the Object Oriented Framework and one based on an O/R DBMS.

The application will have a layered structure, flexible enough to adapt to different experimental set-ups and integration of user-friendly interactive access. It will consist of:

- Interactive Session Distributed Database Access Service (ISDB),
- Distributed Database Installation, Replication and Stripping Service (DBINST),
- Interactive Session Distributed Processing (ISDP) implemented on top of Interactive Session Distributed Manager and Worker (ISM, ISW) using MPI or similar message passing interface,
- Interactive Session Data Mining Service (ISDM), including Neural Network Algorithms.

In the interactive analysis in HEP, data location plays a central role: typical interactive requests will run on O(TB) of distributed data, incurring hour-long transfer/replication times for the whole dataset (1 TByte is transferred in >8000 s. over a Gigabit network). So this operation, if needed, should take place once and in advance to the interactive session.

Data mining services based on supervised and unsupervised learning and two solutions processing in worker nodes and on the DBMS side will be elaborated. At first, the process involves several training steps, where most of the time is spent to optimise a neural network. A typical NN with a 10-15-15-1 structure took somewhere between several hours to several days to train with 1MB events on a PIII 600 MHz. In an interactive work with datasets typical for LHC, the objective of training the NN in a few minutes results in the distribution of the work

across  $O(10-100)$  Grid nodes. Optimization of resources may be based on a compromise between node processing power and the overhead involved in distributed information exchange over a network. The NN algorithm scales perfectly from 2 to 100 nodes in a local cluster with fast Ethernet.

#### 4.4 WEATHER FORECASTING AND AIR POLLUTION MODELLING

This application [WP1.4] includes

- sea wave modelling with the Coupled Ocean/Atmosphere Mesoscale Prediction System,
- data mining applied to downscaling weather forecasts,
- an implementation of air-pollution models.

The application is data-intensive and performance-intensive which requires efficient management and transfer of gigabytes of information throughout a wide area. A large community of researchers shares massive data sets and institutions often create local copies and replicas of the simulated and observed data sets to overcome long wide-area data transfer latencies. The data management should not only provide security services, but also contain tools able to determine where all the existing copies are located, to enable a user to decide whether to access existing copies or create new ones to meet the performance needs of their applications.

The weather and sea wave forecasting requires

- an atmospheric, weather prediction model,
- various sea wave models,
- visualisation tools (like GRADS and VIS5D).

Data mining requires

- distributed meteorological databases,
- statistical packages with specific nonlinear data mining techniques on modeled and observed data archives,
- visualisation tools.

Air pollution simulation requires

- an air pollution sulfur transport model (STEM II),
- services which allow the user to change the set of parameters driving the model according to the results of simulations,
- visualisation tools.

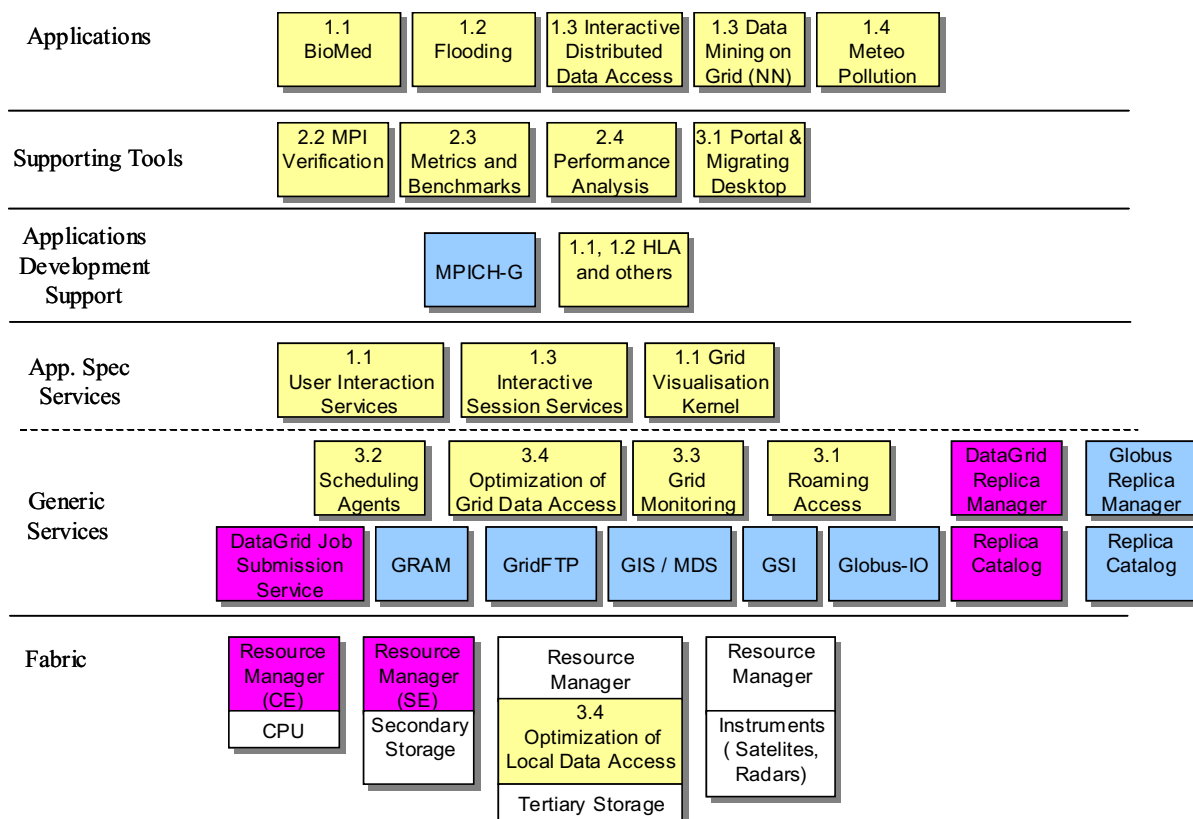
This application also requires services which allow users to check the status of their own pending jobs, to use results from similar, earlier runs and to retrieve job output

## 5 ARCHITECTURE OVERVIEW

After analysing the preliminary version of the application Software Requirements Specification documents [CrossGrid] the CrossGrid Technical Architecture Team has elaborated the first version of the CrossGrid architecture, presented at the first meetings of Project Work Packages. The draft was subsequently modified and improved.

Fig. 1 presents the current version of the general overview of the CrossGrid architecture. The software components which will be developed within the framework of the Project and those from Globus and DataGrid are organized in layers similarly to the approach presented by I. Foster and C.Kesselmann [GridNCI]

- Applications – interactive applications, the basic characteristics of which were given in the previous chapter,
- Supporting Tools – for development of applications and the portal with a migrating desktop,
- Application Support – two components used for organization of communication,
- Application-Specific Grid Services – new services which enable interactive execution of applications on the Grid, designed for specific CrossGrid applications,
- Generic Services – new Grid services with universal features,
- Fabric – for optimization of local data access and for instrument management.



**Figure 1: Overview of the CrossGrid Architecture.**



**CROSSGRID ARCHITECTURE REQUIREMENTS AND  
FIRST DEFINITION OF ARCHITECTURE  
Working Draft**

---

All applications of the CrossGrid require authentication and authorization as every action taken either by user (through the portal), or by any software component has to be secure. Therefore we will use Globus and DataGrid security systems as essential components of the CrossGrid architecture.

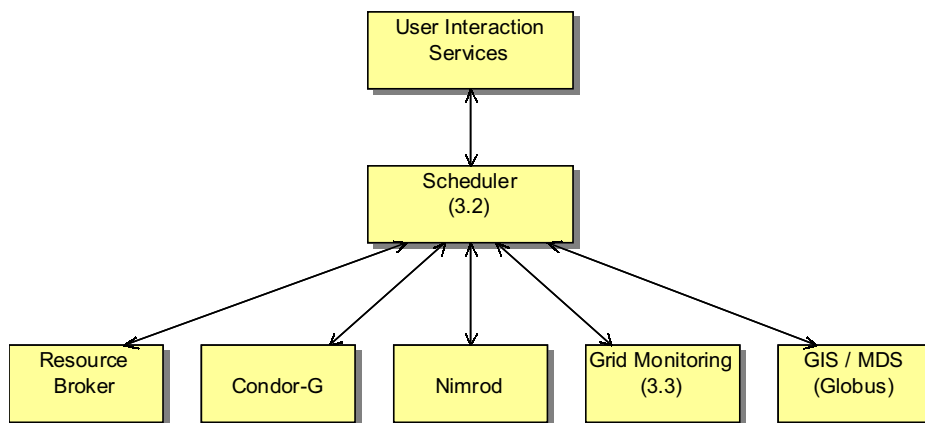
The five following sections describe the functionality of each component and its interaction with other grid services and tools. For a description of Globus [Globus] and Datagrid [EDG] please refer to the appropriate websites.

## 6 APPLICATION-SPECIFIC GRID SERVICES

### 6.1 USER INTERACTION SERVICES

The User Interaction Services should enable control of data flow between various Grid components connected together for a given interactive application. It should also synchronize the simulation and visualization, and – according to the required type of simulation – this service should enable a plug-in mechanism for the required components. Software interfaces to the User Interaction Services are presented in Fig. 2. Together with the CrossGrid scheduler, this module will cooperate with underlying resource managers and information services:

- Resource broker (both DataGrid and Globus),
- Condor-G system for High Throughput Computing,
- Nimrod tool for advanced parameter study,
- Grid monitoring service and Globus GIS/MDS.



**Figure 2: Interaction of User Interaction Services with other grid components.**

The User Interaction Services should make it possible to start complex interactive applications and to steer the simulations. During the simulation the user should be able to cancel a given job as a result of an inspection of the job output. The interactive application should be re-started immediately order to achieve a response time as low as possible. The service should enable an advance reservation and the use of priorities with regard to user jobs and job pre-emption appear to be valuable mechanisms to support this requirement.

The User Interaction Services module will cooperate with the HLA module, the main tasks of which will be to:

- maintain a preselected ordering of simulation events,
- provide specific communication primitives.

We are currently investigating two solutions:

- porting the HLA system to the Grid environment,
- developing a Grid-oriented module which would encapsulate HLA functionality.

## 6.2 INTERACTIVE SESSION SERVICES

This component is a specific type of User Interaction Services for HEP applications. Its main modules are:

- An Interactive Session Distributed Database Access Service which provides actual access to the data. XML will be used for the queries and the result set,
- Interactive Session Workers for interactive data analysis and data mining in a distributed environment (with MPI); they will control work distribution,
- a Distributed Database Installation, Replication and Stripping Service,
- an Interactive Session Data Mining Service.

## 6.3 GRID VISUALIZATION KERNEL

The Grid Visualization Kernel (GVK) will address the problem of interconnecting distributed simulation sources with visualisation clients. The idea of the GVK is to provide a middleware layer extension for scientific visualisation, which allows interactive, near real-time visualization for Grid applications on arbitrary visualization devices. The GVK consists of three distinct modules:

1. An input interface for delivering the simulation data to the GVK.
2. An output interface for delivering the visualisation data from the GVK to the output device.
3. The GVK itself, which connects the simulation and the visualisation via their respective interfaces.

---

## 7 GENERIC CROSSGRID SERVICES

### 7.1 SCHEDULER

The CrossGrid resource management system will be based on self-adaptive scheduling agents. The main goal is to schedule distributed applications submitted to the Grid. The scheduling agents (SA) will decide where, when, and how to run parallel jobs on Grid resources as specified by a set of policies, priorities, requirements and limits. The scheduler is responsible to deciding on the allocation of application tasks to computing resources in a way which guarantees that applications are conveniently, efficiently and effectively executed. This scheduler will be an extension of the DataGrid Workload Manager System. SA will automate most of the activities related to resource discovery and system selection. SA will schedule and manage two types of jobs:

- Simple jobs consisting of single applications (sequential or parallel) ,
- Composite jobs consisting of collections of single applications (each of which can be either sequential or parallel) which exhibit some sort of inter-dependency.

The applications will be described using a specific job description language.

The functionality of the CrossGrid scheduler should include:

- Matching the user application requirements and the resources available on the Grid (the characteristics of Grid resources will be retrieved from both monitoring and information services).
- Resource selection, which will be implemented with heuristics responsible for selecting the actual set of resources to be used by an application from a pool of all resources which fulfill that application's requirements. The selection criteria will be guided by the user-specific optimization criteria.
- A submission service, which performs the actual submission of a job to the resource manager of the Grid. If required, the resource manager should guarantee a simultaneous co-allocation of all the resources determined by the resource selection process.

Software interfaces to the scheduler are presented in Fig. 2. As previously stated, the scheduler will cooperate with

- Resource broker (both DataGrid and Globus),
- Condor-G system for High Throughput Computing,
- Nimrod tool for advanced parameter study,
- Grid monitoring service and Globus GIS/MDS.

### 7.2 OPTIMIZATION OF DATA ACCESS

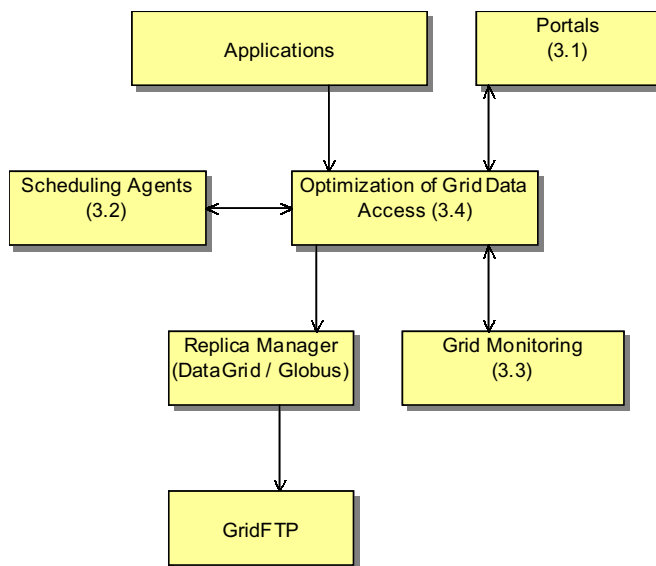
The Optimization of Data Access Service consists of a Component-Expert System, a Data-Access Estimator and a GridFTP Plugin.

The Component-Expert Architecture is a heuristic system for selecting of components (data handlers), which allow flexible data access. It optimizes the usage of the storage systems through proper selection of storage strategies due to different applications' requirements and existing storage capabilities. A human expert should prepare the decision rules.

The Data Access Estimator estimates the bandwidth and the latency of data availability inside a Storage Element. This estimation does not include network transmission aspects. This anticipation is used by the Replica Manager to make informed decisions. The estimation of the data-arrival latency is different for HSM, TRLFM, Secondary Storage and relational databases.

The GridFTP Plugin is designed to expand the capabilities of the GridFTP server, allowing the usage of CES.

The interaction of the Optimisation of Data Access Service with other CrossGrid components is presented in Fig. 3.



**Figure 3: Interaction of Grid Data Access with other Grid components**

### 7.3 GRID MONITORING

Grid monitoring facilities within the CrossGrid work by delivering the low-level data intended for automatic extraction of high-level performance properties and on-line tool support, oriented towards application-bound activities, as well as data for the needs of network-oriented activities on the Grid.

The monitoring system consists of:

- an OMIS-based application monitoring system, OCM-G,
- additional services, (SANTA-G), for non-invasive monitoring, and
- Jiro-based services for Grid-infrastructure monitoring.

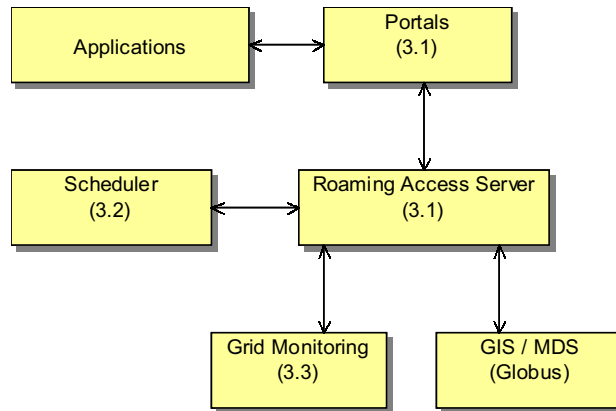
OCM-G is a distributed monitoring system for obtaining information on and manipulating parallel distributed applications. The purpose of this system is to provide a basis for constructing tools which support parallel application development. The benefit of using it is that it constitutes an autonomous monitoring infrastructure, accessible via a standardized interface, which various tools can use. This approach provides abstraction and increases modularity – the tools themselves can be developed independently from the monitoring system.

The Jiro-based services for Grid infrastructure monitoring are intelligent components for obtaining information from and Grid hardware devices. The application of this software is to gather information from hardware devices, make autonomous decisions based on this information and take necessary actions.

SANTA-G services are a specialized non-invasive complement to other more intrusive monitoring services. The objectives are to allow information captured by external monitoring instruments to be introduced into the Grid information system, and to support analysis of performance using this information.

The monitoring system will be developed in such a way that it will be scalable, flexible and easy to configure. The CrossGrid monitoring system should have interfaces to all application-specific services, to Grid generic services and to tools developed within the framework of this Project (see Figs. 2, 3, 4 and 5).

## 7.4 ROAMING ACCESS



**Figure 4: Interaction of Roaming Access Server with other Grid components**

The services that will allow the user to access his environment from remote computers consist of a Migrating Desktop, the application portals and the Remote Access Server. The Remote Access Server manages user profiles, user authorization and authentication and job submission. These features are available through a user interface.

## **8 FABRIC COMPONENTS**

### **8.1 OPTIMIZATION OF LOCAL DATA ACCESS**

The software which enables optimization of access to local data is called the Tape Resident Large Files middleware. This middleware is placed on top of an existing HSM system. It is intended to speed up read access to large files (>100MB) stored on tapes, by means of reducing latency times.

### **8.2 RESOURCE MANAGER FOR INSTRUMENTS**

Within the framework of the CrossGrid Project we consider developing an application-oriented Resource Manager for instruments like radars and satellites providing input for the flood crisis team support application.

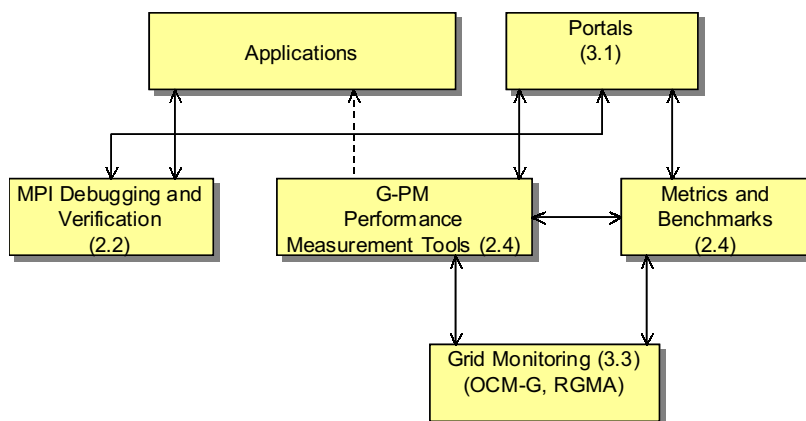
## 9 TOOLS

### 9.1 TOOLS FOR APPLICATION DEVELOPMENT

The CrossGrid application programming environment consists of the following tools:

- an MPI code debugging and verification tool to detect and report erroneous use of MPI communication functions,
- a Grid benchmark suite to gauge the performance of a given Grid configuration,
- a performance evaluation tool to support measurement and prediction of an application's performance on the Grid.

The possible interfaces between this tool environment and other Grid services are given in Fig. 5.



**Figure 5: The tool environment and its interaction with Grid service components.**

### 9.2 PORTAL AND MIGRATING DESKTOP

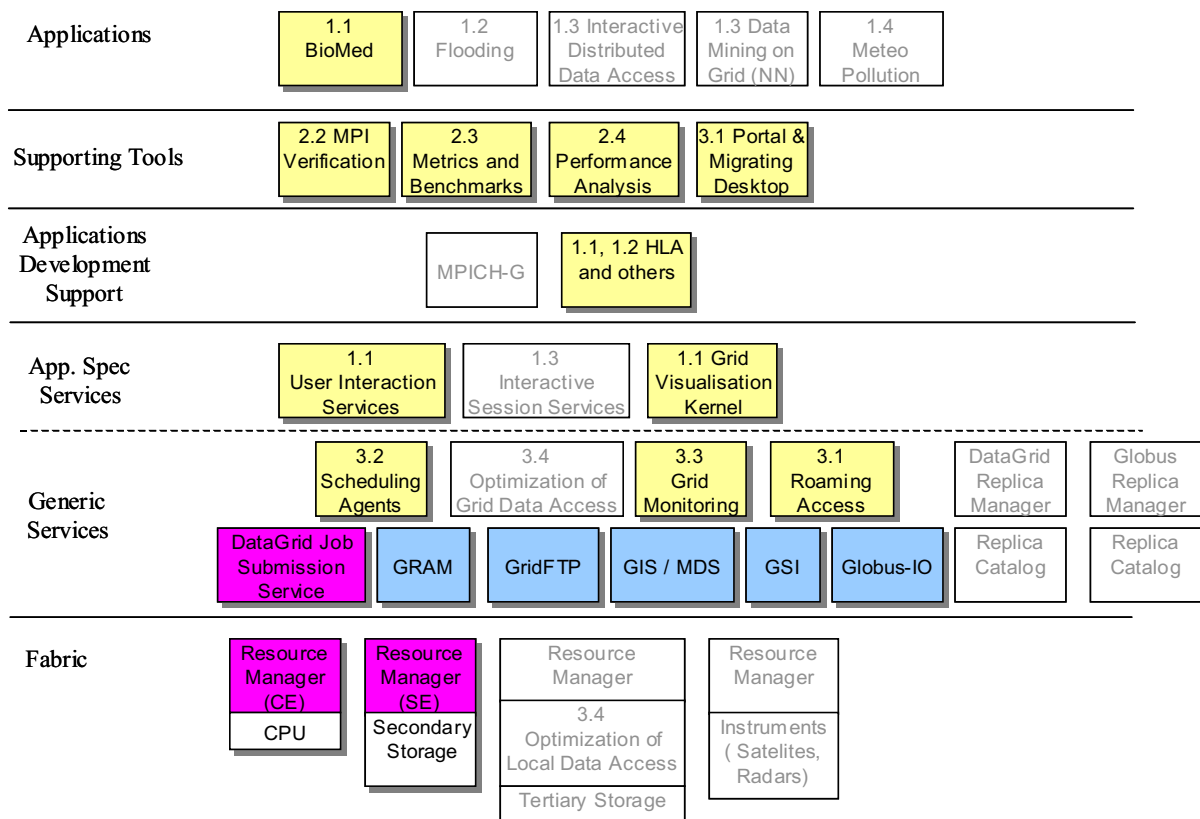
The Portal will provide a unified and consistent window on the CrossGrid environment. The Grid tools will be integrated into the portal providing easier access for users of the Grid applications. The CrossGrid portals will securely authenticate and authorize users and it will allow them to view pertinent resource information obtained and stored in a remote database. The Migrating Desktop will offer a transparent user work environment, independent of the system version and hardware and enable users to access Grid resources and local resources from remote computers independently of their localization and terminal type.

## 10 GRID ORGANIZATION FOR SPECIFIC APPLICATIONS

The figures in the following subsections present the configuration of the Grid services and tools for each application of the Project.

### 10.1 BIOMEDICAL APPLICATION

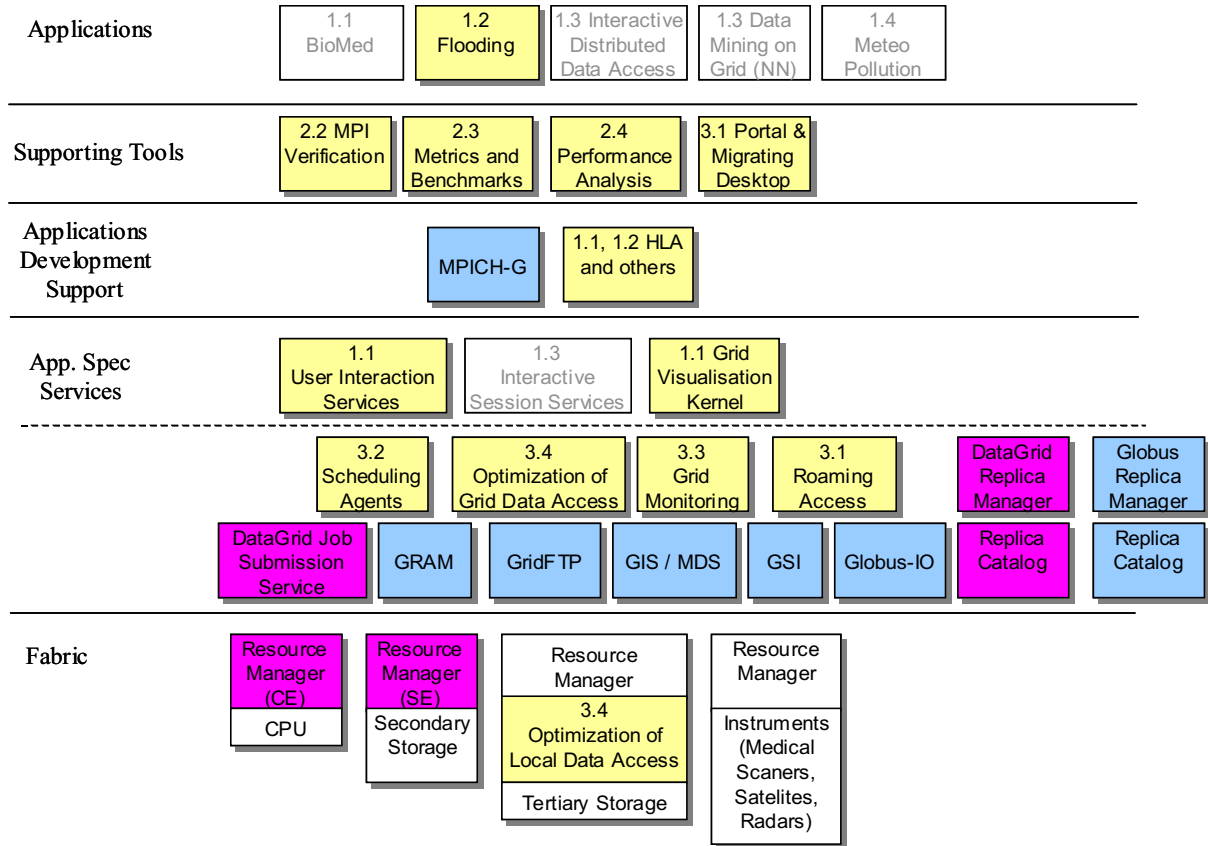
The biomedical application is performance-sensitive in that it implies real-time interaction with users of the various visualization devices. Most of the tools required in this application is being developed insitu by the WP1 team. Grid interaction will therefore be limited to the portal and various supporting tools. The WP1 team is also developing the Grid Visualisation Kernel (see section 5.3). Because of performance concerns, individual work units will run on single, selected sites, hence job distribution via MPICH-G is not required.



**Figure 5: Component dependencies for WP1.1.**

## 10.2 FLOODING CRISIS TEAM SUPPORT APPLICATION

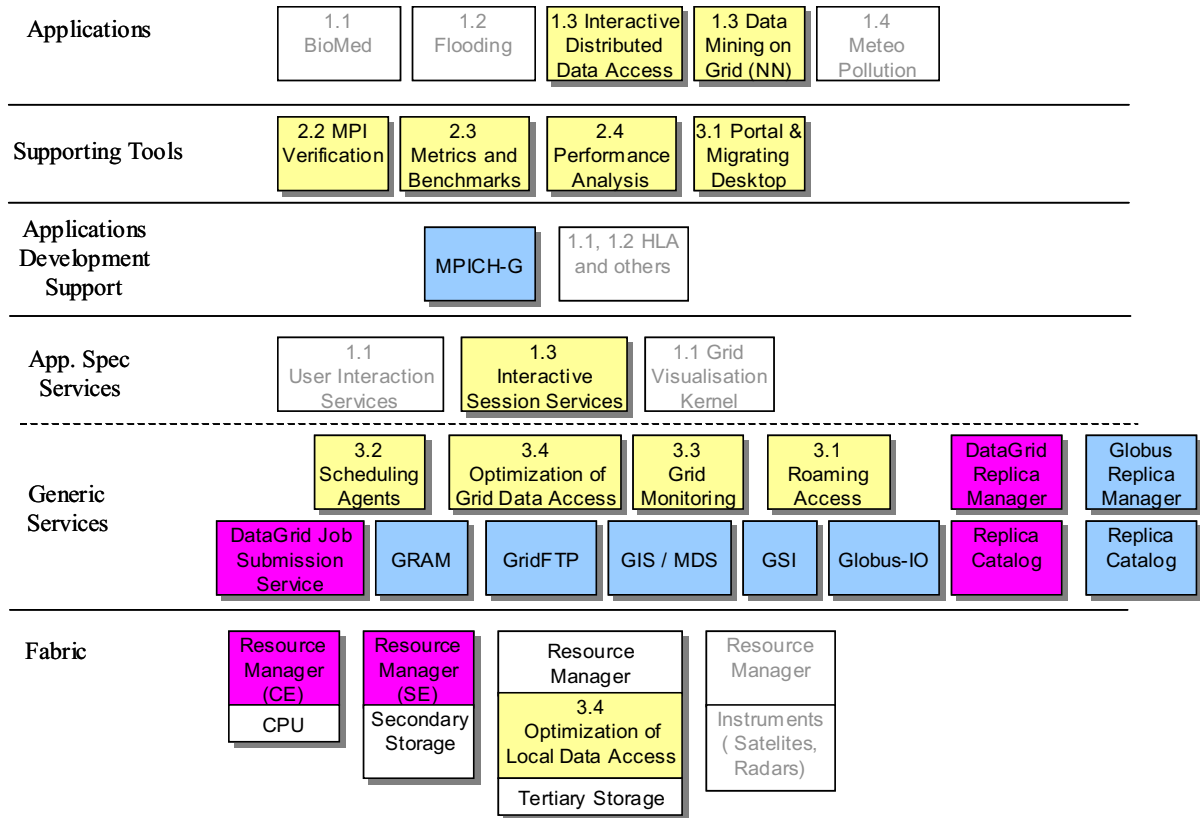
Like all other applications, the flooding app will make use of all Grid supporting tools. The Grid Visualisation Kernel may also be used to handle graphics.



**Figure 6: Component dependencies for WP1.2.**

### 10.3 DISTRIBUTED DATA ANALYSIS IN HEP

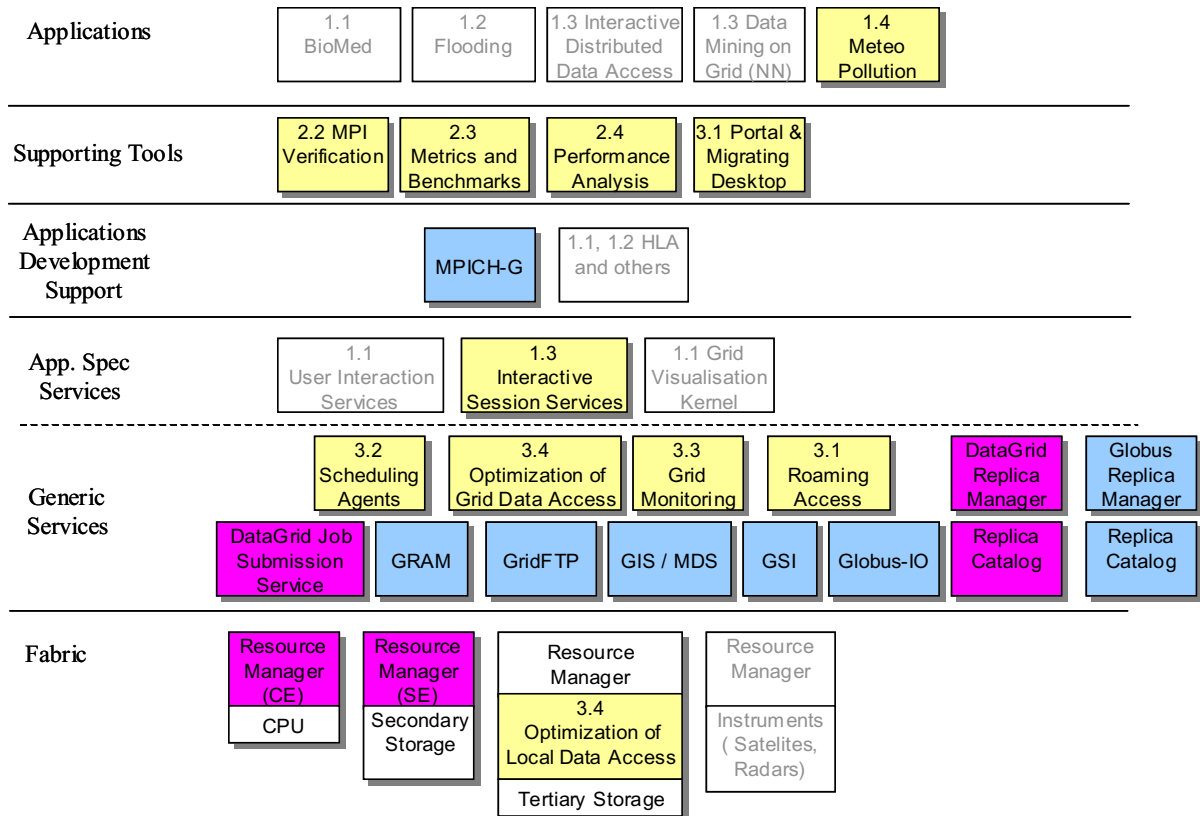
The High Energy Physics application places a heavy burden on both computing power and network throughput. A special, dedicated set of user interaction services will be developed for this application (see section 5.1). HLA will not be used here and neither will the Grid Visualization Kernel.



**Figure 7: Component dependencies for WP1.3.**

### 10.4 WEATHER FORECASTING AND AIR POLLUTION MODELING

WP1.4 shares the Interactive Services module with WP1.3 as both applications are being developed in cooperation. HLA is not required in this instance.



**Figure 8: Component dependencies for WP1.4.**

## 11 GENERAL RULES FOR CROSSGRID SOFTWARE DEVELOPMENT

All components of the CrossGrid software will be developed according to the following rules:

- Iterative improvement consisting of fast prototyping, design, implementation, testing on testbeds, evaluation and further improvement,
- Modularity of all components,
- Open source approach with relation to the Grid services and tool environment,
- Precise software documentation,
- Close collaboration with other Grid projects.

All software will be produced according to the evolutionary life-cycle model with well established phases of specification, development and validation. The key issue in this approach is collecting and improving the Software Requirements Specifications which are the starting point for the design of software architecture (on a high level), detailed design, implementation, and testing.

## **12 CONCLUDING REMARKS**

This set of documents presents an overall view of the CrossGrid architecture, its components, their functionality and relations between them as well as with components from Globus and DataGrid projects. Deliverables from Work Packages WP1, WP2, and WP3 provide detailed descriptions of these aspects. The most important open question is which technology should be used to efficiently implement the required functionality and how to organize interfaces between components. To solve this problem we must obtain more information and working experience with the recently proposed Open Grid Services Architecture [OGSA].

## 13 REFERENCES

- [GridNCI] Foster, I., and Kesselman, C.: *The Grid: Blueprint for a New Computing Infrastructure*
- [OGSA] Foster, I.: *Physiology of the Grid* – draft:  
[www.globus.org/research/papers/ogsa.pdfv](http://www.globus.org/research/papers/ogsa.pdfv)
- [WP1.1] Work Package 1.1 Deliverable D1.1.1
- [WP1.2] Work Package 1.2 Deliverable D1.2.1
- [WP1.3] Work Package 1.3 Deliverable D1.3.1
- [WP1.4] Work Package 1.4 Deliverable D1.4.1
- [CrossGrid] The CrossGrid Project Homepage – [www.eu-crossgrid.org](http://www.eu-crossgrid.org)
- [Globus] The Globus Project Homepage – [www.globus.org](http://www.globus.org)
- [EDG] The European DataGrid Project Homepage - [www.eu-datagrid.org](http://www.eu-datagrid.org)