EnviroGRIDS Data Storage Guidelines

Title	EnviroGRIDS Data Storage Guidelines
Creator	Dorian Gorgan (UTCN)
Creation date	10.06.2009
Date of last revision	30.10.2009
Subject	EnviroGRIDS basic functionality, geospatial and Grid infrastructures interoperability, Grid oriented architecture, data repository and data models, Web portals, EnviroGRIDS Portal, SWAT execution over the Grid
Status	Validated
Type	Word document
Description	Guideline explaining how data repositories should organize, and distribute, and make information available.
Contributor(s)	Dorian Gorgan, Victor Bacu, Nicolas Ray, Andrew Maier
Rights	Public
Identifier	EnviroGRIDS_D2.2
Language	English
Relation	Interoperability guideline (D2.1) Sensors guideline (D2.3) Remote sensing guideline (D2.4) GRID infrastructure sustainability (D2.5) Technical report and software package of grid services supporting massive data management (D2.7)

Abstract:

This document describes the Service Oriented Architecture (SOA) of the enviroGRIDS infrastructure aiming at providing the user with access to SWAT model execution over the Grid, Geospatial functionality, and distributed Earth Science data. The document was prepared as a guideline for data storage over the Grid, highlighting the main issues emerging from conceptual and technological solutions of software components, data repositories, data management, Grid oriented processing, Grid portal, and interoperability between geospatial and Grid infrastructures.

Executive Summary

EnviroGRIDS (Building Capacity for a Black Sea Catchment Observation and Assessment System supporting Sustainable Development) aims to develop a Spatial Data Infrastructure (SDI) targeting this region and link it to the EGEE infrastructure. A large catalogue of environmental data sets (e.g. land use, hydrology, and climate) will be gathered and used to perform distributed spatially-explicit simulations to build scenarios of key environmental changes.

In this document we describe the architecture of the enviroGRIDS infrastructure as a guideline for data storage, and highlighting the main issues emerging from conceptual and technological solutions of software components, data repositories, data management, Grid oriented processing, Grid portal, and interoperability between geospatial and Grid infrastructures.

The enviroGRIDS project development is influenced by the projects requirements, partners' expertise, as well as the ongoing work in standardization and achievements in interoperability of geospatial and Grid infrastructures (SOA [13], OGSA [62], OGSA-DAI [61], OASIS [50], GEOSS [39], INSPIRE [46], OGC [56], gLite [33], G-OWS [42], etc).

We consider the Geospatial Web Services and Grid Platform as important technologies supporting the development of the enviroGRIDS infrastructure. Geospatial platforms provide spatial data oriented specialized services such as storing, management, processing, and visualization. Grid technology hides the complexity of underlying infrastructure, providing techniques for data management and security, as well as the abstraction mechanism needed to deal with heterogeneous resources. The Grid platform supports single sign-on to distributed resources, transfer of large datasets at high speed, setting up of virtual organizations, maintenance of information in central and distributed catalogues, and efficient resources management.

The enviroGRIDS system will comply with Service Oriented Architecture providing secure and persistent services over the Grid, and as stateless services over the Web. Services encapsulate the basic functionality provided to user applications by both Grid and Geospatial interoperable infrastructures (see section 2.2):

- *Data Management*: provides the basic operation on data repositories (e.g. data access, transfer, replication, storage metadata). Catalogues oriented operations are supported by AMGA Metadata Catalogue [25], which is the metadata service used in the gLite middleware.
- *Security and User Management*: provides the user the functionality needed to work with VOMS database, in order to support user authentication, authorization, and credential management. VOMS support the implementation of particular policies for data access and use.

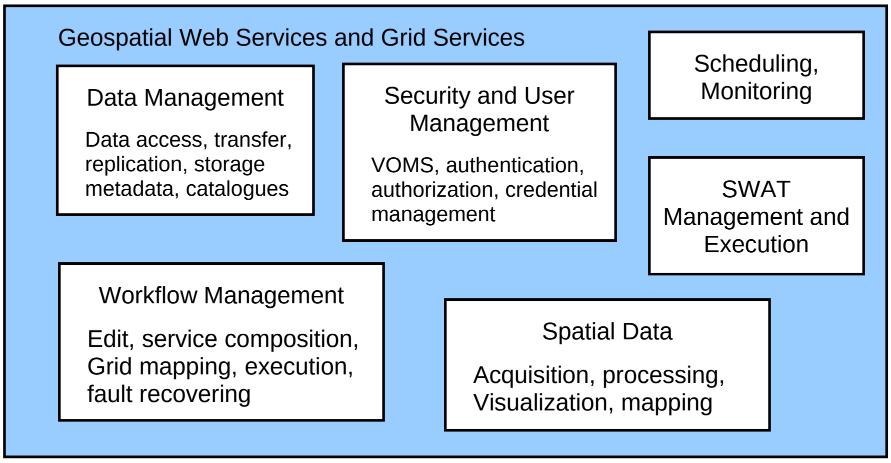

Geospatial Web Services and Grid Services
Data Management
Data access, transfer, replication, storage metadata, catalogues
Security and User Management
VOMS, authentication, authorization, credential management
Scheduling, Monitoring
SWAT Management and Execution
Workflow Management
Edit, service composition, Grid mapping, execution, fault recovering
Spatial Data
Acquisition, processing, Visualization, mapping

Figure 1 EnviroGRIDS Services provided through the interoperability of the Geospatial and Grid infrastructures

- *Scheduling*: provide optimal resource allocation and sharing. Static or dynamic load balancing provide the best efficiency, costs, and use of available resources.
- *Monitoring*: support evaluation of the execution performance, and statistical analysis.
- *SWAT Management and Execution*: provide the functionality to control the execution over the Grid of the SWAT modules and related data.
- *Workflow Management*: supports the graph description of the processing, service composition, Grid mapping, workflow interpretation and execution, and fault recovering.
- *Spatial Data Acquisition*: support the working with sensors. They supervise the sensor status, data acquisition and transformation, store, and processing.
- *Visualization and GIS Mapping*: support data visualization in graphical user interfaces of the interactive applications, and maps generation and visualization.

The present document is divided into nine chapters. Chapter 1 presents briefly the enviroGRIDS project, and the purpose of this document. Chapter 2 is an introduction to the enviroGRIDS project describing the objectives, and the layered functional architecture of the enviroGRIDS system, with a short description of data repository, main services, and applications. Chapter 3 deals with interoperability of geospatial and grid infrastructures. Chapter 4 describes the Grid oriented enviroGRIDS architecture and the related architectural standard patterns, the impact on data management, and geospatial and Grid interoperability. Chapter 5 is a description of the concepts that will be used for data repositories, data models, data replication and storage, metadata, data movement, data discovery services, and security. Chapter 6 presents the Web and Grid portals architectural and functional solutions. Chapter 7 presents the SWAT model. Chapter 8 is a presentation of the main technologies involved in the enviroGRIDS implementation. Finally, the chapter 9 presents some conclusions and the main recommendations that can be derived at this stage.

Contents

List of figures

1 Introduction

1.1 Purpose and scope

EnviroGRIDS (Building Capacity for a Black Sea Catchment Observation and Assessment System supporting Sustainable Development) is a 4-years project funded under the EC Seventh Framework Programme, aiming to address the subjects of ecologically unsustainable development and inadequate resource management in the Black Sea Catchment area. The project will develop a Spatial Data Infrastructure (SDI) targeting this region and linked to the Enabling Grid for E-sciencE infrastructure (EGEE). A large catalogue of environmental data sets (e.g. land use, hydrology, and climate) will be gathered and used to perform distributed spatially-explicit simulations to build scenarios of key environmental changes.

The purpose of this document is to provide the architectural design (combining SOA, OGC, OASIS, and Grid approaches) of the enviroGRIDS infrastructure software, focusing on data storage issues. It provides guidance for the design, implementation and integration activities, carried out in particular by WP2 - Spatial Data Infrastructure, and Task 2.2 - Interoperability and data storage.

To reach its objectives the enviroGRIDS projects keeps focused on continuous cycle of permanent enhancement by which enviroGRIDS (quality of services, usability, performance, etc) is evaluated and validated according with user needs on modeling, studying and visualizing of environmental models based on Soil Water Assessment Tool (SWAT). The improvements are based on results of research activities, taking into account developments and constraints in the areas of standardization, technologies and policies.

This document has to be considered as a reference document as we expect continuous modifications and extensions of the proposed system while we gain experience with practical implementations, feedback from our users and partners, and evolution of the requirements.

1.2 Document structure

EnviroGRIDS Storage Guidelines is divided into nine chapters. Chapter 1 (this chapter) is a short presentation of the enviroGRIDS project, and the purpose of this document

Chapter 2 is an introduction to the enviroGRIDS project describing the objectives, and the layered functional architecture of the enviroGRIDS system. It concerns with the short description of data repository, main services, and applications.

Chapter 3 deals with interoperability of geospatial and Grid infrastructures highlighting the INSPIRE Directive, the OGC Interoperability Program, Web services, G-OWS achievements, and mainly the interoperability between Earth Science and Grid infrastructures.

Chapter 4 describes the Grid-enabled enviroGRIDS architecture and the related architectural patterns such as SOA, OGSA, OGSA-DAI, and the impact on data management and geospatial and Grid interoperability.

Chapter 5 emphasize the main concepts concerning data repositories, data models, data replication and storage, metadata, data movement, data discovery services, and security.

Chapter 6 presents the Web and Grid portals architectural and functional solutions by three generations of the Grid portals.

Chapter 7 presents the SWAT model, its components, data input and output related requirements, and model execution.

Chapter 8 is a presentation of the main technologies involved in the enviroGRIDS implementation. It presents the gLite technology available in EGEE, the GENESI-DR project, Uniform Resource Management URM GeoPortal, and BASHYT - Basin Scale Hydrological Tool Decision Support System.

Finally chapter 9 sketches the conclusions and the main recommendations of this document.

2 EnviroGRIDS System

The Black Sea Catchment is recognized for its ecologically unsustainable development and inadequate resource management leading to severe environmental, social and economical problems. The FP7-funded enviroGRIDS project (April 2009 – March 2013) will address these issues by developing a Spatial Data Infrastructure (SDI) targeting this region and linked to the EGEE infrastructure. A large catalogue of environmental data sets (e.g. land use, hydrology, and climate) will be gathered and used to perform distributed spatially-explicit simulations and to build scenarios of key environmental changes.

2.1 Main Objectives

During his 4-year timeframe, the enviroGRIDS project aims at the following objectives for which the EGEE infrastructure will be instrumental:

- A high resolution (sub-catchment spatial and daily temporal resolution) water balance model will be applied to the entire Black Sea catchment using the Soil Water Assessment Tool (SWAT). This tool will first be gridified to be used on the EGEE infrastructure.
- Access to real time data from sensors and satellites will provide early warning and decision support tools to policy-makers and citizens. These data need to be streamlined into the grid-enabled enviroGRIDS SDI to ensure fast computation and dissemination of results.
- Because spatial data is very heterogeneous in format and quality across the European community, urgent efforts are needed to organize and standardize spatial data to improve its interoperability. The enviroGRIDS SDI will rely on the development of policies, technologies, data, common standards, standard practices, protocols and specifications such as those of the Open GIS Consortium (OGC). Through cataloguing, the grid infrastructure will help implementing and sharing standardized data sets.

By taking a watershed approach involving 15 Eastern-European countries, enviroGRIDS is the first truly trans-national effort to address many societal issues in the region. This system will incorporate a shared information system that operates on the boundary of scientific/technical partners, stakeholders and the public. It will contain an early warning system that will inform in advance decision makers and the public about risks to human health, biodiversity and ecosystems integrity, agriculture production or energy supply provoked by climatic, demographic and land cover changes on a 50 year time horizon. It will significantly build local, national and regional capacity on Earth Observation Systems through active contribution to the GEOSS and INSPIRE initiatives.

The strong Grid component of the project will foster data interoperability, and will most certainly trigger new directions of research or alternative ways of analyzing high resolution data sets. The large user community of SWAT may greatly benefit from a gridified version of the software, and a sustainable Virtual Organization will be created in this regard.

2.2 EnviroGRIDS Layered Functionality

EnviroGRIDS is a distributed system built on a Service Oriented Architecture (SOA) that allows a flexible use of services over heterogeneous architectural components and technologies. The functionality provided by services could be used anywhere over the computing infrastructure by open standards and communication protocols.

The functionality and the layered architecture of the enviroGRIDS system are presented in Figure 2. The lower level is the data level. The following levels are Grid Infrastructure and Middleware. The Grid infrastructure is provided by the EGEE network, on which the gLite middleware is running. The upper levels consist of a Web portal hosting various services.

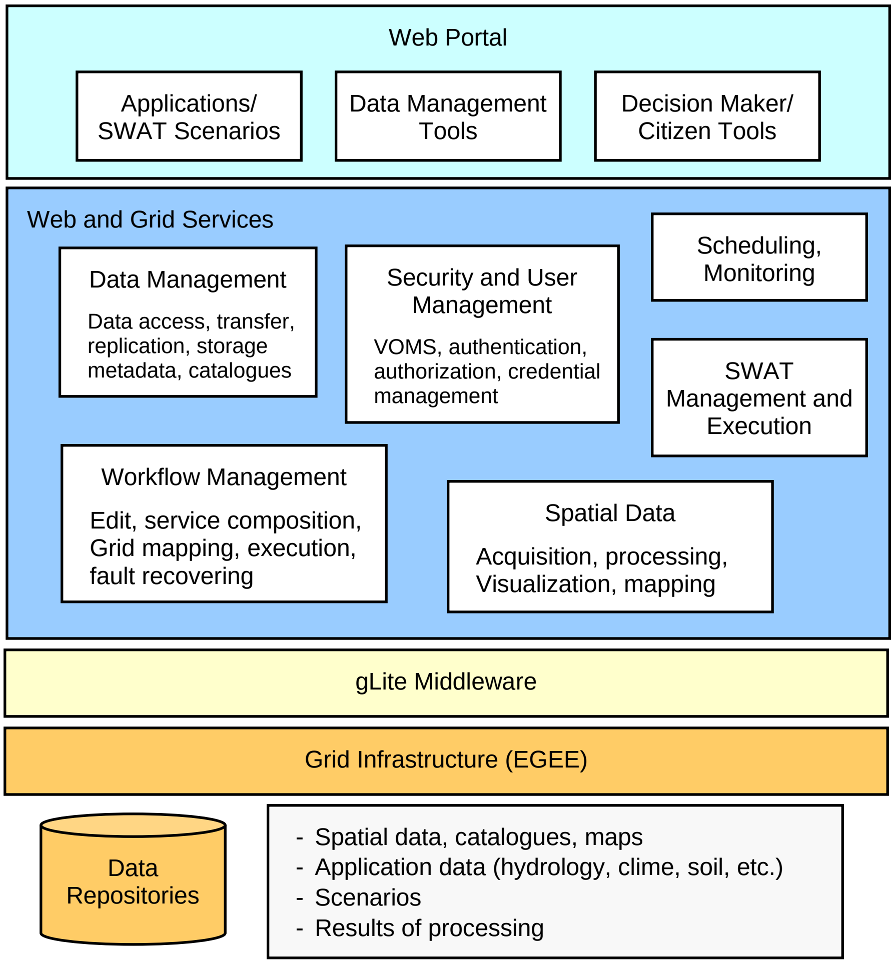

Web Portal
Applications/ SWAT Scenarios
Data Management Tools
Decision Maker/ Citizen Tools
Web and Grid Services
Data Management
Data access, transfer, replication, storage metadata, catalogues
Security and User Management
VOMS, authentication, authorization, credential management
Scheduling, Monitoring
SWAT Management and Execution
Workflow Management
Edit, service composition, Grid mapping, execution, fault recovering
Spatial Data
Acquisition, processing, Visualization, mapping
gLite Middleware
Grid Infrastructure (EGEE)
Data Repositories
- Spatial data, catalogues, maps
- Application data (hydrology, clime, soil, etc.)
- Scenarios
- Results of processing

Figure 2 EnviroGRIDS functional Layers

2.2.1 Data Repositories

Data Repositories consist of stored data and associated functionalities required to manage the repository. The repository stores raw data (e.g. spatial data) as well as processed data (e.g. maps, tables). They store application data, which are specific for each application type and instance (e.g. hydrology, climate, soil, etc). The register stores metadata catalogues that support the search, discovery and use of distributed data by the user applications, and processing services.

2.2.2 Resource Management and Data Processing Services

This level provides the basic services available over the Grid as secure and persistent services, and over the Web as stateless services. The services encapsulate the basic functionality provided to user applications. Such service categories are:

- *Data Management*: provides the basic operation on data repositories (e.g. data access, transfer, replication, storage metadata). The catalogues oriented operations are supported by AMGA Metadata Catalogue [25], which is the metadata service used in the gLite middleware.

- *Security and User Management*: provides the user the functionality needed to work with VOMS database, in order to support user authentication, authorization, and credential management. VOMS support the implementation of particular policies for data access and use.
- *Scheduling*: provide optimal resource allocation and sharing. Static or dynamic load balancing provide the best efficiency, costs, and use of available resources.
- *Monitoring*: support evaluation of the execution performance, and statistical analysis.
- *SWAT Management and Execution*: provide the functionality to control the execution over the Grid of the SWAT modules and related data.
- *Workflow Management*: supports the graph description of the processing, service composition, Grid mapping, workflow interpretation and execution, and fault recovering.
- *Spatial Data Acquisition*: support the working with sensors. They supervise the sensor status, data acquisition and transformation, store, and processing.
- *Visualization and GIS Mapping*: support data visualization in graphical user interfaces of the interactive applications, and maps generation and visualization.

2.2.3 EnviroGRIDS Applications

The enviroGRIDS Portal exposes to the user a set of tools and applications of which functionality is composed of the services provided by the underlying level. There are four types of interactive applications and tools available through the enviroGRIDS Portal:

1. *Applications/ SWAT Scenarios Development Tools*: the user may develop various scenarios for natural phenomena (e.g. through a graph based language) and use cases, and perform their execution over the Grid. The user may visualize the results and analyze statistical data.
2. *Data Management Tools*: data administrators and data provider may access, upload, update, and organize spatial data.
3. *Decision Maker Tools*: provide the user possibility to develop and execute various scenarios by different data series, and to analyze and make predictions on the phenomenon evolution. Graphical data visualization and mapping are available as well.
4. *Citizen Tools*: provide the citizen, as an Internet visitor, the execution of a given set of scenarios by limited set of data, and graphical visualization of the results.

3 Interoperability of Geospatial and Grid Infrastructures

3.1 Overview

Earth Science Infrastructures is becoming very large and involve a lot of distributed resources. Geospatial interoperability technologies are able to use in an easier manner important volumes of distributed heterogeneous geospatial data and services. Open Geospatial Consortium (OGC) has developed interoperable Web service specifications, such as Web Coverage Service (WCS), Web Map Service (WMS), Web Feature Service (WFS) and Catalog Service for Web (CSW), for promoting geospatial interoperability in a distributed environment [55]. These specifications are widely used by the geospatial community for sharing data and services [5], [21]. Throughout Earth Science, geoprocessing tasks normally involve a sequence of inter-related steps of computations within the Web service environment. There is a need for cooperation and security mechanisms between any two geospatial Web services. Grid infrastructure should provide secure interoperation and collaboration of distributed heterogeneous resources, including as well the compatibility with geospatial technologies.

Interoperability and metadata sharing are two key technologies in discovering and enabling access to usable resources (i.e. information, data, services, models, etc.). Interoperability and metadata sharing are achieved by adopting a Service-Oriented Architecture (SOA) approach applying International Standards and best practices, e.g. INSPIRE IR, GEOSS IOC architecture rules, ISO 19100, and OGC, OGF, W3C and OASIS specifications.

3.2 INSPIRE − Infrastructure for Spatial Information in the European Community

INSPIRE (INfrastructure for SPatial InfoRmation in Europe) [46] is a European Directive (entered into force in May 2007 and fully operational by 2019) that aims at creating an European Union Spatial Data Infrastructure. This will enable the sharing of environmental spatial information among public sector organizations and better facilitate public access to spatial information across Europe [12].

The main purpose of INSPIRE is to support the formulation, implementation, monitoring, and evaluation of Community environmental policies. To ensure that the spatial data infrastructures of the Member States are compatible and usable in a Community and transboundary context, the Directive requires that common Implementing Rules are adopted in a number of specific areas:

- Metadata
- Data Specifications
- Network Services
- Data and Service Sharing
- Monitoring and Reporting

3.3 OGC Interoperability Program

OGC's Interoperability Program [55] is a global, hands-on and collaborative prototyping program designed to rapidly develop, test and deliver proven candidate specifications into OGC's Specification Program, where they are formalized for public release. In OGC's Interoperability Initiatives, an international team of technology providers works to solve specific geoprocessing interoperability problems. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments, and interoperability support services – all designed to encourage rapid development, testing, validation and adoption of open, consensus based standards specifications.

OGC has expressed interest for OGC Web Services, Phase 6 (OWS-6) Testbed activity to advance OGC's open framework of standards based on priority geospatial and location based interoperability requirements [55]. The OGC Interoperability Team recommended the following areas for the OWS-6 initiative:

- Web Enablement Sensor (SWE)

- Geo Processing Workflow (GPW)
- Aeronautical Information Management (AIM)
- Decision Support Services (DSS)
- Compliance and Interoperability Test and Evaluation (CITE)

The *Geo Processing Workflow* (GPW) thread aims at building on the progress of previous testbeds with a focus on maturing the interoperability and capabilities of OGC web service in a service-oriented architecture with particular emphasis to address OGC web service security issues. To satisfy mission-critical goals, the architecture must not only provide for integration of a wide variety of service capabilities and resources, it must do so and ensure authenticity, integrity, quality and confidentiality of services and information. To meet these goals, the following task areas have been identified:

- Asynchronous Workflow and Web Services Security
- Data Security for OGC web services
- Data Accessibility
- Web Processing Service (WPS) Profiles – Conflation, and Grid processing
- Geography Markup Language (GML) Application Schema Development and Shape Change Enhancements

3.4 OGC Web Service standards

The following OGC Web Service standards have been defined for geospatial data:

- The OpenGIS® *Web Map Service Interface Standard* (WMS) [59] provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layers and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined or not.

- OpenGIS *Web Feature Service* (WFS) [58] allows a client to retrieve and update geospatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services. The specification defines interfaces for data access and manipulation operations on geographic features. Via these interfaces, a Web user or service can combine, use and manage geodata from different sources. A Transactional WFS includes the optional Transaction operation to insert, update, or delete a feature.

- OpenGIS *Web Coverage Service* (WCS) [57][53] supports the ability to publish and search collections of descriptive information (metadata) about geospatial data, services and related resources. Providers of resources use catalogues to register metadata that conform to the provider's choice of an information model; such models include descriptions of spatial references and thematic information. Client applications can then search for geospatial data and services in very efficient ways.

- The *Catalogue Service for the Web* (CSW) [52] is one binding defined in the OpenGIS Catalogue Services Specification . The Catalog standard defines common interfaces to discover, browse, and query metadata about data, services, and other potential resources.

- OpenGIS *Web Processing Service* (WPS) [60]. This is designed to standardize the way that GIS calculations are made available to the Internet. WPS can describe any calculation (i.e. process) including all of its inputs and outputs, and trigger its execution as a Web Service. WPS supports simultaneous exposure of processes via GET, POST, and SOAP, thus allowing the client to choose the most appropriate interface mechanism. The specific processes served up by a WPS implementation are defined by the owner of that implementation. Although WPS was designed to work with spatially referenced data, it can be used with any kind of data. WPS makes it possible to publish, find, and bind to processes in a standardized and thus interoperable fashion. Theoretically it is transport/platform neutral (like SOAP), but in practice it has only been specified for HTTP. It is best described as a non-REST-ful RPC type service although it does comply with most of the REST principles.

- OpenGIS *Sensor Web Enablement* (SWE) enables developers to make all types of sensors, transducers and sensor data repositories discoverable, accessible and useable via the Web. The main adopted or pending OpenGIS Standards in the SWE framework include:
    - *Observations and Measurements* (O&M) - The general models and XML encodings for observations and measurements.
    - *Sensor Model Language* (SensorML) - standard models and XML Schema for describing the processes within sensor and observation processing systems.
    - *Transducer Markup Language* (TML) - Conceptual model and XML encoding for supporting real-time streaming observations and tasking commands from and to sensor systems.
    - *Sensor Observation Service* (SOS) - Open interface for a web service to obtain observations and sensor and platform descriptions from one or more sensors.
    - *Sensor Planning Service* (SPS) - An open interface for a web service by which a client can 1) determine the feasibility of collecting data from one or more sensors or models and 2) submit collection requests.
    - *Sensor Alert Service* (SAS) - Standard web service interface for publishing and subscribing to alerts from sensors.
    - *Web Notification Services* (WNS) - Standard web service interface for asynchronous delivery of messages or alerts from SAS and SPS web services and other elements of service workflows.

3.5 Interoperability between Earth Science and gLite Infrastructure

G-OWS Working Group [43] is concerned with interoperability between Earth Science and gLite infrastructures. The group concerns with implementation specifications for OWS (OGC Web Services) on gLite platform through the following activities [16], [18]:

- Integrate OWS with gLite as "back-end" processing environment
- Integrate OWS with gLite data management services and tools
- Empower OGC Web Services with the gLite security and Virtual Organization management services
- Design and implement gLite reference implementations of OWS for SDI community

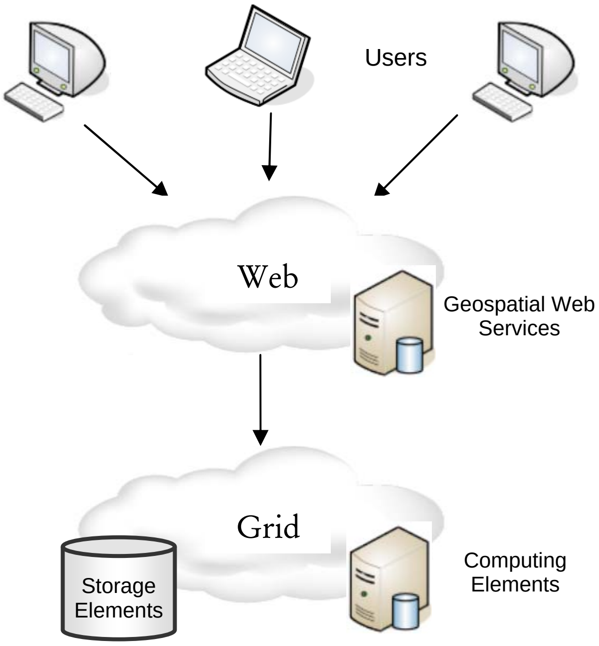


Figure 3 User, Web and Grid interaction in Geospatial and Grid middleware interoperability

Mazzetti and Nativi have highlighted in [16][12] and [18] the main issues and challenges facing the OGC WS and gLite interoperability:

- Semantics mismatch
  - Grids handle files, data, instruments, jobs, etc., at resource provision level
  - Geo-information services handle coverage, observations, measurements, models, etc., at resource content/application level
- Architectural mismatch:
  - OWS implements a Web Service-Oriented-Architecture
    - Stateless services
    - Synchronous communication between components
    - No embedded security. It has to be provided by other components and levels
  - gLite implements the Grid paradigm
    - Stateful services
    - Asynchronous and not fully event based communication
    - Strong embedded security based on the VO paradigm (by authentication, authorization, certificates)

In CYCLOPS Project [28] the G-OWS group experiments the integration of geospatial services and grid technologies by using WPS for disaster management. The project proves the approach of porting the applications by two use cases: Wildfires risk assessment – RISICO, Flash flood risk management – ALHTAIR. In both applications the OGC-WCS service achieves the input and output data related operations, and the OGC-WPS service supports the processing management.

Mazzetti and Nativi have presented in [16] the main benefits of porting the application to Grid technology consists of:

- Scalability. The grid infrastructure provides high processing and storage capabilities allowing to:
  - Improve output data resolution (e.g. in the RISICO project the processing capacity grows up from the 1km to 100m resolution)
  - Improve models complexity
  - Widen the covered area (e.g. RISICO: from local to national)
  - Improve the time-of-response(e.g. RISICO: from hours to minutes)
  - Time-of-response is independent of resolution/coverage

- Flexibility and Interoperability. The geospatial services layer allow to:
  - Integrate/assimilate new and heterogeneous input data (e.g. running RISICO in France/Italy border)
  - Integrate outputs in a higher level application chain
  - Facilitate models interoperability and composition
  - Be interoperable with other standard based infrastructures (e.g. INSPIRE, GEOSS)

Finally, the developers of CYCLOPS have provided mechanisms for the identification of spatially referenced data required for processing, the initialization of the processing, and the management of the outputs access by the client. The WPS service publishes a generic interface, and it can be used to wrap other existing and planned OGC services that focus on providing geospatial processing services. Therefore, WPS service provides and hides the grid processing capabilities to the geospatial services infrastructure.

3.6 Geospatial Interoperability in Globus based Grid Environment

In their paper [5], the authors analysis the technology for enabling the OGC-based geospatial interoperability in a Globus based Grid environment. The paper presents a new Grid-enabled catalogue services model for secure registry, discovery and access of geospatial data and services. The model combines the information schemas of Grid Metadata Catalog Service (MCS)/Replica Location Service (RSL) and the OGC Catalog Service for Web (CSW). The model exploits as well the geospatial metadata standards including ISO 19115 [47], ISO 19115-2, FGDC Content Standard for Geospatial metadata [30], and NASA ECS Metadata [49].

The paper presents the development of the Grid enabled CSW (GCSW) service. The service preserves the OGC CSW interface while providing the naming and location transparency by mapping Grid MCS/RLS information model to OGC CSW information model. The paper proposes the metadata model for data types and service types and integrates metadata with CSW model to facilitate the service interoperability and chaining. The authors have developed an intelligent Grid Service Mediator (iGSM) to extend the other geospatial Web services (e.g. WCS, WMS) to the Grid environment for geospatial service interoperability.

The user interfaces just with geospatial services. A request for geospatial services for users is completed through the request and response operations of geospatial services. For this reason, Grid Security Infrastructure (GSI) is combined with geospatial services to authenticate the requester. When the requester is valid, the service request is sent to iGSM to accomplish the request. iGSM then securely queries the Grid enabled CSW service to determine the service and data locations, to analyze if it is necessary for the service to interact with other services, and finally invokes the service execution. If required, data and service are at remote location, it is the duty of iGSM to manage the data and result traffics between working stations.

By the implementation of GCSW, iGSM, and a number of Grid-enabled OGC geospatial services, the authors have successfully extended the application of Grid technology to the Earth Science community for geospatial interoperability. The Grid technology has great potential to support applications in the geospatial discipline. By using Grid technology as the foundation for geospatial data infrastructure, geospatial data and services are securely shared while providing known and standard OGC interfaces to users.

4 EnviroGRIDS Architecture

4.1 Architectural Components

The proposed enviroGRIDS system architecture will encompass the following modules (Figure 4):

- *EnviroGRIDS Portal* includes a set of applications accessed by four types of users: *administrator, data manager, data provider, Earth Science specialist, decision maker*, and *citizen*. Scenarios describe the logics of applications and use cases of specific domains of interest such as hydrology, SWAT based computation, etc.

  A Web portal allows application users to access resources specific to a domain of interest through a Web

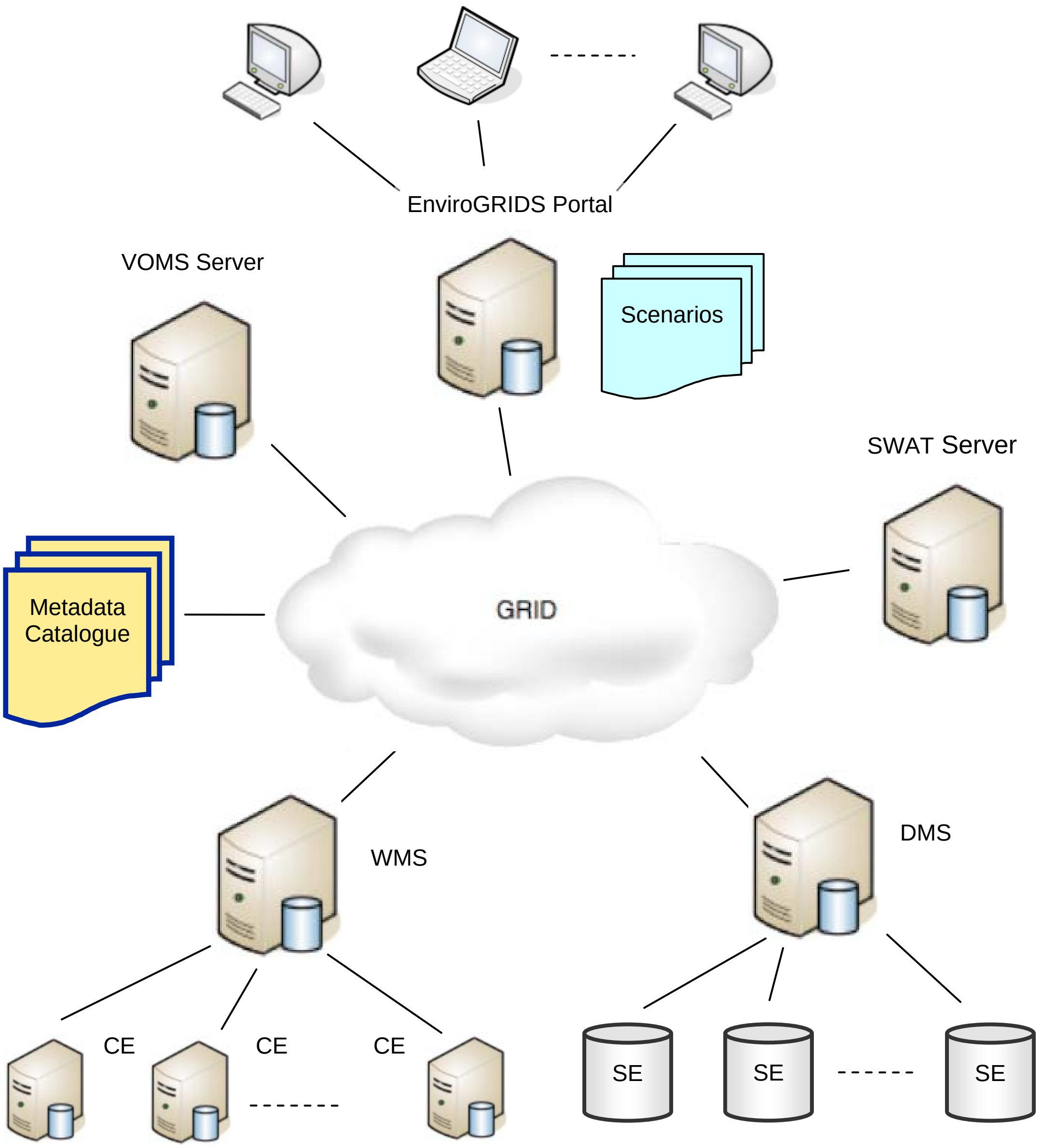


Figure 4 EnviroGRIDS System Architecture

interface. Unlike Web portal, the Grid portal could also provide access to Grid resources. The Grid portal may authenticate users and allow them to access remote resources, support making decisions on the job scheduling, and allow users to access and manipulate resource information stored on a remote database. The Grid portal provides data management tools for operations on data such as enter, update, modify, copy, delete, and replication.

- *VOMS Server* provides a database of user roles and capabilities and a set of tools for accessing and manipulating the database and using the database contents to generate Grid credentials for users when needed. VOMS-aware applications can use the VOMS data to make authentication decisions regarding user requests. To get authentication the user must use his particular authentication data (i.e. certificates).
- *Data Management System repository* consists of gLite nodes of Storage Element (SE) type. Initially, in the development phase, there is a central repository at UNIGE and other two complementary repositories in different locations of the partners. The central repository stores primary data and the secondary repositories store data replicas. Finally both types of data, primary and replicas, will be uniformly resident on both types of SE, primary and secondary. The secondary SE will be actually the partners who provide data or store data resulted from the SWAT related processing.
- *Metadata Catalogue* describe data stored in distributed data repositories, in terms of identifier, type, location, etc. In gLite the user identifies a file by its logical name LFN (Logical File Name). Other identifiers such as GUID, SURL, and TURL are transparent for the user. They are generated and used by the data management services. The LFC File Catalogue links the LFN identifier to other identifiers of the same data object.
- *Workload Management System* (WMS) broker is the interface between the Portal services and gLite services (e.g. data access, execution, monitoring). The WMS system is the gLite service supporting the distribution and management of the task over the available Grid resources. The user jobs are assigned to the CEs that provide the resources required by the users (e.g. minimum memory capacity, processor capacity, etc). The load balancing mechanism achieves an efficient distribution of the jobs over the available CEs in the Grid infrastructure.
- *Computing elements* (CE) perform the Grid based processing (e.g. SWAT related processing). Within the EGEE infrastructure they are gLite computing elements.
- *SWAT Server* executes the SWAT related processing in a distributed and parallel manner over the Grid. The description of input data and processing logics is encapsulated in the specific scenarios.

4.2 Grid Oriented Architecture

The Grid based solution of the enviroGRIDS architecture offers a series of advantages by which resource management, resource sharing, data and processing distribution, security, are just the most important. The Grid is based on the concept of service. Grid architecture conforms generally with SOA architecture and particularly with the OGSA framework.

4.2.1 Service Oriented Architecture

Service Oriented Architecture (SOA) [13][12] provides a set of principles of governing concepts used during phases of systems development and integration. Such architecture gathers functionality as services provided by different software modules. Rather than defining an API, SOA defines the interface in terms of protocols and functionality. Service orientation requires loose coupling of services with operating systems, and other technologies that support the applications. SOA separates functionality into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them within applications. The services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services.

There are two categories of service concept used in distributed architectures: Web and Grid services. Web services have emerged as popular standards based framework for accessing network applications [8]. Grid service concept adds new capabilities of stateful and dynamic services.

The notion of Web service describes an important computing paradigm supporting the addressing of heterogeneous resources across distributed environments. Web services define a technique for describing accessing software components, methods for accessing such components, and methods for identifying them. Web services are independent on programming language, programming model, and software system. The Web services related standards are mainly defined through W3C (World Wide Web Consortium) organism. Three standards are of particular interest: SOAP, WSDL and UDDI [8].

- *Simple Object Access Protocol* (SOAP) [67]: provides a means of messaging between a service provider and a service requestor. SOAP is a simple XML based formatting mechanism that describes an invocation by Remote Procedure Call (RPC). SOAP is independent on the transport protocol (i.e. HTTP, FTP, JMS, etc).
- *Web Service Description Language* (WSDL [72]): describes a Web service in an XML document. Service interfaces are defined abstractly in terms of messages structures and sequences of simple message exchanges (i.e. operations).
- *Universal Description, Discovery and Integration* (UDDI) [69]: specifies distributed registries of Web services. UDDI supports the publish-find and bind mechanism, through which a service provider describes and publishes the service details, the service requestor interrogates the registry to find the providers of the service, the service binds to the requestor's application using the technical details provided by UDDI.

4.2.2 Open Grid Services Architecture

The Open Grid Services Architecture (OGSA) Framework is developed within the Global Grid Forum (GGF) and expresses the convergence of Web services and Grid computing. It was presented at the Global Grid Forum (GGF) meeting, in 2002, and described initially in the paper [62]. OGSA supports the creation, maintenance, and application of ensembles of services maintained by Virtual Organizations (VOs). A service is defined as an entity providing capabilities such as computational resources, storage resources, networks, programs, and databases.

OGSA is based on several other Web service technologies such as WSDL and SOAP, but it aims to be largely compatible in relation to the transport-level handling of data [1], [9]. OGSA is a distributed interaction and computing architecture based around services, assuring interoperability on heterogeneous systems so that different types of resources can communicate and share information. OGSA has been adopted as grid architecture by a number of grid projects including the Globus Alliance.

A basic concept of OGSA is the service [9] as a network enabled entity that provides some capability through the exchange of messages. Computational resources, storage resources, networks, programs, databases, and so forth are all services. This concept allows considering all components of the environment are virtual. Moreover, OGSA represents everything as a Grid service, which is a Web service conforming a set of conventions and supporting standard interfaces. The interfaces address discovery, dynamic service creation, lifetime management, notification, and manageability (e.g. authorization, policy management, concurrency control, and monitoring and management of large sets of Grid service instances).

Starting from this basic set of interfaces, by which all Grid services are implemented, support the composition of higher order services that can be treated in a uniform way across layers of abstraction.

Grid services offer a set of capabilities. Each service implements one or more interfaces, and each interface defines a set of operations that may be invoked through a sequence of messages. The service interfaces are defined in WSDL language.

The Grid service maintains its internal state during its lifetime. Grid service instance is a particular state of a service. One main difference between Grid and Web services is that the Grid service is stateful, unlike the Web service that is stateless. Moreover the Grid service is dynamic – it could be created and destroyed dynamically.

4.2.3 Open Grid Services Architecture – Data Access and Integration

Open Grid Services Architecture – Data Access and Integration (OGSA-DAI) Project, started in 2002, develops a solution to the Internet-scale data integration. OGSA-DAI is a middleware bundle that allows data resources, such as file systems, relational or XML databases, to be accessed, federated and integrated across the network.

OGSA-DAI supports data resources sharing and collaboration. It concerns with:

- Data accessing structured data in distributed heterogeneous data resources
- Data transformation by exposing data in schema X to users as data in schema Y
- Data integration by exposing multiple databases to users as a single virtual database
- Data delivery to where it is needed by the most appropriate means: - web service, e-mail, HTTP, FTP, GridFTP

4.2.4 Grid Data Management Challenges

The Grid infrastructure adds some challenges to the data management compared with traditional data management systems [23].

- *Heterogeneity*: Data is stored on different systems by using different technologies. For this we need a common interface to storage resources. This interface offers a transparent access layer to the underlying complexity. The Storage Resource Manager (SRM) interface is the gLite solution for this requirement.

- *Distribution*: We need to store data at different locations. In most cases we are not using a shared file system or a common namespace. In addition we need to be able to move data between different locations and also to keep track of data storage location. The gLite middleware offers a File Transfer Service (FTS) to move files between different Grid sites and Catalog for keeping track where data are stored.

- *Data retrieving*: In most cases, the applications and data are not stored at the same place. We need a reliable scheduled file transfer service. For this purpose we can use the File Transfer Service that offers data scheduler, file placement service, transfer agent and file transfer library.

- *Security*: Data must be managed according to the VO membership access control policy. We need a centralized access control service like the gLite's File Authorization Service.

4.3 Geospatial and Grid interoperability in enviroGRIDS

Interoperability is a great challenge in the enviroGRIDS system development. Such a technology can significantly reduce problems associated with archiving, manipulating, analyzing, and utilizing large volumes of geospatial data at distributed locations. The OGC Web Services and the gLite middleware must be able to communicate and interact with each other in order to combine the complex specialized geospatial functionality with high power computation of the Grid.

There are many organizations and working groups concerned with research and experiments on solutions for interoperation of Geospatial and Grid infrastructures. For instance, we could mention G-OWS (gLite-OWS) Working Group [42][32], OGC Interoperability Institute [54], FGDC Geospatial Applications and Interoperability Working Group [31].

Figure 5 presents the enviroGRIDS functional architecture. For interoperability, the architecture includes the Geospatial Oriented Level and the Geospatial and Grid Services Level. The service level includes as well the functionality provided by the SWAT modules execution, Collaborative Working Environment (CWE) and Uniform Resource Management (URM).

According to the SOA concepts and requirements of interoperability between Geospatial and Grid infrastructures each of the main visible functionalities should publish at least service interface. Therefore CWE, URM, and SWAT related functionality should publish as well service interfaces. With these interfaces, the available functionality is visible from enviroGRIDS Portal and consequently, by the end user applications (e.g. Decision

Maker/Citizen oriented Applications). The enviroGRIDS portal should be the single way of the user to get into the system and the portal should face a unique style of the graphical user interface.

The biggest challenge is the interoperability with the Grid (i.e. gLite). Conceptually it is supported by the extension called Geospatial Oriented Level. It should support the mechanisms of integration of the Grid compatibility into OGC Web Services concerning mainly:

- Data management: metadata, catalogues, data access, etc.
- Resource management: storage elements, computing elements, services, etc.
- Job execution: process synchronization, fault recovering, workflow management, etc.
- Security: certificates, single sign-on mechanism, etc.
- Other issues arisen from conceptual, technical and technological particularities.

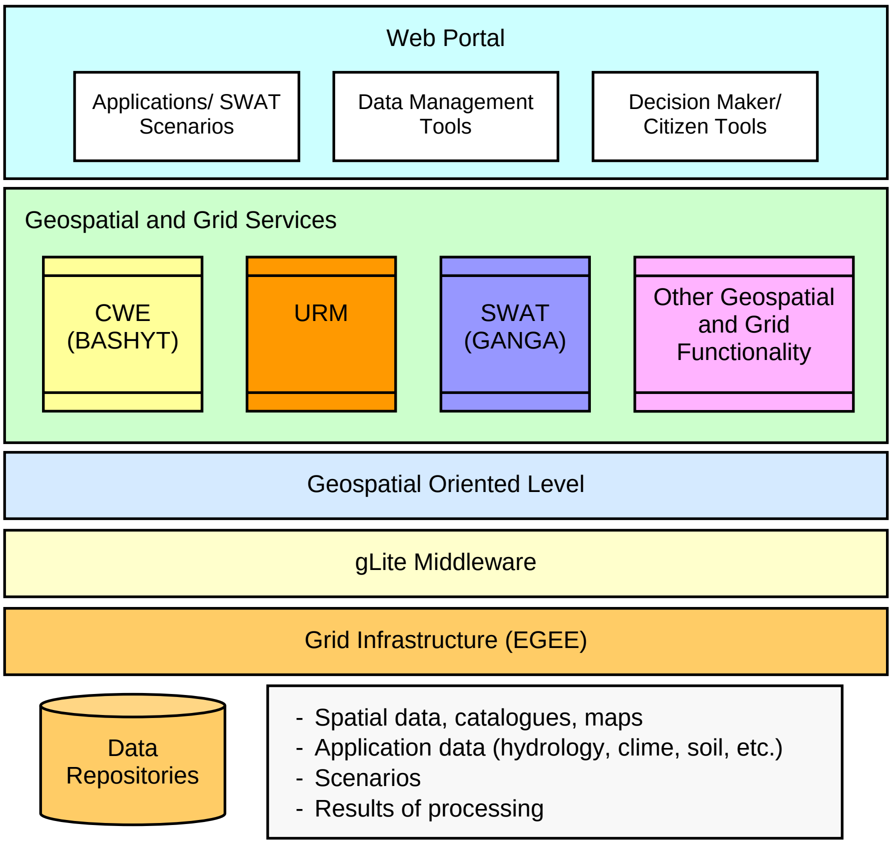

Web Portal
Applications/ SWAT Scenarios
Data Management Tools
Decision Maker/ Citizen Tools
Geospatial and Grid Services
CWE (BASHYT)
URM
SWAT (GANGA)
Other Geospatial and Grid Functionality
Geospatial Oriented Level
gLite Middleware
Grid Infrastructure (EGEE)
Data Repositories
- Spatial data, catalogues, maps
- Application data (hydrology, clime, soil, etc.)
- Scenarios
- Results of processing

Figure 5 Service oriented EnviroGRIDS Functional Architecture

5 Data Repository

The Digital Repository is a complex collection of data. The main functionalities concern with data access and related services provided to users, data organization, data store, administration, search and discover, and secure access and retrieve.

5.1 Data Model

The performances of the data system are influenced by the internal organization of data sources. Different models [22] can be adopted as a solution to integrate different data sources in the Data Grid. These models depend on the source of data, the size and the mode of sharing.

The model taxonomy:

1. Monadic;
2. Hierarchical;
3. Federation;
4. Hybrid.

5.1.1 Monadic Model

The monadic model represents the simplest solution (Figure 6). Here all the data is gathered at a central repository. All the queries are solved at this level, and the data provider is only one node. The data can be gathered from different sources but in this system there is only a single point of access, for example a Web Portal. This Web Portal is also responsible with authorization and authentication of users. Although this model consists of a central repository there can be created different replicas of this repository. In this case, the data can be access also from the replicated locations.

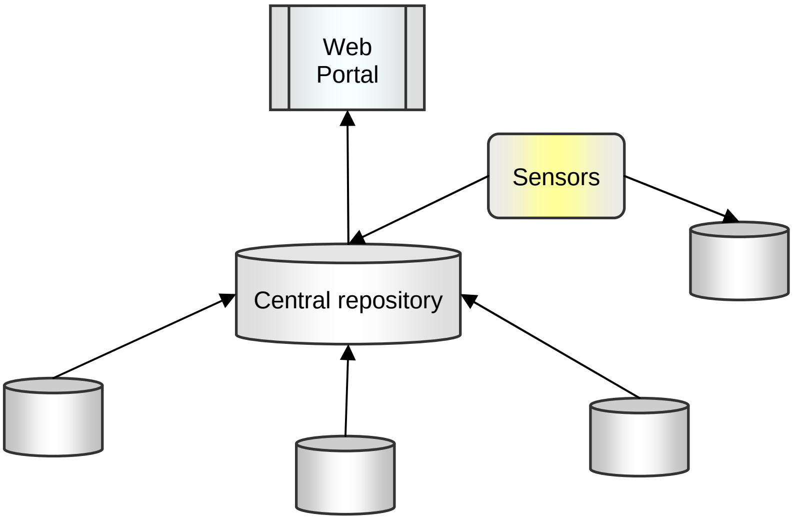


Figure 6 Monadic Data Model

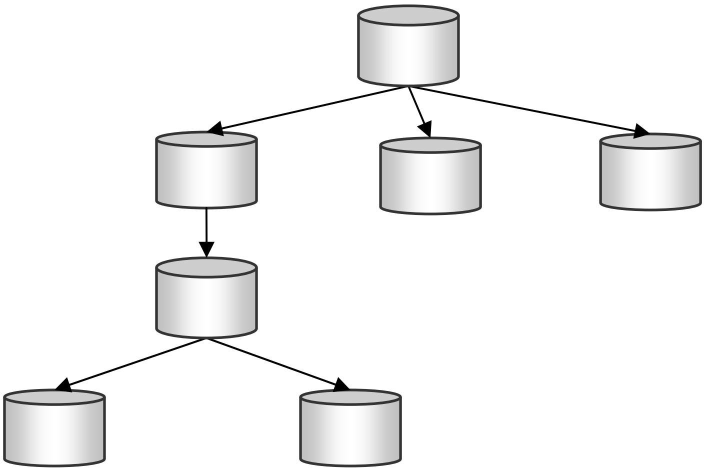


Figure 7 Hierarchical Data Model

5.1.2 Hierarchical Model

In some applications (Figure 7), there is a single source of data (for example the CERN experiment). A subset of data is distributed to the lower levels. The data consistency is much simple to accomplish because of the single data source. The difference from the monadic model is that in this case the data can be accessed from multiple access points, located at different levels in the hierarchy of Data Grid. In this way the users access the needed data from the level to which they have access.

5.1.3 Federation Model

In some cases (Figure 8), different institutions that already have some databases wishes to share their data. Researchers from one institution can have access to data provided by another institution based on authentication. That institution accomplishes the control of each local database. One example of a federation model is in the GENESI-DR, where the data is federated from different institutions. Based on different security policies and authentication methods different users can access that data.

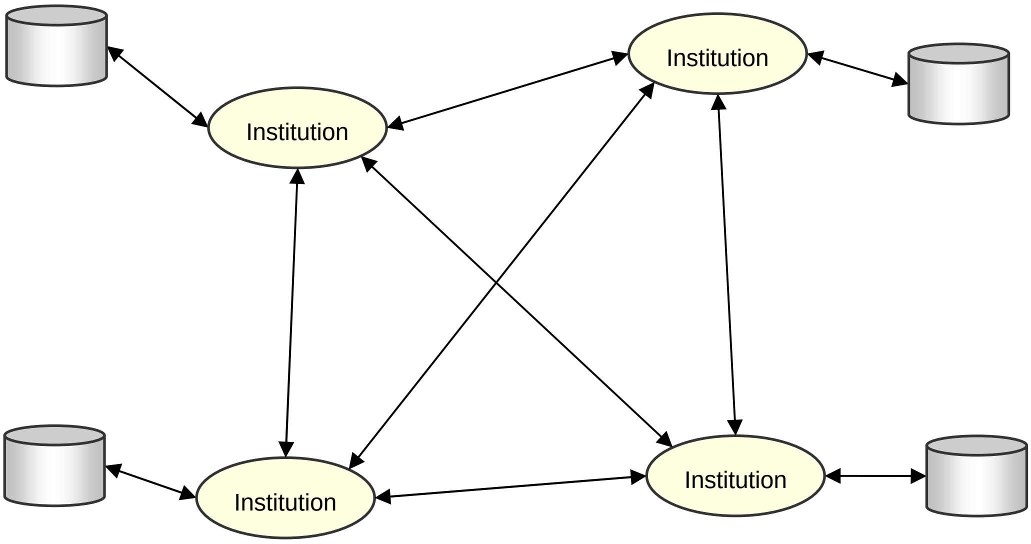


Figure 8 Federation Data Model

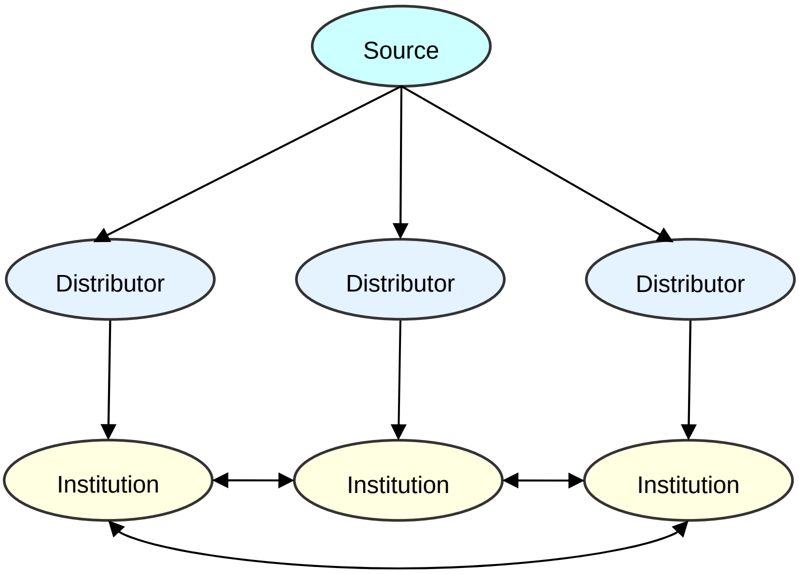


Figure 9 Hybrid Data Model

5.1.4 Hybrid Model

The hybrid model combines the previous models (Figure 9). This kind of model comes out of the need for researchers to collaborate and to share products of their analysis.

5.2 Data Replication and Storage

In a Data Grid the users belonging to a particular Virtual Organization are spread in different geographical areas. Therefore, the data replication is an important feature that ensures the scalability, reliability and also can preserve the bandwidth. The access to a desired data is accomplished through a replica management system that manages and redirects the user to the most appropriate location of the desired data.

A basic replica management system is composed of several storage nodes, a replica manager and a replica catalogue (Figure 10). The replica manager creates and manages the replicas according to the selected replication strategy. The location and the number of replicas for a particular data are discovered by using the replica catalogue. In some systems, the manager and the catalogue are merged into one entity.

The most important elements of a replication mechanism are the architecture of the system and the replication strategy.

A centralized system consists of one master replica. In case of update at this master level, the updates are propagated to the other nodes. The decentralized model is a much complex model, in which it is more difficult to maintain the data consistency. In this peer-to-peer model, the update can occur at any replica node and therefore the nodes need to synchronize with each other. The topology represents the way in which the replica nodes are organized. The mostly used topologies are the hierarchical, flat and hybrid. In the hierarchical topology, the nodes are grouped in a tree-like structure. The update is propagated on definite paths. In P2P systems is adopted the flat topology, and the updates are dependent of the arrangements between the peers. The hybrid topology is based on a hierarchical topology and at different levels are used a flat topology.

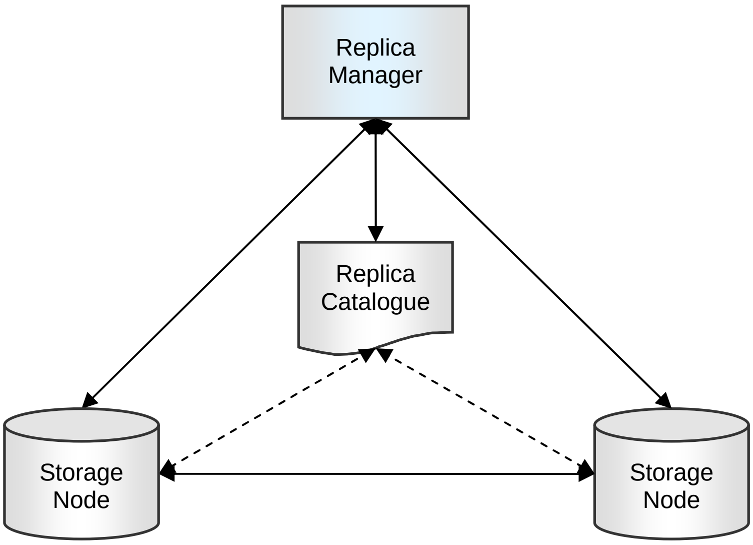


Figure 10 Data replication based architecture

In a complex Data Grid are stored a very large amount of data. Very often, it is difficult to identify a particular dataset from that amount of available datasets. For this reason, a metadata system is necessary. Depending on the datasets, the metadata information is created. General metadata information consists of different file attributes such as creation date, size on disk, physical location, and file checksum. The user-defined attributes consists of a set of properties that depend on the type of data. The metadata can be updated automatically when someone updates a replica, or the users can update them passively when they create new replicas.

In general, the data is updated in one site and the updates must be propagated to the rest of its replicas. This can be done in a synchronous or asynchronous mode. Because the datasets that are stored in a Data Grid are very large in terms of capacity the synchronous mode it is not used very often. In the asynchronous updating mode when the primary copy is changed the updates are propagated to all the other replicas. Another solution is based on notifications, the replica sites subscribe to update notifications, and when the primary replica is updated the notification is sent to all the other replicas.

The granularity is related to the level of subdivision of data. The dataset granularity deals with multiple files at the same time. The next level of granularity is individual files. Other strategies work with smaller subdivisions of files, such as objects or fragments.

The replication strategy objective function can be to maximize the locality or move data to the point of computation.

5.3 Metadata

The Grid applications often require and generate a huge amount of data. Therefore, an efficient mechanism for describing and localization of data is required. This problem can be solved by associating descriptive attributes to the files and answering to the user queries against the associated information.

Some of the basic metadata concepts are briefly described:

Entries – real world entities to which we are attaching metadata information for describing them

Attribute – key/value pair

Schema – set of attributes

Collection – set of entries associated with a schema

Metadata – list of attributes associated with entries

The schemas can be viewed as tables, attributes as columns and entries as rows.

A metadata catalogue is the repository for the metadata associated with files. Metadata can describe any grid entity/object (Figure 11).

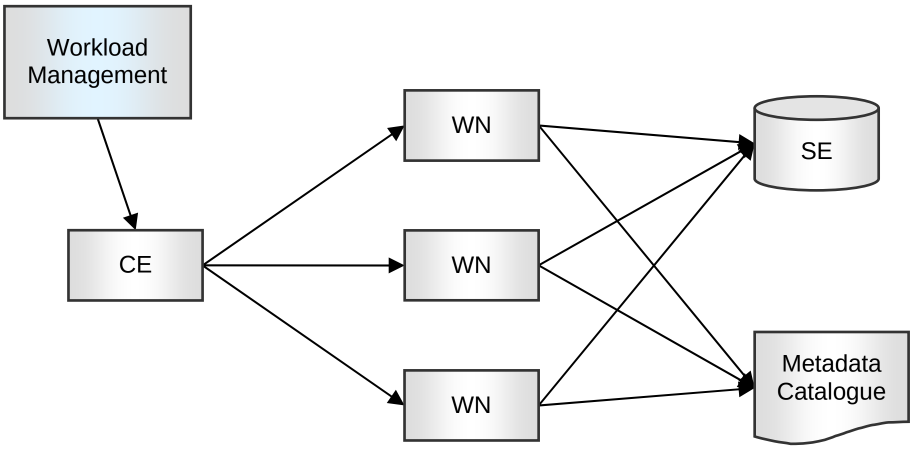


Figure 11 Metadata based architecture

The AMGA Metadata Catalogue [25] is the metadata service used in the gLite middleware. The AMGA stands for Arda Metadata Grid Application. It provides a complete but simple interface. It was design to allow scalability, and it can deal with large number of entries. Grid security is provided to grant different access levels to different users. Some of the AMGA features include the dynamic schemas, the hierarchical organization of collections, flexible queries and support for views, constraints and indexes.

User can modify schemas at runtime. The user can create or delete schemas, add or remove attributes, etc. The collections can contain sub-collections, and the sub-collections can inherit or extend the parent collection' schema. The query language is similar to the SQL language. It is also provided different join type, like inner, outer, left, right, between schemas.

For security, the AMGA uses UNIX style permissions, by creating users and groups. The SSL provides a secure client/server connection. The client authentication is based on username and password, general X509 certificates, and grid-proxy certificates.

The AMGA server comes with different client APIs for C++, Python, Java and Perl. Both the Java API and the C++ client include a command line interface.

AMGA provides a query language similar to the SQL query language. The biggest difference to SQL is that in AMGA's query language tables are referred to as references to directories. The access restrictions ensure that a user cannot infer data from queries if they have not read-access to that data.

AMGA commands:

```
find entry_pattern query_condition

selectattr column_1_query ... column_n_query query_condition

updateattr attr_1 update_query_1 .... attr_n update_query_n
query_condition

updateattr_single attr_1 update_query_1 .... attr_n update_query_n
query_condition
```

5.4 Data Movement

A *de fact* standard for data moving on Grid is the *GridFTP* [44][43]. This protocol is based on a FTP (File Transfer Protocol) and has some extensions based on that protocol. The most important features of the GridFTP protocol include:

- Support for multi source data transfers;
- Support for secure data transfer by using Grid Security Infrastructure (GSI) system;
- Third party managed data transfers are also supported;
- Support for partial file transfer;
- Support for large file transfer;
- Communication channel reuse.

In the case of server-side failure, the GridFTP transfer can be recovered. On the other hand in case of client-side failure there is no way to restart the transfer.

As well as the FTP protocol, the GridFTP uses two channels, a control channel and a data channel [24] (see Figure 12). The Control Channel uses a communication link (TCP) through which are transmitted the commands and the responses. It uses low bandwidth. The commands are transmitted in an encrypted and integrity protected way. The Data Channel represents the communication link for the actual data transmission. It uses high bandwidth. By default it is an authenticated channel. The encryption and integrity protection are optional.

The security is offered by the Grid Security Infrastructure (GSI) protocol used both by the GridFTP server and by the client. In order to improve the transfer speed the GridFTP protocol supports multiple TCP streams in parallel between source and destination. Another useful feature of the GridFTP protocol is the support for partial file transfer. In this way we can access regions of a file by specifying an offset and the desired block length.

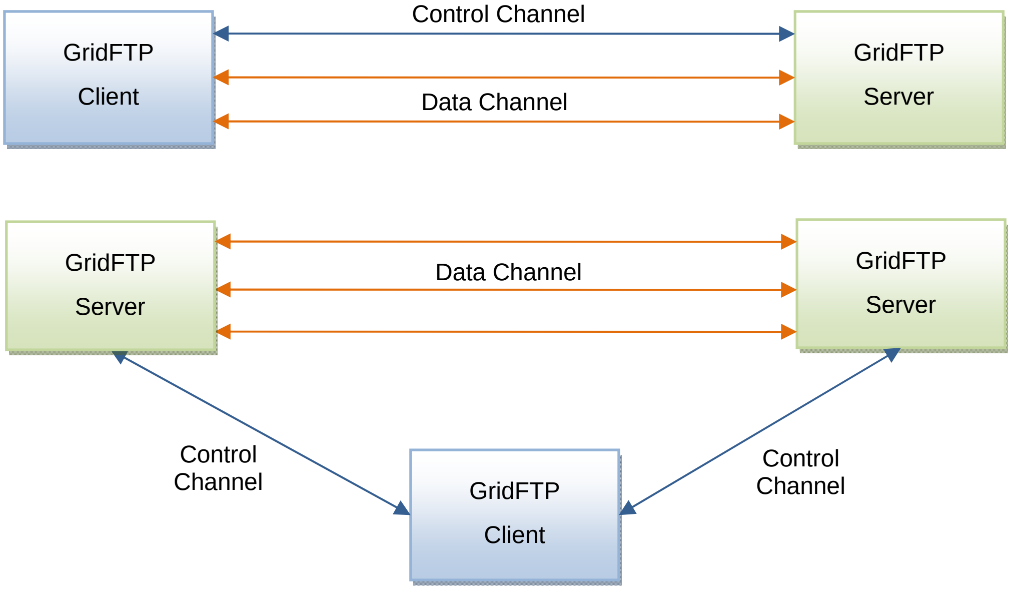


Figure 12 GridFTP Communication model

The GridFTP protocol offers third party transfers. In this case, a file can be copied between two GridFTP servers.

The *File Transfer Service* (FTS) [32] is a data movement service. For general users it provides a reliable point to point movement of files. For site managers it provides a reliable and manageable way of serving file movement. The FTS has three basic API group: job submission / tracking, service / channel management and statistics tools. Job submission is used to submit requests. The services are controlled by administrators by using service / channel management API.

The transfer security is accomplished by using user credentials. The authorization to service is done by using the Grid mapfile mechanism or by using VOMS role. Users submit jobs that can contain multiple copy requests. The feedback or the job status can be viewed with different details, for example can be requested only the overall job status (to determine if the job was completed), or can be requested detailed status of individual file status.

In the FTS service is defined the concept of channel. A channel is a link between two SEs. Channel administrators manage channels. Channels are uni-directional.

5.5 Data Discovery Services

One of the data services existing in Globus is the Replica Location Service (RLS). It is used to manage copies, or replicas, in the Grid environment. For every data stored there is defined a logical file name, and a physical file name. The logical file name is a unique identifier for the contents of a file, where the physical file name is the location of the copy in the storage system. RLS is a registry that maps the logical file names to the corresponding physical file names. The RLS registry can be distributed. In this way, we don’t have a single point of failure.

The RLS uses Local Replica Catalog (LRC) where information related to the logical and physical file names are stored.

The hierarchical structure of the RLS is made possible by using the Replica Location Index (RLI), which comprises of a network of connections between logical and physical file names.

5.5.1 OGSA-DAI

Open Grid Services Architecture – Data Access and Integration (OGSA-DAI) is a middleware that supports data accessing though web services [61]. The main goal is to offer a consistent, data resource-independent way to access, query, update, transform and deliver data resources. It provides access to distributed heterogeneous data resources, it exposes to users data in schema X as data in schema Y by data transformation, it integrates multiple databases as a single virtual database, and delivers data using different protocols (HTTP, FTP, GridFTP, etc.).

Users can expose their own data resources by extending web services. It is compliant with two web services specifications: Web Service Resource Framework (WSRF), and the recommendations Database Access and Integration Services of the OGF DAIS Working Group (WS-DAIR for relational data resources and WS-DAIX for XML data resources).

OGSA-DAI is a middleware bundle that allows data resources, such as file systems, relational or XML databases, to be accessed, federated and integrated across the network. It is an effective solution to a specific data management problem. OGSA-DAI workflows can represent complex data management scenarios, which involve data access, transformation, filtering, and updating of numerous distributed, heterogeneous databases.

OGSA-DAI is much more than another database connection layer between clients and data. It is a server-side data workflow engine designed for data integration, and federation and distributed query support across multiple types of database and file system. By providing an intermediate layer, OGSA-DAI:

- Allows data providers to retain control of their data.
- Provides a layer in which to hide database heterogeneities which can yield thinner clients.
- Provides a place to enforce additional security, for example to hide the actual location of the data, or filter the data according to the rights of clients, and manage access to federations, databases, tables, documents, files, rows, or lines of data.
- Provides a place to define views on read-only databases.

OGSA-DAI is also highly extensible and supports the use of application-specific data resources, activities, security, and presentation layers. It can be viewed as an enabling framework to save development time and allow users to focus on application-specific features, while getting standard functionalities out-of-the-box (e.g. queries, updates, transformations, deliveries). OGSA-DAI is full Java compatible and runs under Windows, UNIX, and Linux. It uses web services and the clients can be written in any language and on any platform that supports web services.

5.5.2 OGSA-DAI for gLite

The OMII-Europe project has developed a presentation layer for OGSA-DAI 3.0 for the gLite middleware called OGSA-DAI GL [61]. The presentation layer acts as an interface between the messaging framework being used and the core OGSA-DAI functionality. OGSA-DAI GL functions in a gLite 3.1 environment and makes use of a VOMS (Virtual Organization Membership Service) which furnishes attribute certificates.

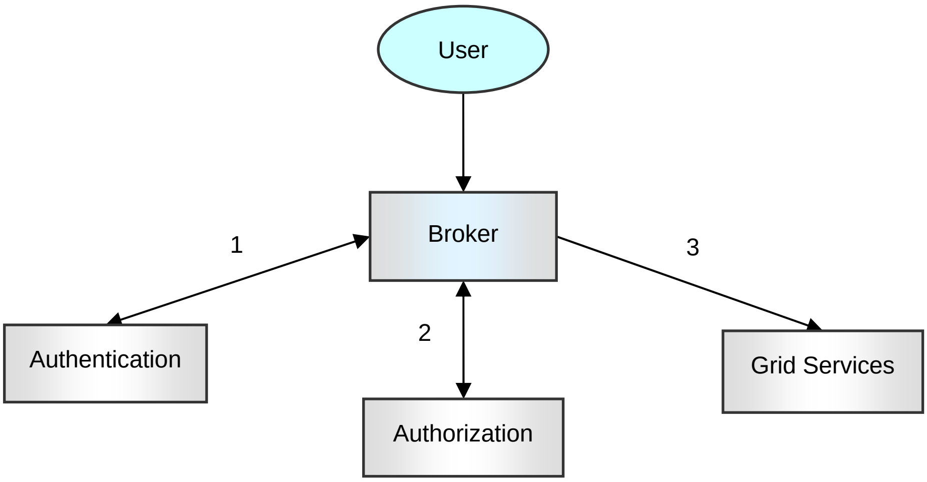


Figure 13 Secure access to Grid services

5.6 Security

Authorization is based on the Virtual Organizations (VO) concept, where a VO is a collection of users (each user with their role in the VO) and resources (computation, storage, etc.). In general, the VOs are based around users that share some common goals. The VO contains resources for a particular use and set of users.

Some security concepts [34][33]:

- Authentication – the way in which the identity of user / site is communicated
- Authorization – what can a user do?
- Encryption – encrypted messages
- Integrity – unchanged messages

The Grid Security Infrastructure (GSI) uses public key cryptography. The basic concept in the GSI authentication is the certificate. In particular, a certificate identifies every user or service.

Every entity (user/machine/software) from a particular VO has a pair of keys: a private key and a public key. A message that is encrypted by one key can be decrypted only by the other one. The sender of a message encrypts the message using the receiver’s public key. Only the receiver private key can decrypt the message.

In Grid, the public key is wrapped into a certificate file. Trusted third parties, known as Grid Certification Authorities (CA), issue these certificates. The private key is stored in an encrypted file that is protected by a passphrase. The pair private/public key is created by the Grid user.

Delegation allows remote process and services to authenticate on behalf of the user. This is achieved by creating a new key-pair from the user’s key-pair. This new key-pair is a single file, known as proxy credential. Proxys are valid for limited operations and they have a limited lifetime (usually 12 hours).

The proxy credential is created by using the *voms-proxy-init* command from the gLite UI.

The security workflow is the following (Figure 13):

1. The user is authenticated based on the certificate issued by a CA
2. The user is authorized by the VO in which is part of
3. Based on authentication and authorization the user can use Grid services

An X.509 Certificate contains the following:

- The owner's public key
- The identity of the owner (DN)
- Info on the CA, the issuer
- Time of validity
- Serial number
- Digital signature of the CA

6 Web Portals

6.1 Web Portal Overview

According to [7] a computational Grid consists of a set of resources involving computers, networks, on-line connected devices, data servers or sensors that are tied together by a set of services, which allow the users of the resources to view the collection as a seamless computing/information environment.

The standard Grid services include the following functionalities

- security services which support user authentication, authorization and privacy
- information services, which allow users to see what resources (machines, software, other services) are available for use,
- job submission services, which allow a user to submit a job to any compute resource that the user is authorized to use,
- co-scheduling services, which allow multiple resources to be scheduled concurrently,
- user support services, which provide users access to "trouble ticket" systems that span the resources of an entire grid.

To make Grid environment to work reliably and efficiently is a difficult task for many reasons. The first of them is the heterogeneity of sub-applications and resources that have to work in a consistent and predictable manner. Another difficulty is to control and synchronize the parallel and distributed execution of different components. There are complex interactions between middleware and application oriented components such as grid services, security, scheduling, and jobs and resources management.

There are three types of Grid application developers and users [7]. The first and the majority are end users who program pre-packed grid applications by using simple graphical or Web interfaces to supply application specific parameters and simple execution configuration data. They do not need to have knowledge about Grid. The second type of Grid programmers know how to build up an application by Grid components and services. The third type is the group of researchers who develop individual components, such a Monte Carlo generator, or a simulator. The first group of users works better by a Grid Portal that is a Web server through which the user configures and runs specific applications.

The Grid portal regarding technology and solutions have progressed by three known generations [7]:

1. The first generation of grid portals concerns with providing computational and data services using Grid-enabled tools in the form of APIs. Portals of this generation are mainly restricted to Grid middleware technologies.
2. The second generation of Grid portals concerns Grid portlets, which are pluggable software components and associated with backend Grid Services. The second generation of Grid portals is still under the development and standardization phase.
3. The third generation of Grid portals is focusing on semantic services as the Grid evolves towards the Semantic Grid. This portal is user centric, concerning on how to solve the user problems, instead of just solving the user problem. The portlets have semantic capabilities, and help user to identify and locate the required portlets.

6.2 First Generation Grid Portals

The first generation of Grid portals provides basic task oriented Grid services, and all the application functionality layers on these services composition [7]. The services provide user authentication and authorization, job submission, process monitoring, and data transfer. The portals are tightly coupled with Grid middleware tools. The portal developers have expertise in Grid technologies, Web and Grid services, and distributed applications. They should build the portal level and the graphical user interface of the application in such a manner that hides to the end user the complexity of Grid architecture and technology.

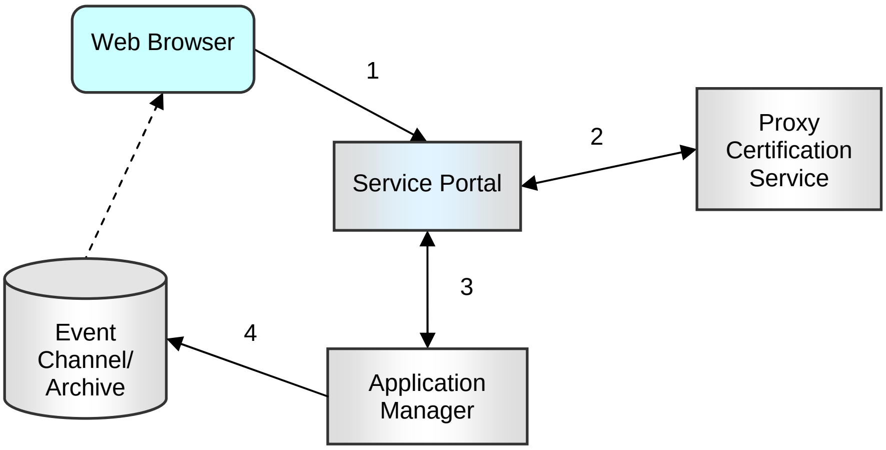


Figure 14 Flow control through Grid Portal [7]

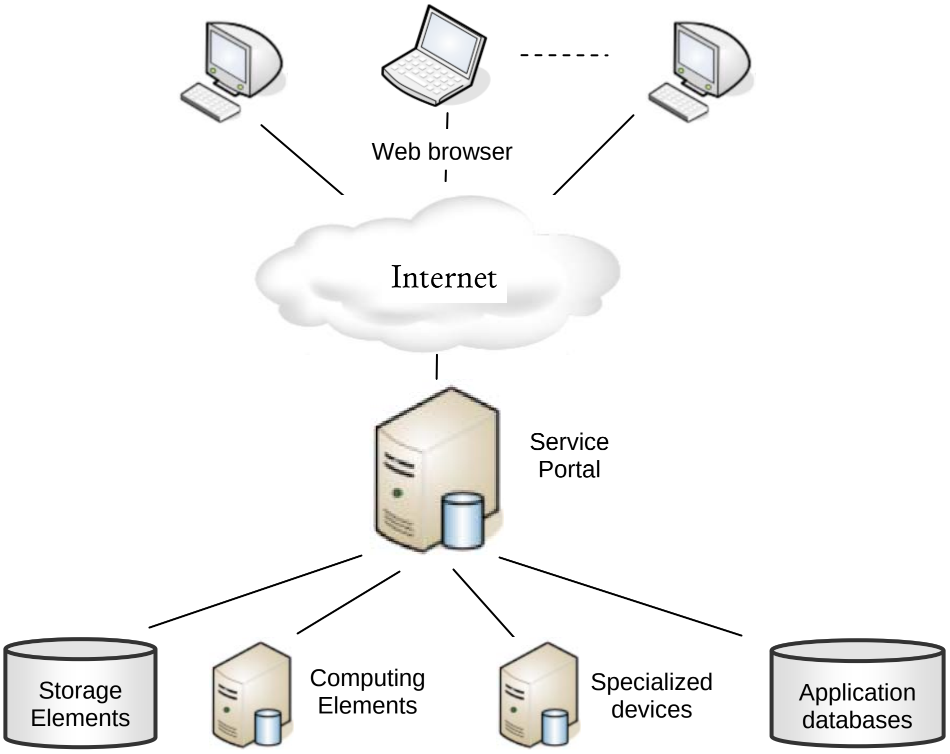


Figure 15 Three-tiered architecture of the Grid portal

In [6], [7] and [19] highlight the three-tiered architecture as the basic model of the first generation Grid portals (Figure 15). The grid portals have the following characteristics:

- The typical three-tiered architecture consists of the first tier of workstations running the Web browsers, a middle tier of Service Portal, which could be effectively a Web Server, and the third tier of backend resources such as storage elements, high computation resources, specialized devices, and application databases.

- The user makes a secure connection by the Web Browser to the Service Portal (step 1, Figure 14).
- The Service Portal then obtains a certificate from a proxy certificate server and uses that to authenticate the user (step 2).
- When the user completes defining the parameters of the task, the Service Portal launches an application manager (step 3) that controls and monitors the task execution over the Grid (step 4).
- The Service Portal delegates the user proxy credentials to the application manager as it may act on the user's behalf (steps 3 and 4).

The first generation Grid portals provide mainly the following functionality:

- *Authentication and Authorization*. Authentication allows the user to enter into the system. Authorization allows the user to access and interact with a particular resource. Essentially for a Grid Portal is to provide a single sign-on mechanism that support the user may access multiple distributed resources through a single authentication.
- *Job Management*. The user may manage the parallel and sequential tasks through the Grid Portal. The user launches the application by the Web Browser in a secure and reliable way, monitors the task status, and pauses and resumes the task execution.
- *Data Transfer*. The user may upload data files required for the execution of the tasks, and download the file resulted by the process execution.
- *Information Services*. The Grid Portal may search for resources required for a given task. The main information characterizing a resource could be: current load status, CPU configuration, free memory, operation system type, file system information, and network status.

6.3 Second Generation Grid Portals

The second generation of portals concerns with portlets. A portlet application consists of several portlets. Actually a portlet is a window in a portal that provides a specific service or information, for instance, a calendar or weather. The portal administrator could be able to install and remove the portlets, and can change the settings of a portlet. The portlets can be created and deleted automatically. For the application developers, the portlets are pluggable modules that are designed to run inside the portlet container of a portal server. The portlet container provides a runtime environment in which portlets are instantiated, used, and finally destroyed [1].

Portlets are specialized and more advanced form of Java Servlets, which run in a portal server environment. The portlets have features that are not available to Java Servlets. One important feature is the built-in support to automatically use of different JSP (Java Server Pages) for different devices. The user may write portlets for various devices, including desktop computers, and hand held devices such as PDA (Personal Digital Assistant) and mobile phones. The system can provide multiple interfaces to different device types for the same underlying business logic [7].

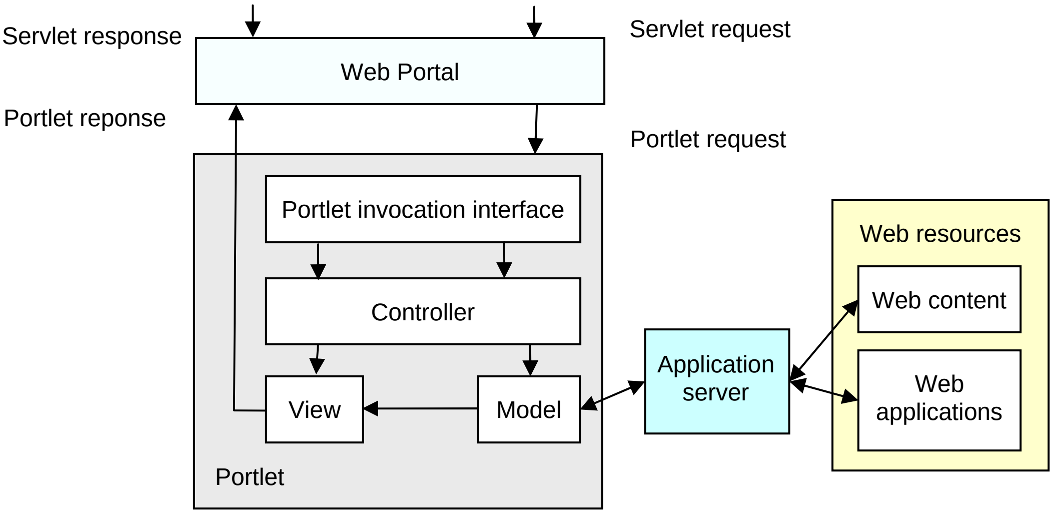


Figure 16 Web Portal by a portlet [7]

Portlets rely on the overall portal infrastructure to access user profile information, participate in window and action events, and communicate with other portlets, access remote content, and lookup credentials, and store persistent data.

There are two working groups that work on the standardization of the portlets: OASIS (Organization for the Advancement of Structured Information Standards), and JCP (Java Community Process).

Java Specifications Requests (JSR 168) is the Portlet specification that enables interoperability between portlets and portals. The specification defines the set of Portlet API for portal composition including the area of aggregation, personalization, presentation, and security.

OASIS WSRP states for Web Services for Remote Portals. The main objective of the WSRP is to define an XML and Web services standard that allows plug-and-play of visual Web services.

Figure 16 presents the access of Web services from Web Portal through a portlet. When the Web Portal receives a servlet request it generates and sends events to the portlet using parameters in the request and then invokes the portlet to be displayed through the portlet invocation interface. The portlet encapsulates the MVC (Model View Controller) model that has separated components for controlling the receiving requests and the invoking commands, for the model that describes application data and logic of accessing Web content and application, and finally, for views representing the results on the device.

In [7] the portlet is combined with a Grid service within a Grid Portlet (Figure 17).

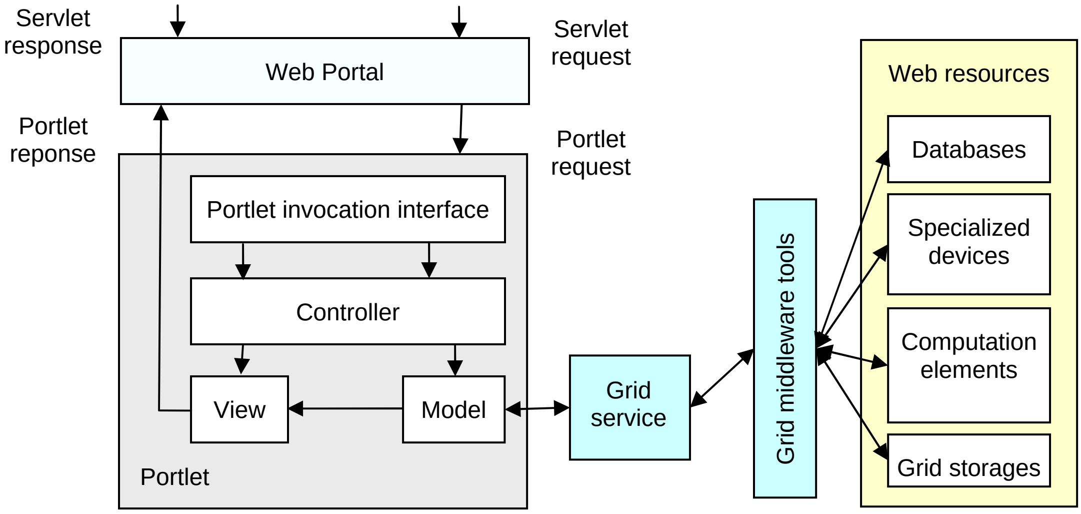


Figure 17 TheGrid Portlet concept [7]

6.3.1 Example of Portlets based Grid Portal

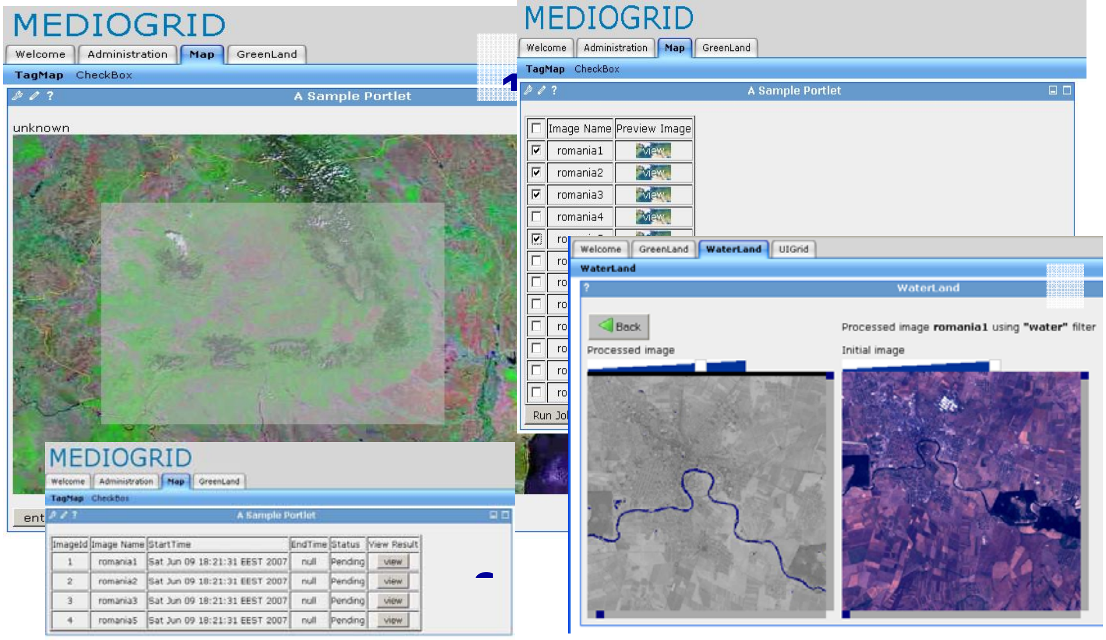
The following sections exemplify the MedioGrid Portal through which the user accesses the Waterland application. The application is available through Web browser, and the user may visualize and process huge and a great number of satellite images using remotely high computation resources provided by the Grid infrastructure and image repository.

6.3.2 Waterland Application

The Waterland application achieves the detection of water boundaries from satellite images [2]. The application uses an algorithm developed by Gond at al. [11], and uses the SWIR, Red and NIR spectral bands of Landsat images. The algorithm consists of several processing steps. In the first phase the NDVI and NDWI are computed from the input image. The next phases are composed of subtractions and means between intermediary result images. In the end the water areas are highlighted.

The Waterland portal allows the interaction between the user and the Grid system (Figure 18). This interaction consists in the selection of a region of interest (from which the user selects the images for the processing phase), visualization of the input satellite images as well as the result images (satellite images with the water areas highlighted) and monitoring the state of the submitted jobs (pending, active, clean up, and done).

MedioGrid [17][12] layers the user interface of these three applications on GridSphere that includes a set of fundamental portlets and services which provide the basic infrastructure necessary for the development and administration of web portals. The new interaction techniques enhance the GridSphere functionality by satellite imagery oriented features such as imagery zoom, scroll, selection by direct manipulation, automatically refresh, and layer management of satellite images and topological data (e.g. street network, rivers, administrative areas).



☐	Image Name	Preview Image
☑	romania1	view
☑	romania2	view
☑	romania3	view
☐	romania4	view
☑	ro	
☐	ro	
☐	ro	
☐	ro	
☐	ro	
☐	ro	
☐	ro	
☐	ro	

ImageId	Image Name	StartTime	EndTime	Status	View Result
1	romania1	Sat Jun 09 18:21:31 EEST 2007	null	Pending	view
2	romania2	Sat Jun 09 18:21:31 EEST 2007	null	Pending	view
3	romania3	Sat Jun 09 18:21:31 EEST 2007	null	Pending	view
4	romania5	Sat Jun 09 18:21:31 EEST 2007	null	Pending	view

Figure 18 Waterland Application available through MedioGrid Portal

6.3.3 MedioGrid Platform Kernel

MedioGrid Platform Kernel [3][12], [17] supports the general development of Grid applications and particularly of the environment domain applications. The kernel contains the basic software components and services which are used by the most environment oriented Grid applications:

- *MediogridService* is the component that supports as Web and Grid service the creation, the execution, and the scheduling of the jobs;
- *MediogridFactory* creates resource entities which keep information about the jobs. By using resources the MediogridService becomes a stateful service;
- *MediogridResource* contains information about the job state, the start and the end time of the job;
- *MediogridOGSADAI* provides access to the image database;
- *MediogridUrlCopy* supports the file transfer using the GridFTP protocol. By this service the worker fetches the image files;
- *MediogridRLS* returns a list of physical images in order to allow the worker to decide where is the most appropriate GridFTP location to download data.

The JobManager functionality is implemented in MedioGrid Kernel by Condor, which is used to schedule, manage and monitor every job created by the MediogridService. The use of Condor as job manager is just one of the possible solutions from others such as PBS and Sun Grid Engine. The MediogridService interface is described in WSDL language. The service provides functions for job specification and monitoring. The functionality of the job can be called through MediogridService from any stand-alone Grid client as well as from another Grid or Web service. By using resources the MediogridService becomes a stateful service. Each client that interrogates the service receives an endpoint reference to a particular resource which stores job information. MediogridFactory is responsible with the creation of new resources. Resources are entities which generally store some kind of information. Particularly in MedioGrid these resources keep information about the current state of the jobs.

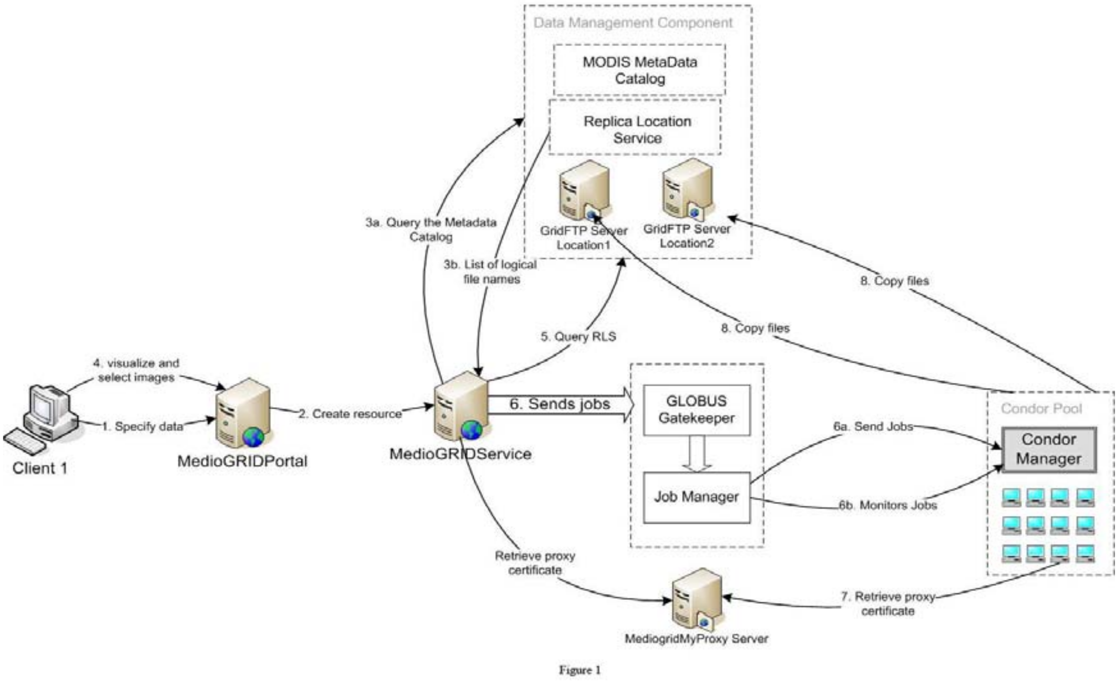


Figure 19 Flow control in MedioGrid system

6.3.4 User Interaction through Grid Portal and Information Flow

MedioGrid Software Platform supports the intermediate level between middleware and applications. The services are combined in order to develop environmental grid applications. The interaction between the user and the grid system consists of the following steps (Figure 19):

1. Through MediogridPortal the user specifies some spatial and temporal parameters for satellite images (step 1)
2. MediogridService creates a resource for every job (step 2)
3. MediogridService interrogates the MediogridOGSADAI service and receives a list of logical file names of images which match the specified characteristics (steps 3a, 3b)
4. The user selects all or a subset of images and the type of processing (step 4)
5. MediogridService interrogates the MediogridOGSADAI service and receives a list of locations of the selected images (steps 5)
6. MediogridService submits the jobs to the Condor pool (6, 6a)
7. Condor pool requires certificates to be allowed transferring data (step 7)
8. Each worker uses MediogridURLCopy and fetches the files that are needed for the computation phase (step 8)
9. The worker performs the necessary processing and uploads the results onto the database server
10. The user monitors the progress of the jobs by interrogating MediogridService (step 6b)
11. The user visualizes the results by MediogridPortal

6.4 Third Generation Grid Portals

The third generation focuses especially on the semantic services a portal could provide. These Grid Portals provide domain-specific and knowledge based services. The portlets have semantic capabilities that can be used to match a portlet candidate or a list of candidates.

6.5 EnviroGRIDS Portal

The enviroGRIDS Portal will expose to the user a set of tools and applications supported by the functionality and components provided through the services. There will be personalized tools for different category of users: system administrator, data provider, Earth Science specialist, decision maker and citizen. Each category of user have particular accessing rights to data and system resources.

The enviroGRIDS portal could adopt a flexible architectural solution which combines the first and the second generations of the Grid portals. It combines the Web services and portlets encapsulating the Grid services. The real solution depends on the practical achievements of the implementation of the interoperability between geospatial and Grid infrastructures.

Functionalities offered by the Web Portal:

- Gather data resources – The Web Portal should allow integration of data resources and creation of metadata which is associated with data
- Visualize data resources – Depending on the access rights a user can search, visualize, download desired data resources
- Administrate data resources – Copy data from one location to another, rename files, create replicas, manage replicas, etc.
- Define scenarios – define scenarios for the SWAT model by defining parameters, choose input resources, etc.
- Execute Earth Science oriented applications
- Run scenarios – execute scenarios over Grid
- Administration tools – management of users, management of portal, etc.

Each user category may access different resources through particular accessing rights:

- Administrator – manages user accounts, creates new users, updates information on the Web Portal, manages data resources, set up virtual organization, administrate the virtual organization membership, etc.
- Data provider – enters datasets in repositories, fills in metadata, visualizes selected data, manages stored data, etc.
- Earth Science specialist – describes and executes SWAT scenarios. Develop use cases (datasets, scenarios, data visualization, results visualization) for particular studies.
- Decision maker – visualizes and analyzes the model execution results for various scenarios. Visualizes statistics on results provided by different use cases.
- Citizen – visualizes data and reports, etc.

7 SWAT - Soil Water Assessment Tool

7.1 Overview

The main processing throughout the enviroGRIDS system is related with SWAT based computation.

SWAT (Soil Water Assessment Tool) is a process based water quality model [66][64]. It is designed to estimate impacts of land management practices on water quantity and quality in complex watersheds. The time stamp on which the model runs is the day time. Complex watersheds are divided into sub-basis. Each sub-basin has relatively uniform land cover and soil properties. Each sub-basin is represented by a one dimensional soil profile, which is connected to down stream profiles by overland flow channels. The sub-basins for a watershed can be spatially referenced one to another.

The SWAT model requires specific information about weather, soil properties, topography, vegetation, and land management practices in watershed. SWAT models water movement, sediment movement, crop growth, and nutrient cycling with this information. Therefore, watersheds with no monitoring data can be modeled, and the effect of changes in input data can be quantified.

The SWAT Model Calibration consists of three steps: Water Balance and Stream Flow, Sediment, and Nutrients.

The development of SWAT model started in the early 1990s. A series of review or expansions were introduces. The mostly used SWAT models are SWAT94.2, SWAT98.1, SWAT99.2, SWAT2000, and SWAT2005.

7.2 SWAT Inputs

1. *Watershed delineation* - define the elevation related properties such as: elevation above sea level, slope, aspect, stream flow network, and distance to nearest stream, and dividing the basin in sub-catchments.
2. *Land use* - The type of vegetation determines many components of the hydrological cycle: total water requirement, irrigation demand, and actual water consumption by evapotranspiration, surface runoff, percolation, and erosion. The number of vegetation types and crops that can be included in SWAT is virtually unlimited. For practical reasons it is important that the level of detail is in correspondence with the question to be answered during the modeling.
3. *Soils* - Soils are the determining factors for hydrological processes such as: surface runoff, infiltration, percolation, lateral subsurface flow, plant water availability, etc.
4. *Hydrological Response Units* - A specific characteristic of the SWAT model is the subdivision of the study area in so-called Hydrological Response Units (HRUs). These HRUs form unique combinations of a specific soil type and land cover type within a sub-catchment.
5. *Irrigation and Reservoirs* - SWAT includes a module for auto-irrigation where the source of water should be specified. A total of five different sources can be specified: canal water, reservoir, shallow aquifer, deep aquifer, or a source outside the basin.

7.3 Model Components

1. *Hydrology*: based on the water balance equation – which is based on: Surface runoff, precipitation, percolation, evapotranspiration & return flow
2. *Weather*: based on: Precipitation, air temperature, soar radiation, wind speed & relative humidity
3. *Sedimentation*: uses the Modified Universal Soil Loss Equation (MUSLE) model: Surface runoff, peak rate flow, soil erodibility, crop management, erosion practices, slope length & steepness
4. *Soil Temperature*: simulated using: Daily maximum and minimum air temperature; & snow, plant, and residue cover for the day of interest plus the four days immediately preceding
5. *Crop Growth*: is based on the EPIC model, which is based on: Daily accumulated heat units, harvest yield, biomass from solar radiation and water and temperature stress adjustment

6. *Nutrients*: Nitrate, organic nitrogen, soluble phosphorus, & organic phosphorus
7. *Pesticides*: uses the GLEAMS model; each has a unique set of parameters: Solubility, half life in soil and foliage, wash-off fraction, organic carbon adsorbsion coefficient & cost
8. *Agricultural Management*: sub-models that simulate: Tillage, irrigation water application, fertilizers, pesticides, and grazing systems

7.4 SWAT input files

Even if the data is provided by partners in various formats and files, it must be formatted into SWAT input files. This chapter briefly presents the SWAT model input files [68]. These inputs for the SWAT model can be grouped by different levels of detail: watershed, sub-basins, HRU and reservoirs.

7.4.1 Watershed level files

- *Watershed configuration file* (.fig) – defines the routing network in the watershed;
- *Basin input file* (.bsn) – defines variables that models physical processes for the entire watershed;
- *Precipitation input file* (.pcp) – daily measured precipitation; optional file;
- *Temperature input file* (.tmp) – daily measured maximum and minimum temperatures; optional file;
- *Solar radiation input file* (.slr) – daily solar radiation; optional file;
- *Wind speed input file* (.wnd) – daily average wind speed; optional file;
- *Relative humidity input file* (.hmd) – daily relative humidity values; optional file;
- *Potential evaporation input file* (.pet) – daily PET values for the watershed; optional file;
- *Weather forecast input file* (.cst) – statistical data used to generate daily climatic data for the subbasins; optional file;
- *Auto-calibration input file* (.cal) – contains data used by the auto-calibration algorithms; optional file;
- *Land cover/plant growth database file* (crop.dat) – contains plant growth parameters;
- *Tillage database file* (till.dat) – contains information about the amount and depth of mixing cased by tillage operations;
- *Pesticide database file* (pest.dat) – information about mobility and degradation for all pesticides simulated in the watershed;
- *Fertilizer database file* (fert.dat) – information on the nutrient content of fertilizers simulated in the watershed;
- *Urban database file* (urban.dat) – information on the build-up/wash-off of solids in urban areas;
- *Watershed water quality input file* (.wwq) – contains information used to model QUAL2E transformations in the main channels; optional file;

7.4.2 Sub-basin level files

- *Subbasin input file* (.sub) – defines climatic inputs and the number and types of HRUs in the subbasin;
- *Weather generator input file* (.wgn) – statistical data used to generate daily climatic data for a subbasin;
- *Pond/wetland input file* (.pnd) – contains information for impoundments; optional file;
- *Water use input file* (.wus) – contains information for consumptive water use; optional file;

- *Main channel input file* (.rte) – contains parameters that define water and sediment movement;
- *Stream water quality input file* (.swq) – contains information used to model pesticide and QUAL2E nutrient transformations; optional file;

7.4.3 HRU level files

- *HRU input file* (.hru) – HRU level parameters;
- *Management input file* (.mgt) – contains management scenarios and specifies the land cover;
- *Soil input file* (.sol) – contains information on physical characteristics of the soil;
- *Soil chemical input file* (.chm) – contains information on intial nutrient and pesticide levels of the soil;
- *Groundwater input file* (.gw) – contains information about the shallow and deep aquifer in the subbasins;

7.4.4 Reservoir files

- *Reservoir input file* (.res) – contains parameters used to model the water and sediment movement through a reservoir; optional file;
- *Lake water quality input file* (.lwq) – contains parameters used to model the nutrients and pesticides movement through a reservoir;

7.5 SWAT Output files

The SWAT output files consist of the following:

- the summary input file (input.std)
- the summary output file (output.std)
- the HRU output file (output.hru)
- the subbasin output file (output.sub)
- the main channel output file (output.rch)

7.6 Execution workflow

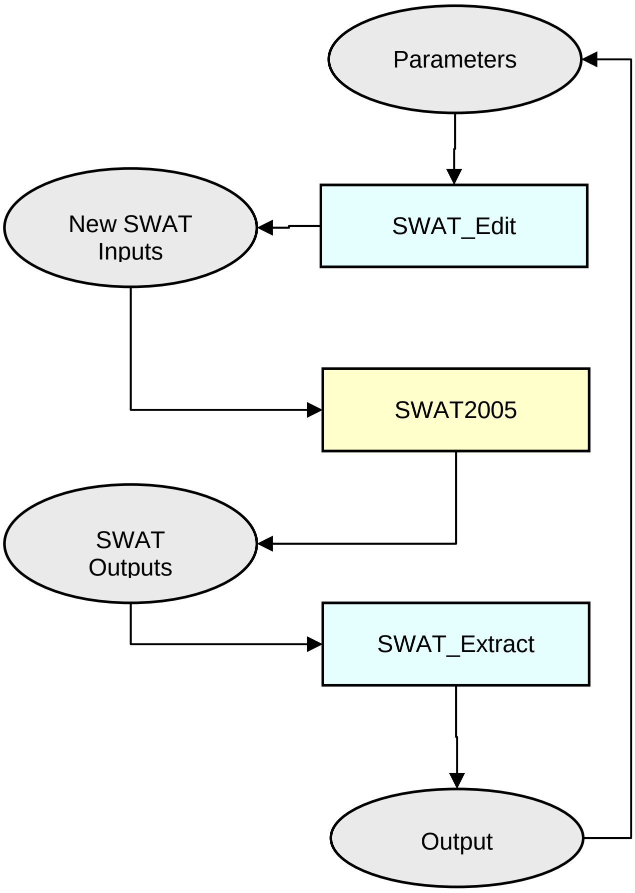


Figure 20 SWAT Calibration Uncertainty Analysis

The main computation of the enviroGRIDS system concerns with SWAT based processing. The Grid execution of the SWAT model consists of a set of user interactions between the user and the SWAT Web Service.

- The user selects a predefined scenario through the Web Portal. The scenario runs on already specified input data.
- The other solution for the user is to create a scenario by specifying the input data, and the parameters required by the SWAT model related Grid processing. The user accesses the Web Portal Data Management functionality which allows him to query the metadata catalogue and to select the input dataset
- According with the specified scenario the system invokes the specific SWAT model execution service
- The SWAT Web Service, based on the input dataset, creates a description file (e.g. JDL format) that is used to run the SWAT model
- The SWAT description file is used by the SWAT executor service that launches and manages the SWAT model execution on CE nodes over the Grid
- The job sends the results to data repository and updates the metadata catalogue. The portal interactive tools support the user to visualize and analyze the results

One possible solution of the SWAT model gridification and execution could be the GANGA software package [37][36].

8 Technologies

8.1 gLite Technology Available in EGEE

8.1.1 gLite Data Management System

The gLite Data Management System consists of a set of data services and data tools. Storage, catalogs, and movement are the basic data services available in the gLite. The Storage Element is the service which allows a user or an application to store data for future retrieval. It manages local storage and interfaces to Mass Storage Systems. It is able to manage different storage systems uniformly and transparently for the user.

The Grid applications often require and generate a huge amount of data. Therefore, an efficient mechanism for describing and localization of data is required. This problem can be solved by associating descriptive attributes to the files and answering to the user queries against the associated information. The AMGA Metadata Catalogue is the metadata service used in the gLite middleware. It provides a complete but simple interface. It was design to allow scalability, and it can deal with large number of entries. Grid security is provided to grant different access levels to different users. Some of the AMGA features include the dynamic schemas, the hierarchical organization of collections, flexible queries and support for views, constraints and indexes. The set of services that are needed to provide file access and storage makes up what we call a Storage Element. In essence, the Storage Element is the Grid service responsible for saving/retrieving files to/from some data store which can provide a wide range of quality of service to the user. The interface for the management of disk and tape storage resources is the Storage Resource Manager (SRM). All the Storage Elements, except the Classical SE, offer a SRM interface. By using this interface the user can request files, store them on disk buffers, etc. In the gLite middleware the files are identified by the user based on their logical file names. The LFNs are managed by the

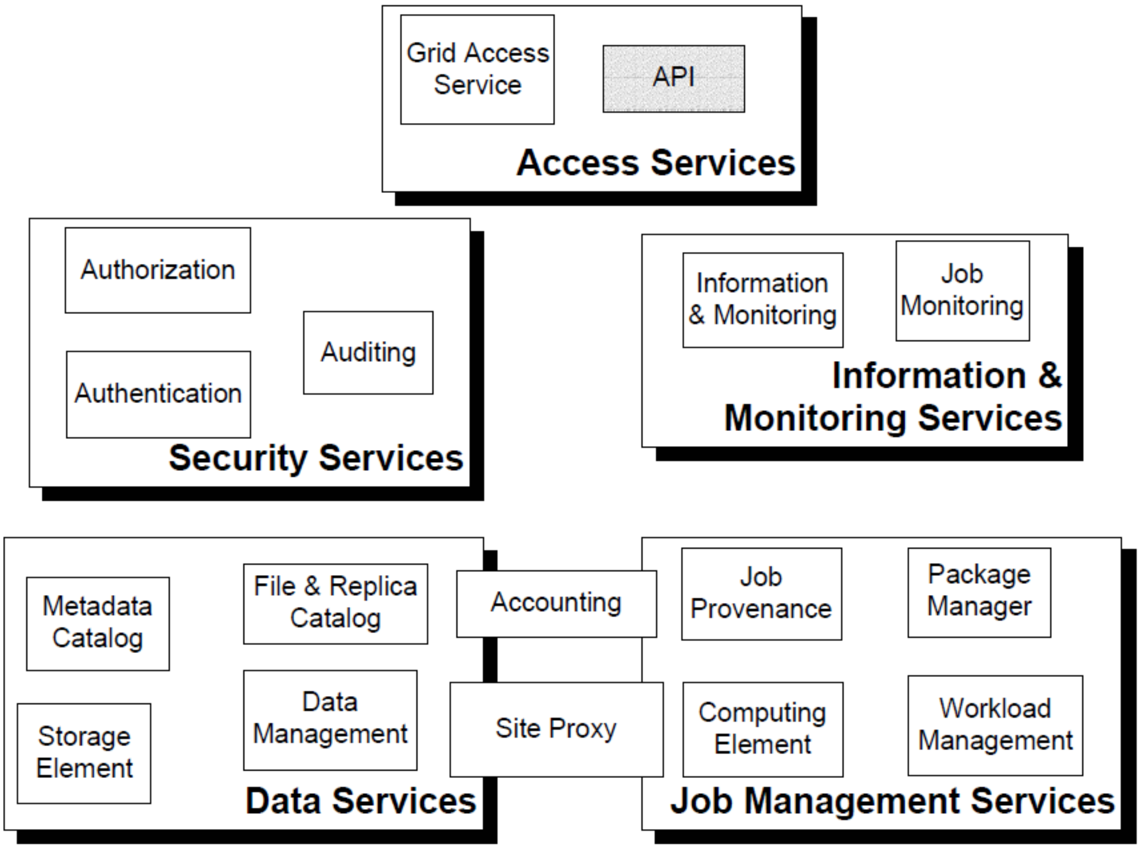


Figure 21 gLite Data Management System

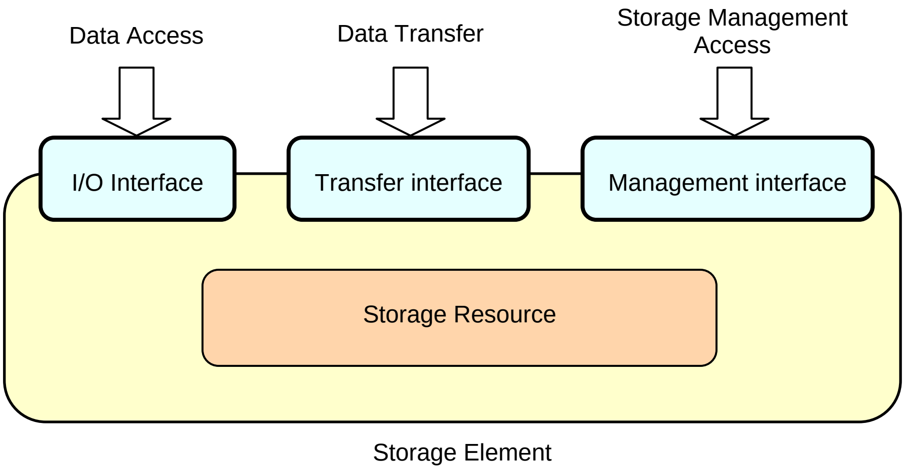


Figure 22 Storage Element conceptual architecture

File Catalog interface. The LFN name is based on the VO specific namespace that maps a hierarchical directory structure. The File Catalogue service maintains mappings between LFN(s), GUID and SURL(s). LFC could either be used as a Local File Catalogue, holding only replicas stored at a given group of site, or a Global File Catalogue, containing information about all files in the Grid. Data catalogues are responsible with the management of the Grid file namespaces, location of files, also to store information about the data and the associated metadata, and to keep data authorization information.

8.1.2 Storage Element

In a Grid environment the data and its availability is managed by the Grid middleware. The middleware interfaces to data storage need to be rich enough to be able to take into account the possible semantic properties of the underlying storage resource hardware. At the same time its usage should be simple enough so that clients of the storage resource do not need to be experts in the handling of it. The interface to data storage needs to abstract out the commonalities of data stores while allowing the usage of specific features that may be available only on a small subset of storage resources.

Storage is tightly coupled to all other aspects of the Grid architecture. Every data access and data movement needs to take the data resource availabilities into account. Storage is also tightly coupled to Grid scheduling. Grid Schedulers need to be able to take the data requirements of a job into account. In order to be able to do so, the storage resources need to expose appropriate interfaces as well. If data resources provide means of storage scheduling and reservation, these need to be exposed accordingly as well. Finally, like any other resource, storage resources also need to be monitored, audited and its usage has to be accountable.

The storage resource is a combination of storage hardware and storage software (for example file systems), and offers a place to store data [24]. Different storage resources solution offers different Quality of Service (QoS) levels. In general, the storage hardware is controlled by software.

A user or an application can store data in a Storage Element [35]. All of the stored data from a SE must be considered read-only, and that data can be changed only by physically remove or replacement. In EGEE there are 4 types of Storage Elements: Classic SE, Castor, dCache and LCG Disk pool manager.

The Storage Element (Figure 22) wraps a level of storage services over the storage resource in order to make the data available in the data architecture. It offers three types of interfaces:

- Data access (ByteIO, DAI, etc.)
- Data transfer (DMIS, GridFTP)
- Storage Management (SRM)

The Classic SE consists of a GridFTP server and an insecure RFIO daemon in front of a physical single disk or disk array. The CASTOR consists of a tape mass storage system that has a disk buffer frontend. The complexity of the tape is hidden to the user by a virtual filesystem (or namespace). A special process, called "stager" manages the migration between disk and tape. Since the storage protocol (insecure RFIO) is not GSI-enabled, only RFIO access from a location in the same LAN of the SE is allowed. dCache consists of a server and one or more pool nodes. The files stored in the SE can be accessed from a single point of access, represented by the server. A single virtual filesystem tree is used to represent the files stored in the pool disks. Another feature of this system is that new nodes can be added dynamically to the pool. LCG Disk pool manager is suitable for relatively small sites (max 10 TB of total space). Like in the dCache, new nodes can be added dynamically to the pool. In addition, a virtual filesystem hides the complexity of the disk pool architecture.

The set of services that are needed to provide file access and storage makes up what we call a Storage Element (SE):

- Storage back-end with all the associated hardware and drivers
- SRM service implementation on the given storage
- Transfer service for a (set of) transfer protocol(s) but at least GridFTP
- Native file access service exposing a POSIX-like I/O interface
- Auxiliary security, optional logging and accounting services

In essence, the Storage Element is the Grid service responsible for saving/retrieving files to/from some data store which can provide a wide range of quality of service to the user.

8.1.3 Storage Resource Manager

The interface for the management of disk and tape storage resources is the Storage Resource Manager (SRM). All the Storage Elements, except the Classical SE, offer a SRM interface. By using this interface the user can request files, store them on disk buffers, etc. In addition, the SRM offers a third party transfer protocol between different endpoints.

In the Grid environment the files are no longer permanent entities. They can be created or removed dynamically according to the specification of the user. The SRMs invoke middleware components for file transfer (such as GridFTP).

The SRM services offer a standard interface to hide characteristics and to allow interoperability.

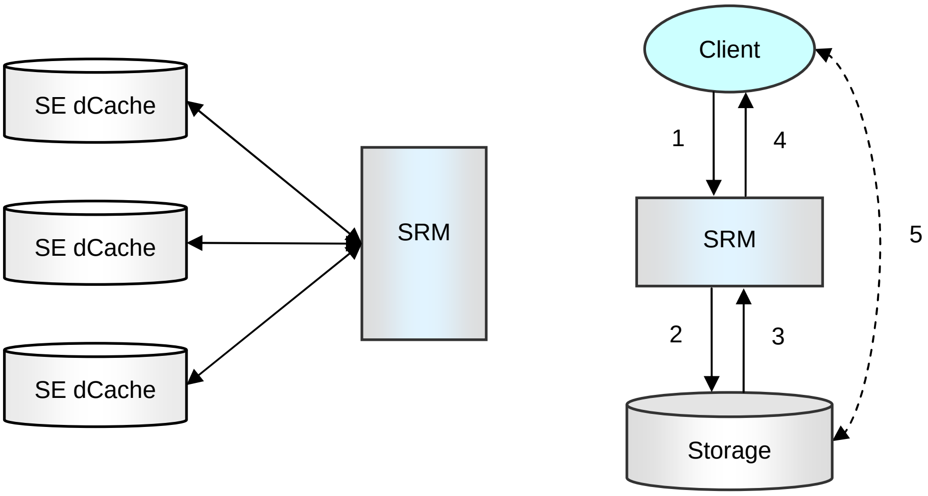


Figure 23 Storage Resource Manager related architecture and its interaction cycle

SRM interactions (Figure 23**):**

1. The client asks the SRM for the file providing an SURL (Storage URL)
2. The SRM asks the storage system to provide the file
3. The storage system notifies the availability of the file and its location
4. The SRM returns a TURL (Transfer URL), i.e. the location from where the file can be accessed
5. The client interacts with the storage using the protocol specified in the TURL

A Storage Element exposes a set of access and control protocols. The SRM interface exposes the control protocol that is implemented by a SRM service (SRM-dCache, SRM-Castor, etc.).

8.1.4 File Names in gLite

In the gLite middleware the user identifies the file by its logical name (LFN – Logical File Name).

LFN (Logical File Name) – represents a logical identifier for a file (user alias). It is unique and mutable (a user can change the file name). The namespaces are built hierarchically and represents a convention for every Virtual Organization.

Example:

```
lfn:<any_string>
lfn:importantResults/Test1240.dat
```

GUID (Global Unique Identifier) – this logical identifier is guaranteed to be unique based on the UUID mechanism. The relation between LFNs and GUIDs is 1:1, for each LFN only one GUID is created. One of the differences from the LFN is that are not mutable, so that a user cannot change this identifier.

Example:

```
guid:<36_bytes_unique_string>
guid:38ed3f60-c402-11d7-a6b0-f53ee5a37e1d
```

Logical Symlinks - Symbolic links always point to an LFN. There may be many Symlinks to an LFN (N:1 relation).

SURL (Storage URL) - specifies the file replicas. One file can have multiple replicas, so that the relation is 1 to many. Every replica has its own SURL. In gLite, SURLs are always fully qualified SRM names, accepted by the Storage Element's SRM interface.

Example:

```
<sfn|srm>://<SE_hostname>/<some_string>
```

In the case of the sfn prefix, the string after the host name is the path to the location of the file and can be decomposed in the SE's access-point (path to the storage area of the SE), the relative path to the VO of the file's owner and the relative path to the file.

Example:

```
sfn://<SE_hostname><SE_Accesspoint><VO_path><filename>
sfn://tbed0101.cern.ch/data/dteam/doe/file1
```

In general, SRM-managed SEs can use virtual file systems and the name a file receives may have nothing to do with its physical location.

Example:

```
srm://srm.cern.ch/castor/cern.ch/grid/dteam/doe/file1
```

TURL (Transport URL) - It is a URL that can be used to actually transfer a file using any standard transport protocol. The TURL is a fully qualified URL starting with the protocol to be used for transfer or direct file access through some native I/O mechanism.

Example:

```
<protocol>://<some_string>
gsiftp://tbed0101.cern.ch/data/dteam/doe/file1
```

The mapping between the SURL and the TURL is managed by Storage Resource Management (SRM) service.

The LFN name is based on the VO specific namespace that maps a hierarchical directory structure. For example a valid LFN name can be the following:

```
/glite/myVO.org/production/run/07/123456/calibration/cal/cal-table100
```

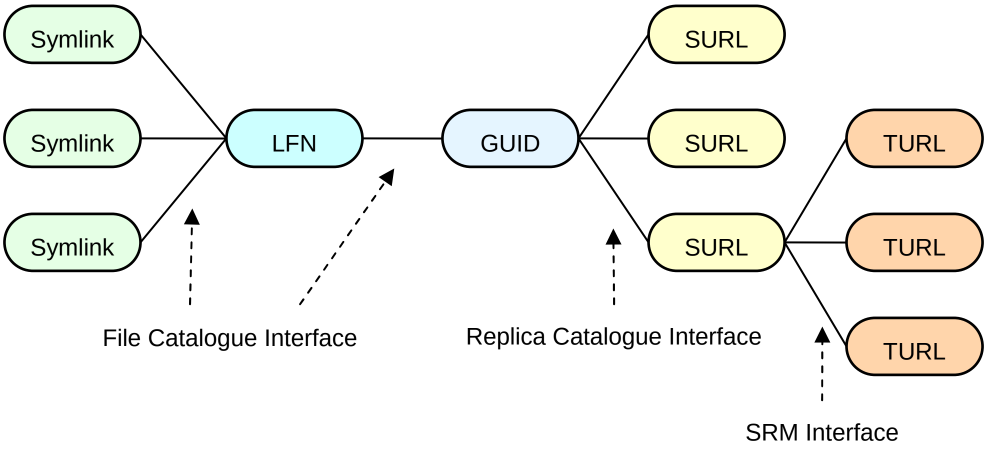


Figure 24 Relationships between different file name types

The LFNs are managed by the File Catalog interface. The gLite Fireman catalog implements this interface. It is responsible for assuring the uniqueness of the LFN. If the same catalog is to be shared among several Virtual Organizations, then the LFN uniqueness has to be assured across VOs. One solution is to include the VO name in the namespace, creating different branches for every VO. Anyway, the uniqueness must be also applied to VO names.

Another concept that is included in the LFN namespace is the directory. The directory may be manipulated just as in a normal file system. It can be listed in the same manner, new entries may be added to it or existing ones renamed and removed. In the previous example LFN, one of the directories is

```
/glite/myVO.org/production/run/07/123456/calibration/
```

A user can also copy logical files from other logical directories into their own directory.

The LFNs and GUIDs must be unique. And both of them represent logical identifiers. The LFN has a human-readable format and also it is mutable. On the other hand the GUIDs are generated and they don't have a human-readable form, and are immutable. The LFN namespace has a hierarchical structure whereas the GUIDs expose no structure at all. The use of LFNs is not mandatory, VOs may decide to use only the GUIDs to identify their files.

8.1.5 File Catalogue in gLite

The File Catalogue service maintains mappings between LFN(s), GUID and SURL(s). In the gLite 3.1 the used File Catalogue is the LCG File Catalogue (LFC). The Data Management tools can discover the catalogue by its endpoint stored in the Information Service. LFC could either be used as a Local File Catalogue, holding only replicas stored at a given group of site, or a Global File Catalogue, containing information about all files in the Grid.

Data catalogues are responsible with the management of the Grid file namespaces, location of files, also to store information about the data and the associated metadata, and to keep data authorization information. The catalogue features are exposed through a series of interfaces:

- *Authorization Base* - methods to set and get permissions on one or more catalog entries
- *Metadata Base* - setting, querying, listing and removing metadata attributes to one or more catalog entries
- *Metadata Schema* - define attributes and group them into schemas

- *Replica Catalog* - expose replication operations (for example to list, add and remove replicas to a file identified by its GUID)
- *File Catalog* - operations on the logical file namespace (create directories, rename logical file entries, manage symbolic links)
- *File Authorization* – contains methods needed to implement a standalone authorization service
- *Metadata Catalog* - add and remove entries
- *Storage Index* – return the list of Storage Elements that contains a replica (identified by its LFN or GUID)

In gLite are available two kinds of LFC [24], a Central LFC and a Local LFC. In the Central LFC, for each VO, one site will publish a global catalogue. This catalogue will record file replicas or dataset entities across the whole Grid. In the Local LFC, local catalogues record the file replicas stored at that site's SEs only.

The LCG File Catalogue provides bulk operations, timeouts and retries for client operations, hierarchical namespace and namespace operations, integrated GSI Authentication and Authorization, etc. LFC stores both logical and physical mappings for the file in the same database obtaining in this way a speed up of operations. This catalogue Treats all entities as files in a UNIX-like file system, and uses a File API also similar to UNIX.

8.1.6 gLite Commands

8.1.6.1 Lfc commands

lfc-chmod: Change access mode of the LFC file/directory

lfc-chown: Change owner and group of the LFC file-directory

lfc-delcomment: Delete the comment associated with the file/directory

lfc-getacl: Get file/directory access control lists

lfc-ln: Make a symbolic link to a file/directory

lfc-ls: List file/directory entries in a directory

lfc-mkdir: Create a directory

lfc-rename: Rename a file/directory

lfc-rm: Remove a file/directory

lfc-setacl: Set file/directory access control lists

lfc-setcomment: Add/replace a comment

8.1.6.2 Replica management

lcg-cp: Copies a grid file to a local destination

lcg-cr: Copies a file to a SE and registers the file in the catalog

lcg-del: Delete one file

lcg-rep: Replication between SEs and registration of the replica

lcg-gt: Gets the TURL for a given SURL and transfer protocol

lcg-sd: Sets file status to "Done" for a given SURL in a SRM request

8.1.6.3 File catalogue interaction

lcg-aa: Add an alias in LFC for a given GUID

lcg-ra: Remove an alias in LFC for a given GUID

lcg-rf: Registers in LFC a file placed in a SE

```
lcg-uf: Unregisters in LFC a file placed in a SE
lcg-la: Lists the alias for a given SURL, GUID or LFN
lcg-lg: Get the GUID for a given LFN or SURL
lcg-lr: Lists the replicas for a given GUID, SURL or LFN
```

8.2 GENESI-DR

GENESI-DR Project [38] offers a framework allowing scientists from different Earth Science disciplines to have access, to combine and to integrate historical and present Earth-related data from space, airborne and in situ sensors available from all digital repositories dispersed all over Europe.

The GENESI-DR Project aims to:

- Provide a base for establishing a world-wide e-infrastructure for Earth Science repositories
- Provide reliable, easy, effective, and operational access to a variety of data sources (space and ground)
- Harmonize operations at key Earth Science data repositories
- Demonstrate effective curation and prepare the frame for long term preservation
- Validate capabilities to access distributed repositories for involving new communities, including education…
- Integrate new scientific and technological derived paradigms

OpenSearch [64] was adopted and promoted in GENESI-DR as the minimal compliance level to develop discovery mechanisms in heterogeneous sites

The GENESI-DR system consists of several sub-systems such as User Interface, Data Management, Security, User Management, Services Management and Workflow Management (Figure 25) [26].

The *User Interface* consists of a Web Portal through which the users can access the GENESI-DR repository. It offers a workspace from which the users has control to search, store and publish data products.

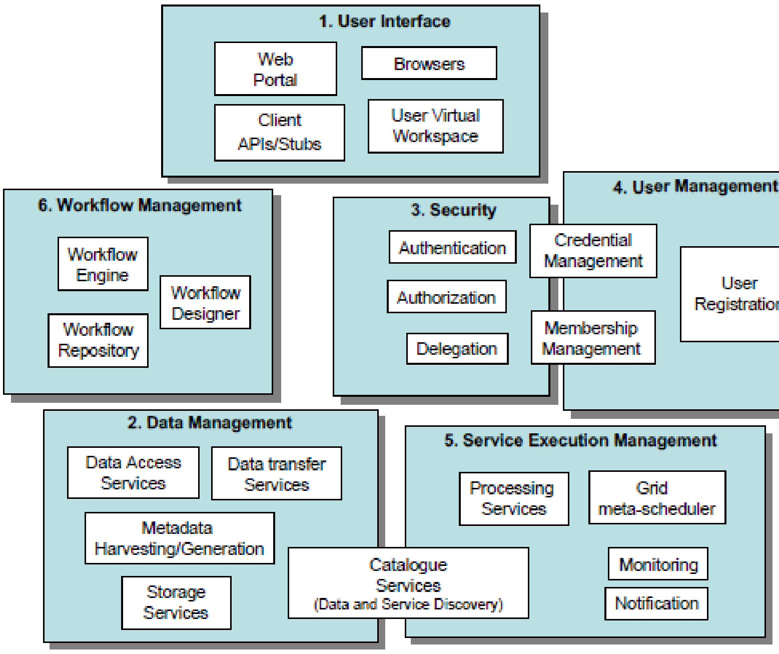


Figure 25 GENESI-DR layered architecture

The *Data Management* sub-system handles metadata harvesting, cataloguing, publishing and storage management of observational and user-generated information.

The Security components are responsible with the Authentication, Authorization and Delegation services that enable the identification of entities (users, systems and services), allow or deny access to services and resources. The GENESI-DR security architecture relies on the Public Key Infrastructure (PKI) available via the X.509 certificate model that uses a concept of trusted third parties (Certification Authorities or CAs) that issue the identity credentials. All users and services are expected to have a X.509 credential.

The registration, storage and management of user information are accomplished by the User Management sub-system. The User Management sub-system is tightly linked with the security infrastructure

The *Services Management* sub-system handles execution, access, monitoring and control of processing services on remote heterogeneous computers.

The *Workflow Management* sub-system allows the composition of services by defining dependencies among them. The interaction between services is expressed in a workflow description that is executed by the Workflow Engine.

The Catalogue Access Service (CAS) is a web service gateway. Through this gateway, the external clients have access to the GENESI-DR Catalogue. The authentication of the users is also the responsibility of this service.

Data access is supported using different protocols and services, GridFTP, HTTPS, BitTorrent.

The Genesi-DR system support different services and functionalities:

- *Web Portal* – offers a single access point for Earth Science information and services, a directory to locate data and resources, direct access to earth-observing satellite data as well as map servers and cartographic resources.
- *Centralised data discovery* – data products and data collections can be discovered through a central querying information point
- *Fast data retrieval* – services allowing the user to directly retrieve data products without any (or transparent) ordering stage. It supports different file transfer and access protocols / mechanisms (e.g.

gridFTP, HTTPS, BitTorrent, OPeNDAP, WCS, WMS, WFS etc.) and the combination / optimization of state-of-the-art technologies (i.e. high-bandwidth networs, peer-to-peer etc.)

- *Membership management* – set of services and tools for user management. Users' communities are modeled as Virtual Organizations. According to the membership and by the played role by a user it can be defined access control to data and processing services.
- *Processing services and hosting of user applications* – services to enable expert users exploiting computational and network resources together with Earth Science data in order to produce the final desired product.
- *Services Orchestration* – services supporting users in chaining, scheduling and executing the tasks that are involved in an experiment. The collection of tasks and their dependencies is the workflow. Individual tasks include searching and retrieving data from archives, mining these data, transforming / processing data objects by means of user defined algorithms or processing services available in the infrastructure.
- *Storage and sharing of user results* – a sort of private user's workspace allowing scientists to manage (search, store & publish) data generated in an experiment in a professionally maintained repository to give them greater visibility and accessibility over time. The researcher must feel she has complete control over when and how one of her experiment data products is made available to a broader community.

8.3 Uniform Resource Management URM GeoPortal

The URM GeoPortal [12] consists of a set of modules and services, which are able to communicate trough interoperable services (OGC, W3C). URM Geoportal is based on Open Source technologies, but it could be integrated with different technologies like MS SQL or ArcSDE. The GeoPortal system uses different OpenSource platforms (MapServer, GeoServer) and contains common visualization, data sharing, and metadata and catalogue functionalities. Additional parts of solution are also tools for management sensor observation and spatial data transformation and processing.

The internal repository uses File system (for shape file, dgn file and raster file like geotiff) and PostGIS as main data repository. It is also possible to connect to MS SQL 2008 spatial extension, MS SQL GeoMedia extension, ArcSDE. Metadata information could be stored in PostgreSQL, MySQL, MS SQL, and Oracle.

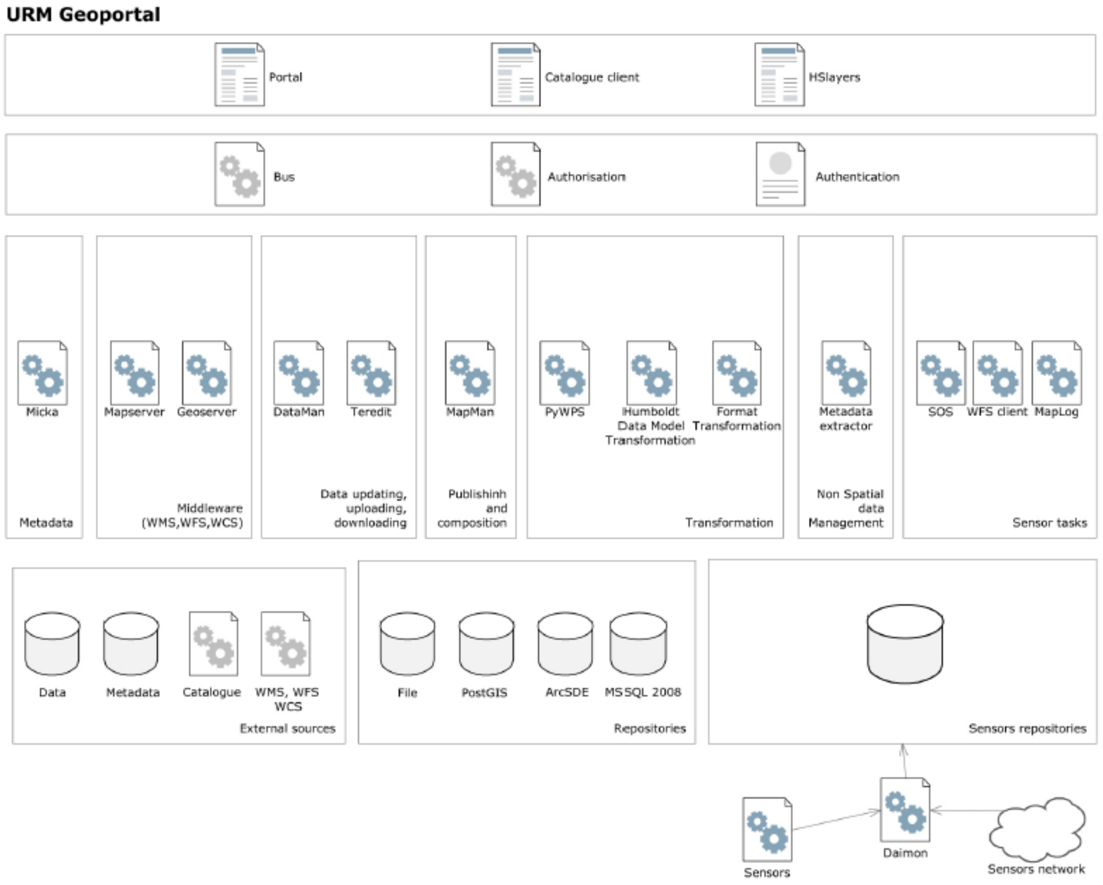


Figure 26 Uniform Resource Management URM GeoPortal

MIcKa is a complex solution for metadata management and for Spatial Data Infrastructure (SDI) and geoportal building. It contains tools for editing and management of metadata for spatial information, web services and other sources (documents, web sites, etc.). It includes their search on the Internet, portrayal in map or downloads to local computer. MIcKA is compatible with obligatory standards for European SDI building (e.g. INSPIRE). Therefore, it is ready to be connected with other nodes of prepared network of metadata catalogues. Metadata are stored in relational database and edited by dynamically generated forms.

The catalogue service is based on OpenGIS® Catalogue Services Specification – profile Catalogue Service for Web (CSW) and OpenGIS® Catalogue Services Specification 2.0.2 - ISO Metadata Application Profile standards. It supports basic operations (GetCapabilities, DescribeRecord, GetRecords, GetRecordById), editing, Inquiry items (according to standards), and extensions (export to GeoiRSS and KML).

For data management, which includes uploading, downloading and updating it uses two tools DataMan and Terredit. DataMan is an application for spatial data management. It supports data management in databases or files. In database, it is possible to store both, vector and raster data, including their attributes. Also for file oriented storage, it supports both, vector and raster data. From raster formats, it currently supports IFF/GeoTIFF, JPEG, GIF, PNG, BMP, ECW, from vector formats ESRI Shapefile, DGN, DWG, and GML. Teredit is a global component system where the components are interconnected via a multilevel architecture. Primary data collection provides a “Mobile GIS Editor” – it can be any available data collected software (e.g. ArcPad for PDA) which is up to standard of functionality and communication. The second side is the internal server of the company where the data will be stored. Teredit constitutes a bridge between these two sides.

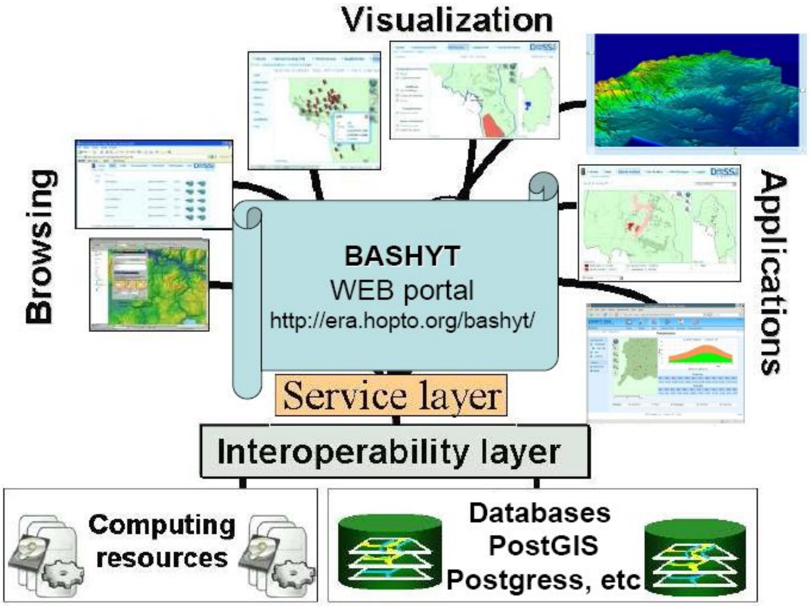


Figure 27 BASHYT Portal

The Map Project Manager (MapMan) is a software tool for users who want to publish local data or create new map projects and compositions from local data and external services. It supports publication of spatial composition from locally stored data (fields or database-stored in DataMan), with external WMS, WFS data services. It supports visualization in web browser using such clients like HSLayers, GoogleMaps, DHTML client, Desktop viewer GoogleEarth, GIS Janitor or publishes data as OGC WebMapService (WMS), OGC WebFeatureService (WFS). All published data are also connected with metadata stored in MIcKA.

8.4 BASHYT

BASHYT DSS (Basin Scale Hydrological Tool Decision Support System) [14][12] is an integrated web portal that exposes hydrological application, based on a watershed scale model, to support decision makers. BASHYT enables the storage, management and querying of data collections, visualization of data through the web GIS. In addition, the portal can create reports (graphs, maps, etc) through automatic standardized procedures. BASHYT has a layered structure (interoperability layer, service layer, application layer, etc.) and is built with open source technologies.

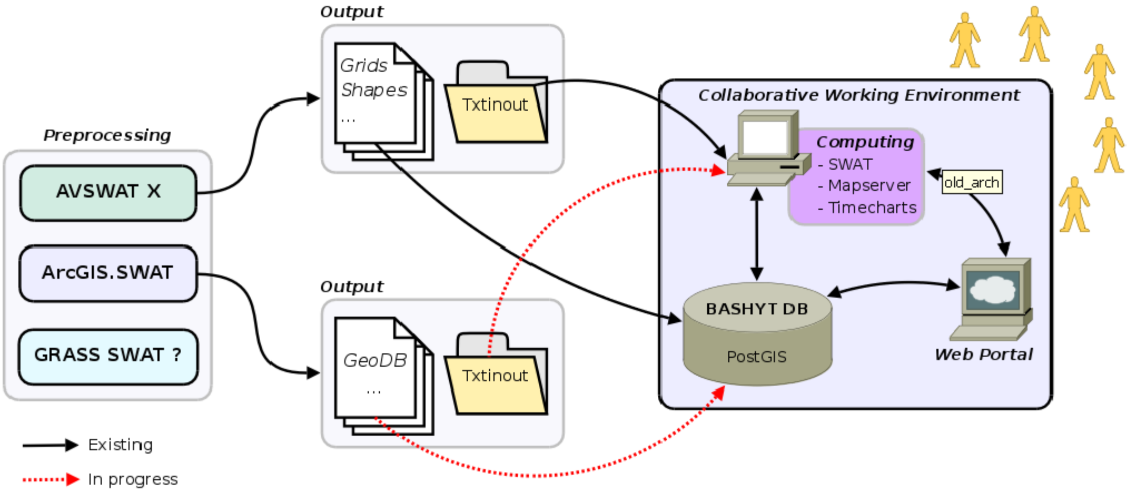


Figure 28 Collaborative Working Environment combines the BASHYT DSS and SWAP functionalities

- Service layer – it is the base system layer. Service layer offers access to different services: web server, application server, database server, GIS server, etc.
- Interoperability layer – contains functionalities and protocols for interoperability services
- Application layer – supports the user interface

The Portal offers a live programming environment and web template features, making the programming and application development available to beginner developers. The software could work with SWAT outputs and produces reports (GIS, tables, charts, contents etc) on the web in an easy fashion.

BASHYT is able to use output files of SWAP through Collaborative Working Environment (CWE), see Figure 28.

9 Conclusions and Recommendations

9.1 Conclusions

The following considerations are important for building the enviroGRIDS data storage and processing solution:

- Earth Science oriented applications require high computation resources to model large geographical areas, process huge distributed data, and provide high quality of user oriented services.
- Geospatial technologies (SDI) offer very specialized functionality for Earth Science oriented applications.
- Heterogeneous and distributed systems expose their functionality by services and use open standards and communication protocols.
- EnviroGRIDS will be a distributed system running on heterogeneous resources.
- Grid oriented technology is able to support distributed and parallel processing, management of heterogeneous resources, share distributed data and processing, and security in complex and collaborative works.
- SWAT model processing uses a great number of resources and data files that must be accessed in a secure and transparent manner for the user.
- SWAT execution over the Grid is a decentralized model. An efficient processing should be based on close locations of data and processing in order to reduce the cost and delay of data transport. This requires data management and replication policy over the Grid.
- Interoperability between geospatial and Grid infrastructures provides the features of both technologies:
  - Specific geospatial complex functionality and high power computation and security of the Grid.
  - Scalability. The grid infrastructure provides high processing and storage capabilities allowing to:
    - Improve output data resolution (e.g. from 100km to 2.5km).
    - Improve models complexity (e.g. collaborative, interactive models).
    - Widen the covered area (e.g. from local to national).
    - Improve the time-of-response (e.g. from hours to minutes).
    - Time-of-response is independent of resolution/coverage.
  - Flexibility and Interoperability. The geospatial services layer allow to:
    - Integrate and assimilate new and heterogeneous input data (e.g. different data formats and data sets).
    - Integrate outputs in a higher level application chain (e.g. SWAT and BASHYT)
    - Facilitate models interoperability and composition.
    - Be interoperable with other standard based infrastructures (e.g. INSPIRE, GEOSS).
- Interoperability is a great challenge. The OGC Web Services and the gLite middleware must be able to communicate and interact with each other.
- There are many organizations and working groups concerning with research and experiments on solutions for interoperation of Geospatial and Grid infrastructures.
- Interoperability attempt faces a set of difficult issues:

- Service composition:
    - Service description: Services in Grid infrastructure are described by WSDL while the OGC web services have their particular XML description.
    - Service interface: Grid services are invoked by SOAP while OGC web services use RPC (e.g. Get and Post commands by HTTP protocol) to send request and receive response.
    - State: Grid services are stateful while OGC services are stateless, excepting OGC WPS.
- Security:
    - OGC does not have embedded security. It has to be provided by other components and levels
    - Grid provides a strong embedded security based on the VO paradigm (by authentication, authorization, certificates)
    - On a Grid, it is essential to be identified and to have mechanisms of encryption, authentication and authorization. OGC does not have specifications to address these issues. Data security for OGC web services should be considered.
- Data accessibility
    - Semantics mismatch
        - Geo-information services handle coverage, observations, measurements, models, etc., at resource content/application level
        - Grids handle files, data, instruments, jobs, etc., at resource provision level
    - Geospatial data format: OGC WFS services use GML language to encode geospatial data.
    - Metadata: structure, content and related services are different for Geospatial and Grid infrastructures. OGC WCS publishes and searches collections of metadata about geospatial data, services and related resources.
    - Catalogue management:
        - It should preserve the OGC CSW interface while providing the naming and location transparency by mapping Grid information model (e.g. RLC) to OGC CSW information model. It should improve the metadata model for data types and service types and integrates metadata with CSW model to facilitate the service interoperability and chaining.
        - In the gLite middleware the files are identified by the user based on their logical file names (LFN), which are managed by the File Catalog interface. The LFN name is based on the VO specific namespace that maps a hierarchical directory structure. The File Catalogue service maintains mappings between LFN(s), GUID and SURL(s). gLite middleware uses the AMGA Metadata Catalogue as the metadata service.

9.2 Recommendations

According to the enviroGRIDS requirements and conclusions enumerated in the previous section, we express the following recommendations:

- Storage resources as well as distributed resources are managed through the secured mechanisms provided by Grid.

- Data storage solution is based on the gLite middleware. The gLite Data Management System provides a set of data services and tools supporting the storage, catalogs, and data transport.
- The Grid applications often require and generate a huge amount of data. This requires an efficient mechanism for data description and localization. One solution could be that of associating descriptive attributes to the files (e.g. semantic annotation).
- Grid security is provided to grant different access levels to different users. Storage Element is the Grid service responsible for saving/retrieving files to/from some data store that can provide a wide range of quality of service to the user. The interface for the management of disk and tape storage resources is the Storage Resource Manager (SRM).
- The File Catalogue service maintains mappings between LFN, GUID and SURL. LFC could either be used as a Local File Catalogue, holding only replicas stored at a given group of site, or a Global File Catalogue, containing information about all files in the Grid.
- EnviroGRIDS architecture layers three main components: Grid Portal, Geospatial infrastructure, and Grid infrastructure.
- EnviroGRIDS architecture consists of three main functional layers: data layer, Grid layer and service layer.
  - The data layer provides data repositories for storing data sets. The datasets consists of raw data (e.g. spatial data) and processed data (e.g. maps, tables). Metadata information are attached to datasets.
  - The Grid layer is provided by EGEE infrastructure and the gLite middleware.
  - The service layer provides services for data management, geospatial functionality, security and user management, scheduling, monitoring, SWAT related management and execution, workflow management, spatial data acquisition, and visualization.
- The enviroGRIDS Portal exposes to the user a set of tools and applications supported by the functionality and components provided through the services. There are available personalized tools for different category of users: system administrator, decision maker and citizen. The portal provides a particular set of tools for SWAT scenarios development, and data management.
- The enviroGRIDS portal should be the single way of the user to get into the system and the portal should face a unique style of the graphical user interface.

References

[1] Atkinson M., *Rationale for choosing the Open Grid Services Architecture*. In Berman F., Fox G., Hey T. (Eds), Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons, pp.199-215 (2003)

[2] Bacu V., Gorgan D., *Resource Measurements for Water Detection Algorithm in MedioGrid Architecture*. ISPDC 2007 - 6th International Symposium on Parallel and Distributed Computing, July 5-8, 2007, Hagenberg, Austria. IEEE Computer Press (2007), pp. 107-114.

[3] Bacu V., Muresan O., Gorgan, D.: *MODIS Image Based Computation of Vegetation Indices in MedioGRID* Architecture. SYNASC – GridCAD Workshop, 28 September, 2006, Timisoara, D. Zaharie et al. (eds) Procs. SYNASC06, IEEE Computer Press (2006), pp. 267-273.

[4] Berman F., Fox G., Hey T. (Eds), *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons (2003)

[5] Chen, A., di, L., Bai, Y.; Wei, Y., *Grid-enabled Web Services for Geospatial Interoperability*. Published by: American Geophysical Union, Fall Meeting 2007,

Available at http://adsabs.harvard.edu/abs/2006AGUSMIN34A..04C

[6] Cunha J.C., Rana O.F. (Eds), *Grid Computing: Software Environments and Tools*, Springer-Verlag (2006)

[7] Dennis Gannon, Randall Bramley, Geoffrey Fox, Shava Smallen, Al Rossi, Rachana Ananthakrishnan, Felipe Bertrand, Ken Chiu, Matt Farrellee, Madhu Govindaraju, Sriram Krishnan, Lavanya Ramakrishnan, Yogesh Simmhan, Alek Slominski, Yu Ma, Caroline Olariu, Nicolas Rey-Cenvaz, *Programming the Grid: Distributed Software Components, P2P and Grid Web Services for Scientific Applications*. Cluster Computing 5(3), pp.325-336 (2002)

[8] Faramarzi M., Abbaspour K.C., Schulin R., Yang H., *Modelling blue and green water resources availability in Iran. Hydrological Processes*, 23, 486–501 (2009). Published online 27 November 2008 in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/hyp.7160

[9] Foster I., Kesselman C., Nick J., Tuecke S., *The Physiology of the Grid: Open Grid services Architecture for Distributed Systems Integration*, Presented at GGF4, Feb. 2002, Available at http://www.globus.org/research/papers/ogsa.pdf (2002)

[10] Foster I., Kessselman C., Tuecke S., *The anatomy of the Grid: enabling scalable virtual organizations*. International Journal of Supercomputing Applications and High Performance Computing, 15(3), Also available at http://www.globus.org/research/papers/anatomy.pdf (2001)

[11] Gond V., Bartholomé E., Ouattara F., Nonguierma A. and Bado I., *Surveillance et cartographie des plans d'eau et des zones humides et inondables en régions arides avec l'instrument VEGETATION embarque sur Spot 4*, International Journal of Remote Sensing, 2004, 25,5. pp. 987-1004.

[12] Interoperability Guideline (enviroGRIDS_D2.1). EnviroGRIDS Project document, October 2009

[13] Kreger H., Estefan J., *Navigating the SOA Open Standards Landscape Around Architecture*

Available at http://www.opengroup.org/projects/soa/uploads/40/20044/W096.pdf

[14] Manca S., Soru C., Cau P., Meloni G., Fiori M., *A multi model and multiscale, GIS oriented Web framework based on the SWAT model to face issues of water and soil resource vulnerability*. Presentation at the 5th International SWAT Conference, August 3-7, 2009

http://ssl-video.tamu.edu/video/C2-Cau.pdf

[15] Mazzetti P., Nativi N., *WPS for Disaster Management CYCLOPS*. Presentation at the 4th GRID and e-Collaboration Workshop – Digital Repositories, ESA, Frascati, 25-26 Feb. (2009)

[16] Mazzetti P., Nativi S., *An interoperability architecture for geospatial resources*, Presentation at the EGEE 09 Conference, Barcelona - 24 September (2009)

[17] Muresan O., Pop F., Gorgan D., Cristea V., "Satellite image processing applications in MedioGRID", in Proceedings of ISPDC06, IEEE Computer Press, (2006), pp 253-260.

[18] Nativi S., *Implementing OGC Web Services on gLite: the G-OWS initiative*. Presentation at the workshop Earth Science community on its way to the Earth Science SSC. EGEE'09 Conference, Barcelona, 21-25 Sept. (2009)

[19] Li M., Baker M., *A Review of Grid Portal Technology*. Published in *Computing: Software Environments and Tools,* Cunha J.C., Rana O.F. (Eds), Springer-Verlag, pp. 126-156 (2006)

[20] Silva V., *Grid Computing for Develpers*, Published by Charles River Media (2006)

[21] Valentine, D. W., Jennings, B., Zaslavsky, I., Maidment, D. R., *Unifying Access to National Hydrologic Data Repositories via Web Services*. Publication: American Geophysical Union, Fall Meeting 2006, Available at http://adsabs.harvard.edu/abs/2006AGUFMIN53C..03V

[22] Venugopal R., Buyya R., Ramamohanarao K., *A Taxonomy of Data Grids for Distributed Data Sharing, Management,and Processing*, ACM Computing Surveys, Volume 38 , Issue 1, ISSN:0360-0300

[23] Veronesi P., *Data Management:File Catalog, Data Movement High level tool*, Presentation at the II Scuola per utenti INFN della Grid, Bologna October 6-9th, 2008

[24] Zappi R., *Data Management Concepts and Storage Services, GridFTP and SRM services*, Presentation at the II Scuola per utenti INFN della Grid, Bologna October 6-9th, 2008

[25] AMGA Metadata Catalogue
http://amga.web.cern.ch/amga/index.html

[26] Architectural Design (GENESI-DR-SA2-DEL-DSA2.2 v1.0)
http://www.genesi-dr.eu/documents/

[27] Condor Project
http://www.cs.wisc.edu/condor

[28] CYCLOPS Project Web Site
http://www.cyclops-project.eu

[29] DCLite4G, Dublin Core Lightweight Profile for Geospatial
http://wiki.osgeo.org/wiki/DCLite4G

[30] FGDC - Federal Geographic Data Committee
http://www.fgdc.gov/metadata/csdgm/

[31] FGDC Geospatial Applications and Interoperability WorkingGroup
http://www.fgdc.gov/participation/working-groups-subcommittees/gaiwg

[32] FTS, File Transfer Service
http://www.gridpp.ac.uk/wiki/FTS

[33] gLite, Lightweight Middleware for Grid Computing
http://glite.web.cern.ch/glite/

[34] gLite Security
https://edms.cern.ch/file/487004/1.1/EGEE-JRA3-TEC-487004-DJRA3.1-1.1.pdf

[35] gLite Storage Element
https://edms.cern.ch/document/594698/

[36] gLite WMS - Workload Management System
http://web.infn.it/gLiteWMS/

[37] Ganga
http://ganga.web.cern.ch/ganga/

[38] GENESI-DR, Ground European Network for Earth Science Interoperations – Digital Repository
http://www.genesi-dr.eu

[39] GEOSS, Global Earth Observation System of Systems
http://www.epa.gov/geoss/

[40] Globus Project
http://www.globus.org/

[41] GML 3.1.1 Application schema for Earth Observation products OGC 06-080r4
https://portal.opengeospatial.org/files/?artifact_id=31065

[42] G-OWS (gLite-OWS) Working Group
https://www.g-ows.org/pmwiki.php

[43] G-OWS Working Group
http://sites.google.com/a/imaa.cnr.it/g-ows/Home

[44] GridFTP
http://www.globus.org/toolkit/docs/4.0/data/gridftp/

[45] GSI, Grid Security Infrastructure
http://www.globus.org/toolkit/docs/4.0/security/

[46] INSPIRE
http://inspire.jrc.it/

[47] ISO, International Organisation for Standardization
http://www.iso.org

[48] ISO TC-211 Glossary
http://www.isotc211.org/Terminology.htm

[49] NASA - Earth Science Data Terminology and Formats
http://nasascience.nasa.gov/earth-science/earth-science-data-centers/earth-science-data-terminology-and-formats

[50] OASIS SAML/XACML Glossary
http://www.oasis-open.org/committees/security/docs/draft-sstc-glossary-02.pdf

[51] OGC Simple Features Specification for SQL
http://www.opengeospatial.org/standards/sfs

[52] OGC CSW, Catalogue Service for Web
http://www.opengeospatial.org/standards/cat

[53] OGC Glossary
http://www.opengeospatial.org/ogc/glossary

[54] OGC Interoperability Institute
http://www.ogcii.org/

[55] OGC, Open Geospatial Consortium
http://www.opengeospatial.org/

[56] OGF, Open Grid Forum
http://www.ogf.org

[57] OGC WCS, Web Coverage Service
http://www.opengeospatial.org/standards/wcs

[58] OGC WFS, Web Feature Service
http://www.opengeospatial.org/standards/wfs

[59] OGC WMS, Web Map Service
http://www.opengeospatial.org/standards/wms

[60] OGC WPS, Web Processing Service
http://www.opengeospatial.org/standards/wps

[61] OGSA-DAI, The Open Grid Services Architecture – Database Access and Integration
http://www.ogsadai.org.uk/

[62] OGSA Architecture - The Open Grid Services Architecture, Version 1.5
http://www.ogf.org/documents/GFD.80.pdf

[63] OGC – Open Geospatial Consortium Web Services
http://www.opengeospatial.org/projects/initiatives/ows-6

[64] OpenSearch
http://www.opensearch.org

[65] RDF, Resource Description Framework
http://www.w3.org/RDF

[66] Soil and Water Assessment Tool, Official SWAT Web site
http://www.brc.tamus.edu/swat/index.html

[67] SOAP, Simple Object Access Protocol specification
http://www.w3.org/TR/soap

[68] SWAT Input files
http://www.brc.tamus.edu/swat/downloads/doc/swat2005/SWAT%202005%20io.pdf

[69] UDDI, OASIS – UDDI, Universal Description, Discovery and Integration
http://uddi.org/pubs/uddi_v3.htm

[70] VOMS, Virtual Organization Membership Service
http://vdt.cs.wisc.edu/components/voms.html

[71] Wikipedia encyclopedia
http://en.wikipedia.org/

[72] WSDL, Web Service Description Language
http://www.w3.org/TR/wsdl

Terminology

The terms and definitions used in this document are according with terminologies published in known glossaries such as ISO TC-211 Glossary [48], OGC Glossary [53], OASIS SAML/XACML Glossary [50], and Wikipedia encyclopedia [71]. We have added new terms and definitions when it was necessary.

Definitions

Access Control: A process by which use of resources is regulated according to a security policy and is permitted by only authorized system entities according to that policy.

Access Rights: A description of the type of authorized interactions a user or a program can have with a system. Examples include read, write, execute, add, modify, and delete operations.

Actor: A coherent set of roles that user of system plays when interacting with this system.

Attribute: Feature that describes the range of values an element may hold.

Authentication: To confirm a system entity’s asserted principal identity with a specified, or understood, level of confidence.

Authorization: The process of determining, by evaluating applicable access control information, whether a subject is allowed to have the specified types of access to a particular resource. Usually, authorization is in the context of authentication. Once a subject is authenticated, it may be authorized to perform different types of access.

Catalogue: A collection of entries, each of which describes and points to a feature collection. Catalogues include indexed listings of feature collections, their contents, their coverage, and other metadata. It registers the existence, location, and description of feature collections held by Information Community. Catalogues provide the capability to add and delete entries. At a minimum Catalogue will include the name for the feature collection and the location handle that specifies where this data may be found.

Clearinghouse: It provides a central access point for value-added topical guides which identify, describe and evaluate Internet-based information resources. In our system a clearinghouse contains metadata that supports a unique presentation and query of the partner data repositories. The clearinghouse provides information on each application what data could access.

Computational viewpoint: Viewpoint in RM-ODP that is concerned with the functional decomposition of the system and its environment into a set of services that interact at interfaces. This viewpoint captures the details of components and interfaces without regard to distribution.

Data clearinghouse: Collection of institutions providing digital data, which can be searched through a single interface using a common metadata standard.

Data product: Dataset or Dataset series that conforms to a data product specification. Data product specification is a detailed description of a dataset or dataset series together with additional information that will enable it to be created, supplied to and used by another party.

Dataset: A Dataset is an abstract object. It corresponds to the ideal of a data set, independent of a physical form or an encoding in which it is being distributed. It represents an identifiable collection of data.

Dataset series: Collection of datasets sharing the same product specification.

Digital Repository: A complex collection of datasets. The main functionalities concern with data access and related services provided to users. Data repository could support the following functionalities on data: organize, store, administrate, secure access and retrieve, search and discover.

Engineering viewpoint: RM-ODP viewpoint on a system concerns mainly with the interaction between computational components supporting communication, software processing, and clustering, without detailed knowledge of the components and computational infrastructure.

Enterprise viewpoint: RM-ODP viewpoint on a system describing the business or organizational perspective, context, purpose, scope and governing policies.

Feature: Abstraction of a real world object and phenomenon. In data repository a feature has a digital representation.

Feature collection: A group of features having common metadata and formal relationships. The feature collection can be identified as another feature at different abstraction levels.

Implementation view: Part of Information viewpoint that specifies how information must be represented within a working enterprise in different phases (e.g. access, processing).

Information viewpoint: RM-ODP viewpoint concerns with the meaning and relationships of data in different phases, operations, and components.

Grid computing: A service for sharing computer power and data storage capacity over the Internet (by CERN). The ability, using a set of open standards and protocols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet (by IBM).

Grid Unique Identifier (GUID): A non-human-readable unique identifier for a data item.

Infrastructure for spatial information: Metadata, spatial data sets and spatial data services; network services and technologies; agreements on sharing, access and use; and procedures, established, operated or made available in accordance with particular specifications.

Interoperability: Possibility for spatial data sets to be combined, and for services to interact, without repetitive manual intervention, in such a way that the result is coherent and the added value of the data sets and services is enhanced.

ISO 19115: most widely used international standard ISO for describing geographic information and services. It provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data.

Logical File Name (LFN): An alias used to refer to some data item. It is created by a user.

Metadata: Data describing another data or service. In the web service based system, the metadata is XML-encoded and stored in catalogues and registries, in order to support operations such as search, discover, and retrieve data and services.

Metadata dataset: The set of metadata describing a specific dataset.

Middleware: Software in a distributed computing environment that mediates between clients and servers.

Point-in-time recovery: in the context of computers is a system whereby a set of data or a particular setting can be restored or recovered from a time in the past.

Policy: A set of rules, sometimes described through an algorithm as an obligation.

Portal: A Web site that provides a view into a universe of content and activity through a variety of links to other sites, communication and collaboration tools, and special features geared toward the community served by the portal.

Portlet: From user's perspective, a portlet is a window in a portal that provides a specific service or information, for instance, a calendar or weather. From the application development perspective, the portlets are pluggable modules that are designed to run inside the portlet container of a portal server. The portlet container provides a runtime environment in which portlets are instantiated, used, and finally destroyed [7]. Portlets rely on the overall portal infrastructure to access user profile information, participate in window and action events, and communicate with other portlets, access remote content, and lookup credentials, and store persistent data.

Portrayal: The presentation of information to humans, e.g., a map. In the context of the Web, portrayal refers to how data is presented to the user.

Protocol: A set of semantic and syntactic rules that determine the behavior of entities that interact.

Schema: Formal description of a model or a structured framework. A metadata schema specifies the order and types and labels of information elements describing a data set.

Server: A particular instance of a service. The server denotes sometimes the computer running a service.

Service: A computation performed by a software entity on one side of an interface in response to a request made by a software entity on the other side of the interface. It is a collection of operations that allows a user to evoke a behavior or a value.

Service interface: Shared boundary between an automated system or human being and another automated system or human being.

Service metadata: Metadata describing the operations and (geographic) information available at a server.

Site URL (SURL): The location of an actual piece of data on a storage system.

Spatial data: Any data with a direct or indirect reference to a specific location or geographical area.

Spatial data services: Operations which may be performed, by invoking a computer application, on the spatial data contained in spatial data sets or on the related metadata.

Spatial data set: An identifiable collection of spatial data.

Spatial object: An abstract representation of a real world phenomenon related to a specific location or geographical area.

Spatial metadata: Information describing spatial data sets and spatial data services and making it possible to discover, inventory and use them.

Storage resource: A combination of the storage hardware and the controlling software (e.g. file system).

Storage resource manager (SRM) services: Middleware components that provides dynamic space allocation and file management. SRM services implement SRM interface.

Storage element (SE): A Grid service that allows Grid users to store and manage files together with the assigned space.

Storage area: Defines a portion of the total available space that can span different kinds of storage devices within a Storage Element.

Storage component: Identifies a specific storage with certain quality properties.

Technological viewpoint: Viewpoint on an ODP system and its environment that focuses on the choice of technology in that system. RM-ODP viewpoint concerned with the underlying infrastructure in a distributed system. It describes the hardware and software components used in a distributed system.

Transport URL (TURL): Temporary locator of a replica plus an access protocol understood by a Storage Element.

Use case scenario: A possible sequence of real world events used as a test case for specifying or testing information systems designed to help manage such events.

Universally Unique Identifier (UUID): An identifier standard used in software construction, standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE). The intent of UUIDs is to enable distributed systems to uniquely identify information without significant central coordination.

Virtual Organization: In grid computing, a Virtual Organization (VO) is a group who shares the same computing resources (e.g. data, processors, software components, hardware infrastructure).

Web Service: A Web Service is defined by the W3C as a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine based processing format (e.g. WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

X.509: ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) standard for a public key infrastructure (PKI) for single sign-on (SSO) and Privilege Management Infrastructure (PMI). X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

Abbreviations and Acronyms

API	Application Programming Interface
BES	Basic Execution Service
CA	Certification Authority
CE	Computing Element
CERN	European Organization for Nuclear Research
CWE	Collaborative Working Environment
DBMS	Data Base Management System
DCE	Distributed Computing Environment
DR	Digital Repository
CSW	Catalogue Service for Web (OGC)
EGEE	Enabling Grids for E-sciencE
ESA	European Space Agency
GENESI-DR	Ground European Network for Earth Science Interoperations – Digital Repositories
GEO	Group on Earth Observation
GEOSS	Global Earth Observation System of Systems
GIS	Geographic Information System
gLite WMS	Workload Management System

GT	Globus Toolkit
GUID	Globally Unique Identifier
HTTP	Hypertext Transfer Protocol
INSPIRE	Infrastructure for Spatial Information in Europe
ISO	International Organization for Standardization
LFC	LCG File Catalogue, LHC Computing Grid File Catalogue, Large Hadron Collider Computing Grid File Catalogue
LFN	Logical File Name
OAIS	Open Archival Information System
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OGF	Open Grid Forum
OGSA	Open Grid Service Architecture
OSF	Open Software Foundation
OWS	OGC Web Service
PFN	Physical File Name
RB	Resource Broker
RDF	Resource Description Framework
RM-ODP	Reference Model for Open Distributed Processing (ISO/IEC 10746)
SDI	Spatial Data Infrastructure
SE	Storage Element
SEE-GRID-SCI	South Eastern Europe-GRID eInfrastructure for regional eScience
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SURL	Storage URL
SWAT	Soil and Water Assessment Tool
TC-211	Technical Committee within ISO for Geographic Information/ Geomatics standards
TURL	Transport URL
UDDI	Universal Description Discovery and Integration
UNIGE	University of Geneva
URL	Uniform Resource Locator
URM	Uniform Resource Management
UUID	Universally Unique Identifier
VO	Virtual Organization
VOMS	Virtual Organization Membership Service
W3C	World Wide Web Consortium
WCS	Web Coverage Service implementation specification produced by the OGC

WFS	Web Feature Service implementation specification produced by the OGC
WM	Workload Manager
WMS	Web Map Service implementation specification produced by the OGC
WPS	Web Processing Service implementation specification produced by the OGC
WS	Web Service
WSDL	Web Services Description Language
WSRF	Web Services Resource Framework
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language