SIMDAT

Data Grids for Process and Product Development using Numerical Simulation and Knowledge Discovery

Project no.: 511438

Grid-based Systems for solving complex problems – IST Call 2

Integrated project

SIMDAT

D.2.1.2 First integrated Grid infrastructure

Start date of project: 1 September 2004 Duration: 48 months

Due date of deliverable: 2005-09-30

Actual submission date: 2005-11-07

Lead contractor for this deliverable: IT Innovation

Revision: 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participant (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Copyright

Table of contents

Executive Summary

This report is a public version of the document of record for the first integrated Grid infrastructure software delivered to SIMDAT partners on 3rd June 2005, as specified in SIMDAT Annex 1- “Description of Work” [1]. The selection of Grid infrastructure components was made by application sectors based on evaluating publicly available middleware components. A variety of technologies were selected with aerospace and automotive sectors deploying GRIA, pharma deploying web services and E2E security from GEMMS and meteo deploying J2EE, web services, AuthZ and AuthN. The first Grid infrastructure software release was based on GRIA 4.1.0, which provided various functional and usability enhancements to satisfy initial SIMDAT requirements. Feedback from PM12 prototype evaluation will be factored into the Grid infrastructure roadmap, which will define how Grid infrastructure components developed in Pharma, Meteo and the wider community can be incorporated into future releases of Grid infrastructure software either by integration or by demonstrating interoperability using appropriate standards.

1 Introduction

This report is a public version of a ***document of record for the first integrated Grid infrastructure software*** delivered to SIMDAT partners on 3rd June 2005, as specified in SIMDAT Annex 1- "Description of Work" [1]. The document describes how the application activities selected Grid infrastructure components in Section 2 and then describes the main software release from WP2 in Section 3. Section 4 describes how Grid infrastructure has been deployed during the PM12 prototypes and Section 5 provides some initial conclusions and next steps for the Grid infrastructure software.

The results of PM12 prototype evaluation and a review of the wider Grid infrastructure landscape will be documented D.2.1.3: Report on SIMDAT Integrated Grid Infrastructure to be delivered at PM16.

2 Selection of Grid Infrastructure components

The selection of Grid infrastructure components was made by application sectors following a detailed requirements capture phase and review of the state-of-the-art in Grid technologies provided by WP2 [2]. The selection was based on evaluating and selecting publicly available middleware components to ensure that the development of the PM12 prototypes could begin quickly. The state-of-the-art presented various Grid infrastructure technologies some of which were available immediately (GRIA, OMII, UNICORE) and some with imminent release schedules (gLite, GT4). The emphasis during the analysis was to allow the application activities to come to their own conclusions and select Grid technology that meets the needs of their application domain rather than prescribing a single Grid infrastructure.

In the end, the application sectors selected a variety of technologies to support their PM12 prototypes. The aerospace and automotive sectors chose GRIA for the PM12 prototypes due to the GRIA's explicit focus on industrial Grid deployments and immediate availability. The automotive sector intends to review the state-of-the-art after PM12 following the release and stabilisation of gLite and GT4. The meteo sector concluded that Grid technology was still emerging and that the state-of-the-art in Feb-2005 would not meet their data grid requirements. However, OGSA-DAI and gLite had potential and would be re-evaluated after PM12. The pharma sector decided to develop a web services Grid leveraging E2E security components developed during the GEMSS project.

The following sections describe the first integrated Grid infrastructure delivered by WP2. The release was based on GRIA 4.1.0, which provided various functional and usability enhancements to satisfy SIMDAT requirements. Future releases of Grid infrastructure software will begin to address interoperability challenges between middleware components deployed in each sector. We will begin to see a move away from monolithic Grid infrastructures to a set of services that can work together in a coherent way to delivering the needs of the application activities.

3 First Integrated Grid Infrastructure Software Description

GRIA provides a virtual organisation (VO) model based on business-to-business service provision, supporting business models and processes that allow collaborative resource sharing for industry. GRIA uses process based access control (PBAC) and contextualised web services to constrain how resources are assigned and in which context they can be accessed. The existing GRIA business process defines four resource types: account, resource allocation, job and data tasks along with the interactions between the client and service provider.

A business relationship between a client and service provider is represented as an account. When a client requests an account the service provider checks the client's credit-worthiness and assigns the client an account with a certain credit limit. Typically, a business will have a one or more accounts with a given service provider representing distinct financial budgets i.e. project or organisation budgets. Accounts are requested by users within a client organisation who are authorised to spend the budget. Once the account is open, the client may apply for any number of resource allocations. This is done through a tendering process whereby the client submits a request detailing resource requirements (e.g. data transfer quantities and CPU time requirements) and the service provider responds with an offer of an allocation and a price, which can be accepted or rejected by the client. A client buys resource allocations for use within a project or department.

Within a resource allocation, data stagers may be set up and jobs may be executed. A data stager may be thought of as a container that can hold only one file. The transfer of data into and out of a data stager is metered and billed back to the account via the resource allocation. The client controls read and write access to a data stager. A job is also associated with a resource allocation. When a job is started, the job service is informed which data stager(s) to download input data from and which data stager(s) to upload output data to. In order for the job service to access these data stagers, access rights to the data stagers must have previously been granted to the job service by the client.

GRIA uses hierarchical contexts to represent business processes. At each level, there is a well-defined subject and conversational history that the context refers to. This approach is strongly based on existing business processing mechanisms. For example, a B2B relationship is likely to involve a client opening an account with a service provider organization. The account is the subject of a conversation at this level, and the account number is the service provider's identifier for that particular relationship – in other words a context identifier. Within the context of this account, orders are raised by the client, and within an order, there may be many items, each of which may be delivered and invoiced separately. Thus there is a hierarchy of contexts, allowing each individual document to be traced back to the top-level account. To represent this in GRIA, we use three nested levels of conversations,

- A conversation referring to the account opened by a Budget Holder client with a service provider;
- A conversation referring to a quality of service agreement between a User client and service provider; and
- Conversations referring to individual jobs and data stores that use resources allocated under a quality of service agreement.

This forms a tree-like structure representing the relationship between a client and service provider. At any instant, the contextual hierarchy captures the status of the client-server relationship (e.g. outstanding jobs or total amount owed).

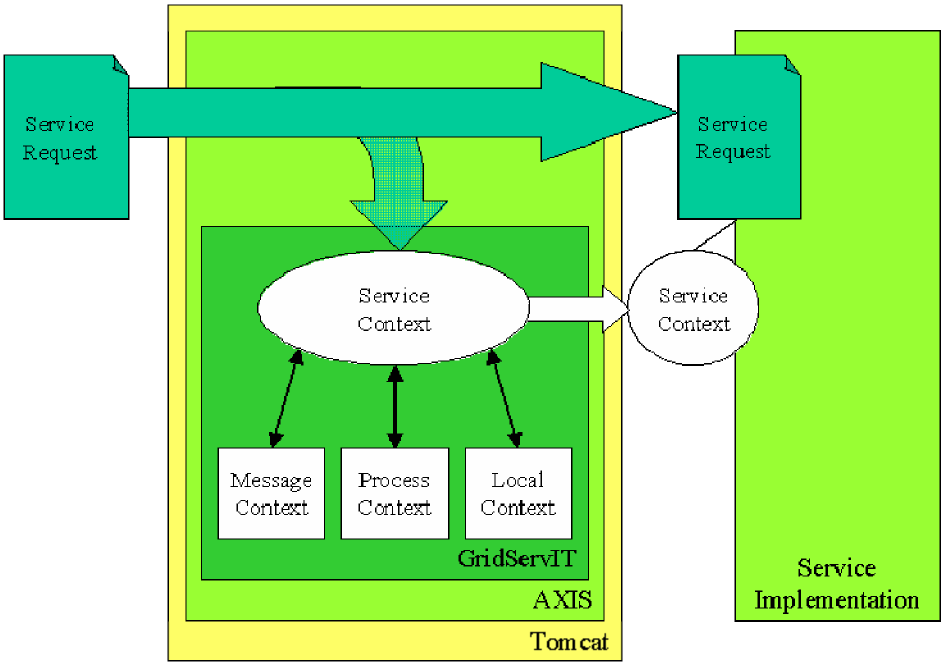

Figure 1: GRIA support for service context

The GRIA service container GridServIT provides context management for service implementations. When a request is received, GridServIT builds a service context and makes this available to the service implementation for authorisation decisions and business logic. The service context itself is created from message, process and local context as shown in Figure 1. The message context contains information relating to the specific request such as WS-Security headers. The process context represents the state of the conversation between the client and service provider on which the current request is operating. The local context represents the environment that the service is deployed in such as the Tomcat service container and local machine.

GRIA uses mature, off-the-shelf software components as its application container and web server such as the Apache HTTP Server, Tomcat and Axis (See Figure 2). GRIA advocates a multiple "rings of defence" security strategy by adopting solutions from the world of web-based e-Commerce systems, and working within, rather than around the operational security best practices developed there. For network security GRIA recommends that services and associated resource clusters be placed in a De-Militarised Zone (DMZ), which is conventional best practice for most web-based business systems. GRIA services are designed so that each SOAP invocation is reasonably short-lived, relieving problems with TCP connection time-outs. There are also no "call-backs" from the service to the client (e.g. for notification). These features mean that GRIA works with conventional HTTPS proxies and firewall configurations. There is no need for the system administrator to open any non-standard tunnels in the firewall, and client-side users continue to benefit from its protection. GRIA also uses both transport- and message-level security. At the transport level, GRIA (through Apache) enforces mutual authentication of HTTPS connections. This ensures that an unauthenticated client cannot access a GRIA service as the HTTPS connection will be dropped during the SSL handshake, before any data reaches the Web Service message processor. When the HTTPS connection is established with an authenticated user, message-level security is used to enable separate authentication (and non-repudiation) of message content using WS-Security standards. The authenticated user along with the service operation is then used as the basis for business process enforcement using Process Based Access Control component.

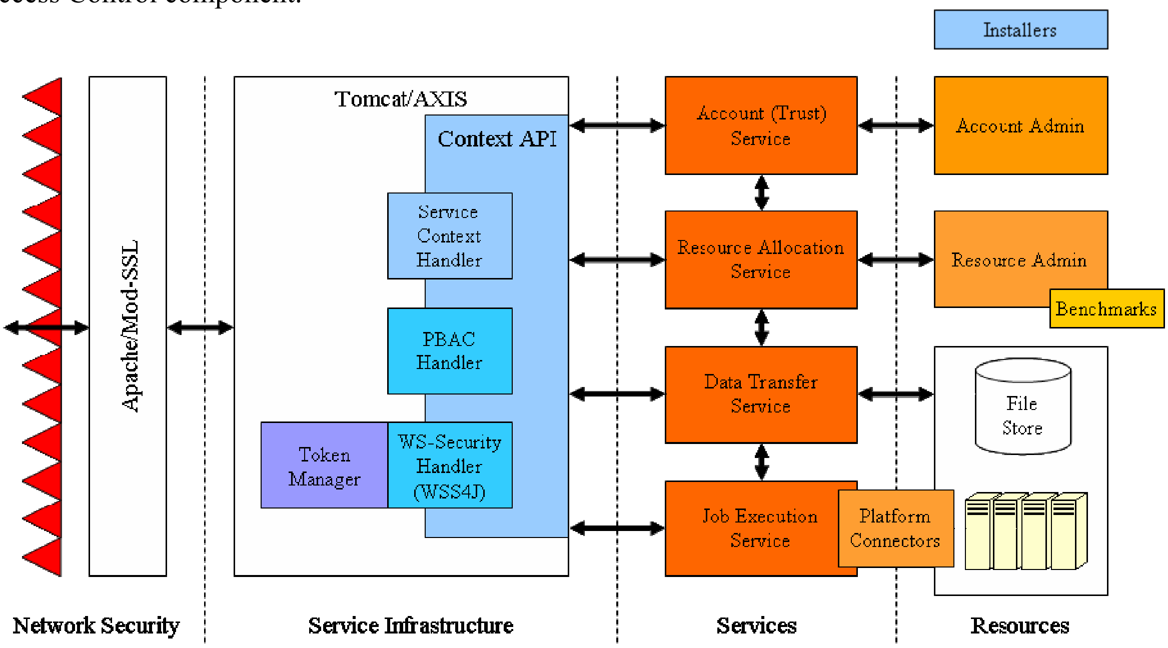

Figure 2: GRIA service provider

Higher-level services that are compliant to WS-I 1.0 Basic Profile, and use WS-I 1.0 Basic Security Profile build on top of this security and authorisation core. GRIA provides four services:

- **Account Service**. This supports the creation and management of accounts, each representing a trust relationship between the service provider and a user (the account holder), who is responsible for all service access on their account. The account holder can then allow others to access services, and monitor usage on the account.

- **Resource Allocation Service**. This allows remote users to request and be granted (or denied) allocations of computation and data storage capacity at the service provider site. The service incorporates a resource capacity model, which is used to determine whether capacity is available to meet a requested allocation.
- **Data Storage Service**. This allows remote users to upload and download data files to the service provider. The Data Service also supports delegation of access rights (for read or read-write access) to other users or service providers.
- **Job Execution Service**. This allows remote users to start, monitor or kill computational jobs, executed by the service provider. The Job Service can be configured to support multiple applications (mapped to different web service endpoints), fetching input from and writing output to data storage services.

The GRIA job service supports applications that run in batch mode and executed either locally or via a resource manger environment. Applications can be executed on a variety of underlying computing platforms for job execution using Platform Connector scripts e.g. from single computers to clusters of workstations or even supercomputers (See Figure 3). The Platform scripts decouple the job service and application bindings from resource managers ensuring that various execution platforms can be used without changing the job service itself or the applications.

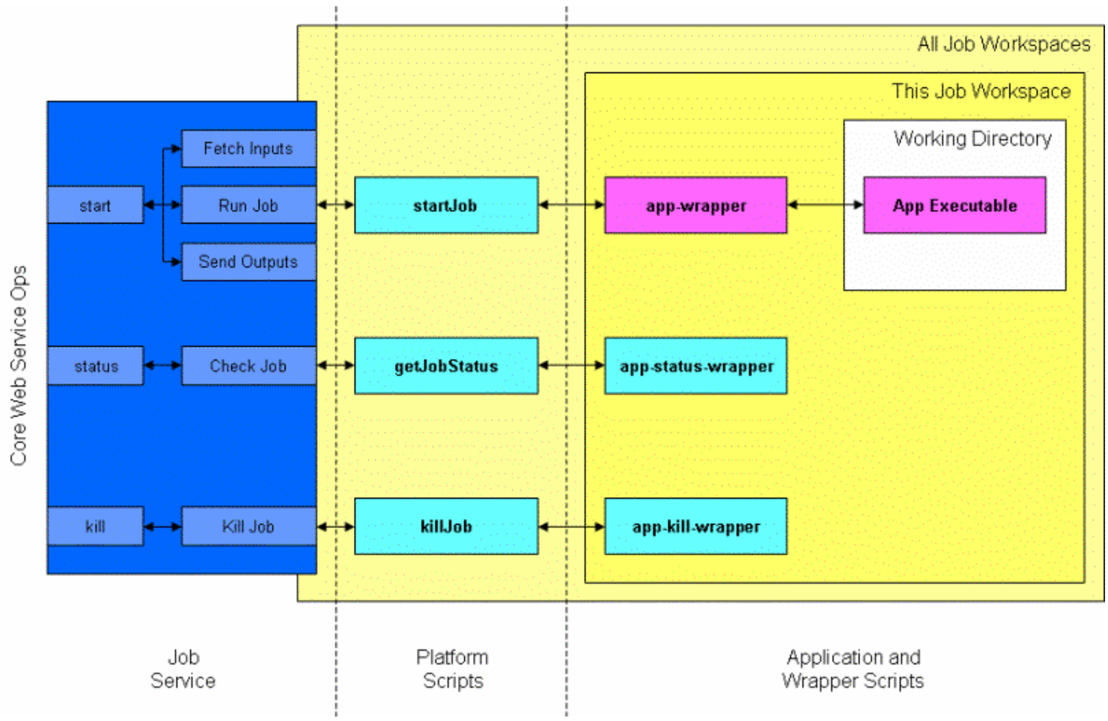

Figure 3: Execution platforms and applications

The GRIA job service can manage jobs via three web service operations, i.e.

- Start the job;
- Monitor the status of the job; and
- Terminate a job.

For each of these operations the job service invokes the corresponding Platform Script with appropriate arguments to provide the required functionality. GRIA is pre-supplied with platform scripts for local execution (default) and working template scripts for PBS, and Condor job managers. GRIA service providers are expected to provide their own Platform scripts for their preferred resource manger. Platform scripts can have strong dependences on both the resource manager and the underlying OS. GRIA does not impose any restrictions on the implementation of these Platform scripts, although in practice scripting languages can provide the required functionality, e.g. handling processes, etc.

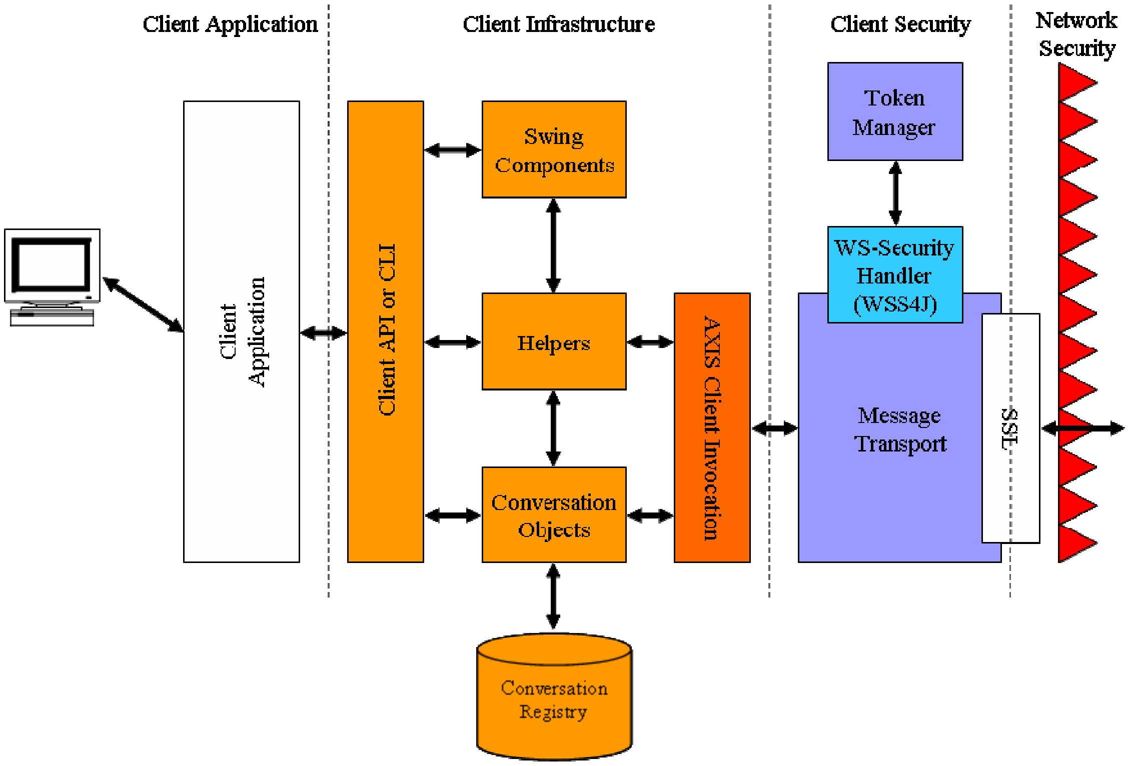

Figure 4: GRIA client API

On the client side, a toolkit and Java API is provided, which can be used to simplify the programming of applications that use GRIA Services (See Figure 4). The client API uses the Axis invocation framework along with helper classes to manage conversations with GRIA service providers. Application developers can either use the helper classes or access the services directly using a WS-I profiles compliant web services invocation framework.

3.1 GRIA Enhancements for SIMDAT

GRIA was initially developed in the EC IST GRIA project which ended Oct-2004. The final software delivered by the project was functionally good but robustness, usability and maintainability needed improvement to support commercial Grid deployment in SIMDAT. Also, additional functional enhancements were identified following an analysis of both the application and higher-level technology work package requirements.

GRIA 4.1.0 provided extensive usability enhancements to help users install and administer the software. Previous versions of GRIA typically took a few days to install and required in-depth knowledge about target platforms and the GRIA software itself. The new release makes the installation much simpler by utilizing Tomcat's web application archive (WAR) deployment facility. Users can now upload a GRIA WAR file directly to a Tomcat server and provide configuration information using the GRIA service provider administration portal (See Figure 5). The administration portal is very intuitive and leads the user through all aspects of the installation. If problems exist with the configuration the user is notified directly. The administration portal also supports upgrade from previous GRIA installations whilst maintaining previous configuration options.

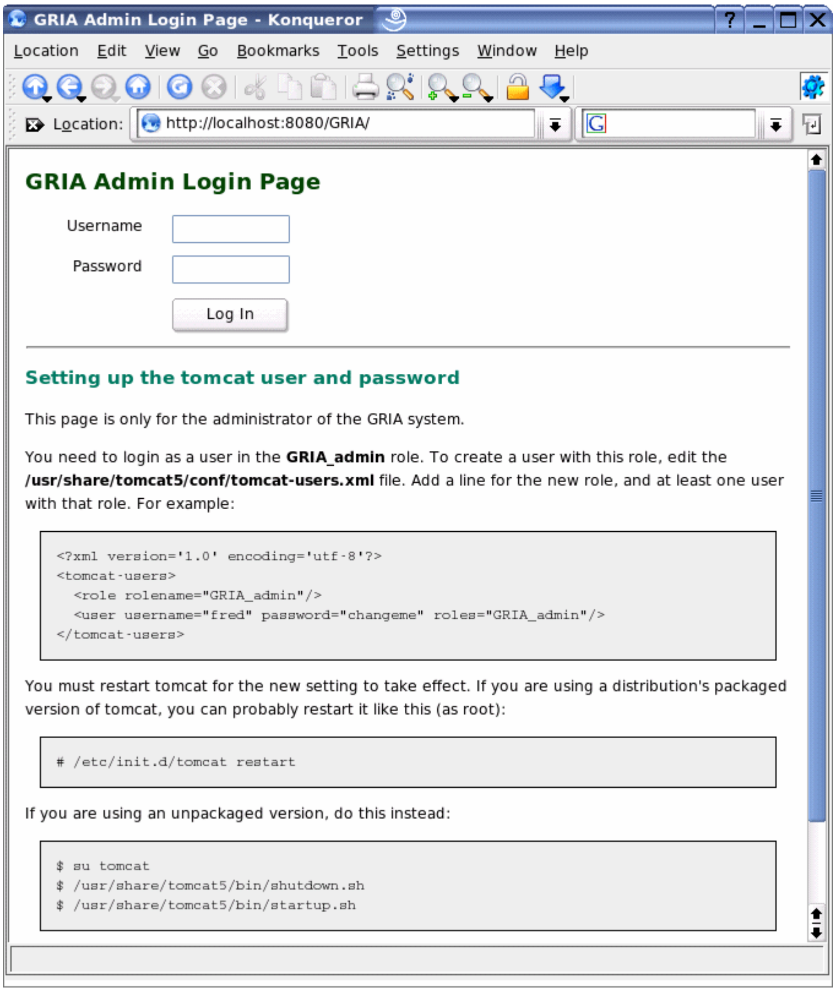

Figure 5: GRIA administration tool

The application requirements showed that SIMDAT users need Grid infrastructure software to be portable between many operating systems and hardware platforms. GRIA is written in Java and is therefore inherently portable to most operating systems. However, providing deployment documentation and testing on every operating system/Linux distribution would be too costly. Through discussions with the application sectors, target operating systems were identified that would be supported for deployment in SIMDAT. GRIA was originally documented and tested for the SuSE Linux distribution. This was extended to support Fedora Core 3. Fedora was selected due to its close relationship with the Linux Ret Hat distribution required by the automotive sector. Fedora is more upstream than Red Hat with more frequent release intervals and newer features where more inline with a Grid RTD project.

The application activities use various queuing systems to manage access to computational clusters. As described in Section 3, GRIA integrates these queuing systems using Platform Connector scripts. In the aerospace and automotive sector both Condor and PBS were already deployed. Additional Platform Connector scripts where developed to support both systems.

A major objective of the PM12 prototypes in the aerospace and automotive sector was integration between Grid infrastructure and workflow technologies Taverna/Freefluo, Inforsense and Optimus. All of the technologies provide a composition tool that allows users to compose workflows from a set of available services/tasks. GRIA was extended to provide a basic service registry of applications deployed at a service provider. The registry allows service providers to publish application metadata and clients such as workflow technologies to discover application metadata for use during workflow composition.

4 Grid Deployments for PM12 Prototypes

This section provides a brief summary of the Grid infrastructure deployments achieved for the PM12 prototypes. A more detailed description can be found in the associated deliverables [3, 4, 5, 6, 7].

4.1 Aerospace

GRIA has been deployed as the Grid infrastructure to support the aerospace application scenario. GRIA's capability to support dynamic virtual organisations allows a project manager to create a multidisciplinary design team where distributed engineers working in different organisations can participate in the design and development of a low-noise/high-lift landing system. Figure 6 shows the overall architecture for the aerospace scenario. Each organisation within the aerospace virtual organisation operates as a GRIA service provider offering engineering services that form part of the overall aerospace application including an initial optimisation function (UoS), aerodynamics (BAE) and aero-acoustics (EADS).

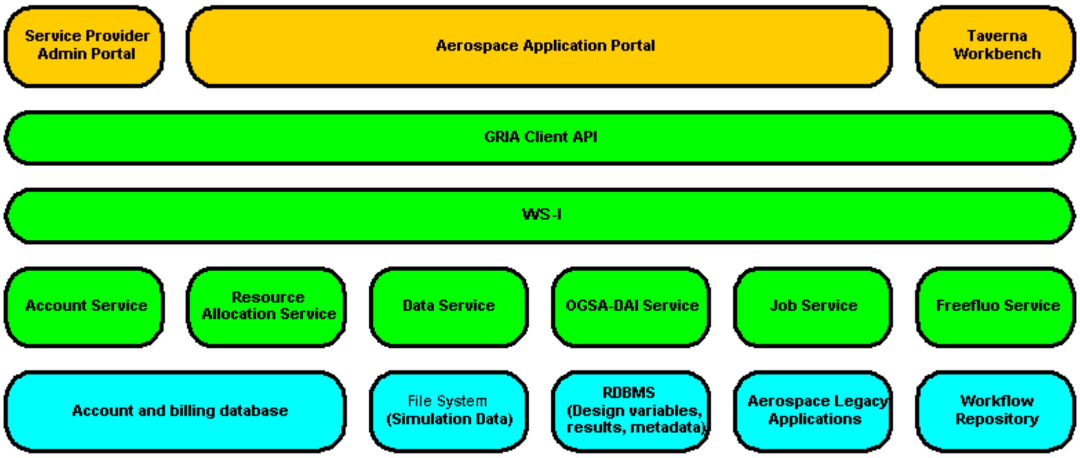

Figure 6 Aerospace architecture

Each service provider hosts six core services:

- Account service records resource usage for billing purposes
- Resource allocation service manages and assigns computation and data resources hosted by the service provider
- Data service stores simulation data files
- OGSA-DAI service provides access to database resources for storing metadata
- Job service executes legacy applications
- Freefluo service executes application workflows containing legacy jobs and post processing tasks for writing metadata to OGSA-DAI

Three user interfaces will be provided including:

- A service provider administration portal that allows service providers to manage resources such as accounts, data and applications
- Taverna workbench allows expert engineers to develop application workflows and deploy them to a Freefluo service
- Aerospace application portal provides a business-focused interface to the aerospace design workflow.

4.2 Automotive

The automotive sector has developed three prototypes each demonstrating progress towards different SIMDAT objectives. The first prototype from Audi is based on the simulation data and process management system (SDM system) MSC.SimManager. For PM12 the prototype will focus on the implementation and intra-enterprise deployment of distributed database access, grid infrastructure and analysis services components. Later on, the prototype will be extended to support virtual organisations addressing inter-enterprise issues such as firewalls, security, encryption, etc.

The second prototype from LMS/Noesis concentrates on the coupling between workflows and analysis services in the automotive scenario. The main purpose of the LMS SAMD is to develop a prototype demonstrating the current state-of-the-art in workflow definition and execution based on analysis services. GRIA was selected as the Grid infrastructure after examination of several Grid and web services frameworks. OPTIMUS has been integrated with GRIA demonstrating workflow composition and execution on the Grid. The prototype will be extended to compose and execute workflows over multiple service providers accessing a multi-CPU computer cluster.

The third prototype from RENAULT/IDESTYLE demonstrator uses Grid-enabled PSE environments to between automotive OEMs and partners. RENAULT involves numerous partners or subsidiary companies in the design process of vehicles, IDESTYLE in particular. The Grid-enabled PSE will enhance collaborative work between RENAULT and its partners by sharing analysis services and particularly by sharing or exchanging data in a more convenient and secure way. GRIA has been deployed to support this prototype specifically to provide distributed analysis services, secure data transfers and delegation to a trusted partner. The GRIA services have been integrated with Inforsense workflow technology to support the overall design process.

4.3 Meteorology

The meteo sector is developing a common infrastructure for the collection and sharing of distributed meteorological data: the Virtual Global Information Systems Centre (V-GISC). The prototype is based on the V-GISC architecture Data Communication Infrastructure (DCI) [5]. The aim of the prototype is to validate that V-GISC can be built on a distributed and loosely coupled Grid architecture. Grid technologies will be used to offer external interfaces to the V-GISC and to federate the partner’s data repositories. The main objectives of the demonstrator are the following:

- Build the virtual database foundations
- Implement the authentication service of the VO
- Implement part of the Access services

A combination of web service and Grid technologies such as OGSA-DAI has been used to integrate the legacy databases within a virtual database. The authentication service (AuthN) has been integrated to deliver single sign-on credentials to users. AuthN is based on the Public Key Infrastructure (PKI) allowing X509 user authentication. For the PM12 prototype a trivial authorization service (AuthZ) has been implemented that grants access to data for any authenticated user.

A portal providing a basic access to the virtual database functionalities has been built. This portal provides interactive access using web technologies and batch access using Grid technologies. This demonstrates that programs and humans can access the virtual database service. It also demonstrates that the Virtual database service can be integrated in a wider Grid.

4.4 Pharmaceuticals

The Pharma prototype coordinates pharmaceutical services, data and resources in order to support industrial and scientific pharmaceutical research. The prototype extends SRS software to provide client federation of multiple SRS services deployed across different domains. The system consists of three main components:

- Directory Service (Semantic Broker)
- Node Broker (SRS Nodes)
- Federated Portal (User Interface)

The Pharma prototype is deployed on an untrustworthy network, the internet, for communication of highly sensitive data and to give access to services and resources. The pharma prototype uses deployed a web services Grid with end-to-end security components from the GEMSS Grid developed by NEC. Pharma users require that access is granted only those who are authorized, that the data transmitted does not get eavesdropped, manipulated and is authentic. NEC's E2E security toolkit currently provides a security mechanism that can ensure a high level of security of the data transmitted from one Grid endpoint to another, even across intermediaries. In future, the E2E toolkit will be extended to provide dynamic access control mechanism that supports a variety of access models typical of Pharma collaboration scenarios.

5 Conclusions

The first integrated Grid infrastructure software has been successfully delivered and deployed in four SIMDAT application prototypes. The prototypes demonstrate how the Grid infrastructure can support inter-Enterprise collaboration for the design of complex products. These prototypes include:

- **Aerospace (BAE, EADS, CEDG):** pan-European inter-enterprise multidisciplinary (optimisation, aeroacoustics, aerodynamics) collaborative configuration design of a low-noise, high-lift landing system
- **Automotive 2 (LMS, MSC, AUDI)**: Analysis service provision
- **Automotive 3 (Renault/ESI/IDEStyle):** Virtual organisation for collaborative car design supporting confidentiality constraints of components between organisations
- **Automotive Knowledge (Audi, SCAI, Inforsense)**: Distributed data mining using WEKA toolkit

Following the delivery of the PM12 prototypes the Grid infrastructure will be evaluated by end-users. This evaluation will result in new requirements that will be factored into the Grid infrastructure roadmap. In addition, other Grid infrastructure technologies developed and deployed within Pharma and Meteo sectors will be identified and where appropriate incorporated into future releases of Grid infrastructure either by integration or by demonstrating interoperability using appropriate standards.

The results of PM12 prototype evaluation and a review of the wider Grid infrastructure landscape will be documented D.2.1.3: Report on SIMDAT Integrated Grid Infrastructure to be delivered at PM16.

6 References

1. SIMDAT Annex 1 Description of Work
2. D2.1.1 Consolidated Requirements Report, Roadmap and SIMDAT Infrastructure Design
3. D17.1.1 Initial version of the application scenario, demonstrating connectivity and the operation of the model problem
4. D11.1.2 Documentation of Software Design and Initial Implementation for SIMDAT Automotive Demonstrators (Integrates D9.1.2, D10.1.1, D11.1.3)
5. D20.1.1 Specification of Meteorological scenario for connectivity test case
6. D20.1.3 SIMDAT Meteo Demonstrator Technical Design
7. D14.1.2 Production of first working prototype