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

Deliverable

D2.3.1 Third report on Grid-infrastructure state-of-the-art and SIMDAT Infrastructure roadmap

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

Due date of deliverable: 01/09/2007

Actual submission date: 24/10/2007

Lead contractor for this deliverable: IT Innovation Centre

Revision: 1.1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination level		
PU	Public	
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)	X

Revision history

Date	Version	Author	Modification
06/10/2007	1.0	Mike Boniface, David Sansome, Stephen Philips, Panos Melas (IT Innov) Changtao Qu (NEC)	Initial version
08/10/2007	1.1	Mike Boniface	Added job service enhancement section, description of aerospace scenarios and executive summary
09/10/2007	1.2	Mike Boniface	Final version including updates from SIMDAT reviewers

Copyright

Table of contents

Executive Summary

This document is the deliverable D2.3.1 "Third report on Grid-infrastructure state-of-the-art and SIMDAT Infrastructure roadmap" of the EU IST-2002-511438 SIMDAT project, as specified in the Annex 1- "Description of Work". The document presents a description of the technology developments for WP2 Integrated Grid Infrastructure for the first part of the Knowledge phase PM31-PM36 and roadmap for the knowledge phase PM37-PM42.

The knowledge phase enhancements were planned to be delivered in two phases. The first phase focuses on capabilities that can be developed and delivered in sufficient to for integration and deployment at the 3rd SIMDAT review meeting in November 2007. The developments targeted new requirements identified during the connectivity and interoperability phase evaluations and included client-side registry supporting group-based resource publication and discovery, significant performance optimisations of the SLA management service and various client interface and API enhancements. The job service resource manager integration architecture has been refactored based to allow multi-resource managers to be deployed at a single GRIA service provider and to allow resource managers and applications to report usage metrics that can be constrained using an SLA. The Semantic Broker has been interfaced with GRIA to provide service discovery and selection infrastructure to support an industrial procurement business process. The architectural design of longer term features proceeded in parallel to the first phase to ensure that capabilities, identified as barriers to adoption, (a licensing infrastructure, and client-side policy based management) continues and is delivered in time for the forth Integrated Grid infrastructure release due at the end of the knowledge phase.

The third Integrated Grid Infrastructure software has now been adopted by the auto, aero and pharma application sectors. The prototypes in each sector continue to exploit the advanced security and manage delivered by WP2. A summary of the prototypes is given in below (detailed descriptions are available in the relevant deliverables: D9.3.1, D10.3.1 and D11.3.1)

- Aerospace (BAE, EADS, CEDG, MSC, LMS): pan-European inter-enterprise multidisciplinary (optimisation, aeroacoustics, aerodynamics, structures, RSM) collaborative configuration design of a low-noise, high-lift landing system (D11.3.1)

- Automotive (Audi/MSC/LMS/Ontoprise): Inter-domain data integration between CAD, CAE and CAT teams

- Automotive (Renault/ESI/IDEStyle): Crash Compatibility Testing for collaborative car design supporting confidentiality constraints of components between organisations (D9.3.1)

- Automotive Knowledge (Audi, SCAI, Inforsense): Distributed data mining using WEKA toolkit (D9.3.1)

- Pharma B2B (GSK, Galapogas, NEC, Inforsense): B2B Bioinformatics collaborations (D11.3.1)

- Pharma Federated Prototype (ULB, SCAI, NEC, UKA, Inforsense, EMBL):: ixodus workflow with P2P data distribution (D10.3.1)

Three requirements capture and evaluation phases of SIMDAT have identified 40 requirements for the Integrated Grid Infrastructure. 82% of all requirements are now completed and 88% of all priority 1 requirements. All other priority 1 requirements are now in the work plan and intend to be delivered in the forth Integrated Grid infrastructure. During the final phase the Integrated Grid infrastructure will be supported and enhanced to address bugs and critical feature requests inhibiting

the adoption of the technology within industry. The infrastructure will be deployed within real-world demonstrators and pilot applications with both SIMDAT industrial partners and external partners interested in evaluating the technologies

Introduction

1.1 Purpose

This document is the deliverable D2.3.1 "Third report on Grid-infrastructure state-of-the-art and SIMDAT Infrastructure roadmap" of the EU IST-2002-511438 SIMDAT project, as specified in the Annex 1- "Description of Work" [1].

The document presents a description of the technology developments for WP2 Integrated Grid Infrastructure for the first part of the Knowledge phase PM31-PM36 and roadmap for the remaining project months PM37-PM48. The document describes the hardening of the SLA service based on evaluation of the initial version, extensions to the job submission service to support requirements from analysis services, licensing architecture, integration with the semantic broker and the relationship between SIMDAT developments and the state-of-the-art. The document describes the technology uptake by the application activities during the knowledge phase and early requirements identified for future work to support the SIMDAT demonstration phase. These requirements will be evaluated again during the evaluation of the knowledge phase prototypes. Finally, the document outlines a roadmap for SIMDAT Grid infrastructure developments for the remaining months of the project.

The intended audience for this document are the application and technology partners within the SIMDAT consortium, as listed in Section 3 of [1], as well as the European Commission Services. SIMDAT partners (stakeholders) have a diverse range of expertise representing both application sectors and horizontal technology activities. The document is structured to give a view of grid infrastructure from each application and technological stakeholders' perspective.

2 Technology Improvements

The knowledge phase enhancements were planned to be delivered in two phases. The first phase focuses on capabilities that can be developed and delivered in sufficient to for integration and deployment at the 3rd SIMDAT review meeting in November 2007. The developments targeted new requirements identified during the connectivity and interoperability phase evaluations and included client-side registry supporting group-based resource publication and discovery, significant performance optimisations of the SLA management service and various client interface and API enhancements. The job service resource manager integration architecture has been refactored to allow multi-resource managers to be deployed at a single GRIA service provider and to allow resource managers and applications to report usage metrics that can be constrained using an SLA. The Semantic Broker has been interfaced with GRIA to provide service discovery and selection infrastructure to support an industrial procurement business process. The architectural design of longer term features proceeded in parallel to the first phase to ensure that capabilities, identified as barriers to adoption, (a licensing infrastructure, and client-side policy based management) continues and is delivered in time for the forth Integrated Grid infrastructure release due at the end of the knowledge phase.

2.1 SLA Management Service Enhancements and Editors

The SLA service was significantly updated to provide optimised application usage monitoring and usability enhancements. New Capacity and SLA Template editors were provided to allow service

[1] SIMDAT Annex 1 Description of Work

providers to update their available resources easily and to author and publish new offerings to customers without having to write XML.

Initially the SLA service does not know about any metrics. As the service discovers metrics through capacity or SLA template files being uploaded, or by receiving usage reports from functional services, the list of metrics in the service's administration page is added to. When a metric is discovered by the SLA service in one of these ways it is stored and its definition (in terms of the human-readable fields) will not be changed by subsequent events. Sometimes the initial metric definition needs to be changed, and for this purpose the "Edit" button is provided. The form also provides a "Delete" button for metrics that are no longer required (though a metric may of course be rediscovered by the service), and a "New metric" button for defining a metric from scratch.

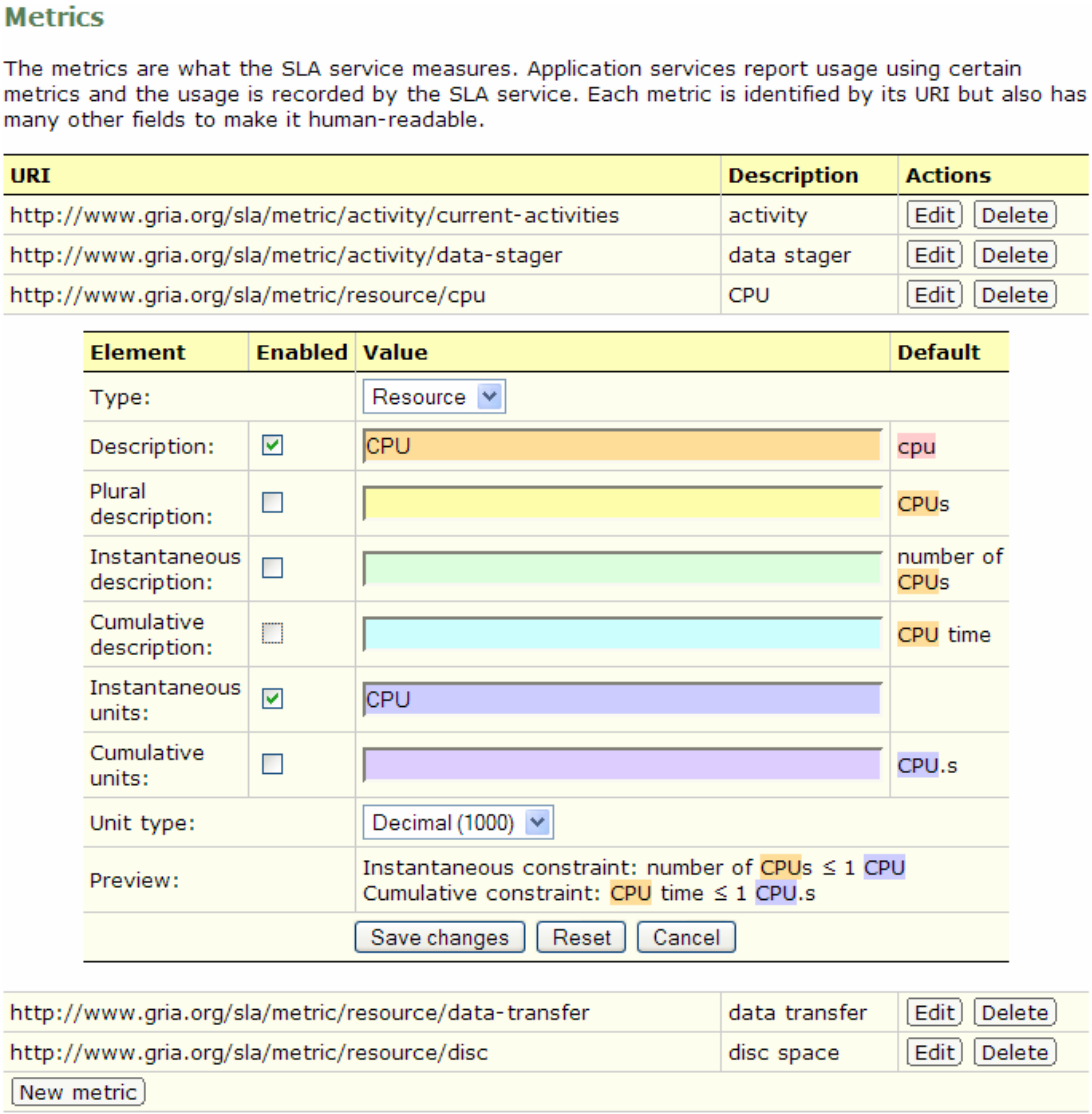

Figure 1: SLA Template Metric Editor

The metric editor is quite complicated because of the large number of fields that are provided to describe the metric. A lot of the time most of the fields can be left disabled and the system will automatically generate sensible defaults. For instance, in the screenshot above, the description has been set to "CPU" (over riding the default of "cpu" inferred from the URI) but the plural description field has not been enabled as the default of "CPUs" is correct.

When defining or editing a metric, it is best to start at the top of the form and work downwards as the default values for later fields come from the earlier fields. Each input field in the form is coloured to help understanding. The defaults shown on the right hand side are generated by taking a field and adding further text. So for instance, in the screenshot, the default for the "Instantaneous description" is displayed as "number of CPUs". This shows that the "CPU" text has come from the "Description" field (coloured orange). The preview section at the bottom of the editor gives two complete examples of how a constraint using this metric would be described.

2.2 Job Submission Service Enhancements

2.2.1 Multiple Resource Managers

The GRIA Job service has been extended to support analysis service requirements. The GRIA architecture is flexible enough to use a variety of underlying computing platforms to run jobs e.g. from single computers to clusters of workstations or even supercomputers. The following sections of this document give an overall picture of the various components of the GRIA Job Service, and then describe some common deployment scenarios.

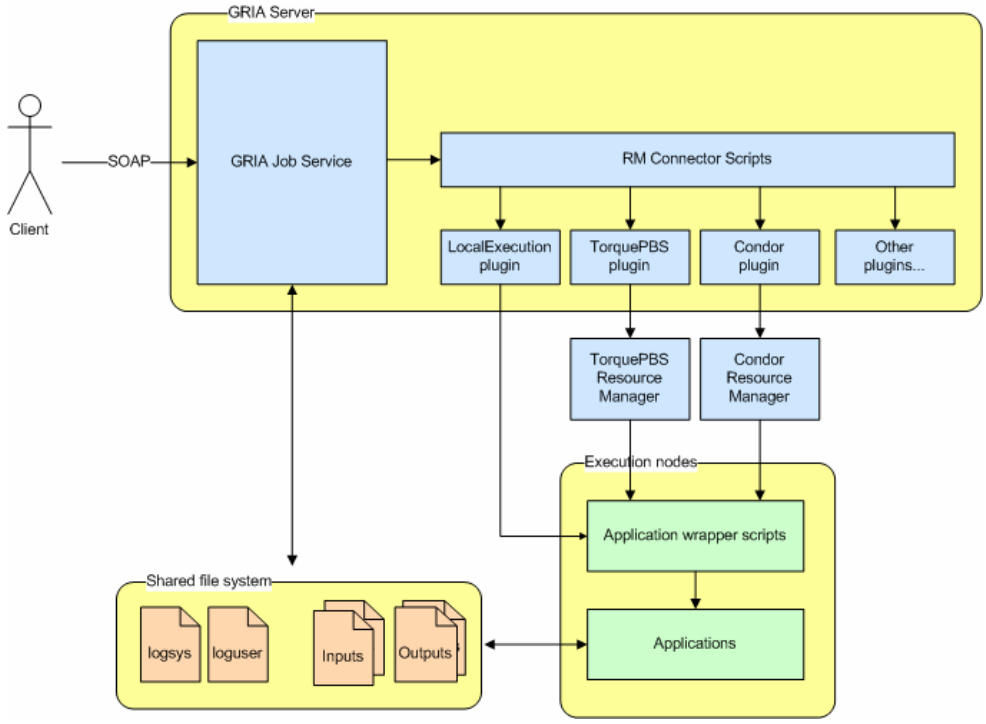

Figure 2: Overview of the job service architecture

The GRIA Job Service is separated into several distinct components: (colours relate to the above diagram)

- The Resource Manager Connectors - The GRIA Job Service can submit jobs to different resource managers such as TorquePBS and Condor, or even run them on the local machine using the LocalExecution plugin. It is able to do this thanks to the Resource Manager Connector layer - a plugin architecture written in Python that abstracts away resource manager-specific details and presents a single interface for submitting and monitoring jobs. Service providers can configure the job service to use the existing Resource Manager Plugins or they can write their own to interface with custom configurations. Version 5.2 of the GRIA Job Service can have any number of RM Connector Plugins loaded at the same time. Selecting which one to use for an individual job is done in a series of steps.

- The Application Wrapper Scripts - Each Application deployed on the GRIA Job Service needs to have a couple of small wrapper scripts installed alongside it. These scripts are responsible for providing the application with the correct files from the shared filesystem, and making sure the outputs from the application are written or copied back to the correct location. Optional wrapper scripts can also be written to cancel a job gracefully and report progress and usage information specific to that application (eg. frames rendered by a graphics package)

- The Shared Filesystem - When the user creates a job, the GRIA Job Service creates a directory for it on the shared filesystem. The administrator should ensure that this directory can be read from and written to by both the job service running on the server, and the application wrapper scripts running on the execution nodes.

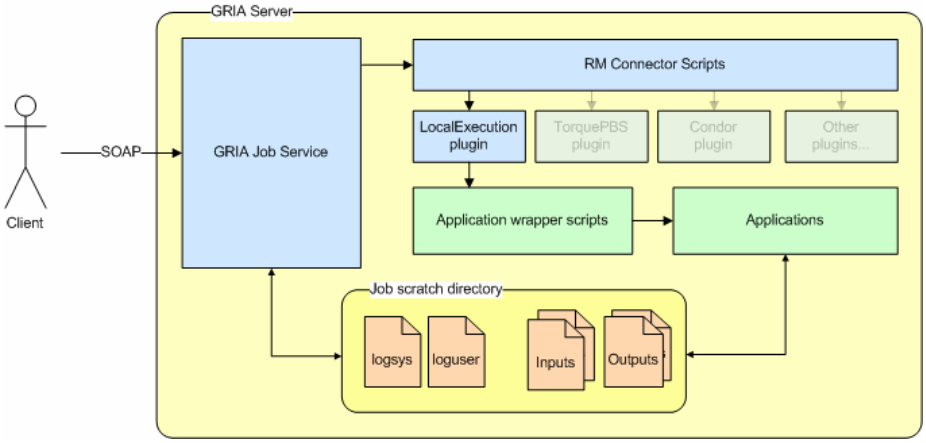

Figure 3: Local Execution Architecture

When configuring the job service, the administrator must be careful to ensure that the different files and executables can be accessed by the correct components in the system.

- The application executables should be accessible by compute nodes only, and they should be read only. Installation of the applications can be either local per compute node or over disk space shared among all nodes.

- The application wrapper scripts should be accessible by both the compute nodes and the job service. Like the application executables they should be read only, and can either be installed locally or part of a shared filesystem.

- The job's scratch directory should be accessible by both the compute nodes and the job service. The job service must have write permission on the entire directory, but applications themselves only need to access the work/ subfolder. The application wrapper scripts could setup a chroot jail inside this directory to ensure nothing else on the filesystem can be accessed. Note that the scratch area cannot be copied between compute nodes, instead it has to be exported as a shared disk space.

Figure 3 shows how GRIA can be configured to run applications locally on the server machine running Tomcat. Because of its simplicity and minimal configuration, this deployment is commonly used for demonstrations and testing. This is the default configuration for the GRIA Job Service assuming the administrator does not set up the TorquePBS or Condor plugins. Note that it is not advisable to leave the GRIA Job Service configured like this in a production environment.

The GRIA Job Service can be configured to use any number of Resource Managers. Deciding which one to use for a submitted job is done as a three-step selection process. In Figure 4, we use five made-up Resource Manager Plugins - RM1, RM2, RM3, RM4 and RM5.

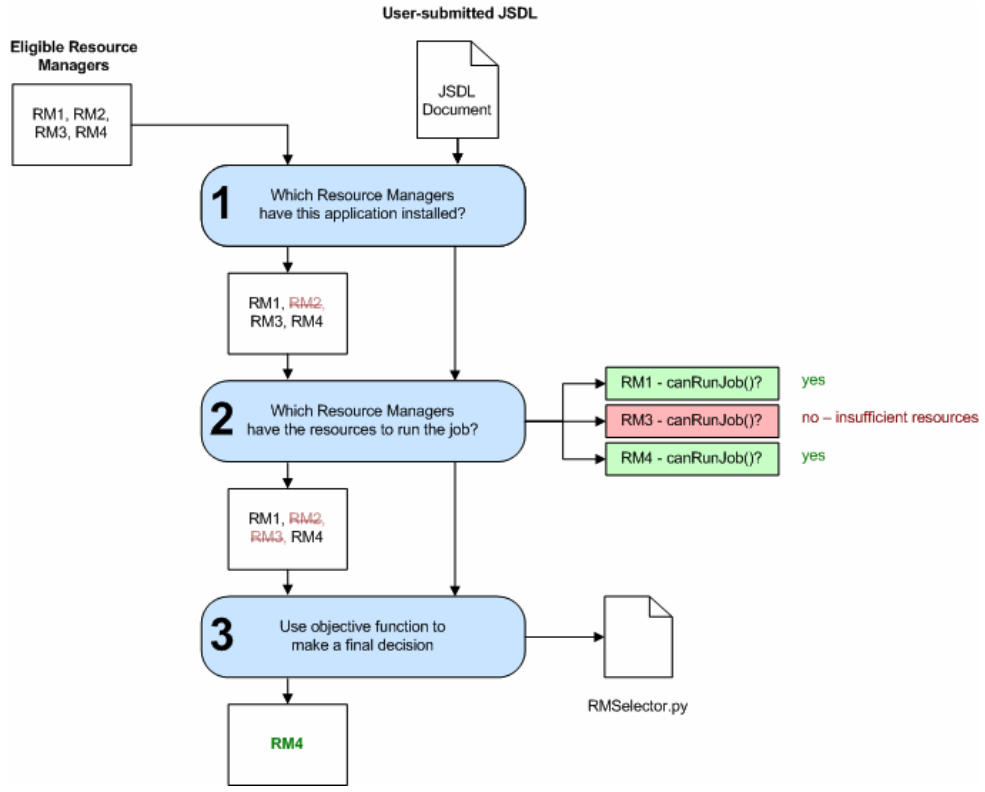

Figure 4: The decision process for selecting a resource manager

The Job Service starts by compiling a list of all the enabled RM Connector Plugins that are installed and enabled. The list can be viewed from the Job Service administration page (Figure 5) - all the plugins in the list that are not greyed out are enabled, and will be used in this selection process.

When the service administrator deploys a new application on the job service, they can indicate which resource managers have the application installed. Any resource managers not selected will be immediately excluded from the selection process, and no jobs for that application will be able to run on them.

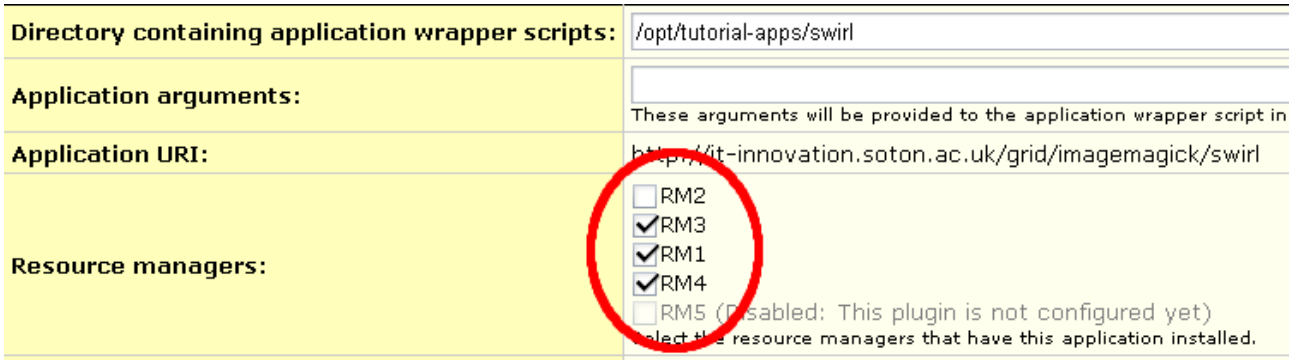

Figure 5: Selecting resource managers for an application

At this stage the Job Service asks each RM Connector Plugin whether it has enough resources to run the job being submitted. The plugins will typically look at the resource requirements section of

the JSDL, and then query the actual resource manager to see if it is able to run the job. Some checks that the plugins might do include:

- Checking whether the operating system and system architecture requested by the submitter is available on any of the compute nodes.

- Checking whether there is a compute node with enough memory to run the job.

The final decision as to which Resource Manager to use for a job is made by a Python script - RMSelector.py. The default implementation of this function is to just choose the first plugin available, but administrators can override this behaviour.

The Python interface for selecting a plugin is very simple. The Job Service will look for a Python function called selectPlugin, and call it with two arguments:

1. job - a Job object, containing information gathered from the JSDL including the job's name and its resource requirements.

2. plugins - a list of RMConnector derived objects - representing all the plugins that made it to Step 3 in the selection process. The selectPlugin function is expected to return one of these objects.

A very simple example is given below. This always selects the plugin named "RM4".

```
#!/usr/bin/env python
def selectPlugin(job, plugins):
    for plugin in plugins:
        if plugin.__class__.__name__ == "RM4":
            return plugin
    return None
```

Once the administrator has written this script, they can instruct the Job Service to use it by entering its path in the configuration page:

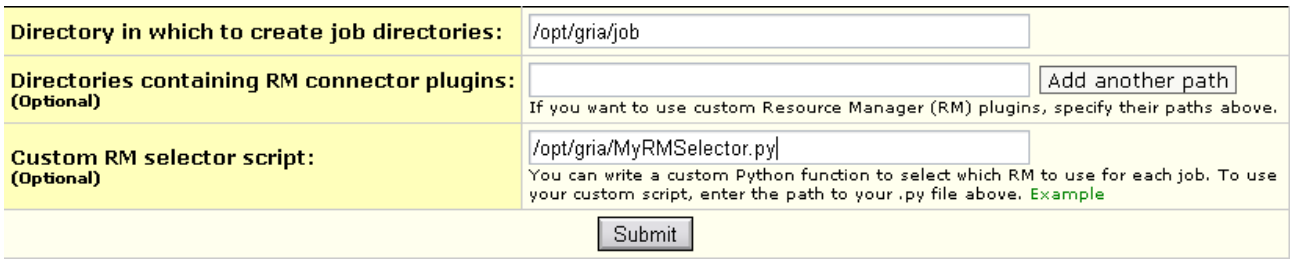

Figure 6: Using another RM Selector

2.2.2 Standard CPU Time

The GRIA Job Service receives usage reports about the current CPU utilisation of running jobs. It forwards these usage reports to the SLA service so that users can be billed according to how much CPU time they are using on compute nodes.

If the Job Service is configured to submit jobs to a heterogeneous cluster (i.e. consisting of machines with different specifications), the administrator might want users to be billed more for CPU time if their jobs are executed on faster machines. The GRIA Job Service can adjust the amount of reported CPU usage according to the measured performance of the node on which the job is running.

The process of recording the performance of each node is not automatic - the administrator must run a benchmark on each individual machine on which a job could be run. 1 standard CPU second is defined as 1 second of full CPU utilisation on a Pentium III 1GHz processor. This gives a performance of 54.3 Mflops/s using the Linpack java Benchmark.

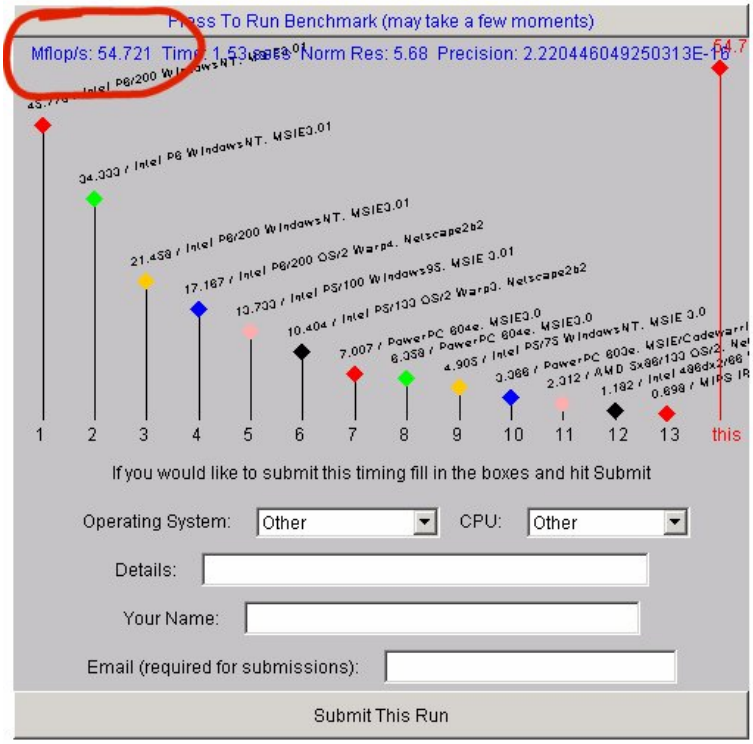

Figure 7: Benchmark output

2.3 Integration with JBoss Application Server

Integration with existing Enterprise infrastructures is a key requirement to overcome barriers to adoption of the Grid. Many Problem Solving Environments (PSEs) in SIMDAT are based on the leading open-source JBoss enterprise application service [2]. The Integrated Grid infrastructure deployment interface is based on enterprise Java mechanisms but had only been tested with the Tomcat container. This has now been extended to include JBoss to provide secure and dynamic B2B capabilities that can extend existing enterprise applications beyond domain boundaries. Full instructions for deploying GRIA in JBoss were developed and delivered to the automotive partners for simpler integration of LMS Tec.Manager and MSC SimManager both of which are based on JBoss.

2.4 Semantic Discovery Integration

Until PM36, the Semantic Broker (SB) has been integrated with both GRIA 5.0.1 and GRIA 5.1 middleware, and further deployed on a server in NEC for the public access. In Fig. 1 we illustrate the SB deployed in the GRIA 5.1 environment.

[2] http://www.jboss.org

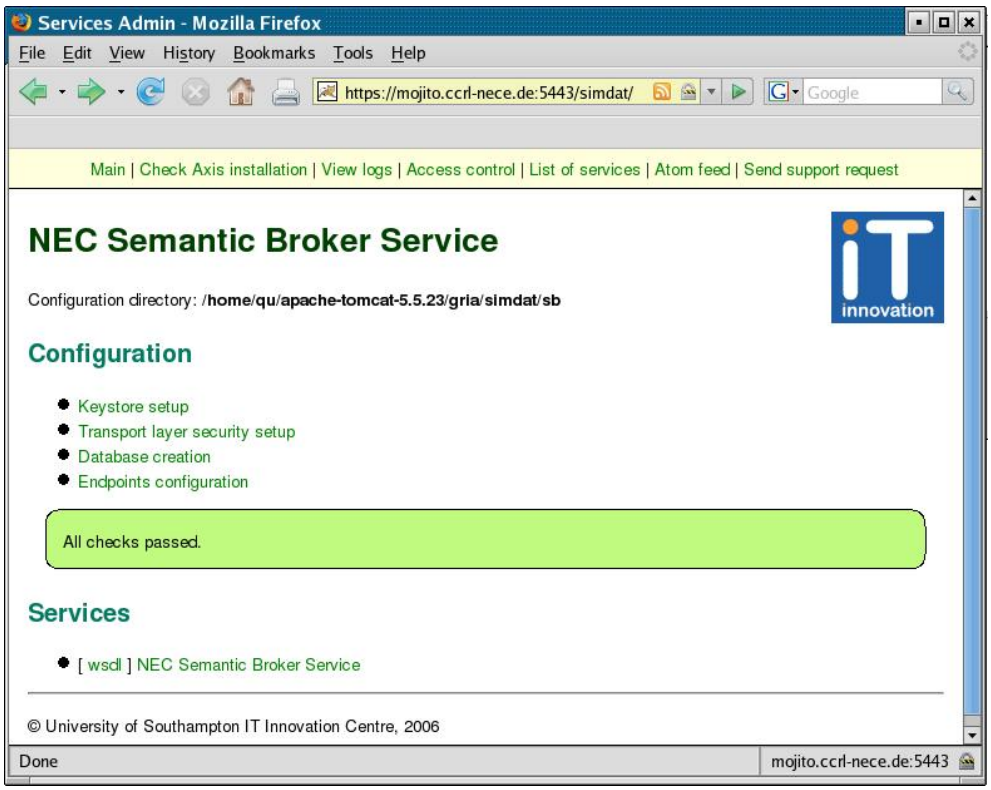

Figure 8: SB deployed in the GRIA 5.1 environment.

For the integration, following work has been done during the reporting period:

1. The SB interface is re-factored through using OWL-S as pure on-wire protocol to transport both SB queries and query results. This effectively avoids the need for transferring complex Java objects on-wire within the GRIA middleware, which can greatly simplify the integration process. Based on the GRIA client framework, we have developed two types of SB GRIA clients: (1) a SB client plug-in, which can directly be deployed in the standard GRIA client framework to let users experience the SB service through a GUI; and (2) a standalone SB client, which can demonstrate the usage of SB client side APIs for invoking SB's functionalities.
2. The SB client side APIs have been developed to simplify the usage of the SB at the client side. In particular, a GRIA client wrapper API has newly been developed, which can efficiently wrap the invocation details of the SB service. As a result, through the SB client side API, users can effectively be hidden from the complexity of both the GRIA middleware and OWL-S syntax.
3. The integration of the SB with the GRIA middleware enables SB to directly make use of GRIA's security mechanism. As a result, the access to the SB is now secured by GRIA PBAC module, which provides a dynamic access control mechanism to authenticate users and sequentially allow them to access authorized services and service operations. When a user tries to access the SB resource by invoking an operation on the SB Web service, the Process Based Access Control (PBAC) system checks that the user is authorized to perform the requested operation on that resource.

Based on the PBAC, in SB we have defined three different user process roles according to the typical usage of the SB, and further assign their respective authority on different SB operations, as listed in Table 1. Fine-grained access control over SB is thus achieved at the service operation level, which can ensure the completeness of service annotations.

Process roles	Allowed SB operations
SB Admin	getServiceMatchings, publishAnnotation, removeAnnotation, retrieveAnnotation addadmin, removeSBAdmin,getSBAdmin, addSBAnnotator, removeSBAnnotator, getSBAnnotator addSBUser, removeSBUser, getSBUser getSBServiceInfomation destroySBResource
SB Annotator	getServiceMatchings, publishAnnotation, removeAnnotation, retrieveAnnotation addSBAnnotator, removeSBAnnotator, getSBAnnotator getSBServiceInfomation
SB User	getServiceMatchings getSBServiceInfomation

Table 1: PBAC based access control over the SB GRIA service.

In terms of PBAC, sets of GRIA match rules are defined for determining a user's process roles, which can be changed dynamically thus allow users to delegate access to the SB resource to other users. The match rules are basically based on the X.509 user certificates, and can be manipulated via the SB service interface through access control operations such as *addSBAnnotator*, *addSBAdmin*, etc. These rules can also directly be manipulated via a Web interface by the SB service administrator. As an example, in figure 9 we illustrate several match rules in SB for defining the user process roles.

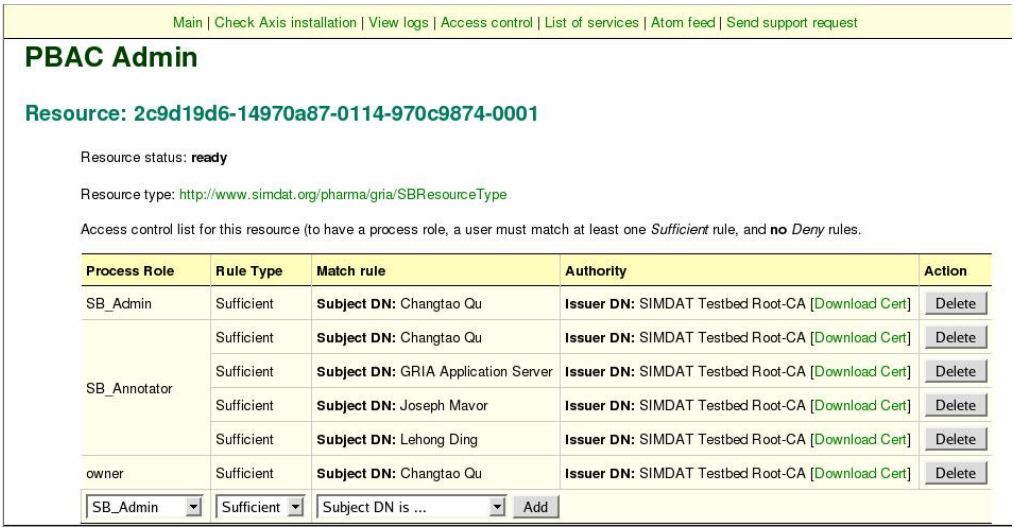

Figure 9: Assign process roles to SB users

Until PM36, the SB GRIA service based on both GRIA 5.0.1 and GRIA 5.1 is maintained in the SIMDAT Pharma Grid testbed. The service is successfully used by project partners in the SIMDAT workflow toolkit (InforSense) and service annotation toolkits (Dynamo from ULB and TUAM from SCAI).

2.5 Licensing Architecture

A critical barrier to adoption for the Grid is distributed licensing models. Two aspects need to be addressed 1) technology that can support issuance and monitoring of Grid licenses and 2) changes in Independent Software Vendor (ISV) business models. Early in the knowledge phase meetings were held with the leading license management vendor Macrovision. Although there was expression of interest between Macrovision and project partners, no further technical collaboration was achieved. WP2 decided that 1) could be addressed in SIMDAT to enable 2) to be explored with ISV's such as ESI Group.

WP2 has developed an architecture based on the SLA management service that allows a customer to procure a license from an ISV and use that license at a 3rd party service provider in a secure and accountable way. The proposed architecture ensures that the ISV maintains relationships with their customers and monitors where and how their customers are using the software.

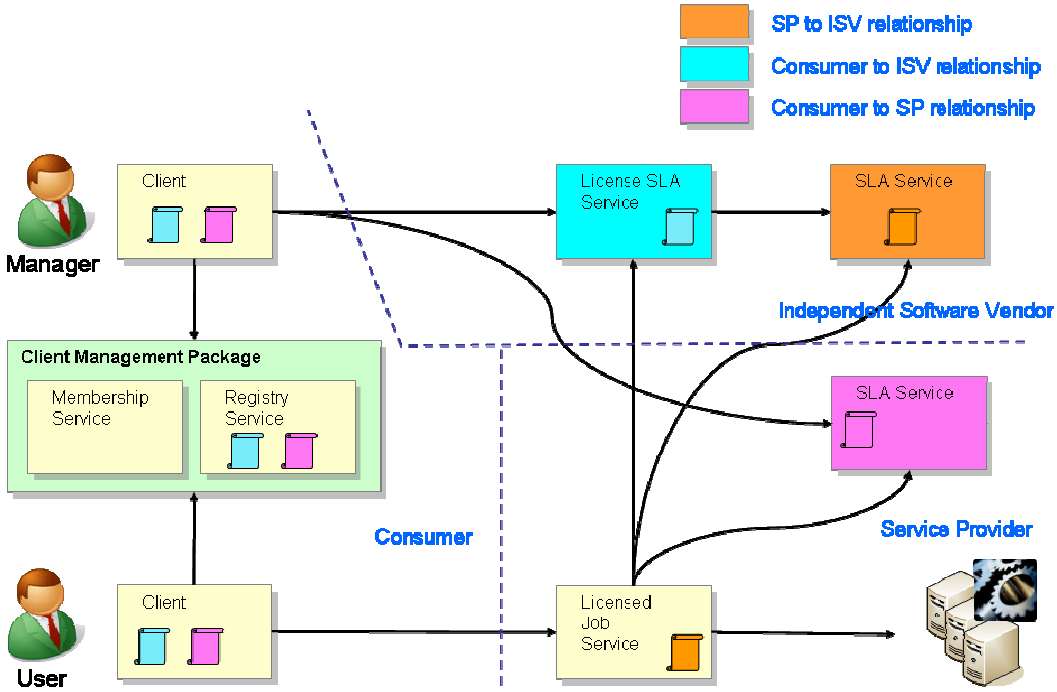

Figure 10: GRIA's licensing architecture

Figure 10 shows the stakeholders and the services that need to be deployed. The sequence of events is described below:

1. The service provider agrees an SLA with the ISV to run their software
2. The manager at the consumer agrees a license with the ISV (this is effectively an SLA)
3. The manager at the client organisation agrees an SLA with the service provider to use their resources (computation, storage and applications)
4. The resources (SLAs) are placed in the client organisations client management system, groups and policies are set up

5. A user discovers SLA, license and any required security tokens from the client management system

6. A user runs a job, passing over license and SLA to the service provider

7. The service provider checks the SLA to ensure that the activity is permitted and sufficient resource is still available to the client

8. The service provider checks with the ISV that they are permitted to run the application and that the customer has a valid license.

9. If all checks are successful the service provider can run the application

3 State-of-the-art analysis

Inter-enterprise service-oriented collaborations have now been proved in many major sectors and companies are now adopting the technologies in real-world pilot applications within their businesses. Many of the architectural principles for creating and maintaining business relationships are now understood and frameworks implementing the underlying policy management capabilities exist that build on standards that are now largely approved or at least in a recommendation phase. There are still differences of opinion about the future of some core infrastructure standards between IBM and Microsoft but at least they are still talking through the WS-Convergence initiative.

It is agreed that different collaboration models need to exist ranging for traditional virtual organisation to flexible supply chains but in each case organisations are responsible for their own trust decisions building on bi-lateral service level agreements for the provision of a wide range of IT assets such as applications, information, computation and storage resources depending on the business value exchange. These technologies will be the building blocks for the market-based service economy that will emerge over the next few years as organisations adjust their business models and operating policies to the changing market conditions that incorporate more flexible and adaptive working beyond their traditional organisation boundaries.

The Grid and Web Services brands have now largely converged under the banner of service-oriented architectures (SOA) and service-oriented infrastructures (SOI). Over the next 12 months we will see further convergence between overlapping technologies as the enterprise IT data centre evolves. Virtualisation, cluster management and multi-core processing solutions will jostle for position. Virtualisation will incorporate much of the load-balancing capabilities from traditional cluster computing approaches and we will see an increased support for virtualisation within hardware. We will also see an increase in mergers and partnerships between SOA providers and enterprise solution providers to meet the increasing demand from customers for extended enterprise.

.NET has now firmly established itself as a framework for inter-enterprise service-oriented infrastructures after many years of promotion and technology development. The move of Microsoft into the HPC arena last year further emphasises their commitment. GRIA continues to evolve to support dynamic B2B collaboration through service provision for industry and commerce. UNICORE 6 has been released to provide an HPC infrastructure building on both Web Service and OGSA specifications and is now a better reference implementation of OGSA than Globus. Within the e-Science community two important Grid infrastructure activities continue. The second phase of the EGEE project continues supporting academic research across Europe based on the gLite middleware. In the UK, OMII-UK continues to provide software and support to enable a sustained future for the UK e-Science community. A recent change of strategy has seen OMII focus on software solutions for e-Science.

European Grid and SOA research has been significantly restructured. Grid has largely become an e-Science research infrastructure activity with industry and commerce focusing on software, services and SOA. NESSI has established itself as a major player in European research within FP7. The NESSI-Grid Strategic Research Agenda has evolved through open meetings and public comment and it is expected that many NESSI approved Integrated Projects will be funded during the first call focusing on the identified challenges for business.

4 Work provided to the application activities

During the Knowledge Phase the application activities further adopted Grid infrastructure technologies. Aero, auto and pharma sectors deployed GRIA 5.1 and started to explore how service-level agreements can support provisioning and management B2B and B2A (Business to Academic) collaborations. The meteo sector completed an in-depth analysis of technologies to support their virtual organisation. WP2 supported the meteo sector providing an analysis of technologies and standards whilst proposing solutions for specific operational challenges. The improved stability of the integrated Grid infrastructure, using open Web Service and Grid specifications, has allowed application sectors to develop extensive proof of concepts that can be evaluated within their businesses. The following list provides a description of the integration tasks with other work packages and application transfer has been explored during the Knowledge Phase.

The third Integrated Grid Infrastructure software has now been adopted by the auto, aero and pharma application sectors. The prototypes in each sector continue to exploit the advanced security and manage delivered by WP2. A summary of the prototypes is given in below (detailed descriptions are available in the relevant deliverables: D9.3.1, D10.3.1 and D11.3.1)

- Aerospace (BAE, EADS, CEDG, MSC, LMS): pan-European inter-enterprise multidisciplinary (optimisation, aeroacoustics, aerodynamics, structures, RSM) collaborative configuration design of a low-noise, high-lift landing system (D11.3.1)

- Automotive (Audi/MSC/LMS/Ontoprise): Inter-domain data integration between CAD, CAE and CAT teams

- Automotive (Renault/ESI/IDEStyle): Crash Compatibility Testing for collaborative car design supporting confidentiality constraints of components between organisations (D9.3.1)

- Automotive Knowledge (Audi, SCAI, Inforsense): Distributed data mining using WEKA toolkit (D9.3.1)

- Pharma B2B (GSK, Galapogas, NEC, Inforsense): B2B Bioinformatics collaborations (D11.3.1)

- Pharma Federated Prototype (ULB, SCAI, NEC, UKA, Inforsense, EMBL):: ixodus workflow with P2P data distribution (D10.3.1)

WP2 has supported all of the application sectors with most effort going to the meteo and auto activities. WP2 has provided an in-depth analysis of the trust domain VO model and shown how dynamic access control components can be used to implement their solution. Significant work has been done with the automotive sector in prototypes addressing inter-domain collaboration that goes beyond traditional organisation boundaries, including:

- Analysed the CCT scenario and provided components based on WP2 security and management capabilities that allows two OEM's to perform crash compatibility tests whilst protecting the IPR of those involved

- Analysed the requirements and developed a design for the integration of external meshing suppliers
- Analysed security models for integrating different domains within Audi (CAD, CAE, CAT)

4.1 Aerospace Client Usage Patterns

WP2 has developed different service-oriented infrastructure deployments to support different collaboration patterns typically seen within the aerospace industry and how the Integrated Grid infrastructure security and management services can be configured. These scenarios have been explored with BAE, EADS both from a SIMDAT and external perspective with companies such as Airbus.

In the connectivity and interoperability phase the aerospace prototype collaboration model was based on a value network, where BAE accessed at design service at the UoS who subcontracted engineering expertise to other suppliers. This did not ideally match the requirements for the prototype where a prime contractor is responsible for orchestrating the collaboration. To solve this problem the aerospace partners developed a virtual organisation by federating GRIA membership services (See Figure 11. Each organisation created a membership group that included engineers working in the SIMDAT project allowing them to maintain control of their team. An additional group was created at the prime contractor that logically represented the virtual organisation (VO). Each membership group was added to the VO group so that members of each organisation group could get access to a VO security token. This scenario shows how bi-lateral relationship management can be used to form virtual organisation collaboration models with the benefit of each organisation maintain control of the teams and resources

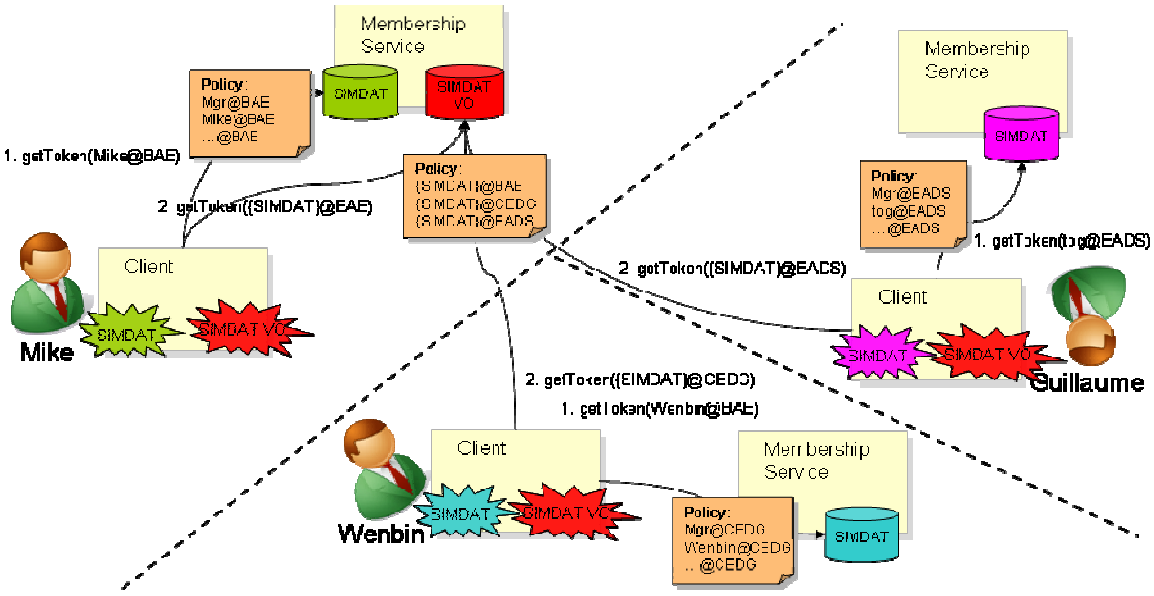

Figure 11: Federating membership services to create a VO

Another common scenario is where the prime contractor wants to share data with a subcontractor. In this scenario the subcontractor maintains a team of engineers working on a project but needs access to specific design data for their work. A manager at the prime contractor adds the membership group rule to the data service policy. Contract engineers can then get a security token from their local membership group that is then used to access the prime contractor's data directly.

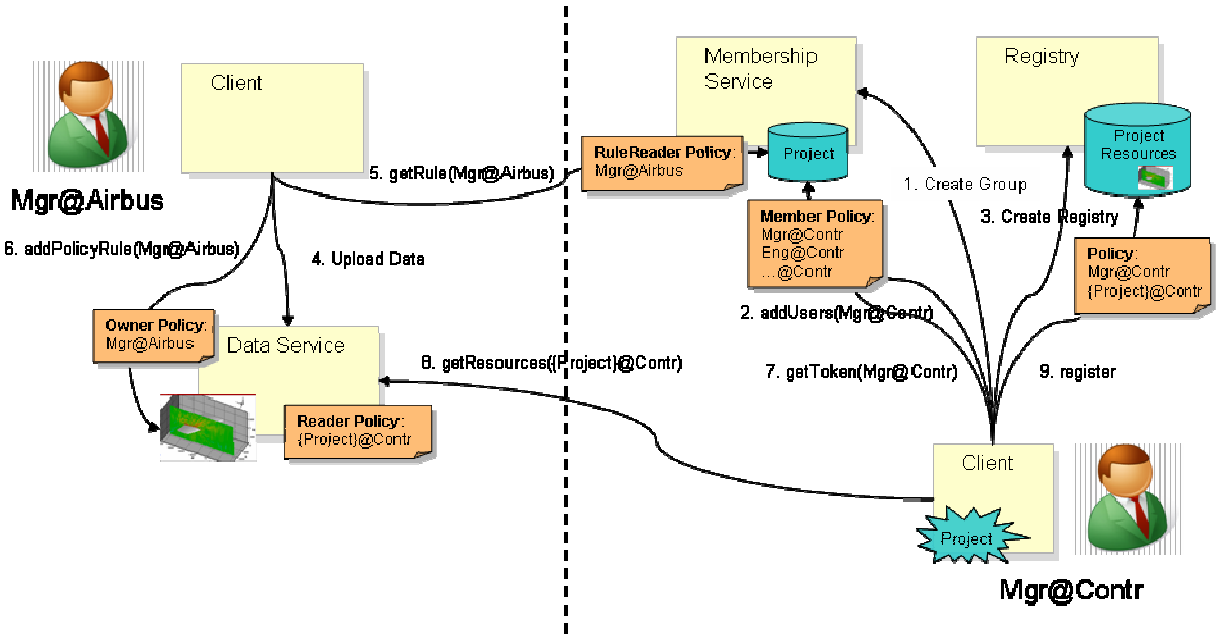

Figure 12: Sharing data with a subcontractor

The final scenario is where the prime contractor wants to share computation, storage and application resources with a contractor. The manager at a prime contractor defines a SLA templates and creates an SLA for their collaboration with the contractor. The contractor is able to use the SLA to access the prime contractor's job service and execute a prime contractor's application. The contractor then takes the output of that application and can then uses it as input for a local application.

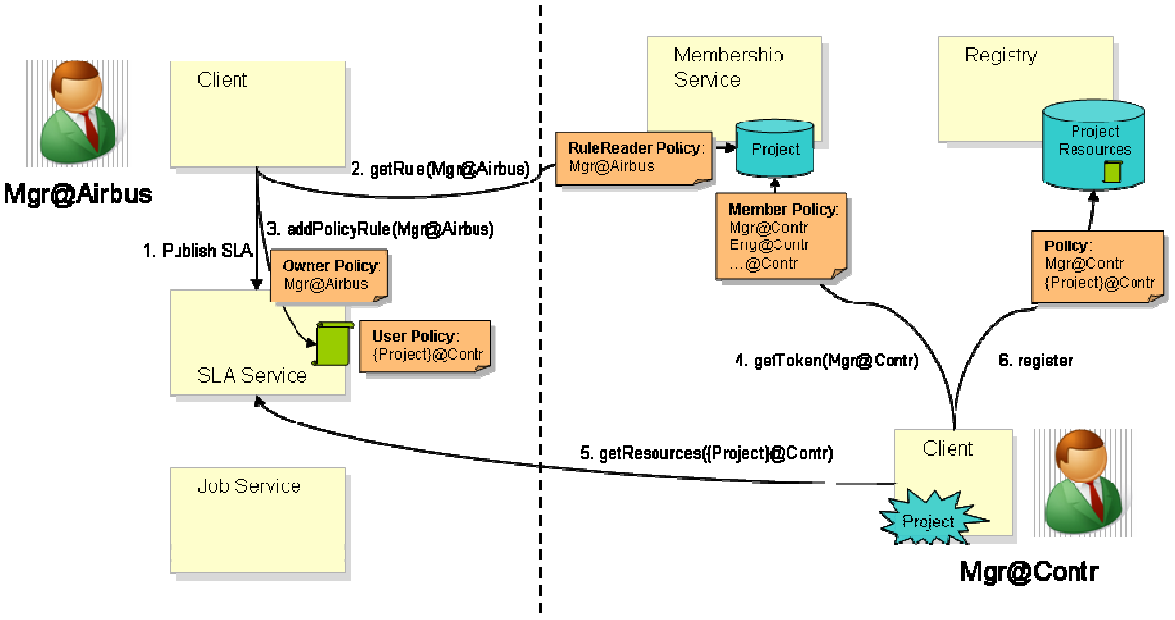

Figure 13: Providing resources to a subcontractor

We showed how the Integrated Grid infrastructure can support these different scenarios by exploiting the benefits of the service oriented architecture and dynamic security policies. No changes needed to be made to the software and within a few hours proof-of-concepts were delivered. The final results were demonstrated to Airbus

4.2 Crash Compatibility Prototype Components

WP2 has developed orchestration components for the automotive activity for the crash compatibility test prototype. The prototype was an extension of the OEM/Supplier integration where intellectual property needed to be protected between participating stakeholders. The solution adopted, as discussed in deliverable D4.2.3/D4.24 [3], is for the trusted third party to play an active role in job execution and data ownership. The principle is that the TTP is the owner of all data stagers and runs the jobs under the direction of the OEM orchestrating the crash compatibility test. As owner of the input data, the TTP can assign appropriate access permissions to each OEM for the inputs they need to provide.

In the implementation three applications are deployed at the trusted third party as shown in Figure 14 (Green rectangles signify OEM 1 data and red rectangles signify OEM 2 data):

- Setup: responsible for creating the jobs and data stagers with the appropriate permissions (inputs: certificate-A, certificate-B, outputs: job-EPRs)
- Assemble: responsible for creating the assembly from the both cars (inputs: input-A, input-B, outputs: log-A, log-B, model)
- Run: responsible for executing the crash compatibility test and creating the results (input: model-EPR, outputs: output-A, output-B)

OEM-A starts the process by running the 'setup' application, passing in the certificates of OEM-A and OEM-B. The setup application:

- Creates assemble job.
- Transfers ownership of the job to OEM-A.
- Transfers ownership of input-A and log-A to OEM-A.
- Transfers ownership of input-B and log-B to OEM-B.
- Creates a run job.
- transfers ownership of the job to OEM-A.
- Transfers ownership of output-A to OEM-A.
- Transfers ownership of output-B to OEM-B.
- Stores the EPR of the assembly "model" output locally

[3] D4.2.4 Impact of SIMDAT Grid Technology development on Application activity business processes…"

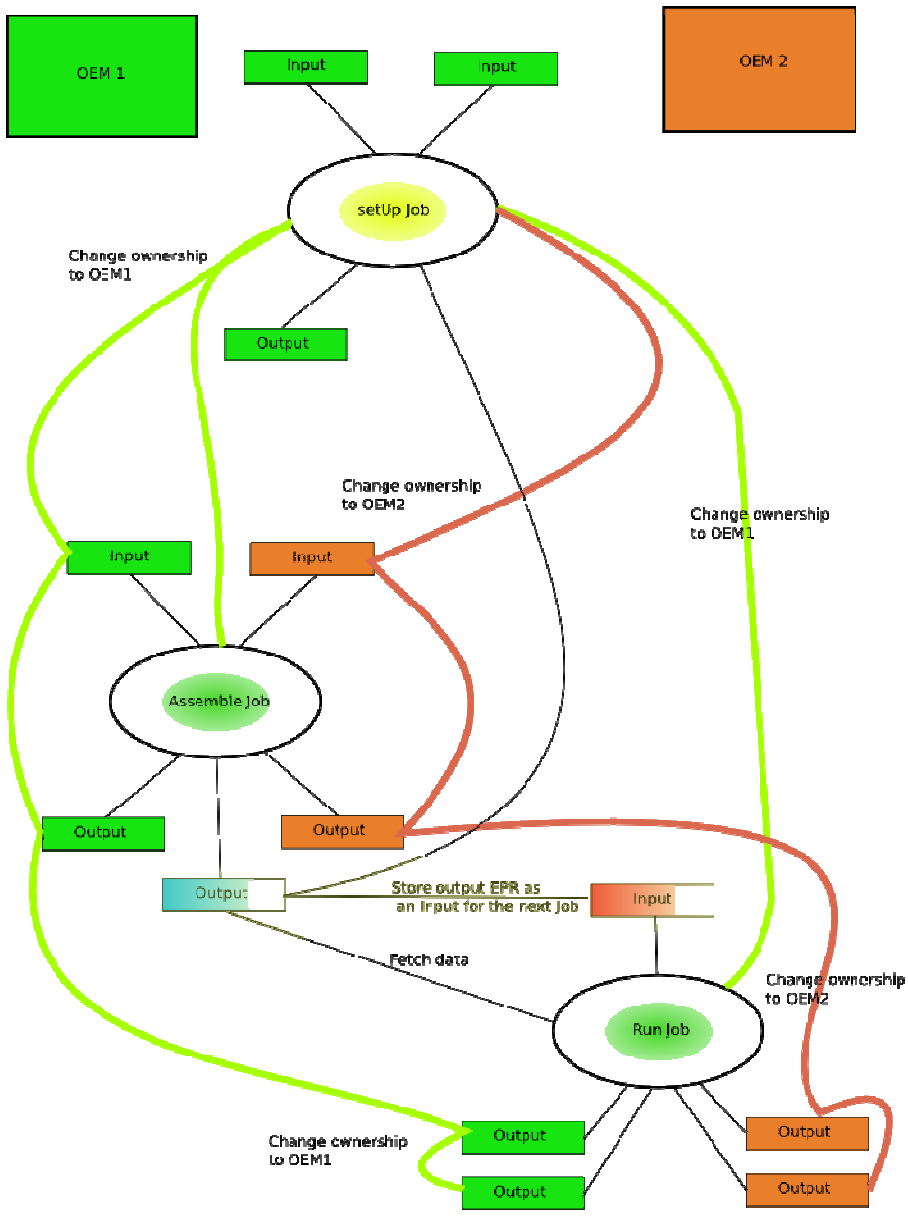

Figure 14: Crash compatibility test workflow at TTP

The assembled "model-EPR" input remains at the TTP and is not given to either OEM as the information needs to be protected. Once the setup has been completed the test can be executed. The process includes the following steps:

- OEM-A uploads to input-A.
- OEM-B uploads to input-B.
- OEM-A runs the job.
- OEM-A checks log-A.
- OEM-B checks log-B.

When the logs look OK, OEM-A runs the assembly job. This job copies from the data stager whose EPR is in its input (this will be a local file) and then runs. When it is done:

- OEM-A downloads output-A.
- OEM-B downloads output-B.

The setup application was implemented as a Java application using the GRIA API and delivered and successfully deployed by the automotive partners

5 Specific and modular requirements documentation

SIMDAT has an iterative development cycle and the project is currently in the 3rd phase. Three requirements capture and evaluation phases of SIMDAT have identified 40 requirements for the Integrated Grid Infrastructure. 82% of all requirements are now completed and 88% of all priority 1 requirements. All other priority 1 requirements are now in the work plan and intend to be delivered in the forth Integrated Grid infrastructure. WP2 expects further requirements to be identified following Infrastructure accreditation in WP4 and early insights into this process have influenced the final workplan for the remaining phases of the project (See section 6).

- Green: completed
- Yellow: in progress
- Red: not planned
- Grey: another work package

ID	Description	Applications	Priority	Complexity	Status
EM2	Support integration of multiple backend queuing systems with a single analysis service	Auto, Aero	1	2	Ongoing
EM3	Allow users or service providers to pause and resume jobs	Auto	1	2	Ongoing
EM5	Support a standard analysis service interface	Auto	1	1	Ongoing
EM6	A client shall be able to specify their exact requirement for the number of input and output stagers	All	1	2	Ongoing
EM7	A client shall be able to describe a job's resource requirements at job creation time.	All	1	3	Ongoing
EM8	A client shall be able to specify command line arguments at job creation time	All	1	3	Ongoing
EM9	A client shall be able to view application status information, in addition to the overall status of an activity	All	1	3	Ongoing
EM10	Clients shall be able to copy data from local stagers to the job working directory during job execution	All	1	2	Ongoing
EM11	Clients shall be able read intermediate output data generated by applications that have been terminated	All	1	2	Ongoing
EM12	Resource Managers shall be able to report platform usage metrics for SLAs	All	1	1	Ongoing
EM13	Applications shall be able to report application usage metrics SLAs	All	1	1	Ongoing
EM14	Application Metadata shall support command line arguments, stager arrays, modifiable stagers and optional stagers	All	1	1	Ongoing
IS7	Semantic broker integration with discovery infrastructure	Pharma	1	1	Ongoing
R5	Support Grid license distribution	Auto, Aero	1	1	Ongoing
S2	Support trust relationships that are grounded in out-of-band policies rather than the ability to pay for a service	Meteo	1	1	Ongoing
S7	Convergence between GRIA and NEC DAC security architectures	Pharma	1	1	Ongoing
S8	Integration with home authentication schemes	Aero, meteo	1	1	Ongoing
S9	Portal security	Auto, Meteo, Pharma	1	1	Ongoing
IN3	Integration with Visual Composer	Auto	2	3	Ongoing
U2	Provide a technical FAQ	All	2	3	Ongoing

ID	Description	Applications	Priority	Complexity	Status
U4	Improve exception handling and error messages	All	2	3	Ongoing
CI3	Client-side evaluation of service provider performance and heart-beat checking.	Pharma	3		Not Planned
EM4	Provide a simple way of giving parameters to an analysis service	Auto	3		Not Planned
S1	Compliance with organisation security policies and standards such as ISO/IEC 17799/27001	All	3		Not Planned
S3	Support auditable logging	Aero, auto	3		Not Planned
CI1	Provide a generic Job submission portal	Aero, Auto	4		Not Planned
EM1	Support interactive real-time visualise of engineering simulations	Auto	4		Not Planned
R3	Support a resource reservation model	Auto	4		Not Planned
CI2	Refactor GRIA 4.3.0 Java API	Auto	1	1	Completed
IS1	Support analysis services discovery	Auto, Aero	1	1	Completed
IS2	Support information discovery	Pharma, meteo	1	1	Completed
IS3	Support for building registry services for data and analysis service in the automotive SOA –Architecture	Auto	1	1	Completed
IS6	Extended registry for service selection	Aero	1	1	Completed
R1	Allow QoS commitments with softer commitments on resources and time periods	Auto, Aero	1	1	Completed
R2	Support a very generic QoS model handling a wide class of application services	Auto	1	1	Completed
R4	Support management of application and data resources	Auto	1	1	Completed
S4	Allow client organisations to locally manage access control lists for remote resources provided by other organisations	Meteo	1	1	Completed
S5	Decentralised relationship management	Meteo	1	1	Completed
S6	Assign access rights to services (per application suite)	Auto	1	3	Completed
U5	Improvement availability of service usage information	Auto	1	3	Completed
U6	Support other databases that PostgreSQL	Auto, Aero	1	2	Completed
U7	Refactor GRIA 4.3.0 services so that they can be used independently	Pharma, Auto, Aero	1	1	Completed
WSC1	Standard mechanism to access stateful resources using web services (WSRF)	Auto	1	1	Completed
U1	Improve GRIA 4.3.0 documentation	All	2	3	Completed
U3	Provide additional metadata about stateful resource ids	Auto	2	3	Completed
U8	Use a web-based installer for the keystore generation	Auto	2	3	Completed
IN2	Deployment of OGSA-DAI within Tomcat and JBoss	Auto	4		Another WP
IS4	*Store and query product performance data*	*Aero*	*4*		*Another WP*
IS5	*Workflow storage, retrieval and search*	*Aero*	*4*		*Another WP*
R7	*Data lifecycle management on metadata*	*Aero*	*4*		*Another WP*

6 Roadmap and targets for PM45

The SIMDAT Grid solution portfolio has now stabilised and Grid technology choices have now been made by the application sectors. WP2 continues to be responsible for the core, security,

service management, execution management and information services all of which have been addressed throughout the knowledge phase through Grid technology workshops and infrastructure developments.

During the final phase the Integrated Grid infrastructure will be supported and enhanced to address bugs and critical feature requests inhibiting the adoption of the technology within industry. The infrastructure will be deployed within real-world demonstrators and pilot applications with both SIMDAT industrial partners and external partners interested in evaluating the technologies. A licensing infrastructure will be provided allowing ISV's, customers and service providers to use and monitor distributed application licenses building on GRIA's SLA management service.

The tasks in the final project phase aim to support the application sectors in the deployment of demonstrators that pilot SIMDAT Grid infrastructure technologies. The support will ensure that critical bugs and feature requests can factored into the technical roadmap and to make sure that production deployment of the technology can be achieved after the project end. This includes:

- Finalisation of infrastructure components for production deployment
- Service provider decision support tools
    - audit analysis (past)
    - enhanced status monitoring (present)
    - policy-based management (future)
- Address new requirements and fixes resulting from infrastructure accreditation process
- Support and adaptation for new application demonstrators
    - address requirements detailed in the demonstrator specifications
    - additional support for new partners engaged in the demonstration process

7 Conclusion

The objective of the PM31-PM36 period was to design and develop advanced Grid infrastructure capabilities (enterprise identity integration, Grid licensing) that can support the knowledge phase prototypes and deliver a Grid solution that can be evaluated in demonstrators during the final stage of the project whilst supporting application activities in understanding how to deploy the third Integrated Grid infrastructure within their organisations and how new features such as services level agreements impact business models and operating policies.

The technology developments were structured in two phases to ensure that other workpackages had sufficient time to integrate for the 3rd review. Technology improvements have been made to the SLA management service to improve robustness and usability and the job service to align better with analysis service requirements. The semantic broker has been integrated with GRIA to provide secure semantic discovery and architecture has been proposed for Grid-based licensing based on SLAs that will be evaluated by ESI. The level of support to application activities has increased as meteo and auto address security aspects of their deployment and aim to build virtual organisations for their prototypes. Specific orchestration components have been developed for the crash compatibility testing.

The roadmap for the final project phase has been defined to make sure that production deployment of the SIMDAT Integrated Grid infrastructure can be achieved at the project end following extensive proof of concepts being deployed and demonstrated at industrial partner sites.