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

D.2.1.1 Consolidated Requirements Report, Roadmap and SIMDAT Infrastructure Design

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

Due date of deliverable: 01.03.05

Actual submission date: 07.04.05

Lead contractor for this deliverable: IT Innovation Centre

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

1 Introduction

1.1 Purpose

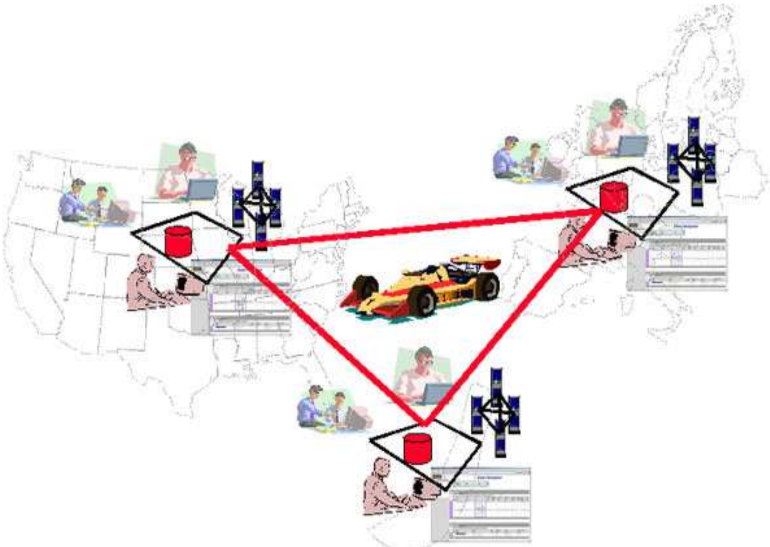
This document is the public version of the deliverable D2.1.1 WP2 Consolidated requirements report, roadmap and SIMDAT infrastructure design for the SIMDAT Integration project IST-2002-511438.

The document presents a consolidated specification of the SIMDAT grid infrastructure requirements as derived from analysis of the state-of-the-art and requirements discussions with SIMDAT application sectors and technology activities. The purpose of this deliverable is to provide a written statement of the detailed goals of WP2 integrated grid infrastructure that can be agreed by all partners as a reference at the start of implementation work. The document states what the software will do and also proposes a high-level architecture defining the scope of grid infrastructure in reference to other technology activities.

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.

1.2 Scope

Modern commercial processes deployed to manage the design, development and production of products - whether these are automobiles, aircraft, drugs, or services such as meteorology - are highly complex. In every case these processes are further complicated by external factors. Such factors include increasingly stringent regulatory environments and the commercial pressure to collaborate in order to share (or mitigate) technical and/or financial risk. The challenge for SIMDAT is to develop and deploy technology and techniques that will improve the ability of organizations to collaborate in a flexible and dynamic fashion. This collaboration is required at a deep technical level, with applications, databases and resources talking directly to one another in a controlled and secure fashion.



The complex problems to be solved all involve multiple data repositories describing aspects of the product and process development. Typically in different departments and at different sites, these currently not directly linked with each other. The Pharmaceutical sector is most advanced in integrating data repositories. Tools like SRS [35] from LION or Discovery Link [18] from IBM allow different databases and flat files to be interlinked. However, these data repositories need to be available at a common location. Therefore copies of remote data repositories are exchanged and updated periodically. In other industrial sectors like the automotive and aerospace industries, concepts for interlinking these different distributed data repositories do not yet exist.

Design of many products is essentially multidisciplinary, involving the solution of complex problems that are correlated with each other. Minimizing the risk of injury to a pedestrian in a car accident conflicts with the mechanical stability of the bonnet, and a compromise has to be found. The designer of one subsystem needs to know about the design changes of other development teams, to get direct access to simulation results of other disciplines and to get seamless access to simulation methods for all disciplines in order to successfully apply multidisciplinary optimization tools.

Correlation of data generated in different departments or at different sites within a global organization is a key problem for all industries represented in SIMDAT. Its solution requires distributed data access with a clear definition of the semantics of the databases involved, and enables the retrieval of relevant information even though it might not be simply represented in any single database. Integration through a data Grid requires not only basic mapping of semantics between the major data repositories involved but also brokering of applications that serve analysis and mining procedures. Dynamic object assembly will be necessary to create new objects that are compliant with data mining and data analysis tools. Special attention must to be paid to security, e.g. where third-party suppliers have need-to-know access to data and correlation may provide insight into confidential processes.

Knowledge services will add enormous value to virtual data repositories. Using knowledge discovery tools on a virtual repository containing all details of a design process creates the opportunity to extract and formalize successful strategies for design improvement.

The strategic objectives of SIMDAT are:

- to test and enhance Data Grid technology for product development and production process design,
- to develop federated versions of problem-solving environments by leveraging enhanced Grid services,
- to exploit Data Grids as a basis for distributed knowledge discovery,
- to promote defacto standards for these enhanced Grid technologies across a range of disciplines and sectors,
- to raise awareness for the advantages of Data Grids in important industrial sectors.

Four application sectors have been selected to cover the full range of issues to be addressed in design, development and production of complex products and services: the aerospace, automotive and pharmaceutical industries, and meteorology. For each sector a complex problem has been identified as a use-case for the project. The number of sectors addressed will be extended during the project, embracing additional applications with challenging demands to drive SIMDAT forwards.

Seven key technology layers have been identified as important to achieving the SIMDAT objectives[1]:

- an integrated Grid infrastructure, offering basic services to applications and higher-level layers;
- transparent access to data repositories on remote Grid sites;
- management of Virtual Organizations;

[1] "Grids for Integrated Problem Solving Environments: Status and Research Perspectives vs. Requirements from an Industrial Viewpoint", Bonn, April 2003. http://www.cordis.lu/ist/grids/event-announcement.htm

- scientific workflow;
- ontologies;
- integration of analysis services; and
- knowledge services.

1.3 References

1 Atkinson, et al, Web Service Grids, see http://www.nesc.ac.uk/technical_papers/UKeS-2004-05.pdf

2 AuthN and AuthZ, http://www.ucsf.edu/its/planning/authnz/

3 CSF, http://www-106.ibm.com/developerworks/grid/library/gr-meta2.html - N10041

4 D.3.1.1: Consolidated requirements report and SIMDAT distributed data repository access design.

5 D.4.1.1: Summary of operational requirements for VO deployment and management in each application sector, together with available state-of-the-art analysis and implementation plan.

6 D.5.1.1: Report on application of workflow engine.

7 D 6.1.1: Ontology technology requirement and implementation plan.

8 D.7.1.1: Consolidated evaluation and comparisons documentation for recommended implementation plan for integration of analysis services

9 D.9.1.1: Consolidated automotive requirement statement

10 D.11.1.1: Definition of SAMD reference model and use case

11 D12.1.1: Consolidated pharma requirements specification

12 D14.1.1 Technical specification of pharmaceutical scenario prototypes

13 D.15.1.2: Consolidated aerospace requirements statement

14 D15.1.1: Detailed scenario description and architecture document (Aero)

15 D.18.1.1: Consolidated meteorology requirements

16 D.20.1.1 Specification of meteorological scenario for the connectivity test case

17 D. De Roure, M. Surridge, 'Interoperability Challenges in Grid for Industrial Applications', http://www.semanticgrid.org/GGF/ggf9/gria/

18 Discovery Link, http://www-1.ibm.com/industries/healthcare/doc/content/resource/insight/941644105.html

19 The EGEE middleware architecture, published by the EGEE project consortium, at https://edms.cern.ch/document/476451.

20 GRIA, http://www.gria.org/

21 GGF, http://www.ggf.org/

22 Globus Toolkit 2.4 Reference, http://www.globus.org/gt2.4/

23 Globus Toolkit 3.2 Reference, http://www-unix.globus.org/toolkit/docs/3.2/index.html

24 GT4, http://www-unix.globus.org/toolkit/docs/development/4.0-drafts/GT4Facts/index.html

25 Ian Foster (ed), *The Open Grid Services Architecture, Version 1.0*, http://www.globus.org/ogsa/

26 I. Foster, C. Kesselman, S. Tuecke., *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, http://www.globus.org/research/papers/anatomy.pdf

27 I. Foster, C. Kesselman, J. Nick, S. Tuecke, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002, http://www.globus.org/research/papers/ogsa.pdf

28 JSDL 0.9.2 draft, https://forge.gridforum.org/projects/jsdl-wg/

29 M. Little, J. Webber, S. Parastatidis, *'Stateful Interactions in Web Services'*, http://www.sys-con.com/story/?storyid=44675&DE=1

30 M. Surridge, '*Virtual Organisations and GRIA* V2', http://www.gria.org/

31 OGSA-DAI Overview, http://www.ogsadai.org.uk/docs/current/doc/DAIOverview.html

32 OntoBroker, http://www.ontoprise.de/products/ontobroker_en

33 P2P Dynamic Networks, Sams Publishing, http://www.developer.com/java/ent/article.php/10933_1496861_4

34 Semantic Web Community Portal, http://www.semanticweb.org/

35 SRS, www.lionbioscience.com/

36 UDDI 3.0.1 Specification, http://udi.org/pubs/uddi_v3.htm

37 UNICORE, http://www.unicore.org/downloads.htm

38 WEKA Data Mining Toolkit

39 WSDL 1.1 Specification, http://www.w3.org/TR/wsdl

40 WS-Security Profile Scenarios, http://www.ws-i.org/Profiles/BasicSecurity/SecurityScenarios-1.0-20040614.pdf

41 WS-I Overview, http://www.ws-i.org/docs/20030115.wsi.introduction.pdf

42 WS-Resource Framework 1.0 reference, http://www.globus.org/wsrf/specs/ws-wsrf.pdf

43 OASIS WS-Notification TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

44 Design of the EGEE gLite middleware external interfaces, published by the EGEE project consortium, at https://edms.cern.ch/document/487871.

45 D.J.Marvin, M.Surridge and S.J.Taylor, Grid Resources for Industrial Applications, Procs 2004 IEEE International Conference on Web Services, p402.

1.4 Overview

The objective of this deliverable is to document the analysis of the Grid infrastructure requirements of the four SIMDAT application activities in the context of the state-of-the-art of Grid technology. The requirements and conclusions are based on the findings of face-to-face meetings with application sectors and their related requirements and test case specification deliverables [9, 10, 11, 12, 13, 14, 15, 16].

At the start of the requirements elicitation phase a state-of-the-art analysis of Grid technologies was undertaken based on a functional classification of features and how they are support in by each technology . Existing infrastructure such as GRIA [20, 45] and Unicore were examined along with the emerging technologies GT4 [24] and gLite [19]. This provided each application sector with sufficient knowledge of each Grid technology to make informed choices for the 12-month demonstrator and beyond.

The remainder of the document is structured as follows. Section 2 provides an overall description of Grid infrastructure and its functional characteristics putting the technology in perspective with other solutions and initiatives. Section 3 defines the specific requirements of each application sector looking at both short and longer-term objectives. Section 4 provides a discussion of the technology activity requirements focusing on how grid infrastructure would need to be extended to support the technology in the context of the application activities. Section 5 provides an initial roadmap for the implementation of PM12 demonstrators and section 6 offers some concluding remarks and next steps for Grid infrastructure within the SIMDAT project.

2 Overall Description

2.1 Grid Perspective

The Grid was devised in the mid-1990s in the USA, as a way of dynamically and seamlessly sharing resources between (mainly academic) organisations. The most successful Grid middleware from the 1990s was Globus, superseded in 2001 by “Globus Toolkit v2” or GT2 [22]. These early Grid systems came out of the HPC community, and the basic “service” they supported was just a hardware node on which clients could remotely execute programs, communicating via custom protocols. However, they did use a service-oriented “publish-find-bind” mechanism to locate machines on which to run, and they did form an organisation known as the Global Grid Forum [21] to attempt to standardise the protocols used by Globus.

By this time, the UNICORE technology [37] emerged from some German and EC projects, and its developers joined the Global Grid Forum. In 2000-01, the first efforts were made to achieve interoperability between UNICORE and Globus, which were pursued via an EC project (GRIP). While GRIP achieved some success, it is clear that there were architectural problems (especially in the area of security) that made full interoperability impossible between Globus and UNICORE.

By late 2001, it was clear that the original Globus technology and standards were too difficult to use, and unable to cope with the many “middleboxes” found in real-world systems. Web Services provided a new way to address these problems, and in early 2002, the Globus team and IBM announced a new “Open Grid Services Architecture” (OGSA) initiative [25,26,27] which would rebuild the Grid by using and “extending” Web Services. Their first implementation was GT3 [23], which stabilised during 2003, and met the “Open Grid Services Infrastructure” (OGSI) specification, which was a concrete realisation of the original OGSA.

In 2002-03, the Grid was beset with controversy over the OGSA movement, coming essentially from two sources:

i) some felt that the OGSA movement was not necessary, especially those that had invested in GT2 or its associated GGF "standards" which were now to be discarded by the Globus team;

ii) some felt that OGSA was an abuse of Web Services: the "object oriented" OGSA paradigm could never fully exploit Web Services technology and would lead to problems with higher level standards emerging from the mainstream Web Services community.

Eventually, the first group of objectors were overcome, and the concept of an OGSA founded on Web Services is now widely accepted, though many projects continued to prefer GT2 over GT3 on grounds of code maturity and stability. The second group of objectors eventually prevailed, and efforts were made to close the gap with mainstream Web Services developments. In early 2004, the Globus Alliance with IBM and others launched a new collection of standards called the "Web Services Resource Framework" (WS-RF) [42], part of which (concerned with notification) was later decoupled to become "WS-N" [43]. These proposals were made directly to OASIS (not GGF), built on existing and emerging Web Service standards, and are seen as a key step that allows convergence between Web Services and the Grid:

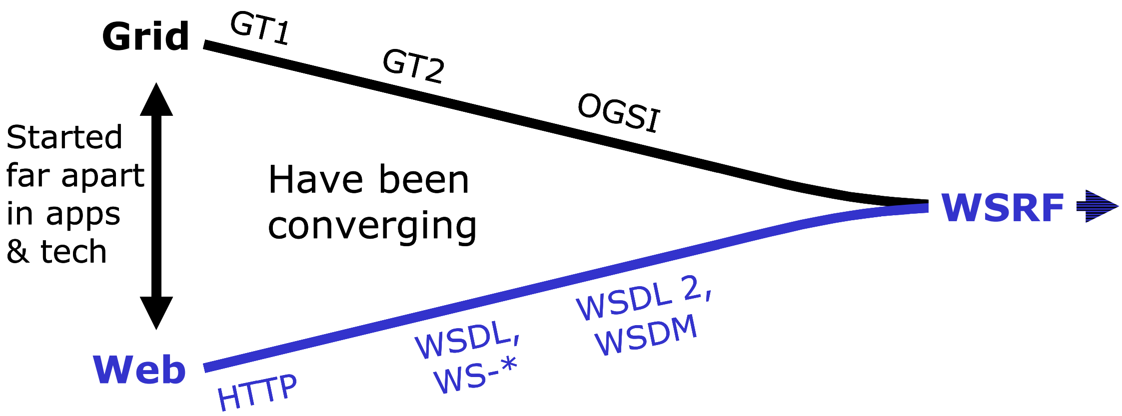


Figure 1. "Grid and Web Services: Convergence: Yes!"
(From WS-Resource Framework: Globus Alliance Perspectives, Ian Foster, Jan'04)

WS-RF certainly is more compatible with wider Web Services standards (and their likely future development), but remains somewhat controversial. This is partly because it retained many of the original OGSA "object-oriented" concepts, and partly because some of the Web Service standards it uses are not yet widely agreed or accepted.

Meanwhile, projects starting in 2001-2004 had to decide whether to (a) ignore OGSA and use GT2, (b) wait for GT3 (or now GT4), or (c) build their own Web Services middleware [1]. Several projects including GRIA took the third approach, avoiding OGSA altogether while the controversy over its future raged. This produced new Grid middleware based on pure Web Services, and led a new group of objectors (though not the GRIA developers) to conclude that Web Service toolkits would meet all their needs, and that Grid infrastructure (whether OGSI or WS-RF) was no longer needed.

The arguments on both sides over the need for OGSA (or any Grid infrastructure) really comes down to two questions: "What is the Grid?" and "How is it different from Web Services?". These questions are still being debated, but a practical position is that Grids have:

- long-lived interactions: e.g. jobs may run for weeks, data transfers last for hours, etc.,
- large-scale processing, storage and transfer requirements per application, and
- inter-site sharing of resources, capabilities and even know-how to support collaborative applications.

Web services (at least up to now) don't exhibit these features. Most web services are provided by a single site, and involve short-lived interactions (e.g. booking a flight may take a few minutes, running a Google search just a few seconds). Although the server-side infrastructure may include large servers farms connected to data warehousing (e.g. at Google), very little computational resource is consumed by a single application. The Grid therefore represents the leading edge in terms of application duration, extent and intensity.

To deal with the above characteristics, Web Services (up to WS-I 1.0 [41]) are not enough. Clients (or their applications) may come and go, or even change location while long-lived jobs are running. Services may go down and have to be moved to new locations. Data transfers may be interrupted. Users want trusted collaborators to share in these long-lived interactions, yet applications and service provider facilities must remain secure. For Grids working in industry (and increasingly in academic research), service providers want large-scale computations to be fully accounted and paid for, while users demand quality of service guarantees in return. This in turn drives the need for redundancy (and competition) between service providers, and for data caching, replication or more general "overlay networks" already found in some P2P file sharing systems [33], etc.

The key step in dealing with these long-lived, large-scale, shared and potentially mobile or replicated entities is to "contextualise" all messages sent to services. The "context" is used to refer (in a time- and location-independent way) to the entity the message is about. The service can then take the action indicated by the message, and apply it to the correct entity. One also has to create mechanisms for generating and managing these "context ids". The result is a "programming model" of the Grid in which applications assemble and manipulate long-lived entities known in WS-RF as "resources" (e.g. data, processes, computer systems, relationships) that are created, managed and accessed via services.

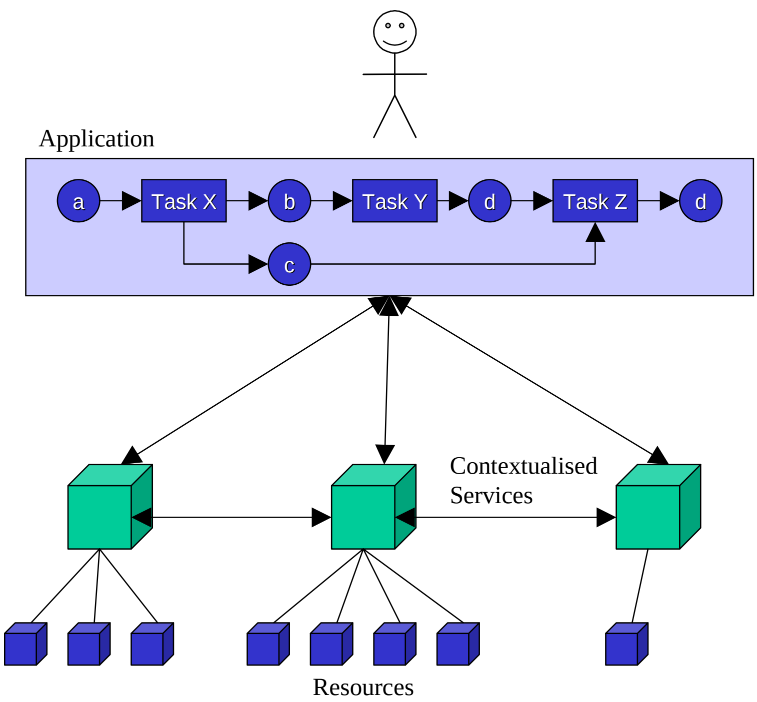


Figure 2. Grid Programming Model

In Figure 2, it should be noted that logical entities in the application space correspond to logical resources that may be replicated across services or even mobile. To cope with this, overlays can be set up, allowing non-trivial and possibly dynamic mappings between logical and physical ids.

The Grid concepts from Figure 2 can be implemented at the application level using the basic WS-I standards. In GRIA, we did this for resource accounting, for quality of service, and for secure sharing of data, all based on well-defined B2B trust models between users and (multiple) service providers of file storage, transfer and processing services. Our location-independent “context ids” are simple Uniform Resource Identifiers (URI), passed as parameters to the services. The resources identified by these URI are data sets, computational jobs, computer time or storage allocations, and accounts set up between service providers and trusted clients. As indicated in Figure 2, these resources are not only accessed directly by the client application, but also between collaborating services. To do this securely requires a model of delegation from clients to services, which is why all Grids including GRIA (but few conventional Web Services) provide a delegation model as a core part of their security infrastructure [30].

In future, we want the core features to become part of a standards-compliant architecture, so application developers can use them more easily, and so they can choose between different (reusable) implementations. The original OGSI attempted to do this by mapping “resources” to an object-oriented model of service lifecycle, but the Web Services community rejected this. The WS-RF proposals use contextualised services (similar to GRIA), but the details are still somewhat controversial and WS-RF has yet to prove its value. The challenge of standardising the Grid programming model and associated management services is therefore still unfulfilled, but we do now understand much better what is needed.

2.2 Grid Infrastructure Functionality

This section provides a general description of the functionality requirements for basic Grid infrastructure. Typically, existing Grid infrastructures provide a subset of this functionality and specific behaviour may differ between implementations. This functional classification forms the basis for grid infrastructure requirements analysis for both application and technology activities.

Figure 3 shows an overview of the main components of Grid infrastructure and their dependencies. The follow sections describe each of the green components. The data management and workflow components are coloured white as these requirements are documented in D3.1.1 [4] and D5.1.1 [6] respectively.

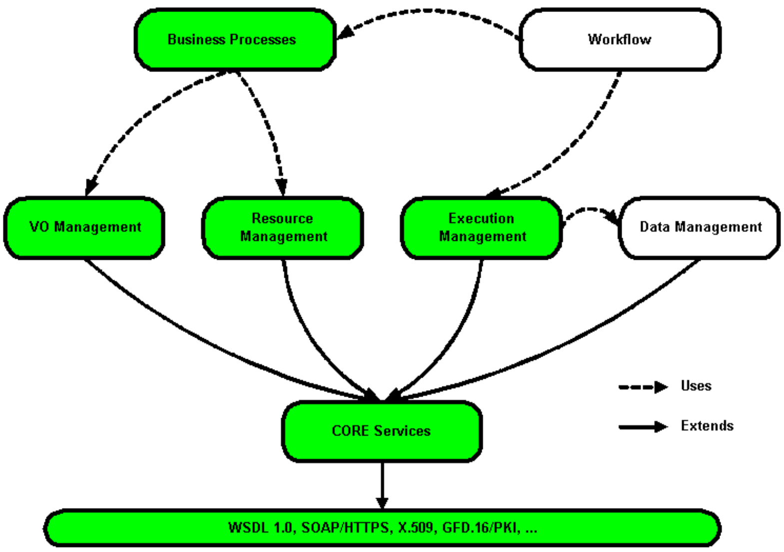


Figure 3: Grid architecture overview

2.2.1 Execution Management

Execution management is concerned with the creation and execution of tasks on the Grid, for example, a Nastran batch job submitted to LSF, an interactive parameterised design service or complete workflow. Execution management is responsible for managing the lifecycle of tasks including determining where tasks can and should be executed, setting up the tasks (input/output data) and monitoring the execution. Execution management is especially important to the Grid in circumstances when tasks need to be dynamically rescheduled, for example, if a service provider needs to move a task to another resource to meet a service level agreement.

Execution management implementations may provide differing approaches such as a simple prioritised batch job queue to a workflow enactment engine that schedules and manages the workflow allowing for QoS requirements and optimum data to process distribution. Execution management services may also interact with other services such as schedulers and brokers.

2.2.1.1 Service Discovery

Service discovery is concerned with how a service consumer discovers a service deployed on the Grid. Grid services descriptions typically have a lot of metadata including functional (operation and data) and non-functional (security, privacy policies, etc) descriptions. The metadata is also likely to be updated frequently. Service discovery implementations include centralised registries and P2P techniques. Current standard service descriptions have focused on syntactic issues, which in most cases still require human readable specifications for service integration. To improve interoperability semantics are needed to allow software to understand the meaning of data and a service’s function allowing improved service discovery, automated orchestration and mediation. Existing registries such as UDDI 1.0 [36] are not well suited to the grid as the data model is restrictive, it is usually centralised and does not support dynamics well.

2.2.1.2 Notification

Notification is concerned with providing a mechanism for services to notifying other distributed components that an event has occurred. Grid deployments need to operate across organisation boundaries where firewall restrictions callbacks from an outside organisation difficult. Notification is especially important for interactive services.

2.2.2 Virtual Organisation (VO) Management

VO management is concerned with providing infrastructure capabilities that allow individuals and organisations to collaborate and share resources within the constraints of their business needs. VO management allows individuals within organisations to assign and control access to resources to trusted individuals. VO structures and dynamics vary depending the business, for example, B2B service provision requires a fast VO where client sets up a trusted relationship with a service provider close to the time when the Job is executed. This is contrary to a team of engineers within different organisations agreeing to collaborate over a year in the development of a complex product.

2.2.2.1 Business Processes

Business processes are concerned with providing a conversation model between organisations that allows users such as engineers and scientists to collaborate for some agreed objective. Requirements for Grid infrastructure tend to focus on the analysis process executed by the application users rather the constraints under which the resources are shared. Typically, grid infrastructures provide services that support an implicit business processes.

2.2.2.2 Authentication

Authentication is concerned with identifying individuals and organisations using security credentials. Various mechanisms exist for representing security credentials within trust domains such as X509 certificates, Kerberos Tokens and basic username/password. The heterogeneity of the

Grid means that individuals may access resources located within different trust domains where identity mapping may be required between different credential types.

2.2.2.3 Federation

Federation is concerned with distributing trust agreements among decentralized security and policy domains. Federation lets access-management functions span diverse organizations, platforms and applications. Federation requires that an organization trust each collaborating organisation to authenticate its own users' identities. In a federated environment, a user can log on to their home domain and access resources transparently in external domains subject to various policies defined by home and external administrators.

2.2.2.4 Confidentiality

Confidentiality is concerned with ensuring that data remains private and not disclosed to unauthorised individuals intentionally or unintentionally. Two main security mechanism exist asymmetric and symmetric encryption.

2.2.2.5 Integrity

Integrity is allows a recipient to verify that data has not been changed during transmission. Integrity also provides non-repudiation stopping a sender from denying that the data was sent.

2.2.2.6 Authorisation

Authorisation is concerned with deciding which individuals have the necessary rights to access to resources. Access control decisions are based on policies that can be implemented in a variety of ways. For example, an authorisation policy could simply list a set of roles or could be more complex incorporating application context such as the ability to pay, a valid license, state of a conversation, etc.

2.2.2.7 Delegation

Delegation is concerned with delegating access rights from individuals to services or other individuals. Delegated rights are usually for a limited lifetime to avoid misuse, for example within the scope of a specific job or account. Models of delegation differ between grid infrastructures such as proxy certificates or direct updates to access control lists.

2.2.2.8 Accounting

Accounting is concerned with tracking the usage of resources and providing this information to higher level services such as auditing, authorisation, scheduling, load balancing, etc.

2.2.2.9 Auditing

Auditing is concerned with producing a record of security related events to allow an administrator to determine if a VO is adhering to required authorisation and authentication policies

2.2.3 Resource Management

Resource management is concerned with the management of different types of resources available on the Grid. Resources management operates at different levels depending on whether the resource is physical (machines, storage systems), logical (applications, metadata) or conceptual (projects, organisations). Resource management is also concerned with managing the grid infrastructure, which is important for providing Service Level Agreements.

Resource management requires a resource model that describes the different resources accessed on the grid for functions such as assigning resource, managing execution and monitoring tasks. Defining a standard resource model for the multitude of available resources that can be understood by resource management services is a significant challenge. Existing standards such as CIM schema are starting points for creating resource models, however, most existing grid infrastructures are only just working towards this.

3 Application Requirements

This section describes the SIMDAT application sector requirements on Grid infrastructure. For each application activity a description of how Grid infrastructure will improve collaborative working is provided.

3.1 Demonstrators

In the first 12 months application activities will develop demonstrators that will explore how Grid technologies can improve enterprise and inter-enterprise collaborating working.

- **Aero:** VO for collaborative multi-disciplinary (aeroacoustics, structural, aerodynamics) aerospace design
- **Auto**
  - *Distributed Product Development 1 (Audi)*: federation scenario showcasing
    - MSC.SimManager Federation
    - Data extraction from two MSC.SimManager data sources for data mining usage
    - Ontological integration of CAE and CAT databases
  - *Distributed Product Development 2 (Renault/ESI/IDEStyle)*: VO for collaborative car design supporting confidentiality constraints of components between organisations
  - *Meta-scheduling (LMS)*: Optimus/GRIA integration for meta-scheduling on the grid
- **Meteo:** Meteorology portal that provides access to virtual meteorology database with supporting VO for access control
- **Pharma:** SRS federation
- **Data Mining:** Technology demonstrator showing how data mining applications (WEKA) can be deployed within Grid infrastructure
- **Ontological Integration:** Technology demonstrator showing how Ontobroker [32] can be used to integrate local disparate data sources

3.2 Aerospace

The aerospace industry deals with highly complex products that have data creation, management and curation requirements that span hundreds of collaborating organisations over a 50-year lifecycle. Partners on a product team need to collectively manage thousands of inter-related processes and this leads to expend considerable time and effort in the access, transmission, control, translation and sharing of data. The aerospace sector will develop and deploy existing and emerging Grid technologies and concepts to improve collaborative engineering of sophisticated products.

The development of aerospace products requires the collaboration of various engineering disciplines such as aeroacoustics and aerodynamics that are increasingly distributed within different organisations. Each discipline relies on a variety of commercial and bespoke in-house PSE's and analysis tools to help solve complex design problems. The design models are complex and simulations are computationally expensive often executed on computational clusters accessed through batch submission systems such as PBS, Condor, etc. When designing a product engineers develop a dataflow that incorporates various analysis applications. The dataflow is used to explore a design space and hence find the optimum design solution. Typically today's engineers codify these dataflows using scripts that read/write/translate data and execute local applications. Results are stored on the file systems and managed through a manual procedure.

A problem with this approach is that intra-enterprise collaboration is difficult and inter-enterprise collaboration is almost impossible within the constraints of most businesses. Collaborative engineering requires organisations and individuals to share resources within the constraints of their business. For example, if a script needs to execute an application on a machine provided by another organisation remote access has to be configured. This is difficult to maintain and usually involves systems administrators who may not be the right person to authorise access to the resource because they do not understand the business needs. If the application is not directly accessible collaboration across organisation boundaries has to be achieved using out-of-band distribution of data files.

Other problems with aerospace dataflows include data formats and management. A typical dataflow may contain a diverse set of both proprietary and standard data formats. In some cases, engineers develop analysis codes that read and write data files in proprietary formats. Connecting two codes together requires both in-depth knowledge of the file formats and development of bespoke conversion utilities that are inherently difficult to reuse.

The aerospace sector is looking to define new business models that will demonstrate how engineers can collaborate more effectively across organisation boundaries. The initial scenario will simulate the multi-disciplinary collaborative configuration design of a low-noise, high-lift landing system. The scenario is typical of sub-system design problems in the context of, say, future-concept, unmanned cargo vehicles that require an ability to use airfields in noise-sensitive locations. The scenario is one use-case selected from many possible alternatives in the product lifecycle that will be used as a “model problem” to drive the development and deployment of Grid technology.

A project manager working for a prime contractor initiates the business process by assembling a project team with the required core competencies to solve the design problem. The project manager identifies engineers directly with the organisation responsible for co-ordinating the design and searches for service providers with an appropriate trust and Quality of Service credentials. The service providers advertise services and respect negotiated quality of service.

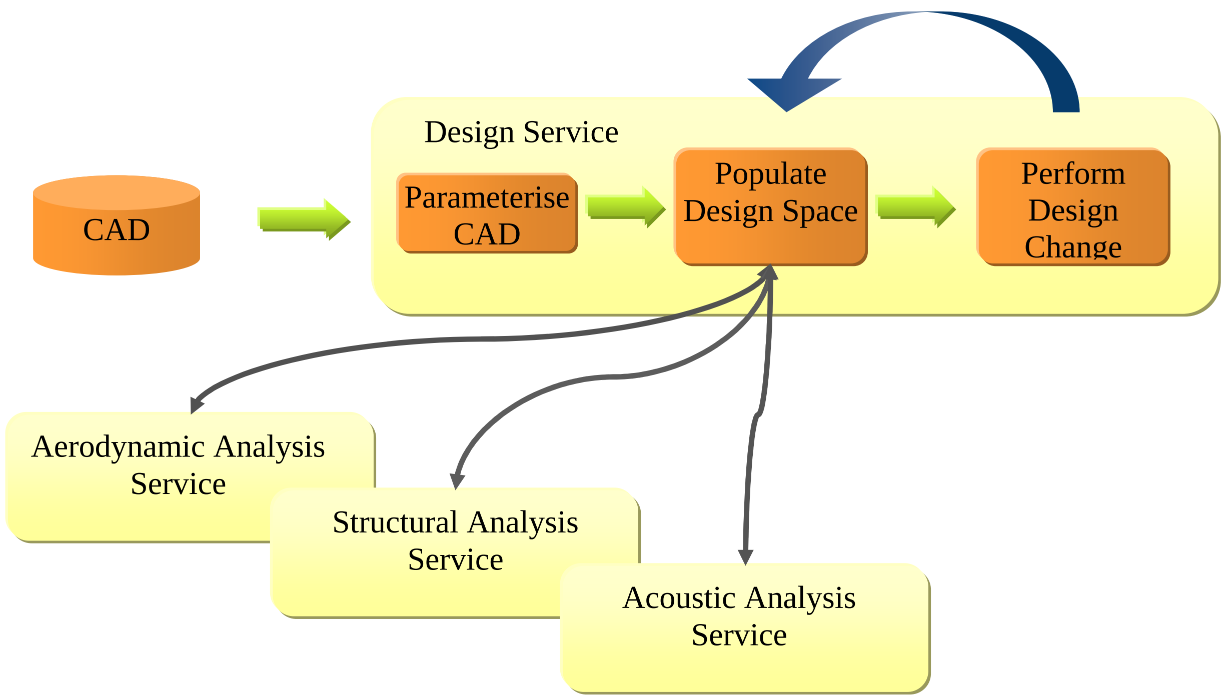


Figure 4: Aerospace application scenario

The objective of the initial aerospace application scenario is to develop a multi-disciplinary optimization workflow that can be executed across boundaries of organisations within the constraints of an agreed business process. The project team will consists of engineers directly working for the prime contractor and then set of service providers that specialise in aeroacoustics, structures and aerodynamics. The focus will be on virtual organisations and the analysis process (services and workflow).

The aerospace activity has concluded that GRIA can provide almost all the necessary grid infrastructure functionality for this development. The explicit business process and security mechanisms are well aligned with the needs to the aerospace virtual organisation. GRIA's simple file-compute Grids is also sufficient for the proposed data requirements.

3.3 Automotive

Within the automotive application activity the architectural vision is to start with tightly coupled propriety systems and progress to a loosely coupled web services Grid architecture based on open standards. This should be done to move from application centric (MSC NASTRAN / ESI PAMCrash) to problem centric analysis services (crash/meshing/assembly). Therefore there is a need to switch from isolated discipline specific databases (CAE/CAT) to transparent secure inter-organisation data access and integration.

Based on the two main stakeholders in this application activity – AUDI/SEAT on one hand and RENAULT on the other hand – there are two basic initial scenarios. The group centred around AUDI is interested in enabling intra-organisation multi-disciplinary simulation (CAE/CAT; crash/NVH) based on MSC.SimManager and MSC NASTRAN. The group centred around Renault is looking into improving the interaction between manufacturer and supplier to evolve a collaborative CAE simulation outsourcing scenario based on ESI PAMCrash.

Regarding the software environments at the end of the project, i.e. after 48 months, the following is planned:

Managing distributed simulation data is a key component of SIMDAT. Within the SIMDAT environment, there is not only data to be consumed and disseminated, but also the analysis services produce a substantial amount of data, which has to be persistent in SIMDAT. As part of persisting the data, its full pedigree needs to be stored, for example in meta databases. The foundation of the project is the assumption that AUDI will use individual implementations at various engineering sites in a coordinated way (meaning with a number of homogeneous assumptions). The implementation will consist in applying information Grid technology to allow analysts in a given location from their MSC.SimManager based workbench (aka CAEBench) to access and act on objects that are stored and initially managed on another location as well as combined objects of the two locations in local engineering actions.

Crash compatibility is a highly involved task. Physical testing of two cars crashing into each other is only possible in special crash test facilities, which allow the simultaneous acceleration of two cars. Such facilities are typically not available at car manufacturers themselves. In addition, there is an increasing demand for performing the so called compatibility crashes at different impact angles, originating from the USA. Using the results of SIMDAT, the goal is to be able to conduct compatibility crash simulations between cars from different car manufacturers over the SIMDAT Grid infrastructure. The Grid security technologies deployed within SIMDAT will allow each car manufacturer to see only the results of his car. The envisioned coupling of CAD, CAE and CAT data will enable the car manufacturer to more quickly identify and resolve any potential design shortcomings discovered in these Grid enabled compatibility crash simulations.

The other automotive application activity Renault/IDEStyle/ESI is concerned with a scenario regarding the interaction between the original equipment manufacturer (Renault) and a supplier (IDEStyle) to evolve a collaborative crash simulation outsourcing scenario based on ESI PAMCrash. The original manufacturer outsources a certain part of the design of a car to a supplier while preserving control and knowledge of the whole vehicle's design for himself. Therefore only the part to be developed by the supplier and its immediate environment must be known to the supplier while on the other hand remaining secret to the original manufacturer. Work on the model as a whole is given to a trusted third party for calculation. Results on the outsourced parts are given back only to the supplier while results on the rest of the car are given only to the original manufacturer. Independently from the scenario the manufacturer and the supplier negotiate for the full disclosure of the data afterwards.

The data access prototype will show the interoperability of two simulation data management systems at different locations. Grid technology will be used to enable this. For the 12-month demonstrator it is assumed, that each of the sites for the distributed product development will use MSC.SimManager for the testing and improvement of the functional behaviour of car designs. Grid technology is used to federate MSC.SimManager and provide access to the distributed underlying databases. A first step into this federation is subject of the demonstrator for the automotive activity.

Access to distributed databases and distributed data vaults is precondition for the 48 months demonstrator. The 12 months demonstrator will show the comparison of two car crashes whose data is stored at different sites. Furthermore there is a need to have the possibility to create additional post-processing objects (PPOs) on both sides for deeper investigation of the car crashes. On every side the car projects of the other side should “feel” like the local car projects in the navigation frame of MSC.SimManager.

In the design phase there is the opportunity to visit various approaches to solve the distributed simulation data management challenge. From a pure web services based external information broker agent to a middleware provided federated database solution, all will be challenged and individually evaluated to identify possibilities and dependencies brought to the project. Finally, after considering key features and probable necessary adaptations of MSC.SimManager attached to each alternative, the best solution will be chosen.

For this demonstrator a comparison report with data on different sites is to be generated. The demonstrator will use the hardware and software environment, which will be set up by MSC for the Grid infrastructure demonstrator. For this purpose AUDI has generated two geometric variants of the SAMD car and will perform a number of crash simulations and compatibility crash between the two models. The data access demonstrator is performed using crash simulations on the SAMD car version 2 (coupe) and version 3 (cabriolet) which Audi provides. Model data will be made available on the two reference installations of MSC.SimManager at MSC. Grid technology will be used to fetch the crash evaluation data from the different reference installations.

The second demonstrator is consistent with the long-term project goal. Initially only a light demonstrator will be set up. This demonstrator will then evolve in requirements and functionality. The OEM (Renault) uploads data (requirements, CAD, meshes, material law…) for the supplier (IDEStyle). IDEStyle downloads Renault data. IDEStyle does the job with the data (pre processing of PAMCRASH model, calculation, post processing). Renault follows up the tasks and consults the draft deliverables of IDEStyle during virtual project meetings (meshes, PAMCRASH model, analysis). After validation by IDEStyle and Renault, data is transferred to Renault.

The Grid infrastructure needs to have a clearly defined interface to analysis services. Batch as well as interactive analysis services will be run on the system. Services for resource management and job scheduling are mandatory.

For distributed data repositories access and integration a robust and efficient data transport to and from analysis services, including large data volumes (average about two Gbytes per simulation, six

Tbytes per year) must be present. It must be possible to separate metadata and data. A standardisation and federation at conceptual level (door, dashboard, etc) is needed. Very important for the end user is the audit trail preservation.

From a security and privacy point of view access control for data and services, secure data exchange between collaborating organisations, and IPR protection for component suppliers in collaborative crash simulation is mandatory.

Initially the automotive application scenarios require a pragmatic approach. Thereby the delivery of the 12-month prototypes is ensured. Grid infrastructure needed and distributed data repository access software should by and large be available today with only small changes called for. There is a need to start prototyping as soon as possible.

GRIA provides a good starting point. There is support for batch integration of file compute analysis services as well as trust relationships for OEM/Supplier collaborations. The Grid software is based on WS-I standards and integrates fairly easy with OGSA-DAI [31].

OGSA-DAI provides a good starting point. OGSA-DAI WS-I will be used in the project. Thereby the efficient access to large data repositories is enabled. OGSA-DAI supports distributed 'same schema' queries for CAE/CAT integration using minimum common data set.

Consortium partners will provide support and intellectual engagement to partners within the project for both GRIA and OGSA-DAI.

3.4 Meteorology

In 2003 the World Meteorological Organization (WMO) approved the concept of Future WMO Information System (FWIS). The FWIS will provide a single coordinated global infrastructure for the collection and sharing of information in support of all WMO and related international programmes.

WMO has defined a virtual structure for the FWIS, which contains three main actors:

- National Centres (NC)
- Global Information System Centres (GISC)
- Data Collection and Production Centres (DCPC)

As a first step towards the establishment of FWIS, the national weather services of France, Germany and the United Kingdom have volunteered to jointly implement a *Virtual* GISC (V-GISC). The three V-GISC partners form a cluster and enjoy equal rights and mutually support one another. The European Centre for Medium-Range Weather Forecast (ECMWF) and the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT) are part of the project as DCPCs.

The V-GISC is a distributed database that will provide users with transparent access to datasets located at Météo-France, DWD, the UK MetOffice, ECMWF and EUMETSAT. The V-GISC concept is being developed within the meteorology activity of the SIMDAT project.

Some key elements of the project are:

- Improve visibility of and access to data through a comprehensive discovery service based on metadata development,

- Add value to existing datasets by enabling diverse databases to be used as a unique virtual resource,
- Offer a variety of reliable delivery services,
- Provide a global access control policy managed by the partners and integrated into their existing security infrastructure.

A user can use the V-GISC according to one of the five global use cases:

- Access data and metadata
- Provide data and metadata
- Manage VO
- Manage V-GISC infrastructure (data communication infrastructure - DCI)
- Monitor and control the V-GISC

The DCI is the backbone of the V-GISC. It is the software infrastructure developed to provide the V-GISC services. The DCI consists of several nodes hosting metadata and data.

The goal of the meteorology application activity is to generate a meteorology portal that provides access to a virtual meteorology database with supporting VO for access control.

Data reside and are managed by each partner. The three main partners (Météo-France, UK Met Office, DWD) will be closely connected and will be seen as a unique entity offering a collection of datasets to the users. To offer this unified view, the distributed database must be easily interfaced with the actual systems (flat file repository, meteorological database, relational databases). It should also be enough flexible to easily add a new database containing new datasets. For resilience and performance reasons the metadata will be synchronized between the partners and part of the data (the real-time datasets) is replicated on at least two sites. For example if the DWD site database is down the Météo-France site database or/and the UK Met Office site database must be able to deliver the real-time data to DWD users. The replication is configurable and is managed by each partner (can be activated or deactivated). The distributed database will manipulate two different kinds of datasets:

Real-time data: This can be the observation data, model outputs, post-process data, time critical products or warnings. The real-time term is applicable to the data only within 24-48 hours after the data birth date. After that, the data is treated as a non-real time data. These datasets are produced several times within a day and should be distributed within one hour. The average size of daily observations dataset is 58 Megabytes. Currently more and more satellite data is produced and used within the meteorological community and the quantity of daily real-time data is growing rapidly (2 GB produced daily).

The V-GISC will develop performance targets with respect to for example internal reliability and timeliness of data exchange with its neighbours depending on the types of data being exchanged. These targets will be published as part of the operating specifications of the V-GISC.

Non real-time data: This is the data contained within the meteorological archive. These data are usually stored on tapes and implies asynchronous retrievals. For example ECMWF runs an archive called MARS. So far 1 Petabyte has been archived and is accessible to ECMWF users. The Data Communication Infrastructure must be able to handle a dataset of few Terabytes but not in a time-critical request.

Metadata are a critical component of the data communication infrastructure. Metadata are required for the discovery, browsing and access. The internal DCPCs will use this infrastructure to update the V-GISC catalogue. The metadata updates will be synchronized among the partners.

The catalogue will be accessible to anybody and a discovery service will be implemented. A query interface will be offered to request sets of data. A subscription service to the datasets will be also implemented. The users will subscribe to different data and will receive it daily when available.

Quality of service mechanisms will be implemented. The cost of a request will be estimated in term of resources necessary to offer the service. The request will be then placed in a queue and the position in the queue will depend on parameter such as the cost of the request, the user's priority. Prioritization mechanisms will be implemented in order to deliver data, such as warning messages, as soon as they are received by the V-GISC.

The data communication infrastructure DCI has to be built to federate all the partners' data repositories. Among the main challenges that have to be solved in order to build the DCI is the implementation of a virtual database providing the following services:

- Create a unified view of all the shared datasets through a distributed catalogue.
- Define a metadata format containing information to locate and identify the data, to describe the data access policy and to describe the available meteorological data for discovery.
- Maintain the distributed catalogue amongst the partners using synchronization mechanisms
- Give access to the legacy meteorological databases
- Implement data replication and cache mechanisms
- Preserve the data integrity

Another challenge is the implementation of data access services:

- Collection and dissemination services that support various efficient and reliable transport mechanisms
- Quality of service (QoS): traffic prioritization, queuing mechanisms
- Discovery service by browsing a hierarchical catalogue or using a keyword search engine
- Interactive interface authorizing humans to easily access the data
- Batch interface authorizing programs to easily access the data

The objectives of the meteorology demonstrator are 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 demonstrator will focus on virtual organisations and data access. A virtual organisation will be created that represents the V-GISC collaboration with VO level services for authorisation and authentication and resource management. Users will access a virtual distributed data catalogue through a portal at each participating organisation that allows the discovery, retrieval and analysis of meteorology data.

The meteorology activity has decided to implement the each VGSIC node and associated business processes using J2EE accessing authentication and authorisation services provided by AuthN and AuthZ respectively [2]. The activity will evaluate OGSA-DAI as a technology for their virtual database and will monitor WSRF developments.

3.5 Pharmaceuticals

In the last decades, the advances in the life science sector have facilitated the rapid acquisition of vast amounts of data e.g. in the diverse genome sequencing projects or in high throughput screening of compounds against drug targets. Academic and industrial researchers in the life sciences community use this data for various experiments. These experiments allow the scientist to investigate or verify a hypothesis that they may have about a particular problem or domain. Such *in-silico*[2] experiments are, by their very nature, hypothesis driven, ad-hoc and highly specialised to the particular problem they are associated with. For example, in medicine, sequences provide a basis for the study of susceptibility to disease and the development of new preventative and therapeutic approaches where as, in cell biology, the interactions between components of cellular circuitry can be studied. The pharmaceuticals activity will deploy and use Grid technology to provide added value to existing data integration technology in supporting collaborative bioinformatics experiments.

Life science research is supported by a collaboration of research institutions called EMBNet that provide national scientific communities throughout the world with access to high performance computing resources, specialised databases and up-to-date software. Each research institution within EMBNet is known as an EMBNode and is responsible for maintaining a set PSE's, analysis tools and bioinformatics databases such as (EMBL, SWISS-PROT, etc). There are approximately 1200 different data providers and each EMBNode maintains a subset of the overall data. New data is published to the community when researchers report their results through academic papers. It is a precondition for paper publication that the experimental results are available in the public domain. Researchers can access EMBNet resources by providing a nominal yearly fee that gives access to all data and analysis services along with a disk quota that can be extended on request.

In industry, researchers also use public data providers but augment this information with proprietary data generated within their organisation. Pharmaceutical companies do not typically access EMBNet directly but maintain in-house databases due to confidentiality constraints. Even the knowledge of the types of queries being performed is commercially sensitive information as it can give competitors information about current drug targets.

The cost of distributing and maintaining databases produced by public data providers is a significant problem for both EMBNodes and pharmaceutical companies. Bioinformatics data is generated at an incredible rate and databases can by updated on a daily basis. Up-to-date data is important to researchers, as additional sequence data can significantly change the results of some analysis. Organisations typically schedule database updates according to their business needs, for example, daily or bi-weekly. The maintenance of databases requires organisations to manually monitor data providers for new database releases and acquire the release either by direct download or CD. Updates can be computationally expensive, for example, updating an EMBL database (>400G Bytes) within a SRS server can take days to calculate the necessary indexes (obviously depending upon the target platform).

Data providers distribute data in a variety of standard (XML, ASN .1) and proprietary data formats (FASTA, GenBank, SwissPrott). Researchers execute cross-database queries using tools like SRS that provides integration of these diverse and complex data structures. At present, to do a cross-database query SRS requires all data and indexes to be co-located. The pharmaceutical activity is looking to federate SRS so that distributed cross-database queries can be supported reducing the need to support an entire set of databases at a single server.

Researchers use a wide variety of domain specific PSE's and analysis tools to support their experiments such as SRS for cross-database sequence similarity searching. Some analysis tools such as those adopting brute force algorithms can be computationally expensive lasting many hours. Researchers traditionally chain together database searches and analytical tools, using complex scripts to overcome incompatibilities in data formats, or by manually cutting and pasting between web interfaces. These *in silico* experiments are usually undertaken without support for the scientific process of managing, sharing and reusing the results, their provenance, and the methods used to

[2] In silico experiments are procedures using computer based information repositories and computational analysis adopted for testing hypotheses or to demonstrate known facts

generate them. Management of the scientific process is important should the data be judged erroneous at a later date. The researcher can then determine the results that need to be ignored or experiments that should be rescheduled. Currently, there are no standards for database description metadata and versioning information. Each data provider adopts its own approach with some providers' not even publishing version information; these have to be derived from the date of download. For multiple distributed SRS servers containing replica databases it is important that database versions are synchronized. The database lifecycle is currently managed manually or by a technology called PRISMA.

The pharmaceuticals activity is looking to develop new business models for carrying out bioinformatics research that will enable collaborative experiments within the commercial and academic communities that supports a managed scientific process. These business models are in the early stages of development and will be further defined during the development of the first demonstrator. An example possible business model could be based on a project team assembled by a pharmaceutical company for the development of a new drug. The team could consist of a set of trusted data providers and analysis service providers along specialist scientific teams within the company. There may be situations when smaller organisations could provide specific areas of expertise in the development process and could join the virtual organisation. Researchers define *in silico* experiments as workflows that querying data sources and executing analysis services. When a researcher executes a workflow the infrastructure manages generated data and derived knowledge along with provenance about the experiment allowing other scientists to interpret the experiments context. As the scientists perform experiments results are generated that should shared with authorised team members. Periodically, when results are seen to provide value beyond the project team they are validated and published the wider community either within a proprietary company database or public database.

The objective of the initial pharmaceuticals demonstrator is to develop a federated version of SRS. The focus will be to solving the public bioinformatics data distribution and synchronization problem faced by organisations within EMBNet running SRS servers. The demonstrator may also look at providing bioinformatics analysis tools as services that are invoked from SRS. Some aspects of VO's will be addressed such as certification authorities and message security but business models and wider aspects of virtual organisations to support collaborative resource sharing will not be the focus. The pharmaceuticals activity has decided to base the implementation on web services and will integrate NEC's E2E framework providing authentication of individuals using X509 credentials along confidentiality and integrity of messages.

4 Technology Requirements

This section describes the requirements of SIMDAT technology activities on Grid infrastructure. For each technology activity a general discussion of issues relating to the integration of the technology layer with Grid infrastructure is given in the context of the application scenarios. This section does not try to identify specific requirements as with section 3 but tries to identify gaps in the existing technology and potential longer term solutions currently being proposed within the Grid and web service communities.

4.1.1 Virtual Organisations

The administration of virtual organisations (VO) is a fundamental principle driving all Grid infrastructure technology. Grid developments have been driven by the need for organisations to collaborate and share resources for some common purpose. Grid infrastructure implementations have developed to support different types of virtual organisations each exhibiting specific characteristics and implicit business models. Therefore, the means by which a Grid infrastructure supports virtual organisations is a key decision when selecting an appropriate technology.

In traditional Grids such as those described by the pharma EMBNet and the meteorology application sectors the concept of the virtual organisation plays an important role. The VO becomes a tangible manifestation of the collaboration, representing and facilitating a common purpose across the collaboration. The VO is persistent, resourceful, and may develop characteristics of a real organisation, such as logically centralised administration and management structures. The formation of the VO focuses on defining rules of membership and operation along with assigning resources for VO-level services to operate on. The management of such a VO involves keeping track of all its members including authentication credentials, roles and access rights. It also involves keeping track of all the computational resources available to the VO through a logically centralised registry, accessing them through portals or job submission services. Grid technologies such as EGEE supporting a centralised VO model suited for academic Grids. EGEE provides centralised services for VO administration (VOMS) and resource management (GAS). There is a user accounting service appropriate to this model, but billing is still undefined, and presumably will initially target user billing (or quotas) at VO level.

In business-to-business Grids, such as those described by aerospace, automotive and pharma B2B scenario a different style of VO is required. For industrial collaboration VO participants require management of their own resources according to their own interests rather than centralised VO management. The VO must support transient B2B federation of resources under user control to perform specific projects or applications. For example, in the aerospace sector the assembly of a project team requires the discovery of service providers for the design of complex products. There could be circumstances when a new service provider is needed to execute an important job, where they would be discovered, would join and then leave the VO in a short period of time. In this case, business relationships may be short lived and terminate rapidly with little or no prior infrastructure apart from the ability to discover service providers. Grid technologies such as GRIA provide a decentralised P2P VO model more appropriate to B2B Grids. Each organisation participating in a collaborative relationship can assign and control access to resources they want to share.

The support for VO's that enforce policy-driven business processes would add much flexibility to Grid infrastructure. This would allow industry define business models that meet their needs rather than adopting those supported implicitly by a specific infrastructure. This is described further in section 4.1.2.1 Business processes.

4.1.2 Workflow

Workflow is concerned with providing a mechanism that allows users and developers to connect services together to create higher-level business and analysis processes. In the web services community, the development of open standard workflow languages such as BPEL4WS, WSCI, and BPML have tried to reduce the complexity required to orchestrate web services and increase interoperability, which cannot easily be achieved using proprietary business protocols. In the Grid community, the concept of workflow for service composition has received little priority within the design and development of current infrastructure and is not even discussed within the current OGSA specification. Workflow is only just beginning to be recognised as the key technology that will enable user and applications to effectively and efficiently program the Grid.

The integration of web service workflow standards with Grid infrastructure is however non-trivial. The Grid requires long-lived interactions and inter-site sharing of resources, capabilities and even know-how to support collaborative applications. Grid infrastructures have developed to support these requirements by providing a programming model based on stateful resources that are managed using resource and execution management services that are accessed within a secure context. Although, web services may also be concerned with these issues the programming model is based, as one might expect on web service standards which to not currently address these concerns. For example, BPEL4WS workflow enactors have no understanding of how resources can be modelled using WS-Resource and referenced using WS-Addressing as proposed by WSRF. There are many other issues such as security, federation and optimization that are related to the integration of workflow and Grid infrastructure that are likely to be research topics over the next few years. In the short term, it is likely that specific workflow engines will be deployed on specific grid infrastructures as standards such as WSRF and BPEL4WS evolve and converge. The following sections discuss the significant integration issues facing workflow and grid infrastructure.

4.1.2.1 *Business processes*

Virtual organisations are based business models and processes that allow collaborative resource sharing. The Grid provides the infrastructure that enforces a business processes. For example, authorising access to a service only if a user has the ability to pay and a resource allocation with an agreed QoS.

The business process is additional to that created for the analysis workflow and related to assigning and accessing resources. Figure 5 shows the relationship between a business and analysis processes for business-to-business service provision. The user defines a workflow for the analysis process and wants to out-source the FEM solver to a 3rd party service provider. The business process allows the user to discover a service provider, create accounts and negotiate resource allocations with an acceptable quality of service. The user can then execute the analysis workflow and on completion the service provider can bill the user's account.

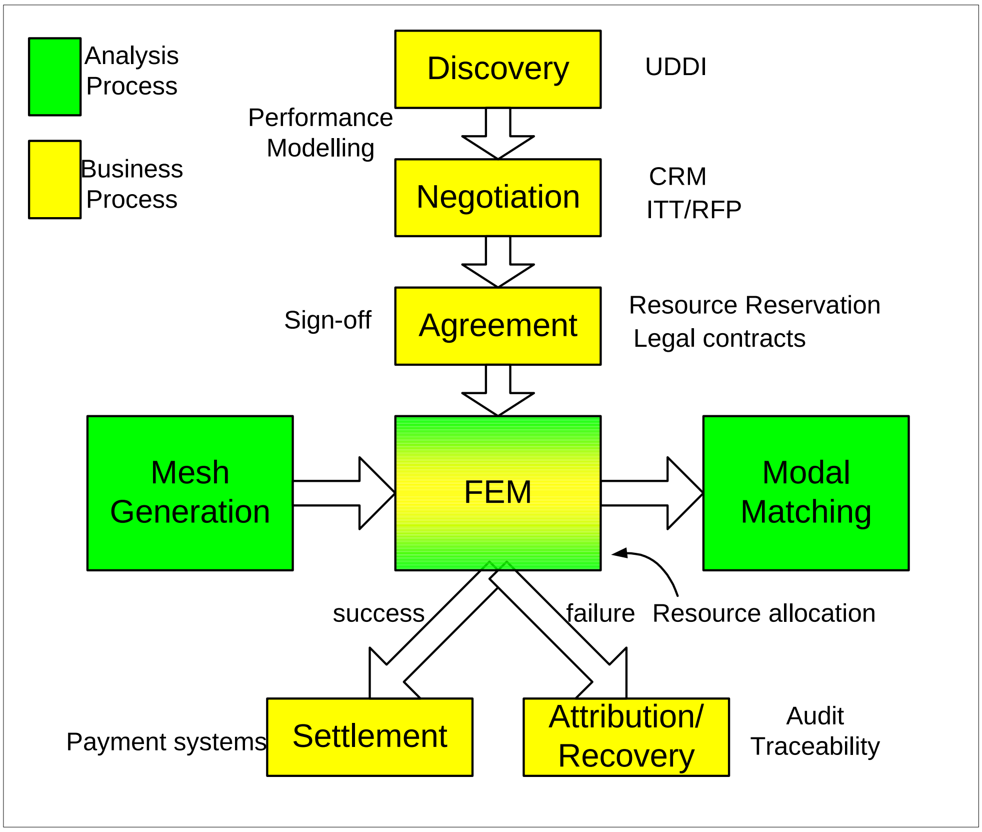


Figure 5: Workflow and business processes

In most scenarios, binding Grid resources to abstract workflows can be done in advance of executing the analysis workflow. The service broker described in 4.1.3 would need to execute business process workflows when selecting appropriate services. There are dynamic issues in the business process that need to be considered such as service providers having different negotiation protocols and resource models depending upon their business needs. In the aerospace and automotive scenarios, we can see two different business models for agreement and settlement. EADS wants to provide services to different customers with dynamic agreement and settlement whereby IDEStyle tend to have an out-of-band agreement and cost for a specific project design cycle. However, as discussed in 4.1.1 current grid infrastructure tends to support a fixed implicit business model and does not provide capabilities that allow VO participants to define flexible business processes.

4.1.2.2 Architecture and Federation

Architecture is concerned with the structure, relationship between services and data and behaviour of distributed Grid applications. Workflow is crucial for architecture as a means for users to compose services into larger applications. Workflow technology currently support structure using hierarchical workflows but do not currently considers the security implications of services operated in different security domains.

The architectural considerations for workflow and grid infrastructure are important for all application sectors. In the aerospace scenario, the workflow is hierarchical with the design workflow calling aerodynamics, structures, and aeroacoustics workflows provided as services from BAE, Boeing and EADS respectively. In automotive and pharmaceuticals the federation of existing software such as MSC CAE Bench and Lion's SRS require specific infrastructure capabilities.

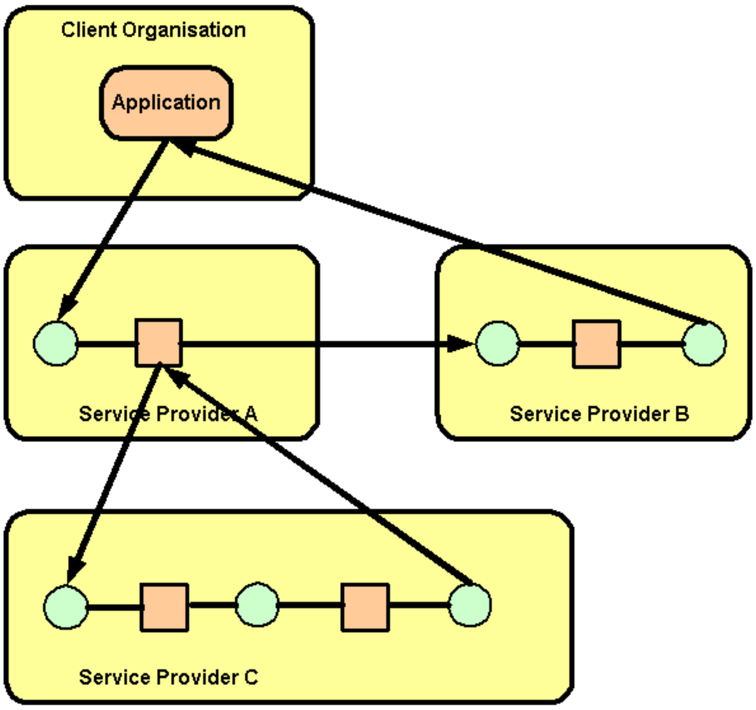


Figure 6: Workflow and architecture

Figure 6 shows how workflow can be used to define the architecture of a Grid application over four organisations. The client application uses services from A and B with the underlying implementation of A using services provided by C. Firstly, to support scenarios as shown in Figure 6, workflow enactors such as Inforsense and Freefluo must be deployed as services on the Grid rather than just as clients to Grid services.

Secondly, the client organisation must be able to authorise service provider A to write data to the data store at service provider B. The grid infrastructure must support a delegation model that allows services to act on behalf of the originator. Grid infrastructure technologies provide different delegation models for this purpose including Globus's proxy certificates and GRIA's process based access control. Figure 6 also shows service provider A accessing services provided by service provider B. One might think it acceptable for the client organisation to authenticate with A and A to authenticate with C. This is how most enterprise applications operate with service-side credentials accessing databases or other services. However, for service-oriented architectures with inter-enterprise collaboration a service must act on behalf of the client when accessing other resources. In our example, the client organisations must tell C what A can do on their behalf.

Implementing delegation at a workflow level is an active research topic. When defining a workflow a user should be able compose applications from services and data sources provided by different organisations. These workflows will have implicit delegation requirements depending upon how the services and data are located. There a few architectural choices that need to be discussed about how delegation models can be incorporated into workflow technologies:

- The workflow composition tool allows users to manually delegate. This would require tasks to be bound to grid resources.
- The workflow language is extended to support delegation.
- An intermediate scheduling component could process the workflow, bind to grid resources, delegate as necessary

Finally, the identity federation is important for services operating in different security domains. Standards such as WS-Federation are investigating how to support identify federation, however, this remains an early research topic and is not currently supported by Grid infrastructures. In SIMDAT, the early demonstrators will focus on services operating within a single security domain where the root of trust is an agreed certificate authority.

4.1.2.3 Notification

Notification is concerned with providing a mechanism for services to notifying other distributed components that an event has occurred. Notification is essential to support SIMDAT scenarios such as computational steering workflows in the aerospace and users notification for long running jobs. Existing Grid infrastructure does not support notification but rely on polling mechanisms, however, standards such as WS-BaseNotification and WS-Eventing should soon be adopted.

Figure 7 shows a workflow from the pharma sector workflow where a scientist wants to monitor and steer a parameterised simulation. The infrastructure should provide the client application with regular notifications of the simulation progress so it can be monitored and even displayed to the scientist using an appropriate tool. The user should be able then change workflow parameters and workflow enactor and grid infrastructure should reschedule relevant services.

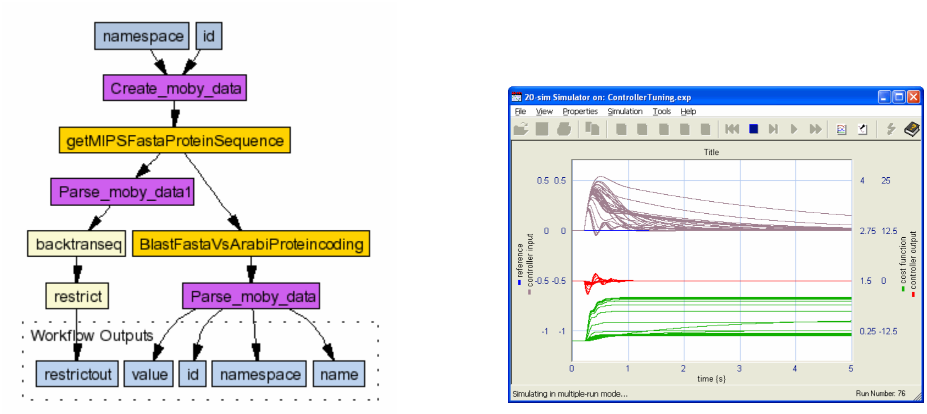


Figure 7: Workflow and notification

4.1.2.4 Meta-scheduling

Meta-scheduling is concerned with scheduling analysis tasks within workflows over different organisations based on metadata provided by the infrastructure that describes the resource model (systems, networks, applications and services) along with relationships between the resources. A meta-scheduler provides a consistent interface for users into the scheduling system for a grid coordinating communications between multiple heterogeneous schedulers that operate at the local or cluster level.

A meta-scheduler must take into account the distributed nature of Grid applications. Figure 8 and Figure 9 shows a simple scenario how the location of data components can affect the execution time. In Figure 8, the service running at A writes output data to a data stager co-located at A. When A completes the services running at B must pull the data twice across the network. However, if the data stager was located at B rather than A, as shown in Figure 9, the data only has to be distributed once. Now clearly we have parallel tasks at B so the execution time may not be increased but there would certainly be an increased network traffic for very large data sets. This would certainly be an issue for low bandwidth network connections.

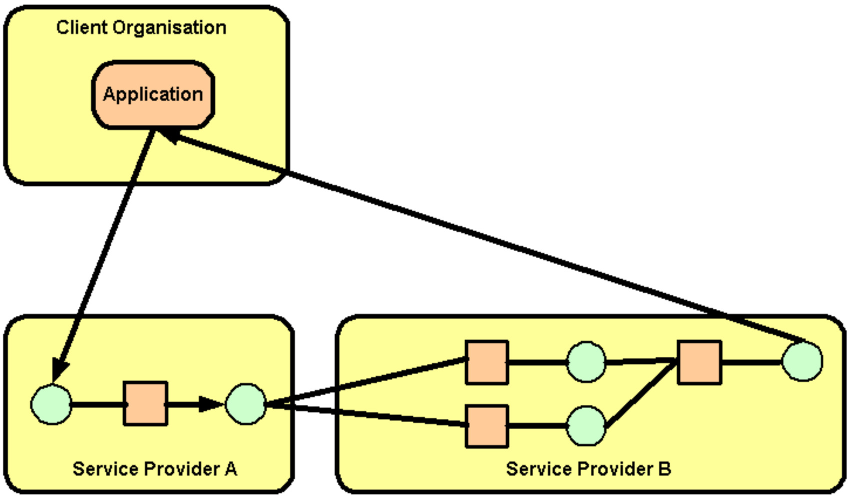


Figure 8: Workflow and optimization

Software such as LMS Optimus provides meta-scheduling over existing resource management systems such as Condor, PBS, LSF and will be further developed in SIMDAT to provide optimization capabilities on GRIA.

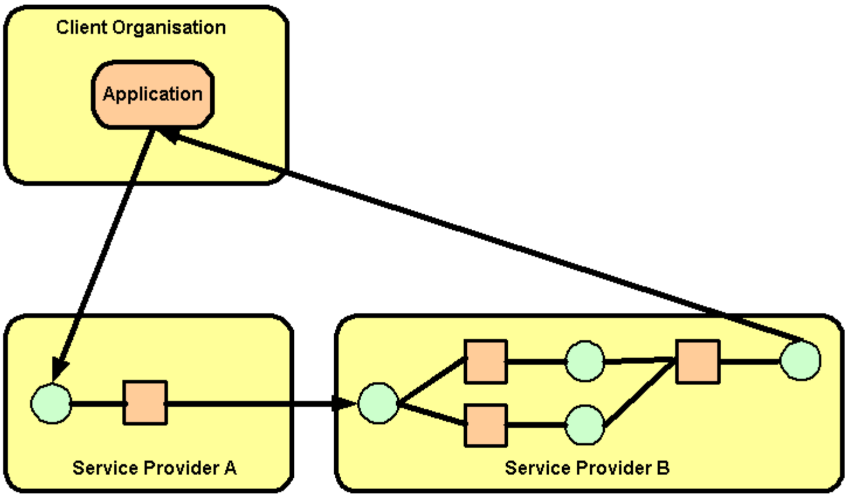


Figure 9: Workflow and optimisation

4.1.2.5 Computational Steering

Computational steering is concerned with the ability for a user to interact with and influence the execution of a process at runtime. This ability is of particular benefit for long-running and resource intensive applications where changes to runtime execution of an application process are required. Provision of support for runtime steering can obviate the need to restart an application process and suffer the overhead and other costs of setup and execution that have been already incurred, before the point in execution at which the intervention is required.

Computational steering in the aerospace activity requires may require changes to input and intermediate data sets and other environment settings used by iterative processes, in order to explore parameter spaces or correct inappropriate inputs. Furthermore, the facility may be provided to influence the control flow of execution of an application process at runtime. Computational steering for error recovery and optimisation of execution is another example, and support for checkpointing is important here.

Some of the challenges in supporting computational steering in the Grid infrastructure include facilitating changing application input data sets at run time in parallel environments. Furthermore, applications may provide custom steering interfaces and integration of these application interfaces or providing support for generalised interfaces may be required.

4.1.2.6 Checkpointing

Checkpointing of a process is the facility to pause its execution and capture its internal state, at one or more points during its execution. This facilitates a partial restart (or rollback) of the process to a previous state, without necessarily having to restart execution from the beginning. This is of particular benefit for steering long running processes, because if a partial restart is required, the setup overhead and some of the execution overhead, as well as other costs, are avoided.

Checkpointing is important for error recovery and provision of transaction support in execution of applications. Furthermore, a checkpoint for a process may be duplicated for multiple sequential or parallel executions of the process. In a Grid context, this might be done to support dynamic load balancing and optimisation of execution of the process on a computational cluster. Another reason for duplication of process execution from a particular checkpoint involves the case where a user wishes to conduct multiple executions of a process with different input data or other environment changes. This might be done to explore parameter spaces, for example.

The application is typically responsible for managing its checkpoints and providing facility to restart from a particular state. However, grid infrastructure might expose and perhaps generalise application-specific checkpoint and rollback features.

4.1.3 Ontologies

Ontologies are concerned with creating a formally agreed description of a domain consisting of concepts and relationships between concepts that are used to share information. Ontologies are key for semantic interoperability and supporting Grid dynamics [17]. Ontologies will be used in SIMDAT to describe service and data semantics that will support the development of service discovery and data integration components.

Service discovery is considered to be a capability that should be provided by Grid infrastructure. Service registries such as UDDI do not provide enough metadata capabilities to describe service semantics. Consider a design engineer wanting to out-source a crash simulation to a service provider in the UK for the cheapest price over a period of time. The engineer also wants to find post-processing services that can translate the simulation output to an IGES format that can be displayed by a appropriate visualisation tool.

The engineer can search the UDDI registry by providing a standard products and services code for Application Service Providers (81112106) using a global classification like UNSPSC and a geography code for UK. However, there are a number of problems with this process. Firstly, the engineer has to go through all the businesses found to check their services. These services could be anything related with application service provision and not only to crash simulation services. With a large Grid community the number of results are unlikely to be manageable. Secondly, it is not possible in UDDI to enforce a relationship between the service names and their function. WSDL [39] only provide the signature of the operations of the service, that is, the name, parameters and the types of parameters of the service. Discovering services by name may not be always very meaningful since a service name could be anything and in any language e.g. pamCrash, nastranCrashService. Finally, it is hard to identify complementary services since there is no mechanism to define relationships among services and data they operate on. In this case, the engineer cannot easily discover a service to translate the simulation output to an IGES format.

These limitations are not inherent in the UDDI specification, but rather stem from the lack of semantic descriptions. The OWL-S initiative from W3C is an OWL-based service ontology, which supplies service providers with a core set of markup language, constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form.

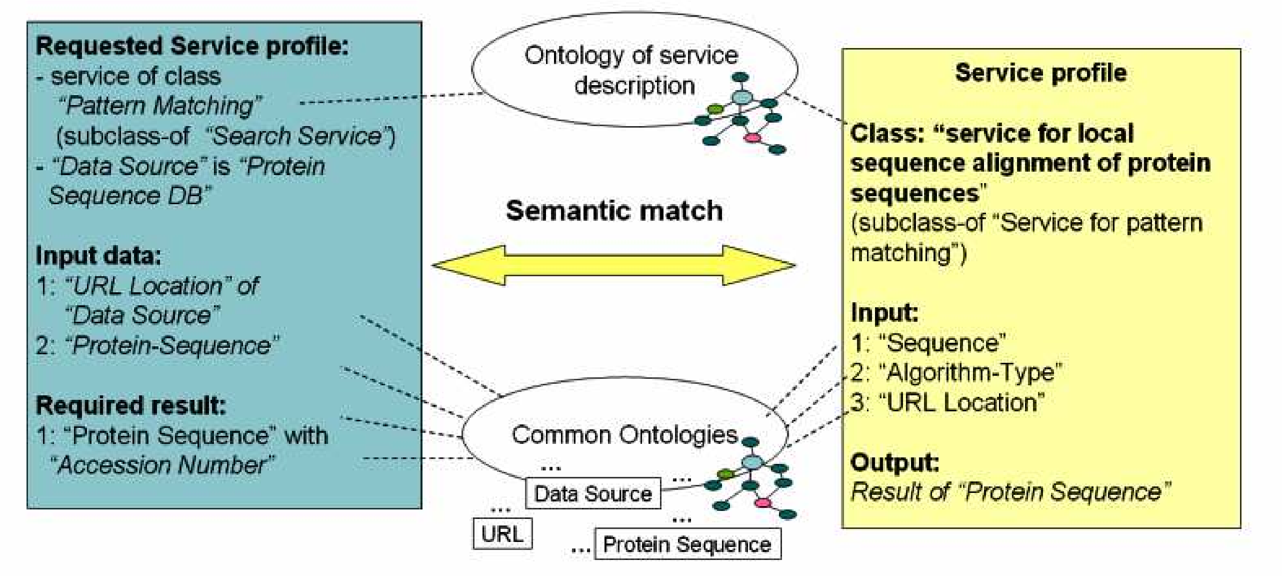


Figure 10: Semantic service descriptions in pharma

Figure 10 shows how ontologies can be used to describe the semantics of services in the pharma sector where many services operate on String types representing sequence data. The OWL-S service profile describes what the service does in terms of its functionality and data. In this case the service has been classified "service for local sequence alignment of protein sequences" with inputs "Sequence", "Algorithm-Type", "URL Location" and output "Protein Sequence". A class within an ontology is used to represent each classification so an application can understand the meaning of function and data. Describing the functional characteristics may not be sufficient for most grid applications. Non-functional characteristics such as security, quality of service, etc, are also important and more challenging as they are likely to change frequently.

A semantic broker will be developed that is used to discover services based on these semantic descriptions. A user should be able to define an abstract workflow that includes semantic requirements and submit the workflow to the semantic broker. The semantic broker will discover services and create a concrete workflow that has bound services, which can be submitted to a workflow enactor for execution.

In general, ontologies can be used to describe the semantics of any resource available on the Grid. If all resources such as databases, license servers, authorisation services, etc are accessed using services then ontology may need to be developed to describe both domain and infrastructure services. Ontological integration could then be used to provide interoperability between different Grid infrastructures by providing a common resource model for optimization and negotiation over heterogeneous technologies.

4.1.4 Integration of Analysis Services

Integration of analysis services is concerned with developing strategies for deploying industrial simulation codes into grid-enabled analysis services for use in grid enabled problem solving environments. Problem solving environments are a common trend in the industry as they allow integrating tools and data for product development. Problem solving environments, which are already in productive use in industry, are for example SRS by LION or MSC.SimManager by MSC.Software. These problem solving environments integrate a substantial number of analysis services, like mesh generation tools, data mining algorithms or tools for the prediction of the structure of molecules. Concrete examples are BLAST (Basic Local Alignment Search Tool) for rapid searching of nucleotide and protein databases, PAM-CRASH for automotive crash analysis or LMS OPTIMUS for multidisciplinary optimisation.

The infrastructure must support integration of legacy applications as Grid services, with no modifications. Batch application integration is an essential Grid requirement and is therefore well supported by most infrastructures. The execution management services provided by infrastructure is fundamentally responsible for managing the lifecycle of jobs including setting up the tasks (input/output data) and monitoring the execution. For batch applications, this would include creating a working directory with scratch space that is used for the lifetime of the execution.

The infrastructure must support integration of interactive applications. This requires the development of services that wrap existing applications exposing an interactive interface to users as service operations. Grid developers have experienced serious problems in developing and maintaining Grid services because of changing architecture, capabilities and standards supported by infrastructure. For example, services developed for GT2 had to be significantly changed for GT3 and now again for GT4. Standardisation is essential to support integration of interactive services and interoperability between different Grid infrastructures. For example, a developer should be able to deploy a service in GRIA that can be discovered and invoked from gLite.

WSRF and WS-I and are two initiatives that aim to overcome web service interoperability issues. WSRF is being developed within the Grid community and standardised through OASIS but does not have the backing of all leading vendors in the web service community. WS-I has commitment from all leading vendors, has wider acceptance and interoperability tests but has no explicit stateful resource model requiring stateful resources to be managed by the application rather than the infrastructure. It is still unclear which initiative will be successful and until the reference implementation of WSRF (GT4) is available and can be evaluated WS-I based services are the only sensible option for the 12 month SIMDAT demonstrators.

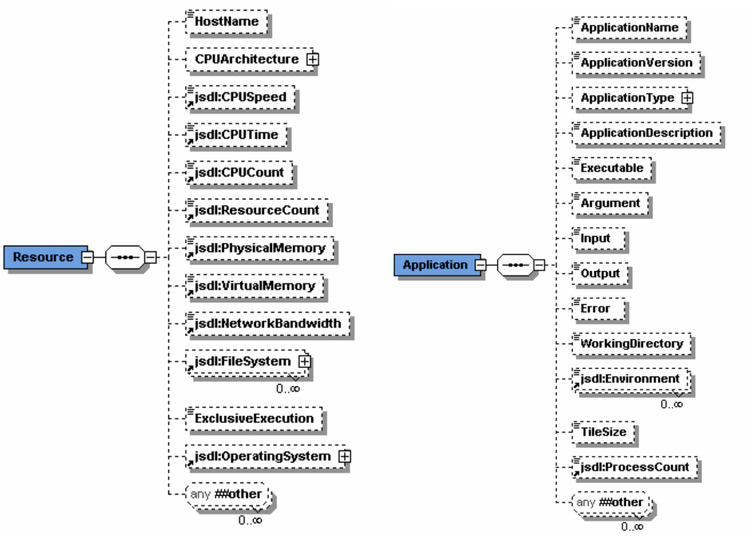


Figure 11: JSDL resource and application schemas

The accuracy of simulation results can be significantly affected by the computer architecture of the execution node and software versions. A user should be able to specify both parameters when requesting resource allocations from a Grid service provider to ensure simulation results that are consistent with local execution environments. Grid infrastructures provide ways to specify these requirements although they are non-standard, for example, GRIA defines an XML schema for Job requirements and UNICORE uses an AbstractJobObject. The emerging JSDL (Job Submission Description Language) [28] specification from GGF is likely to be important standard for describing these requirements. Figure 11 shows part of the JSDL schema for describing resources and applications. In addition to allowing user to describe job requirements JSDL aims to document how to translate JSDL specifications to scheduling languages supported by leading batch systems.

4.1.5 Knowledge Services

Knowledge service are concerned with automatically annotating distributed and heterogeneous data resources. The services will generate metadata repository that can be used for knowledge discovery about annotated text and data objects. The requirements for knowledge services are analogous to those described in section 4.1.5, however, the distributed nature of Grid data resources present's additional problems to traditional knowledge techniques.

Current knowledge technologies can operate on large data sets, however, all data needs to be co-located and integrated. In SIMDAT scenarios, such as the automotive sector, this is not possible as design teams share resources under their own control only authorising access to data necessary for a specific collaboration. Also the data volumes may be prohibitively large. In SIMDAT, to overcome this problem knowledge services will be developed that will allow algorithms to be sent to data resources and return generated metadata resulting from the task.

A demonstrator will be developed using the WEKA data mining toolkit and deployed on GRIA infrastructure with OGSA-DAI for distributed data access.

5 Grid Infrastructure Development Road Map

This section provides a summary of how grid technologies will be deployed and adapted for the SIMDAT PM12 application demonstrators and beyond. The definition of a standard Grid programming model is still under discussion (see section 2) and will significantly evolve during 2005 as technologies such as GT4, gLite emerge and GRIA continue to be developed. For PM12 the application activities have adopted a pragmatic approach by using existing Grid infrastructure and web service technology as the basis for their developments.

Each application sector is integrating Grid middleware into existing vertical applications to provide a demonstration of distributed collaborative working for complex problem solving. All application sectors have existing technology based on J2EE portals where the challenge is to adapt these existing centralised applications to a service-oriented Grid architecture. For example, J2EE applications provide a declarative centralised security model where all services are typically accessed through a single portal such as Lion SRS and MSC.SimManager. For a service-oriented Grid architecture centralised control of this type is not possible, as access to a service can be from any application or service running on the Grid.

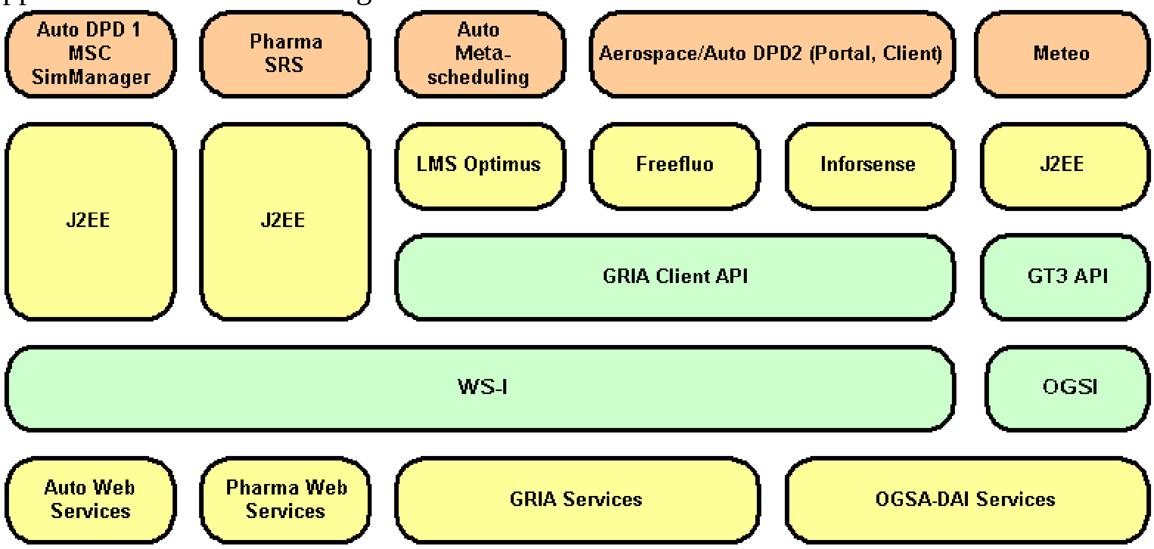


Figure 12: PM12 architecture

Figure 12 shows a high-level Grid infrastructure architecture for all SIMDAT application sectors up to PM12. Auto, Aero and Pharma will develop demonstrators based on a web service Grid conforming to WS-Interoperability. Meteo will develop a virtual data grid based on GT3 OGSI. GRIA will be deployed by Aero and Auto to provide support for the workflow-driven applications executing within virtual organisations.

Although Figure 12 shows how WS-I can provide a common API for distributed services it does not currently meet Grid infrastructure requirements for providing a standardised approach for managing stateful resources, as proposed by WSRF. Clearly, we need a common approach for stateful resources (context) to achieve interoperability between services deployed within heterogeneous Grid infrastructures. However, WSRF is complicated, consisting of various standards with some ambiguities. Programming directly using WSRF will be complicated and therefore an API similar to the GRIA client API is desirable to allow developers to easily and efficiently create Grid applications.

Figure 13 shows proposed generic road-map architecture for Grid infrastructure beyond PM12 that aims to achieve interoperability between different grid infrastructures such as GT4 and GRIA. The Grid service API should be based on WSRF, although the level of compliance may differ between implementations. For example, GRIA could be adapted to reference service context using WS-Addressing endpoint references rather than the web service parameters in the current implementation. However, this is only part of the WSRF specification.

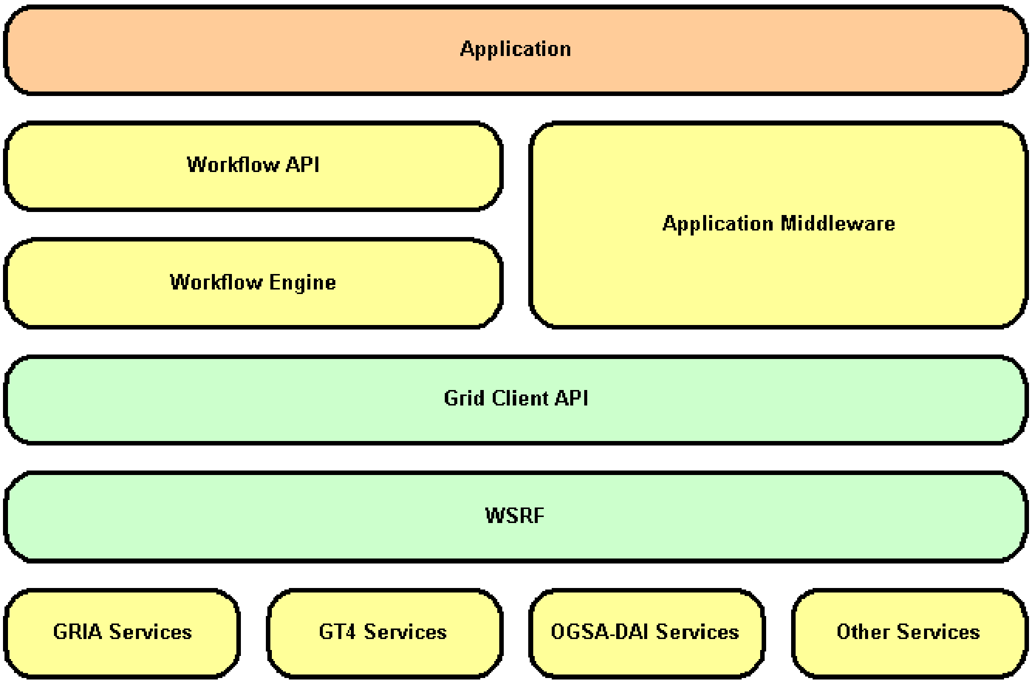


Figure 13: Roadmap architecture

In addition to the need for a standardized Grid service API there is also a need to standardize higher-level services and data models such as resource model descriptions, job specifications and provenance data. Other gaps in the Grid infrastructure technology include workflow as a the Grid programming model, and support for dynamic business processes, identity federation, notification, and dynamic service discovery.

6 Conclusions

We have provided a requirement specification for Grid infrastructure that has been elicited from the application activities during the first six months of the SIMDAT project. We expect that the requirements will evolve as each application activity further understands the potential of Grid technology to provide added value to existing problem solving environments during the development of the 12-month demonstrators.

Examining Grid infrastructure state-of-the-art it is clear that even the core technology which underpins higher-level services such as resource and execution management is still evolving. In future, core features should be part of a standards-compliant architecture, so application developers can use them more easily, and so they can choose between different (reusable) implementations. The WS-RF proposals for contextualised services are still somewhat controversial and WS-RF has yet to prove its value. The challenge of standardising the Grid programming model and associated management services is therefore still unfulfilled.

The application sectors have the challenge of selecting technologies that best fit their scenarios even if they do not provide all of the necessary functionality. We conclude that in the short-term, whilst a standardised Grid programming model is agreed, application activities should base new developments on web service standards such as WS-I. GRIA has emerged as a core technology to support collaborative working in the aerospace and automotive activities because of its availability, adherence to WS-I and explicit support for B2B collaborations. GRIA will be deployed in both sectors during the first 12 months. In the medium term, other infrastructure technologies such as GT4 and gLite should begin to stabilise. Each should then be re-evaluated as potential candidates for deployment.

The document also provides a discussion on how Grid infrastructure integrates with the SIMDAT technology activities. From the analysis, we can see that there are many gaps in existing technology that need to be filled to support the longer-term SIMDAT requirements including standardised job submission, notification, service discovery and workflow. Workflow is important for all application activities and a significant integration challenge for SIMDAT. The aerospace and automotive demonstrators will provide an excellent opportunity to further understand how the two technologies can be integrated with the objective of providing some best practice results and common infrastructure that may be deployed in other application activities.