



## An Introduction to the NextGRID Vision and Achievements V1.0

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
<b>2</b>	<b>THE NEXTGRID VISION .....</b>	<b>5</b>
<b>3</b>	<b>BUSINESS REQUIREMENTS.....</b>	<b>5</b>
<b>4</b>	<b>TECHNICAL CHALLENGES .....</b>	<b>8</b>
4.1	Security-Related Challenges.....	8
4.2	Management Related Challenges.....	8
4.3	Performance Related Challenges .....	11
4.4	Licensing Challenges.....	13
<b>5</b>	<b>ARCHITECTURAL PRINCIPLES.....</b>	<b>13</b>
5.1	Primary Architectural Principles .....	14
5.2	Secondary Architectural Principles.....	21
5.3	Architectural Decomposition.....	22
<b>6</b>	<b>EXPERIMENTS .....</b>	<b>24</b>
6.1	Financial Modelling: Implied Volatility and Derivative Pricing.....	24
6.2	Digital Media Production .....	27
6.3	Supply Chain Management .....	29
6.4	Electronic Data Record Processing.....	30
<b>7</b>	<b>THE GENERALISED SPECIFICATIONS AND COOKBOOK.....</b>	<b>32</b>
7.1	Introduction .....	32
7.2	The NextGRID Cookbook .....	32
7.3	Overview of The NextGRID Generalised Specifications .....	33
<b>8</b>	<b>CONCLUSIONS .....</b>	<b>34</b>
<b>9</b>	<b>ACKNOWLEDGEMENTS AND DISCLAIMER .....</b>	<b>35</b>

## 1 Introduction

This document presents the vision of the NextGRID project and shows how that vision is being realised. It introduces the principal outputs of the project, namely the NextGRID Generalised Specifications and associated Cookbook, which enable commercially viable Grids, conforming to NextGRID architectural principles, to be implemented.

This document addresses a business-orientated audience interested in exploiting the opportunities this new infrastructure creates. It sets out the capabilities and opportunities to be expected from Next Generation Grids; their realisation through the Generalised Specifications and the associated Cookbook; and the project's strategy for making a significant contribution to the development of Next Generation Grids.

In practice, the commercial adoption of Grid technology, in applications that cross organisational boundaries, has not yet happened. This is largely because the balance of risk and reward in a commercial environment is different from that underpinning existing academic Grids. Implicit business models make it difficult for each participant to assess this balance and manage risks. There are several specific factors that are holding back the deployment of multi-organisation Grids supporting multiple applications. These relate to security, service level agreements, software licensing models, pricing, competitive procurement of services, the diverse nature of business services, difficulties in estimating costs in advance of deployment and the perceived and actual difficulty of deploying Grids. A key task of NextGRID has been to understand these barriers to the wider adoption of current Grids beyond their current academic niche and to create an architecture which overcomes those barriers where possible.

The NextGRID vision is of Grids which are economically viable; in which new and existing business models are possible; in which the development, deployment and maintenance of applications are easy; and in which the provisions for security and privacy give confidence to businesses, consumers and the public. The requirements, placed on such an infrastructure to support viable business models, have been derived from real-world applications and have directed the technical work of the project.

The consortium which undertook NextGRID represents the different stakeholders from representative value chains. This is essential to ensure the development of Grid infrastructures driven by viable business models. The consortium comprised:

- Users with expertise in specific domains, where Grids have the potential to improve existing business processes and to stimulate new ways of doing business;
- Hardware and software providers to address support, interoperability and licensing issues;
- Service and infrastructure providers to address the viability of the commercial opportunities enabled by Grids;
- Research organisations and academia to exploit and extend their existing expertise in Grids by identifying new challenges and opportunities.

One long-term impact of NextGRID will be realised through the evolution of standards. This is a slow, but essential part of the process of technological adoption. The

Generalised Specifications developed in NextGRID will have a significant influence on the evolution of these standards.

Information and communication technologies are recognised as having a key role in Europe's transformation into a dynamic, competitive, knowledge-based economy. Sustained success is increasingly reliant on flexibility in business processes, which allows businesses to adapt to a changing global environment. IT applications and services are an essential enabler for this flexibility. To meet this need, there has been a clear shift in the market towards service-oriented IT systems. These allow consumers to obtain a wide range of services from a choice of providers, delivered via a ubiquitous telecommunications infrastructure. The emergence of this infrastructure has allowed users to enjoy permanent global connectivity from a range of wired and wireless devices without needing to be concerned with the technologies and networks involved. Grid technology is an essential component for the establishment of a marketplace, which builds on this global connectivity, where persistent compute and data-intensive applications can be shared securely and where services can be dynamically brokered and traded on a viable commercial basis.

Consequently, Grid applications must be capable of executing on an inter-enterprise, heterogeneous Grid infrastructure. The separate interests of independent stakeholders cannot be resolved a priori as is the case for applications designed to execute in a single domain. This implies that a Grid infrastructure must be capable of bridging the different business models used by different stakeholders at run time.

Grids are already delivering benefits to scientists and to some businesses, but only with an enormous cost in skilled human resources for the deployment and operation of Grids and the software installations on which they depend. This approach is not viable in a general commercial setting for economic, security and other reasons. There are also qualitative differences between academic and commercial business models for service provision. Sustainable commercial business models are essential for the viability of Next Generation Grids and will only be realised through the development of cost-effective and universally applicable technology.

Grids have the potential to make a significant advance beyond the Internet, by turning it from a passive information medium into an active tool for creating and exploring new knowledge and fuelling business and industry. A key issue in this transformation will be the realisation of the capability to compose services from independent sources in a standardised and cost-effective way.

Of course, NextGRID does not address these objectives alone. The participants in NextGRID are the representatives of a much larger community of researchers, technology vendors, service providers and users. This wider community has the critical mass necessary to make Grids a reality. NextGRID is an important catalyst in that overall process.

## **2 The NextGRID Vision**

The overall vision of NextGRID is one of creating an infrastructure to enable new business. This vision is inspired by the emergence of Grid technology from academic research where it has been used for a number of years to support research collaborations. This use by academic research has clearly demonstrated the sharing of resources to enable new ways of working and has enabled the achievement of previously unattainable results. The NextGRID vision now goes beyond academic resource sharing to encompass a dynamic marketplace where resources are available on a commercial basis and where services can be composed, brokered, bought and sold, seamlessly and automatically.

In this vision, Grid-based applications execute on inter-enterprise, heterogeneous Grid infrastructures which encompass at run time, the different business models used by different stakeholders. This implies a cost-effective and universally applicable technology supporting diverse and sustainable business models.

In this vision, a significant advance beyond the Internet is made by turning it from a passive information medium into an active tool for creating and exploring new knowledge and fuelling business and industry. A key issue in this transformation will be the realisation of the capability to compose services from independent sources in a standardised and cost-effective way.

This is a vision of a networked IT infrastructure able to support an unlimited range of applications and business processes throughout their lifecycle. This encapsulates all the necessary resources, including hardware, software, and data and services, available from a complex ecosystem of providers. This vision parallels developments in telecommunications which have now become ubiquitous and invisible with a shift from "occasionally connected" to "occasionally disconnected" to "always connected". Grids will evolve to provide all users with a global, transparent infrastructure providing rich and ubiquitous functionality and a dynamic marketplace for services whilst hiding the complexity of the underlying technology.

## **3 Business Requirements**

We shall now relate this overall vision to a set of requirements placed on Grids for them to enable a dynamic marketplace. These requirements comprise a sufficient set to support viable Grid-based business models and the specification of a viable infrastructure to enable those models, and have been determined by considering real-world business models and through experiments on real-world applications. A set of technical challenges and architectural principles have been derived from these requirements and have directed the work of the project. These challenges relate to security, management, performance and licensing. The architectural principles address SLA-driven dynamics, dynamic federation of resources and a minimal, but sufficient infrastructure. Full technical details are presented later in this document.

In any endeavour, business models are needed to balance risks against rewards. These may be described by explicit rules that govern the interactions between the participants or they may be an implicit consequence of the policies and technical measures used. The

business model adopted by one participant may be disclosed to others but this is not essential. For a business model to be viable, the balance of risks and rewards must make it beneficial for all participants. If the business model is unclear or does not deliver sufficient advantage to each of its participants, it will not be sustainable.

A global Grid marketplace where computing resources, information and services can be bought and sold has been envisaged for several years now, but has not yet been realised, other than in somewhat constrained and limiting instances. Amazon Webservices is an example of this. The ability to select and use service components from a variety of independent sources and integrate them into an application that delivers the functionality and performance desired is essential for the realisation of that global marketplace.

The real promise of Grids is that they will enable new ways of working and new business opportunities, supporting the flexibility and agility that is becoming increasingly important if companies are to be competitive in a global market. To realise that promise an infrastructure needs to be developed which will support the commercially viable use of resources by organisations whose interaction is predicated on the exchange of services for money.

Business relationships are generally codified in contracts. These make all relevant details explicit: defining what is to be provided, relevant business practices and standards to be used, as well as pricing and penalties for breach of contract. In a service-provision relationship, part of the contract is frequently expressed in a service level agreement (SLA). An SLA can be used to provide an explicit context for relationships between Grid entities, be they resources, individuals, or organisations. This context determines many of the managerial and operational policies to be applied within the relationship.

Based on these considerations we can identify the following eight business-driven requirements which a viable Grid infrastructure must satisfy.

**Requirement 1 – Flexible Business Models:** Whilst a number of specific business models likely to stimulate new business ventures have been identified for Grids, the important requirement is to provide support for all viable business models. Like the Web, where several distinct business models operate, Grids must be adaptable to a variety of models. Grid infrastructures must offer real benefits to a diverse range of business applications. Specifically, there must be financial advantages in the Grid approach over conventional in-house solutions.

**Requirement 2 – Specific Quality of Service Terms:** The functional description of a service, used for advertisement or as part of the SLA, does not address all the needs of a consumer or provider. In order to provide market differentiation between providers, quality of service (QoS) terms such as Reliability, Robustness, and Resilience are required to qualify the function provided. The scope for these QoS terms is virtually unlimited, so SLAs must support terms flexibly. Furthermore, to support the management of infrastructure on both the consumer and provider sides, the granularity of the terms must be flexible. Best practice among ISPs is to define management policies based on what is efficient for them, and base SLA terms on what these efficient policies can deliver. This was a major input from the telecommunication partners and NextGRID uses

the same approach. Furthermore, there need to be standard methods, suitable for automation, for consumers and providers of services to agree on service levels, pricing, penalties and the other terms that underpin a commercial relationship.

**Requirement 3 – Dynamic Security:** Even within “closed shop” environments, security is critical. Operational security and operational integrity must have a very high priority in commercial environments. In addition to the common concerns of communication security (authentication, authorization, integrity, confidentiality and non-refutation) and operational security (intrusion detection, accounting, risk assessment, and audit), there is a requirement that dynamically changing trust relationships be supported. The reasons for variations in the trust relationship range from simple accounting types (the user’s account is empty) to significantly complex (a user’s role has changed due to an acquisition by another company). These changes in trust relationships need to be managed throughout the lifetime of a business relationship.

**Requirement 4 – Dynamic Composition:** The NextGRID vision is of services that can be composed to create new more specialised services. In this vision, service composition takes place late in the service provisioning lifecycle, possibly even during execution. This requires that composition primitives, semantic descriptions of service components, and workflow technologies all operate dynamically and in concert with dynamic security.

**Requirement 5 – Economic Sustainability:** The sustainable deployment of Grids in a commercial environment requires strong economic underpinning. The barriers that prevent Grid infrastructures from becoming economically self-sustaining like the Internet must be identified and addressed. Support for the competitive procurement of services is essential to enable the transition to an open marketplace based on real currencies where goods and services are traded.

**Requirement 6 – Privacy:** Privacy, which is effectively the application and management of confidentiality in line with policies, preferences and laws, is essential in a Grid infrastructure where it must be a central aspect.

**Requirement 7 – Facilitated Management:** The ability to manage a Grid infrastructure effectively and efficiently is critical for its commercial viability. Manual management, particularly of a large dynamic infrastructure, is expensive and restricts the number of viable business models. The ability to manage an infrastructure at a minimum cost is an important requirement. Viable Grids need to operate with a high degree of automation because the operational costs of commercial IT are dominated by staff costs. Furthermore, any management infrastructure must integrate with the underpinning business infrastructure, in particular through the implementation and management of SLAs.

**Requirement 8 – Interactive Support:** The provision of interactive support can be decomposed into a number of independent properties of a service relationship, e.g. response time, control, feedback and privacy. Grid implementations must provide mechanisms to specify and support these capabilities in service deployments. This has an impact on both the SLA and on the basic Grid infrastructure.

NextGRID has addressed these requirements through the design of a service-oriented infrastructure which supports the commercial, public and scientific sectors. For all these requirements, as with the Internet, global standards are essential to provide stability and flexibility and to avoid vendor “lock in” and lack of competition.

## **4 Technical Challenges**

In order to realize its vision, NextGRID faced many significant technical challenges, some of which were easily identified from the beginning while others have emerged during the project. These technical challenges have been identified through an analysis of NextGRID’s key application driven experiments and can be classified into four areas: Security; Management; Performance; and Licensing.

### **4.1 Security-Related Challenges**

Applications in important commercial fields, such as finance, drug and automotive design, in the processing of medical and personal data, and in the media industry, have a requirement for absolute security of competitive or personal information. For example in the financial sector regulatory requirements limit access to data even within a single organisation. An important challenge in a Grid system is the mapping of security and confidentiality rules defined by the regulatory authorities and internally within companies.

It is obvious that the inter-organisational character of the Grid requires very strong security and confidentiality measures when it comes to the operation of enterprise business applications that deal with sensitive data. This involves not only legal regulations (data privacy laws, regulatory laws etc.) but also company policies and additional concerns that arise from the shared nature of the Grid paradigm. It is important to ensure isolation on the Grid resource provider side given that a direct competitor might also be a customer of that resource provider.

The NextGRID architecture incorporates a federated (cross domain) security model based on dynamic security token services and supporting dynamic authorisation policies. This approach takes account of the requirements of businesses that will use NextGRID and the constraints of regulatory authorities. The key aspect of security is flexibility, with so many competing requirements. The NextGRID architecture provides a mechanism where, by using standards, the security needs of one partner are made known to the other and parallel infrastructures on both sides allow flexible relationships to be established. The implementation of a local (autonomous) security policy fits flexibly into this framework.

### **4.2 Management Related Challenges**

#### **4.2.1 Availability**

High availability is essential for the core components of any Grid deployed in a commercial environment. This may require the duplication of key components and other measures to ensure the continuing availability of services. Only if inter-organisational Grids can provide at least the same levels of availability, quality, cost effectiveness and

flexibility as intra-organisational Grids, will they be acceptable. That is, hard SLAs must be put in place and enforced. A critical issue is to ensure that SLAs are enforceable and for them to specify what happens when things go wrong so that violations of individual terms do not necessarily invalidate the entire SLA. It is also essential that SLA terms are manageable, and do not create irreconcilable conflicts between the needs for high availability and for high utilisation. The NextGRID architecture provides features which support the implementation of services which can show the required dynamic behaviour, but still be governed by an SLA. In order to provide a greater degree of choice to NextGRID users, the architecture also incorporates a Quality of Experience (QoE) mechanism, which allows users to post and access experience rating with respect to a given service.

### **4.2.2 Service Discovery**

Dynamic components must be provided for the discovery of available services and for describing them. This set of services is expected to be large and constantly changing. On one hand, discovery components must handle frequent service-set updates. On the other hand, they have to provide acceptable look-up throughput to service consumers. Throughput is affected by the number of service queries, which is expected to be large, both because of the large number of clients and because of the need to keep the local view of service availability up-to-date.

NextGRID has addressed this challenge by creating dynamic service registries which attach life times to registrations and which require the periodic update of registration information. To increase scalability and throughput, NextGRID proposes the use of distributed registries that contain different sets of services and that are used by different clients. The NextGRID Registry provides a hierarchical and scalable mechanism to publish information about services and the requirements to use those services. This registry concept is extended in several Generalised Specifications, including the SLA Template Repository and the UDAP, to provide specific discovery information within these contexts.

### **4.2.3 Integration of Legacy Code**

An important challenge in the implementation of a Grid infrastructure is the integration of legacy code either to provide or consume services. Fundamentally, NextGRID must be able to support applications that do not use middleware or allow for code modification via a container or VM-based approach.

To address this challenge NextGRID has developed a model for dynamic orchestration based on workflow description and enactment which supports the integration of legacy applications. Furthermore, the Unified Dynamic Activity Package (UDAP) provides uniform activity management to some classes of application.

### **4.2.4 Usage Control over Resources**

Next Generation Grid systems must support the controlled usage of Grid functionality on both the resource-provider and resource-user sides. For example a service may have over

100 consumers on the user-side supported by a smaller group on the provider-side. Since the actions of Grid users may have a large effect on the budget of their organisation or on the quality of service for other users, the system has to offer facilities to monitor, control and restrict the actions of users in order to follow financial and operational limits. The restriction and control mechanisms have to cover different organisational units and management levels within the companies concerned. Teams not normally associated with the development and deployment of the system will exercise much of this management. Due to the large number of Grid users, the tight latency demands of applications and the strong coupling with the services provided, it will be necessary to have a distributed, usage-control system to prevent management becoming a bottleneck.

NextGRID has addressed this challenge by creating a policy framework used in SLAs and based on semantics encoded within the framework.

#### **4.2.5 Dynamic Configuration**

It is desirable that a Grid infrastructure can be easily reconfigured, for example to speed up service invocations or to increase their security level. That means that the Grid infrastructure should be able to switch between different protocols or deal with protocols of different complexity. For example, the system should be able to switch messaging from a secured transmission over a non-secured network to a faster, unsecured transmission over a secured network. Furthermore the Grid should be able to handle simultaneously multiple security standards, billing systems etc. for different Grid services. Flexibility and dynamicity in configuration are key issues.

The NextGRID Naming Generalised Specification allows for dynamic changes in service deployment, e.g. location or protocol changes. NextGRID Dynamic security provides the necessary information for security federation and also contains references to the Service Level Agreement associated with the interaction.

#### **4.2.6 Interoperability and non-exclusive resources**

In addressing the needs of business, the issue of interoperability arises. In a Next Generation Grid, business applications not only have to deal with different processor families, operating systems and database systems, but also with different Grid middleware. To protect investments, it is expected that the majority of customers will focus on middleware with commercial backing from one or multiple major vendors. However, even then, the middleware layer will need to be interoperable with other products.

NextGRID builds on many standards and has driven many others. NextGRID has also remained agnostic to some low-level issues, so allowing for more adaptability. Basically NG has trodden the line between flexibility (support for many authentication mechanisms) and standards (WSDL and WSs).

The NextGRID architecture incorporates a minimum infrastructure to support interoperability at the low levels. The use of SLAs, the ability to encapsulate services using composition primitives and the support for workflow based orchestration are key elements of NextGRID that address the issue of interoperability.

#### **4.2.7 Non-exclusive access to resources**

Another challenge occurs because the Grid resource management system cannot rely on exclusive access to resources. One of the reasons is that within a data centre, a Grid will share the resources with traditional infrastructures and enterprise resource management systems. A specific example of this would be the sudden loss of a Grid resource that was also part of a fail-over infrastructure in a high-availability system.

The NextGRID architecture uses an operational management framework to allow service providers to manage resources. This includes the ability to discover and seamlessly bring into play external resources if local resources are insufficient. Once again the SLA is fundamental to enabling this by giving the service provider flexibility in how it provides the agreed level of service. SLAs are used to identify when it is required to migrate an application to alternative resources. Naming is used to move Grid users to other resources in the event of failover. The service consumer, for their part, does not need to understand how the service is provided, and changes in configuration are transparent.

#### **4.2.8 Data Management Technologies**

Many business applications are database intensive. Other applications, however, require more complex management capabilities, such as data federation and replication. Given the possible distance between the database server and the compute node in a Grid, appropriate mechanisms to handle aspects such as latency in the resource-discovery process are needed. Furthermore, the owner of data may wish to manage remote access to those data.

NextGRID had developed a number of data management technologies to address this challenge, such as the Unified Dynamic Activity Package (UDAP) which is an information model technology, an information discovery framework based on query propagation and technologies to integrate and aggregate data.

### **4.3 Performance Related Challenges**

Performance-related issues always pose a challenge in any process-based system, be it a computer or human-based process. At the architectural level it is difficult to ensure good performance because this depends on the resources comprising the Grid and on their interconnection. Consequently in NextGRID the strategy has been to permit performance-based Service Level requirements and model them explicitly in the architecture. The following three subsections highlight three examples based on application experiments that need to be addressed between enterprises engaged in a Grid-based interaction. Note that performance requirements of the applications themselves are internal to the service provider, easily managed with SLAs. They have little direct effect on the architecture of the Grid. However, the architecture must enable high performance implementations so that attractive SLAs can be offered.

### **4.3.1 Data Rates**

In financial applications, there are high or peak data rates for fast access to market-data sources. In video processing, the amounts of data that must be handled, even in the simple case of a 15-minute film, are very large. The storage capabilities of the environment must be enhanced so as to deliver the result in a satisfactory timeframe. In turn, the data-transfer service requirements are equally large. Available network bandwidth must be maximised in order to create an environment capable of providing cost-effective solutions.

In addition to providing the SLA infrastructure which allows the service provider to manage the risks of offering a service where the data rates are demanding, NextGRID has investigated and developed Parallel HTTP, a high-performance data transfer mechanism, to address this particular need and improve performance.

### **4.3.2 Scalability**

For NextGRID to succeed, it must be possible to implement the architecture in a scalable way. This scalability must be in terms of the numbers of users, service providers, complexity of services offered and usage patterns. These issues have been taken account of throughout the architecture. The architecture allows multiple distributed information sources (registries) to exist, and the security, discovery and SLA negotiation mechanisms are decentralised.

The requirement to establish SLAs need not constrain scalability; a service provider can offer an SLA where multiple invocations of the service can be made under a single SLA. This enables scalability in terms of the number of service requests by a user, since the overhead of setting up the SLA does not dominate. The NextGRID innovation has been a composable, dynamic, and hierarchically organised architecture to allow such scalability. The multi-level SLA cascade is a good example of this, see Figure 5 on page20.

### **4.3.3 Interactive Access**

In general, response times should be as fast as possible as latency is critical in many financial and other classes of applications. NextGRID has identified a number of concepts that can be assessed independently to identify the specific requirements of a Grid when providing a generalised notion of interactivity, leading to requirement R8 in Section 4.3.3. These are response times, degree of control over the applications, feedback from and interactivity with applications.

It was not NextGRID's aim to provide dramatic improvements in network latency and bandwidth. What NextGRID can do through the SLA and policy mechanisms, is to allow a service consumer to understand in business terms the level of interactivity being offered, and service providers to manage the risks of offering such a service with a given level of technology available. If a service provider offers a service which is very demanding in terms of its interaction with the user, then policies can be written to ensure that the terms are met. These policies might include, for example, always ensuring that key aspects of the service are carried out 'in-house', rather than outsourced.

## 4.4 Licensing Challenges

Existing software licensing models do not often sufficiently address the requirements of a distributed environment, particularly if the coalition is built dynamically and if the same service is provided simultaneously to different Virtual Organisation under different conditions. There are many different licensing models, addressing different aspects such as who (an individual or an organisation) is permitted to use the application, where the application may run, and whether these two aspects are fixed or may float. Many of these licensing models are poorly suited to Grid scenarios. The challenge in NextGRID is to determine appropriate licensing conditions and terms for the offering of application codes over the Grid. Finally, the underlying business model of the software vendor and sometimes their licensees always drives licensing. NextGRID has taken account of this challenge by ensuring that it supports a wide range of licensing models, so that the architecture does not constrain the business models that could be used for developing and distributing software over the Grid.

## 5 Architectural Principles

This section outlines the primary and secondary architectural principles that form the basis of the NextGRID design. These principles are drawn from the challenges posed in Section 4 and motivated by the NextGRID vision presented in section 3. A decomposition of the NextGRID architecture is presented at the end of this section.

The primary architectural principles form the foundation for dynamics and interoperability and make NextGRID feasible. In NextGRID these comprise:

- **SLA-Driven Dynamics:** SLAs are critical building blocks in the NextGRID infrastructure and their dynamic behaviour is central to the operation of any Grid following the NextGRID architecture;
- **Dynamic Federation:** The dynamic federation of resources is a key factor in establishing operational business Grids;
- **Minimal Grid Infrastructure:** Any Grid needs to be simple to ensure ease of maintenance and wideness of applicability. However it needs to have sufficient features to enable it to support viable business models.

The secondary architectural principles form the foundation for development, deployment, and management and make NextGRID efficient. In NextGRID these comprise:

- **Dynamic Service Lifetime:** Services must persist when they are needed, but vanish when they are no longer required;
- **Dynamic Content Support:** Service content must be able to be augmented and evolve during the lifetime of a service;
- **Manageability:** Dynamic Grids must be manageable autonomously and such solutions must scale to encompass large-scale Grids;

- **Discovery:** Any Grid must enable the discovery of services by a range of methods;
- **Open Design and Development Process:** Next Generation Grids will be highly distributed and composed of services from a range of providers. Such components must be interoperable and subject to sufficient commonality of design.

In order to present the architecture of NextGRID, some form of architectural decomposition is necessary. This is presented in the final part of this section. As a result of the analysis of the evolving cycles of the NextGRID architecture, this decomposition is represented by four concepts: Schemas, Management Systems, Functional Systems, and Orchestrators.

## **5.1 Primary Architectural Principles**

The primary architectural principles underpinning NextGRID fall into three categories.

**SLA-Driven Dynamics:** The Service Level Agreement (SLA) is central to the conceptual model of NextGRID and therefore forms a key aspect of the underlying NextGRID infrastructure. All interactions in NextGRID are predicated by an SLA, dynamically created and aimed at ensuring that the relationship between provider and consumer is well defined and understood. Follow-up capabilities allow for monitoring, violation management, and audit. The SLA-based approach to all non-functional (as well as functional) aspects of NextGRID provides a uniform framework for the management and operation of all QoS aspects of NextGRID such as performance, security, provenance management, adherence to privacy regulations, etc.

**Dynamic Federation:** As a dynamic Grid infrastructure, NextGRID provides extensive capabilities for service construction and composition, including the composition of traditional interfaces, various forms of workflow-enabled orchestration and support for dynamic extension of the capabilities of services.

**Minimal Grid Infrastructure:** All services operating in a NextGRID environment can expect to find, but are not required to exploit, a minimal level of capabilities either available in the environment or exhibited by peer services. These capabilities are further refined as communication protocols and languages, behavioural interfaces available on all services, management services from the environment, and a common infrastructure of underlying schemas. A delicate design balance has been required here: too much sophistication in the minimal capabilities and the ability to deploy and maintain NextGRID effectively would be lost; too little and there would not be enough infrastructure to build Grids dynamically or manage them once instantiated.

### **5.1.1 Service Level Agreement Driven Dynamics**

An SLA defines the nature and consequences of an interaction between a service provider and a consumer. The agreement is made in some business context, which may include decisions made by each party leading up to the agreement, the presence of an endorser for the agreement and simply some prior conditions that make the terms of the agreement acceptable to both sides. An SLA is typically established before deploying a service and

covers the whole lifecycle including execution and monitoring through to decommissioning. However, it is also possible to form an SLA with an existing service, e.g. through a federation process orchestrated by an existing consumer that produces new interactions with other consumers. SLAs therefore have a huge influence on all aspects of the service, from as early as design time, to the infrastructure the service is deployed and executed on and the monitoring components that will be required for the provider to offer a service successfully in a NextGRID environment.

Prior to the creation of an SLA, each party is operating within a private context. Following the creation of an SLA, a shared context is established. All the content of this shared context is embodied (directly or indirectly) in the SLA. Note that in the creation of an SLA, elements of the two parties' private context must align in some way to allow the two parties to create the SLA and hence establish a shared context for business. These elements that align two parties so that SLAs can be created may be common to both parties (e.g. the same trusted Certificate Authority), but they are not considered "shared". Note that while many aspects of context, following the creation of an SLA, can be shared or agreed (e.g. roots of trust, expected QoS, etc), some should not (e.g. no commercial service provider is likely to reveal their resource plan to a consumer).

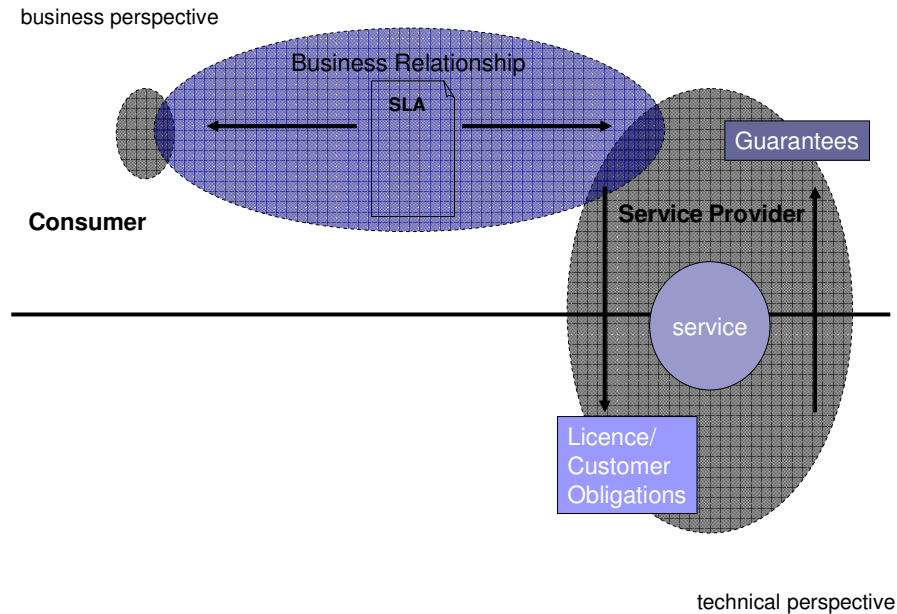
#### **5.1.1.1 SLA Structure and Contents**

A SLA exists between two parties, the service provider and the consumer. However, in some instances there is a role for third parties. This may be in providing independent verification of monitoring information in the case of disputes over violations and to give confidence to the consumer that the provider is fulfilling their obligations. However, by building a robust and non-ambiguous SLA framework, the need for trusted third parties can be reduced and replaced with the provider and consumer performing their own monitoring in a mutually trusted way.

Considerable effort in NextGRID has been focused on the structure of a SLA, so that it can provide all the information that the other components require, in a standard, structured way, which allows for automated and therefore more economic processing. We see the SLA as containing not only information relating to the specific guarantees offered on the performance of the service, what we categorise as "Dynamic Terms" but also to the commercial "Due Diligence" terms. These terms describe the policies in place in the environment in which the service will be deployed and executed. We describe these terms as the "Static Terms" as they are less likely to change between many SLAs. The inclusion and context of these terms have drawn on the experience of Service Providers and industry bodies. The static terms can refer to policies on holding transactional information, especially for finance applications, and therefore influence the lifetime of the SLA.

In the Dynamic Terms we identify high-level terms, that are closer to those understood by consumers or applications and guarantees are based on these terms. These high-level terms are derived from a number of more system-specific low-level monitoring points that are provided by the service, but which are not so obvious to consumers and which may not be visible to them. The period during which the collection of data from these monitoring points takes place may be different from the lifetime of the SLA. This is due

to the requirements outlined earlier that mandate the long-term responsibilities placed upon service providers in certain regulated industries.



**Figure 1: Business level SLAs and technical resource management**

To make this work, mapping mechanisms are needed as shown in Figure 1:

- to translate business-level objectives defined in an SLA into resource management policies that can be applied at the technical level within the service provider environment;
- to translate technical-level monitoring information into business level consequences that can be compared with an SLA, used for steering the management of the service, and used to provide meaningful feedback to the consumer.

Note that the technical-level management is not only concerned with resources and performance. For example, it may be necessary to use licensed software to fulfil an SLA. This could be mapped to a simple resource (in this case licence) allocation policy. However, if the SLA says the customer should provide the software licences, then the mapping will be onto a security policy that restricts access unless the customer provides evidence (e.g. a security token from the software vendor) to prove they have provided the necessary licences.

### 5.1.1.2 Protocol

Negotiation of an SLA should be as flexible as possible, but at the same time aligned with the negotiated services lifetime. It is for example usually counterproductive to use a protocol needing a higher time span to negotiate than is expected to perform the requested service. There may exceptions to this, but in general the creation of an SLA is

an overhead and efforts to minimize it are valuable. Except in very special circumstances, SLA negotiation must be many times cheaper than the value of the service.

To keep the effort (and cost) as low as possible, NextGRID advocates the “discrete offer” protocol: The Service Provider offers the Service Customer some discrete services (e.g. Service A, Service B, Service C or “Gold”, “Silver”, “Bronze”), from which the Service Customer may choose. In the discrete offer model, there is no scope for negotiation as the parameters of the offered services are fixed. Note that the customer may also make the offer and have it accepted or rejected by the Service Provider in a symmetric way. In addition, both provider and customer may advertise template SLAs where terms with a restricted range of values are left to be filled in by the other party. By working from one of these templates the requester (either consumer or provider) may optimize the process. Note that in many cases these templates will need to refer to already agreed contracts or possible standardized terms and conditions in order for this process to be fully automated.

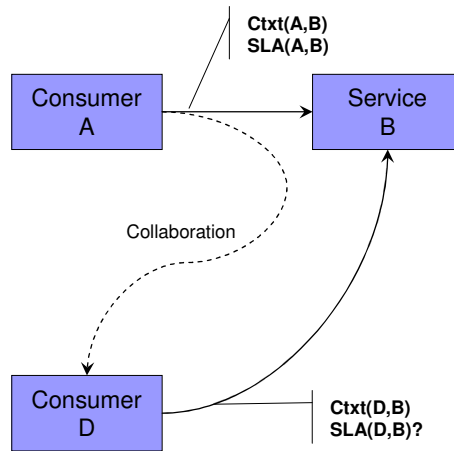
## 5.1.2 Dynamic Federation

### 5.1.2.1 Federation primitives

The NextGRID architecture is intended to support the potentially very rapid dynamic federation of resources to support user communities. Architecturally, we assume that applications will be constructed by composing NextGRID services, each of which has some common properties and behaviours. When executing applications, we can assume that certain core “infrastructure” services or properties are available in the environment of the application. A key requirement is that such federation mechanisms should result in architecturally self-similar structures that are themselves amenable to NextGRID composition rules.

A key aspect of the current NextGRID conceptual architecture is that all interactions will be governed through bipartite SLAs. Based on the understanding that business on the Grid is conducted in the context of an SLA, NextGRID has investigated how service compositions can be automated. At this stage, we hypothesise that there are at least two, and possibly three primitive composition rules that must be analysed. These are:

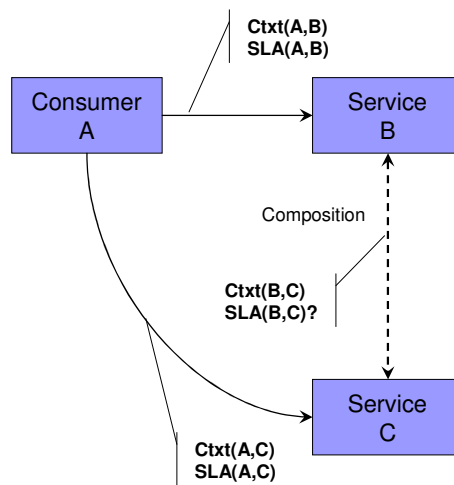
**Resource sharing** is where a consumer of a service shares it with another consumer:



**Figure 2: Resource Sharing (Consumer Federation)**

Resource sharing is strictly a federation between consumers. It makes the two consumers part of a related set of interactions as seen by the service provider. Resource sharing is very important for business Grids.

**Resource Orchestration** arises when a consumer of two services asks them to interact in some fashion:

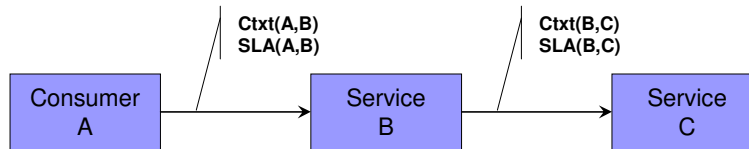


**Figure 3: Resource Orchestration.**

This process effectively combines resources from two service providers to meet the needs of the common consumer.

It is feasible to automate formation of the context and SLA for the B-C interaction, based on the terms covering the A-B and A-C interactions. In NextGRID, we identify general models for automating orchestration processes.

**Resource Encapsulation** arises when a service provider delivers one service to a customer by using another service, with no direct interactions between the provider of the second service and the consumer of the first:



**Figure 4: Resource Encapsulation.**

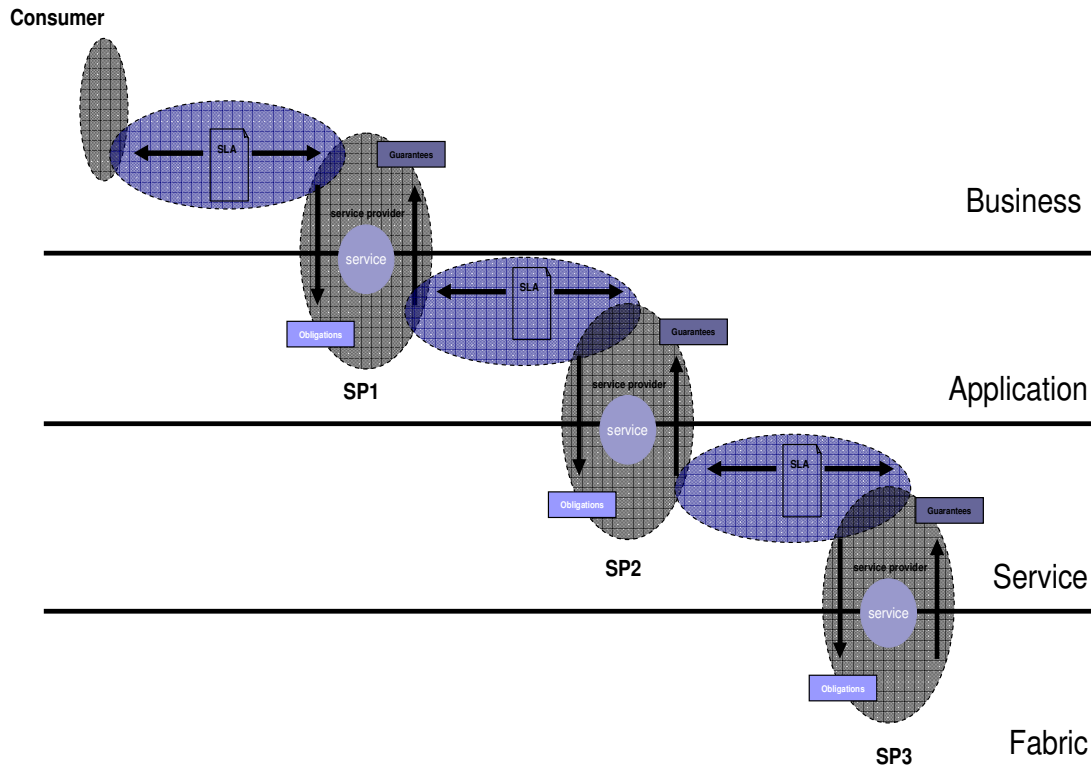
Encapsulation can begin from either side. The intermediary (B) may already be providing service to consumer (A) and then decides to outsource some of the work to an encapsulated service (C). Or the intermediary may have established their relationship with the encapsulated service (C) prior to offering services to the end-consumer (A).

In encapsulation, though the shared contexts between B and C form part of B's view of the context for providing service to A, there is no requirement for B to share this information with A. For this reason, the NextGRID architecture should not assume that any information will be propagated between A and C within this federation pattern.

### 5.1.2.2 Implications for SLAs

Resource sharing and orchestration both involve the creation of new bilateral relationships with a service, initiated by an existing consumer. NextGRID requires that every bilateral relationship should be governed by an SLA. Does this mean the participants in these relationships have to negotiate new SLAs before they can start fulfilling the purpose of the initiating consumer? Investigations in NextGRID suggest this is not necessary, as it should be possible automatically to infer the terms of the new SLA from the terms of the original SLA(s) with the initiating consumer. Thus in Figure 2, the terms covering the B-D interaction should be specified (directly or indirectly) by the SLA between A and B. In Figure 3, B and C should be able to infer the terms for their interaction from their respective SLAs with A. In either case, it may be necessary for A to pass some information (context) between the parties before they start interacting. In all cases, the SLA terms should anticipate and specify the process A should use to initiate the federation, so that bipartite SLAs can be used to support multi-lateral federations throughout their life cycle.

Figure 5 shows an example of this approach, in which 4 distinct levels are identified:



**Figure 5: SLAs and different service levels**

Here the communication (and agreement) between a service consumer and a service provider is on the business level. Instead of mapping directly to the fabric (CPU, networks, etc), this service is provided by encapsulating other services, each governed by its own SLA which is expressed in terms understood by its consumer. The management policies should then specify the requirements to be met by SLAs from the layer below and the monitoring and corrective action to be used to detect and recover from any breaches of those SLAs. Note that it is possible for different layers to be provided by different organisations e.g. a provider of financial services might deliver the business service using its own applications, but use another service provider to supply the fabric via execution and data management services. However, in most cases at least, some encapsulation is likely to be internal to a service provider.

Figure 5 also illustrates the principle of self-similarity in the NextGRID bipartite SLA-driven architecture, in that:

- every SLA will always have a consumer and a provider;
- every SLA has terms understood in the business context of the consumer; and
- every service is implemented by mapping its SLA with consumer(s) to a set of resource management policies, which can be treated as requirements for further services and SLAs.

The necessary resources can then be provided internally or by outsourcing to another service provider. Both of these approaches are fully compatible with the NextGRID architectural approach.

### 5.1.3 Minimal Grid Infrastructure

The key aspects of the Minimal Grid Infrastructure, introduced at the beginning of section 5.1, are the following:

- **Communication** – protocols and languages through which NextGRID communicates.
- **Behaviour** – interfaces implemented by all NextGRID entities.
- **Services** – those services that are always available to users and other services.
- **Schemas** – underpinning schemas of NextGRID.

In the main these interfaces and services are described in the Generalised Specifications, see section 7 for an overview of these.

## 5.2 Secondary Architectural Principles

The secondary architectural principles of NextGRID fall into five categories.

**Dynamic Service Lifetime:** Services implemented on the NextGRID architecture are highly distributed. Both users and providers need to take an active role in managing the lifetimes of services to ensure that they provide their intended function while they are needed, but do not consume resources unnecessarily when no longer required. The large scale of NextGRID means that centralised garbage management cannot scale. The lifetime of services will be negotiated between their users and providers. This needs to be dynamic as circumstances change during the lifetime of a service. The destruction of services must be explicit, once their users have finished with them. Implicit destruction must also be available, as host priorities change or if the users become disconnected from the services.

**Dynamic Content Support:** NextGRID is multi-purpose with the same infrastructure serving different services from multiple domains. Service content, added dynamically and evolving during the lifetime of the service, supports this diversity. The languages describing this content are domain specific and each is appropriate to its purpose. This applies to QoS values and service description and content.

**Manageability:** Dynamic Grids must be manageable. Large distributed Grids have unique management requirements. Such Grids will only be feasible if they are self-managed. The ultimate goal for next generation grids is that they are capable of autonomous management, based on a policy set by their owners, through scalable solutions where the management can be distributed amongst the services.

**Discovery:** The NextGRID architecture must support the full range of discovery methods, using registries as aggregators of information in well-known locations. It must also make relevant information discoverable at the service level.

**Open Design and Development Process:** The next generation of Grids, based on the NextGRID Architecture, will be highly distributed and composed of services developed and maintained by a large number of different entities. Such an ecology will thrive only if all the components are easily interoperable. Any non-interoperability would quickly fragment a Grid into distinct ghettos. Using a common architectural vision and open standards will help to prevent the fragmentation. It is also necessary that any innovations can also percolate quickly throughout a Grid. In order for this to happen, some implementations of the architecture should also be open and available for inspection and extension. These aims are best met using a standards-based, open-source model, along with other approaches, for coding of the infrastructure.

### 5.3 Architectural Decomposition

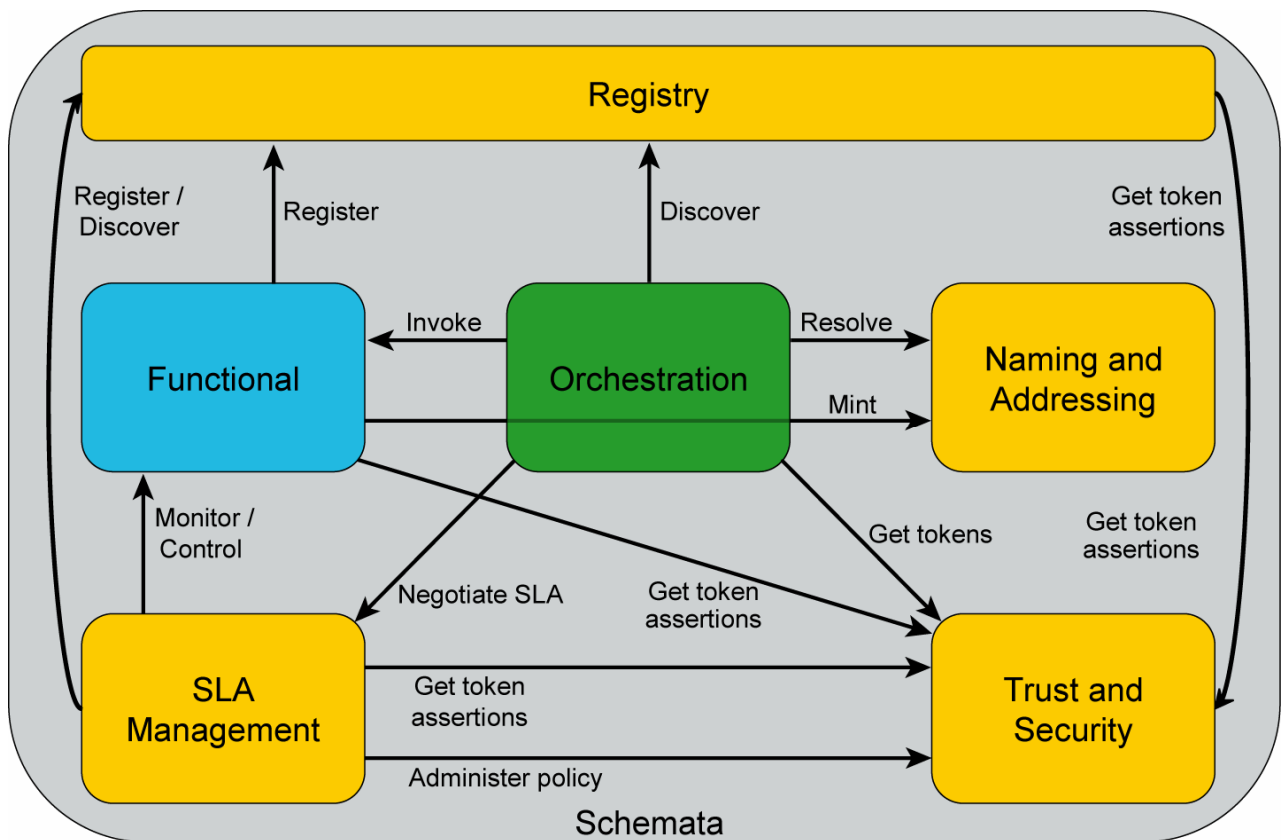


Figure 1

Figure 6: Overview of NextGRID Component Model and basic interactions

In order to communicate the architectural vision of NextGRID, some form of system decomposition is necessary. This is a more complex task than it appears at first. Frequently systems can be decomposed into a “layered” architecture, where each layer communicates only with the adjacent layers. However, the increasing complexity of Grid systems has resulted in the erosion of this simple approach, with some aspects of the system (e.g. security and provenance) spanning all layers of the architecture. Like Service Oriented Knowledge Utilities, NextGRID has moved away from the layered architecture approach. Likewise, the hierarchal inheritance-style decomposition, as employed in

object-based systems, is not as effective in service-oriented systems. Whilst the Service Oriented Architecture (SOA) approach requires more effort for implementation, its primary goal is to produce a more robust distributed system.

The difficulty in depicting the NextGRID Architecture (and any SOA for that matter) is that the compositional possibilities are virtually unlimited. They are not even restricted to single inheritance as in some object-based architectures. As a result, what a service-orientated architecture looks like depends on many parameters, some of which only exist at execution time. This is particularly true of NextGRID, where there is a strong emphasis on dynamic composition of services and capabilities. NextGRID has tackled the problem of decomposing its architecture through the regular analysis of the various iterations of the architecture. Here we have used the results of this process to present a conceptual view of NextGRID as it has evolved. Figure 6 depicts this decomposition and some of the interactions expected between components.

The NextGRID architecture can be decomposed into four concepts: Schemas, Management Systems, Functional Systems and Orchestrators. This decomposition is outlined below.

### **5.3.1 Schemas**

In order for the components of a system to communicate, a common language must be devised. In NextGRID, a number of schemas have been created that convey information from one part of the system to another. The primary categories for these schemas are: Message Schema, defining the content for addressing, securing, and defining function and quality of service for “on the wire” messages; Security Schema, defining the format for policy and token content; Service Level Agreement Schema, defining the negotiation language for consumer/provider agreements; and Service Description Schema, defining the framework for service discovery. In the published specifications, these schemas are included with the specifications to which they are related.

### **5.3.2 Management Systems**

These components provide a minimal environment in which NextGRID services function. On their own, Management Systems do not provide any “end user” functionality and are thus truly for supporting the infrastructure only. Each support system roughly parallels one of the schema discussed above, i.e. Naming and Addressing, Security, SLA Management and Repository, Operational Management, and Registry components. The bulk of the NextGRID Architecture is concerned with these systems.

### **5.3.3 Functional Systems**

The functional components of NextGRID provide the conceptual framework for anything that can be “done” using NextGRID. The detailed definition of the actual functions performed are not part of the NextGRID Architecture, although they can be roughly categorised in terms of their relationship to data which are access and transfer, processing, storage, and storage. For the base line functionality of these, NextGRID has defined some infrastructure specifications, which can be extended as use cases demand.

### **5.3.4 Orchestrators**

Orchestrators, typically represented by the Grid Virtual Infrastructure Model (VIM), facilitate and manage the dynamic composition of services in NextGRID. Orchestrators range from simple invocation to complex workflow processing engines.

## **6 Experiments**

The purpose of the experiments in NextGRID was two-fold. Firstly, they determined requirements that the applications place on the NextGRID architecture. Secondly, the experiments provided an important test of the architectural concepts and gave important feedback, enabling appropriate revisions of the architectural concepts to be made.

The NextGRID applications have been derived from four areas: Financial Modelling; Digital Media Production; Supply Chain Management; and Electronic Data Record (EDR) processing. In section 4 a number of challenges were discussed. These were drawn from an analysis of these applications. In this section, these applications are briefly described and the conclusions drawn and lessons learned from them presented.

### ***6.1 Financial Modelling: Implied Volatility and Derivative Pricing***

#### **6.1.1 Background**

The financial sector currently depends heavily on process and data-intensive computations to deliver competitive advantage. The two financial applications investigated were implied volatility and derivative pricing. The former treats the relationship between the theoretical price of a stock option and its volatility (likelihood that its value will change). The focus of the latter application is on complex derivatives. At present such tasks are largely performed onsite within a closed network. Secure, on-demand external provisioning of these computations represents a real opportunity for the EU financial services industry to increase competitiveness, manage risk and increase profits.

##### **6.1.1.1 Implied Volatility**

Implied volatility is a measure of the change in value predicted for a stock option based on the current market price. It reflects the current market conditions and hence can change constantly. Accurate and timely estimation of this parameter is essential for managing stock portfolios.

Within the stock market, stock and stock options can be purchased. Stock signifies an ownership position within a corporation. Options represent an option to buy (in the case of a call option) or sell (in the case of a put option) a set amount of stock from/to a third party at a set price (the strike price) in the future (the maturity date of the option). An option is purchased from a third party and if it is profitable on the maturity date (for example, the strike price of a call option is less than the current value of the stock, allowing the holder of the option to buy the stock more cheaply than would otherwise be possible) it will be exercised – otherwise it will be left to expire.

When the stock market is open, stocks and option prices are constantly being updated. Stock options are normally priced using the Black-Scholes model. This equation contains a volatility parameter, which cannot be observed in practice. There is a one-to-one relationship between the theoretical price of a stock option and its volatility. Unfortunately there is no closed form solution for implying the volatility from the stock option price. If the volatility is known, trades can be executed to take advantage of volatility spikes. The implied volatility, corresponding to a given price, must be calculated using a numerical method; a Newton-Raphson iterative process is normally used, which is computationally expensive. The peak rate of the option market is 120,000 prices per second for each of which an implied volatility value must be computed.

### **6.1.1.2 Derivative Pricing**

Derivatives are financial instruments whose value changes in response to the changes in underlying variables. The main types of derivatives are futures, forwards, options, and swaps.

The steadily increasing number of traded financial derivatives and the demand for more and more complex derivatives tailored for special purposes require continuously increasing capacities for the numerical valuation of these financial instruments. The price estimation for these complex products is already demanding, but their constant revaluation within the financial institutions' risk assessment, hedging and portfolio optimization processes can increase the computational resource requirements significantly. Regulation authorities amplify the need for fast and reliable computing capacities by imposing higher standards for availability, system integrity and security on financial institutions. At the same time, financial institutions feel the pressure to cut development times and costs for the introduction of new products and the need to reduce their IT costs.

Grid-based solutions will become more and more important in the financial sector, since they offer a higher utilization of existing resources and offer the additional possibility to increase the computational capacity temporarily by adding external resources in the case of peaks in demand.

### **6.1.2 Challenges and Requirements**

The challenges for the NextGRID architecture that arise from these applications comprise:

- Access to large market data sources with high peak data rates
- High availability for services (non-availability means absence from the market);
- Data access constraints (e.g. for back, middle and front office) governed by legal requirements (varying with different jurisdictions);
- Fast dynamic service discovery;
- Rapid response time
- High levels of security and support for multiple users and groups.

These are related to the following NextGRID requirements:

- **R2: Specific Quality of Service Terms:** It must be possible for the service users (financial traders) to understand the terms of the service in a manner meaningful for them. This could be expressed in terms of transaction rates and response times.
- **R3: Dynamic Security:** Security is of critical importance in these applications, and the need to manage multiple users and groups leads to the dynamic requirement.
- **R4: Dynamic Composition:** High availability leads to the need for customers and service providers to be able to switch suppliers if problems are encountered with the services they are using. On the end-user side this means that fast discovery mechanisms and a means of quickly establishing a new SLA are needed, while on the service provider side the same facilities are needed if failures occur with out-sourced services. In the latter case the client does not usually need to know about configuration changes due to the NextGRID encapsulation principle. This allows service providers to simplify the SLA, and gives them freedom to manage resources and provide a robust service. This requirement is also motivated by the need to develop new tools for a rapidly evolving market place, particularly with respect to derivative pricing.
- **R6: Privacy:** The financial industry is one of the most highly regulated. The architecture must allow the legal constraints, with respect to privacy and confidentiality, to be met.
- **R8: Interactive Support:** The applications are demanding in terms of the user interactions.

### 6.1.3 Architectural Validation

These experiments have validated the ability of NextGRID to implement a unified model for the representation of a service that provides a single interface for the registration and discovery of three service types. They have also shown that the NextGRID security model is viable and can operate across heterogeneous domains. Areas for further developments in the security area have also been identified.

The implied volatility experiments used a NextGRID UDAP component to orchestrate interactions between the parties allowing them to register centrally and discover resources. The experiment demonstrated that the NextGRID UDAP framework is capable of wrapping compute power, a financial web service and numerous data resources into a Financial Services Grid by orchestrating service registration and discovery. This financial scenario significantly benefited from the NextGRID architectural concept UDAP together with the NextGRID registry.

The implied volatility application architecture itself has also radically evolved from one concerned with the transfer of very large volumes of data throughout a network to one where software services are moved instead. This is an example of the influence of architectural features on an application, thereby enabling new business models.

The derivative pricing experiments concentrated on the security aspects of the NextGRID architecture, demonstrating the feasibility of providing a complete end-to-end security framework across enterprise boundaries. NextGRID endpoint security components were used to support secure control of the application from a Microsoft Excel client as

typically used in banks. An experimental scenario for pricing of complex options was composed in which results have to be available within seconds or less in order to adapt prices and hedging strategies to market changes, and to allow fast and reliable pricing of products tailored to the needs/demands of potential customers. The experiment showed that the NextGRID security model works and thoroughly tested the implementations of the NextGRID security components.

Further experiments examined more complete deployment scenarios involving core business aspects of the application. In particular, the NextGRID Operational Management package, coupled with the introduction of NextGRID SLAs, enabled a redesign of the Application Management package. The one-user-one-service approach was replaced by a service pool in which users share a pool of services, with one service pool for each of the SLAs on offer. This approach enables a more flexible and efficient usage of the supplier's available resources. The experiments showed that NextGRID architecture enables the transition from a business model based on a purely high performance computing grid application to a business grid application properly reflecting organisational structures.

Although NextGRID has made only limited improvements in technology to improve interactivity, the SLA mechanism allows the service provider and consumer to understand what level of interaction is guaranteed. This allows higher specification services to be offered when new technology becomes available.

## ***6.2 Digital Media Production***

### **6.2.1 Background**

Almost all films and commercials made today use computer graphics animations to implement the special effects that the artists want to depict on the screen. Additionally the number of video productions, made using exclusively compute-based techniques, is growing. Directors can use computer graphics animations, to enhance productions with outstanding features and special effects. This has been made possible by the accelerating rise of the 3D graphics and animation field, and is seen an opportunity for Grid based IT services.

The underling principle is to model the physical world's objects with 3D structural models that are covered with textures to imitate reality. 3D rendering involves several techniques for covering the 3D models with textures, bump mapping and visual effects, such as light sources and clouds. Software applications exist for building and rendering 3D scenes; these applications demand significant computational resources for all but the most trivial of scenes. The large number of objects, textures, light sources and effects, like shiny surfaces and fog, are all factors that limit the design of a scene due to the cost of providing the necessary dedicated computational power.

In this experiment, NextGRID explored the opportunity for external providers to offer services to special effects artists using Grid technologies, and studied whether the technical challenges of doing so could be met. In such a scenario the special effects companies, many of which are small, would use well-defined services (including the price) rather than incur the high costs of ownership of high performance computers.

Service providers would benefit from economies of scale in running a powerful computing resource and achieve economic sustainability by pricing services according to their added value to the consumers.

## 6.2.2 Challenges and Requirements

The challenges of the provision and use of a grid-enabled digital media application include:

- Ability to describe and discover the available resources in the grid infrastructure;
- Ability to discriminate between resources in order to select those that could meet the user's needs.
- Support for multiple users and groups on the consumer side
- Support for complex and dynamic workflows

These challenges are related to the following requirements from Section 3

- **R1: Flexible Business Models:** A future Digital Media grid eco-system with multiple suppliers of services will only be viable if flexible business models are supported.
- **R2: Specific Quality of Service Terms:** The QoS terms need to be expressed in ways that the service users will understand. In this application this could be cost per frame rendered.
- **R3: Dynamic Security:** Security is needed so that artists and designer only get access to the results they should. Service customers need to manage groups of users, leading to the requirement for a dynamic security model.
- **R4: Dynamic Composition:** In order to effectively execute the potentially complex workflows which might be generated, dynamic composition of services is a requirement.
- **R7: Facilitated Management:** Service providers will need to manage their resources carefully and efficiently to be economically viable. There can be a great variation in the computational complexity when rendering scenes, and effective management of resources to meet SLAs is needed.

## 6.2.3 Architectural Validation

The application was used to investigate how Quality of Service (QoS) information could be used by a client to select services. The experiments demonstrated that consumers were able to make an explicit service provider selection based on their requirements and the experience of the previous job executions. This underlined the importance in the architecture of decision support components to assist with service selection.

A second application experiment investigated SLA-based workflow scheduling. During the experiment, appropriate NextGRID components were integrated as well as specifications (SLA, Policy and Event Schema) formulated in order to fulfil all the application requirements. The experiment implemented end-to-end NextGRID SLAs and dynamic workflows, and includes complete SLA lifecycles proving that NextGRID SLAs are applicable to complex business scenarios. In addition the use of NextGRID SLAs in

each step provides SLA dynamics for the application, while the creation and execution of dynamic workflows validated the NextGRID dynamic federation model.

These experiments provided valuable feedback to the schemas used in NextGRID, based on real application requirements.

## **6.3 Supply Chain Management**

### **6.3.1 Background**

Supply Chain Management (SCM) is the process of planning, implementing, and controlling the operations of the supply chain as efficiently as possible. Supply Chain Management spans all movement and storage of raw materials, work-in-process inventory, and finished goods from point-of-origin to point-of-consumption. It is a highly complex application, which needs to be tailored to the needs of specific business. Most companies use third party commercial software running on dedicated systems. If certain challenges can be met, Grid computing offers opportunities in this domain by enabling different alternative business models, incorporating for example Software as a Service (SAAS), which could benefit both customer and service provider.

The SCM application experiment in the NextGRID Project was based on the SAP Web Application Server (SAP Web AS) which is the core platform for hosting SAP business solutions. This delivers a complete set of capabilities for collaboration, planning, execution, and coordination of the entire supply chain network. The creation and management of sales orders and the initiation of delivery notes are handled in this process. Other processes in SCM include Materials Management, Logistic Execution and Production.

Given the characteristics of the traditional architecture, it is obvious that the SAP Web AS is an excellent candidate for application outsourcing to a computational Grid to cover peak loads.

### **6.3.2 Challenges and Requirements**

The challenges for the SCM application in a Grid environment include:

- Delivery of agreed quality of service
- Data intensive workloads
- Management of service-side resources to meet fluctuating demand.
- Cross-organisational security
- Flexible pricing models to meet customer demands.

These challenges are related to the NextGRID requirements as follows:

- **R2: Specific Quality of Service Terms:** A general requirement for SCM (and most business applications) is to guarantee a certain service level with regard to response time versus throughput.
- **R3: Dynamic Security:** The use scenario requires crossing administrative domains and manages several user roles dynamically.

- **R5: Economic Sustainability:** The service offering includes added value by provisioning application components on top of any hardware platform.
- **R7: Facilitated Management:** Automation of application and operational management must be demonstrated. Dynamic scalability is needed to meet the needs of future application development.

### **6.3.3 Architectural Validation**

NextGRID carried out investigations into the use of an Operational Management Framework (OMF) with the objective of managing a Grid of servers hosting SAP Dialogue Instances (processes that carry out the SCM functions) according to policies derived from SLAs. A first version of the SAP Application Management Framework was developed allowing the dynamic provisioning of SAP Dialogue Instances (Web AS) triggered by NextGRID events and policies. The NextGRID OMF and the Application Management Framework were deployed at the Service Provider. Policies were used to determine actions to take if additional computing resource was required. The main benefit from the application side in this SCM scenario is that NextGRID dynamic management of Dialogue Instances reduces the total cost of operation. The NextGRID architectural concepts used introduce high flexibility via policy-controlled management and allow event driven operations of business applications.

Further experiments showed the integration of the NextGRID Business Management components. These are the parts of the architecture, which deal with the business relationships, i.e. such things as billing and accounting. Suggestions for amendments to the schema for events, policies and SLAs were made.

## **6.4 Electronic Data Record Processing**

### **6.4.1 Background**

One of the most-data intensive processes that must be carried out by a telecommunications operator is the analysis of e-data regarding the use of its network services. Data related to calls are stored in Electronic Data Records (EDR) files. These files are processed continuously in high-end machines and then stored in a Data Warehouse that is used by several business processes. Although it is unlikely that a telecommunications operator will, in the near future, outsource its databases and data warehouses, it may well need to add extra computational resources for EDR processing. In this case, the data would lie inside the telecommunications operator's intranet and computational resources be provided by a third party. NextGRID carried out experiments to investigate what architectural requirements would be needed to support these new business models for the EDR processing application.

### **6.4.2 Challenges and Requirements**

The experiments focused on the discovery of external computational resources, flexible security policies and the need for data high transfer rates. The following challenges were identified:

- Dynamic discovery of services.
- High data transfer rates.
- SLA infrastructure to manage new business relationships

These are related to the NextGRID requirements as follows:

- **R1: Flexible Business Models:** The scenario of using external grid-based resources is a new business model for EDR processing.
- **R2: Specific Quality of Service Terms:** The customers of the service will be interested in SLAs which describe the QoS in terms of this application. For example the terms could include the number of transactions per unit time.
- **R3: Dynamic Security:** As with several other applications studied, the different roles that the clients may have in the business relationship require dynamic security.
- **R6: Privacy:** There will be regulatory and business needs for data privacy in this application scenario.
- **R7: Facilitated Management:** Service providers will need to manage their resources to meet fluctuating customer demand.

### 6.4.3 Architectural Validation

The results of the experiments have revealed ways to avoid data transfer bottlenecks. These techniques have been fed back into the overall architecture. Furthermore they have validated the NextGRID SLA framework and dynamic security and trust mechanisms, and led to enhancements in the SLA framework.

A performance model was developed for the EDR Processing application regarding data transfer strategies to the service provider. Simulation with this model indicated that an “on demand” data transfer strategy would increase the scalability of the application significantly. Additionally, dynamic security experiments were performed with NextGRID components using a Grid-enabled EDR Processing prototype. The experiments included the examination of complex distribution patterns to improve scalability and performance of the application. The evaluation showed that NextGRID dynamic security could effectively support the “on demand” data transfer scenario with a reasonable overhead, and without imposing unmaintainable and costly security policies.

In further experiments, the NextGRID Business and Application Management components were integrated into the EDR Processing environment. An experimental scenario was developed where the customer owns a huge data set that must be processed. A lack of sufficient local resources usually makes it difficult to meet the time requirements of this process if performed exclusively at the customer’s site. Thus the customer publishes the data set as a NextGRID secure service and searches for a service provider to process it. As soon as a suitable service provider has been found he receives a job submission. The service provider then distributes the job to several machines. Each machine asks the customer data service for only the data subset it will process. This strategy significantly improves scalability and performance of the application. The NextGRID architectural concepts provide a coherent set of specifications to manage

security, trust, service discovery, SLA negotiation as well as QoS management and thus cover the whole “transaction” life-cycle in the EDR Processing scenario.

The EDR experiments also showed the value in reducing development times of using NextGRID orchestration support tools (Workflow Editor and Adaptive API).

## **7 The Generalised Specifications and Cookbook**

### **7.1 Introduction**

The overall vision of NextGRID is one of creating an infrastructure to enable new business. NextGRID has worked towards the realisation of such an infrastructure through the development of an architecture which support that realisation. This has been the motivation behind the work undertaken by the research organisations and commercial enterprises participating in the project. The resources needed for the implementation of a commercial grade Grid infrastructure are many times greater than those available in NextGRID to research such infrastructures. Consequently, the approach adopted by the project is to publish Generalised Specifications for the implementation of NextGRID-compliant components from which interested parties could implement a NextGRID-compliant infrastructure, either as a whole or as parts. A Generalised Specification provides the precise definition of an interface that may be supported by any number of components within the system. Each component is an identifiable software implementation of one or more generalised specifications. A component is designed to provide a given function and may implement several Generalised Specifications. An implementation of NextGRID is an integrated collection of components, based on Generalised Specifications that addresses a business need.

To complement the Generalised Specifications, the NextGRID Cookbook has been developed. This Cookbook presents a series of examples relating to several application scenarios which show how NextGRID-compliant components can be combined to construct a Grid addressing a range of commercial applications.

### **7.2 The NextGRID Cookbook**

The NextGRID Cookbook is a guide for system designers and developers who want to implement a Grid. It consists of a set of examples showing how specific software components can be designed, implemented and combined. It addresses an audience which has a basic understanding of Service Oriented Architectures. This Cookbook shows how these components interact so that systems of varying complexity can be built to satisfy the needs of particular applications.

The components presented fully comply with the NextGRID architectural principles. That is, they can be combined to form a complete, fully operational Grid infrastructure to address real business requirements. The components offer extensive capabilities for service construction and composition. The NextGRID Cookbook offers complete guidelines to design and implement these components and construct Grids from them. It provides extended examples which demonstrate how this can be done. The main concepts that govern the Cookbook are:

- **The Component Specifications:** The components are the independent, elemental entities that can be combined to form a Grid. . They will typically implement one or more Generalised Specifications.
- **The Descriptions of Systems of Components:** These are examples of the use of one or more components to address a particular business goal.
- **The Description of Application Scenarios:** These are application scenarios demonstrating the use of the components.
- **The Generalised Specifications:** These define and describe the interfaces to the components of NextGRID thereby enabling the design and implementation of a Grid.
- **The Integration Diagram:** This depicts the interactions of the NextGRID components.

The NextGRID Cookbook presents a set of examples for implementing a Next Generation Grid architecture. A culinary analogy is used, referring to ingredients, recipes and meals. The ingredients are a pool of software components that NextGRID has specified. The recipes are ways of combining these ingredients to provide specific functionality in a Grid environment. In this way a new system of more complex components can be created which following the culinary analogy can be thought of as recipes. By combining these further, one can implement higher level scenarios (until reaching the level of application scenarios) that could be regarded as ‘meals’. In other words the Cookbook provides guidelines for people to implement application scenarios by combining components.

More information about the Cookbook can be found on the NextGRID website [www.nextgrid.org/GS](http://www.nextgrid.org/GS)

### ***7.3 Overview of The NextGRID Generalised Specifications***

The NextGRID Generalised Specifications are divided into four categories as listed below.

#### **Management Systems**

- Naming and Addressing
- Security
- SLA Management
- SLA Template Repository
- Registry
- NextGRID Base Profile
- UDAP Framework
- Operational Management

#### **Functional Systems**

- Data Transfer
- Data Storage
- Data Processing
- Data Access

## **Orchestrators**

- Workflow including simple invocation

## **Schemas**

These specifications take the form of a series of individual documents which can be downloaded from the NextGRID website. In addition to these individual documents, an overall guide (or ‘cookbook’) is included, giving details of how the specifications should be used together.

### **7.3.1 Availability**

The Generalised Specifications became available in the second half of 2007 with the full set completed in Spring 2008. Full details are published on the project website [www.nextgrid.org](http://www.nextgrid.org) in connection with the associated Cookbook.

## **8 Conclusions**

The NextGRID vision is of future grids, which are economically viable; in which new and existing business models are possible and profitable; in which development, deployment and maintenance are easy; and in which the provisions for security and privacy give confidence to businesses, consumers and the public. This vision looks far beyond the academic roots of the Grid in aiming to support fully the requirements of organisations and individuals from business and the public sector. It embraces a service-oriented infrastructure, which is as ubiquitous and transparent as the Web is today and which provides support for commercial, public sector and scientific applications. The complete realisation of this vision will take much longer than the duration of NextGRID and involves a Global effort combining resources many times greater than those deployed in this project, particularly with respect to implementation and deployment. Nevertheless the work undertaken in NextGRID constitutes an important step on the path to that realisation – the creation of the Generalised Specifications; the blue print for the next generation grid.

The principal outputs of NextGRID comprise the Generalised Specifications and associated Cookbook, which enable commercially viable Grids, conforming to NextGRID architectural principles to be implemented. Associated implementations of all NextGRID components were also produced by the NextGRID project. These have all been tested in isolation and in conjunction with other components as part of the NextGRID experimental evaluation of the architecture.

In practice, the commercial adoption of Grid technology, in applications that cross organisational boundaries, has not happened. This is largely because the balance of risk and reward in a commercial environment is different from that underpinning existing Grids. Implicit business models make it difficult for each participant to assess this balance and manage risks. There are several specific factors that are holding back the deployment of multi-organisation Grids supporting multiple applications. These relate to security, service level agreements, software licensing models, pricing, competitive procurement of services, the diverse nature of business services, difficulties in estimating

costs in advance of deployment and the perceived and actual difficulty of deploying Grids. A key task of NextGRID has been to understand the barriers to the wider adoption of current Grids beyond their current academic niche and to create designs, which overcome those barriers where possible.

This document takes the NextGRID vision, relates it to the business requirements appropriate to the realisation of that vision, derives technical challenges from those requirements, relates the technical challenges to an overall architecture and represents that architecture in a set of Generalised Specifications.

The technical challenges have been derived a priori from the business requirements and have been measured against real-world needs through the analysis of application-driven experiments, described in section 6. The challenges and their experimental evaluation have together identified the primary and secondary architectural principles that form the basis of the NextGRID design. The three primary architectural principles are based on service level agreements, on dynamic service composition and on the specification of a minimal Grid infrastructure. The secondary architectural principles are derived from these primary principles and comprise dynamic service lifetime, dynamic content support, manageability, discovery and open design.

Together this framework of vision, challenges, analysis of application-driven experiments and use of architectural principles has taken forward the work of the project and resulted in a legacy which is the Generalised Specifications and their contribution both to the implementation of Grids and to their impact on standards.

## **9 Acknowledgements and Disclaimer**

NextGRID has received research funding from the EC's Sixth Framework Programme. This document expresses the views of the authors and not those of the European Commission. Neither the members of the NextGRID Consortium nor the European Commission are liable for any use that may be made of the information contained in this document.