



P1.2.1 and P1.3.1 Architecture Analysis and Design Report

Workpackage:	1	Architecture
Author(s):	Greg Kohring	NEC
	Kostas Tserpes	NUTA
	David Snelling	FLE
Authorized by	David Snelling	FLE
Doc Ref:	P1.2.1, P1.3.1	
Reviewer	Malcolm Atkinson, EPCC	EPCC
Reviewer	Lennart Johnsson	KTH
Dissemination Level	Public	

Date	Author	Comments	Version	Status
March 10	GK, KT, DS	Initial Draft	V1	Draft
March 10	SvdB	Revision 2	V2	Draft
April 6	DS	Revision 3, from comments	V3	Draft

1 Executive Summary

1.1 Analysis Overview

As part of the NextGRID Architecture Analysis, NTUA has designed and documented an architectural Component Model motivated by the NextGRID Conceptual Architecture modules and by various Business Models provided by WP3.

In addition to the Component Model, a number of use cases and scenarios for the specific Business Models are provided. These have been sketched using Together Design Community Edition. The diagrams are posted on the NextGRID BSCW¹ in a similar style to the Design sub-task deliverables.

The purpose of the Analysis is to provide input for processing by the Design sub-task throughout the project lifetime, producing a definition of those services that are required for an economically sustainable Grid.

It is worth noting that the Component Model's emphasis is on the way that the components interact, producing snapshots of the current operation's workflow. Part of our intent is that this component model provides input to the project by triggering the business aspects of the Grid's dynamics.

The results produced by the analysis task fall short of what was hoped for by the project. The primary reason for this was the relatively late (i.e. three months later than expected prior to the removal of the pump priming phase) WP3 (Business Models). This was partly remedied by using a draft WP3 deliverable as well as feedback provided by NextGRID partners, namely HLRS and IT-Innovations. We consider this issue to be an artefact of the start-up phase of the project, following the loss of the "pump priming" phase during negotiation. See D1.1 for a discussion of this.

1.2 Design Overview

As NextGRID has adopted a service-oriented architecture (SOA), the design is organized around the following major constituents:

- Interfaces
- Messages
- Data Types
- Metadata

The NextGRID components will interact through well-defined interfaces, by sending structured messages containing well-defined data types (syntactically SOAP encoded elements, backed by the

¹ The BSCW is only accessible to project partners, but it's contents can be made available on a case by case basis.



data type semantics defined in UML). Metadata is structured data used to describe any of the other constituents. In addition to the above elements, the design also contains some “grid patterns” whose objective is to demonstrate how the other elements interact to perform Grid functions.

It is not the goal of the design subtask to model any particular Grid implementation; rather, the goal is to create an abstract model of NextGRID, which can then be mapped on to system specific implementations. Hence, all design elements are high-level abstractions.

1.3 Choice of Tools

The NextGRID Analysis and Design deliverable is stored on the NextGRID BSCW server as a Together Designer project. The Together Designer Community Edition, which is freely available from the Borland web site, was chosen as the primary UML design tool because it offers the complete UML modelling language in a professional, commercial grade product. Furthermore, since the Community Edition is free, basing the project on it should encourage participation of groups outside of NextGRID in the process of maintaining and extending the design beyond the end of the project. An alternative would have been to store the design files in XMI format; unfortunately, XMI is available in several versions and different UML tools support different XMI versions, leading to compatibility problems. Every detail regarding this work can be extracted “en block” from the Design-Analysis repository.

The following sections list, and briefly describe, all the elements currently in the analysis and design subsections of the BSCW server.



2 Contents

1	Executive Summary	2
1.1	Analysis Overview	2
1.2	Design Overview	2
1.3	Choice of Tools	3
2	Contents	4
3	Analysis Elements	5
3.1	Component Model	5
3.1.1	Components	5
3.2	Scenarios	7
3.3	Use Cases	10
4	Design Elements	11
4.1	Interfaces	11
4.2	Messages	12
4.3	NextGRID data Types	12
4.4	Metadata	12
4.5	Grid Patterns	12

3 Analysis Elements

3.1 Component Model

Figure 1 is a part of the proposed Component Model for the NextGRID Architecture. It is specifically oriented towards Grid Business Models and it is based upon the workflow diagram produced in the work of the Dynamics Work Package (WP5). A number of facilities described in NextGRID Conceptual Architecture are addressed in this component model. The current level of analysis of these features is to recognize them, they will be refined in later iterations of the Model.

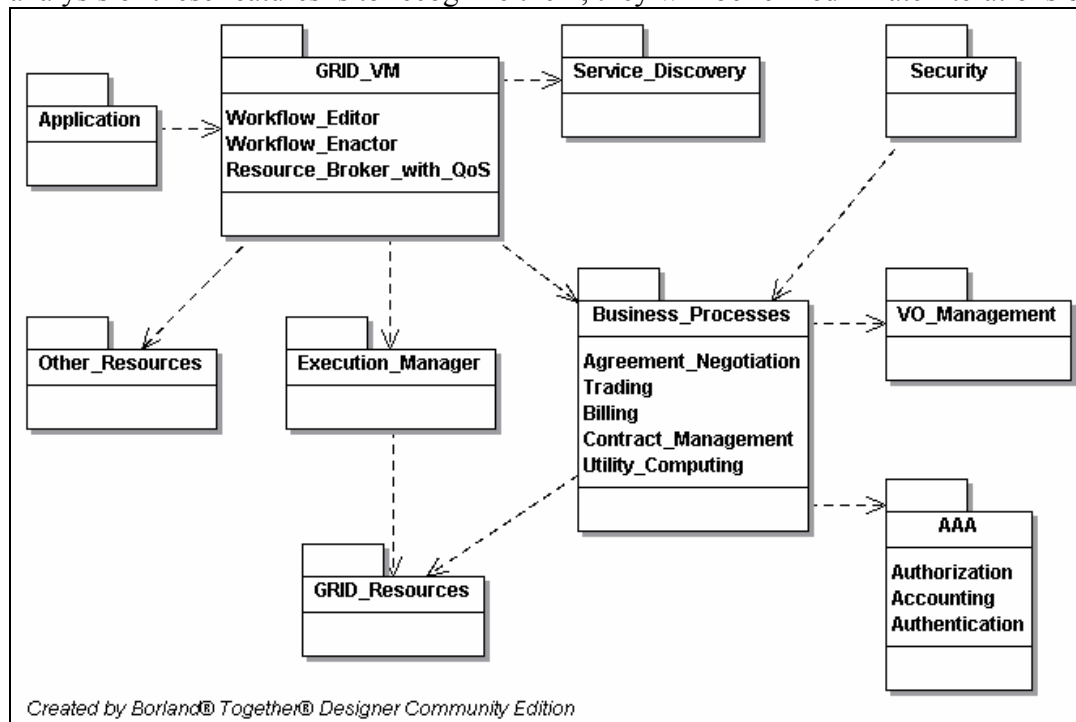


Figure 1. Component Model

3.1.1 Components

- **Application:** A Grid enabled application. Each application should be able to provide input for a grid-enabled service.
- **GRID VM:** This is the Grid workflow enabled Virtual Machine. It is an abstraction that serves the client and presents a "single system image" to what is a distributed Grid environment. In this part of the grid in which all the required actions for the support of dynamic workflow are conducted. This part also ensures the quality of service. The sub components of the GRID VM are the Workflow Editor, the Resource Broker with QoS, and the Workflow Enactor

- **Service discovery:** This component is responsible for the discovery of new services as well as for the storage of information about existing ones.
- **Security:** This component is responsible for providing information about the security methods used for each service that needs to be contacted. When a component needs to communicate with a service the Security Component is contacted in order to provide all the required information and mechanisms.
- **Other Resources:** This component incorporates all the services and resources that are not grid (middleware) resources.
- **Execution Manager:** This component includes all the existing or other managers for the execution of services at the lower level of the grid middleware. Possible candidates are the UNICORE execution manager or the EGEE scheduler, but it is possible to use any kind of grid execution manager.
- **Business Processes:** This component is responsible for the communication with the resources in order to reach an agreement. It is also where the knowledge about the different ways to communicate with a resource is stored. This component also communicates with the VO management component to acquire information from VOs and also with the AAA component for authorization, accounting and authentication issues needed for the service communication.
- **VO Management:** This component provides information about the specific VO it manages. It is used where the selected service is part of this VO. It provides the user role in this VO as well as the VO policies.
- **AAA:** This part is described in the facilities documents ‘Authorization’, ‘Authentication’ and ‘Accounting/Auditing’.
- **GRID Resources:** This component represents all the Grid enabled resources with installed middleware. These resources could be execution resources, data resources or other grid services.

3.2 Scenarios

Three general scenarios using the Component Model are presented. These scenarios were generated following feedback from liaison with the AkoGriMo IST. Summaries of the scenarios follow below, full details of each scenario are provided on the NextGRID BSCW repository.

- In the Generic Scenario, a user issues a workflow, which might be simply the application data that triggers the whole procedure in the Grid Environment. The Grid Virtual Machine interacts with the Business processes in order to initiate processes such as VO Management, Authentication, Authorization, and Accounting as well as Grid Resources. See *Figure 2. Generic Scenario*.
- The Full Scenario involves the procedure that is employed when the user-issued workflow employs business processes. Numerous NextGRID facilities are involved in this scheme. See *Figure 3. Full Scenario*.
- The Business Process Scenario is derived from the use cases in the Information Business Models document. See *Figure 4. Grid VM Scenario*.

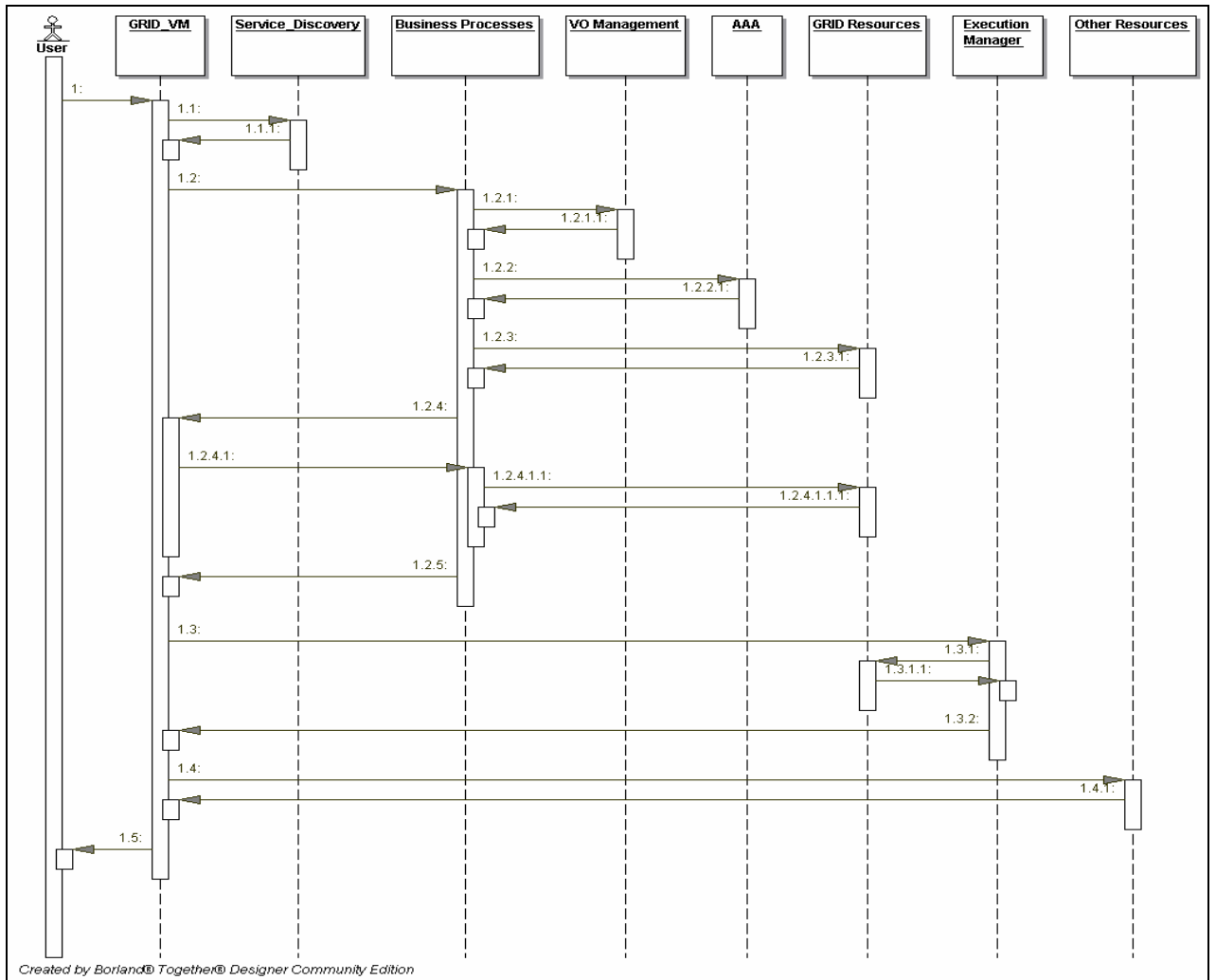


Figure 2. Generic Scenario

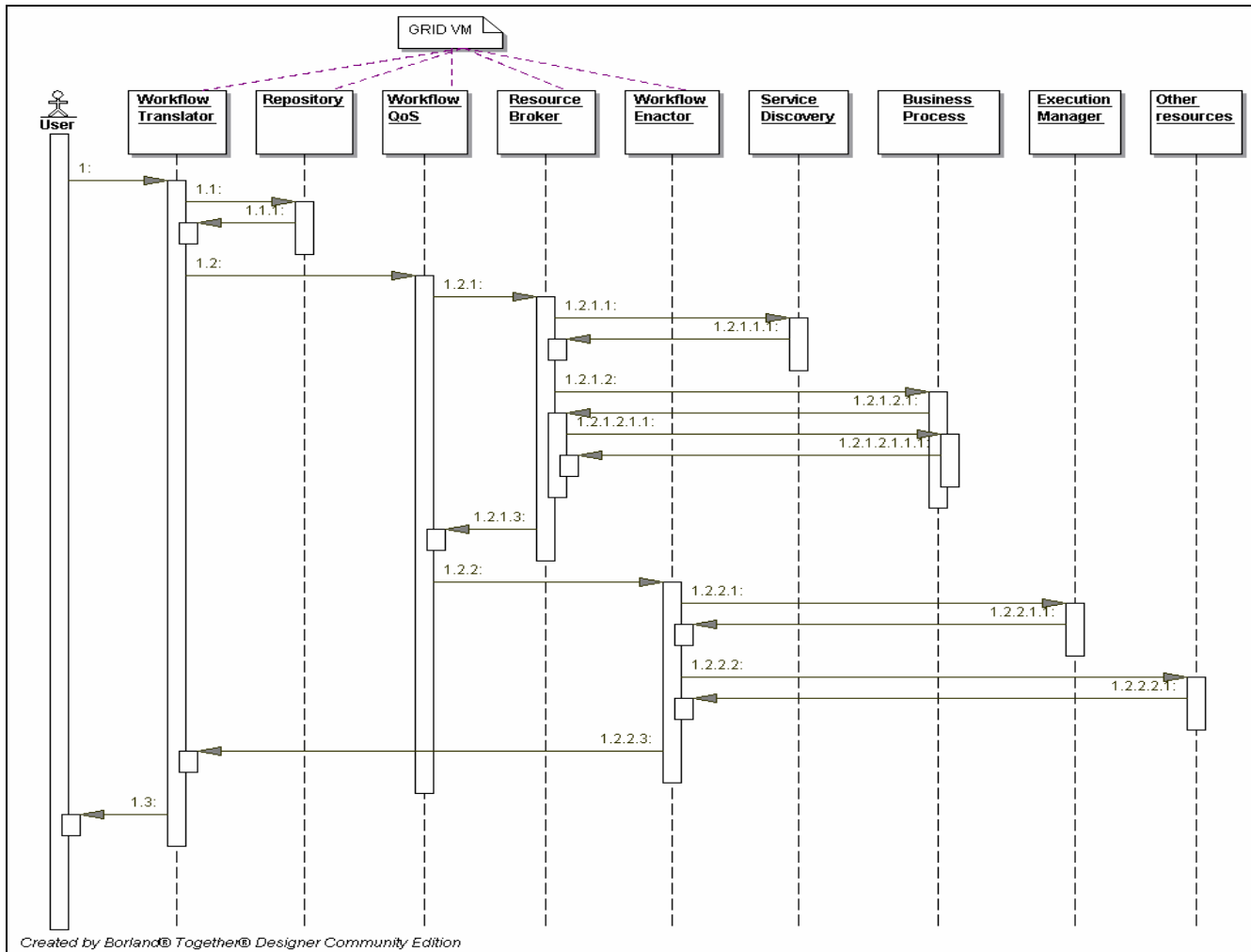


Figure 4. Grid VM Scenario

3.3 Use Cases

Further analysis of the Scenarios lead to a number of use cases. Each use case is provided in detail in a table along with accompanying diagrams and can be found in the NextGRID BSCW Analysis Repository. They are based on the flow of the work in the Component model and they have been designed using Together Design Community Edition. The following is a list of these use cases with a short description.

- **User Submission:** User submits the job to the system with the job parameters and the job requirements for resources, QoS, etc.
- **Service Information Retrieval:** The GRID VM communicates with the Service Discovery to get information about available and supported services.
- **Abstract Workflow Input to QoS:** The Workload Enactor sends the abstract workflow to the Resource Broker with QoS component, which creates a concrete workflow meeting the user's requirement.

- **Workflow QoS Interaction with Resource Broker:** The Workload QoS component contacts the GRID VM Resource Broker to obtain QoS quotes with respect to the required services.
- **Resource Broker Interaction with Service Discovery:** The Resource Broker contacts the service discovery to find available services.
- **Utility Computing Interaction with Resource Broker:** The Resource Broker contacts the Utility Computing, which is part of Business Processes, to obtain generic provision meeting business process requirements.
- **Agreement Negotiation Enactment:** The Utility Computing contacts the Agreement Negotiation component, also part of Business Processes, to contract for resource access.
- **Resource Broker Interaction with Utility Computing:** The Resource Broker contacts the Utility Computing component, part of Business Processes, to obtain a list of dedicated service instances with associated agreements.
- **Contract Management Enactment:** The Utility Computing contacts the Contract Management component, part of Business Processes, to track the evolution of a contract (agreement) for service access and utilization.
- **Workflow Enactor Enactment:** The Resource Broker with QoS contacts the Workflow Enactor to perform service workflows.
- **Execution Manager Enactment:** The Workflow Enactor contacts the Execution Manager to perform individual operations or execute tasks.
- **Resources Enactment:** The Workflow Enactor contacts the Other Resources to provide services outside the Grid environment itself.

4 Design Elements

4.1 Interfaces

- **Token Manager:** The Token Manager is used to manage security tokens. It may be used to manage the tokens for an entire virtual organization (VO), or those associated with only a single site of a larger VO, or those associated with only a single user.
- **Registry:** A service registry is used to tabulate services in order to facilitate their discovery by clients wanting to use the service.
- **Metadata Manager:** The Metadata manager is used to manage metadata for a given service. This includes making the service's own metadata available to other services, as well as obtaining and managing the metadata of other services.
- **Secure Context Manager:** The Secure Context Manager is used to manage security contexts, i.e., to establish a secure channel for exchanging confidential information. This is accomplished by enabling the end-points to agree on a shared, secret key for encrypting all messages.



4.2 Messages

In contrast to the standard SOAP message format, which consists of a textual header and an unstructured body, messages in NextGRID use a framing protocol, which admits structure in both the header and body elements.

4.3 NextGRID data Types

The opaque data types defined by NextGRID are meant to serve as place holders for platform specific data types. When the NextGRID architecture model is mapped onto a platform specific model, then these opaque types will take on specific types.

4.4 Metadata

- **Policy:** The policy metadata describes a service's policies in the form of a structured document.
- **Service Description:** The service description metadata describes the interfaces supported by a given service in the form of a structured document.

4.5 Grid Patterns

- **Dependency Lookup:** Demonstrates how a service publishes its description document to a registry where the description can be found by a client wanting to use the service.
- **Policy Based Security:** Demonstrates how a client accesses a service's security policy and then uses the information therein to construct messages which conforms with that policy.
- **Secure Context:** Demonstrate how a context can be established for the purpose of securely exchanging multiple messages between a client and a server.