Software Requirements Specification for the InterGrid Architecture

Marcos Dias de Assunção
Grid Computing and **D**istributed **S**ystems (GRIDS) Laboratory
The University of Melbourne, Australia
marcosd@csse.unimelb.edu.au

August 6, 2007

Contents

Revision History

Name	Date	Reason For Changes	Version
Marcos	04-08-2007	N/A	0.1

1 Introduction

1.1 Purpose

This document presents an overall description of the InterGrid architecture and its main software requirements. The InterGrid aims at enabling Grids to interconnect with one another through InterGrid Gateways (IGGs) that mediate access to the Grid resources. This way, included are the specifications of the Resource Manager Agent employed by the provider sites, the Distributed Virtual Environment (DVE) Manager used by clients to instantiate a DVE and the InterGrid Gateway that represent a Grid, or organisation, member of the InterGrid. Some of the system aspects presented here are based on a system model already implemented in GridSim. This is the version 0.1 of the requirements specification.

1.2 Document Conventions and Common Terms

Throughout the document, some acronyms and conventions are used. The list of common terms and their acronyms is presented as follows:

- Resource Management Agent (RMA): Responsible for managing the resources at the level of an individual resource provider.
- Distributed Virtual Environment (DVE): It is an execution environment or a network of virtual machines that can span more than one resource provider. This can be viewed as a virtual Grid.
- DVE Manager: Component of the architecture responsible for initiating a DVE by instantiating the virtual machines at the resource providers and setting up the services required by applications.
- InterGrid Gateway (IGG): An IGG is assigned with limited provisioning rights over shares of resources provided by resource providers within an individual Grid. An IGG can exchange shares with other IGGs or acquire shares from other IGGs based on the peering agreements established between them.
- Slot: corresponds to a resource (physical or virtual) with a given capacity.

1.3 Intended Audience and Reading Suggestions

This document is intended for software developers, documentation writers and for general discussions on the implementation decisions regarding the InterGrid architecture.

1.4 Product Scope

The InterGrid, as a final product, is expected to enable the creation and management of execution environments – here also termed DVEs – composed of resources

from multiple resource providers. The IGGs enable the allocation and exchange of resources across Grids. The main idea is to blend virtualisation technology, distributed execution environments, and gateway based resource allocation, thus enabling the instantiation of DVEs that can provide a look and feel of a dedicated infrastructure to applications such as scientific workflows.

1.5 References

- For more information regarding the overall InterGrid architecture we refer to [1].
- The work in [2] describes the resource provisioning at a resource provider site.
- The assertions issued by resource providers and a simple mechanism used between the IGGs is presented in [3].

2 Overall Description

2.1 Product Perspective

The InterGrid *should* leverage existing Grid and virtualisation technology. However, it has to evolve to enable the exchange of resources among Virtual Organisations (VOs) or Grids for the creation of execution environments. Figure 1 presents an abstract view of the InterGrid architecture, its main components and summarises their interactions.

A resource provider provides shares of resources to a Grid by registering the available resources as slot assertions at the IGG; this registration is performed by its RMA. A client application can require resources by requesting them from a DVE Manager; the client also provides additional information about the configuration to be performed once the resources are obtained along with required services. The IGG mediates or exchange resource shares with other IGGs based on the needs of the applications in an individual Grid and the peering policies. Once the resources are obtained, the DVE Manager initialises them and deploy the required services.

The InterGrid expects a minimum set of features from the RMA. The resources provided by to the InterGrid can be physical or virtual resources (i.e. virtual machines). Therefore, it is expected that the RMA is able to collect information from the resource provider and based on the provisioning decisions of the later, issue assertions stating how many and when the resources will be available to the Grid. Also, when a DVE Manager presents a permission to the RMA, the latter must be able to allocate the resources and initialise them. This can correspond to fetching the image required to initialise a virtual machine and carrying out initial network configuration. Although we assume that this is the context in which the InterGrid will operate, nothing impedes one from using resource managers such as Aneka to

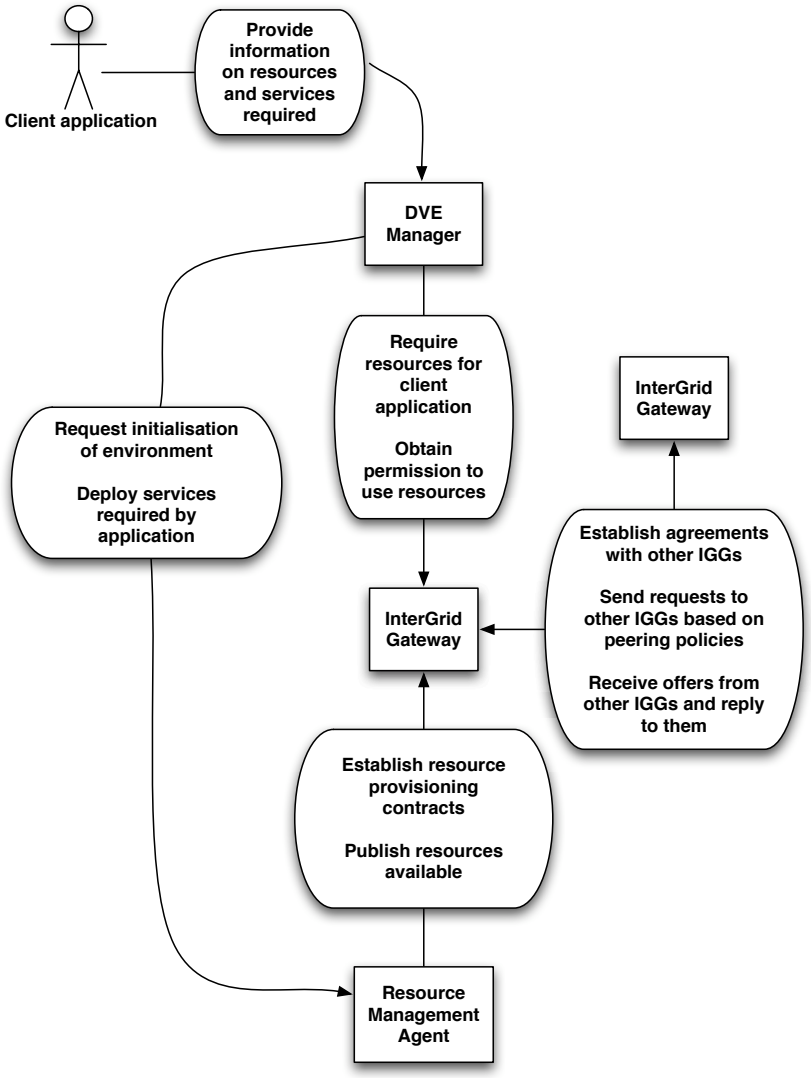

Figure 1: Abstract view of the InterGrid architecture.

reserve, allocate nodes and deploy the required services. However, this can limit the expected features of the InterGrid as discussed later in this document.

2.2 Product Features

The major features of the InterGrid are organised according to its main components. The features of the RMAs are as follows:

- Collect information from the resource management system at the provider site.

- Publish availability of resources to the gateways as slot assertions based on the provider's provisioning decisions.
- Handle resource use permissions given by the IGG to DVE Managers.
- Initialise resources and perform initial host and network configuration.

The main features of InterGrid Gateways are described as below:

- Receive assertions and update the slot inventory.
- Select and assign resources to DVEs based on the Grid-level provisioning policies.
- Negotiate upon and acquire resources from other IGGs.
- Provide resources to other IGGs based on the IGG's provisioning policies.

DVE Managers present the following features:

- Handle requests from a client application.
- Acquire resources from the InterGrid.
- Contact the RMAs at the resource provider sites.
- Deploy services on the resources allocated.
- Manage resources of a DVE and trigger allocation of additional resources or release allocated resources.
- Destroy a DVE.

2.3 User Classes and Characteristics

The users of the InterGrid can be classified in different ways. First, the users (i.e. client applications) can be classified according to the application life span. Applications can be long-lived and short-lived. Second, the applications can be classified with respect to the resources required. Some applications can explicitly require a cluster of nodes that must be provided by an individual resource provider; others can require resources or clusters from multiple providers; whereas other applications can require resources from multiple Grids. Other applications can require clusters of resources regardless their location (i.e. they do not explicitly specify to which Grid or resource provider the resources belong). Finally, the applications can be classified according to their QoS requirements; the requirements can be hard or soft.

2.4 Operation Environment

The IGG *should* provide Web Services interfaces and be implemented in Java. The RMA *should* expose its features as Web Services interfaces and should utilise free virtualisation technologies such as Xen and QEMU. The security infrastructure *could* rely on X.509 certificates by leveraging Grid Security Infrastructure (GSI). Messages exchanged by the components of the architecture should follow Grid standards provided by Open Grid Forum when applicable.

2.5 Design and Implementation Constraints

The RMA *could* rely on Aneka. However, the use of Aneka can limit the features of the RMA because the former does not support virtualisation, performance isolation among other features.

3 System Features

This section contains the main features of the proposed system. This is a preliminary list of features, which should be further improved.

3.1 Registration of Resource Assertions

This feature corresponds to the registration of resource assertions by the RMA in the IGG. Figure 2 presents the use case for this feature.

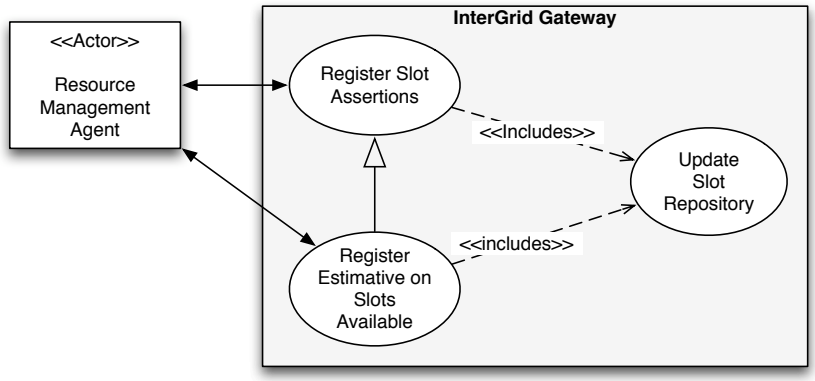

Figure 2: Registration of resource assertion.

Stakeholders and Interests:

- RMA: Wants to inform the IGG gateway about the slots (i.e. resources) available at the resource provider.

- IGG: Wants to update the repository with the information provided by the RMA.

Assumptions:

- The scenario considers that a trustful relationship between the RMA and IGG has already been established.
- The format for the slot assertions has been defined.
- The communication protocol used between RMA and IGG has been defined.
- RMA is aware of the resource provider's provisioning decisions.

Pre-Conditions: RMA has obtained information on resource available from the resource management system utilised by the resource provider. This information includes the number of slots available, their characteristics and the period of time over which they will be available.

Use Case Initiation: The registration process can start when (a) there are resources available to be used by the Grid and there are no outstanding assertions (i.e. RMA has not issued any assertion for the slots available beforehand); (b) a previously issued slot assertion is about to expire.

Use Case Dialog:

1. RMA, based on information obtained from the resource manager, issues a slot assertion and sends it to the IGG.
2. If the (start time + duration) of the assertion is smaller than (current time + minimum time to initialise resources), IGG rejects the assertion.
3. Else, IGG updates the slot repository with the information provided by RMA.
4. IGG checks if the assertion is issued to renew another assertion previously sent by the RMA.
5. If the assertion is a renewal, then IGG updates the allocations of resources to DVEs.
6. IGG generates a registration ID and returns it to RMA.

Alternatively:

1. Instead of publishing complete information about the slots available, RMA can provide an estimative of slots available.

Use Case Termination:

- The slot assertion is successfully registered.
- The registration may timeout.
- The assertion corresponds to a set of slots already published by the RMA.

Post-Conditions: RMA saves the ID of the registration.

Cancel: Changes to the slot repository or to the RMA's status have to be rolled back.

3.2 Client Application Requires Resources

A user wants to allocate a set of nodes at one (or more) resource provider(s) on which she wants to deploy an application (Figure 3).

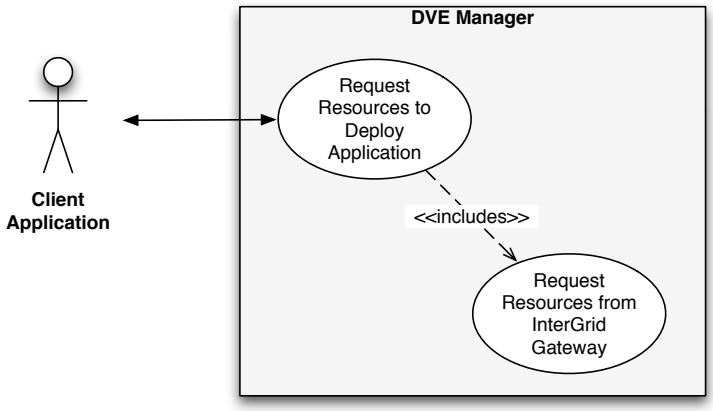

Figure 3: Client application's resource request.

Stakeholders and Interests:

- Client Application (i.e. User): Wants to allocate clusters of resources from resource provider(s) to deploy services and execute an application.
- DVE Manager: Acts on behalf of the client application by converting the client request into slot requests that are sent to the IGG.

Assumptions:

- A language for specifying the configuration of the environment, services and resources has been identified.
- The scenario considers that a trustful relationship between the Client Application and DVE Manager has been established.
- Policies that govern how the DVE Manager requests resources from the IGG have been identified.

Pre-Conditions:

- Authentication has been performed.

Use Case Initiation: A user (i.e. client application) wants to allocate a set of nodes because she wants to deploy a given application, so she contacts the DVE Manager to do so.

Use Case Dialog:

1. User provides information regarding the slots required (i.e. number of slots, types and time duration) and required services to the DVE Manager.
2. DVE Manager extracts the information regarding the slots, and convert it into slot requests.
3. DVE Manager sends slot request(s) to IGG.
4. Once a response from IGG is obtained, DVE Manager informs the client application.

Use Case Termination:

- The request is fulfilled.
- The request timeout is reached.
- The request cannot be accepted.

Post-Conditions: DVE Manager informs the client application.

Cancel: Any partial allocations must be rolled back.

3.3 DVE Manager Requests Slots from IGG

A DVE Manager requests resources from the IGG on behalf of a client application as illustrated in Figure 4.

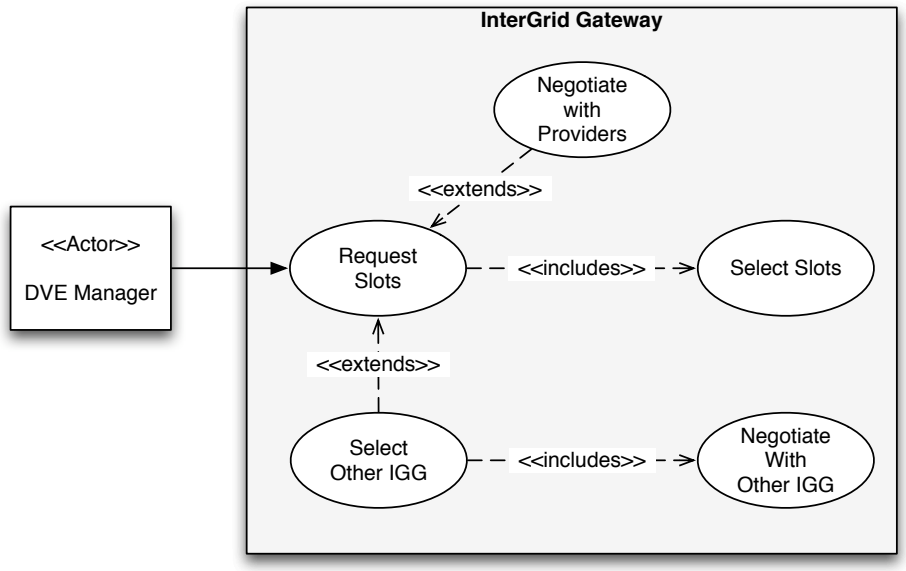

Figure 4: Client application's resource request.

Stakeholders and Interests:

- DVE Manager: Wants to acquire slots necessary to deploy the services required by the user.
- IGG: Has limited provisioning rights over the slots provided and allocates them according to Grid-wide provisioning policies. IGG contacts other IGGs when slots from other Grids are required.

Assumptions:

- The format for the slot requests sent by the DVE Manager to IGG has been defined.
- DVE Manager trusts the IGG.
- IGGs have pre-established contracts amongst them.

Pre-Conditions:

- Authentication has been performed.
- IGG's resource selection policy has been defined.
- IGG's provisioning policies have been defined.

Use Case Initiation: Upon the receiving of user's request (i.e. client application), the DVE Manager tries to acquire slots from the IGG.

Use Case Dialog:

1. DVE Manager creates a request for a set of slots and sends the request to IGG A.
2. IGG A performs admission control by checking whether the DVE Manager is allowed to request the resources or not.
3. IGG A checks whether the allocation will not breach the provisioning policy.
4. IGG A verifies whether there are slots available to serve the request based on its resource selection policy.
5. If there are slots available, IGG A creates a permission to use the slots.
6. Else
   (a) If the IGG has estimations on resources that will be available at resource providers then
      i. Negotiates with resource providers for resources.
      ii. Creates a permission for the resources acquired.
   (b) If the request cannot be fully served by the IGG A
      i. it selects a peering IGG B which can provide the slots required.
      ii. IGG A creates an ID for the negotiation with IGG B and adds it in the outstanding negotiations.
      iii. IGG A makes an offer (i.e. informs the other IGG B about the interest in acquiring slots) and waits for a response.
      iv. If the response is accept, IGG A requests a permission to use slots from IGG B.
7. IGG A returns the permission to DVE Manager.

Use Case Termination:

- The request is fulfilled.
- The request timeout is reached.
- The request cannot be accepted.
- The request is partially fulfilled.

Post-Conditions: DVE Manager triggers the configuration of the resources and the deployment of services.

Cancel: Any partial allocations must be rolled back.

3.4 Handling InterGrid Requests

An IGG receives a request from another IGG and evaluates whether it can serve the request (Figure 5).

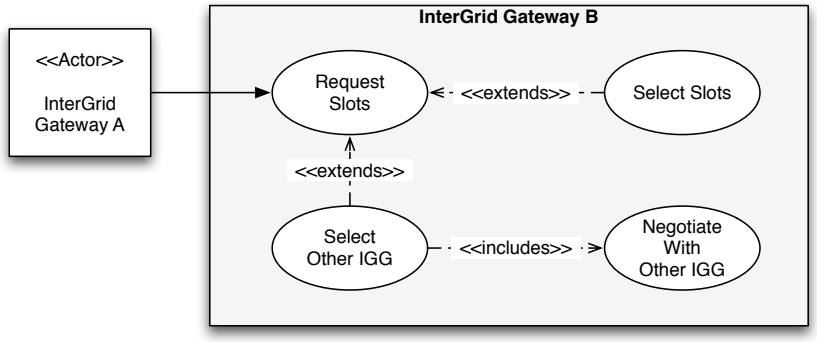

Figure 5: IGG receives a request from another IGG.

Stakeholders and Interests:

- IGG A: Wants to acquire slots from IGG B.
- IGG B: Evaluates the request from IGG A and selects slots.

Assumptions:

- The policies that underlie the peering between the two Grids have been defined.

Pre-Conditions: Authentication has been performed.

Use Case Initiation: IGG A sends a request to IGG B.

Use Case Dialog:

1. IGG B receives from IGG A a request for resources.
2. If the request is related to a negotiation started by IGG B, then return reject.
3. IGG B performs admission control by evaluating whether IGG A is allowed to request that number of resources.
4. If there are slots available, IGG A creates a permission to use the slots.

5. Else

   (a) If the request cannot be fully served by the IGG B, it *may* select a peering IGG which can provide the slots required.

   (b) IGG B makes an offer (i.e. informs the other IGG about the interest in acquiring slots) and waits for a response.

   (c) If the response is accept, IGG B requests a permission to use slots from the other IGG.

6. Slot use permission is returned to IGG A.

Use Case Termination:

- The request is accepted.
- The request timeout is reached.
- The request cannot be accepted.
- The request is partially accepted.

Post-Conditions: The permission(s) received have to be sent to the DVE Manager that request the resources. In addition, the IGG has to update any information regarding slots obtained from peering IGGs.

Cancel: Partial allocations have to be cancelled.

3.5 Initiation of Virtual Machines and Service Deployment

DVE Manager requests to RMA the initialisation of resources, performs initial configuration and deploys required services (Figure 6).

Stakeholders and Interests:

- DVE Manager: Wants presents RMA with a permission to use resources and wants them initialised.
- RMA: Initialises the resources.

Assumptions:

- DVE Manager has already received a slot use permission from the IGG.

Pre-Conditions: Authentication has been performed.

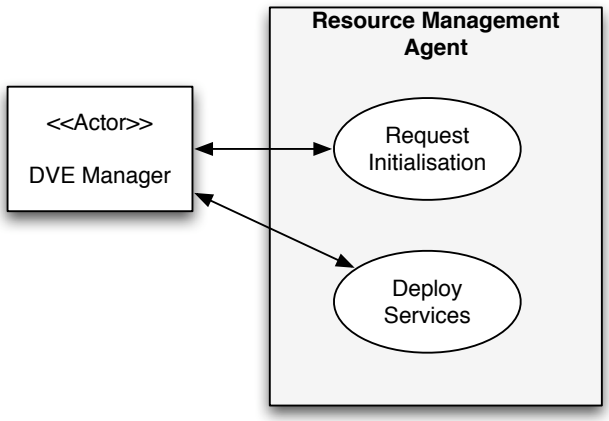

Figure 6: Resource initialisation and service deployment.

Use Case Initiation: DVE Manager requires the initialisation of resources at the resource provider site.

Use Case Dialog:

1. DVE Manager sends the slot use permission to resource provider.
2. Resource provider fetches the virtual machine images from repository, start them and provide initial configuration.
3. Resource provider notifies DVE Manager.

Use Case Termination:

- The resources are initialised.
- The message timeout is reached.
- The resource provider refuses the permission.
- The permission cannot be fully honored.

Post-Conditions: DVE Manager performs the deployment of services.

Cancel: Any resources allocated have to be released.

4 External Interface Requirements

4.1 Communication Interfaces and Protocols

The architecture components should expose Web Services interfaces. The data formats for requests and protocols utilised should follow the standards proposed by the Open Grid Forum.

References

[1] Marcos Dias de Assunção and Rajkumar Buyya. Intergrid: A case for internetworking islands of grids. *Concurrency and Computation: Practice and Experience (CPE)*, (In press, accepted on May 8, 2007).

[2] Marcos Dias de Assunção, Werner Streitberger, Torsten Eymann, and Rajkumar Buyya. Enabling the simulation of service-oriented computing and provisioning policies for autonomic utility grids. In *Proceedings of the 4th International Workshop on Grid Economics and Business Models*, Rennes, France, August 2007. Springer.

[3] Marcos Dias de Assunção and Rajkumar Buyya. A peak load management mechanism for enabling resource exchange amongst grids. Under Review, July 2007.