



Requirements Document: Grid Application Toolkit

Author(s):	Gabrielle Allen, Tom Goodale, Thomas Radke, Ed Seidel
Document Filename:	GridLab-1-D.1-0001-GAT-Requirements
Work package:	Grid Application Toolkit
Partner(s):	PSNC,ZIB,MU,SZTAKI,VU,ISUFI,CARDIFF,GRIDWARE, COMPAQ,NTUA
Lead Partner:	MPG
Config ID:	GridLab-1-D.1-0001-1.0
Document classification:	INT

Abstract: This document presents the requirements for WP1 – Grid Application Toolkit – of the Gridlab project. General requirements for users and developers of the Gridlab project are presented, and from these specific requirements necessary for WP1 are derived. These in turn lead to requirements which must be met by other workpackages.





Contents

1	Introduction	2
1.1	Definitions and Assumptions	2
2	General Gridlab Requirements	2
3	Developer Requirements	3
3.1	Application Developers	3
3.2	Service Developers	4
4	Grid Application Toolkit Requirements	4
5	Requirements which other workpackages must meet for the GAT	4
5.1	All workpackages	5
5.2	WP-5 – Testbed Management	5
5.3	WP-6 – Security	5
5.4	WP-13 – Exploitation and Dissemination	5

1 Introduction

This document presents the requirements for WP-1 – Grid Application Toolkit (GAT) – of the Gridlab project. The requirements are split into four parts – general requirements, developer requirements, the resulting GAT requirements, and, finally, the requirements these place on other workpackages within the GridLab project.

These requirements will be used to design the GAT used by the GridLab project.

1.1 Definitions and Assumptions

- **End User**

An End User of the GridLab environment is a user who does not write any software. The user's interaction with the environment will either be via command line tools provided by the project or the portal.

- **Application Developer**

An Application Developer is a software author who develops software which the end users interact with directly. E.g. simulation codes, portals, command line tools to examine the environment.

- **Service Developer**

A service developer is a software author who develops software which the end user has no direct interaction with. This software is accessed via defined interfaces by application developers.

2 General Gridlab Requirements

Experiences in the Grid community and examination of the annex lead to the following.

An end user in a grid environment requires:

RU-1 An application oriented environment.

RU-2 To be able to run applications on resources of one or more virtual organisations. This must accommodate firewalls, all architectures (e.g. Windows XP, Fujitsu, ...).

RU-3 Flexible, easy-to-use, simple interfaces to resources, jobs, and data (including compiling, tracking jobs, cataloguing data).

RU-4 A means to make efficient and effective use of resources.

RU-5 To be able to run in a trustable, secure, environment. E.g. secure access to machines, data transfer, software, etc.

RU-6 Fault tolerance.

RU-7 The ability to work in a disconnected environment.

RU-8 Mobile working.

RU-9 The ability to run in environments as minimalistic as laptops with no grid infrastructure to fully deployed Virtual Organisations (VOs).

RU-10 Easily deployable infrastructure.

RU-11 Complexity should be hidden as much as possible.

- RU-12** A well supported software and environment (e.g. documentation, user training, examples, helpdesk).
- RU-13** The interfaces must be informative and transparent, with good error and status messages as well as access to audit trails.
- RU-14** Provide reproducible and consistent behaviour.
- RU-15** Provide a collaborative infrastructure.
- RU-16** Enable the user to have absolute control or delegate control to the grid environment.
- RU-17** The infrastructure must cater for all classes of applications, from lightweight to largescale.
- RU-18** The infrastructure must be extensible and future proof.
- RU-19** Clear, clever, customisable and extensible (graphical) user interfaces (e.g. skins support).
- RU-20** The infrastructure must provide capabilities to customise choice of service implementation (e.g. using efficiency, reliability, first succeeding, all).
- RU-21** The infrastructure should enable multiple services providing the same functionality to be used. E.g. multiple resource managers each dealing with a specific set of (possibly overlapping) machines.

Additionally GridLab users may require:

- RUO-1** Open Source distribution of application grid software
- RUO-2** National language support (NLS).

3 Developer Requirements

3.1 Application Developers

A developer for applications in a grid environment requires the following to meet the above requirements:

- RAD-1** An application oriented environment
- RAD-2** A “Grid Application Toolkit” (requirements listed in §4).
- RAD-3** Easy access to a Grid Testbed providing Grid resources (machines, Grid software components and services). The testbed must be stable but also able to simulate real-world behaviour with all pathological cases of failure conditions.
- RAD-4** The infrastructure must allow fast prototyping, testing and use of applications and services.
- RAD-5** Guidelines for making effective use of grid environment. E.g. adding checkpointing, portability, fault tolerance capabilities to applications.
- RAD-6** Must be able to develop applications as independently of service development as possible.
- RAD-7** It must be easy to grid-enable legacy codes

3.2 Service Developers

Service developers additionally require:

RSD-1 They should be able to make use of services from other projects easily (although this could simply mean including these as other GridLab services).

RSD-2 They must be able to develop services as independently of other service development as possible.

4 Grid Application Toolkit Requirements

The resulting requirements the Grid Application Toolkit must meet are:

- RGAT-1** Portability.
- RGAT-2** Easy to build and deploy.
- RGAT-3** Thread safe.
- RGAT-4** Robust and fault tolerant.
- RGAT-5** A consistent API (i.e. naming scheme, function calls, language bindings).
- RGAT-6** Light-weight.
- RGAT-7** Support for all relevant programming languages.
- RGAT-8** Provide error and diagnostic features (logging capability).
- RGAT-9** Developer documentation and examples.
- RGAT-10** Work in any environment (mobile, disconnected, ...).
- RGAT-11** Contain test suites to verify the functioning of each GAT component individually and collections of components.
- RGAT-12** Make use of and enforce security policy.
- RGAT-13** Provide transparent access to security infrastructure (e.g. multiple credentials).
- RGAT-14** Allow applications to be developed and deployed irrespective of the actual state of the grid.
- RGAT-15** Short, simple, clear APIs which allow both access to full functionality of capabilities and access via more abstract calls.
- RGAT-16** Provide extensibility (e.g. provide capabilities through dynamic, plug-in architecture possibly with multiple implementations).
- RGAT-17** Decision making, intelligence and fault tolerance at all levels.
- RGAT-18** The ability to discover and use services dynamically.
- RGAT-19** APIs for monitoring.
- RGAT-20** Instrumented to enable audit-trail functionality.
- RGAT-21** Open source.

5 Requirements which other workpackages must meet for the GAT

The above requirements drive a set of requirements which other workpackages must meet. There are general requirements for all workpackages, and requirements for specific workpackages.

5.1 All workpackages

RWPA-1 Must be able to utilise multiple virtual organisations.

RWPA-2 Short, simple, clear APIs which allow both access to full functionality of capabilities and access via more abstract calls.

RWPA-3 Where possible, APIs provided by services should have a clear correspondence to those provided by the GAT.

5.2 WP-5 – Testbed Management

RWP5-1 Should be able to simulate a “real-world” grid environment – e.g. simulate failure modes.

RWP5-2 Heterogeneous – should contain as many different architectures as possible.

5.3 WP-6 – Security

RWP6-1 Support for multiple credentials.

RWP6-2 Fail-safe.

RWP6-3 Security policy must support disconnected operation.

RWP6-4 Support/accommodate collaborative working practices.

RWP6-5 User control of security policy (e.g. GSI, PKI, local services).

RWP6-6 Clear instructions and APIs.

5.4 WP-13 – Exploitation and Dissemination

RWP13-1 Clear documentation.

RWP13-2 Examples.

RWP13-3 HOWTOs.

RWP13-4 FAQs.