
# Requirement Analysis

| | |
|---|---|
| Author(s): | André Merzky, Florian Schintke and Thorsten Schütt |
| Document Filename: | GridLab-08-RA-0001-RequirementAnalysis |
| Work package: | Data Management and Visualization |
| Partner(s): | ZIB |
| Lead Partner: | ZIB |
| Config ID: | GridLab-08-RA-0001-1.0 |
| Document classification: | PUBLIC |

**A**bstract: This documents describes the requirements for WP8 of the GridLab project - "Data Management and Visualization". It is based on user scenarios as provided by the application work packages, and on the requirements of WP1 - "Grid Application Toolkit (GAT)".

# Contents

# 1 Scope of this Document

This document will derive the requirements to the GridLab WP8 – Data Management and Visualization. For that, we first review the user requirements, application developer requirements and requirements of all work packages depending on WP8. From these, we derive our own set of requirements.
Requirements in the sense of this document include

- functionality requirements,
- design requirements,
- architectural requirements.

These have to fit into the overall scope of the GridLab project.

The User Requirements and Application Developer Requirements as listed in this document are not specific to WP8, but hold generally for all WPs of the GridLab project. They have been derived from discussions with end users and application developers (Cactus Team, Triana Team), and from discussions with WP1 developers (GAT).

The requirements to WP8 as identified in this document have to be met by the WP8 architecture, WP8 design and WP8 implementation. The detailed plans for these are described in separate documents.

## 1.1 Terms used

Terms used throughout this document (as requirement, application, work package, architecture etc.) are used in the sense as described in Annex 1 of the GridLab project proposal or as defined by the GridLab technical Board.

Some terms are described in the next subsection, mainly concerning terms specific to WP8.

## 1.2 About Data

Throughout this document, we talk about data, data sets, data files, checkpoint files and so on. These terms are defined in this section.

**Data** Data are information; raw facts. Data can be input into a program and processed in various ways. Data can be produced by by programs, devices and user input (via devices). Data can be *volatile* (have a very short lifetime, e.g. inside the main memory of a computer) or *persistent* (have an unlimited lifetime, e.g. in a archived file).
Data can be distributed, e.g. stored in files located on different storage systems, or live inside of an application running on multiple resources.
Data are often structured.
In this document, the term *data* refers to any one of: *data set*, *data file* or *collection of data files*.

**Data Set** Data with collective appearance are often gathered in data sets. Data in data sets usually share a common set of meta data, and are of the same or complementary types.

**Data File** A File is an entity living on a storage system, containing a stream of bytes, and can be addressed by a unique name. Often, files are annotated with Meta Data as time of creation, owner, security information and size. Data Files are files containing data sets.

**Data Stream** A Data Stream is a flow of Data from A to B. It is similar to a Data File but in general allows no seek operation. Stream can be converted to files (caching, buffering) and vice versa (dumping).

**Collection of Data Files** Data Files sharing the same set or a similar set of Meta Data can be gathered in Collections. Collections are not physical entity, but a virtual object containing information about Data Files contained in the Collection.

**Meta Data** Meta Data are data about Data Sets, Data Files and Collections. Meta Data can contain most different information, from size of data over information about data structures to arbitrary annotation strings provided by a user or application.
Meta Data are usually stored in Meta Data Directories (MDS). Data Management Services (DMS) provide operations on Data, Data Files, Data Sets and Data Stream and guarantee a consistent update of the Meta Data entries in the MDS.

**Checkpoint Data** Checkpoint Data are Data Sets created by an application or system utility. They usually contain all information necessary to completely restore and restart an application in a status similar to the one it was in before creating the checkpoint data.

**Checkpoint File** Checkpoint Files are Data Files containing Checkpoint Data. Checkpoint Files can be part of a Collection containing the complete set of Checkpoint Data of an application.

**Migration** An application moving during runtime from one resource to another is performing a Migration. This process does also involve the movement of Data (usually Checkpoint Files) in the same direction. This subprocess is called Data Migration.

As this document is concerned about the management and visualization of data, we will also mention a number of possible data sources. Of main interest to the project are data produced by applications or experiments. Also of interest can be data produced or gathered by middleware services or systems, like monitoring systems, information systems. All this data could potentially also visualized, with major emphasis on visualization of application data.

# 2 GridLab Requirements and User Scenarios

From the Annex 1 and elsewhere given user scenarios, and from discussions with application developers, numerous general requirements have been identified and must be met by the general GridLab infrastructure. These are listed in the requirements document of WP1. Together with the user requirements and application developer requirements also listed there, these form the basis of the requierements as identified here.

This section reviews the user scenarios, and extracts all information relevant to data management and visualization. From these information, and from other general user requirements as listed in the WP1 document, the requirements for WP8 are derived.

### Data Management Scenarios (DMS):

From the application scenarios of Annex following Data Management situations/processes are to be enabled by the GridLab project:

**DMS1:** migration of data files from A to B,

**DMS2:** accompanied selection and migration of data files from A to B,
**DMS3:** fast transport of data sets/files from A to B,
**DMS4:** discover data sets,
**DMS5:** locate data sets,
**DMS6:** archival of data files,
**DMS7:** recombination of parted data sets/files,
**DMS8:** requests information about a data set or data file and its contents/history/...,
**DMS9:** requests information about a storage system and its optimal I/O parameters/variables,

### Visualization Scenarios(VS):

From the application scenarios of Annex one follow following Visualization situations/processes are to be enabled by the GridLab project:

**VS1:** visualization of past data sets,
**VS2:** visualization of online data,
**VS3:** visualization *output* for further use,
**VS4:** visual interaction with simulation code at runtime, and
**VS5:** visual interaction with the Grid environment.

## 3 Requirement Analysis for WP8

From the requirements listed in the previous sections, it is possible to derive a well defined set of requirements, which have to be met by WP8 architecture, design and implementation[1]. It is also possible to derive a well defined set of requirements to all other work packages, on which WP8 depends.

### Requirements which WP8 has to meet

#### Functionality Requirements (FR):

Following functionality must be provided by WP8:

**FR1:** migration of data,
**FR2:** archival of data,
**FR3:** replication of data,
**FR4:** recombination of distributed/parallel written data,
**FR5:** annotation of data,
**FR6:** location of data[2],
**FR7:** discovery of data[3],
**FR8:** remote (online and offline) visualization of data,
**FR9:** adaptive visualization (progressive, hierarchical) of data,
**FR10:** data transformation[4],

[1]Please note, that a data set in the following list can mean either (a) a volatile data set living inside an application, or (b) a data file, or (c) a annotated collection of data files.
[2]Location: For a known data, return the exact location of the data.
[3]Discovery: for a set of attribute-value pairs, find a matching set of data.
[4]transform data from format A to format B

**FR11:** data extraction[5], and
**FR12:** secure data transport[6].

**Quality Requirements (QR):**

Following quality constraints must be respected by WP8:

**QR1:** be application oriented,
**QR2:** be usable on all types of resources,
**QR3:** be usable in firewalled environments,
**QR4:** be usable in disconnected environments,
**QR5:** be usable in minimalistic environments,
**QR6:** support and enforce use of security policies,
**QR7:** support synchronous and asynchronous operation,
**QR8:** provide abstractive interfaces,
**QR9:** provide complete set of interfaces,
**QR10:** provide ability to be discovered,
**QR11:** provide ability to discover and use services dynamically,
**QR12:** provide audit trails and verbose error messages,
**QR13:** provide a test suite,
**QR14:** be well documented,
**QR15:** be extensible and 'future proof',
**QR16:** be informative, transparent,
**QR17:** be lightweight (where possible),
**QR18:** be robust and fault tolerant,
**QR19:** be adaptive to variable Grid infrastructure,
**QR20:** be able to recover from interruptions,
**QR21:** be independent from other services (if possible or necessary),
**QR22:** behave consistent and reproducible,
**QR23:** allow integration of 3rd party software/services, and
**QR24:** allow instrumentation for monitoring purposes.

# 4 WP8 Requirements to other Work Packages

To meet the requirements identified in the previous section, the work of WP8 has to be able to utilize other services. Certain functionality and properties are required *from* these services in order to achieve a successful outcome of our work. These requirements are listed below. We distinguish between requirements we *must have* in order to be able to do our work at all, and requirements which would be *nice to have* and which would ease our work significantly[7]. If features in the ENR section cannot be provided by the respective WPs, WP8 has to provide them itself, but possibly tailored to WP8 needs.

[5] transform data set A into subset B
[6] including authorization, authentification and encryption
[7] Nice-to-have can also mean that the architecture or design would be simpler and cleaner if the respective requirement is met by the other WP.

### External Requirements we MUST HAVE (EMR):

Please note that requirements to WP7 (Adaptive Components) are dealt with in a separate document issued by WP7: "Use Cases for Adaptive Components" (Title: GridLab-7-UCR-1-UseCasesReport - deliverable 7.1).

**EMR1:** WP5 simple access to testbed resources,
**EMR2:** WP6 simple interface to security functionality,
**EMR3:** WP6 support for multiple credentials,
**EMR4:** WP6 secure channel from A to B, not encrypted,
**EMR5:** WP10 ability to store custom meta data,
**EMR6:** WP10 ability to search meta data by regex matches to values in key-value pairs,
**EMR7:** WP11 hooks for audit trails.

### External Requirements which would be NICE TO HAVE (ENR):

**ENR1:** WP5 ability to simulate resource failures,
**ENR2:** WP5 installation support,
**ENR3:** WP6 secure encrypted channel from A to B,
**ENR4:** WP10 regex searches for keys for meta data,
**ENR5:** WP11 network forecast for bandwidth and latency.
**ENR6:** WP11 Monitor Storage Systems (space)

## 5 Outlook

Having identified these requirements, the next step is to design an architecture for the working package, which (a) fits into the global GridLab architecture and (b) meets all requirements above. Some requirements may not be enforced by that architecture, but have to be met at implementation level. The implementation plan for Work Package 8 should cover the identified requirements completely.