

EUBrazilCC

EU-Brazil Cloud infrastructure Connecting federated resources for Scientific Advancement

D3.3 Prototype of the CSGrid Adaptation Mechanisms

Contract number:	FP7-614048 / CNPq 490115/2013-6
Start Date of Project:	1 October 2013
Duration of Project:	28 months
Work Package:	WP3 – Infrastructure
Due Date:	M10 – 31/01/2015
Submission Date:	31/01/2015
Partner Responsible for the Deliverable:	PUC-Rio
Dissemination Level:	PU - Public
Nature:	P - Prototype
Author(s):	Maria Julia Dias de Lima (Tecgraf PUC-Rio)
Reviewer(s):	Erik Torres – UPV



Change Log

Version	Date	Description	Author(s)
v0.1	29/01/2015	Initial content	Maria Julia de Lima (Tecgraf PUC-Rio)
v0.2	30/01/2015	Updated based on comments and suggestions from reviewer Erik Torres	Maria Julia de Lima (Tecgraf PUC-Rio)

Document Review

Review	Version	Date	Reviewers	Comments
QA	v0.1	29/01/2015	Erik Torres – UPV	Initial content reviewed
Final	v1.0	30/01/2015	Erik Torres – UPV	Final version edits

Table of contents

Table of contents.....	2
Disclaimer.....	4
Executive Summary.....	4
1 Introduction.....	4
2 RELEASE NOTES.....	7
2.1 Adaptations on the description of execution resource properties	7
2.2 Adaptations on the description of job execution requirements	9
2.3 Adaptations on information about commands	9
2.4 Support to additional execution parameters.....	10
2.5 Provides a plugin mechanism that supports the development of server extensions ..	11
2.6 Support to the execution of SGAs on the same process as the CSGrid server.....	11
2.7 Extends data transfer mechanism between CSGrid server and SGA.....	11
2.8 Migration of the OpenBus SDK from 1.5 to 2.x.....	12
ACRONYMS AND ABBREVIATIONS	13

List of figures

Figure 1 - CSGrid middleware overview 6

Disclaimer

EUBrazilCloudConnect EU-Brazil Cloud infrastructure Connecting federated resources for Scientific Advancement (2013-2015) (hereinafter “EUBrazilCC”) is a Small or medium-scale focused research project (STREP) funded by the European Commission under the Cooperation Programme, Framework Programme Seven (FP7) Objective FP7-ICT-2013.10.2-EU-Brazil Research and Development cooperation, and the National Council for Scientific and Technological Development of Brazil (CNPq) of the Brazilian Ministry of Science and Technology (MCT) under the corresponding matching Brazilian Call for proposals MCT/CNPq 013/2012.

This document contains information on core activities, findings, and outcomes of EUBrazilCC project, and in some instances, distinguished experts forming part of the project’s External Expert Committee. Any references to content in both website content and documents should clearly indicate the authors, source, organization and date of publication.

The document has been produced with the co-funding of the European Commission and the National Council for Scientific and Technological Development of Brazil. The content of this publication is the sole responsibility of the EUBrazilCC Consortium and its experts and cannot be considered to reflect the views of the European Commission nor the National Council for Scientific and Technological Development of Brazil.

Executive Summary

CSGrid provides facilities for resource management and algorithm execution in a distributed and heterogeneous computing environment. In the EU Brazil Cloud Connect Project (EUBrazilCC) project, the CSGrid middleware will provide the base for transparent access to grid and cloud computing infrastructures. In the context of EUBrazilCC, adaptation requirements have been identified as necessary to enhance its flexibility and interoperability, facilitating its integration with different resource management systems for grid and cloud computing, and for improving security. These requirements are described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

D3.3 is a prototype deliverable that is intended to show the scientific and technical feasibility of the requirements identified in EUBrazilCC for the CSGrid software. This document serves to the purpose of accompanying the release of the prototype, facilitating the access links to the implementation and documentation of the prototype. Therefore, the information contained in this document is subject to changes as a result of any changes/updates the CSGrid development team makes in the prototype. The latest version of the information is available at the website of the prototype.

The document is organized as follows. Section 1 CSGrid is briefly contextualized. Section 2 summarizes changes made in CSGrid in order to meet part of the requirements described in the Deliverable D3.1.

1 Introduction

CSGrid provides facilities for resource management and algorithm execution in a distributed and heterogeneous computing environment. The main components of the CSGrid middleware are illustrated in Figure 1.

Typically, a CSGrid server runs on a single machine and is responsible for controlling the resources available to its clients. These resources include the data repositories and the execution hosts to which users submit their jobs. A CSGrid server has access to a number of different execution hosts, possibly encompassing clusters as well as individual machines with specific hardware capabilities. A node daemon, called an SGA, accepts requests for the execution of algorithms and for information on system state and job execution. Upon activation, an SGA registers itself with the CSGrid Server.

OpenBus is a CORBA-based middleware used to integrate multi-platform and multi-language systems based on a service-oriented architecture. It offers a service directory, access control mechanisms and peer-to-peer communication. CSGrid uses the OpenBus to authenticate and publish the following services components:

- **OpenDreams** (OpenBus Distributed Resource and Algorithms Management Service): offers a set of operations for the submission, monitoring and control of jobs on SGA's remote computational resources. It also provides access to the CSGrid user's algorithms configuration, abstracting their parameters and requirements. The OpenDreams interface is based on OGF's DRMAA 1.0 specification.
- **Hierarchical Data Service**: offers a set of operations for accessing and manipulating files in the CSGrid user's project area. User files are organized by projects, and users can share their files in a specific project with other users. Each project represents a data area, organized as hierarchical structures that store related files. All data files needed as input to programs submission should be previously available in a selected user project. Submitted programs also write output files in the associated project area.
- **SGA Monitor Service**: offers a set of operations for monitoring the SGA's execution nodes.

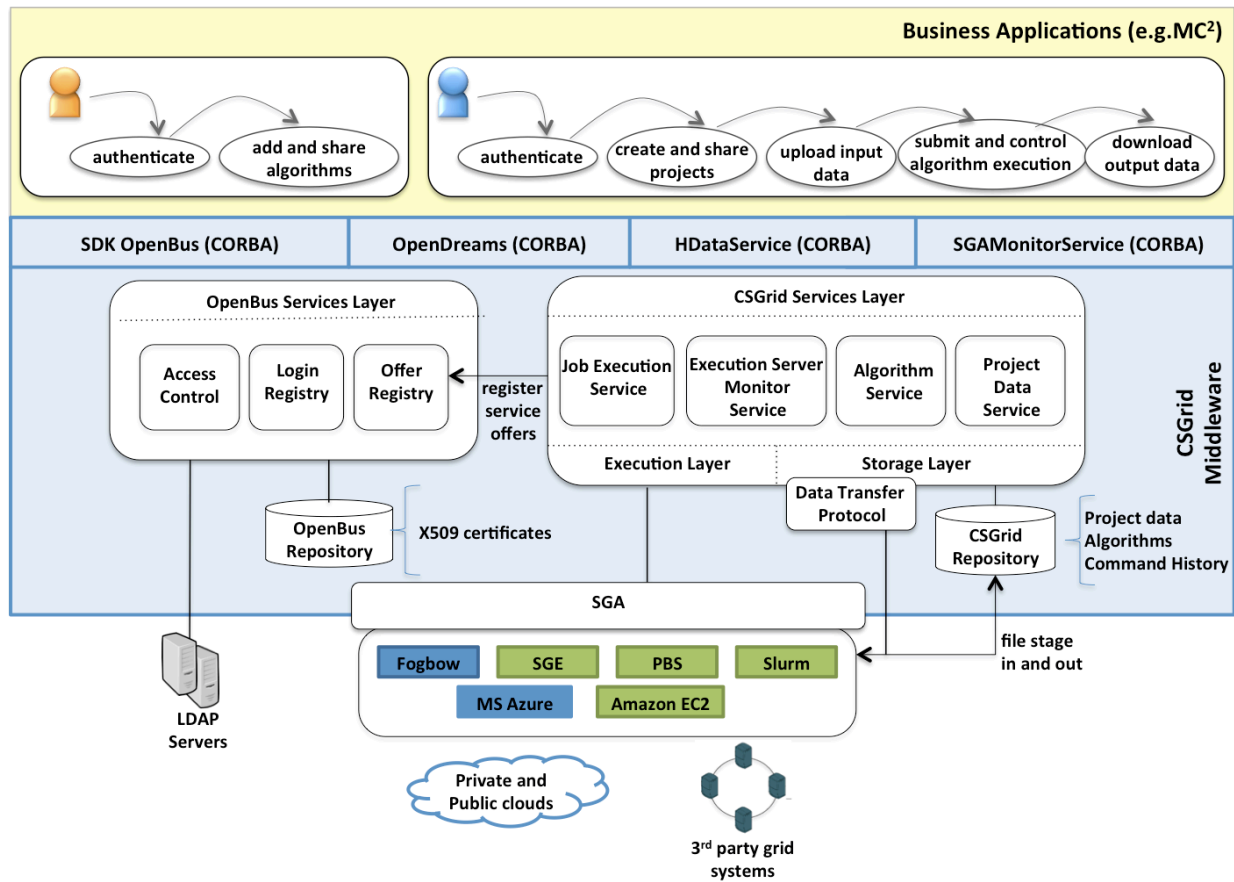


Figure 1 - CSGrid middleware overview

In the EUBrazilCC project, adaptations in this middleware are required to enhance CSGrid flexibility and interoperability, facilitating its integration with different resource management systems for grid and cloud computing, and for improving security. The following requirements are described on Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware:

ADAPTATIONS FOR INTEGRATION WITH OTHER GRID AND CLOUD SYSTEMS

- R1.1 Flexibility on the description of execution resource properties
- R1.2 Flexibility on the description of job execution requirements
- R1.3 Flexibility on information about commands
- R1.4 Support to additional execution parameters
- R1.5 Support to the execution of SGAs on the same process as the CSGrid server
- R1.6 Support to the implementation of listeners during commands lifecycle
- R1.7 Support to job groups

ADAPTATIONS FOR SECURITY REQUIREMENTS

- R2.1 Migration of the OpenBus SDK from 1.5 to 2.x
- R2.2 Privacy on CSBase communications
- R2.3 Authentication of SGAs at the CSGrid server
- R2.4 Support for single sign-on

The present document, D3.3 Prototype of the CSGrid Adaptation Mechanisms, reports the CSGrid 2.0 release notes related with the above requirements. The CSGrid 2.0 installation package and further documentation are available on CSGrid home site (<http://www.tecgraf.puc-rio.br/csgrid>).

2 RELEASE NOTES

2.1 Adaptations on the description of execution resource properties

The SGA sends description and status data to the CSGrid server in order to allow the server to perform scheduling and monitoring tasks. This data is sent both at the SGA initialization, when static properties are provided, and periodically, when dynamic properties are sent to the server. In the CSGrid 2.0 Prototype, these static and dynamic properties are consolidated in an information map (a key-value map), unified in a single set of properties of the SGA:

- Mandatory properties that are handled by the CSGrid Server implementation;
- Any additional information to be used by extension points (plug-ins) added to the CSGrid server.

The methods `registerSGA` and `updateSGAInfo` defined by the SGAManager IDL of CSGrid 2.0 use the new `SGAProperties` structure that represents the SGA properties (mandatory and optional) as a Dictionary (key-value map).

```
struct Pair {
    string key;
    string value;
};
typedef sequence<Pair> Dictionary;
typedef sequence<Dictionary> NodeProperties;

struct SGAProperties {
    Dictionary properties; /** SGA Properties */
    NodeProperties nodesProperties; /* SGA Node Properties */
};

interface SGAManager {
    // other methods

    boolean registerSGA(in SGADaemon sga, in string sgaName,
        in SGAProperties infos, out long updateInterval);

    boolean updateSGAInfo(in SGADaemon sga, in string sgaName,
        in SGAProperties infos);
};
```

The mandatory properties used by CSGrid 2.0 server were defined as constants fields in SGA IDL.

```
const string SGA_NAME = "csbase_sga_name";
const string SGA_ENABLED = "csbase_sga_enabled";
const string SGA_ALGORITHM_ROOT_DIR = "csbase_algorithm_root_directory";
const string SGA_CSFS_HOST = "csbase_csfs_host";
const string SGA_CSFS_PORT = "csbase_csfs_port";
const string SGA_CSFS_ROOT_DIR = "csbase_csfs_root_dir";
const string SGA_FILE_SEPARATOR = "csbase_file_separator";
const string SGA_PROJECT_ROOT_DIR = "csbase_project_root_directory";
const string SGA_SANDBOX_ROOT_DIR = "csbase_sandbox_root_directory";
const string SGA_HAS_DISK_ACCESS = "csbase_has_disk_access";
const string SGA_JOBS_INFO = "csbase_jobs_info";
const string SGA_NODE_CLOCK_SPEED_MHZ = "clock_speed_mhz";
const string SGA_NODE_LOAD_AVG_1MIN_PERC = "csbase_load_avg_1min_perc";
const string SGA_NODE_LOAD_AVG_5MIN_PERC = "csbase_load_avg_5min_perc";
const string SGA_NODE_LOAD_AVG_15MIN_PERC = "csbase_load_avg_15min_perc";
const string SGA_NODE_MEMORY_RAM_FREE_PERC = "csbase_memory_ram_free_perc";
const string SGA_NODE_MEMORY_RAM_INFO_MB = "csbase_memory_ram_info_mb";
const string SGA_NODE_MEMORY_SWAP_FREE_PERC = "csbase_memory_swap_free_perc";
```

Note: This issue is related with the requirement R1.1 (Flexibility on the description of execution resource properties) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

2.2 Adaptations on the description of job execution requirements

The CSGrid server matches jobs requirements and SGAs properties in order to select suitable SGAs for the execution of such jobs. In the previous version of CSGrid, the expression of these execution requirements was currently fixed to a limited set of hard-coded requirements. In CSGrid 2.0 the method `executeCommand` defined by SGADaemon IDL of CSGrid 2.0 was updated adding a dictionary (key-value map) to represent job execution requirements.

```
interface SGADaemon {
    //...

    SGACommand executeCommand(in string command, in string cmdid,
        in Dictionary extraParams, in Dictionary requirements);
};
```

Note: This issue is related with the requirement R1.2 (Flexibility on the description of job execution requirements) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

2.3 Adaptations on information about commands

In addition to the current state of a command executed on CSGrid, the SGA of CSGrid 2.0 also provides additional information about commands (in execution and finalized). This information is represented in a dictionary (key-value) and can be used by the logic implemented in command state observers or any other CSGrid server extension points available.

The SGACommand IDL of CSGrid 2.0 supports dynamic map for representing job execution properties.

```
struct RunningCommandInfo {
    sequence<Dictionary> processInfo;
    Dictionary executionInfo;
};

interface SGACommand {

    //...

    RunningCommandInfo getRunningCommandInfo();
};
```

Mandatory properties already used by CSGrid 2.0 server are defined as constants fields in IDL.

```
const string COMMAND_PID = "csbase_command_pid";
const string COMMAND_PPID = "csbase_command_ppid";
const string COMMAND_STRING = "csbase_command_string";
const string COMMAND_EXEC_HOST = "csbase_command_exec_host";
const string COMMAND_STATE = "csbase_command_state";
const string COMMAND_PROCESSOR_ID = "csbase_command_processor_id";
const string COMMAND_MEMORY_RAM_SIZE_MB = "csbase_command_memory_ram_size_mb";
const string COMMAND_MEMORY_SWAP_SIZE_MB = "csbase_command_memory_swap_size_mb";
const string COMMAND_CPU_PERC = "csbase_command_cpu_perc";
const string COMMAND_CPU_TIME_SEC = "csbase_command_time_sec";
const string COMMAND_WALL_TIME_SEC = "csbase_command_wall_time_sec";
const string COMMAND_USER_TIME_SEC = "csbase_command_user_time_sec";
const string COMMAND_SYSTEM_TIME_SEC = "csbase_command_system_time_sec";
const string COMMAND_VIRTUAL_MEMORY_SIZE_MB =
"csbase_command_virtual_memory_size_mb";
const string COMMAND_BYTES_IN_KB = "csbase_command_bytes_in_kb";
const string COMMAND_BYTES_OUT_KB = "csbase_command_bytes_out_kb";
const string COMMAND_DISK_BYTES_READ_KB = "csbase_command_disk_bytes_read_kb";
const string COMMAND_DISK_BYTES_WRITE_KB = "csbase_command_disk_bytes_write_kb";
```

Note: This issue is related with the requirement R1.3 (Flexibility on information about commands) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

2.4 Support to additional execution parameters

When submitting a command for execution to an SGA, the system shall allow additional parameters to be passed to the selected SGA, so that specific configurations required by 3rd party grid and cloud systems may be added in execution time. The SGADaemon IDL implemented by CSGrid 2.0 supports dynamic map representing job any additional parameter needed by SGA.

```
interface SGADaemon {
    //...

    SGACommand executeCommand(in string command, in string cmdid,
        in Dictionary extraParams, in Dictionary requirements);
};
```

Note: This issue is related with the requirement R1.4 (Support to additional execution parameters) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

2.5 Provides a plugin mechanism that supports the development of server extensions

The CSGrid 2.0 prototype offers a plugin mechanism that provides a runtime engine that dynamically discovers and loads "plug-ins" to the server. The CSGrid 2.0 is able to load two new types of plugins:

- **SGADaemon:** extends the server with new SGA gateways running with the CSGrid server (see Section 2.6)
- **SGADataTransfer:** extends the server with new mechanisms that transfer input and output data from the CSGrid server repository of user's projects and algorithms to the execution host (see Section 2.7)

The CSGrid 2.0 adopts the Java Plugin Framework (<http://jpf.sourceforge.net>).

Note: This issue is related with the requirement R1.5 (Support to the execution of SGAs on the same process as the CSGrid server) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

2.6 Support to the execution of SGAs on the same process as the CSGrid server

The CSGrid 2.0 prototype allows the loading of SGADaemon plugins that implement the same interface as the SGA does and that execute on the same process as the CSGrid server. With this, flexibility is enhanced to implement SGA gateways that use Java and communicate remotely with other systems.

Any new SGA daemon must implement the interface ISGADaemon that extends sgaidl.SGADaemonOperations, a pure Java interface generated from the SGADaemon IDL compilation. The SGA daemon shall be deployed in the plugin repository configured by CSGrid server administrator. The following manifest defines the extension point to connect with new SGADaemon extensions:

```
<?xml version="1.0" ?>
<!DOCTYPE plugin PUBLIC "-//JPF//Java Plug-in Manifest 1.0"
"http://jpf.sourceforge.net/plugin_1_0.dtd">
<plugin id="csbase.sga" version="1.0.0">
  <extension-point id="connector">
    <parameter-def id="class"/>
    <parameter-def id="name"/>
  </extension-point>
</plugin>
```

Note: This issue is related with the requirement R1.5 (Support to the execution of SGAs on the same process as the CSGrid server) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

2.7 Extends data transfer mechanism between CSGrid server and SGA

In CSGrid 2.0, each SGA daemon can define its own mechanism of data transfer, including input and output data and algorithms binaries. Besides the mechanism CSGrid already supports, others can be developed and dynamically loaded, using plugin extensions. The

property “csbase_transfer_name”, provided by the SGADaemon, defines the name of the mechanism the CSGrid server uses.

Any new SGA data transfer mechanism must implement the interface ISGADataTransfer sgaidl.SGADaemonOperations and deployed in the plugin repository configured by CSGrid server administrator. The following manifest defines the extension point used to connect with new data transfer mechanisms extensions:

```
<?xml version="1.0" ?>
<!DOCTYPE plugin PUBLIC "-//JPF//Java Plug-in Manifest 1.0"
"http://jpf.sourceforge.net/plugin_1_0.dtd">
<plugin id="csbase.sga.datatransfer" version="1.0.0">
  <extension-point id="mechanism">
    <parameter-def id="class" />
    <parameter-def id="name" />
  </extension-point>
</plugin>
```

Note: This issue is related with the requirement R1.5 (Support to the execution of SGAs on the same process as the CSGrid server) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

2.8 Migration of the OpenBus SDK from 1.5 to 2.x

The CSGrid 2.0 uses the OpenBus SDK 2.0. OpenBus 2.0 enhances its security mechanisms to ensure that communication identification cannot be imitated or emulated by other parties. Such support is essential to guarantee the security of sensible services exported using OpenBus that accept or deny requests according to the identity of the caller. For this reason, it is essential to migrate CSGrid to OpenBus 2.0, which provides better security support in public or insecure networking environments.

The runtime of CSGrid 2.0 depends of the following libraries available on Tecgraf Maven Repository (<http://maven.tecgraf.puc-rio.br:8081/nexus/>):

```
<dependency>
  <groupId>br.puc-rio.tecgraf.openbus</groupId>
  <artifactId>openbus-sdk-core</artifactId>
  <version>2.0.2.0</version>
</dependency>
<dependency>
  <groupId>br.puc-rio.tecgraf.openbus</groupId>
  <artifactId>openbus-sdk-legacy</artifactId>
  <version>2.0.2.0</version>
</dependency>
<dependency>
  <groupId>br.puc-rio.tecgraf.openbus</groupId>
  <artifactId>openbus-sdk-assistant-core</artifactId>
  <version>2.0.1.0</version>
</dependency>
```

Note: This issue is related with the requirement R2.1 (Migration of the OpenBus SDK from 1.5 to 2.x) described in the Deliverable D3.1 – Adaptation Requirements for CSGrid Middleware.

ACRONYMS AND ABBREVIATIONS

Acronym	Definition
CNPq	National Council for Scientific and Technological Development of Brazil
CORBA	Common Object Request Broker Architecture
DRMAA	'Distributed Resource Management Application API
EC	European Commission
Amazon EC2	Amazon Elastic Compute Cloud
EUBrazilCC	EUBrazilCloudConnect - EU-Brazil Cloud infrastructure Connecting federated resources for Scientific Advancement
Fogbow	Fogbow project - Federated, opportunistic and green cloud
FP7	Framework Programme Seven
mc2	Minha Cloud Científica (My Scientific Cloud)
MCT	Brazilian Ministry of Science and Technology
OGF	Open Grid Forum
RMI	Remote Method Invocation
SGA	Servidor de Gerência de Algoritmos (Algorithm Management Server)
SGE	Sun Grid Engine
SLURM	Simple Linux Utility for Resource Management
SSO	Single Sign-on
STREP	Small or medium-scale focused research project
TORQUE/PBS	TORQUE Resource Manager
WP	Work Package