EGI-InSPIRE

TECHNOLOGY STUDY FOR CTA VIRTUAL TEAM PROJECT

Science Gateway and Single Sign On User requirements

Date:	**18/02/2013**
Document Status:	**Draft**
Dissemination Level:	**Public**
Document Link:	**https://documents.egi.eu/document/xxxx**

Abstract

This document has been written to summarize User Requirements for Science Gateway and associated Single Sign On functionality.

I. DOCUMENT LOG

Version	Date	Comment	Author/Partner
0.1	4/02/2013	Creation and collect of existing requirements	N. Neyroud, C. Barbier
0.2	18/02/2013	Translation to Science Gateway requirements	N. Neyroud, C. Barbier

II. APPLICATION AREA

This document is an internal report produced by the members of the Technology study for CTA EGI Virtual Team project, run under the EGI-InSPIRE NA2 virtual team framework. Further information is available at https://wiki.egi.eu/wiki/Virtual_team.

III. TERMINOLOGY

A complete project glossary is provided at the following page: http://www.egi.eu/about/glossary/.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Scope of the document

This document is related to two software products, Science Gateway and associated Single Sign On in the context of CTA experiment.

- Web-based Science Gateways are community-specific set of tools, applications, and data collections that are integrated together via a web portal, providing access to resources and services from a distributed computing infrastructure. These gateways can support a variety of capabilities including workflows handling, virtualization of software and hardware, visualization as well as resource discovery, job execution services, access to data collections, applications and tools for data analysis. A Science Gateway enables community members to define and perform custom research scenarios or other types of use cases.
- Single Sign On solutions simplify access control to multiple related, but independent software systems. Through SSO a user logs in once and gains access to all systems he/she needs to carry out activities without being prompted to log in again at each of them. SSO solutions can be relevant for the human and automated workflows of the Observatory that need to use software, data and resources from multiple systems.

1.2 Purpose of the document

This public version of User Requirements document is strongly linked to the CTA requirements already validated by the CTA consortium; it's completed with more precise requirements collected using a social network between CTA and EGI through the support teams of the National Grid Initiatives. The objective of this document is to be able to study in a second step the most suitable solutions from EGI and its partners that are capable to address the CTA requirements.

1.3 Definitions, acronyms and abbreviations

CTA	Cherenkov Telescope Array
ESFRI	European Strategy Forum for Research Infrastructures
DCI	Distributed Computing Infrastructure
CTAO	CTA Observatory
MC	Monte Carlo simulation
SSO	Single Sign On
VObs	Virtual Observatory
PI	Principal Investigator

1.4 *Reference documents*

R1: “ESA software engineering standard issue 2”
http://www.esa.int/TEC/Software_engineering_and_standardisation/

1.5 *Users*

The following users have been identified:

The Observatory "The scientific facility with observation site(s), and additional supporting facilities at additional sites, which delivers the required performance for CTA"

User "A scientific user of the CTA Observatory (CTAO) who may be a Guest Observer, a Privileged User or an Archive User"

Guest Observer "A member of the scientific community who is granted access to a specific subset of CTA data, associated with a successful proposal to the Observatory"

Privileged User "A scientific user of the CTA observatory with access to data and/or software and/or services not available to all Users"

Archive User "A scientific user of the CTA observatory who makes use of archival data, as opposed to data associated with a specific proposal"

Principal Investigator "A member of the scientific community named in a proposal to be the contact for all interactions with CTA Observatory”

1.6 *Overview*

The following sections have been written using a two steps methodology: first phase using the existing CTA requirements then, in a second phase, using the social network to gather additional user requirements concerning two software products:

a) Web based Science Gateways operated for the CTA community, making Distributed resources and services from the NGIs accessible for CTA members;

b) A Single Sign On (SSO) authentication, internationally federated, mechanism that would make web-based Science Gateways accessible for the CTA community.

This document is organized following the ESA-PSS-05 Software Life Cycle standard and follows the user requirements recommended table of contents.

2 GENERAL DESCRIPTION

By definition, a Science Gateway is a "community-development set of tools, applications, and data that is integrated via a portal or a suite of applications, usually in a graphical user interface, that is further customized to meet the needs of a specific community". Specific to CTA, a Science Gateway provides access to resources and services from a distributed computing infrastructure.

For the CTA community, the personalization of human-computer interface and management of complex access rights is one of the main requirements of this product and explains why this project is composed of two sub-products: Science Gateway and Single Sign On solutions.

Single sign-on (SSO) is a property of access control of multiple related, but independent software systems. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them. Conversely, Single sign-off is the property whereby a single action of signing out terminates access to multiple software systems. As different applications and resources support different authentication mechanisms, single sign-on has to internally translate to and store different credentials compared to what is used for initial authentication.

2.1 *Product perspective*

The Science Gateway software product will require to be strongly linked to the Single Sign On software for the authentication process and management of access rights. Its main functionality will be to integrate external CTA web applications as science tools, data pipeline or monitoring tools, providing data and software access to a large set of users from scientists up to expert-users or operators.

2.2 *General capabilities*

This section should describe the main capabilities and why they are needed. This section should describe the process to be supported by the software, indicating those parts of the process where it is used.

2.3 *General constraints*

This section should describe any items that will limit the developer's options for building the software.

This section should not be used to impose specific requirements or specific design constraints, but should state the reasons why certain requirements or constraints exist.

2.4 *User characteristics*

This section should describe those general characteristics of the users affecting the specific requirements.

Many people may interact with the software during the operations and maintenance phase. Some of these people are users, operators and maintenance personnel. Certain characteristics of these people, such as educational level, language, experience and technical expertise impose important constraints on the software.

Software may be frequently used, but individuals may use it only occasionally. Frequent users will become experts whereas infrequent users may remain relative novices. It is important to classify the users and estimate the likely numbers in each category. If absolute numbers cannot be stated, relative numbers can still be useful.

2.5 *Operational environment*

A clear description of the real world that the software will operate in should be built up, as the user requirements are captured.

This description of the operational environment must clearly establish the problem context.

In a system development, each subsystem will have interfaces to other, external, and systems. The nature of these exchanges with external systems should be specified and controlled from the start of the project. The information may reside in an Interface Control Document (ICD), or in the design documentation of the external system.

The roles and responsibilities of the users and operators of software should be established by defining the:

- *Characteristics of each group (e.g. experience, qualifications);*
- *Operations they perform (e.g. the user of the data may not operate the software).*

3 SPECIFIC REQUIREMENTS: SCIENCE GATEWAY

The specification of user requirements is the process of organizing information about user needs and expressing them in a document.

A requirement is a "condition or capability needed by a user to solve a problem or achieve an objective". This definition leads to two principal categories of requirements: "capability requirements" and "constraint requirements".

3.1 Science Gateway requirements

Capability requirements describe the process to be supported by software. Simply stated, they describe 'what' the users want to do.

A capability requirement should define an operation, or sequence of related operations, that the software will be able to perform. If the sequence contains more than approximately five related operations, the capability requirement should be split.

The operations should be organized to describe the overall process from start to finish. Where there are many operations to describe, it is recommended that they are grouped hierarchically to help manage the complexity.

Operations may be routine, (e.g. normal tasks) or non-routine (e.g. error handling, interruptions). Non-routine operations may be grouped separately from those related to the normal processing.

In the Software Requirements Definition Phase, capability requirements will be analyzed to produce a set of functional requirements. If duplication of capability requirements occurs, the analyst may be able to replace them with a single functional requirement. A single function may support a process at many different times, therefore a function can map to many capability requirements.

Quantitative statements that specify performance and accuracy attributes should form part of the specification of capability. This means that a capability requirement should be qualified with values of: capacity; speed; accuracy.

The performance attribute is the combination of the capacity and speed attributes.

UR-SG-0010 The Science Gateway must be able to propose to each user its authorized applications associated with its authorized data.

UR-SG-0020 The Science Gateway must provide access to the data products resulting from the Guest observers' proposals, the MC simulation results and the Instrument Response Function.

UR-SG-0030 Data Management applications that need to be accessible via the Science Gateway are: High-level Data analysis (Science tools), Full Data analysis chain, MC simulation tools.

UR-SG-0040 The Data Management Monitoring applications that need to be accessible via the Science Gateway are: Status information on the standard pipeline processing of a specific proposal subset of data, CTA performance tools.

UR-SG-0050 Each application must provide associated documentation.

UR-SG-0060 Each application must provide an associated user support.

UR-SG-0070 The Science Gateway must provide an application for Observatory operation user and community feedback.

3.1.1 Performance

The speed attribute states how fast the complete operation, or sequence of operations, is to be performed. Each capability requirement should be attached with a quantitative measure of the speed required. There are various ways to do this, for example the:

- *number of operations done per unit time interval;*
- *time taken to perform an operation.*

For example: '95% of the transactions shall be processed in less than 1 second', is acceptable whilst, '95% of the transactions will be done as soon as possible' is not.

Note that a system may react quickly to a command but take quite a long time to complete the operations requested. Such 'response' requirements should be stated as HCI requirements.

UR-SG-0100

3.2 Constraint requirements

Constraint requirements place restrictions on how the user requirements are to be met. The user may place constraints on the software related to interfaces, quality, resources and timescales. Users may constrain how communication is done with other systems, what hardware is to be used, what software it has to be compatible with, and how it must interact with human operators. These are all interface constraints. An interface is a shared boundary between two systems; it may be defined in terms of what is exchanged across the boundary. Interfaces are important kinds of constraints. The user may define external interfaces (i.e. state how interactions with other systems must be done) but should leave the developers to define the internal interfaces (i.e. to state how software components will interact with each other). Users may constrain the quality required of the final product. Typical quality characteristics are: adaptability, availability, portability, security and safety.

3.2.1 Communications interfaces

A communications interface requirement may specify the networks and network protocols to be used. Performance attributes of the interface may be specified (e.g. data rate).

The ISO reference model for Open Systems Interconnection, with its seven layers of abstraction, can be used for describing communications interfaces. This means that a communications interface requirement should use terminology consistent with the model. Communications interface requirements should avoid mixing the layers of abstraction.

UR-SG-0200

3.2.2 Hardware interfaces

A hardware interface requirement specifies all or part of the computer hardware the software is to execute on. This may be done by stating the make and model of the device, physical limitations (e.g. size, weight), performance (e.g. speed, memory), qualifications (e.g. project approved, space qualified) and also perhaps whether any hardware selected has to be derated (e.g. for operation at altitude). Environmental considerations that affect the selection of hardware may be stated (e.g humidity, temperature and pressure).

UR-SG-0300

3.2.3 Software interfaces

A software interface requirement specifies whether the software is to be compatible with other software (e.g other applications, compilers, operating systems, programming languages and database management systems).

UR-SG-0400

3.2.4 Human-Computer Interface

A Human-Computer Interface (HCI) requirement may specify any aspect of the user interface. This may include a statement about style (e.g. command language, menu system, icons), format (e.g. report content and layout), messages (e.g. brief, exhaustive) and responsiveness (e.g. time taken to respond to command). The hardware at the user interface (e.g. colour display and mouse) may be included either as an HCI requirement or as a hardware interface requirement.

UR-SG-0500 The Science Gateway provides a single web interface to users.

3.2.5 Adaptability

Adaptability measures how easily a system copes with requirements changes. Adaptable (or flexible) systems are likely to live longer, although the extra design work needed may be extensive, especially for optimizing modularity. An example of an adaptability requirement is: 'it shall be possible to add new commands without retesting existing commands'.

In the operations and maintenance phase the software may undergo continuous adaptation as the user requirements are modified by experience.

When considering the adaptability, note that any change involves some risk, and to change reliable parts of the system may not be acceptable.

UR-SG-0600 The Science Gateway and related applications must be flexible and adaptable enough to integrate new features and new data access rules during the period of operations and 10 years after the CTA decommissioning.

3.2.6 Availability

Availability measures the ability of a system to be used during its intended periods of its operation. Availability requirements may specify:

- *mean and minimum capacity available (e.g. all terminals);*
- *start and end times of availability (e.g. from 0900 to 1730 daily);*
- *time period for averaging availability (e.g. 1 year).*

Examples of availability requirements are:

· 'the user shall be provided with 98% average availability over 1 year during working hours and never less than 50% of working hours in any one week'; 'all essential capabilities shall be at least 98% available in any 48 hour period and at least 75% available in every 3 hour period'.

When a system is unavailable, some, or even all, of its capabilities cannot be used. A loss of capability is called a 'failure' and is caused by one or more 'faults'. The average time between the occurrences of faults internal to the software (i.e. 'bugs') measures the 'reliability' of the software. The average time taken to fix such faults measures its maintainability. A system may also become unavailable due to external factors (e.g. loss of input service).

Users only need to state their availability requirements. The availability requirements are decomposed into specific reliability and maintainability requirements in the SR phase.

UR-SG-0700 The availability of the Science Gateway and related applications must be > 98%.

3.2.7 Portability

Software portability is measured by the ease that it can be moved from one environment to another. Portable software tends to be long lived, but more code may have to be written and performance requirements may be more difficult to meet. An example of a portability requirement is: 'the software shall be portable between environments X and Y'. Portability can be measured in terms of the number of lines of code and/or the number of modules that do not have to be changed to port the software from one computer to another. Either absolute or relative measurements can be used. If migration to another hardware base or operating system is intended, then any requirements to run with different hardware and software interfaces should be stated as portability requirements. New interfaces should be described (e.g. name the new operating system or computer hardware).

UR-SG-0800 The Scientific Gateway and related applications must be portable enough to be maintained over the period of operations and 10 years after CTA decommissioning.

3.2.8 Security

A system may need to be secured against threats to its confidentiality, integrity and availability. For example, a user may request that unauthorized users be unable to use the system, or that no single event such as a fire should cause the loss of more than 1 week's information. The user should describe threats that the system needs to be protected against, e.g. virus intrusions, hackers, fires, computer breakdowns.

The security of a system can be described in terms of the ownership of, and rights of access to, the capabilities of the system.

A secure system protects users from their own errors as well as the malicious interference, or illegal activities, of unauthorized users.

UR-SG-0900 The Science Gateway must be able to propose to each user its authorized applications associated with its authorized archive data.

UR-SG-0910 A specific user can be simultaneously Privileged user, Principal Investigator or Guest observer for one or more specific subset of data, Archive user for all the public Archive data.

UR-SG-0920 All data obtained by CTA observatory must be made public (for Archive users?) through an archive following a period of proprietary use. The proprietary period may depend on the type of observations.

UR-SG-0930 Access rights for users and data rights are defined in a CTA data access policy document. The shareholders may define or change access limitations to the archive.

3.2.9 Standards

Standards requirements normally reference the applicable documents that define the standard.

Two kinds of standards can be specified: process standards and product standards. Examples of product standards are export file formats and legal report formats. Examples of the process standards are product assurance standards and accounting procedures to be followed. Adherence to process standards should be specified in the Software Project Management Plan.

A standards requirement may specify the methods that are to be employed by the developers in subsequent phases. Such methods must be compatible with the life cycle defined in ESA PSS-05-0.

3.2.10 Resources

The resources available for producing and operating the software are a constraint on the design. If this information is available then it should be stated in the Software Project Management Plan in terms of one or more of financial, manpower and material limits. As with any other product, the quality and sophistication of a software product are limited by the resources that are put into building it.

Resource requirements may include specifications of the computer resources available (e.g. main memory). They may define the minimum hardware that the system must run on (e.g. a 486 PC with 4 Mbytes of memory). Care should be taken to include only the necessary resource constraints.

3.2.11 Timescales

A constraint on the design of the software may be the acceptable timescales for its development and production. Requirements for the achievement of specific life cycle milestones may be stated in the Software Project Management Plan.

4 SPECIFIC REQUIREMENTS: SINGLE SIGN ON

The specification of user requirements is the process of organizing information about user needs and expressing them in a document.

A requirement is a “condition or capability needed by a user to solve a problem or achieve an objective”. This definition leads to two principal categories of requirements: “capability requirements” and “constraint requirements”.

4.1 Single Sign On requirements

UR-SSO-0010

4.1.1 Performance

UR-SSO-0100

4.2 Constraint requirements

4.2.1 Communications interfaces

UR-SSO-0200

4.2.2 Hardware interfaces

UR-SSO-0300

4.2.3 Software interfaces

UR-SSO-0400

4.2.4 Human-Computer Interface

UR-SSO-0500

4.2.5 Adaptability

UR-SSO-0600

4.2.6 Availability

UR-SSO-0700

4.2.7 Portability

UR-SSO-0800

4.2.8 Security

UR-SSO-0900

4.2.9 Standards

4.2.10 Resources

4.2.11 Timescales

5 APPENDIX