Heliophysics Integrated Observatory

Project No.: 238969
Call: FP7-INFRA-2008-2

Community Interaction Service Developers Guide

Draft

Title:	**Community Interaction Service – Developers Guide**
Document No.:	HELIO_TCD_S3_002_TN
Date:	17 June 2012
Editor:	**Dr Gabriele Pierantoni**, TCD
Contributors:	
Distribution:	Project

Revision History

Version	Date	Released by	Detail
0.1	17 June 2012	Dr. Gabriele Pierantoni	1st draft
0.1	01 October 2012	Dr. Gabriele Pierantoni	Minor Corrections

Note: Any notes here.

Table of Figures

Introduction

The Community Interaction Service (CIS) offers services for authentication, authorization and the management of user's preferences.

CIS supports two types of Authentication and Authorization: Spring-based security and Grid certificate-based security.

According to its dual nature, the CIS can be deployed in two modalities as sketched in Figure 1.

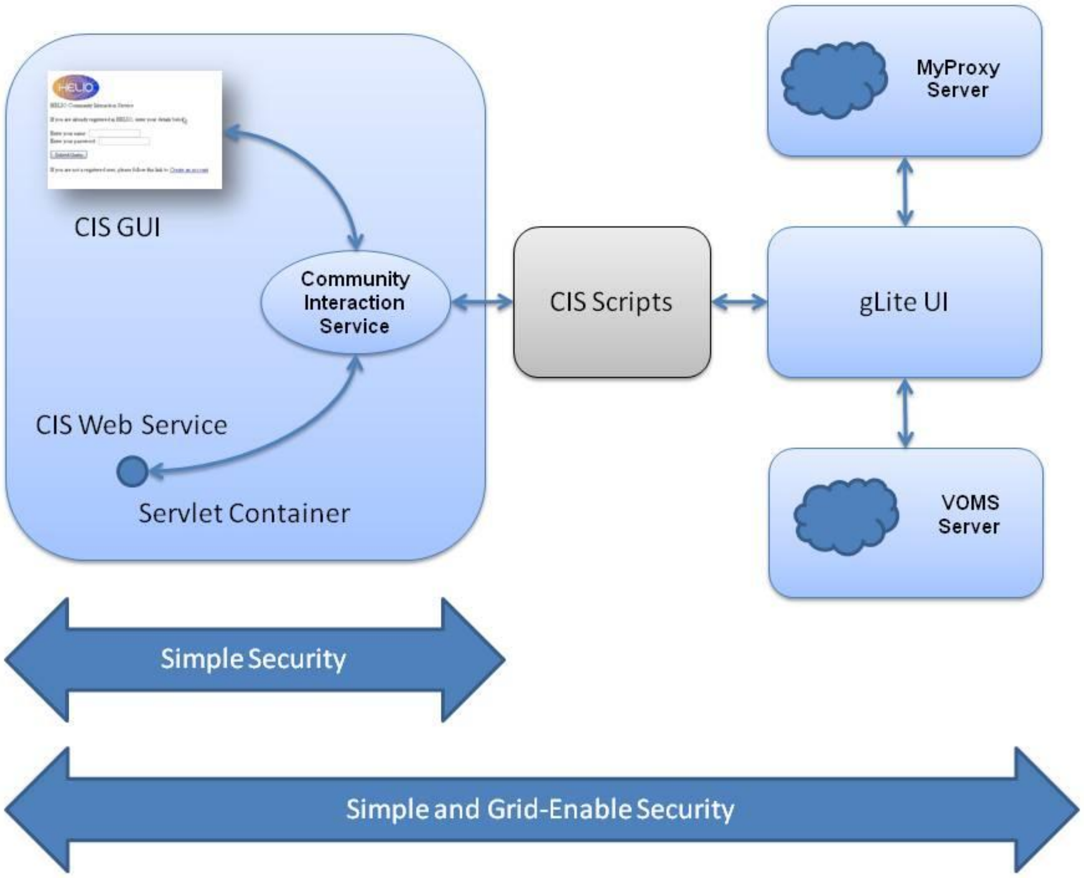

Figure 1, different deployment profiles of CIS

When the CIS is used in spring-security mode only, its stack is described in Figure 2.

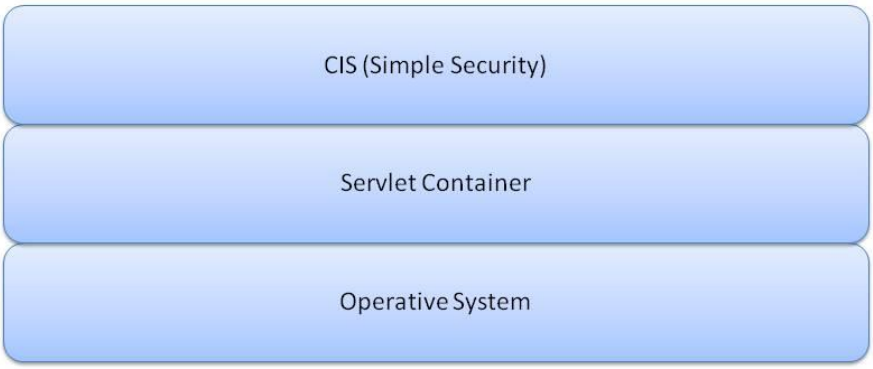

Figure 2, CIS stack in simple modality

In this case, the CIS simply consists of a web service application that comprehends an identity repository, the main application, its web service interface and a GUI.

When a higher degree of security is required, the CIS can be used in conjunction with a MyProxy and a VOMS service to cater for certificate-based security as sketched in Figure 3. It is important to notice that these two modalities of use are not mutually exclusive and, provided that an instance of a MyProxy service is available to the CIS, they can be both used concurrently.

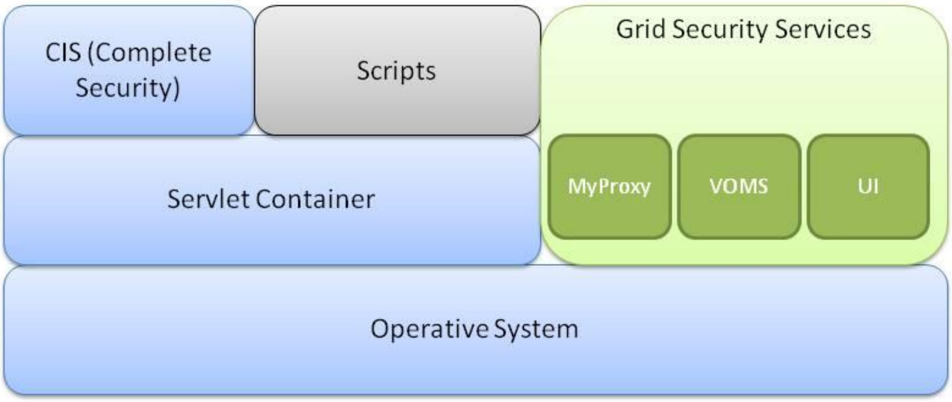

Figure 3, CIS stack in full modality

CIS also caters for user's preferences. All HELIO users can define a set of preferences with which they can define how to access the HELIO services and display their results.
The preferences structure is based on the structure of HELIO services.

- **Service**: the HELIO service that has to apply the user-defined preferences. An administrator can add or remove any number of services.
    - Field: the field of the HELIO service that has to apply the user-defined preferences. Administrators can be remove or add any number of fields for each service
    - Value: each field have a value associated with it. All users can modify this value; administrators can set the default value that is applied to all new users.

Suggested Reading

To build the service	
Java (compile service)	http://www.oracle.com/technetwork/java/javase/overview/index.html
Tomcat (web container to host the service)	http://tomcat.apache.org/
Maven (build system)	http://maven.apache.org/ (or use a plug-in to your development environment)
To install the service	
MyProxy (Main web page)	http://grid.ncsa.illinois.edu/myproxy/
MyProxy (Installation Manual)	http://grid.ncsa.illinois.edu/myproxy/install.html
MyProxy (Documentation)	http://grid.ncsa.illinois.edu/myproxy/doc.html
Grid Certificates (Grid Ireland Introduction)	http://cagraidsvr06.cs.tcd.ie/ca/getting-a-cert.html#introduction
Grid Certificates (Introduction to Grid Computing)	www.redbooks.ibm.com/redbooks/pdfs/sg246895.pdf
VOMS (Main page)	http://vdt.cs.wisc.edu/components/voms.html
VOMS (Documentation Page)	http://vdt.cs.wisc.edu/documentation.html
gLite UI (Main page)	http://glite.web.cern.ch/glite/packages/R3.2/sl5_x86_64/deployment/glite-UI/3.2.1-0/glite-UI-3.2.1-0-update.html
gLite UI (Documentation page)	https://edms.cern.ch/document/722398/

System Requirements

- Java 1.6
- Servlet Container i.e. Tomcat greater than 5.5, GlassFish, JBOSS. For the purpose of this document, installation is in Tomcat.
- MyProxy (Only if certificate-based security is required)
- VOMS (Only if certificate-based security is required)
- gLite 3.2 UI (Only if certificate-based security is required).

Service Installation

CIS primary installation is in TCD, these instructions will create a mirror service. The service can be installed as simple (spring only) security or full (spring and certificate-based) security.

Downloading the Source

- Community Interaction Service is installed as a war file on a servlet container:

Building the Service

No building is necessary.

Installing the Service

It is assumed installation of a servlet container Tomcat.

Simple Installation

The simple installation requires that the war file is deployed in the servlet container.
The CIS configuration is contained in the CIS.conf file that must be deployed in the tomcat directory. When the CIS is deployed in simple modality, the CIS.conf file contains only one entry that is the location of the repository file that holds the information is in a directory that could be read and written by the tomcat user.

Full Installation

In addition to the steps described in the Simple Installation section, requires that:

- The directory of all the CIS scripts defined in the CIS.conf file be readeable and executable by the TOMCAT user but cannot be written.
- An instance of a gLite UI 3.2 is installed on the machine as per https://twiki.cern.ch/twiki/bin/view/LCG/GenericInstallGuide320 and https://twiki.cern.ch/twiki/bin/view/LCG/GenericInstallGuide320#The_UI
- An instance of a MyProxy server is installed (as per http://grid.ncsa.illinois.edu/myproxy/install.html) and can be reached by the machine hosting the CIS.
- An instance of a VOMS is installed (as per https://twiki.cern.ch/twiki/bin/view/EGEE/VomsInstallGuide_32) and can be reached by the machine hosting the CIS.

Structure of the Code

The CIS code is divided into different packages:

- ***helio-shared*** - Utilities classes shared among different components of CIS and other services are in the shared project,
- ***helio-cis-server-lib*** - This package contains all the core functionalities of the CIS server,
- ***helio-cis-server-ws*** – This package contains the web-service interface of the CIS
- ***helio-cis-server-gui*** – This package contains the graphical user interface of the CIS
- ***helio-cis-client*** – This package contains the client to the web-service interface of the CIS

Dependencies

The CIS dependencies are managed by the maven project of HELIO. They are described package by package by their pom.xml files.

Logging

CIS logs to the ***catalina.out*** log file.

Maintaining & Extending the Service

The CIS exposes a GUI that can be used to manage the user database

Fixing problems

Contact mailto://helio-services@majordomo.mssl.ucl.ac.uk

Adding new Data

The CIS exposes a GUI that can be used to manage the user database

Download, Build, Install

If the requirement is to view the source code to make an extension or correct a possible error, follow instructions below for bulding from the source code repository.

Downloading Source

The CIS source code is part of the HELIO-vo project in sourceforge. The main page is: http://sourceforge.net/projects/helio-vo/.
You can download the complete source code using svn issueing the command
svn co https://helio-vo.svn.sourceforge.net/svnroot/helio-vo helio-vo

Building CIS

CIS is built using Maven. It is recommended to issue a ***maven clean*** command before building with a ***maven package*** command. This will build a war file helio-cis-server/target directory.