



***Heliophysics
Integrated
Observatory***

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

Community Interaction Service (CIS)
User Manual
Draft

<i>Title:</i>	Community Interaction Service – User Manual
<i>Document No.:</i>	HELIO_TCD_S3_002_UM
<i>Date:</i>	01 October 2012
<i>Editor:</i>	Dr. Gabriele Pierantoni , Trinity College Dublin
<i>Contributors:</i>	
<i>Distribution:</i>	Project



Revision History

Version	Date	Released by	Detail
0.1	10/06/2012	Gabriele Pierantoni	First Draft
0.1	23/07/2012	Anja Le Blanc	Added workflow description
0.2	09/08/2012	Gabriele Pierantoni	Second Draft (added SOAP interface)
0.3	01/10/2012	Gabriele Pierantoni	Minor Corrections

Note: Any notes here.

Table of Figures.....	1
Introduction	2
About the CIS	2
Simple Security Profile	3
Certificate-based security Profile	5
User Preferences.....	7
How to Access the CIS.....	8
The Graphical User Interface	8
Simple User Operations.....	8
Grid User Operations.....	8
Administrator Operations	8
Login or create an account	8
Choice of role	9
Choice of simple-role actions.....	9
Change password.....	9
See the user's preferences	10
Modify the user's preferences	10
Remove your account.....	11
Add MyProxy details.....	11
Choice of admin-role actions	11
Remove an account	11
Modify or add standard preferences.....	12
Remove standard preferences.....	12
To remove a field of an existing service	12
Add a user to the administrator list	13
Other Interfaces	14
Sample Workflows	16
How to Use the CIS.....	17
Appendix A	18
Setup the certwizard	18
Load the grid certificate into the certwizard	18
Setup the certificate authority.....	20
Test that the time on the user machine is updated.....	22
Setup the MyProxy server	23
Test the validity of the certificate.....	24
Upload the certificate to MyProxy	25
Appendix B.....	27
Works Cited.....	28

Table of Figures

Figure 1, Simple Security use of the CIS	3
Figure 2, Grid Security use of the CIS	5
Figure 3, Login Screen for the CIS	8
Figure 4, Simple User Actions	9
Figure 5, Change a password	9
Figure 6, View the user's preferences.....	10
Figure 7, Modify the user's preferences	10
Figure 8, Insert MyProxy Information	11
Figure 9, Select the administrator actions	11
Figure 10, Modify the default preferences	12
Figure 11, Promote a user to Administrator	13
Figure 12: Example workflow using CIS for authentication	16
Figure 13, Load the grid certificate (as pem files) into the certwizard	18
Figure 14, Load the grid certificate (as p12 files) into the certwizard	19
Figure 15, Select the certificates for the Grid Ireland Certificate Authority.....	20
Figure 16, Select the pkcs7 format for the Grid Ireland Certificate Authority	21
Figure 17, Select the downloaded certificates	21
Figure 18, Setup the Grid Ireland Certificate Authority	22
Figure 19, Testing that the time is correct.....	22
Figure 20, Enter the MyProxy details for Grid Ireland.	23
Figure 21, test if the settings are correct.	24
Figure 22, Select the MyProxy server.	25
Figure 23, enter the login details.	25

Introduction

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

About the CIS

Authentication and Authorization, or more broadly, the security services deal in HELIO with two main types of services: the first, with simple requirements, perform only locally authentication and authorization while the second type of services, with more stringent requirements, not only need local authentication and authorization but also use services that need high security levels like grids or distributed storage.

To cater for both services, the CIS issues a spring-compliant security token (for the services that perform only local authentication and authorization) that can also include information for higher level security services.

The following sections describe the usage of the CIS for each of these functionalities.

Simple Security Profile

The CIS offers the means to manage accounts for HELIO and issues an authentication token that is compliant to the spring security framework (Spring Security) for local authorization.

The interaction with the first type of service is sketched in Figure 1

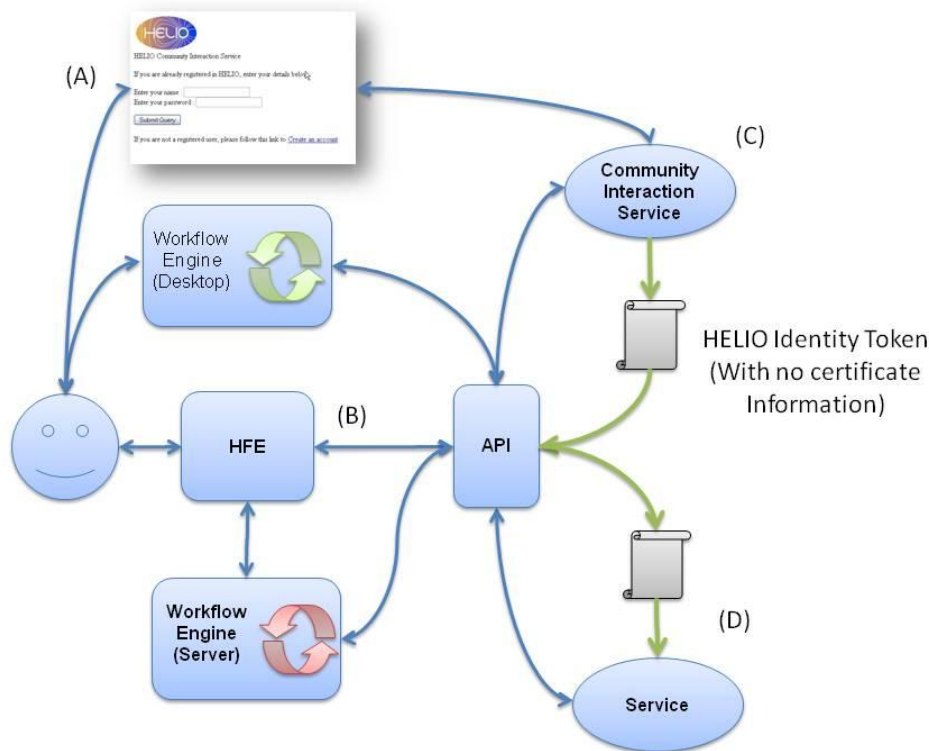


Figure 1, Simple Security use of the CIS

To use the CIS for authentication and authorization with a simple security profile, the steps described in Figure 1 should be performed:

1. Creation of an account with HELIO
 - 1.1. The user creates her/his own account in the CIS (This step can also be performed programmatically through the API)
2. Authentication and Authorization
 - 2.1. The program running on behalf of the user, when it needs an Authentication Token (HELIO Identity Token, or HIT) to be issued on her/his behalf, invokes the authentication method from the CIS.
 - 2.2. The CIS, checks that the user is a HELIO registered user and issues the Authentication Token.
 - 2.3. The Authentication Token is then sent to the services that can use it to perform spring-compliant authentication and authorization.

Step 1 has to be performed only once, steps 2.1 – 2.3 are executed whenever a user uses the HELIO system (through its main front end, or through other means such as the TAVERNA workbench)

Certificate-based security Profile

HELIO also interacts with services that require authentication and authorization based on grid certificates; to support this scenario the CIS connects to the HELIO components as sketched in Figure 2. In addition to the components and steps of Figure 1, the user has a Grid Certificate and uploads it to the CIS whenever she/he wants to use it for her/his authentication.

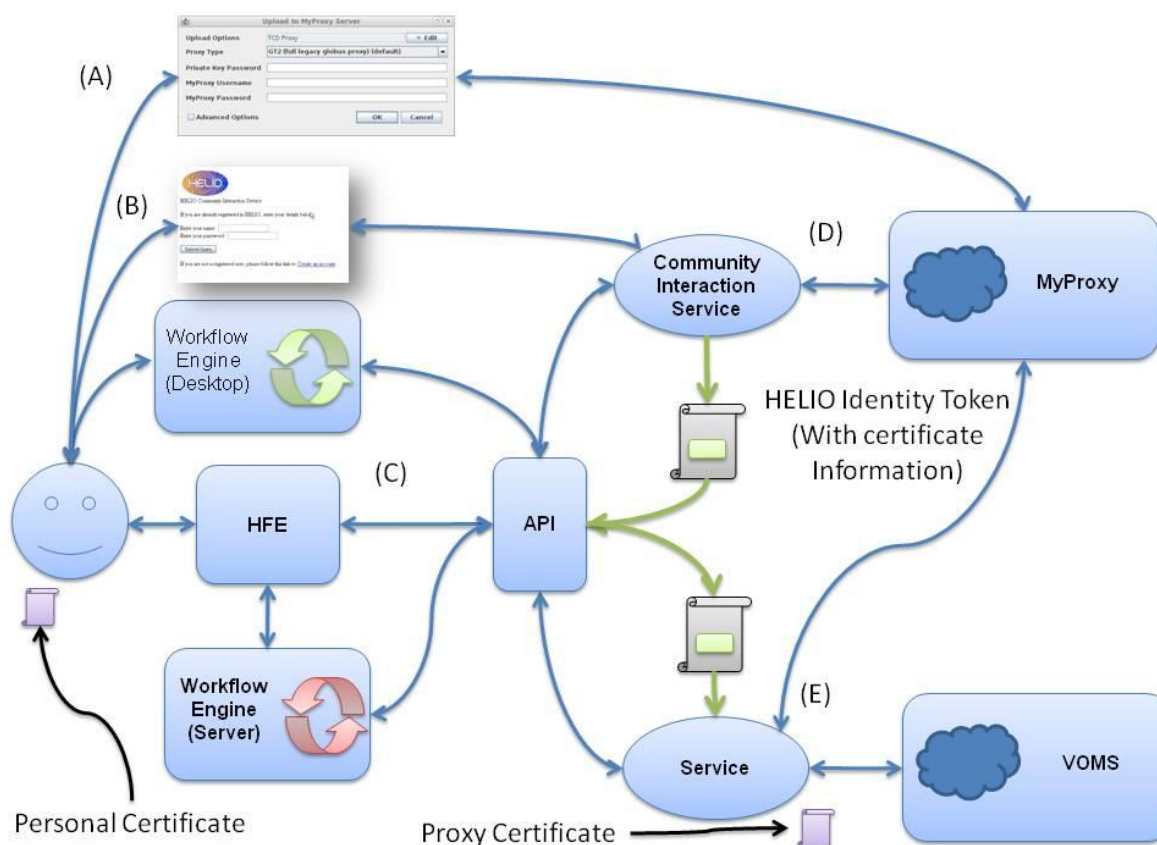


Figure 2, Grid Security use of the CIS

To use the CIS for certificate-based authentication and authorization, the steps described in Figure 1 should be performed:

1. The user requests a certificate if she/he does not have one already.
 - 1.1. Depending on the Nationality, the user who wants to obtain a Grid Certificate should contact her/his national Certificate Authority.
 - 1.2. The user requests membership to the HELIO Virtual Organization in <https://cagraidsvr10.cs.tcd.ie:8443/voms/vo.helio-vo.eu/StartRegistration.do>
2. Creation of an account with HELIO
 - 2.1. The user creates her/his own account in the CIS (This step can also be performed programmatically through the API)
3. The user uploads her/his certificate to the CIS

- 3.1. The user uploads her/his own certificate to the MyProxy Service (connected to the CIS) and defines her/his proxy login and password on the MyProxy service. This step can be performed using the tool available in <http://tools.ngs.ac.uk/ngstools/certwizard/myproxy.jnlp> and the instructions available in <http://www.ngs.ac.uk/tools/certwizard>.

Details on the procedure to setup the certwizard are available in Setup the certwizard.

Details on the procedure to use the certwizard to upload the certificates into MyProxy are available in Upload the certificate to MyProxy

- 3.2. The user adds her/his MyProxy login and password to her/his account on the CIS.
4. The user access HELIO with certificate-based security
 - 4.1. The program running on behalf of the user, when it needs an Authentication Token (HELIO Identity Token, or HIT) to be issued on her/his behalf, invokes the authentication method from the CIS.
 - 4.2. The CIS, checks that the user is a HELIO registered user issues the Authentication Token with the embedded MyProxy information.
 - 4.3. The Authentication Token is then sent to the services that need to perform Authentication and Authorization. If a services needs grid security, it can then locally download the proxy file from the MyProxy service.

Step 1 has to be performed only once although, depending on the different national policies, Grid Certificates are usually re-issued on a yearly basis.

Step 2 has to be performed only once.

Step 3 has to be when the user wants to start using certificated-based security. The validity of an update certificate is usually of a week.

Steps 4.1 – 4.3 are executed whenever a user uses the HELIO system (through its main front end or through other means such as the TAVERNA workbench) and requests certificate-based authentication.

User 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. Once a user is created it is assigned a set of standard preferences (that can be edited only by administrators), these standard preferences, once they are assigned to the user, can be customized to her/his liking.

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.

How to Access the CIS

The service is now accessible at <http://cagnode58.cs.tcd.ie:8080/helio-cis-server/>

The Graphical User Interface

The CIS exposes a simple user interface (<http://cagnode58.cs.tcd.ie:8080/helio-cis-server/>) to manage the accounts. The CIS caters for four main kinds of operations:

Simple User Operations

These operations are allowed for all HELIO users. Simple HELIO users can create and remove their account, change their password and modify the user preferences associated to their profile.

Grid User Operations

If a user owns a Grid Certificate, it can add her/his MyProxy account information. This information will be added to her/his HELIO Authentication Token and will be used by the services that require certificate-based security.

Administrator Operations

A user which is granted administrator privileges can modify the standard preferences that are given to all news users and can promote/demote users to the administrator role.

Login or create an account

Here a user can either login or create your account. When a new account is created, it will be awarded simple user (i.e. not administrator) privileges.



HELIO Community Interaction Service

If you are already registered in HELIO, enter your details below

Enter your name :

Enter your password :

If you are not a registered user, please follow this link to [Create an account](#)

Figure 3, Login Screen for the CIS

To create a new account the following fields must be entered:

- name
- email (not mandatory)
- password

Choice of role

Here a user that has successfully logged can choose the profile she/he wants to use (simple or, if she/he is entitled, administrator)

Choice of simple-role actions

Here a user can select the actions to perform



Figure 4, Simple User Actions

Change password

Here a user can change her/his password

The image shows a user interface for the HELIO Community Interaction Service. At the top is the HELIO logo, which consists of the word "HELIO" in white capital letters inside a blue and orange oval. Below the logo, the text "HELIO Community Interaction Service" is displayed. Underneath this, the text "Change Password for gab" is shown. Below that, there are two input fields: "Enter your new password:" followed by a text box, and "Re-enter your new password:" followed by a text box. At the bottom, there is a button labeled "Submit Query".

Figure 5, Change a password

See the user's preferences

Here a user can see her/his preferences

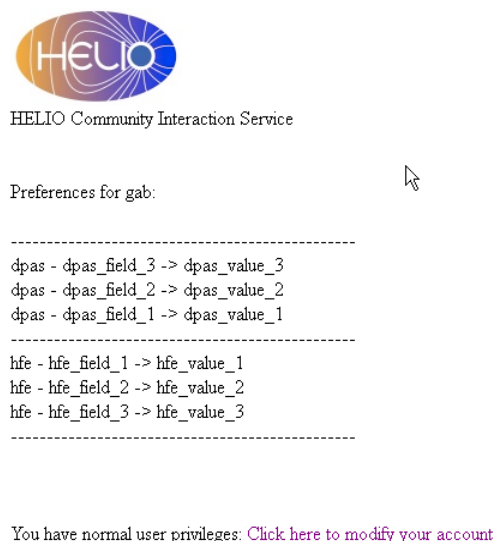


Figure 6, View the user's preferences

Modify the user's preferences

Here a user can modify the value of her/his preferences. A normal user cannot modify the general schema of the preferences (i.e. add/remove services or add/remove fields for services)

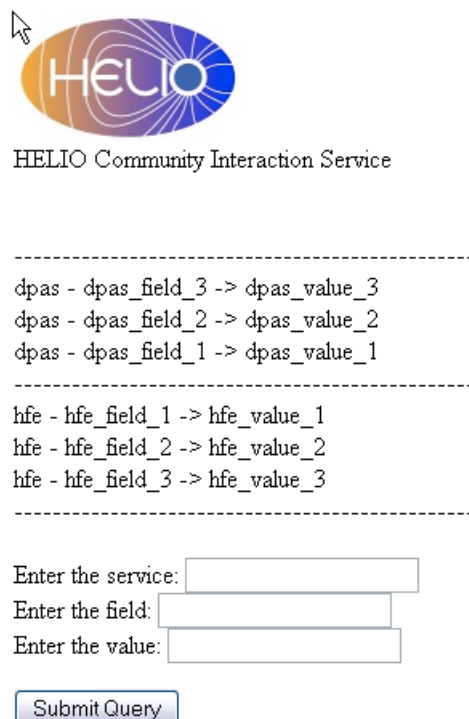


Figure 7, Modify the user's preferences

Remove your account

Here a user can remove her/his account

Add MyProxy details



HELIO Community Interaction Service

Add MyProxy information for gab

Enter your proxy login name:

Enter your proxy password:

Re-enter your proxy password:

Figure 8, Insert MyProxy Information

Here a user can add the login and password that HELIO will use to access the Grid Certificate stored in the MyProxy service.

Choice of admin-role actions



HELIO Community Interaction Service

Community Interaction Service - Administrator Page

- [Remove a user](#)
- [Modify or add a standard preference](#)
- [Remove a standard preference](#)
- [Add a user to the administrator list](#)
- [Remove a user from the administrator list](#)

Figure 9, Select the administrator actions


Here a user that has chosen to operate as an administrator can select the actions to perform

Remove an account

Here an administrator can remove the account of another user.

Modify or add standard preferences

Here an administrator can modify the default value of a preference, add a new field for an existing service or add a new service and a new field.



Now the standard preferences are :

```
dpas - dpas_field_3 -> dpas_value_3
dpas - dpas_field_2 -> dpas_value_2
dpas - dpas_field_1 -> dpas_value_1
hfe - hfe_field_1 -> hfe_value_1
hfe - hfe_field_2 -> hfe_value_2
hfe - hfe_field_3 -> hfe_value_3
```

Enter the service:

Enter the field:

Enter the value:

Figure 10, Modify the default preferences

To modify the default value:

- Put an existing service in the "Enter the service" box
- Put an existing field (for that service) in the "Enter the field" box
- Put the default value "Enter the value" box

To add a new field (with a default value) to an existing service

- Put an existing service in the "Enter the service" box
- Put a new field in the "Enter the field" box
- Put the default value "Enter the value" box

To add a new field (with a default value) for a new service

- Put a new service in the "Enter the service" box
- Put a new field in the "Enter the field" box
- Put the default value "Enter the value" box

Remove standard preferences

Here an administrator can remove a field for an existing service or remove a service (with all its preferences).

To remove a field of an existing service

- Put the service in the "Enter the service" box
- Put the field you want to remove in the "Enter the field" box

To remove a service (with all its preferences)

- Put the service in the "Enter the service" box

Add a user to the administrator list

Here an administrator can promote a simple user to administrator



HELIO Community Interaction Service
Select the account from this list : [gab, anja]

Enter the account to be promoted to administrator:

Figure 11, Promote a user to Administrator

Other Interfaces

The CIS functionalities are also accessible through SOAP protocol using the WSDL present at <http://cagnode58.cs.tcd.ie:8080/helio-cis-server/cisService?wsdl> and <http://cagnode58.cs.tcd.ie:8080/helio-cis-server/cisService/cisService?wsdl>.

This interface exposes the following methods:

- **String test(String parameter):** a simple method to test that the service is running, the parameter is not relevant but it will be returned as part of the string being returned.
- **Boolean validateUser(String name, String pwdHash):** returns true if the user defined by the name and the password hash (the function to obtain the password hash is part of the security utilities, explained in Compute password hash)
- **void addUser(String name, String pwdHash):** adds a user and the password hash to the CIS, the user will be given the standard preferences and will have only simple user role at first.
- **public void addUserWithEmail(String name, String email, String pwdHash):** adds a user, her/his email and the password hash to the CIS, the user will be given the standard preferences and will have only simple user role at first.
- **public void removeUser(String name, String pwdHash):** removes the user defined by name with credential pwdHash, if the hash of the password does not match with that stored in CIS, the method will raise an exception and will not remove the user.
- **public void removeAnotherUser(String user, String requester, String requesterPwdHash):** removes the user defined by user, in order for this command to succeed, the requester identity (requester) and her/his password hash (requesterPwdHash) must belong to a valid user with administrator role.
- **public boolean isUserPresent(String name):** returns true if the user is present.
- **public void changePwdHashForUser(String name, String oldPwdHash, String newPwdHash):** Changes the stored password hash for the user, the old password hash (OldPwdHash) must be the one stored in the CIS otherwise the method will raise an exception.
- **public String getPreferenceForUser(String user, String service, String field):** returns all the preferences for the user expressed as a single string
- **void setPreferenceForUser(String name, String pwdHash, String service, String field, String value):** sets the preferences for the service and field for the user identified by name and authenticated with her/his password hash. The structure of preferences in the CIS is explained in “See the user’s preferences.”
- **public void setStandardPreference(String userName, String computeHashOf, String prefService, String prefField, String prefValue):** sets the standard preferences (that apply to all new users) for the service and field. The user identified

by userName and authenticated with her/his password hash must have Administrator role”

- **public void removeServiceInStandardPreference(String userName, String computeHashOf, String prefService):** Removes the defined service from the standard preferences. The user identified by userName and authenticated with her/his password hash must have Administrator role”
- **public void removeFieldInStandardPreference(String userName, String computeHashOf, String prefService, String prefField):** Removes the defined field from the defined service from the standard preferences. The user identified by userName and authenticated with her/his password hash must have Administrator role”
- **public Set<String> getRolesForUser(String name):** Returns all the roles granted to the user
- **public boolean validateUserAndRole(String name, String pwdHash, String role):** returns true if the user is valid and if he is granted the defined role.
- **public HashMap<String, HashMap<String, String>> getAllPreferencesForUser(String name):** returns all the preferences of a user as an hashmap of hashmaps (upper level for services, lower level for fields)
- **public HashMap<String, HashMap<String, String>> getAllStandardPreferences():** returns all the standard preferences as an hashmap of hashmaps (upper level for services, lower level for fields)
- **public Set<String> getAllUserNames():** returns all the names of the users present in the CIS.
- **public Set<String> getAllUserNamesWithRole(String role):** returns all the names of user that are granted a certain role present in the CIS.
- **public void promoteAnotherUserToAdministrator(String userName, String computeHashOf, String anotherAccount):** grants a user (anotherAccount) administrator role, the user identified by userName and authenticated with her/his password hash must have Administrator role”
- **public void demoteAnotherUserFromAdministrator(String userName, String computeHashOf, String anotherAccount):** revokes the administrator role to a user (anotherAccount), the user identified by userName and authenticated with her/his password hash must have Administrator role”

void addGridInfoForUser(String name, String pwdHash, String gridInfo): Allows a user to add the crypted information that will be used by high-security services to use grid-certificate-based security. The string gridInfo contains in crypted version the username and

password of the proxy user. (the function to obtain the gridInfo is part of the security utilities, explained in

The method `String computeHashOf(String password)` returns the hash of the password. This method is part of the `SecurityUtilities` part of the helio-shared component.

- Encrypt Grid Information)

Sample Workflows

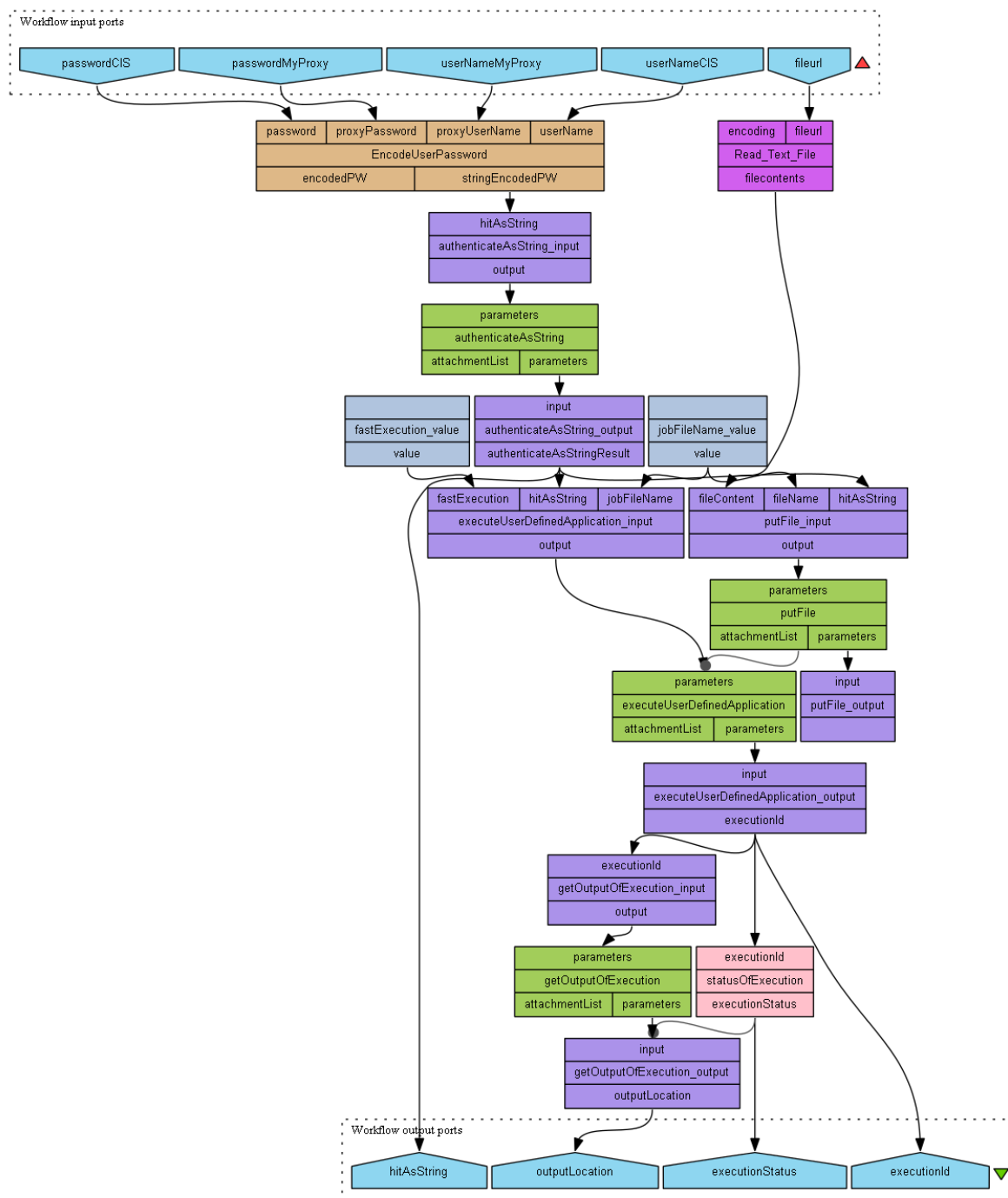


Figure 12: Example workflow using CIS for authentication

The workflow in Figure 12 uses the CIS to authenticate to the service which returns the string of the authentication token. The authentication token can be presented to the processing service (HPS) to accept user code for execution on the grid. In this workflow the URL to the users execution code is given as an input to the workflow. Additionally the users need the HELIO login and password and the login and password to the myProxy service. Note: the user's certificate must be member with the HELIO group in the virtual organization.

How to Use the CIS

Before using the CIS you need to upload your Grid certificate to the myProxy service (see "Upload the certificate to MyProxy" in Appendix). We assume the user is successfully registered and their grid identity (certificate) is associated with the HELIO virtual organisation.

To enable security the user has to encode his usernames and password before sending them to the CIS service. Helio-shared-[version].jar (URL) provides the functionality to encode this information.

```
import eu.heliovo.shared.common.SecurityUtils;
import eu.heliovo.shared.common.SerializationUtils;
import eu.heliovo.shared.common.HIT;

SecurityUtils      secUtilities      =      new SecurityUtils();
SerializationUtils serUtilities      =      new SerializationUtils();

stringEncodedPW =serUtilities.toString( new HIT(userName,
        secUtilities.computeHashOf(password),
        secUtilities.prepare(proxyUserName, proxyPassword)));
```

The those encoded string can be used as input to the 'authenticationAsString' function of the CIS. The returned string from this function call should be used as input 'hitAsString' of the functions 'putFile' and 'executeUserDefinedApplication' of the processing Service.

The remaining notes concern the processing service which follows after the CIS authentication. Please note that the order of execution of the functions is significant.

1. upload of executable code; function 'putFile' parameters file content – the string of the code ('fileContent'), the file name of the code ('fileName'), and the returned security token from the CIS ('hitAsString')
2. execute the code on the processing service; function 'executeUserDefinedApplication' with the parameters 'fastExecution' (supported values: true, false), 'jobFileName' that is the name you have given your code in Step 1, and the 'hitAsString', the returned security token from the CIS; the returned value is the 'exectuionId' which you will need to access your computation.
3. check the status of the execution; function 'getStatusOfExecution' the parameter is the execution Id returned by the previous step. This function need to be called until the status has changed from 'Running'. Please note that the execution on Grid facilities usually requires the queuing of the job until it will be executed. It is not advisable to check very frequently (twice a minute is sufficient).
4. request results; function 'getOutputOfExecution' with the input parameter of the execution Id (output of step 2). As result of this operation you receive the URL to the location of your output directory. There are usually two directories in that, one with the results and one with the error message.

Appendix A

The Certwizard is a tool that allows the management of Grid Certificates and the connections to the MyProxy services

Setup the certwizard

Once the certwizard has been installed in the user's computer, it must be configured with the following actions.

Load the grid certificate into the certwizard

The first step once the user has obtained her/his certificate from her/his national certificate authority is to configure the certwizard to use it. If the certificate is available as pem files, then the procedure is described in Figure 13, if the certificate is available as p12 file, then the procedure is described in Figure 14.

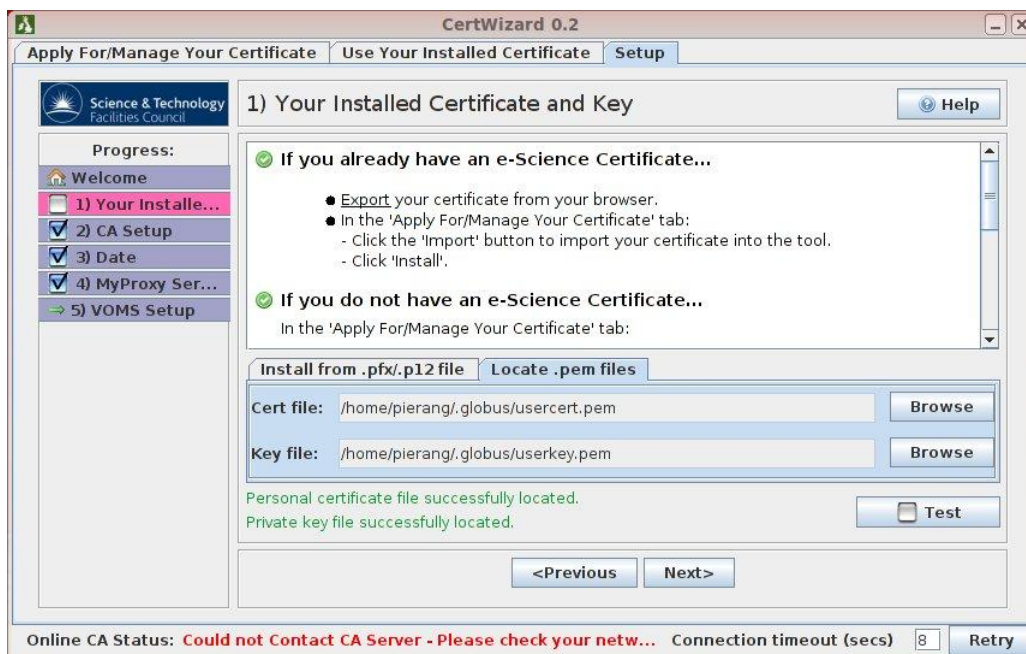


Figure 13, Load the grid certificate (as pem files) into the certwizard

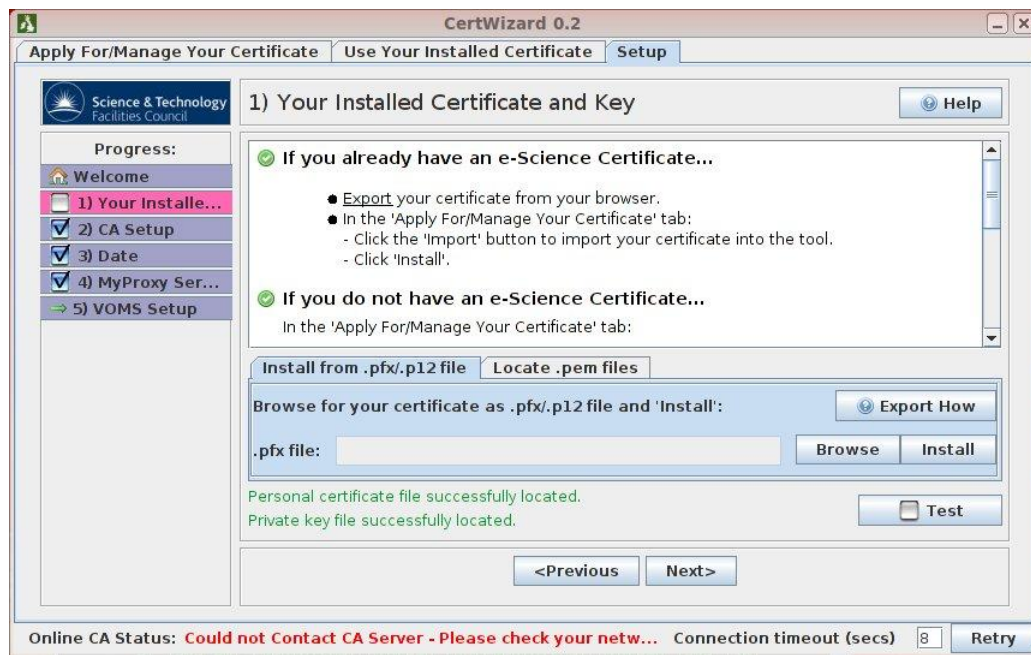


Figure 14, Load the grid certificate (as p12 files) into the certwizard

Setup the certificate authority

After loading the user certificate to the wizard, the user must set the Certificate Authority to the wizard; to add the Grid Ireland Certificate Authority to the wizard the user must:

- Download the Grid Ireland Certificate Authority certificate from <https://www.tacar.org/cert/list> in pkcs7 format (Figure 15 and Figure 16)
- Setup the Grid Ireland Certificate Authority in the certwizard by pressing the ‘Add New’ button and selecting the location of the downloaded certificate (Figure 18 and Figure 18).

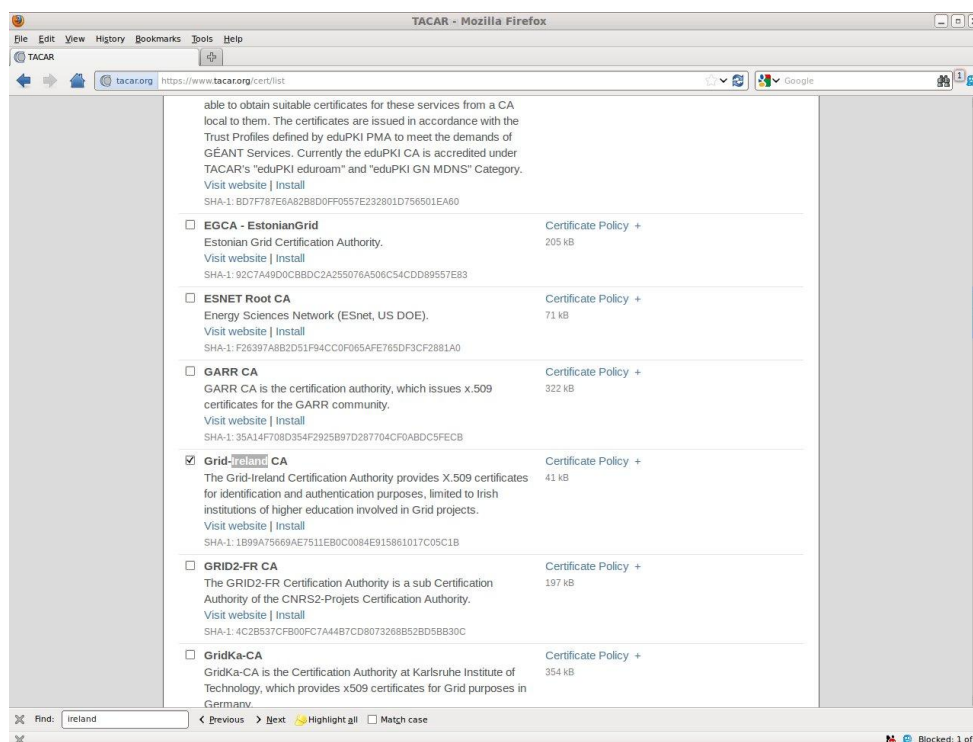


Figure 15, Select the certificates for the Grid Ireland Certificate Authority

Service Name – Admin Guide

Version 0.1

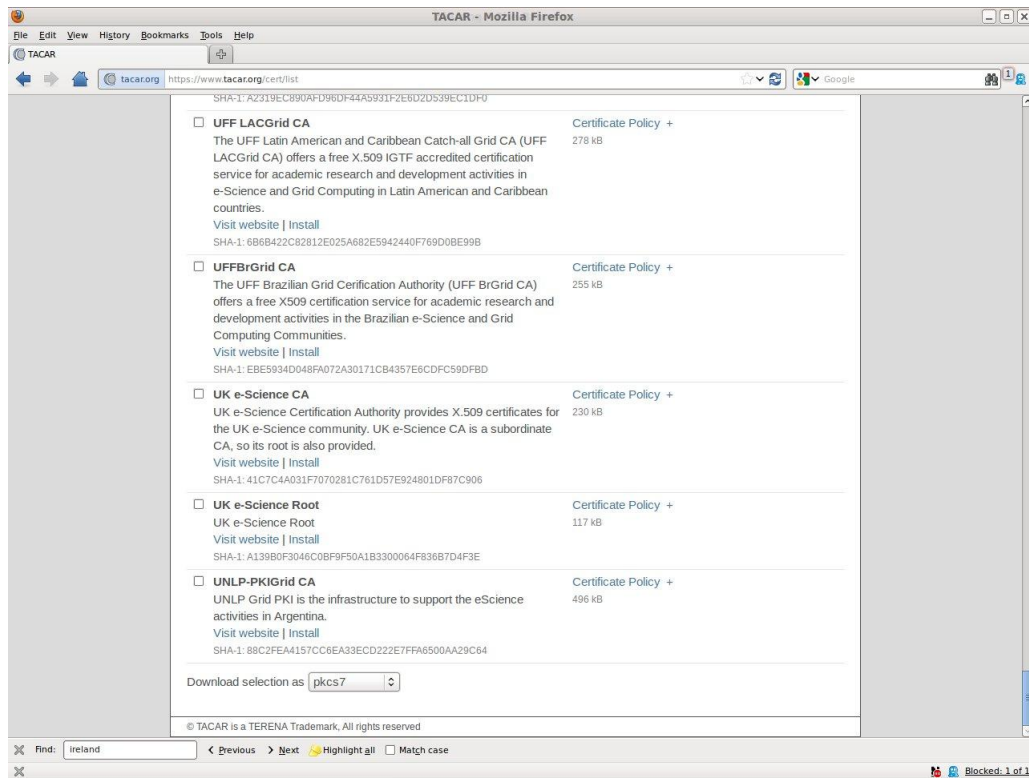


Figure 16, Select the pkcs7 format for the Grid Ireland Certificate Authority

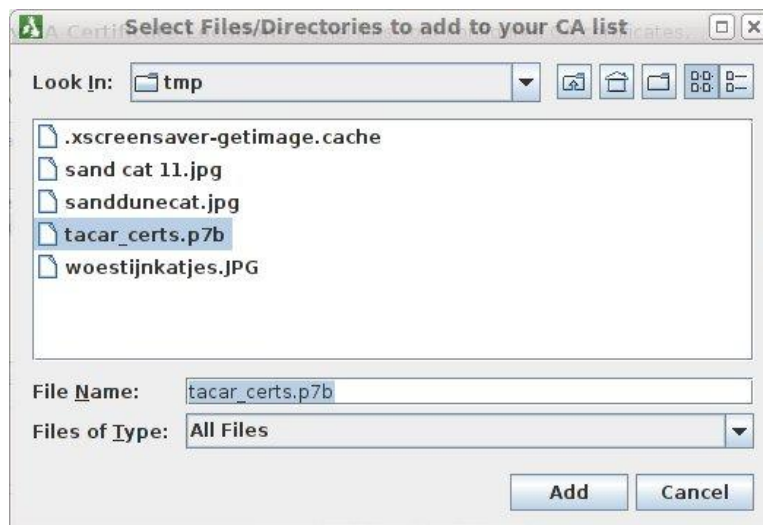


Figure 17, Select the downloaded certificates

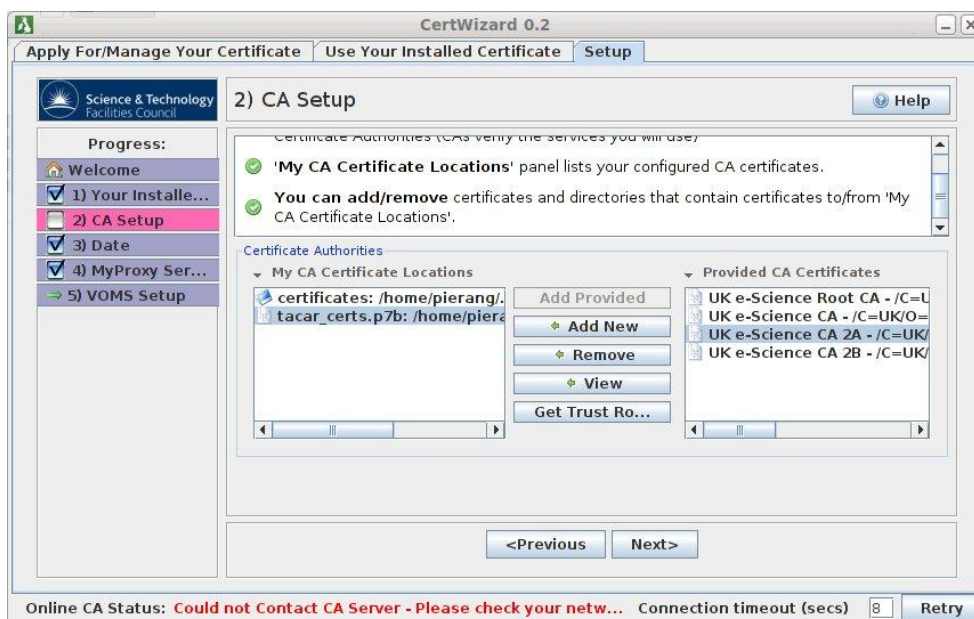


Figure 18, Setup the Grid Ireland Certificate Authority

Test that the time on the user machine is updated

Certificate-based security requires that the machine where the wizard runs has the time set correctly within a certain tolerance; the page described in Figure 19 allows to test that the time is correct within the required tolerance.

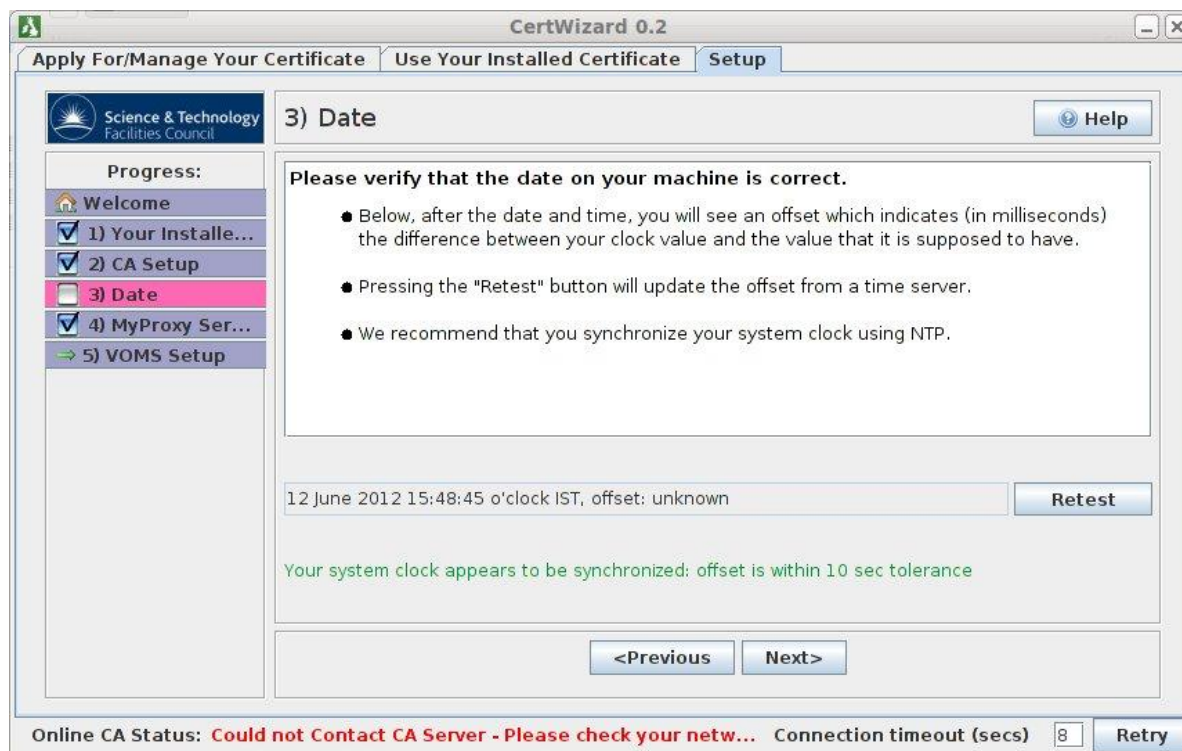
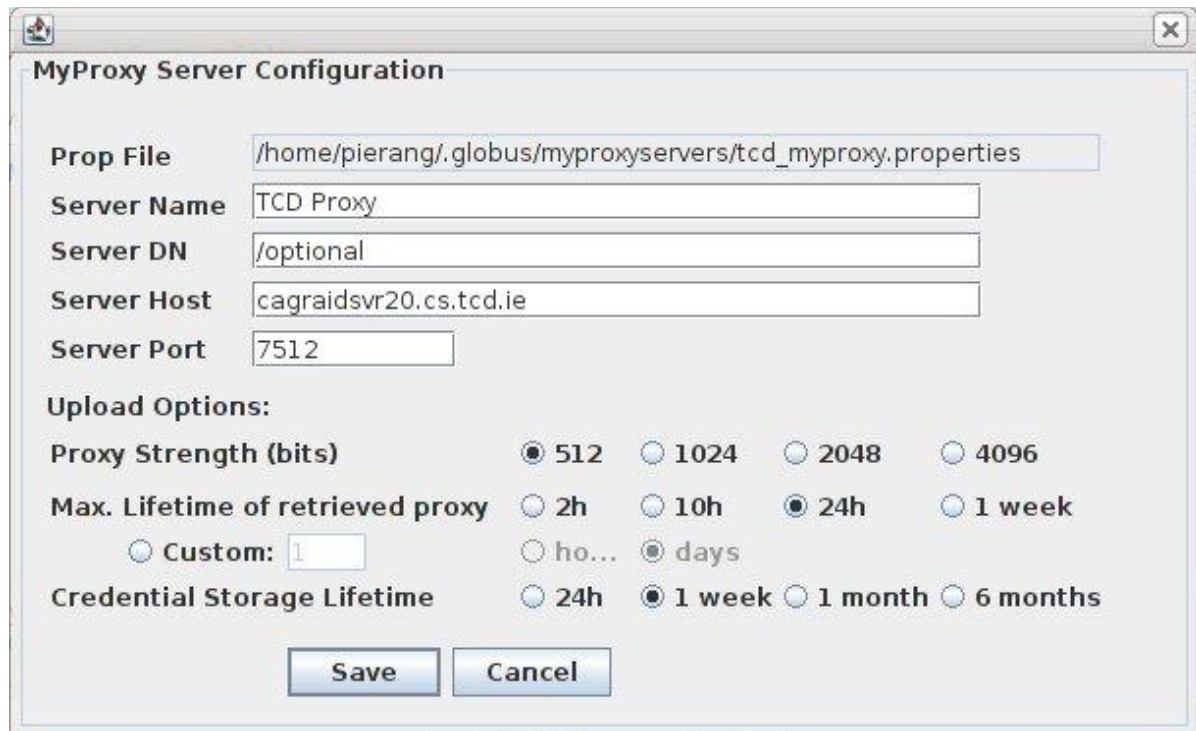


Figure 19, Testing that the time is correct

Setup the MyProxy server

The final step in the setup in the certwizard is to enter the details of the MyProxy server used by Grid Ireland. The details of the MyProxy server are defined in Figure 20.



The image shows a 'MyProxy Server Configuration' dialog box with the following fields and options:

- Prop File:** /home/pierang/.globus/myproxyservers/tcd_myproxy.properties
- Server Name:** TCD Proxy
- Server DN:** /optional
- Server Host:** cagraidsvr20.cs.tcd.ie
- Server Port:** 7512
- Upload Options:**
 - Proxy Strength (bits):** ☒ 512 ☐ 1024 ☐ 2048 ☐ 4096
 - Max. Lifetime of retrieved proxy:** ☐ 2h ☐ 10h ☒ 24h ☐ 1 week
☐ Custom: 1 ☐ ho... ☒ days
 - Credential Storage Lifetime:** ☐ 24h ☒ 1 week ☐ 1 month ☐ 6 months
- Buttons:** Save, Cancel

Figure 20, Enter the MyProxy details for Grid Ireland.

Test the validity of the certificate

Once the certwizard has been configured, the user can test if the settings are correct by checking the status of her/his certificate as described in Figure 21.

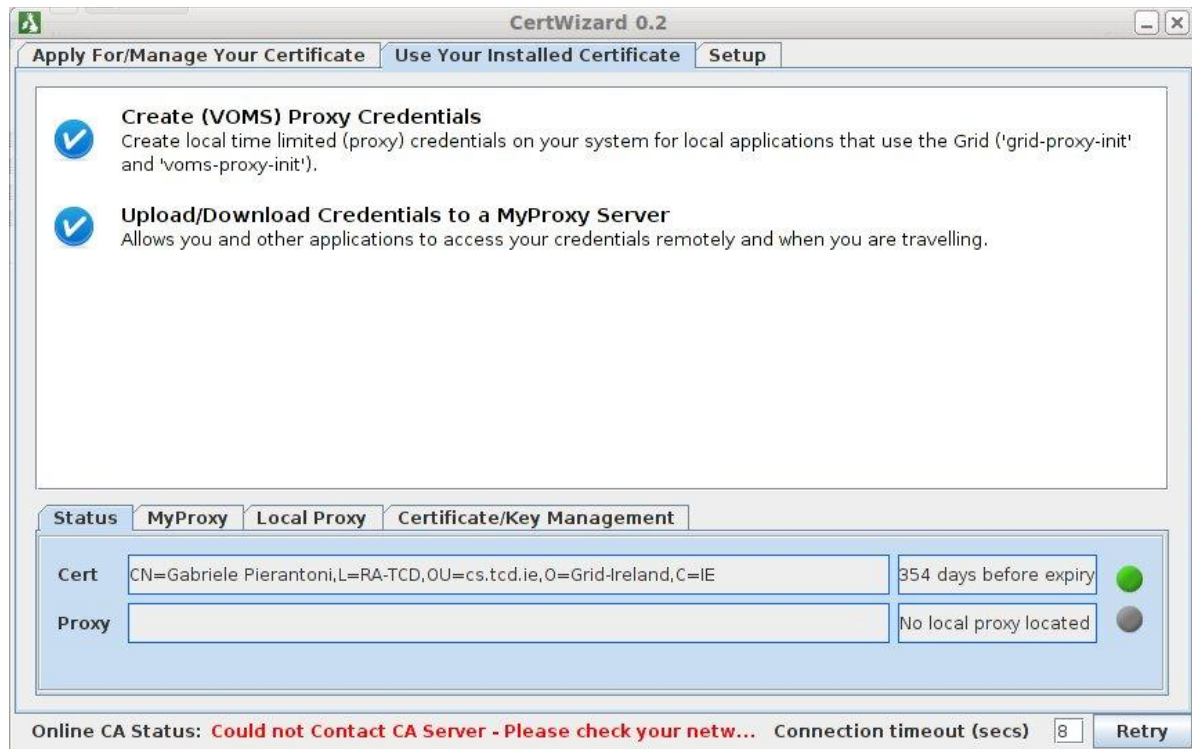


Figure 21, test if the settings are correct.

Upload the certificate to MyProxy

Once the certwizard has been configured, the user can upload her/his certificate to the MyProxy server. Once this last step has been completed the user will be able to use grid-enabled security.

In order to perform the upload, the user must select the right MyProxy server (Figure 22) and then upload the certificate to the MyProxy server (Figure 23).

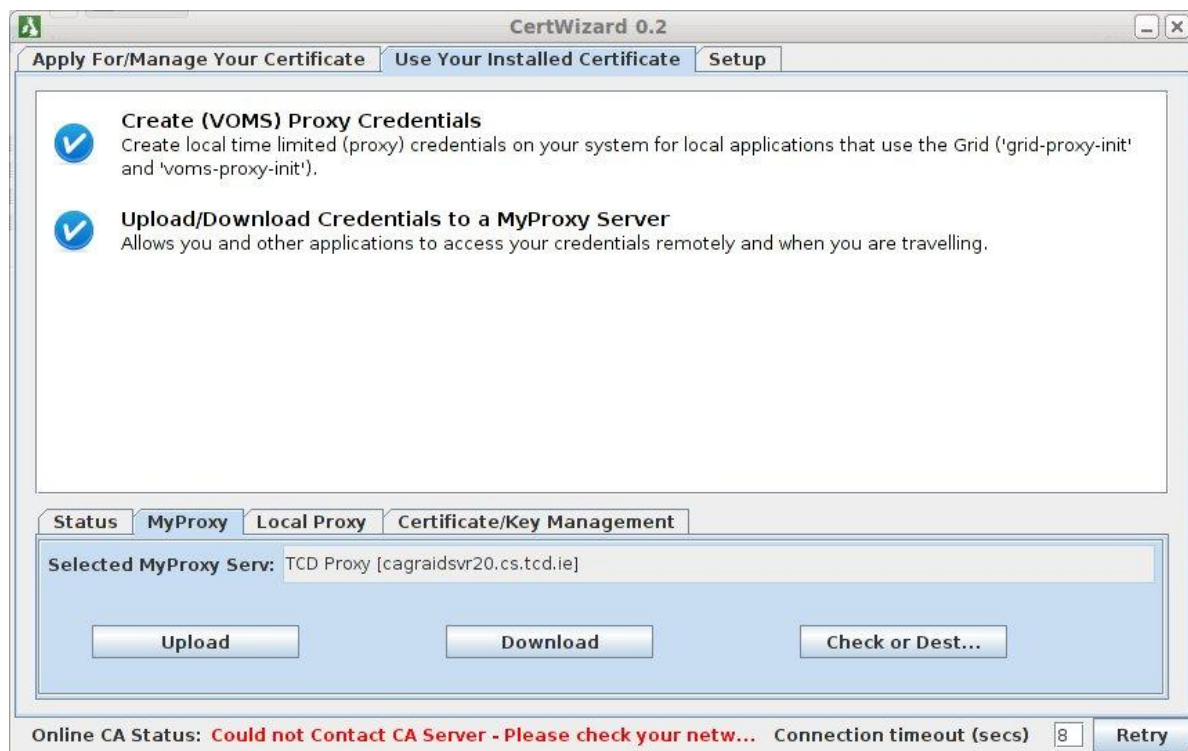


Figure 22, Select the MyProxy server.

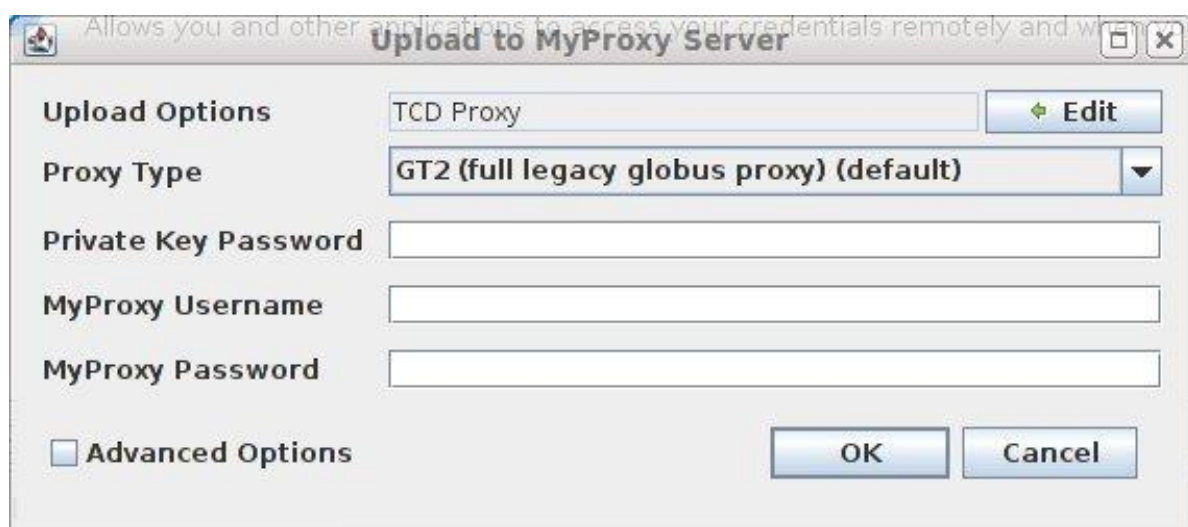


Figure 23, enter the login details.

The login details are:

- **Private Key Password:** This is the password that the user has defined for her/his certificate when it requested it from her/his certificate authority.
- **MyProxy Username:** This is the same username used to add the MyProxy information to the CIS (Add MyProxy details).
- **MyProxy Password:** This is the same password used to add the MyProxy information to the CIS (Add MyProxy details).

Appendix B

Compute password hash

The method `String computeHashOf(String password)` returns the hash of the password. This method is part of the `SecurityUtilities` part of the `helio-shared` component.

Encrypt Grid Information

The method `String prepare(String proxyUserName, String proxyPassword)` returns a crypted string that contains user name and password to access the proxies stored in the `MyProxy` server. This method is part of the `SecurityUtilities` part of the `helio-shared` component.

Works Cited

Spring Security. (n.d.). Retrieved from Spring Security: <http://static.springsource.org/spring-security/site/>