

EDGI repository Administrator Manual

1. Introduction

This manual documents the EDGI application repository. Section 2 describes the key entities, actors and use cases. Section 3-9 describe the GUI and section 10 describes the servlet interface and GridFTP access.

2. Entity, Actor, and Use-case Specification

Entity Definitions

Application. This entity represents an application for example modelling and simulation applications. It describes the inputs and outputs and explains what the application does. However it does not actually contain any files necessary to run the application itself because there can be different implementations available e.g. for different operating systems.

Implementation. This entity represents an implementation of an application. It strictly follows the input and output definitions of the application and implements the functionality given in the application description. It contains or references (via e.g. URLs) all the files and also holds other data/metadata necessary to run the application on a given platform. An implementation goes through a validation process and is eventually deployed on a resource. Implementations have a list of sites where they are or can be installed.

Platform. This entity describes in which desktop Grid and/or service Grid environment the implementation can be executed.

Files. This entity contains the implementation files required to run the applications.

Actor Definitions

E-scientist. This actor is typically a scientist (likely from other disciplines than Computer Science or Information Technology) who wants to run applications either through a user friendly interface or through a user interface he/she is familiar with. For this he/she wants to search and browse the repository to find those applications he/she is interested in.

Application Developer. This actor is a computer scientist who has knowledge of middleware and applications to be run on this middleware. He wants to enable e-scientists to run their applications using the EDGI infrastructure. To achieve it he has to be able to define applications, implementations and sample inputs (also called sample configurations). After implementing the applications the actor marks them as non-validated. Their applications should go through a validation process. The actor is also interested in the outcome of the validation process and any comments on applications, implementations and sample files.

Application Validator. This actor is a computer scientist who wants to test implementations created by application developers and give feedback. The actor should find non-validated applications submitted by application developers, download packages and sample inputs and

attempt to run the applications. After successful validation he gives a feedback about the application and makes it available for Administrators by marking them as validated.

Desktop Grid Administrator. This actor manages a desktop grid, such as a BOINC or ExtremWeb resource. He wants to allow e-scientists and application developers to use available resources. To achieve it the actor should install applications he can trust. He does this by finding and downloading validated implementation packages relevant to the grid type, operating system and architecture he manages. He also wants to give feedback on the implementation he uses to help the application developers.

Modified Computing Element (mCE). This actor represents the modified Computing Element which interacts with the repository to submit application from service grids to desktop grids. The EDGI Repository will work with ARC, gLite and Unicore modified CEs.

Use Cases

There are five use cases representing five actors using the EDGI infrastructure. At one end is the E-scientist's use case and at the other end is the Desktop Grid Administrator's use case. The Application Developer's and the Application Validator's use case are between them. The use cases are built on each other. For example operations included in the E-scientist's use cases are available for all other actors.

E-scientist

This actor is the consumer of the contents of the repository to use services based on it (see Figure 1).

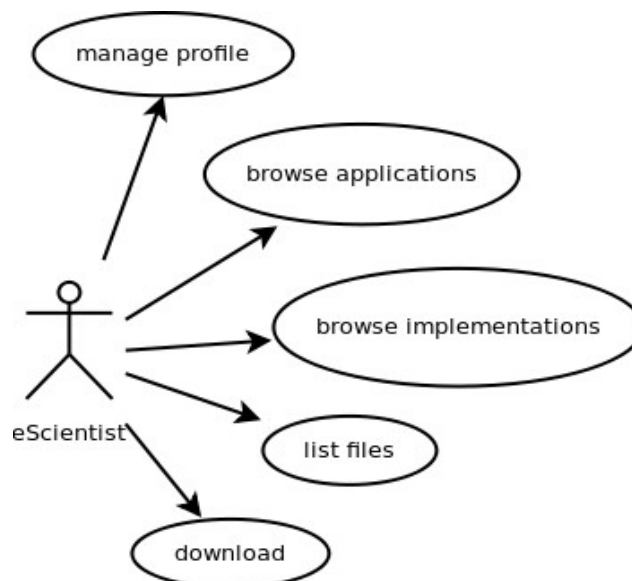


Figure 1 E-scientist's Use Case

- **Manage profiles (see section 3)**
It enables e-scientist to manage their profiles, i.e upload, display, modify and delete their data.
- **Browse applications (see section 7)**
Browse includes filtering and listing applications based on their metadata.
- **Browse implementations (see section 8)**
It allows e-scientists to browse the implementations of applications stored in the repository.

- **List files (see section 7 and 8)**
E-scientists can list files related to the application selected by the “Browse applications” operation.
- **Download (see section 7 and 8)**
E-scientists can download application and implementation files.

Application Developer

This actor is the creator and maintainer of the repository contents (see Figure 2).

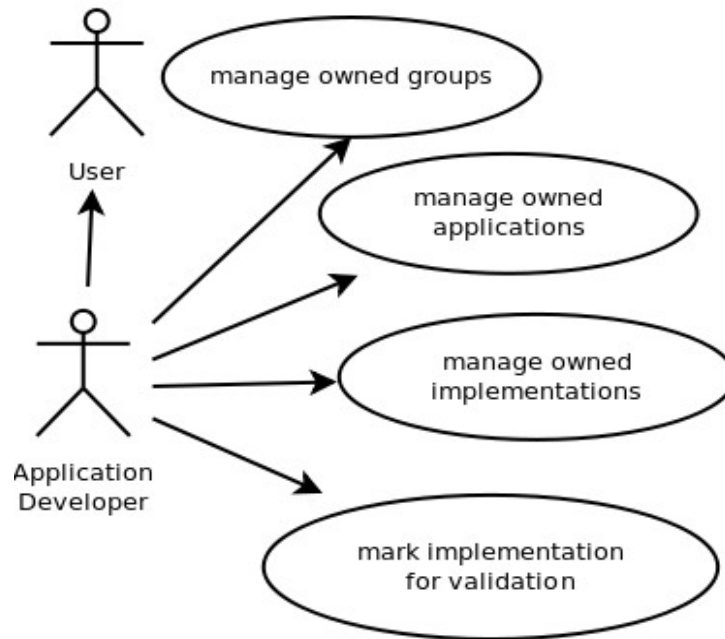


Figure 2 Application Developer's Use Case

- **Manage owned groups (see section 5)**
Application developers can add and remove actors from their groups.
- **Manage owned application (see section 7)**
Application developers are allowed to upload, modify, delete and download their applications of the repository.
- **Manage owned implementation (see section 8)**
Application developers are allowed to upload, modify, delete and download implementations of their applications.
- **Mark implementation for validation (see section 9)**
Application developers are allowed to mark their applications available for validation.

Application Validator

Application validators download application packages from the repository marked for validation, do the validation, change the status of the applications and give feedback. However they do not provide any other services. Their job is only to test/validate applications (see Figure 3).

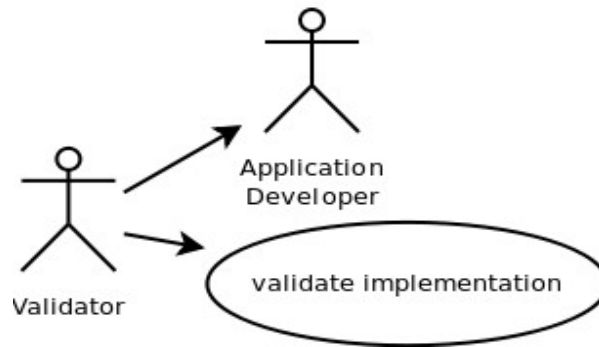


Figure 3 Application Validator's Use Case

- **Validate implementations (see section 9)**

Validators are allowed to validate implementations marked by application developers for validation. The validation includes the following operations: download implementations and files needed to run them, and change the status of the implementation from non-validated to validated after a successful validation.

Repository Administrator

This actor is responsible for the management of the EDGI Application Repository (see Figure 4).

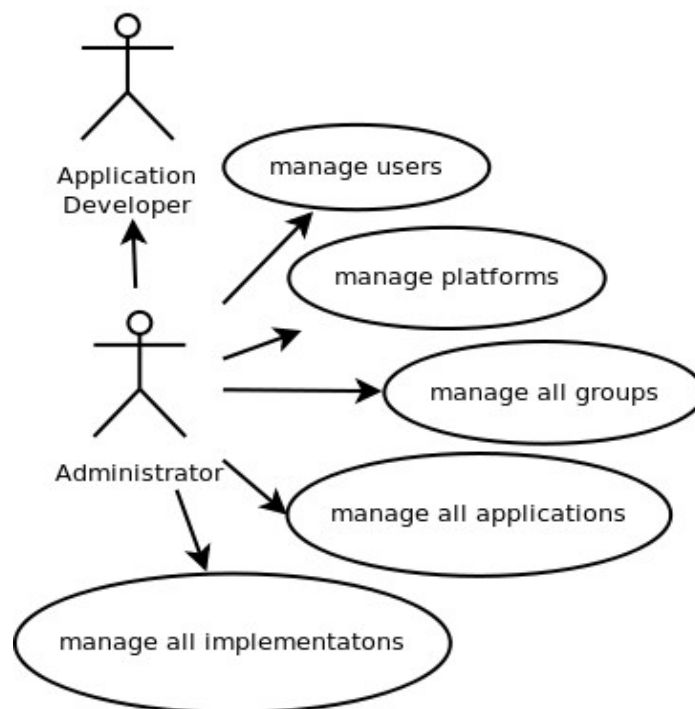


Figure 4 Desktop Grid Administrator's Use Case

- **Manage users (see section 4)**

It enables administrators to manage actor data, i.e. register users, display, modify and delete their data.

- **Manage all groups (see section 5)**

It enables administrators to manage groups, i.e. create and delete groups, display and modify group data.

- **Manage platforms (see section 6)**

It enables administrators to manage platform, i.e. add and remove platforms, display and modify platform's data.

- **Manage all applications (see section 7)**
Administrators are allowed to upload, modify, delete and download applications to/from the repository.
- **Manage all implementations (see section 8)**
Administrators are permitted to upload, modify, delete and download implementations of the applications of the repository.

Modified Computing Element

This actor represents the modified Computing Elements as an external component which interacts with the repository (see Figure 5).

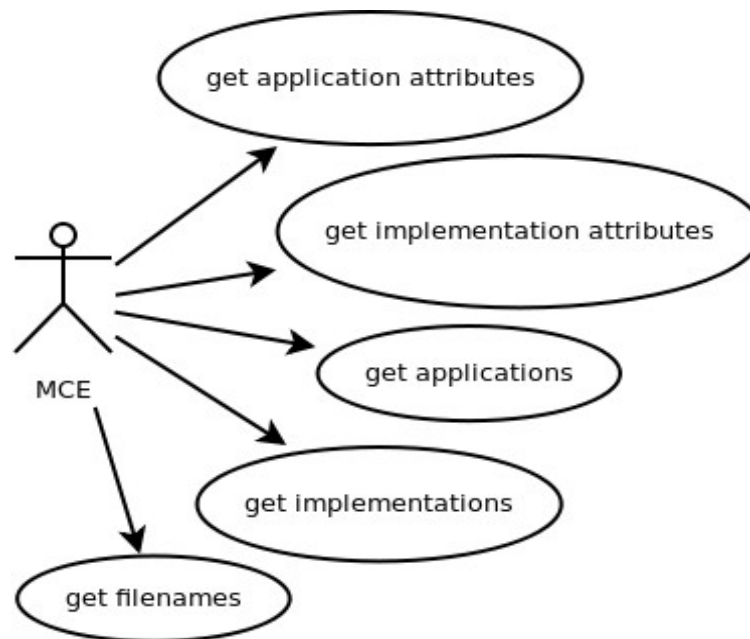


Figure 5 Modified Computing Element's Use Case

- **Get application attributes (see section 10)**
mCE is able to get application attributes.
- **Get implementation attributes (see section 10)**
mCE is able to get implementation attributes.
- **Get applications (see section 10)**
mCE is able to download applications stored in the repository.
- **Get implementations (see section 10)**
mCE is able to download implementations of applications stored in the repository.
- **Get files (see section 10)**
mCE is able to get files required by implementations stored to run them.

Repository model

Users represent all actors (e-scientists, application developers, application validators, desktop Grid administrator and modified Computing Elements). They may own applications and can access implementations of applications to run them (see Figure 6).

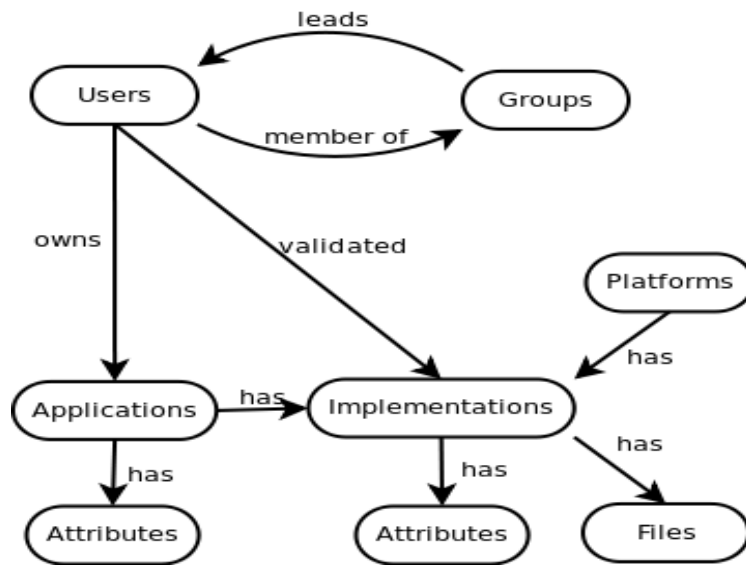
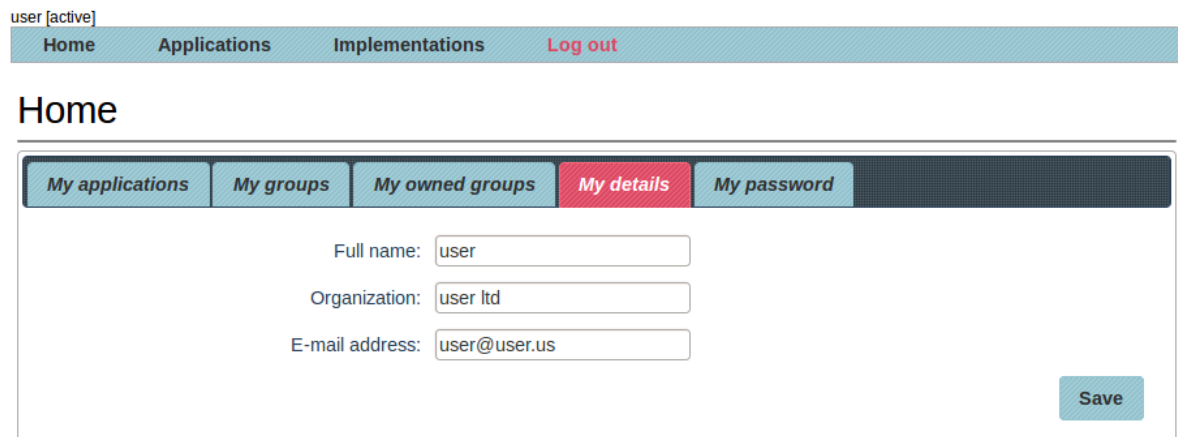


Figure 6 Repository Model

The repository enables application developers and repository administrator to create and manage groups. Groups support controlled access to applications and their implementations.

2. Manage User Profile

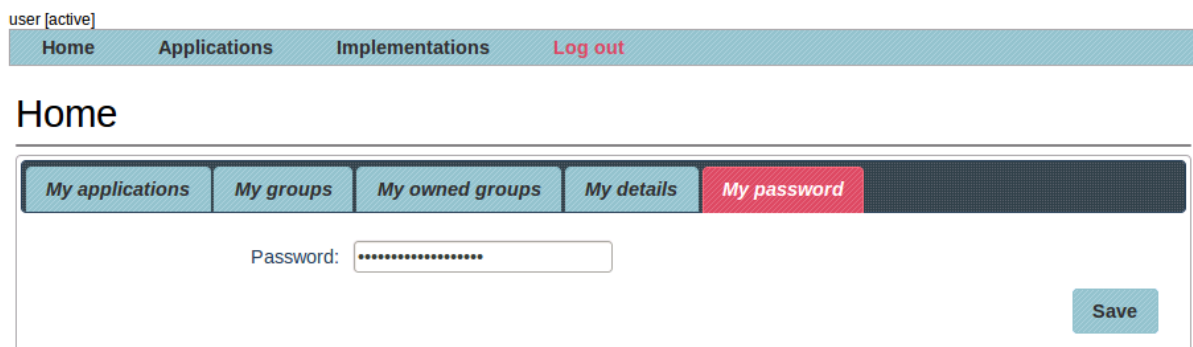
To change user details go to Home->My Details, where the full name, organisation and email address can be updated, as illustrated in figure 7.



The screenshot shows a web interface for managing a user profile. At the top, there is a header bar with the text "user [active]" on the left and navigation links: "Home", "Applications", "Implementations", and "Log out" (highlighted in red). Below the header, the word "Home" is displayed. A secondary navigation bar contains five tabs: "My applications", "My groups", "My owned groups", "My details" (highlighted in red), and "My password". The main content area of the "My details" tab contains three form fields: "Full name:" with the value "user", "Organization:" with the value "user ltd", and "E-mail address:" with the value "user@user.us". A "Save" button is located at the bottom right of the form.

Figure 7 Updating User Details

To change the user password, go to Home->My Password and define your new password. See illustration in figure 8.



The screenshot shows the "My password" page in the same user profile management interface. The header bar is identical to Figure 7. The secondary navigation bar has the same tabs, but "My password" is now highlighted in red. The main content area contains a single form field labeled "Password:" with a masked input (represented by dots). A "Save" button is located at the bottom right of the form.

Figure 8 Change user password

4. UserManagement

Create users

Administrators can create new users through the *Action -> New* tab on the *Users* page.

Mr Sam Smith [active, validator, admin]

Home	Applications	Validation	Users	Groups	Platforms	Log out
------	--------------	------------	-------	--------	-----------	---------

Users

Users

25 (1 of 1)

New

Login name	Full name	Organization	Roles
admin1	Mr Sam Smith	Westminster	active, validator, admin
user1	User One111	CPC one	active
test1	Mr Test One	ONE	active, validator
noam	Noam Weingarten	cpc	active, validator, admin
user01	user01	qqqss	active
user02	user02	qerr	active
vdor01	vdor1	sssss	active, validator
vdor02	sgsgsg	gsgsgg	active, validator
admin01	lllll	lllll	active, validator, admin
tryws 1	white space in name	wwwwwwqqq	active
user03	user03	qqqqqq	active
montest1a	monday test active 1	test	active
monad1	mon test admin 1	est	active, admin
monval1	mon valid remake	test	active, validator
monus1	user01	testqq	active
monother	non group member	test	active
mongroupee	group member	test	active
zsolt1	dfgsdg	sdfgsdg	active
monvalgroupee	validador groupee	test	active, validator

Figure 9 Users page

Browse and list users

The Users page displays all users. It also allows searching users based on their Login name, Full name or Organization and list users whose data meet the search criteria.

Clicking on user's login name opens the user page which displays user related data and enables changing it.

Modify user details

user01

Actions

Details	Password	Groups	Owned groups
---------	----------	--------	--------------

Full name:

User 01

Organization:

test

E-mail address:

test@a.com

Active:

☒

Validator:

☐

Administrator:

☐

Save

Figure 10 User page – Details

Administrators can display data of the selected user, particularly:

The *Details* tab presents users' *Full name*, *Organization*, *E-mail* and [role](#).

The *Password* tab displays the password.

The *Groups* tab displays the groups this user is a member of.

The *Owned groups* tab displays the groups this user is the leader of.

Administrators can use

the *Details* tab to change users' *Full name*, *Organization*, *E-mail* and [role](#).

the *Password* tab to change the password.

Roles

The user page allows the definition of the following roles:

- | | |
|----------------------|--|
| Active | user (or e-scientist). |
| Validator | user with the additional rights to <i>validate</i> implementations for example application developers. |
| Administrator | users who have full control over groups, users, applications and implementations. |

Delete users

From the Users page administrators can use the *Action* -> *Delete* tab to delete a user.

Note: if there are any user associations, i.e. owned groups or applications, the ownership of these entities should be transferred to someone else before deleting the user.

5. Group Management

Create groups

The *Actions* -> *New* tab on the Groups page can be used to create a new group.

Browse and list Groups

Mr Sam Smith [active, validator, admin]

Home	Applications	Validation	Users	Groups	Platforms	Log out
------	--------------	------------	-------	--------	-----------	---------

Groups

25 (1 of 1)

Actions
New

Group name	Leader
group1	user01
group3	user1
group4	user1
group5	user01
group6	user1
group7	user01
group10	test1
groupnew	admin1
group01	admin1
group7a	admin1
gg1	admin1
mongrp1	user01
zsoltgroup1	zsolt1
zsoltgroup2	zsolt1
mygroup1	monus1

Figure 11 Groups page

The Groups page can be used to list Groups and search by Group name.

Manage groups and group members

Mr Sam Smith [active, validator, admin]

Home	Applications	Validation	Users	Groups	Platforms	Log out
------	--------------	------------	-------	--------	-----------	---------

group6

Actions
Delete

Details Users

Leader: user1

Change leader

Figure 12 Group page - details

Administrators and group owners can click on a group to open the Group page where they can modify the data of the selected group.

From this page

the *Details* tab can be used to change the leader of the group

the *Users'* tab can be used to list, add or remove members

Delete Group

Administrators and group owners can use the *Actions* -> *Delete* tab to delete groups.

6. Platform Management

Create Platform

The Actions -> New tab can be used to create new platforms.

Browse and list platforms

Mr Sam Smith [active, validator, admin]

Home	Applications	Validation	Users	Groups	Platforms	Log out
------	--------------	------------	-------	--------	-----------	---------

Platforms

Platforms

25 (1 of 1)

Actions

New

Name	Description
linux-amd64-sles-10sp1	SUSE Linux Enterprise with service pack 1 runni...
boinc-windows-xp-32	Windows XP desktop grid using BOINC ttttt
new platform	1111111
monday	77699

Figure 13 Platforms page

The Platforms page can be used to list and browse current platforms using their Names.

Modify Platform

By clicking on a platform, its description can be modified.

Delete Platform

From the Platforms page, the Actions -> Delete tab can be used to delete platforms.

Note: Associated implementations should be either removed or associated with a different platform.

7. Application Management

Create Applications

The Actions -> New tab on the Applications page can be used to create a new application, and specify its details.

Note: The application creator will be the initial owner. Actors other than administrators can only associate applications with groups they own.

List and search applications

Mr Sam Smith [active, validator, admin]

Home Applications Validation Users Groups Platforms Log out

Applications

25 (1 of 1) New

Name	Owner	Group	Description
app15	admin1	group01	fefefef
app3	user1	group1	test
dsp	noam	group1	Details about the application
aa2222	test1	group10	hjhjhjh
appn1222	user01	group4	qqqqq
monapp1	monus1	mongrp1	test1
monap2	monus1	mongrp1	testsat
monNP	monus1	mongrp1	no permissions
monGP	monus1	mongrp1	Group Read
monGRD	monus1	mongrp1	Group Read Download
monGRDM	monus1	mongrp1	Group Read Download Modify
monGROR	monus1	mongrp1	Group Read Other Read
monGRDOR	monus1	mongrp1	Group: Read Download Others: Read
monGRDMOR	monus1	mongrp1	Group: Read Download Modify Others: Read
monGRDMORD	monus1	mongrp1	Group: Read, Download, Mdlify Others: Dwnld, ...
monaaa	monus1	mongrp1	aaaa
todelete	monus1	mongrp1	qqq

Figure 14 Applications page

The Applications page can be used to list and search applications by Name, Owner or Group.

Modify applications

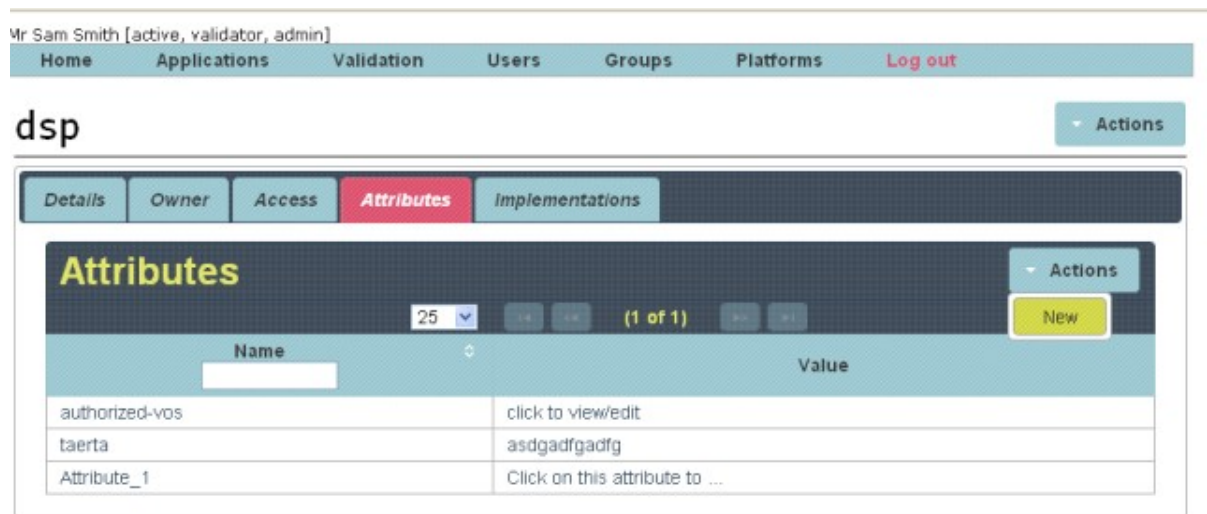


Figure 15 Application page - attributes

Clicking on an application opens the Application page.

The *Details* tab allows administrators and application owners to edit the description of the application.

The *Owner* tab can be used to change the ownership of the application

The *Access* tab can be used to change the group the application is associated with, and the [access controls](#) of the applications.

The *Attributes* tab can be used to list, add and modify attributes of the selected application.

The *Implementations* tab can be used to [manage implementations](#) of the selected application.

Application Attributes

Application attributes can be listed by clicking on the Attributes tab of a particular application.

Table 1 describes the metadata attributes and provides example values. Figure 16 presents the application metadata structure. These attributes allow straightforward categorisation of applications and

improve significantly the browsing and search operations. The input and output attributes with their sub-attributes define inputs and outputs of applications. The

configuration attribute specifies values of input parameters passed to application inputs and they can also specify example outputs.

Figure 17 illustrates the application attribute page and the attribute table, where the left column contains the

attribute names and the middle column the attribute values. The right column

presents operations which can be performed on attributes as actions. There are three supported actions (or operations): add, edit and remove, depending on a particular attribute.

In addition, extra attributes as key-value pairs can be also defined for any application by clicking on Actions->Add new attributes. Attribute keys have to be unique and should not contain character '!'. Note that the attribute table is not automatically synchronised with the underlying database. In order to save changes, click on the save button.

Application metadata			Example value	Description
id			1001	application identifier
name			Factorial	application name
description			test app for mCE devs	application description
domain			Mathematics	scientific domain
keywords			factorial, integer	application keywords
inputs	\			list of input files
	file0001	\		first input file
		logical filename	1.txt	filename that the application expects
		description	this file contains an integer	file description
outputs	\			list of output files
	file0002	\		first output file
		logical filename	2.txt	filename that the application expects
		description	this file contains the factorial of the input integer	file description
configurations	\			List of input/output configurations
	conf0001	\		First configuration
		file0001	input.dat	example file for file0001
		file0002	output.dat	example file for file0002

Table 1 Application metadata attributes

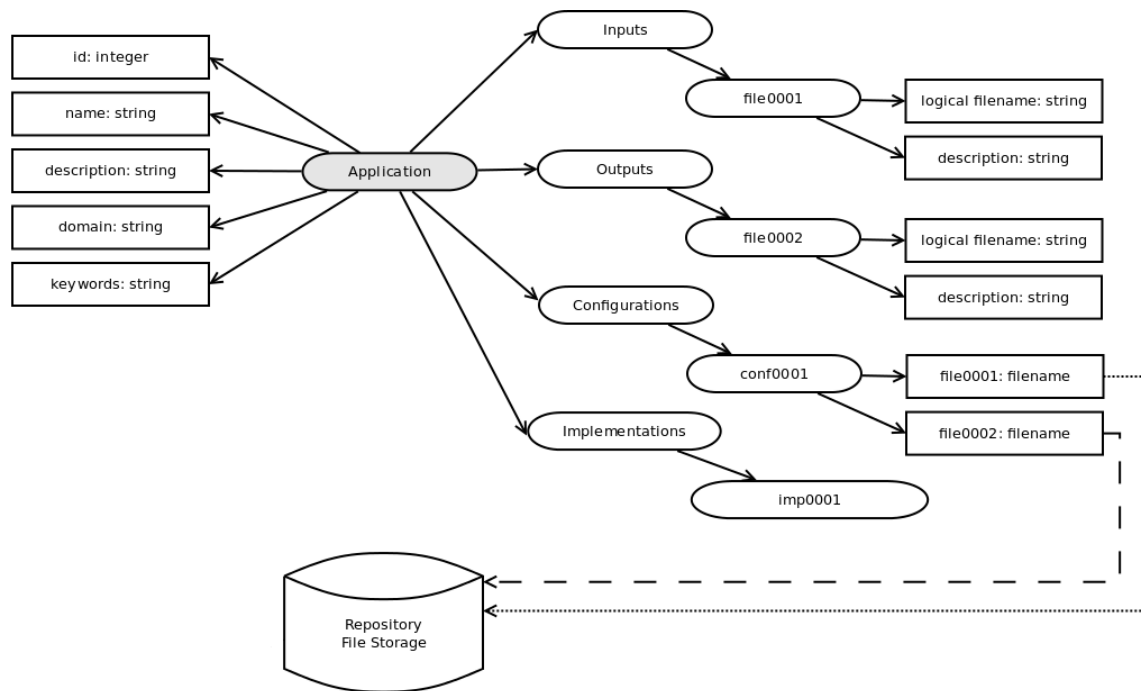


Figure 16 Application metadata structure

Application: Autodock

Actions

Details

Owner

Access

Attributes

Files

Implementations

Attributes

Save

Actions

Name	Value	Actions
▼ inputs		Add
▼ port0001		Remove
description	Protein molecule files packaged into a single archive	Edit
logical filename	inputs.zip	Edit
▼ port0002		Remove
description	AutoDock docking parameter file	Edit
logical filename	docking.dpf	Edit
▼ outputs		Add
▼ port0003		Remove
description	AutoDock docking log file	Edit
logical filename	log.dlg	Edit
▼ configurations		Add
▼ conf0001		Add Remove
port0001	example_inputs_1.zip	Edit Remove
port0002	example_docking_1.dpf	Edit Remove
port0003	example_docking_log_1.dlg	Edit Remove
domain	bioinformatics	Edit
keywords	autodock, docking, protein, receptor, ligand	Edit

Figure 17 Application attribute table

Application's access control

Mr Sam Smith [active, validator, admin]

Home Applications Validation Users Groups Platforms Log out

dsp Actions

Details Owner Access Attributes Implementations

Group:

Read	Download	Modify
Group: <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Others: <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Published: ☐

Save

Figure 18 Application page - access control

The *Access* tab, allows an administrator or an application owner to specify various types of access to either group-members or other users.

Management of application files

Management of files is the same in the case of both applications and implementations. Application files can be uploaded, downloaded and deleted similarly as in the case of implementations described in the next section. Application file management view can be opened by clicking on the files tab of a particular application.

8. Implementation Management

Add implementation to application

Implementations of an application can be added by using the Action -> New tab.

Note: Only application owners of the application and administrators are allowed to add implementations to applications.

List implementations of an application

Mr Sam Smith [active, validator, admin]

Home Applications Validation Users Groups Platforms Log out

dsp Actions

Details Owner Access Attributes **Implementations**

Implementations 25 (1 of 1) New

Platform	Version	Status
boinc-windows-xp-32	1.0	old
boinc-windows-xp-32	4.1	ready
linux-amd64-sles-10sp1	1.0	deprecated
linux-amd64-sles-10sp1	3.0	validated
linux-amd64-sles-10sp1	4.4	compromised

Figure 19 Implementations page

The Implementations page can be used to search and list implementations of applications. The implementations can be searched by Platform, Version or [life-cycle Status](#).

Modify Implementation

Mr Sam Smith [active, validator, admin]

Home Applications Validation Users Groups Platforms Log out

monGP / edit_implementaion (boinc-windows-xp-32) Actions

Details Validation Attributes **Files**

Platform: boinc-windows-xp-32

Version: edit_implementaion

Save

Figure 20 Implementation page - Details

Administrators can modify implementations of any application.

The *Details* tab can be used to modify the Platform and version of the selected implementation

The *Validation* tab can be used to see the validation [life-cycle status](#).

The *Attributes* tab can be used to list, add and modify attributes of the selected implementation

The *Files* tab can be used to manage files held in the repository for selected implementation of the application.

Implementation attributes

Implementation attributes can be opened by clicking on the attributes tab of a given implementation as illustrated in figure 22. The left column of the attribute table contains attribute names and the middle column attribute values. The right column

presents operations which can be performed on attributes as actions.

Similarly to applications, the metadata template is used to help the definition of most common attributes. The four key attributes

are: exec/bundle, dependency, configuration, and VOs. The exec/bundle attribute, is a reference to the binary executable in the case of Service Grids and application bundle in the case of Desktop Grids. The

dependency attribute can be any requirement of the particular implementation. It can be for instance files, executables or libraries required for execution. Configuration attributes resolve these

dependencies.

VO attributes allow the definition of different VOs where the implementation can be submitted. For each VO a site list can also be provided.

Table 2 describes each attribute and provides example values. Figure 21 illustrates the metadata structure. Additional metadata attributes can be added by clicking on Actions->Add new attribute. Attribute keys have to be unique and should not contain character '.'. In order to save the attribute table, click on the save button.

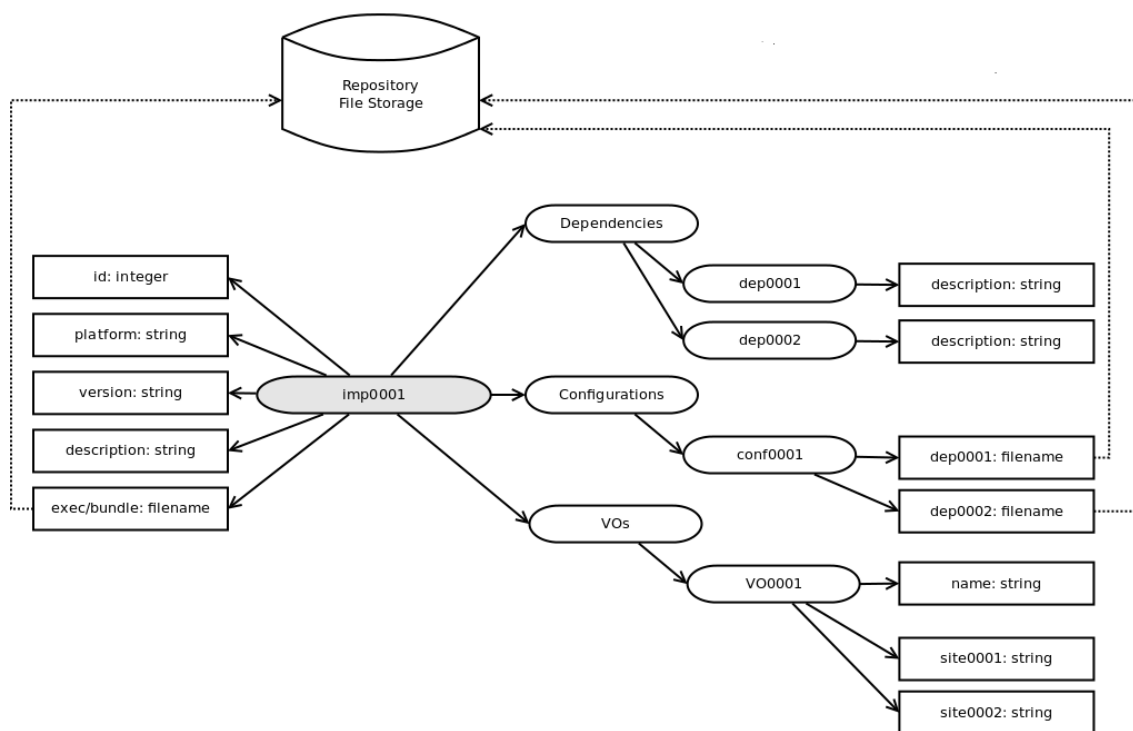


Figure 21 Implementation metadata structure

Application: dsp
Implementation: 1.0 (gLite)

[Actions](#)
[Details](#)[Validation](#)[Attributes](#)[Files](#)

Attributes

[Save](#)[Actions](#)

Name	Value	Actions
▼ dependencies		Add
▼ dep0001		Remove
description	A JDL file that contains all information needed for gLite submission.	Edit
title	Job Descriptor	Edit
▼ configurations		Add
▼ conf0001		Add Remove
dep0001	example_dsp_job_descriptor_1.jdl	Edit Remove
▼ VOs		Add
▼ vo0001		Add Remove
name	desktopgrid.vo.edges-grid.eu	Edit
site0001	ce1.grid.edges-grid.eu:2119/jobmanager-edges-extremadura	Edit Remove
site0002	ce1.grid.edges-grid.eu:2119/jobmanager-edges-uow	Edit Remove
site0003	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwla1	Edit Remove
site0004	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwripub	Edit Remove
site0005	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwripub	Edit Remove
▼ vo0002		Add Remove
name	demo.vo.edges-grid.eu	Edit
site0001	ce1.grid.edges-grid.eu:2119/jobmanager-edges-uow	Edit Remove
▼ vo0003		Add Remove
name	gilda	Edit
site0001	ce1.grid.edges-grid.eu:2119/jobmanager-edges-extremadura	Edit Remove
site0002	ce1.grid.edges-grid.eu:2119/jobmanager-edges-uow	Edit Remove
site0003	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwla1	Edit Remove
site0004	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwripub	Edit Remove
site0005	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwripub	Edit Remove
▼ vo0004		Add Remove
name	seegrid	Edit
site0001	ce1.grid.edges-grid.eu:2119/jobmanager-edges-extremadura	Edit Remove
site0002	ce1.grid.edges-grid.eu:2119/jobmanager-edges-uow	Edit Remove
site0003	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwla1	Edit Remove
site0004	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwripub	Edit Remove
site0005	ce1.grid.edges-grid.eu:2119/jobmanager-edges-xwripub	Edit Remove
description	Statically complied binary for gLite worker nodes.	Edit
executable/bundle	dsp	Edit
desktop-grid-ids	http://mishra.lpd.sztaki.hu/edgidemo/ xw.lri.fr	Edit Remove

Figure 22 Implementation attribute table

Manage implelention files

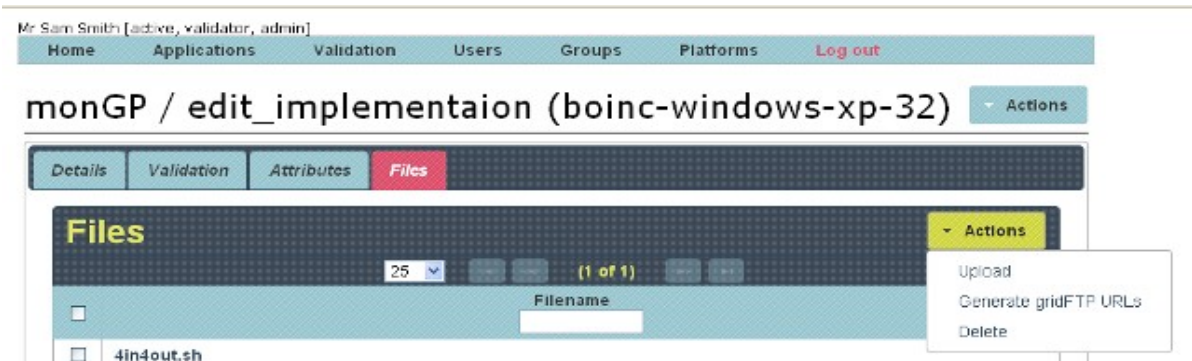


Figure 23 Implementation page - Files

Upload files to the repository

The *Actions* -> *Upload* tab from the *Files* pane can be used to upload files to the repository for the selected implementation of the application.

Download files from the repository

Administrators, as well as application owners and others (as-per [access controls](#) specified) can click on the live-link of each file of a particular implementation.

Users without permissions to download Files will only be able to see the names of the files, but there will not be live-links to download said files.

Delete files from the Repository

The *Actions* -> *Delete* tab from the *Files* tab can be used to delete selected files from the repository for that implementation of that application.

Generate gridFTP URLs of files the repository

The *Actions* -> generate gridFTP URLs from the *Files* pane can be used to generate gridFTP URLs of files. (For use in *jdl* files for Modified Computing Elements.)

9. Implementation Validation

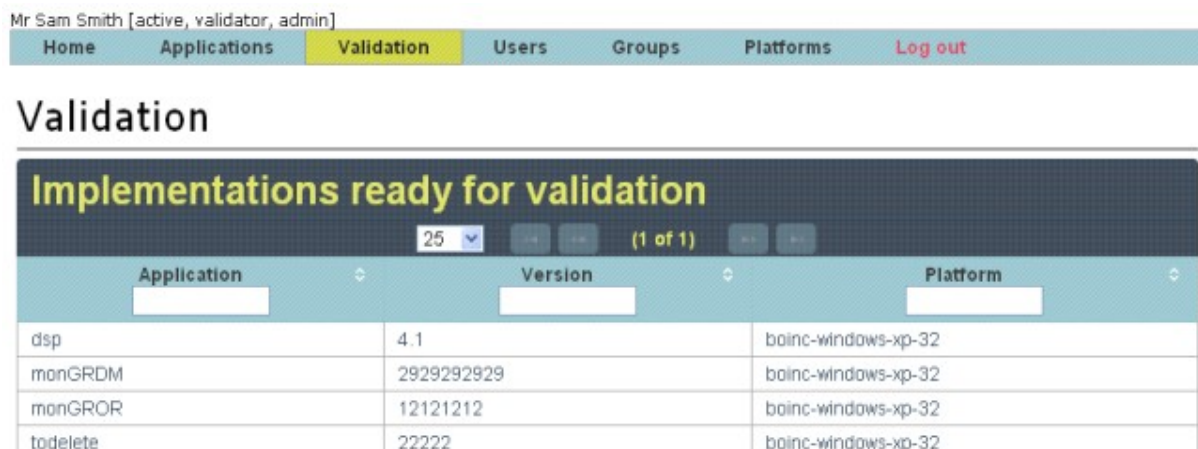


Figure 24 Implementation validation page

The *Validate* pane on the main menu lists all implementations which are ready for validation. Clicking on each will enable the validator to view the details of the implementation and make a decision.

The *Actions* control in the *Validation* pane within the implementation details will list the available actions, given the stage in the life-cycle and permissions available to that user.

The whole section above should be revised.

Change validation status of Application

Application owners are allowed to perform the following status changes

- Submit a new implementation of their own application as being ready for validation.
- Marking an implementation of their own application as being old.
- Marking an implementation of their own application as being deprecated.
- Marking an implementation of their own application as being compromised.
- Deleting an implementation of their own application.

Validators are only allowed to perform the following status changes

- Approve for validation any implementation which is currently ready for validation.
- Deny for validation any implementation which is currently ready for validation.

Administrators can perform any validation status change.

Implementation life-cycle

Figure 25. Depicts the life-cycle of an implementation.

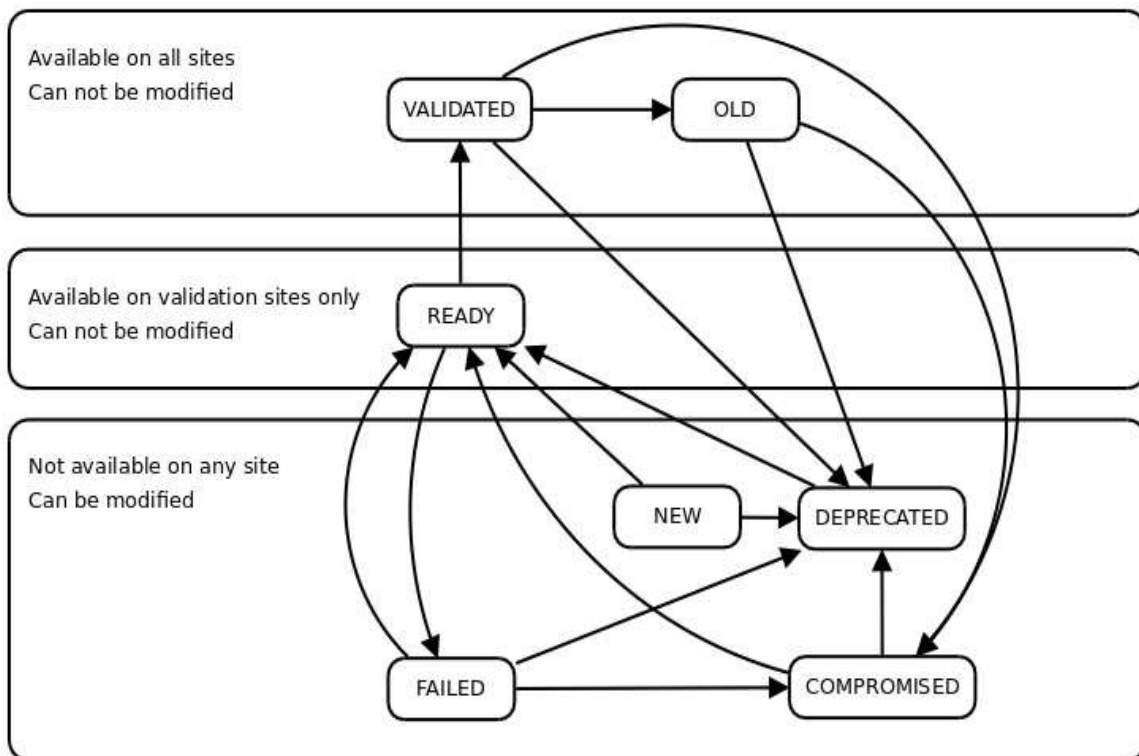


Figure 25 Implementation life-cycle

10. Servlet manual and GridFTP access

All servlets are available at `http://hostname:1234/webapp-context-root/mce/servletname`, where `webapp-context-root` is defined when the webapplication is deployed in glassfish.

Query applications

Get list of all applications and their ID numbers

`http://hostname:1234/webapp-context-root/mce/getapps`

Get list of applications with implementation attributes or specified value

`http://hostname:1234/webapp-context-root/mce/getapps?`

`impattrname=attribute.attribute name&impattrval=attribute value`

Query implementations

Get list implementations of one or more application

`http://hostname:1234/webapp-context-root/mce/getimps?`

`appids=applicationID1+applicationID2....`

Query attributes of applications

Get all the application attributes of one or more application(s)

`http://hostname:1234/webapp-context-root/mce/getappattr?`

`appids=applicationID1+applicationID2....`

Get the values of specific attributes of one or more application(s)

`http://hostname:1234/webapp-context-root/mce/getappattr?`

`appids=applicationID1+applicationID2....&attrnames=attribute name 1+attribute name 2.....`

Query attributes of implementations

Get all the implementation attributes of one or more implementation(s)

`http://hostname:1234/webapp-context-root/mce/getimpattr?`

`impids=implementationID1+implementationID2....`

Get the values of specific attributes of one or more implementation(s)

`http://hostname:1234/webapp-context-root/mce/getimpattr?`

`impids=implementationID1+implementationID2....&attrnames=attribute name 1+attribute name 2.....`

Get the URL's for implementations

Get the URL's of input files of one or more implementations

`http://repository:1234/webapp-context-root/mce/getfileurls?`

`impids=implementationID1+implementationID2....`

GridFTP access

The GridFTP service should provide read-only access and by default it should not require authentication. This way anyone can access the files placed in the repository, no username/password, is needed and any valid user proxy should be accepted.