Project Number: **RI-312579**

Project Acronym: **ER-flow**

Project Full Title:

Building an European Research Community through Interoperable Workflows and Data

Theme: **Research Infrastructures**
Call Identifier: **FP7-Infrastructures-2012-1**
Funding Scheme: **Coordination and Support Action**

Deliverable D3.1
Study of the Adaptation Options of the Simulation Platform

Due date of deliverable: 31/01/2012
Actual submission date: 31/01/2013
Start date of project: 01/09/2012
Duration: 24 months

Lead Contractor: University of Westminster
Dissemination Level: PU
Version: 3.2 (Final)

ER-flow is supported by the FP7 Capacities Programme under contract no RI-312579

1 Table of Contents

Contents

2 List of Figures and Tables

2.1 Figures

2.2 Tables

3 Status and Change History

Status:	**Name:**	**Date:**	**Signature:**
Draft:	Gabor Terstyanszky	09/01/13	n.n. electronically
Reviewed:	Peter Kacsuk		n.n. electronically
Approved:	Steve Winter		n.n. electronically

Table 1. Deliverable Status

Version	**Date**	**Pages**	**Author**	**Modification**
1.0	09/01	p6-8	Terstyanszky	Introduction + simulation platform sections
1.1	15/01	p9-12	Terstyanszky	Enhancement request + adaption options sections
1.2	18/01	p13	Terstyanszky	Adaption options - implementations
2.0	21/01/13		Terstyanszky	Completing the adaption option section
2.1	22/01/13		Terstyanszky	Revising the completed sections
3.0	24/01/13		Terstyanszky	first near complete version
3.1	30/01/13		Kacsuk	Revising Chapter 8
3.2	30/1/13		Winter	Added Table of Contents. Approved document

Table 2. Deliverable Change History

4 Glossary

Axx	Major Enhancement Request
ASM	Application Specific Module
CGI	Coarse-Grained Interoperability
DCI	Distributed Computing Infrastructure
EGI	European Grid Infrastructure
FGI	Fine-Grained Interoperability
Ixx	Minor Enhancement Request
SSP	SHIWA Simulation Platform
SZTAKI	Magyar Tudomanyos Akademia Szamitastechnikai Kutato Intezete
UoW	University of Westminster
VO	Virtual Organisation
WF	Workflow
WE	workflow engine
WP	Work package

Table 3. Glossary

5 Introduction

Workflows have become essential to integrate expertise of the application (user domain) and infrastructure domain (Distributed Computing Infrastructures - DCI) in order to optimize and support research communities. Workflows help e-scientists to formalize and structure complex scientific experiments to enable many significant scientific discoveries. Workflows represent, streamline and automate the analytical and computational steps that e-scientists need to go through from data selection and integration, computation and analysis to final data presentation and visualization. Research communities have developed different workflow systems and created large numbers of workflows to run experiments. These workflow systems are different in workflow description languages, enactment strategies and middleware providing access to infrastructures. It takes a significant effort and time to learn how to use workflow systems, and it requires specific expertise and skills to develop and maintain workflows. As a result, creating, running and maintaining workflows require substantial efforts and expertise. E-scientists would prefer workflows sharing, i.e. automatic porting of workflows across workflow systems and DCIs to optimise their efforts. Currently, the major obstacle of workflow sharing is that workflow systems are not compatible. E-scientists hesitate to learn new workflow systems to migrate their experiments to other workflow systems as this is a time-consuming and error prone process.

To address workflow interoperability the “Sharing Interoperable Workflows for Large-Scale Scientific Simulations on Available DCIs” (SHIWA) project [5.1] developed two interoperability concepts: Coarse-Grained (CGI) and Fine-Grained Interoperability (FGI). The CGI concept is based on workflow engine integration. It manages workflows as black boxes, i.e. e-scientists can use the workflows as they are and they cannot change them. The FGI concept is based on workflow translation. It is a white box approach which enables modification of workflows after translation. The SHIWA project elaborated and deployed a production-level CGI service on the SHIWA Simulation Platform (SSP) and offers a prototype-level FGI solution.

WP5 collected the requirements of the four research communities represented in the ER-flow project. WP3 in collaboration WP5 investigated how the existing simulation platform can run workflows of these communities and how the platform should be extended with required new and upgraded features and services. These work packages compiled a list of workflow systems, repositories and DCIs used by these communities. They also defined the level of interoperation and integration that should be achieved between the simulation platform and these workflow engines, repositories and DCIs.

The ER-flow project provides support for user communities to develop/select and execute workflows based on the Coarse-Grained workflow Interoperability concept. There are several reasons behind the CGI-based support. First, there is production-level CGI service available on the SHIWA Simulation Platform. Second, it requires one to four weeks to extend the CGI support for a new workflow system while it needs one to three months to develop FGI solution for a new workflow system. Third, in contrary to the SHIWA project, the ER-flow project will focus more on end users (or e-scientists) than workflow developers. End users want to run workflows previously developed and tested rather than elaborate new workflows. As a result, they do not need the white box approach of the FGI concept which allows modification of workflows.

This report will give a short overview of the SHIWA Simulation Platform in Section 6. It will describe the users, user scenarios and enhancement requests in Section 7. Finally, it will outline how WP3 will address the short-term enhancement requests.

6 Overview of the SHIWA Simulation Platform

The SHIWA Simulation Platform (SSP) addresses the challenges of (i) executing workflows of different workflow systems as non-native workflows (ii) combining workflows of different workflow systems into meta-workflows and (iii) running meta-workflows on different DCIs. The simulation platform contains the SHIWA Science Gateway and provides access to the SHIWA VO resources which belong to different DCIs of the European Grid Infrastructure (EGI), particularly DCIs based on ARC [6.1], gLite [6.2], Globus [6.3] and Unicore [6.4] middleware.

SHIWA Science Gateway contains a portal (SHIWA Portal), a submission service (GEMLCA or SHIWA Submission Service), a workflow repository (SHIWA Repository), a legacy code repository (GEMLCA Repository) and a proxy server (SHIWA Proxy Server) to support the whole workflow life-cycle using the Coarse-Grained Interoperability concept. The SHIWA Portal is built on the WS-PGRADE portal technology. It enables certificate/proxy and DCI resource management. It is integrated with the WS-PGRADE Workflow System which is used as native workflow engine in the simulation platform. The portal offers a graphical workflow editor, a workflow engine and workflow execution monitor. The SHIWA Repository stores the workflow (workflow graph, workflow description, workflow implementation and configuration). The SHIWA Submission Service incorporates the GEMLCA Repository which stores the workflow engine data (legacy code description and workflow engine binaries) and workflow executables. Workflow and workflow engine developers can describe, modify and delete workflows and workflow engines through the repository GUI and portal's admin portlet, respectively. The SHIWA Repository offers a wide-range of browse and search features for e-scientists and workflow developers. To support non-native workflow execution the SHIWA Submission Service imports the workflow from the GEMLCA Repository. This service either invokes locally or remotely pre-deployed workflow engines or submits workflow engines with the workflow to local or remote resources to execute workflows. The SHIWA Proxy Server enables management of multiple certificates when the workflows are executed on different DCIs and VOs.

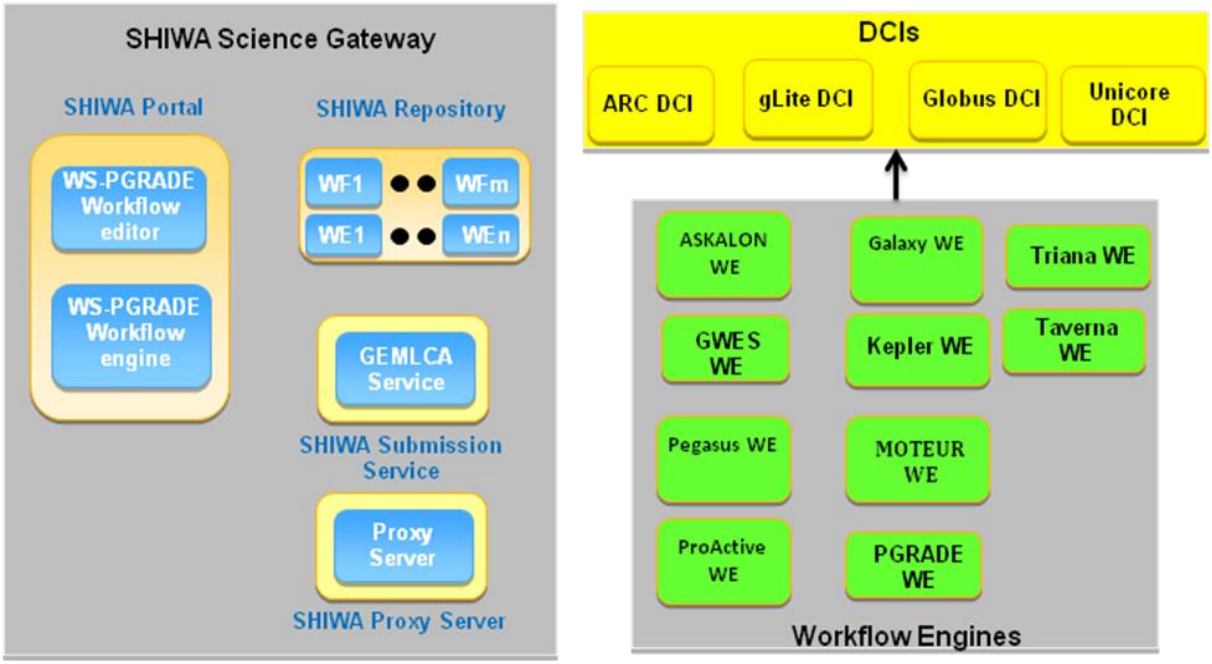

Figure 6.1: SSP CGI Architecture

where $\mathbf{WF_1,\ldots WF_m}$ - workflow data and metadata,
$\mathbf{WE_1,\ldots WE_n}$ - workflow engine data and binary

Currently the simulation platform provides CGI support for the following **workflow systems**: ASKALON [6.5], Galaxy [6.6], GWES [6.7], Kepler [6.8], MOTEUR [6.9], WS-PGRADE [6.10], Pegasus [6.11], ProActive [6.12], Taverna [6.13] and Triana [6.14]. It should be emphasized that it requires minimum efforts to extend the CGI support for other workflow systems. The SHIWA project consortium created the **SHIWA VO** to enable user authorization across the DCIs providing resources for the workflow execution. The project deployed the dedicated shiwa-workflow.eu VOMS server to allow access to the VO. The VO incorporates resources from the Austrian, British, Dutch, French Hungarian, and German National Grid Infrastructures.

The ER-flow project has three technology provider partners: CNRS, SZTAKI and UoW. These partners are responsible for adaption and management of the SHIWA Simulation Platform. Particularly, UoW manages the SHIWA Portal, the SHIWA Repository and the SHIWA Submission Service which are deployed at the University of Westminster. CNRS handles the SHIWA Proxy Server. SZTAKI is responsible for upgrading the SHIWA Portal based on user requirements. The ASKALON, Galaxy, Kepler, MOTEUR, P-GRADE Taverna and Triana workflow system have been deployed at the University of Westminster while the others were installed remotely.

7 Enhancement Requests and Adaptation Options of the SHIWA Simulation Platform

The SHIWA Simulation Platform supports three types of users:

- End user (or e-scientists)
- Workflow developer
- Workflow system developer

The **end users** (or e-scientists) are the domain researchers (e.g. biologists, chemists, etc.) who would like to run a workflow in order either to reproduce an experiment represented as a workflow in the SHIWA Repository or to make their own experiment with their own data sets. To run workflows they want to search/browse the SHIWA Repository to find and download workflows to the SHIWA Portal to execute them. The **workflow developers** are computer scientists who have knowledge of workflows and of the workflow system which execute these workflows and of the middleware where workflows are executed. They want to enable end users to run workflows. To achieve it they define workflows and their implementations and configurations, and they upload all this data to the repository and make it public. The **Workflow system developers** are computer scientists who develop the various workflow systems. They upload the workflow system's description into to the SHIWA Repository to support execution of non-native workflows.

The SHIWA project elaborated twelve user scenarios targeting both workflow developers and end users. These user scenarios cover the Coarse- and Fine-Grained workflow Interoperability and the Distributed Computing Infrastructure Interoperability. The ER-flow project will support the following four user scenarios which are based on the CGI concept:

1 **Publish executable workflows in the SHIWA Repository**
   actors: workflow developers
   services: SHIWA Repository
   SHIWA Desktop (optional)
   operations: Actors can upload, upgrade and delete workflows (data, metadata and executables). This scenario requires registration with the SHIWA Repository. There are two ways to upload workflows into the repository. The actors can either use the "Table View" of the repository to specify workflow data or they can use the SHIWA Desktop to publish workflows in the repository. Currently, the SHIWA Desktop is integrated with the ASKALON, MOTEUR and Triana workflow system.
2 **Browse and search executable workflows in the SHIWA Repository**
   actors: end-users + workflow developers + workflow system developers
   services: SHIWA Repository
   SHIWA Portal
   operations: Actors can browse or search the repository in "Browse View" and "Table View" and select those workflows they want to execute. This scenario does not require registration to access the workflows stored in the SHIWA Repository. Actors can access the repository through the repository GUI or through the Repository Portlet of the SHIWA Portal.
3 **Create and Run non-native workflows available in the SHIWA Repository**
   actors: end-users + workflow developers + workflow system developers
   services: SHIWA Portal
   SHIWA Repository
   SHIWA Submission Service
   operations: Actors would like to run a non-native workflow found in the SHIWA Repository on the SHIWA Portal. They create the abstract and the

concrete workflow representing the non-native workflow selected in user scenario No. 2 in the SHIWA Portal. They configure the concrete workflow retrieving the relevant information about the workflow from the SHIWA Repository. They can use either the default input parameters or define their own. Next, the actors submit the workflow through the portal. Finally, the submission service retrieves the workflow executable from the SHIWA Repository and submits it to the workflow system.

Remark: After creating and configuring the workflow the actors can re-submit them at any time skipping the editing and configuration phase.

4 Create and execute meta-workflows through the SHIWA Simulation Platform

actors: end-users + workflow developers + workflow system developers

services: SHIWA Portal
SHIWA Repository
SHIWA Submission Service

operations: Actors would like to combine workflows found in the SHIWA Repository. Chosen workflows are written in different workflow languages. Actors create the abstract and the concrete workflow combining native jobs and workflows plus non-native workflows in the SHIWA Portal. They configure the concrete workflow retrieving the relevant information about the non-native workflows from the SHIWA Repository. Next, the actors submit the workflow through the portal. Finally, the submission service retrieves the required workflow executables from the SHIWA Repository and submits them to the associated workflow systems.

The SHIWA project targeted mainly the workflow developers and the workflow system developers. The ER-flow project is extending the target group to end users. WP5 collected the prospective applications of the Astrophysics, Computational Chemistry, Heliophysics and Life Science community which will be ported to the SHIWA Simulation Platform. WP3 analysed these applications and defined their requirement towards the simulation platform (See Table 7.1)

	workflow system	portal	repository	Infrastructure
Astrophysics	WS-PGRADE	WS-PGRADE	none	Italian NGI
Computational Chemistry	GWES Unicore	WS-PGRADE	none	German NGI
Heliophysics	Taverna	none	myExperiment	Irish and UK NGI
Life Science	GWES, MOTEUR WS-PGRADE	WS-PGRADE	SHIWA Repository	Dutch and German NGIs

Table 7.1: Application Requirements towards the Simulation Platform

These requirements raise two issues. First, the CGI concept does not support the Unicore workflow system. Second, the simulation platform does not have access to Italian and Irish NGI resources. WP3 will discuss these outstanding issues with the involved user communities and elaborate the relevant solutions.

WP3 presented the SHIWA Simulation Platform at the ER-flow kick-off meeting and collected feedback from the four user communities represented in the project consortium. The user communities compiled an enhancement request list including major and minor enhancements. (See Table 7.2 and Table 7.3)

7.1 Major enhancement request:

No.	major enhancement requests	partner	schedule
A01	To upgrade the **SHIWA Portal** to offer better performance and more reliable services.	UoW	ST
A02	To simplify **workflow description** using the SHIWA CGI bundle and SHIWA Desktop concept for WS-PGRADE workflows	SZTAKI + UoW	ST
A03	To implement **single-sign-on** to the SHIWA Portal and the SHIWA Repository and use the SHIWA Portal as a single point of entry to the simulation platform	SZTAKI + Uow	ST
A04	To support **robot certificate** to offer simple access for both workflow developers and end users eliminating current certificate management issues. This solution should offer proper log information about users who submitted workflows, preferably via a GUI.	SZTAKI	ST
A05	To provide either the **end-user interface** or **ASM based interface** for e-scientist considering their requirements	UoW	MT
A06	To re-engineer the **SHIWA Submission Service** replacing GT4 with web services and upgrading its functions	UoW	MT
A07	To **monitor the components of the simulation platform** (portal + repository + submission service + workflow systems)	SZTAKI + UoW	MT
A08	To create and support **personalized access to the SHIWA Repository**, i.e. access to the public or the private view of the repository considering the user profile	UoW	MT
A09	To collect information about workflow execution(domain to which workflow belongs, user who initiated the execution, etc) about executed workflows including both meta and sub-workflows	SZTAKI	MT
A10	To define and publish the **SHIWA Repository API** to support upload, listing, searching and downloading workflows	UoW	MT
A11	To define and publish the **SHIWA API** to expose the simulation platform services which manage workflow execution	SZTAKI + UoW	MT
A12	To create a **personalised view** (my own workflows, my favourite workflows, etc.)	UoW	MT
A13	To **integrate** login with **EGI SSO, facebook, gmail**, etc. (login is needed only for a few, restricted activities. Don't use login where there is nothing to protect)	SZTAKI + UoW	MT
A14	To enable **automatic workflow execution environment deployment** on cloud plus the consequent workflow execution on cloud	SZTAKI + UoW	LT
A15	To support **workflow downloading** in CGI bundle format without the SHIWA desktop concept	SZTAKI + UoW	LT
A16	To extend **ASM with SHIWA API** to allow users to manage workflow data and execution	SZTAKI	LT
A17	To support **monitoring of non-native sub-workflow execution** and retrieve proper fault information	SZTAKI	LT

Table 7.2: Major Enhancement Requests

Legends: ST – short-term MT – medium-term LT – long-term

WP3 defined three categories of the major enhancement requests: short-, medium- and long-term requests and identified the partners who will be involved in addressing these requests. Currently the work package is working on short-term enhancement requests and will complete most of these by M06. WP3 will investigate the medium-term enhancement requests in M07 and will identify those requests which it will implement.

7.2 Minor enhancement request:

No.	major enhancement requests	partner	schedule
I01	**To check** the **workflows** and allocate them to the proper and relevant domains (image processing, Image processing should be merged, etc.) in the SHIWA Repository	UoW	ST
I02	To support proper **workflow versioning** in the SHIWA Repository	UoW	ST
I03	To remove **welcome page of the SHIWA Repository** and provide direct access to the browse page of abstract workflows	UoW	ST
I04	**Controlling characters** entered into the SHIWA Repository (lowercase, uppercase, white spaces should not matter in the repository)	UoW	ST
I05	To enable **automatic re-submission of failed non-native sub-workflows**	SZTAKI	ST
I06	To assign **URL to each workflow implementation** uploaded to the repository, the URL should guide to workflow implementation stored in the repository	UoW	MT
I07	To create and **display lists of the latest 5-10 workflows** and the most frequently used workflows	UoW	MT
I08	To enable **access to the simulation platform via Facebook**	SZTAKI	MT
I09	To enable **downloading Taverna 2 workflows** from the myExperiment repository	UoW	MT
I10	Input type 'File' should be more specific for example identifying the file format? (e.g. GIF file for ImageMerger wf)	UoW	MT
I11	Validation checks abstract workflow but no concrete (or instance workflows) are validated. It should be explained on the repository GUI.	UoW	ST
I12	To **transfer workflows** uploaded to the SHIWA Repository to the EGI App database	UoW	MT
I13	To allow users to **move** from the end user view to the **developer view** by a single click assuming that user will access the end user view after login	SZTAKI	MT
I14	To **test workflows** uploaded to the repository on cloud or local resources by a single click	???	MT

Table 7.3: Minor Enhancement Requests

Legends: ST - short term MT - medium term LT - long term

Similarly to the major enhancement requests WP3 defined three categories of minor enhancement requests: short-, medium- and long-term requests and identified the partners who are involved in addressing these requests. The work package is working on short-term enhancement requests and will complete these by M05. WP3 will investigate the medium-term enhancement requests in M06 and will identify those requests which it will implement.

8 Addressing Enhancement Requests

8.1 Major Enhancement Requests

8.1.1 Upgrading the SHIWA Portal (A01)

Issue: In the portal the job configuration window presents the list of all simulation platform enabled workflows. Currently there are more than 100 workflows and this number increases on regular basis. As a result, it's difficult and error-prone to find the workflow binaries.

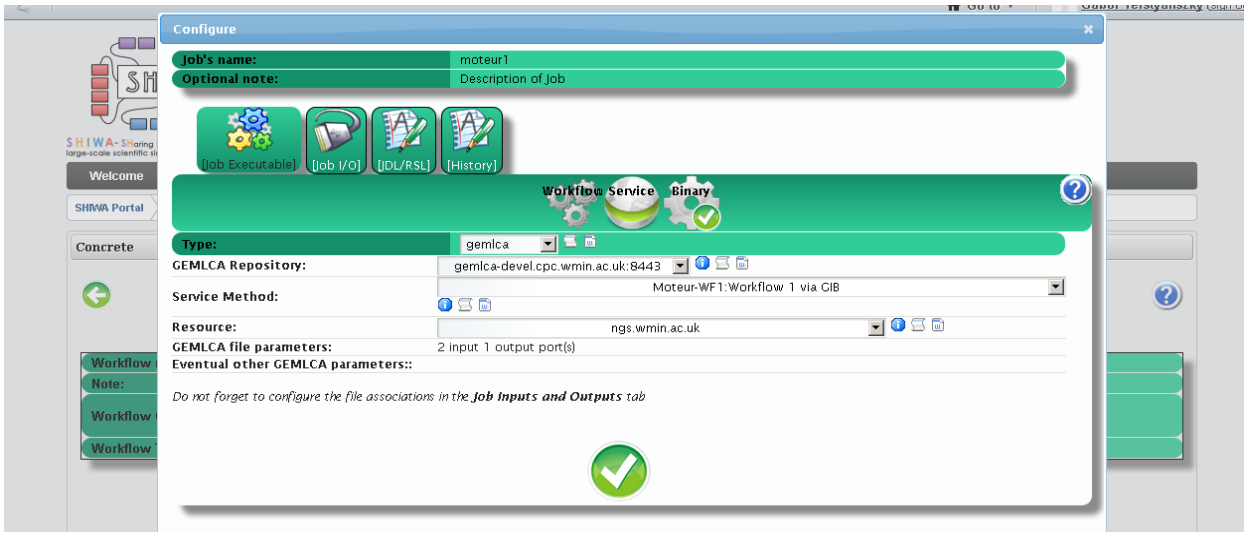

Figure 8.1: Configuring the Job Executable in the Old Portal

Solution: The SZTAKI team added a new feature to the portal and the UoW team upgraded the SHIWA Portal to gUSE v3.5.2. The new portal offers upgraded job executable and job port configuration services (See Fig. 8.2 and Fig. 8.3). The major improvement is that workflows previously selected in the SHIWA Repository are displayed at the top of the list of platform enabled workflows and users should not get through the whole list. For example at the top of the list is the previously selected Kepler workflow. (See Fig. 8.3)

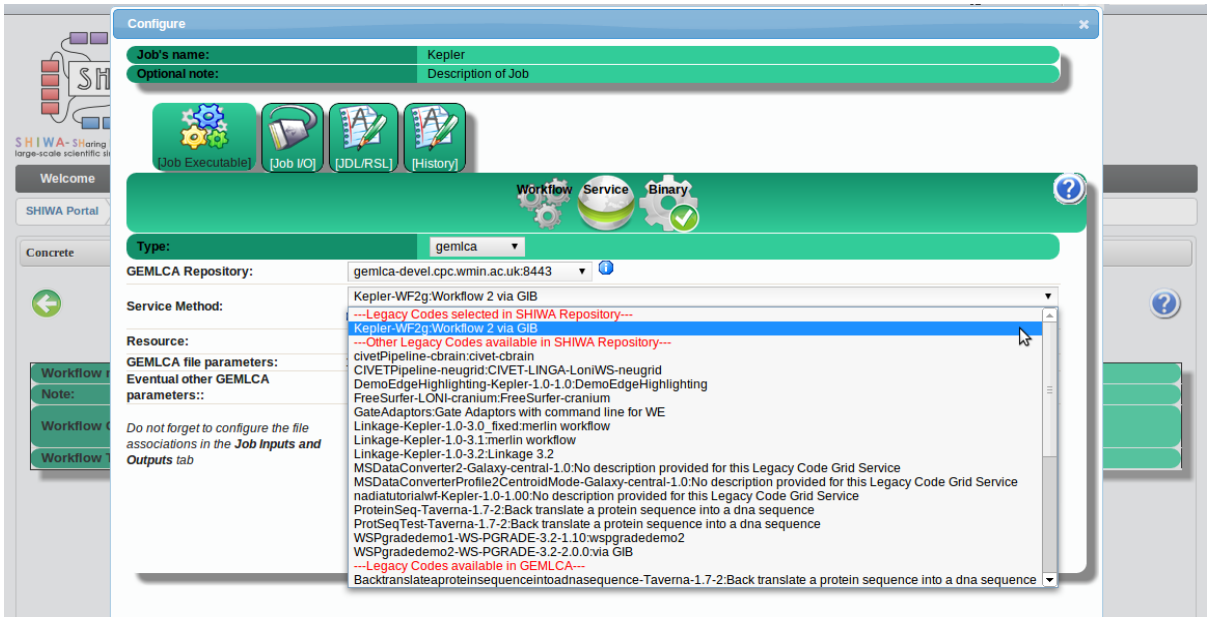

Figure 8.2: Configuring the Job Executable in the New Portal

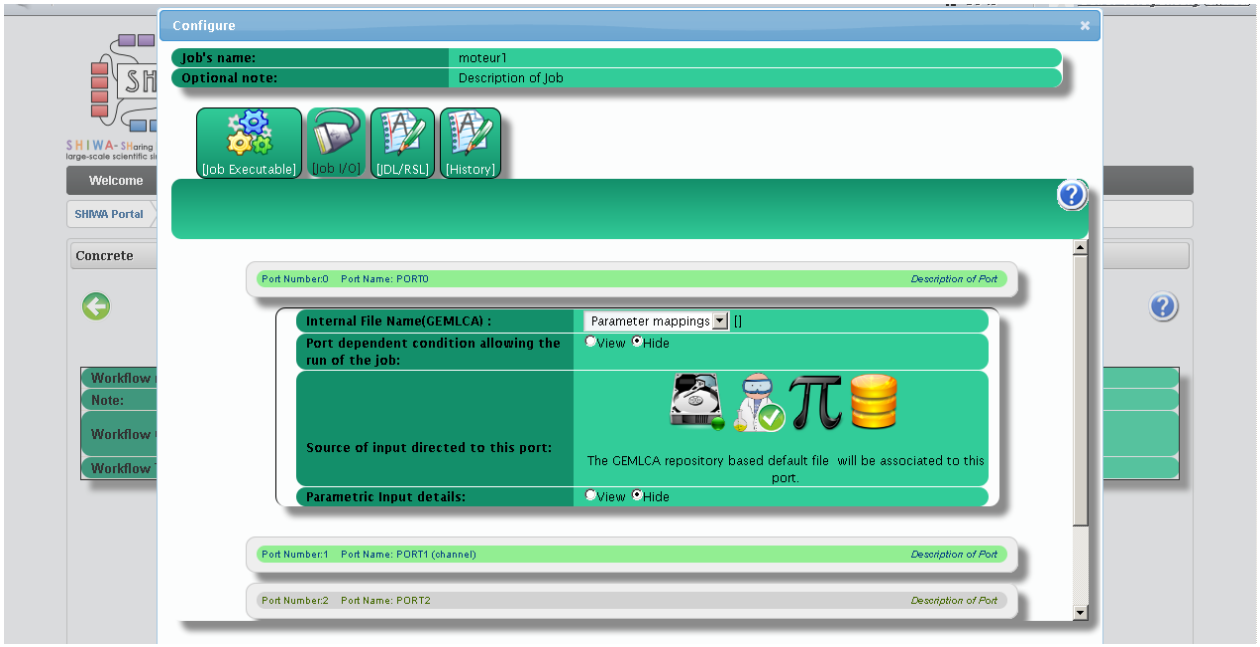

Figure 8.3: Configuring the Job Ports

The new SHIWA Portal also offers further advanced services such as accessing several cloud systems via the CloudBroker Platform that is integrated with gUSE since version 3.5.0. Via the CloudBroker Platform users of the ER-Flow community can create workflows that can run on the following cloud systems: Amazon, IBM, Eucalyptus, OpenNebula, OpenStack. This possibility opens a new horizon of executing workflows not only in grid systems but also in cloud systems. Since the SHIWA portal is based on gUSE further enhancements of the SHIWA portal can be seen in the gUSE sourceforge repository at: https://sourceforge.net/projects/guse/

8.1.2 Uploading workflows to the repository (A02)

Issue: Currently there are two options to upload a workflow into the SHIWA Repository: upload data manually or through the SHIWA Desktop. In the first case workflow developers should enter manually data and meta-data of abstract and concrete workflows through the repository interface. This process requires knowledge of the SHIWA Repository's data format and the workflow system. In the second case the workflow developer creates the workflow in the workflow system which he/she is familiar. The SHIWA Desktop generates the CGI bundle from the workflow and uploads the bundle to the SHIWA Repository which translates bundle data into the workflow representation of the repository. At this moment ASKALON, MOTEUR and Triana workflow systems are integrated with the SHIWA Desktop solution.

Solution: Since WS-PGRADE is used and will be used in the SHIWA Simulation Platform SZTAKI is developing the SHIWA Desktop solution for WS-PGRADE.

WS-PGRADE/gUSE system supports exporting WS-PGRADE applications to SHIWA repository in two formats. Both are based on SHIWA Desktop Bundle API that was developed to be able to describe a workflow signature. It has two versions: CGI Bundle and FGI Bundle. The CGI bundle stores only the input/output interface of a workflow and the workflow description. An FGI bundle contains all executables and input and expected output files associated to the nodes together with the workflow description. Both formats are independent from the language itself in which the workflow is stored therefore, we can place

the workflow into the bundle described either in native WS-PGRADE workflow language or in the common IWIR language. WS-PGRADE/gUSE 3.5.2 supports exporting in SHIWA Desktop FGI Bundle with IWIR language, which means that before creating the bundle, the native gUSE workflow must be translated to IWIR language. This approach has a limitation: due to the different expression power of the WS-PGRADE and IWIR languages, this translation restricts the set of the exportable applications.

Supporting ER-Flow community by allowing native gUSE export to SHIWA Repository from WS-PGRADE/gUSE system is going to be achieved in two steps. At the first step we took into consideration of the requirements of the gUSE community. For them it makes sense to allow to export workflow from WS-PGRADE/gUSE into FGI Bundle but with a workflow structure described in native WS-PGRADE language. This solution enables the community to share WS-PGRADE workflows as white-box applications and besides the structure won't be converted to IWIR enabling some features (e.g. Mixed dot and cross parameter field generalization for one job) supported by WS-PGRADE but not by IWIR. The solution is in testing phase.

The second step allows the whole ER-flow community (namely not just gUSE, but Moteur or Askalon users) to export WS-PGRADE bundles from WS-PGRADE/gUSE. It is based on the CGI format of the SHIWA Desktop Bundle API therefore, workflows in this format cannot be imported and edited as white-box applications, just as black-boxes. This solution is being under development.

8.1.3 Single sign-on to the SHIWA Simulation Platform (A03)

Issue: Users need accounts to access the SHIWA Simulation Platform, particularly to reach the SHIWA Portal and the private interface of the SHIWA Repository. The private interface permits access to non-validated and validated workflows. Users can upload, upgrade and delete workflows through this interface. The public interface allows access only to validated workflows. It enables only browse and search operations. Sysadmin assigns the portal and repository accounts with the same login name and password but it does not enable single sign-on. Users can access the repository either through the portal or the repository interface. According to the first option users have to login into the portal and they can access the public interface of the repository through the repository portlet. If they want to use the private interface they have to login in the repository. In the second option they can reach the private interface without login or they have to login through the repository interface to access the private interface.

Solution: There should be two user profiles: workflow developer and end user profile. The ER-flow project should create a profile-based sign-on operation. According to this solution workflow developers will be able to access the portal and repository without the current limitation after the login operation, i.e. clicking on the repository portlet the public interface of the repository will be available. At the same time there will be no change in the login operation for end users.

8.1.4 Supporting robot certificate on the SHIWA Simulation Platform (A04)

Issue: Workflow developers and workflow developers use X509 certificate on a regular basis because many of the workflow systems submit the service to grid resources. E-scientists may have certificates but it is not typical. Based on the previous experience on the one hand they do not want to get certificate, they do not want to manage it, i.e. to upload the certificate and download certificate proxys. On the other hand they want to use resources of different

DCIs with as little limitations as possible. They consider certificates as one of the constraints which prevent them from using grid infrastructures.

Solution: SZTAKI proposed to use robot certificate in the SHIWA Portal by extending the gUSE framework. The solution is based on the assumption that workflows are pre-defined and validated data processing applications which makes the usage of robot certificate compliant with the EGI regulations. In this solution the portal, the VO to which the portal provides access, the portal manager are all individually and collectively responsible and accountable for all interactions with the infrastructure. The portal administrator should manage robot certificates which should be hidden from end users. Currently SZTAKI is working on the robot certificate support. (See Fig. 8.4)

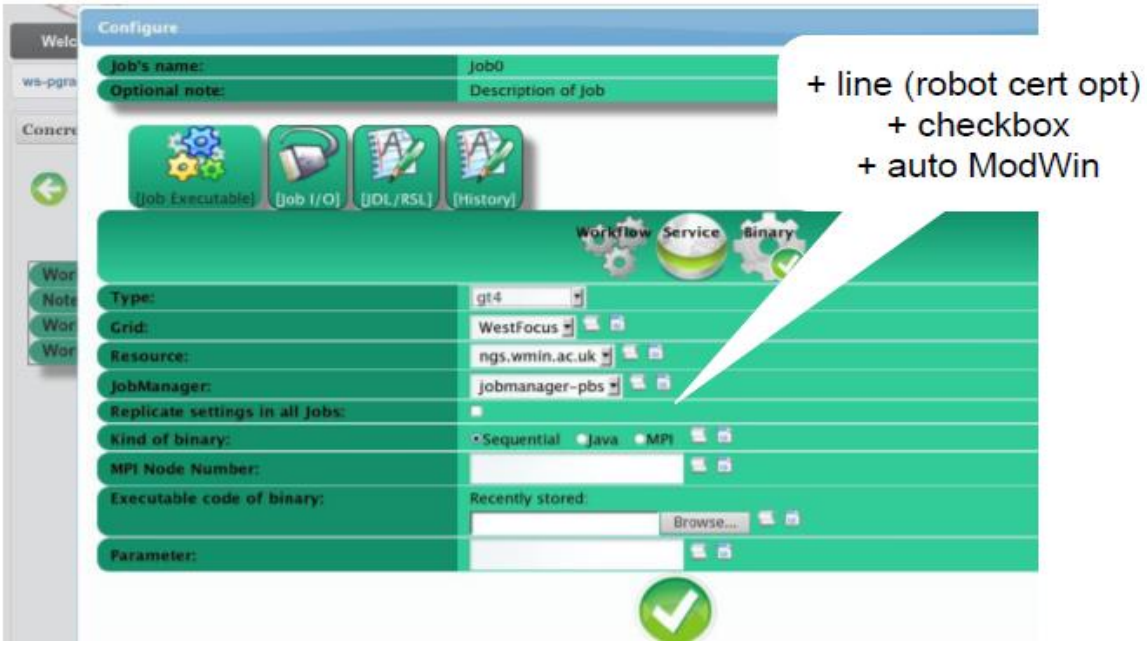

Figure 8.4: Managing robot certificate in the portal

8.1.5 Providing application oriented interfaces

Issue: In the SHIWA project most of the users were workflow developers who used the advanced user interface of the SHIWA Portal. This interface enables editing abstract workflows, creating concrete workflows, configuring, executing and monitoring them. According to the end users' feedback they want to run applications and not workflows. They do not even want to know anything about workflows. They expect that workflow developers create and test workflows and present them as a single entity.

Solution: The WS-PGRADE portal offers two options to provide application oriented interfaces: end user interface or ASM.

Figure 8.5: Configuring an application in the end user interface

In the first option users have to login as end users to have access to the end user interface. They can select an application, specify its input and output configuration (See 8.5) and submit it. The portal provides a monitoring page to be able to follow the progress of the application. (See Fig. 8.6)

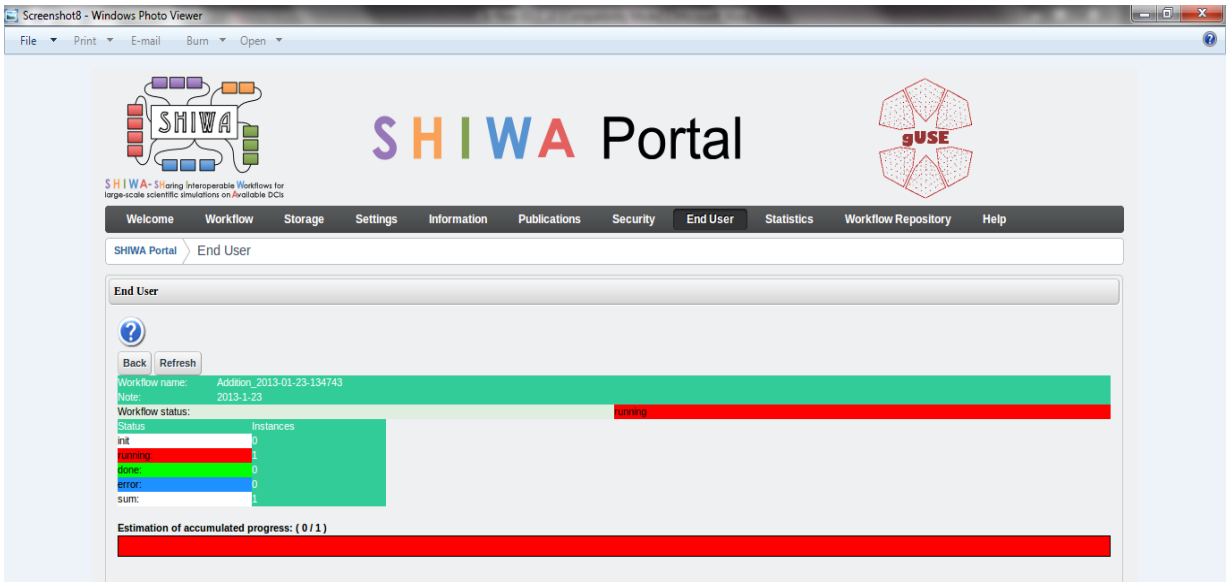

Figure 8.6: Monitoring the application in the end user interface

ASM is customised version of the end user interface. While the end user interface is part of the WS-PGRADE portal to have an ASM interface it requires development efforts. Fig 8.7 and Fig. 8.8 present an ASM based configuration and monitoring page, respectively.

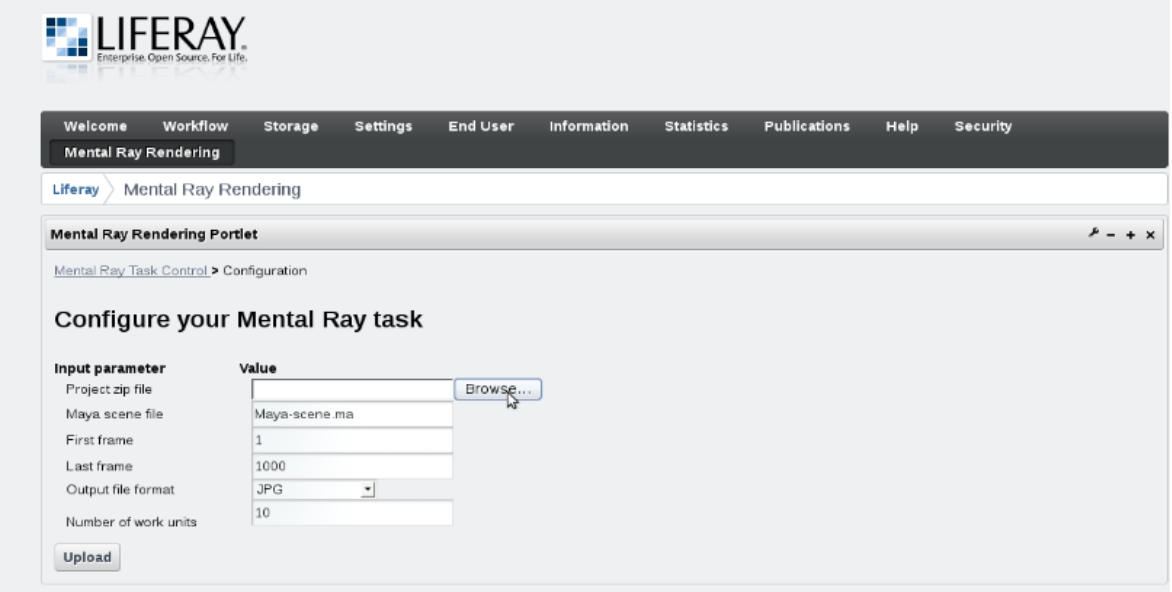

Figure 8.7: Configuring the application in ASM interface

Figure 8.8: Monitoring the application in ASM interface

WP3 and WP5 have to investigate all applications to be supported to define which one will be used through the ASM and which one through the end user interface.

8.1.6 Upgrading the SHIWA Submission Service (A06)

Issue: There are two repositories in the current simulation platform: GEMLCA and SHIWA Repository. The first one manages platform enabled workflows and workflow engines (data required for their execution and their executables) while the second one only workflow data. Currently the GEMLCA Repository is inside the GEMLCA Service (or SHIWA Submission Service). Having two repositories workflow and workflow system developers have to manage them through the SHIWA Portal's admin portlet and SHIWA Repository's user interface. The GEMLCA Service has further constraints. First, it is based on GT4 middleware which is not supported anymore properly and is perfomance bottleneck. Second, it has a too complicated architecture partly because of GT4 partly because of the GEMLCA Repository. Third, many of the backend plugins in the backend layer are out of date.

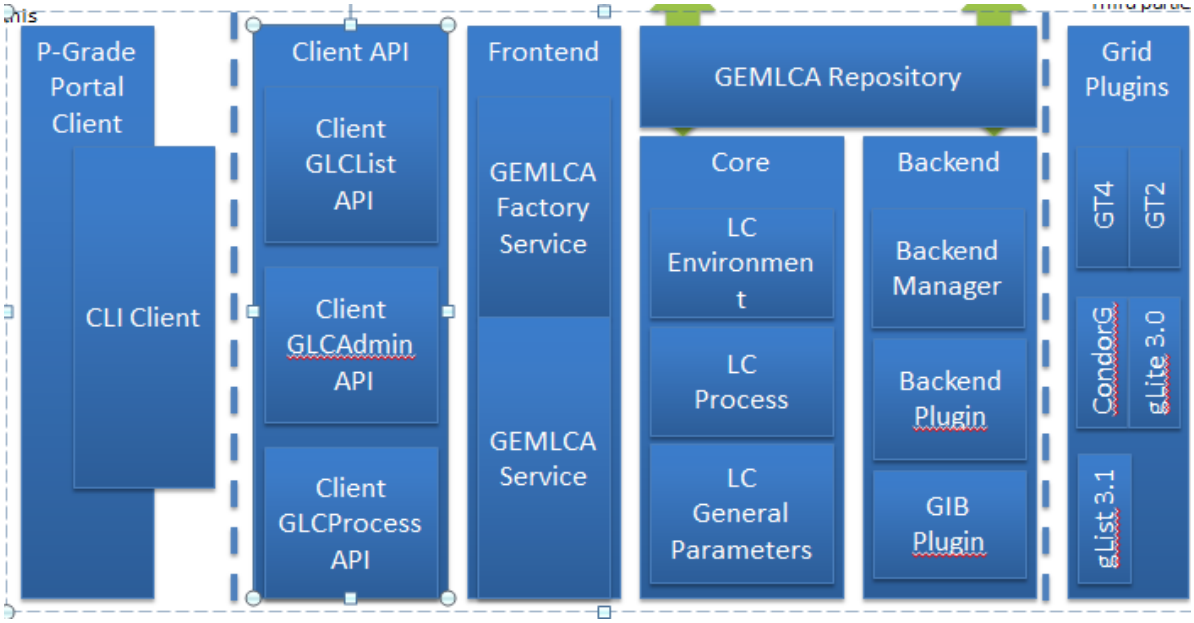

Figure 8.9: GEMLCA Architecture

Solution: We are proposing a minor re-engineering of the GEMLCA Service. The upgraded submission service should use the available portal and repository interfaces with which the users are familiar (See Fig 8.10). This service should use the client API of the current version to maintain compatibility with the SHIWA Portal. The GEMLCA Repository should be eliminated and the SHIWA Repository should manage both workflows and workflow engines. Finally, the upgraded submission service should access the infrastructure through the DCI Bridge instead of the backend plugins (See Fig. 8.11).

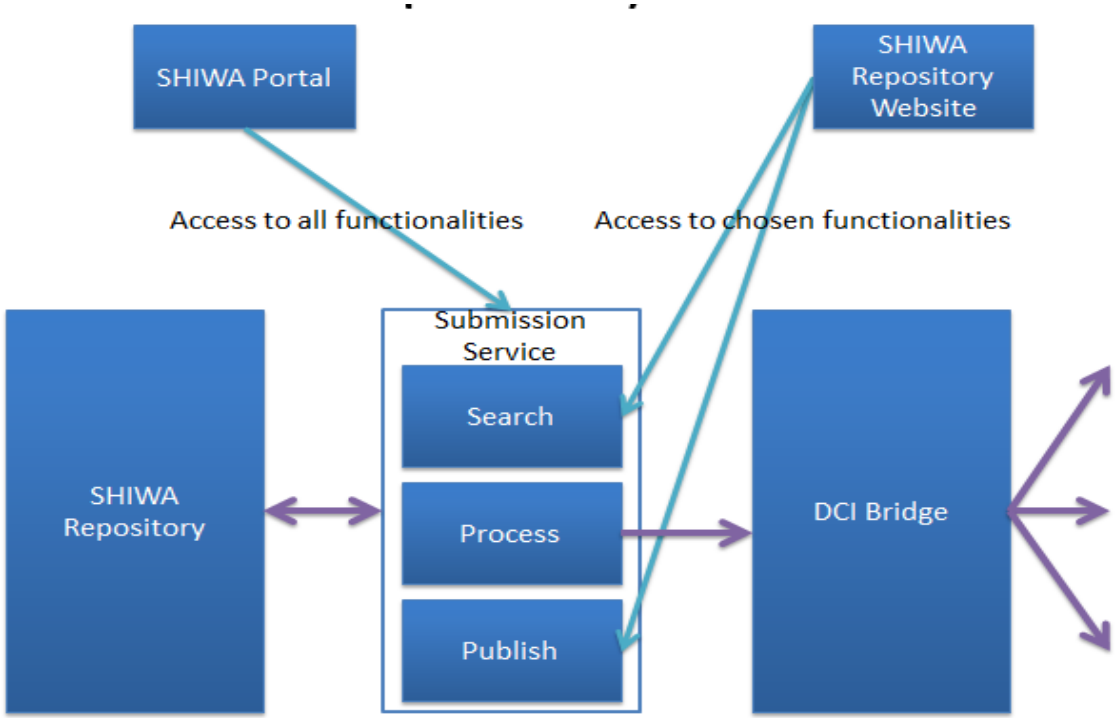

Figure 8.10: Upgraded SHIWA Architecture

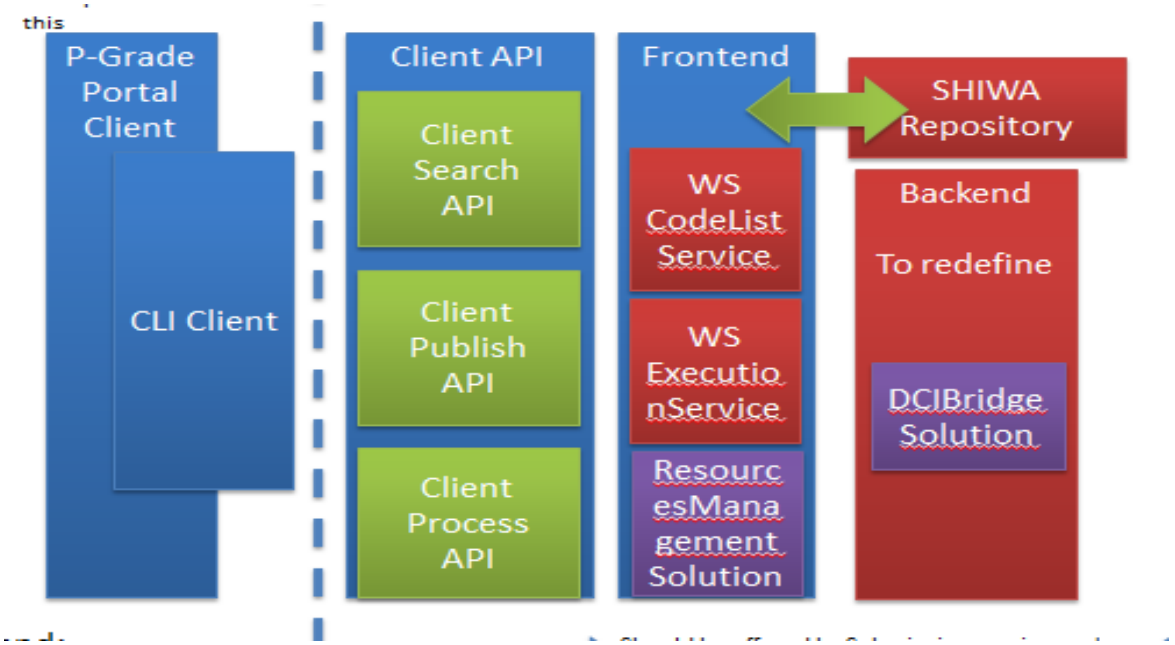

Figure 8.11: Re-engineered GEMLCA Architecture

8.2 Minor Enhancement Requests

The UoW team addressed issues I01 – I04. The current portal version provides manual sub-workflow re-submission through the “Suspend” and “Resume” function. This function enables users to suspend the meta-workflow and re-submit the failed sub-workflows. Since SZTAKI and UoW are working on the short-term major enhancement requests they will address this issue later.

9 Conclusion

The major goal of ER-flow is to support the user communities with a convenient workflow development and execution platform that enables the CGI-based interoperability of the used workflows. The SHIWA project has made the major technical developments to achieve this goal. The ER-flow project collects user feedbacks and requirements to further improve the workflow development and execution platform. This deliverable showed the collected user requirements and described how the ER-flow project intends to improve the workflow development and execution platform based on these user requirements.

References

[5.1] SHIWA: www.shiwa-workflow.eu

[6.1] M.Ellert et al.: Advanced Resource Connector middleware for lightweight computational Grids". Future Generation Computer Systems 23 (2007) 219-240.
[6.2] Foster, I. “Globus Toolkit Version 4: Software for Service-Oriented Systems.” Journal of Computer Science and Technology (Springer) 21, 2006, pp 513-520.
[6.3] gLite. http://glite.web.cern.ch/glite/.
[6.4] Erwin, D.W., and D.F. Snelling: “UNICORE-a Grid computing environment.” Concurrency and Computation: Practice and Experience (Springer) 14 (2002): 1395-1410.

[6.5] Thomas Fahringer, Radu Prodan, Rubing Duan, Jüurgen Hofer, Farrukh Nadeem, Francesco Nerieri, Stefan Podlipnig, Jun Qin, Mumtaz Siddiqui and Hong-Linh Truong, et al: ASKALON: A Development and Grid Computing Environment for Scientific Workflows, Workflows for e-Science, 2007, Part III, 450-471, DOI: 10.1007/978-1-84628-757-2_27
[6.6] Goecks, J.; Nekrutenko, A.; Taylor, J.; Galaxy Team, T: "Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences". *Genome Biology* **11** (8): R86. doi:10.1186/gb-2010-11-8-r86. PMC 2945788. PMID 20738864. http://www.pubmedcentral.nih.gov/articlerender.fcgi?tool=pmcentrez&artid=2945788.
[6.7] GWES - http://www.gridworkflow.org/kwfgrid/gwes-web
[6.8] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, Yang Zhao: Scientific workflow management and the Kepler system, Concurrency and Computation: Practice and Experience, Special Issue: Workflow in Grid Systems, Volume 18, Issue 10, pages 1039–1065, 25 August 2006
[6.9] T. Glatard, J. Montagnat, D. Lingrand, X. Pennec: "Flexible and efficient workflow deployement of data-intensive applications on grids with MOTEUR" in International Journal of High Performance Computing Applications (IJHPCA), 22(3):347-360, SAGE, August 2008.
[6.10] Z. Farkas and P. Kacsuk: P-grade portal: A generic workflow system to support user communities. Future Generation Computer Systems, 27(5):454 - 465, 2011.
[6.11] Ewa Deelman , Gurmeet Singh , Mei-hui Su , James Blythe , Yolanda Gil , Carl Kesselman , Gaurang Mehta , Karan Vahi , G. Bruce Berriman , John Good , Anastasia Laity , Joseph C. Jacob , Daniel S. Katz: Pegasus: a framework for mapping complex scientific workflows onto distributed systems, Scientific Programming Journal, 2005, Vol. 13, pp 219-237
[6.12] BR Odgers, JW Shepherdson, SG Thompson: Distributed workflow co-ordination by proactive software agents, In Proceedings of Intelligent Workflow and Process Management, IJCAI-99 Workshop, 1999
[6.13] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn: Taverna: a tool for building and running workflows of services. In Nucleic Acids Research, vol. 34, Web Server issue, pp. 729-732, 2006.
[6.14] I. Taylor, M. Shields, I. Wang, A. Harrison: The Triana Workflow Environment: Architecture and Applications. In Workflows for e-Science, pages 320-339. Springer, New York, Secaucus, NJ, USA, 2007.