Federated Computing on the Web: the UNICORE Portal

Maria Petrova-El Sayed*, Krzysztof Benedyczak†, Andrzej Rutkowski† and Bernd Schuller*
*Jülich Supercomputing Centre
Forschungszentrum Jülich GmbH, Germany
Email: {m.petrova, b.schuller}@fz-juelich.de
†Interdisciplinary Centre for Mathematical and Computational Modelling
Warsaw University, Poland
Email: golbi@icm.edu.pl, rudy@mat.umk.pl

***Abstract*—As modern science requires modern approaches, vast collaborations comprising federated resources on heterogeneous computing systems rise to meet the current scientific challenges. Due to their size and complexity these computing systems become demanding and could further complicate the scientific process. As a result, scientists are overrun with the necessity of additional technical experience that lies outside their domain.**

UNICORE is a middleware which serves as an abstraction layer to mask the technical details and ensure an easy and unified access to data and computation over federated infrastructures. The Portal is the newest client in the UNICORE portfolio providing web access to data and computing systems. With the rising demand of having an up-to-date user-friendly graphical interface and access to computational resources and data from any device at any point in time, the Portal meets the contemporary needs of the dynamic world wide web.

This paper describes the functionality of the Portal and its advantages over other clients. It also discusses security and authentication methods offered by the Portal and presents use cases and client customisation from our practice. It concludes with ideas about future work and extension points for further simplification for the scientific use.

I. Introduction

Complex scientific projects involve multiple computer simulations and calculations. In order to achieve their goals, scientists need to benefit from heterogeneous computing systems and resources, which can be distributed all over the world. The working process in such an environment is very challenging for scientists with limited technical background. UNICORE is a middleware software that ensures a smoother path for researchers, allowing them to concentrate on the actual scientific problem by reducing the amount of required technical knowledge. The development of UNICORE [1] dates back to 1996 and was initiated in an effort to provide access to the three largest German High-Performance Computing (HPC) centres. Its core design principles comprise: abstraction of resource-specific details, openness and extensibility, security, operating system independence and autonomy of resource providers. The software is available as open source from the SourceForge repository [2] under a commercially friendly licence.

The traditional end-user clients in the UNICORE portfolio are the UNICORE Commandline Client (UCC) and the desktop UNICORE Rich Client (URC). However, in the current era of technological advancement the world wide web is the main means to enable access to data. In an answer to contemporary demands, the UNICORE Portal provides a web-based Graphical User Interface (GUI). It offers not only a user-friendly GUI for work with distributed computing systems, but also serves as an access point to compute centres. Web technologies have multiple advantages, for instance, constant availability from an arbitrary location.

The remainder of this paper is structured as follows. Section II describes the functionality of the Portal. Supported authentication methods are covered in section III, which includes a concise presentation of the new Unity [3] component that plays a crucial role in many of the new development and integration scenarios. Section IV focuses on client customisations and extensions from our practice. In section V we discuss some advantages of the Portal by making a high-level comparison between the different UNICORE clients as well as by comparing the Portal to other existing web solutions. Finally, in section VI we summarise and conclude with future work.

II. Functionality of the UNICORE Portal

The UNICORE Portal facilitates computation in federated systems and makes the grid more accessible. It is also used as a convenient access point to a particular compute centre.

The functionality of the Portal covers generic use cases. Before being able to compute or manage data via the Portal, the user needs to authenticate himself at login. More information on available authentication methods is presented in section III. After successfully logging in, the user is directed to the home page of the Portal. The home page is to a great extent configurable by an administrator and can contain either a Really Simple Syndication (RSS) feed by choice, or a personalised design from a local Hypertext Markup Language (HTML) file. The Portal allows multiple customisations (done by an administrator only) like hiding certain views, branding the GUI, adding and exchanging logos and much more.

A. Job Submission

Focus points of the Portal functionality are execution of jobs, fetching of results and presentation of outcomes. By clicking on *New Job* from a flat navigation menu, the user is forwarded to a view for job submission (Fig. 1). Here he can select an application from a pre-filled list, which is being dynamically updated. The information about the applications in this list is fetched from the configuration files of the UNICORE sites which are accessible to the user.

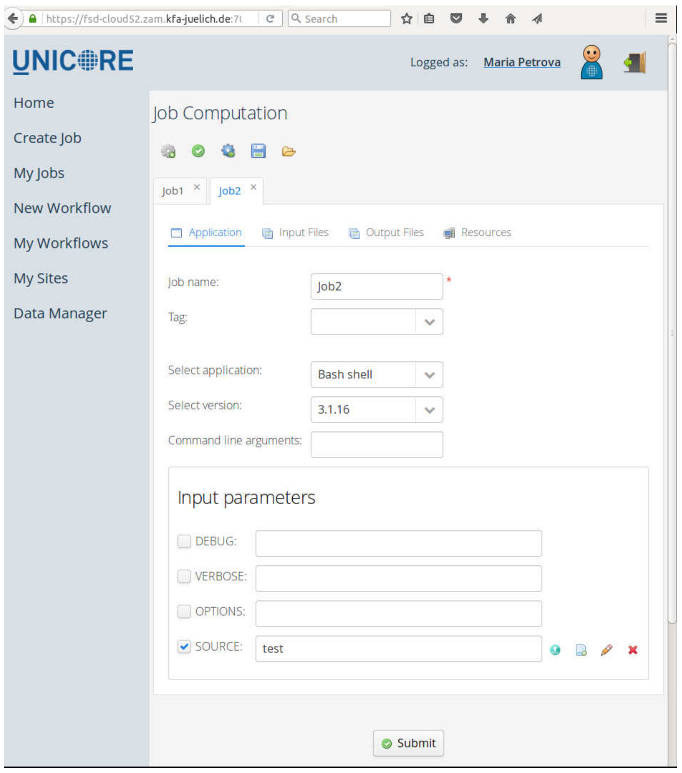
In this view, job descriptions and application specific input parameters can be entered in a graphical form. The latter would have normally been put into a script and manually submitted for execution. A GUI client eliminates the need to look into a detailed job script by automatically generating it in a ready to be submitted state. The user has to only prepare the necessary data and set the desired arguments. He can as well upload files from a remote or local storage. When clicking on the corresponding button, a window with the accessible storages will be open. Then the user can browse a selected storage and navigate through its folders in a manner resembling work within a file system. He also has the possibility to create new files where he can write necessary algorithms or other required input information. Most of the local or remote files can be changed on the fly through a file editor. However, restrictions on the size prohibit opening and editing of too large files from within the Portal.

A stage-out destination of objects to a desired output location can also be specified for every job. For example, the user can define that the job output be copied to a remotely located folder of choice.

Already in this view the user can see which sites he has access to and can choose where to send his job for execution. If none selected, the UNICORE server will take the decision based on a thorough brokering algorithm, which takes into account the needed resources, the load of the sites, etc. The required resources for the successful completion of his job can also be specified by the user in a comfortable GUI form. Furthermore, multiple jobs can be edited at the same time in multiple windows. Full job definitions can be saved and restored, thus allowing for parameter fitting and re-submission.

B. Monitoring

The monitoring of the job results is represented in a dedicated job table on a separate view. The table contains the job name, status, submission time, tags and other details. The update for each status change or a new job submission happens automatically. The user can sort, filter and search the table entries, as well as re-submit, abort or delete jobs. He can also browse the working directory. If the job's execution completed successfully, the working directory contains all job related files including the fetched output results. These can normally be viewed in the browser or downloaded to the local machine. As expected, the user can monitor and operate only with his own data and only after a successful authentication. In case of an unsuccessful job execution, the working directory stays empty as no output has been produced. However, the user can view the reason for failure by clicking on a button which opens a window with more detailed information.



Job name: Job2 *
Tag:
Select application: Bash shell
Select version: 3.1.16
Command line arguments:

Input parameters
- [ ] DEBUG:
- [ ] VERBOSE:
- [ ] OPTIONS:
- [✓] SOURCE: test

Fig. 1. The view for preparation and submission of generic jobs.

C. Data Manager

The Portal offers a powerful data manager which assists users in data-related tasks. The intuitive look & feel ensures a comprehensive GUI for transfer, upload or download of data within different locations. Users are able to create, preview and delete files in the browser as well as edit their content on the fly. The data manager supports both client-server upload/download and server-server transfers with several available data transport protocols. High throughput protocols such as UNICORE File Transfer Protocol (UFTP) [4] are also included.

D. User Workspace

Another typical characteristic is the so called *user workspace*. It enables the storage of data directly in the Portal in order for them to be accessible from anywhere on the web. For example, the data from submitted jobs are being preserved in the workspace. Each user has his own separate workspace, which is always created by default. However, for security reasons the Portal administrator has the possibility to hide the workspace from the user, thus restricting his access. If access is granted, the user has full control over the data from his workspace with the help of the data manager.

E. Workflows

The Portal supports the creation and submission of elementary workflows. With a few mouse clicks the user can compose a graph of jobs with the required connections, parameters, imports and exports. The workflow editor is meant to be simple and intuitive and serves only the basic needs, i.e. only task sequences. All the advanced features, such as conditional and loops, are currently omitted.

Identical to the job table, there is a separate table, containing all submitted workflows and their details. The user can browse through a workflow's working directory in a manner resembling browsing through a file system.

F. Sites view

The sites view contains information about all the accessible to the user sites. The user can track each site's availability, the number of its cores, its installed applications and more. If the geographic location is configured, it will be marked on Google Maps for a clear representation of the infrastructure. The sites view is just informative and has no influence on computing.

G. Configurations

The Portal is designed to be flexible and allows for multiple configurations. All settings including authentication modes, accessible compute centres, graphical customizations of logos, home page, main menu entries and more are defined in a properties file. This file is managed by a so called Portal administrator who is responsible for the initial set up of a Portal instance. The end user normally does not have access to the configuration file and does not need to know or manage the Portal instance itself. His working process starts from the authentication point.

H. Internationalisation

Last but not least, the Portal supports internationalisation. The default language is English. However, adding a new language to the GUI is uncomplicated and does not require any programming effort. All the messages are defined in properties files which can be translated. With a simple configuration from a Portal administrator, the new language will appear for selection together with all available languages at the login screen.

III. Security and Authentication

The certificates of the X.509 Public Key Infrastructure system are commonly used by grid middleware as a base of authentication and trust delegation, needed to coordinate distributed job execution. This approach turned out to be a major obstacle in grid adoption: certificates are difficult to obtain, distribute, refresh and manage. Conversion between different formats, installation in clients, etc. are problematic as well.

Web scenarios are even more troublesome. The Portal is a middleware client and is accessed itself by the user's browser. Therefore, the user's private key cannot be retrieved by the grid client—i.e. the Portal. Even if the user installs a certificate in the browser, it can only be used for the authentication step. As browsers do not support grid-specific trust delegation, which is needed by the grid, this has to be handled differently.

The UNICORE Portal approach to authentication and trust delegation addresses the aforementioned issues. It preserves backwards compatibility and does not scarify an overall security. Authentication in the Portal is designed in a modular way. Each authentication module provides support for a different authentication mechanism. In general these mechanisms can be grouped into two categories: *local mechanisms* and *remote mechanisms*.

The local authentication is handled fully by the Portal. Currently implemented local authentication modes comprise:

- An X.509 certificate—pre-installed in the user's web browser.
- User name and password—for simple installations as it lacks advanced features such as password reset.
- Demonstration account—a shared pseudo account that can be enabled for demonstrative access.

In order to use the UNICORE infrastructure after any of the local authentication mechanisms, the user is prompted to upload a so called trust delegation token to the Portal. The only exception is the demonstrative account. This is a solution to the delegation problem, as the private key stays at the user's machine. The drawback is that it requires the user to start a Java Web Start application (served by the Portal) and provide X.509 credentials to this application. The application generates a time-limited delegation token and uploads it to the Portal. This extra step is executed roughly once a month depending on the user's trust in the Portal. Nevertheless, it can be seen as an obstacle for portal users as it requires possession of a certificate and a potentially difficult action.

To offer first class authentication, the UNICORE Portal also supports a remote authentication over the Security Assertion Markup Language (SAML) protocol. The SAML authentication using the web POST profile is described in the OASIS SAML Standard [5]. Besides the authentication itself, support for an additional trust delegation assertion was added. Such assertion must be generated by a universally trusted third party service and returned to the Portal together with the SAML authentication assertion.

The role of a trusted delegation and authentication authority is fulfilled by a Unity service [3]. Unity is a powerful identity provider, featuring support for outsourcing authentication to different external identity providers, using both OAuth and SAML protocols. Through Unity, a middleware can be configured with an arbitrary login mechanism. Unity can enable one or more from the standard solutions such as password, authentication with data centre Lightweight Directory Access Protocol (LDAP), home Identity Provider (IdP) of a SAML federation, etc. It can even use a social identity provider like Google, Facebook or Github to name a few.

It should be underlined that the configuration of the selected authentication mechanisms is done by a Portal administrator and is transparent for the end user. Thus the choice of allowed authentication sources is fully controllable and the Portal's authentication uses UNICORE trust delegation without any tricks like generation of short lived certificates or storing user's private key in a web server.

IV. Use cases and user customisation

A. Generic use case

Already in its default configuration, the UNICORE Portal can be used for a wide variety of use cases. Access to a federated infrastructure of HPC resources is one of the primary design goals.

As a concrete example for this use case, we would like to mention the Human Brain Project (HBP) [6], a European FET flagship project. The HBP targets a wide range of topics, aiming at a deeper understanding of the human brain and attempting to leverage this understanding for new technological advancements. The HBP operates on HPC resources, integrating major supercomputing sites in Jülich, Barcelona, Bologna and Lugano as well as cloud storage and other resources. The HBP uses a single-sign-on system based on OpenID Connect (OIDC) [7]. With UNICORE deployed as underlying middleware, the UNICORE Portal is used as a simple way to access the HPC and data resources. The Portal is configured to use Unity for authentication, and is thus integrated with the single-sign-on functionality. The user benefits already in this simple scenario from the abstraction provided by UNICORE and the single-sign-on functionality; it is no longer required to know and understand how to log into the various resources, or be aware of the various batch schedulers that are used at the various sites (e.g., Slurm and IBM LoadLeveler). The Portal is used to prepare and submit so called *generic jobs*. The user interface for these generic jobs is generated at runtime based on metadata provided by the UNICORE server.

B. Custom plugins

The generic approach is sufficient for the majority of applications, especially those whose users were traditionally preparing input files in some well-established format. However, there are cases when a more sophisticated interface is welcomed. For example, some applications require special resource allocations such as CPU, number of nodes, used memory or architectures, which might all depend on other application parameters. A specialised application interface can assist with these complex resource settings. Customisations to the interface can also improve the preparation of input. For instance, selecting parts of the image for a visual analysis is much more convenient when clicking on the image itself than by providing numeric pixel coordinates. Even though the generic application description is quite powerful and allows for expressing the most common dependencies, some complex relationships between input settings may require special implementation. Furthermore, for many applications output visualisation is a key feature and its proper placement in the Portal is crucial for effective research.

As the above aspects were known from the very beginning, the core Portal is designed as a base for development of applications or domain specific plugins. A plugin is free to implement an arbitrary user interface as well as to contribute with one or more entries to the main menu. The administrator has control over all menu entries and can select which of them to be shown on the GUI. This allows for tailoring of the Portal to particular domain needs even to a degree where a generic job interface is not presented at all.

The development of a portal plugin is restricted to the base technology of the Portal, which includes Vaadin [8] as the underlying framework for building user interfaces in Java and Spring [9] as a container for objects. These restrictions are counterbalanced by the possibility to use internal portal components and features to speed up the plugin development. For example, multiple graphical elements such as the job table can be re-used from the existing implementation. Already present are also the various user authentication methods as well as a grid security context for grid interaction like asynchronous discovery of resources and jobs. Thus, the development can be focused purely on application domain aspects.

We present here two examples of such custom modules developed in PL-Grid Plus and PL-Grid NG projects. The first one is named *SinusMed*. *SinusMed* processes an input, that being a series of computed tomography (CT) images, which form a 3D image of a human head. It detects all air-filled head areas. Areas are subsequently marked and categorised using reference data sets prepared by medical doctors. Results of the simulation comprise several sets of layered masks on the input image, which must be presented in a visual form.

Another example is the *AngioMerge* application. It is used to synchronise several angiograms, which are taken periodically with an interval of several minutes. The output is supposed to be displayed in 3D, optionally as an animation.

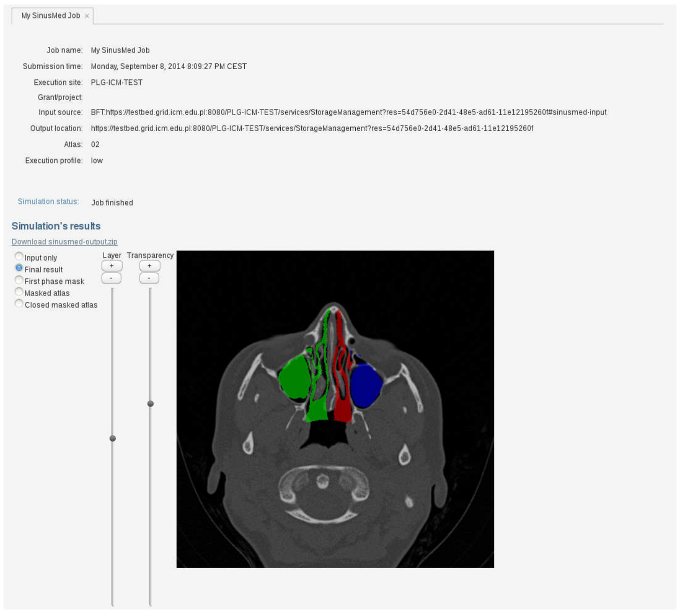
The portal modules for *SinusMed* and *AngioMerge* allow for a simplified job preparation, which focuses on the actual application input and automatically pre-sets the required resources for each application. Therefore, the user is not burdened with additional knowledge about the computing infrastructure being used. What is more, applications submit the jobs with the help of a UNICORE broker so that site selection is fully automated.

The most attractive features of the above application specific interfaces are found in the output visualisation part. The *SinusMed* directly shows the original CT images in the browser (Fig. 2), enabling the overlay of synchronised masking layers, computed by the application. The *AngioMerge* module embeds a WebGL 3D interface (Fig. 3) allowing for interactive viewing of the output and even playing the resulting animation.

V. Comparison with other solutions

A. With Other UNICORE Clients

The UCC is a commandline tool that enables job and workflow management from a shell or scripting environment [10]. It covers all features of the UNICORE service layer including data transfers, job and workflow submission and monitoring, results fetching, etc. Among its characteristic qualities is the possibility to submit multiple jobs in an automated fashion via a batch mode. To speed up the work, new customised commands can be defined by the user. However, the UCC is



My SinusMed Job ×

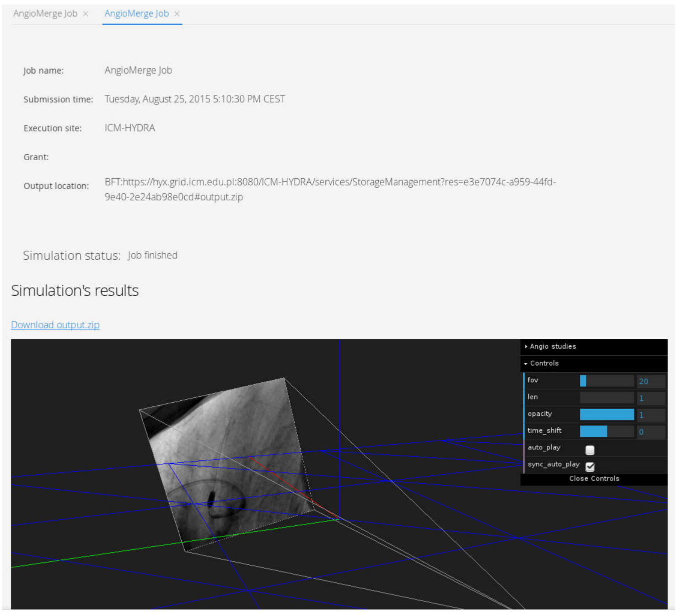
Job name:	My SinusMed Job
Submission time:	Monday, September 8, 2014 8:09:27 PM CEST
Execution site:	PLG-ICM-TEST
Grant/project:	
Input source:	BFT:https://testbed.grid.icm.edu.pl:8080/PLG-ICM-TEST/services/StorageManagement?res=54d756e0-2d41-48e5-ad61-11e12195260f#sinusmed-input
Output location:	https://testbed.grid.icm.edu.pl:8080/PLG-ICM-TEST/services/StorageManagement?res=54d756e0-2d41-48e5-ad61-11e12195260f
Atlas:	02
Execution profile:	low

Simulation status: Job finished

Simulation's results

Download sinusmed-output.zip

Fig. 2. Results of sinuses detection embedded in the Portal interface after a simulation in a low quality (fast) mode.



AngioMerge Job × AngioMerge Job ×

Job name:	AngioMerge Job
Submission time:	Tuesday, August 25, 2015 5:10:30 PM CEST
Execution site:	ICM-HYDRA
Grant:	
Output location:	BFT:https://hyx.grid.icm.edu.pl:8080/ICM-HYDRA/services/StorageManagement?res=e3e7074c-a959-44fd-9e40-2e24ab98e0cd#output.zip

Simulation status: Job finished

Simulation's results

Download output.zip

Fig. 3. Output visualisation of the *AngioMerge* module directly in the portal web interface.

oriented towards experienced computer users, who find working with a terminal faster and more convenient. In contrast, there are scientists who welcome an easy to use GUI.

The URC [11] is a standalone client that provides a detailed graphical interface for the UNICORE functionality. Similar to UCC, the URC supports all features like job and workflow creation and submission, retrieval of results, data transfers and grid browsing. In its essence the software is based on Eclipse [12], which makes it easily extensible via the plugin mechanism. Specific to the URC are the different perspectives, which hide or show features from the UI in accordance to how advanced the user is. The major strength of this client is its powerful workflow editor with constructs like loops and conditional statements for steering the control flow and enabling a fully automated work process.

In comparison to the URC, the functionality of the Portal's workflow editor is elementary. We have taken this approach due to feedback from our users. In order to use the advanced structures, many users still need the assistance of a UNICORE specialist. This is one of the main reasons why we decided to keep the design of the Portal's workflow editor simple and light-weight. Thus, we ensure a comprehensive and intuitive interface that covers the basic need, whereas the complex use cases can benefit from the well-developed and well-established editor of the URC.

Federated computing on the web has multiple advantages to desktop clients. Unlike the UCC and the URC, the Portal is available from any location and can be accessed by various devices. While standalone clients require installation, administration and regular updates done by the user himself, a web portal releases the user of this responsibility, thus saving time and effort.

B. *With Other Existing Web Clients*

There are several existing web portals that concentrate on HPC. One such solution is the InSilicoLab [13]. It is an environment built to assist researchers in their specific domain. In principle it has been designed to be generic, but it is currently used only by projects in chemistry [14]. Similar to the UNICORE Portal, this tool offers access to distributed resources on the web, job preparation, submission and monitoring as well as fetching of results and data management. However, the InSilicoLab has rather specialised into domain-specific interfaces. In contrast, the Portal is more generic with the intention to be an easily extensible solution that can be beneficial in various research domains. The Portal is also more flexible with its numerous authentication possibilities through the integration with Unity.

Another web client for HPC is Adaptive Computing's Moab Viewpoint [15]. The Moab Viewpoint is solely concentrated on job and data management and does not cover any workflow features. However, it offers an interface for creating application templates, which helps in optimizing application run times and reduces errors. Another enticing feature, which is not present in the UNICORE Portal, is the existence of two GUI modes: an end user and an administrator one. The administrator mode enables supplementary functionality like reporting on resource utilisation, troubleshooting of the workload, editing or cancellation of jobs and tracking of node-usage. The Portal can be customized by an administrator through a properties file but this should not be confused with the administrator mode from Viewpoint. The administration in the Portal is not job based and offers no GUI. It concerns the enabled options like languages, authentication methods, menu entries, etc. for a particular Portal installation. One drawback of the Moab Viewpoint is that it is a commercial software, which means extra costs incurring for the users. Furthermore, Viewpoint does not operate in a distributed environment, but only supports work on a singular site.

VI. Conclusion

The UNICORE Portal is an open source solution that facilitates scientific work by enabling federated computing on the web. It offers a convenient and friendly GUI for job and workflow submission, monitoring and data management. The authentication to the Portal is done either locally or with the help of Unity, which allows for using numerous authentication mechanisms, including social and federated ones, without the need to possess an X.509 certificate.

The default Portal implementation covers the generic use case, which is sufficient for most projects. We presented as an example the integration of the Portal in the HBP platform. The Portal is also flexible and, due to its modular design, it is easily extensible by self-developed plugins for specific applications. The creation of a new plugin requires solely the effort that is necessary to add the particularities of the targeted domain, whereas the base portal components and features can be easily re-used. We described two use cases with custom modules, namely the *SinusMed* and the *AngioMerge* applications.

We also made a concise comparison between the Portal and the other clients in the UNICORE portfolio with their strong points and disadvantages. We discussed related web solutions for HPC environments.

The Portal's development does not stop here. It faces multiple milestones on its future path. One of them, which is under current development, is the workflow template feature. The Portal will offer the option to import a ready template file. The latter will be parsed and a familiar graphical form will be automatically generated. Then the user will need to fill the data and submit the updated workflow without having to compose a new one from scratch. This approach will ensure a comfortable parameter fitting, simple re-use of workflows and uncomplicated workflow submission.

Another idea that is also in progress is the integration with data sharing solutions. Scientists will be able to share with others their files, such as workflow templates, from within the Portal's interface.

Acknowledgement

This work was made possible with assistance of the PL-Grid Plus project, contract number: POIG.02.03.00-00-096/10, and the PL-Grid NG project POIG.02.03.00-12-138/13, website: www.plgrid.pl. The projects are co-funded by the European Regional Development Fund as part of the Innovative Economy program.

The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 604102 (Human Brain Project).

References

[1] A. Streit, P. Bala, A. Beck-Ratzka, K. Benedyczak, S. Bergmann, R. Breu, J. M. Daivandy, B. Demuth, A. Eifer, A. Giesler, B. Hagemeier, S. Holl, V. Huber, N. Lamla, D. Mallmann, A. S. Memon, M. S. Memon, M. Rambadt, M. Riedel, M. Romberg, B. Schuller, T. Schlauch, A. Schreiber, T. Soddemann, and W. Ziegler, "UNICORE 6 – recent and future advancements," *Annals of telecommunications-annales des télécommunications*, vol. 65, no. 11-12, pp. 757–762, 2010.

[2] "UNICORE Open Source project page," [accessed: 2016-02-01]. [Online]. Available: http://sourceforge.net/projects/unicore/

[3] "Unity project website," February 2016, [accessed: 2016-02-01]. [Online]. Available: http://www.unity-idm.eu

[4] B. Schuller and T. Pohlmann, "UFTP: High-performance data transfer for UNICORE," in *Proceedings of 7th UNICORE Summit 2011*, ser. IAS Series, no. 9. Forschungszentrum Jülich GmbH, 2011, pp. 135–142.

[5] J. Hughes, S. Cantor, J. Hodges, F. Hirsch, P. Mishra, R. Philpott, and E. Maler, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 OASIS Standard," 3 2005, [accessed: 2016-02-01]. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/

[6] "Human Brain Project," [accessed: 2016-02-01]. [Online]. Available: http://www.humanbrainproject.eu/

[7] "OpenID Connect," [accessed: 2016-02-01]. [Online]. Available: http://openid.net/connect

[8] "Vaadin framework website," [accessed: 2016-02-01]. [Online]. Available: https://vaadin.com/

[9] R. Johnson, J. Hoeller, K. Donald, C. Sampaleanu, R. Harrop, T. Risberg, A. Arendsen, D. Davison, D. Kopylenko, M. Pollack *et al.*, "The spring framework–reference documentation," *Interface*, vol. 21, 2004.

[10] "UNICORE commandline client: User Manual," [accessed: 2016-02-01]. [Online]. Available: http://unicore.eu/documentation/manuals/unicore/files/ucc/ucc-manual.html

[11] B. Demuth, B. Schuller, S. Holl, J. Daivandy, A. Giesler, V. Huber, and S. Sild, "The UNICORE Rich Client: Facilitating the automated execution of scientific workflows," in *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, Dec 2010, pp. 238–245.

[12] "The Eclipse Foundation open source community website," [accessed: 2016-02-01]. [Online]. Available: https://eclipse.org/

[13] J. Kocot, T. Szepieniec, D. Harężlak, K. Noga, and M. Sterzel, "Insilicolab – managing complexity of chemistry computations," in *Building a National Distributed e-Infrastructure–PL-Grid*. Springer, 2012, pp. 265–275.

[14] A. Eilmes, M. Sterzel, T. Szepieniec, J. Kocot, K. Noga, and M. Golik, "Comprehensive support for chemistry computations in PL-Grid infrastructure," in *eScience on Distributed Computing Infrastructure*. Springer, 2014, pp. 250–262.

[15] "Adaptive computing: Viewpoint portal," [accessed: 2016-02-01]. [Online]. Available: http://www.adaptivecomputing.com/products/hpc-products/viewpoint/