# GridSpace Based Virtual Laboratory for PL-Grid Users

Maciej Malawski, Eryk Ciepiela, Tomasz Gubała, Piotr Nowakowski, Daniel Harężlak, Marek Kasztelnik, Joanna Kocot, Tomasz Bartyński i Marian Bubak

ACC Cyfronet AGH, ul. Nawojki 11, 30-950 Kraków, Poland

e-mail: malawski@agh.edu.pl

As the shift from traditional science to *e-science* may impose significant learning curve for scientists the tools accompanying PL-Grid are expected to conceal complexity and low-level issues of underlying technologies and provide a workbench called Virtual Laboratory tailored to users' demands and suited to their style of work.

The experience gathered through development of the **ViroLab** Virtual Laboratory for virologists [VL, VL-ROYAL, VL-WWW], the APPEA runtime environment for banking and media application in **GREDIA** project [APPEA, GREDIA-WWW], the **GridSpace** environment [GS-WWW] for running *in-silico* experiments, has been extended with further deepened user requirements analysis during the initial phase of PL-Grid project. The requirements are continuously collected through inquiries and discussions with interested groups of scientists of various domains such as physics, chemistry and engineering.

Basing on analysis carried out by the end of 2009, bearing in mind the aspects imposed by emerging PL-Grid infrastructure the initial vision of PL-Grid Virtual Laboratory has been drawn up, and its implementation has been launched, yet it's still a subject for further refinements and revisions as the Virtual Laboratory assumes iterative and incremental development, so the current version of the document reflects the design of the Virtual laboratory at the most recent development phase. We foresee that new requirements will emerge and new design decisions are to be taken based on the results of the development phase.

The vision of the Virtual Laboratory developed in the PL-Grid project is to provide tools which facilitate programming and execution of complex applications on the PL-Grid infrastructure. Such applications are referred to as experiments, which means that they go beyond simple and repeatable execution of installed programs. Instead, they may involve experimentation or exploratory programming, where the application consists of multiple steps which are not known in advance and often are selected ad-hoc, based on the results of previous steps. Another important aspect is support for collaborative work between users, so that they can share and reuse the pieces of their applications, such as scripts, code snippets and whole experiments.

In order to meet these objectives the Virtual Laboratory will follow a layered and distributed architecture as shown in Figure 1. The topmost layer is formed by a web portal which constitutes an entry point for the whole Virtual Laboratory. This gives users access to the Virtual Laboratory from any workstation equipped with a web browser. Such an **Experiment Workbench Layer** provides a Portal which is intended as a common tool-rich workbench for all Virtual Laboratory researchers where they perform their daily experimentation, collaborate, communicate and share resources (such as reusable code). Below, there is an **Experiment Execution Layer** where consecutive parts of experiments provided by the users through the Portal are actually evaluated in the context of a particular user account on the so-called experiment host machine. The key concept is that an experiment is developed continuously and in parts. Individual experiment parts are executed by an experiment interpreter which preserves state (namespace, runtime values) between evaluations. Consequently, experiment development time and runtime overlap, and the activities of writing and executing code are intertwined. Such an approach admits introspective, interactive, explorative and dynamic experimentation recorded as experiment code. The Experiment Workbench Layer and Experiment Execution Layer are in constant communication via the SSH protocol family. Below, there is a layer that consists of Gems which are libraries/modules/utilities invoked by experiments at runtime. The **Gem Layer** provides APIs for experiment developers enabling programmatic access to underlying resources (Grid, clusters) and functionality exposed in the form of libraries or services.

At the very bottom of the architecture the **Grid Fabric Layer** is situated. It consists of Grid resources available to the PL-Grid community, including clusters accessible through PBS, Grids available through middleware, external services (e.g. Web Services) and data sources (e.g. RDBMSs) that users may make use of in the course of their research activities.

As the result, the PL-Grid Virtual Laboratory will provide scientists with web-based workbench supporting exploratory, multi-language (we plan to support Python, Ruby, Perl), interactive and

console-like programming. Such an environment will enable publishing and reusing of utilities as libraries called gems, code snippets and whole programs called experiments. Moreover, the basic gems enabling access to computational infrastructure will be provided out-of-the-box. It will be enabled to publish experiment as web application in order to allow broader audience to benefit from them. Also the obtained result data will be a subject to store, publish and share within Virtual Laboratory.

As opposed to existing virtual laboratories workbenches and workflow systems the PL-Grid Virtual Laboratory will support exploratory experimentation programming paradigm and multifaceted collaboration by sharing software, data and ready-to-use web applications.
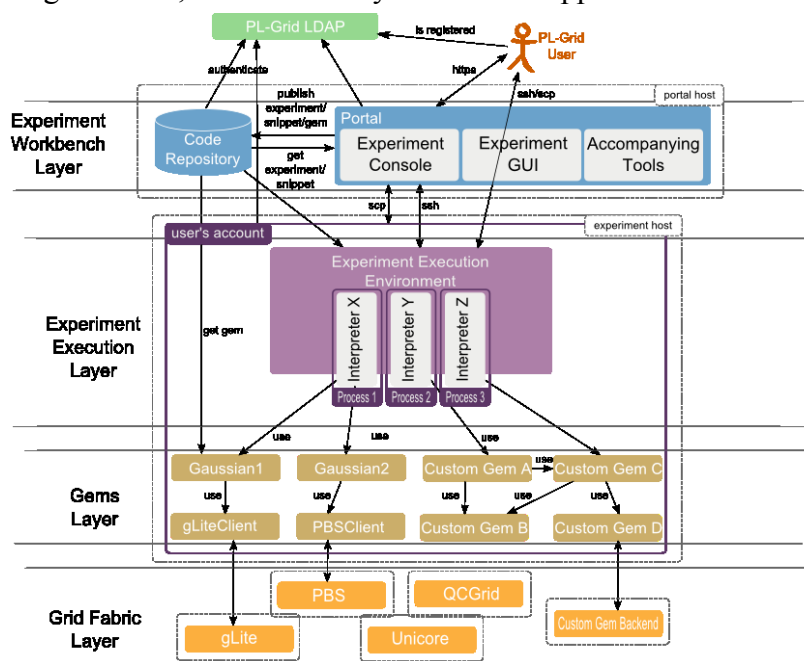


**Figure 1: Architecture of the PL-Grid Virtual Laboratory including logical layers (captions on the left-hand side), physical deployment (dotted boxes), modules (solid boxes) and their interconnections.**

**References**

[VL] M. Bubak, M. Malawski, T. Gubala, M. Kasztelnik, P. Nowakowski, D. Harezlak, T. Bartynski, J. Kocot, E. Ciepiela, W. Funika, D. Krol, B. Balis, M. Assel, and A. Tirado Ramos. Virtual laboratory for collaborative applications. In M. Cannataro, editor, *Handbook of Research on Computational GridTechnologies for Life Sciences, Biomedicine and Healthcare*, chapter XXVII, pages 531-551. IGI Global, 2009.

[VL-WWW] ViroLab Virtual Laboratory: http://virolab.cyfronet.pl

[APPEA] D. Harezlak, P. Nowakowski, M. Bubak, "Appea: A Framework for Design and Implementation of Business Applications on the Grid", Proceedings of Computational Science - ICCS 2008 , 8th International Conference Krakow, Poland, June 2008, volume III, LNCS 5103, Springer, 2008

[GREDIA-WWW] http://www.gredia.eu

[GS-WWW] GridSpace http://gs.cyfronet.pl

[VL-Royal] P.M.A Sloot, Peter V. Coveney, G. Ertayalan, V. Mueller, C.A. Boucher, and M. Bubak: HIV decision Support: from Molecule to Man. Philosophical Transactions of the Royal Society A, vol 367, pp 2691 -

2703, 2009, doi:10.1098/rsta.2009.0043.