Open Marketplace for Simulation Software on the Basis of a Web Platform

A P Kryukov$^1$ and A P Demichev$^2$

Skobeltsyn Institute of Nuclear Physics, Lomonosov Moscow State University, Leninskie gory, 1, Moscow, 119991, Russia

E-mail: $^1$kryukov@theory.sinp.msu.ru, $^2$demichev@theory.sinp.msu.ru

Abstract. The focus in development of a new generation of middleware shifts from the global grid systems to building convenient and efficient web platforms for remote access to individual computing resources. Further line of their development, suggested in this work, is related not only with the quantitative increase in their number and with the expansion of scientific, engineering, and manufacturing areas in which they are used, but also with improved technology for remote deployment of application software on the resources interacting with the web platforms. Currently, the services for providers of application software in the context of scientific-oriented web platforms is not developed enough. The proposed in this work new web platforms of application software market should have all the features of the existing web platforms for submissions of jobs to remote resources plus the provision of specific web services for interaction on market principles between the providers and consumers of application packages. The suggested approach will be approved on the example of simulation applications in the field of nonlinear optics.

1. Introduction
Currently the development of scientific web applications is booming around the world. In particular, today scientific services in the Internet provide the access to various data (for example, http://www.nature.com/sdata), scientific experimental facilities (for example, High Speed Networking with Subaru Telescope in Hawaii, http://www.naoj.org/), and educational materials (Virtual Learning Environment [1]). In many cases researchers need to run a large number of similar computing tasks to solve some problem in a particular application area. Quite often, these tasks are performed by using already existing specialized application packages, the tasks being identical in form and differing only in the values of the input data (parameters). In this case, pre-installation and configuration of commonly used application packages on a computing resource and the availability of specialized web services as well as appropriate user web interfaces allow owners of computing resources to increase efficiency of their use (Software as a Service model; SaaS). The set of specialized interconnected web services and web application interfaces is called a web platform for remote access to computing resources (see, e.g., a short survey in [2]). Development of web platforms for scientific research in different areas are widely underway around the world. Examples of such developments: nanoHUB (http://www.nanohub.org); Yabi (https://ccg.murdoch.edu.au/yabi); e-Science Central (http://www.esciencecentral.co.uk); other examples and discussions see in [2].
Each individual pre-arranged task (a launch of an application package, an access to a data storage, and so on) is called “a tool”. With the access to the tools provided by a platform, a user only has to specify particular values of the input parameters or input files, and the rest of the task description is generated automatically. Thus the user can carry out the generating of the task, its submission, monitoring, and getting the results via a standard web browser. This approach proves to be particularly effective and convenient when the full research requires a set of application packages rather than just one of them. Moreover, often the output of some package is the basis for the generating of the task for the next application package in a series of stages of the research. Such sets of tasks are called workflows. Besides the launch of computing packages, workflow may include other operations such as a query to relational database, visualization of the results or obtaining online data from an operating scientific facility. In addition, providers can offer prepared in advance debugged and ready to execution workflows, constructed of an application software from third-party providers. This is in direct compliance with actively developing concept of Business Process as a Service (BPaaS) [3].

The main functional requirements for the platforms are the following: management of user credentials granting the right to use the available resources; remote administration of the web platform via a web browser; job execution management; data files transfer management; tools (services) management. These requirements correspond to the web platforms with minimal functionality, namely the ones intended solely for remote job execution. In the case of web platforms with more general functionality additional requirements arise.

Further line of development of the web toolkit, suggested in this work, may be related with the improvement of the technology of remote deployment of new application software on resources interacting with the web platforms. This approach will help to overcome an important problem associated with the use of the SaaS model in scientific areas, namely, limited set of application packages offered by SaaS providers. Often, these providers focus on mass servicing of single-type customers and scientific activity is beyond the scope of their interests. Currently, the provision of services for providers of application software in the context of scientific-oriented web platforms is not developed enough. Although some implementations (for example, the nanoHUB and e-Science Central) have services for remote application software deployment, they are still insufficient to ensure the creation of a web platform capable of performing the whole range of tasks characteristic for a free open market of application softwares.

General principles of constructing of the web platforms providing open marketplace for scientific application software in the framework of the SaaS model are discussed in the next section. Section 3 briefly describes possible operation algorithms of such a web platform.

## 2. Open Marketplace Paradigm

In this paper we propose basic features of the design of the web platform through which consumers of web services and their providers can interact with each other, and the platform needs to perform a range of functionalities, which are characteristic for a free open market. These tasks and the corresponding functional modules of the platform under development are presented in the Table 1.

We shall use the following nomenclature: consumer is a platform user requesting software as a service via the platform, provider is a user providing a software to consumers by means of the platform. There are two ways for offering an application software by means of the web platform.

In the first case, the software is deployed on provider’s own resources. Thus the provider offers not only a software, but also the computing resources. In this case the provider can use different types of resources, e.g., supercomputers, clusters, standalone servers, cloud systems. Of course, these computing facilities are not necessarily in property of the provider but can be leased resources (e.g., in a public cloud). This way is especially suits those providers who want to exclude completely access both users and platform administrators to the software (both
**Table 1.** Tasks which are characteristic for a free open market and the corresponding functional modules of the platform under development.

| Open market tasks     | Content of the task                                                                 | Modules (functionality) of the tasks                                      |
|-----------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| Information           | Marketplace gives participants the information about quantity and quality of products | service catalog with search engine;                                      |
| Matchmaking           | Matchmaking of users and SW providers                                                 | search engine of the service catalog; module for blog posting and for assigning rating marks |
| Price setting         | The price is formed on the market on the basis of supply and demand given the competition | price list in the catalog; billing module;                               |
| Regulation            | Market balances the supply and demand                                                 | service catalog with search engine; billing module; module of verifying compliance with the terms of service; module of e-wallets; |
| Stimulating innovation products | Market stimulates the development of innovation products, the reduction of costs and increase the quality | billing module; module for blog posting and for assigning rating marks |
| Coordination          | Market encourages service providers to create the necessary public services with the lowest cost | module for blog posting and for assigning rating marks                     |
| Optimization of market supply | Identification of inefficient products                                                 | logging module; billing module;                                          |

It is worth noting that in the case of provider’s own resources a problem of reliability of delivery of services may appear.

In the second case, the software is deployed on platform resources. In this case provider offers only a software as a service; a fee for the use of computing resources goes to the owners of the platform resources. The web platform have to provide a service for remote installation of the application software by users (providers) themselves. In this case the type of the resources is practically limited to the cloud systems. Indeed, an independent remote installation of application software on supercomputers by users is highly problematic both in terms of security and from a technical point of view: for a deployment of a software the user must deeply know the features of the architecture and software of the specific supercomputer. A natural solution to these problems is the use of virtual machines [5] that provide both safety (isolation of installed software) and feasibility of using the operating system required for the application software to be installed. Thus, since cloud infrastructure provides tools for managing virtual machines, the web platforms allowing software installation must provide access to the cloud systems in the framework “Platform as a Service” (PaaS) model.
3. Web Platform Architecture and Algorithms

Basically, all the web platforms have the typical three-layer architecture (see, e.g., [4]). The first layer is the frontend that provides the user web interface; the second layer is the platform engine responsible for job submitting, obtaining results, execution of workflows, etc. as well as the administration module that is responsible for job management, tool configuration, audit trails and user management; the third layer is a resource manager that exposes data and compute resources to the preceding layer. Also all web platforms have more or less sophisticated security infrastructure.

We will not discuss in details here the standard modules of a web platform responsible for job submission to remote resources (see, e.g., [2] and refs. therein). Instead we concentrate in this work on the specific modules needed for the web platform to operate as a marketplace. Part of such modules were indicated in Table 1. However, in addition the marketplace-like platform have to include a number of other auxiliary services for fully-featured operation. The list of the modules providing marketplace features includes:

- module of software registration;
- service catalog with search engine;
- logging module;
- billing module;
- integrator of conditions for workflows;
- module of verifying compliance with the terms of service;
- software deployment module;
- module of a tool setup;
- module of e-wallets;
- repository of the software codes;
- module for review and blog posting and for assigning rating marks.

Algorithm of the platform operation when a provider is offering his SaaS for consumers depends on the type of resources where the provider deploys the software. In the case of provider’s own resources the algorithms reads as follows (see figure 1; the numbers near the arrows corresponds to the items in the list below). The provider

1. registers himself at the platform and acquires a restricted access to the administrative interface;
2. A. in the case of provider’s own resources, deploys the software on this resources and provide an access to the software via the SSH protocol;
   (a) if the resource is not a standalone server but an HPC system with a batch manager, the latter should be in the list of managers supported by the platform;
2. B. deploys his software into the platform cloud by means of appropriate platform module (this implies the provider gets an access to a virtual machine with an appropriate operating system, etc.; PaaS model)
3. sets up his software and converts it in the web tool;
4. registers his web tool in the platform catalog togeter with
   (a) its description in a machine-readable form (e.g., via a completion of an appropriate web form),
   (b) terms of the service (including the price);
5. on regular basis checks his e-wallet and transfers the digital cash obtained to his bank account; this action should be preceded by consumption and logging/billing of the tools offered by the given provider (the corresponding arrows in the figure are marked with (5)).
The steps 3 and 4 may include setting up and registering debugged and ready to execution workflows, constructed of provider’s own application software or/and software from third-party providers in the spirit of the BPaaS model [3].

We suppose that each user of the platform (both a customer and a provider) has the electronic (virtual) wallet (e-wallet) containing the conditional means (digital cash) within a platform. Users can transfer their funds to the e-wallets of other users. In reality users also shall have opportunity to transfer funds from the e-wallets to real payment systems (like PayPal) or to bank accounts, but in this work we do not consider this process.

**Figure 1.** The algorithm of a provider and market modules operation.

For a consumer the both cases (that is separate resources or platform cloud) are equivalent (the type of resources is transparent for consumers). To use the platform tools, a consumer (see figure 2; the numbers near the arrows corresponds to the items in the list below):

1. chooses from the catalog appropriate tools;
2. designs out of these tools a workflow or select predefined business process (BPaaS);
3. gets terms of services and price for their use;
4. accepts or deny the terms of conditions;
5. if accepts, transfer digital money from his e-wallet to the provider’s e-wallet.
   (a) the module of verifying compliance with the terms of services checks that the conditions are accepted and ample means have been transferred to the providers’ e-wallets;
6. starts the workflow and obtains a result.

An administrator of the web platform provides user registration, moderation of software installation, user support and Quality of Service (QoS) obligations from the side of providers offering their own computing resources.

4. Conclusion
A prototype of such a platform, entitled eSciMart, is currently being developed at SINP MSU. The basis of the prototype platform is web service technologies and, in particular, the service-oriented architecture (SOA) [6], as well as the REST architectural style [7]. The approach
to creation of this web platforms market of application software is based both on the original solutions and on the synthesis and adaptation of the solutions used in research hubs (e.g., nanoHUB; nanohub.org), cloud and grid systems, as well as in on-line application stores. However, unlike the on-line app stores, the platform will not only provide information services for searching the tools needed by users, but also provide the feasibility of direct using of the necessary tools. Thus, the future eSciMart web platform will provide a single entry point both for web service providers and for their customers.

Acknowledgments
This work is supported by the RFBR grant No.15-07-09309.

References
[1] Weller M 2007 Virtual learning environments: using, choosing and developing your VLE (London: Routledge)
[2] Polyakov S P, Demichev A P, and Kryukov A P 2015 Proceedings of the 4th International Young Scientists Conference on Computational Science (Amsterdam: Elsevier) (to be published)
[3] Barton T and Seel C 2014 Proceedings of the Conference on Enterprise Modelling and Information Systems Architectures (Available at: http://subs.emis.de/LNI/Proceedings/Proceedings234/145.pdf)
[4] Hunter A A, Macgregor A B, Szabo T O, Wellington C A and Bellgard M I 2012 High Performance and Cloud computing. Source Code for Biology and Medicine 7 22
[5] James E S and Ravi N 2005 Virtual Machines: Versatile Platforms For Systems And Processes (San Francisco: Morgan Kaufmann)
[6] Erl T 2009 SOA Design Patterns (New York: Prentice Hall)
[7] Richardson I, and Ruby S 2007 RESTful Web Services (Cambridge: O’Reilly Media)