Supporting agricultural communities with workflows on heterogeneous computing resources

Akos Balasko, Robert Lovas, Mark Gergely
Laboratory of Parallel and Distributed Systems
MTA SZTAKI
Kende u. 13-17, Budapest, H-1111, Hungary
{balasko, rlovas, mgergely}@sztaki.hu

Nikos Manolis
Agro-Know Technologies
Grammou 17, 15235 Vrilissia, Athens, Greece
Email: manolisn@agroknow.gr

***Abstract*—In this paper we present a solution for agricultural communities to process large metadata record sets registered on distributed data repositories utilizing distributed computational and storage capabilities. Since significant number of the agricultural web-sites are based on Drupal content management system, we offer the user interface as a Drupal module. For sustainability reasons on-demand service deployment facility is enabled for cloud infrastructure.**

***Keywords*-grid computing, workflow management, on-demand service deployment, content management systems**

I. Introduction

Agricultural research communities are located in a wide, multidisciplinary array of research domains, ranging from molecular genetics to geo-sciences, including also social and economic sciences. The stakeholders at global level generate and use a vast and continuously growing amount of data; we are the witnesses of this data tsunami. Therefore, the current forthcoming infrastructure relies not only on geographically distributed data sources, registries (such as CIARD R.I.N.G. [3]), pre-existing and new data management, processing, and visualization components on top of them but cloud and grid computing resources are also involved in the infrastructure particularly for (meta)data processing purposes. Metadata in this scenario describe basic and detailed information about agricultural data in any form: e.g. experimental values, reports, and complete scientific papers. Complexity of the identified use cases lead to the requirement of defining them as workflow compositions. The workflow-based approach is in-line with the current trends; the High Level Expert Group on Scientific Data outlined a long-term vision in the Riding the Wave report [3] concerning collaborative data infrastructures that support seamless access, use, re-use, and trust of data. In their suggested framework the workflow concept has been positioned at two levels; workflow generation for community support services, and workflow execution at the layer of common data services. This paper follows this recommendation by developing workflows consisting data/metadata aggregators and service providers and offers a solution for handling and transformation of large metadata stored in CIARD R.I.N.G. repositories via content management system interfaces.

To achieve the objective of the communities, the following requirements are identified:

1) Performing the composition of complex processes(indexing, deduplication of metadata records etc.) is required in a completely hidden way
2) Access heterogeneous resources transparently
3) The system must be manageable via content management systems (CMS) (mainly via Drupal [1], [2])
4) The system must offer a sustainable solution.

In technical terms the community would like to have a specific scientific gateway, which is capable to define and enact workflow compositions, and - considering that access through CMS applications is a requirement as well – it must provide a clear API for its core features to by-pass its original user interface.

This paper is organized as follows. Related work is discussed in Section II, then Section III details the main concept to be implemented, while Section IV introduces the component applications ported to the grid infrastructure. Section V introduces the gUSE workflow composition. Section VI shows how to invoke gUSE framework to execute a workflow remotely, while Section VII introduces the concept of Cloud Orchestrator used to achieve on-demand deployment of the science gateway. Conclusion, Future work and Acknowledgement close the paper.

II. Related Work

Science Gateways can be developed in two ways: they can be elaborated from scratch, or using and customizing an available Science Gateway Framework. The prior case is obviously a time-consuming process and requires solid expertise in various fields from scripting the underlying resources to designing the user interface. In addition its maintenance and its sustainability is a problematic task to solve. In contrast, creating a Science Gateway based on an existing framework can ease its building process and maintenance, since the core components are supported by the framework, and the developers can focus on serving the needs of the scientific community only. Many generic science gateway frameworks are in operation. Despite addressing the same achievement, namely to enable the users to utilize remote distributed grid or cloud resources for facilitating scientific research, they offer different possibilities concerning the complexity of the applications to be used by the scientists.

Vine Toolkit [12]: It provides a general set of extensible APIs written in JAVA to ease creating web applications that hide real applications executed on distributed infrastructures for creating science gateways as it is described in [13].

Catania Science Gateway Framework [17]: Another approach to build an interface that provides the basic functionality required for executing distributed applications is Catania Science Gateway Framework (CSGF) by INFN, offering a standardized programming interface specification needed to create an application-domain science gateway. CSGF is made of a set of libraries to manage authentication and authorization mechanisms and to interact with several different kinds of computational (including grid and cloud) resources. The essential part of the CSGF is the Grid-Engine, a tool that implements access to various computational resources (resources) adopting the OGF standard Simple API for Grid Applications (SAGA)[14] and its JAVA implementation (called JSAGA). One example for a science gateway based on CSGF is DECIDE [16] science gateway, which helps the early diagnosis and research on brain diseases.

EnginFrame [7] is a generic cloud Portal that also provides access to Applications, Data, and the HPC compute farm through a standard web browser, developed by NICE s.r.l. It can be customized for specific community needs, e.g. as described in [8], the framework was used to create a community-used web-based interface on the top of iRods Data Grid. It is connected with different workflow engines [11] such as Taverna [9] or Kepler [10] to enable the users to organize their applications in complex structures. It does not contain native workflow enactment system.

InSilicoLab Framework [18] Among others InsilicoLab provides a customisable framework offering comprehensive tools for data management for a science gateway allowing access to the data, its metadata and provenance information about the experiment. The experiments can be executed in distributed resources, and an automatic parallelization possibility is included by enabling parameter sweep execution of an experiment against large and separable input data fields. Pipelines of experiments are also supported, so it provides an intuitive common workflow execution, but implementing complex workflow structures requires programmatic expertise.

AgriDrupal extends Drupal, an open source content management system, with ready-to-use functionalities for agricultural information management. AgriDrupal is conceived as a light and easy-to-use tool - it requires a basic LAMPP environment (a preconfigured Linux, Apache, MySQL, PHP, and Perl software stack) - whose adoption can be sustainable also in small institutions with little information technology support. The added value of AgriDrupal compared to a vanilla installation of Drupal is that an AgriDrupal installation already contains the necessary content types, taxonomies and views to manage a basic agricultural information system. However it provides a high-level agricultural specific user interface, its weak point is the accessing distributed systems for data processing.

Whilst the prior Science Gateway frameworks are promising solutions, most of them allow defining and executing individual jobs only, or support concatenating those jobs in a programmatic way. In contrast, frameworks supporting workflow management are mostly stand-alone applications so they cannot be used remotely as they are. To conclude, these frameworks cannot be used to satisfy the requirements of the agricultural community.

III. Concept

In order to solve aggregation of metadata, we must have a clear view of their structure. Metadata files or as also known, metadata records are grouped in datasets. Several datasets are stored by catalogues sites (targets henceforward) covered by such CMS user interfaces allowing browse and search on them. These catalogue sites are registered in the CIARD R.I.N.G. registry maintained by The Global Forum of Agricultural Research (GFAR) under the umbrella of CIARD. The R.I.N.G. registers meta-information from several types of catalogues such as catalogues containing agricultural news by RSS feeds, or discrete research data etc. The current version of the science gateway deals with two categories: Educational, that uses OAI_LOM metadata format [23] and Bibliographic based on Agris_AP metadata format.

WS-PGRADE/gUSE by MTA SZTAKI is a generic web-based framework, which has been developed to support different scientific communities by automatizing the execution of their process compositions a.k.a. workflows on various heterogeneous and distributed computational resources such as clusters, grids or clouds. Moreover, the system provides APIs for accessing the core features by other web-based or stand-alone applications.

A good option to satisfy the needs of the community, is a customized WS-PGRADE/gUSE workflow manager [19], [20] and science gateway framework because of the following reasons. Various types of resources can be accessed by gUSE, so jobs defined in the system can represent stand-alone applications submitted or pre-deployed on grid or cloud resources or they can represent invocation of web-service interfaces (HTTP, Rest), too. Considering that the processes to be executed are stand-alone applications and the manipulation of the retrieved metadata is done by CouchDB [21] that provides REST-based web-service interface, possibility of accessing different types of resources is a real benefit to be utilized. As gUSE offers a data-driven workflow language with support of a graphical web-based configuration interface, the jobs can be combined together to define a more complex process. In addition gUSE provides APIs for customizing a scientific gateways upon the generic framework or to execute prefabricated workflows remotely (called Remote API). As Remote API is implemented as a simple servlet interface, it can be invoked from any kind of client applications, including content management systems as well. Figure 1 illustrates the idea of the architecture designed for fitting all the requirements. The numbers denote consecutive steps of the harvesting use case.

The main idea is to establish connection between CMS systems and gUSE framework using Remote API by developing a new simplified Drupal module as gateway interface. It

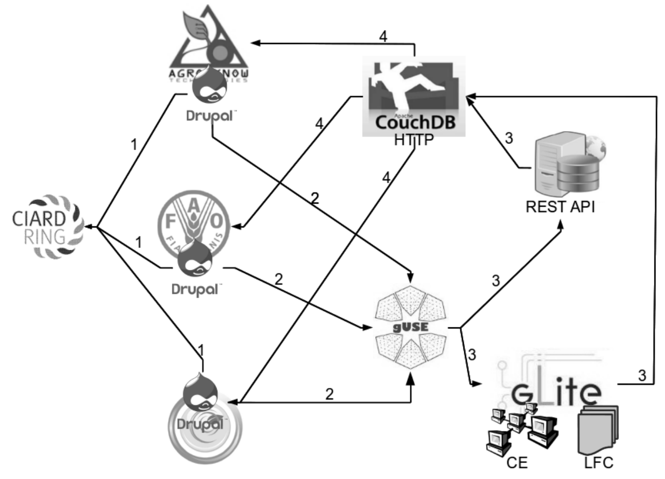


Fig. 1. Overview of the interacting architectural components

enables the users to search and to browse in catalog of CIARD R.I.N.G. (step 1), and to be able to submit and interpret the harvesting workflow (step 2). Then gUSE enacts the workflow by executing harvest and transformer applications pre-deployed on gLite resources and by invoking REST API webservices (step 3). Records are placed on the distributed system and are registered in the Logical File Catalogue (LFC). To avoid complicated security issues required by the applied technology, automated processes are triggered by the register process to publish them for other components outside of the grid using CouchDB(step 4 shows the lines that the harvested metadata sets can be downloaded from the certain Drupal modules). The composition of processes to be executed on grid and service resources is shown in Figure 2. We remark that our main objective was to illustrate the whole flow of the integrated systems can work together, some of the components shown in Figure 2 are skipped in the workflow implementation hence, further improvement of the workflow is required in the future.

The aggregation workflow composition covers the following processes. First of all, all datasets belonging to the given target must be downloaded (harvesting process), then the metadata set is going to be filtered according to the users settings. After this step all of the records must be transformed into the internal format required by the user. The next step is checking the availability of the links of the real data, then after a post-processing method the records are being stored. In case of duplication or if a link in a record is broken, the records are stored, but remarking that they were failed at the validation step.

In order to achieve a sustainable software stack, we realized that creating and maintaining a composition of applications is not enough, since it requires a deployed and accessible, and therefore maintained science gateway as well. Hence, we aimed at on-demand deployment of the systems involved into the calculation to be executed, including the science gateway itself. Moreover, on-demand deployment must be suitable for various types of cloud infrastructures. Therefore, we must review the available software for on-demand service deployment and/or orchestration tools as well.

IV. Components

Aggregation workflow consists of components pre-deployed on grid infrastructure and invokable by applications submitted as simple gLite jobs. These components are the followings:

1) agDataHarvester: agDataHarvester performs harvesting of any dataset exposed via an Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)[5] target. The module is type agnostic with respect to the metadata to be harvested (DC, LOM) and can support harvesting of any metadata format as this is declared in the metadaPrefix field of the verb ListMetadataFormats of an OAI-PMH target. The harvested metadata records are stored in the specified directory path. The application requires the following parameters to be provided as input:

- OAI target: The URL of the OAI-PMH target;
- Destination directory: The directory path where the harvested metadata records will be stored;
- Metadata Prefix: The metadata prefix indicating the metadata schema that records follow.

2) agLOMtoAKIF: agLOMtoAK performs conversion of a set of metadata records with XML binding that follow IEEE LOM metadata format into JSON files that follow the agINFRA internal representation format (AKIF) for harmonization of metadata describing educational resources. The application requires the following parameters to be provided as input:

- Input directory: The directory/root path where the initial metadata records are stored;
- Output directory: The directory path where the transformed metadata records will be stored;
- Error directory: The directory path where the error-prone records will be stored;
- Set name: The collection name to be transformed.

3) agAGRIStoAGRIF: agAGRIStoAGRIF performs conversion of a set of metadata records with XML binding that follow AGRIS application profile (http://aims.fao.org/metadata/sets/agris-application-profile) into JSON files that follow the agINFRA internal representation format (AGRIF) for harmonization of metadata describing bibliographic resources. The application requires the following parameters to be provided as input:

- Input directory: The directory/root path where the initial metadata records are stored;
- Output directory: The directory path where the transformed metadata records will be stored;
- Error directory: The directory path where the error-prone records will be stored;
- Set name: The collection name to be transformed

4) agLinkCheck: agLinkCheck is a stand-alone application for checking if the URL for accessing the learning objects included in the metadata records is broken or not. The metadata records with broken URLs are stored in a separate directory. In

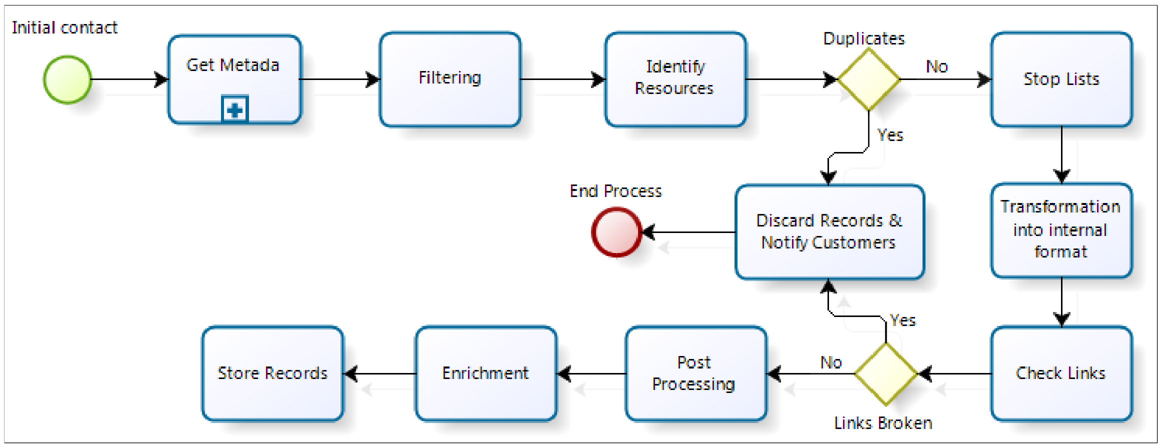


Fig. 2. Concept of data aggregation processing workflow

case of URLs that redirect to other URLs these are considered to be broken and they have to be evaluated by curators. In order to facilitate such kind of tasks after each run of the link checker the numbers of broken links per domain are stored into log files so that it is clear in which domain there are several problems.

V. Workflow Management

This section introduces the workflow developed in gUSE according to the concept shown in Figure 2 and based on the components described in the previous section.

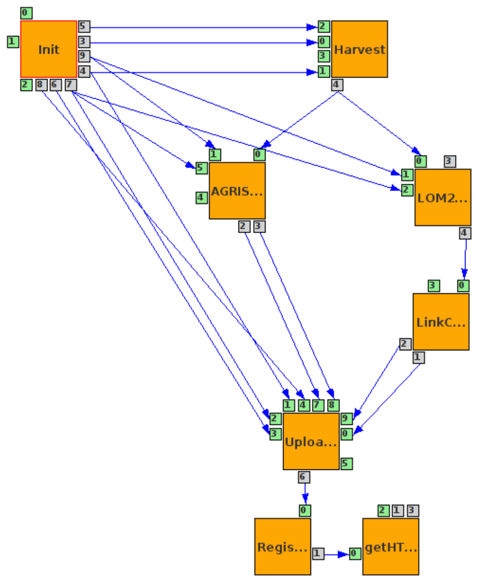


Fig. 3. Implementation of the data aggregation processing workflow

At first, according to semantics of gUSE, job Init is executed, which just systematizes the settings concerning their function. For instance, users can set the URL of the target to be harvested (selected from CIARD R.I.N.G.), but the workflow has more additional options for testing and fine-tuning purposes (setting the path on where the records are stored in the grid, how many record should be harvested from the target and so on). Next process to be executed is job Harvest (it is *agDataHarvester* application introduced in section IV), which executes and manages the real pre-deployed harvest process. Then depending on the metadata format required by the users, one of the next parallel branches Agris and LOM2 + LinkC is executed to perform the needed transformations. These jobs represent *agAGRIStoAGRIF*, *agLOMtoAKIF* and *agLinkCheck* respectively, moreover it must be remarked that LinkC is required just in case if the metadata format is IEEE LOM. Both branches validate whether a record can be transformed to the metadata format, hence two record-sets will be produced by these jobs containing the correct and the wrong records separated. Then the produced folders will be compressed and uploaded by job Upload into the given LFC folder, then job Register registers the files uploaded by REST service invocation to CouchDB resulting that the files can be accessed from outside of the grid. Finally as registration takes time the last job sends a request to CouchDB to get the metadata in-formation about the registered filesets in JSON format. These output files are sent back to the users through the Drupal module providing complete URLs on where the harvested records can be found.

VI. On-demand Execution

gUSE system can be extended to let the community to send a complete workflow via HTTP protocol circumventing the original WS-PGRADE interface. In technical terms this is a servlet extension of WS-PGRADE called Remote API. In this solution members of community do not have to register to the WS-PGRADE portal, namely they don't have to have a valid user account on the portal. Although it means that they can not develop their own workflow, it is not a real restriction since, they can use workflows that were prefabricated by workflow developers. These workflows must be available on the client machine. However, the workflow nodes are going to be executed on an agINFRA [22] infrastructure, which requires authentication (X509 user proxies for grids), using robot certificates for such nodes can resolve this requirement. As the users are unauthenticated in the portal's view, the API call creates a new temporary user before every execution, and the workflow will be submitted on behalf of this temporary user. After downloading the outputs of the workflow, this temporary user and all of the related files and database rows are erased. It shows the main disadvantage of this solution, since the users' executions are not stored on the portal, the users cannot retrieve outputs which have already been downloaded once. To avoid unauthorized execution on the portal, the clients must possess a password set previously with the server to establish the connection. Figure 4 illustrates the workflow's path through the components. At the user side – as it can be

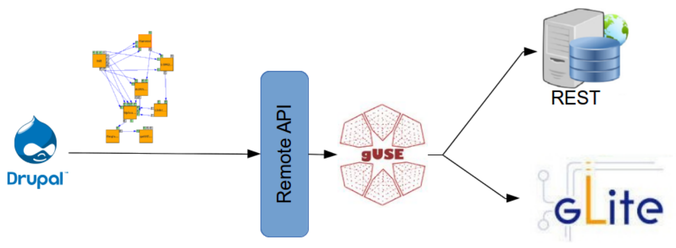


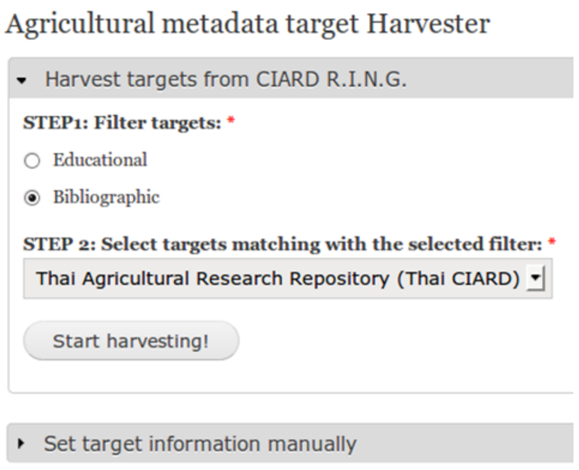
Fig. 4. Overview of On-demand execution in gUSE

seen at the left hand side of Figure 4, there is a Drupal- based CMS. Since Drupal enables reusability of compomenets as modules, a client has been developed to access and submit the aggregation workflow to gUSE via Remote API as a Drupal module. In the users' point of view this client module hides all of the underlying complexity. For example a user would like to get bibliographic metadata sets collected in the Thai CIARD, it can be selected for harvesting in the client interface as it is shown in Figure 5 and by clicking "Start harvesting!" button, the complete harvester workflow are sent to gUSE for enacment together with the current settings, and, after its execution (the interface show its progress), the harvested metadata sets can be downloaded via simple links (shown in Figure 6).

VII. On-demand deployment

Due to sustainability reasons the system including the infrastructure must be reproducible on-demand. To be able to set up the complete infrastructure on-demand whenever it is needed, a new component service called Cloud Orchestrator (CO) has been developed, which is capable of composing and deploying a set of instances of virtual machines (VM) eliminating the installation and configuration difficulties of the software components enclosed to the given VM. It can be used to deploy and start a gUSE instance in any OpenNebula-based cloud, which is preconfigured precisely for accessing agINFRA-related remote computational and storage infrastructure via gLite middleware.

As shown in Figure 7, the on-demand deployment can be initiated by the user. The user is required to complete a web form which contains the name of the requestor, the email


Agricultural metadata target Harvester

Harvest targets from CIARD R.I.N.G.

STEP1: Filter targets: *

○ Educational

[✓ Bibliographic]

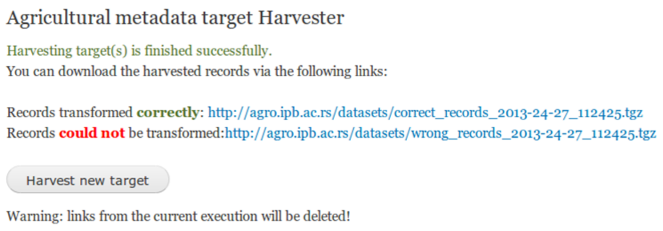
STEP 2: Select targets matching with the selected filter: *

Thai Agricultural Research Repository (Thai CIARD)

Start harvesting!

Set target information manually

Fig. 5. Client Drupal interface - options


Agricultural metadata target Harvester

Harvesting target(s) is finished successfully.

You can download the harvested records via the following links:

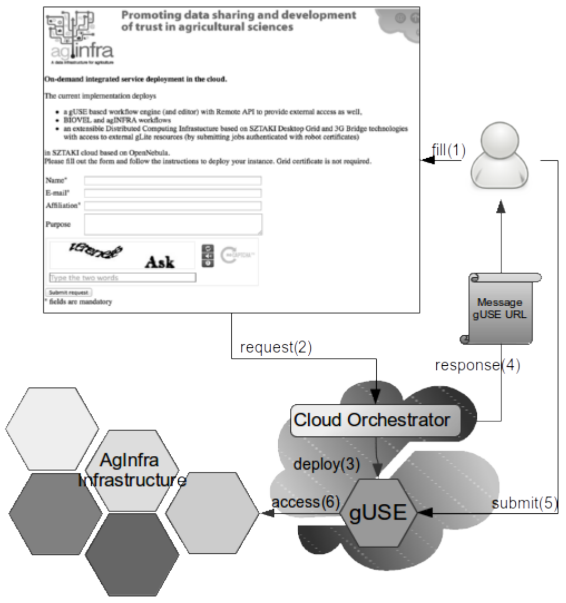
Records transformed **correctly**: http://agro.ipb.ac.rs/datasets/correct_records_2013-24-27_112425.tgz

Records **could not** be transformed:http://agro.ipb.ac.rs/datasets/wrong_records_2013-24-27_112425.tgz

Harvest new target

Warning: links from the current execution will be deleted!

Fig. 6. Client Drupal interface - results screen


Promoting data sharing and development of trust in agricultural sciences

On-demand integrated service deployment in the cloud.

The current implementation deploys

- a gUSE based workflow engine (and editor) with Remote API to provide external access as well,
- BIOVEL and agINFRA workflows
- an extensible Distributed Computing Infrastucture based on SZTAKI Desktop Grid and 3G Bridge technologies with access to external gLite resources (by submitting jobs authenticated with robot certificates)

in SZTAKI cloud based on OpenNebula.
Please fill out the form and follow the instructions to deploy your instance. Grid certificate is not required.

Name*
E-mail*
Affiliation*
Purpose

Type the two words

Submit request

* fields are mandatory

Fig. 7. Process of on-demand deployment of gUSE

address, affiliation, and purpose of the request (step 1). Once the request is approved, the web page forwards it to the CO (step 2). The CO component inspects the request and checks whether it can be deployed into one of the cloud resources. The checking procedure consists of the following basic tests:

1) is the cloud reachable
2) are the required images ready to be used
3) are the supplied credentials working

If it finds a suitable cloud resource, then it initiates the deployment (step 3). The first step of the deployment is to start a VM instance from an image, which contains the WS-PGRADE/gUSE. The requestor email address and a password is contextualized into the VM instance. The portal appliance is prepared to use these contextualized information in order to grant access to the requestor. Once the appliance VM is running, the CO checks the portal that it can serve requests and the supplied credentials are working. After every check passed, the framework notifies the requestor about the successful deployment (step 4). The email notification contains the URL, and the required credentials which are needed to access the portal. A successful deployment results a WS-PGRADE/gUSE framework connected to the available agINFRA infrastructure extended by standard remote API interface which can be invoked from the Drupal interface shown before (step 5 and step 6).

VIII. Conclusions

To summarize, to serve agricultural communities we have integrated gUSE framework with a new Drupal-based interface that can be used from any other CMS. We have developed and introduced a workflow in gUSE for aggregating meta-data datasets contained by various catalogue sites collected by CIARD R.I.N.G. Then due to sustainability reasons, we showed a way to deploy the portal framework on demand.

IX. Future work

Further plan of this work is extending the workflow by adding new transformation components capable to transform different metadata schema. Component of deduplication is also to be included in the next version of the workflow. According to these modifications the Drupal interface must be extended with the possibility of selecting other schemas. In case of on-demand deployment, we plan to extend the Drupal interface with an automated solution for setting the newly deployed gUSE framework as the new end-point, and we plan to develop a set of VMs containing the whole infrastructure (except the datastore component) to be deployed in a cloud infrastructure achieving a complete solution for sustainability issues. It can be done by utilizing the improved features of Cloud Orchestrator.

X. Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 283481 (SCI-BUS) and under grant agreement no 283770(agINFRA).

References

[1] Butcher, Matt, Larry Garfield, and John Wilkins. Drupal 7 Module Development. Packt Publishing Ltd, 2010

[2] Melanon, Benjamin, et al. The definitive guide to Drupal 7. Apress, 2011.

[3] Pesce, Valeria, Ajit Maru, and Johannes Keizer. "The CIARD RING, an Infrastructure for Interoperability of Agricultural Research Information Services." Agricultural Information Worldwide 4,1, 2011

[4] European Commission. Riding the Wave: How Europe can gain from the rising tide of scientific data., http://cordis.europa.eu/fp7/ict/e-infrastructure/docs/hlg-sdi-report.pdf (accessed at 20.03.2014)

[5] http://www.openarchives.org/pmh/ (accessed at 20.03.2014)

[6] https://geodynamics.org/portals/seismo/ (accessed at 20.03.2014)

[7] Torterolo L, Porro I et. al.:. *Building Science Gateways with EnginFrame: a Life Science example*. Proceedings of International Workshop on Portals for Life Sciences, Edinburgh, United Kingdom, September 14-15, 2009.

[8] Venuti N., Torterolo L. et. al.:. *A Service-Oriented Interface to the iRODS Data Grid*. Proceedings of the International Worshop on Science Gateways, Catania, Italy, September 20-21, 2010.

[9] Hull D., Wolstencroft K. et. al.:. *Taverna: a tool for building and running workflows of services*. Nucleic Acids Research, p. 34, 729-732, 2006.

[10] Ludscher B., Altintas I. et. al.:. *Scientific workflow management and the Kepler system: Research Articles*. Concurrency and Computation: Practice and Experience, vol 18, issue 10, p. 1039-1065, 2006.

[11] Bartocci E, Cacciagrano D et al.:. *A Grid infrastructure for managing workflows in bioinformatics applications*. Proceedings of the NET-TAB2006. Santa Margherita, p. 38-44. 2006.

[12] Dziubecki P., Grabowski P. et. al.: *Easy Development and Integration of Science Gateways with Vine Toolkit*. Journal of Grid Computing, pp. 1-15, 26 October 2012.

[13] Dziubecki P. et al.: *Nano-Science Gateway development with Vine Toolkit and Adobe Flex*. Proceedings of the International Worshop on Science Gateways, Catania, Italy, 2010.

[14] Goodale T., Jha S. et.al: *SAGA: A Simple API for Grid Applications*. Computational Methods in Science and Technology, vol. 12., issue. 1, pp. 7-20, 2008.

[15] Lindemann J., Sandberg G.:. *An extendable GRID application portal*. Advances in Grid Computing-EGC 2005, pp. 322-325, 2004.

[16] Ardizzone V., Barbera R. et. al.:. *The DECIDE Science Gateway*. Journal of Grid Computing, pp. 1-19, 2011.

[17] V. Ardizzone et al., Science Gateways for Semantic-Web-Based Life Science Applications ,HealthGrid Applications and Technologies Meet Science Gateways for Life Sciences, Studies in Health Technology and Informatics, volume 175, pp. 119-130, 2012.

[18] J. Kocot, T. Szepieniec, D. Harlak, K. Noga, M. Sterzel: InSilicoLab Managing Complexity of Chemistry Computations. In: M. Bubak, T. Szepieniec, K. Wiatr (Eds) Building a National Distributed e-Infrastructure - PL-Grid - Scientific and Technical Achievements, Springer 2012, ISBN 978-3-642-28266-9, pp. 265-275, 2012.

[19] Balasko A, Farkas, Z., and Kacsuk, P., Building Science Gateway by Utilizing the Generic WS-PGRADE/gUSE Workflow System, Computer Science, vol. 14, no. 2, p. 307, 2013.

[20] P. Kacsuk, Farkas, Z., Kozlovszky, M., Hermann, G., Balasko, ., Karczkai, K., and Mrton, I., WS-PGRADE/gUSE Generic DCI Gateway Framework for a Large Variety of User Communities, Journal of Grid Computing, vol. 10, no. 4, pp. 601 - 630, 2012.

[21] Anderson, J. Chris, Jan Lehnardt, and Noah Slater. CouchDB: the definitive guide. O'Reilly Media, Inc., 2010.

[22] Pesce, Valeria and Geser, Guntram and Protonotarios, Vassilis and Caracciolo, Caterina and Keizer, Johannes Towards Linked Agricultural MetaData: Directions of the agINFRA Project, In AgroSEM 2013, Thessaloniki (Greece), 19-22 November 2013.

[23] IEEE LOM Specification
http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
(accessed at 20.03.2014)