Towards a Global Service Registry for the World-Wide LHC Computing Grid

Laurence Field, Maria Alandes Pradillo, Alessandro Di Girolamo

CERN, Geneva, Switzerland

E-mail: `laurence.field@cern.ch`

Abstract. The World-Wide LHC Computing Grid encompasses a set of heterogeneous information systems; from central portals such as the Open Science Grid's Information Management System and the Grid Operations Centre Database, to the WLCG information system, where the information sources are the Grid services themselves. Providing a consistent view of the information, which involves synchronising all these informations systems, is a challenging activity that has lead the LHC virtual organisations to create their own configuration databases. This experience, whereby each virtual organisation's configuration database interfaces with multiple information systems, has resulted in the duplication of effort, especially relating to the use of manual checks for the handling of inconsistencies.

The Global Service Registry aims to address this issue by providing a centralised service that aggregates information from multiple information systems. It shows both information on registered resources (i.e. what *should* be there) and available resources (i.e. what *is* there). The main purpose is to simplify the synchronisation of the virtual organisation's own configuration databases, which are used for job submission and data management, through the provision of a single interface for obtaining all the information. By centralising the information, automated consistency and validation checks can be performed to improve the overall quality of information provided. Although internally the GLUE 2.0 information model is used for the purpose of integration, the Global Service Registry in not dependent on any particular information model for ingestion or dissemination. The intention is to allow the virtual organisation's configuration databases to be decoupled from the underlying information systems in a transparent way and hence simplify any possible future migration due to the evolution of those systems.

This paper presents the Global Service Registry architecture, its advantages compared to the current situation and how it can support the evolution of information systems.

1. Introduction

Obtaining information about the structure and state of Grid services is a fundamental requirement that enables higher-level Grid functions. Although information describing each Grid service is provided by the service itself, this is not the only source of information about Grid services. Grid infrastructures are managed entities and hence are governed by policies that define among other things, which services *should* be in the infrastructure. For the worldwide LHC Computing Grid (WLCG) [3], there are two information sources that state which services *should* be in the infrastructure: the Open Science Grid's (OSG) [9] Information Management System (OIM) [11] and the Grid Operations Centre Database (GOCDB) [7]. These information sources also provide information on scheduled and unexpected downtimes. The lists of services that *should* be in the infrastructure is used by the WLCG information system [6] to discover the site-level information services and hence all the services that *are* in the infrastructure. The WLCG information system is then used to find further information about the structure and state of available Grid services.

From the Virtual Organisation's (VO) perspective, the primary information source is their own configuration database [2, 8, 10, 12] which is used to drive their experiment framework. This not only contains generic information about services but in addition includes information that is specific to a particular VO such as service annotations using internal naming and semantics. Hence, there is a need to populate these configuration databases from the various information systems that are the sources of information about Grid services. Currently, this requires interfacing to multiple information systems that use different, and in some cases non-standard, interfaces and information models. As the information obtained is provided in different formats, the structures, and in some instances the values, need to be transformed. In addition, any issues relating to the quality of the information or the reliability of the information system need to be handled, which includes addressing inconsistency.

The Global Service Registry (GSR) aims to address these issues by providing a centralised service that aggregates information from multiple information systems and provides a consistent view from which the VO can populate their configuration database. The reasoning behind the introduction of the GSR is provided in Section 2 and its architecture is presented in Section 3. The implementation of the GSR is described in Section 4 along with the results from its initial deployment and some concluding remarks are provided in Section 5.

2. Configuration Databases

The fundamental requirement is the need of the VO to discover the resources at their disposal and to obtain further information about their structure and state so that they can be exploited [5]. From the perspective of the LHC VOs, this is achieved by means of their own configuration databases [2, 8, 10, 12].

For ALICE, a central LDAP server is used [10], which is updated manually based on out-of-band communication. LHCb uses a Configuration Service as part of DIRAC [12] to provide static configuration parameters to all of its distributed components. A single master service is used that supports both automatic and manual updates. Multiple read-only slave services are geographically-distributed to provide both redundancy and load balancing capabilities. Within CMS the SiteDB [8] is used and each site has the responsibility of maintaining information about itself and the services it provides. This is implemented using an Oracle database which can be accessed through both GUI and API Web interfaces. The ATLAS Grid Information System (AGIS) [2] provides configuration and status information about resources, services and topology needed by the ATLAS Distributed Computing applications and services. It is implemented using Django as a high-level Web application framework which provides an Object Relational Mapping functionality to an Oracle database backend. The information stored is both automatically retrieved from other sources and managed directly within AGIS.

From these configuration databases we can conclude the following:

(i) there is a need for a configuration database;

(ii) the requirements are driven by the experiment framework;

(iii) the configuration database is the information source for the experiment framework;

(iv) the information is quasi-static and describes the topology of the infrastructure from the VO's perspective;

(v) both manual and automated update process are requried;

(vi) there is duplication of information and functionality between these configuration databases and other information sources;

(vii) there is duplication between the information and functionality of the configuration databases themselves.

Duplication and inconsistencies can be discovered when the information is aggregated from multiple information sources and hence, centralisation of the information helps to ensure consistency. It also simplifies the annotation or modification of the information as further synchronisation of these updates from disparate information sources is not required. To avoid redundancy and to improve performance, caching mechanisms can be used where required. The configuration database simplifies the maintenance of this cache by avoiding the need to maintain synchronisation with multiple information sources. A centralised service is also easier to manage than a distributed system with respect to adding capabilities.

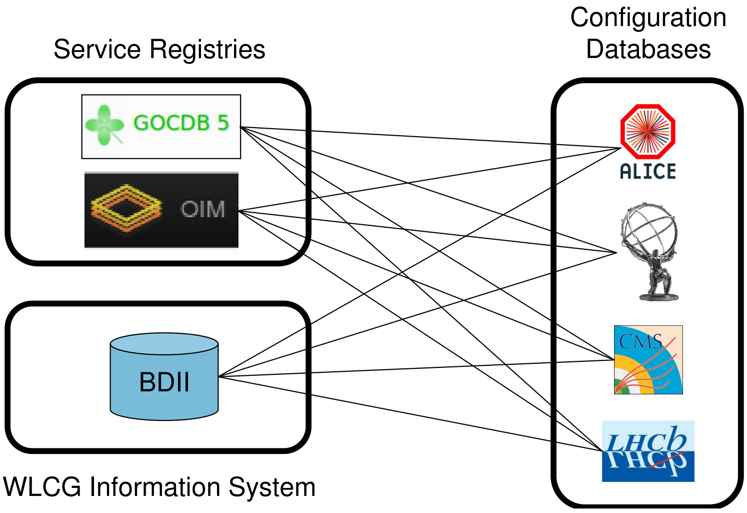


Figure 1. The connectivity between components in the existing scenario

The configuration databases employ both automated and manual update methods. Automated processes are suited to the accurate reproduction of repetitive tasks, such as adding information from an information source. However, as errors are also replicated, the result is dependent upon the validity of the original information unless validation steps are introduced into the process. As automating validation checks can be difficult, manual checking can provide a simple alternative. In addition, manual processes are needed where no information source exists and is provided via out-of-band communication. It must therefore be understood why there are errors in the original information sources and why no information source exists for the information that is required.

The main issue is the duplication of effort required to provide multiple implementations which essentially provide the same function. The current situation is shown in Figure 1.

3. The Global Service Registry

The GSR aims to provide a common solution from where VOs can obtain information about Grid services in the WLCG infrastructure. It follows a layered approach whereby the WLCG infrastructure (e.g. Grid services) provides a broad foundation upon which domain-specific solutions can support the unique requirements for a particular VO. The goal is to ensure that common requirements are met by generic solutions so that the VO can focus their efforts on their unique challenges. In this specific case, the provision of common information is provided by the GSR, leaving the VO to focus on their own annotations. Figure 2 shows location of the GSR in this layered approach and how it relates to the information sources.

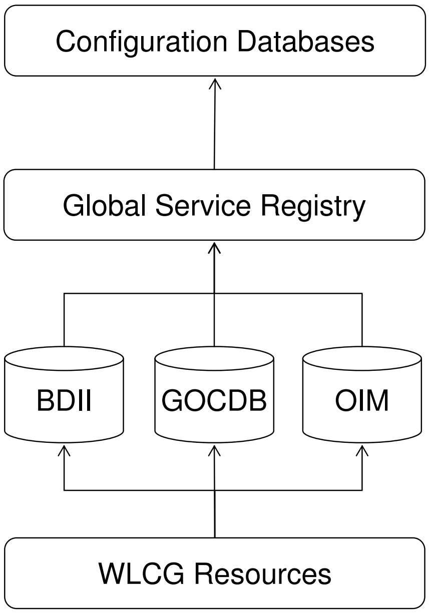


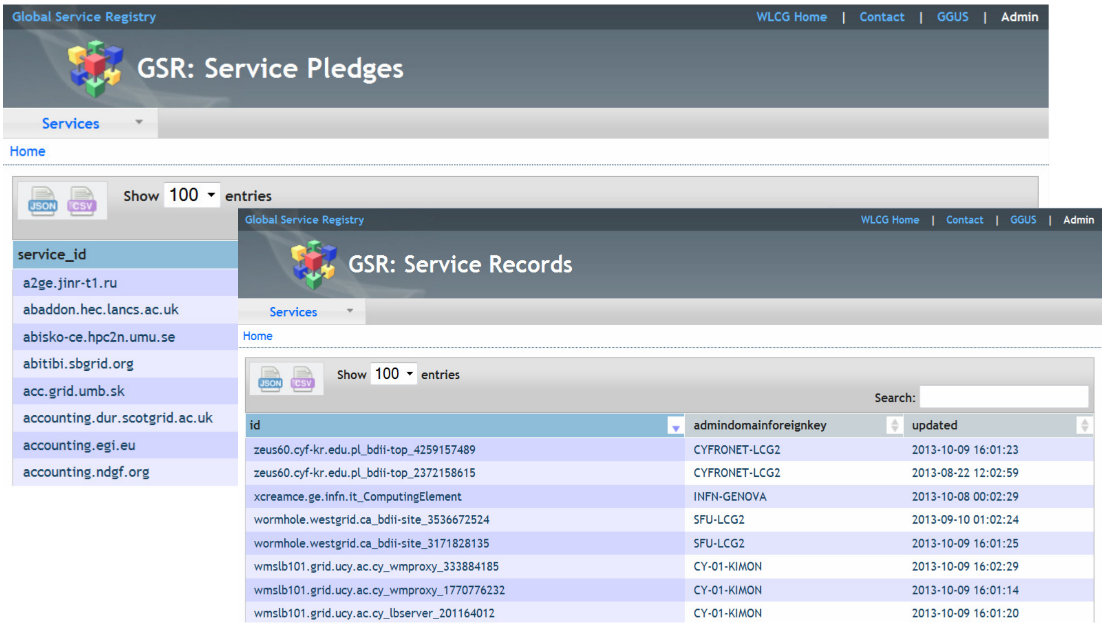
Figure 2. The layered architecture of the Global Service Registry

The GSR provides an intermediate layer to address the challenge of aggregating information from many sources. It is a single interface, a unique entry point about both registered and actual Grid services. This simplifies the task of the VO to gather the information they need for their configuration databases, as they do not have to contact multiple information sources and also protects the VO from possible changes to the information source interfaces.

Internally the GSR uses the GLUE 2.0 information model [4] as an intermediate translation layer, however for ingesting and dissemination, information can be translated from other models through the use of plug-ins. Central to the GSR is a database that stores the information about the WLCG services. This database can be used to discover inconsistencies between information sources. For each information source, a plug-in is created to ingest the information which translates or modifies it if necessary. These plug-ins can be used to provide a hotfix, that is to fix a problem quickly while a more permanent solution is being prepared. This is required, for example, in situations where there is a software defect with an information provider and it takes time to update all the available instances. As the GSR is a single central service, updates can be more easily applied as a distributed roll-out of the update is avoided. In addition, quality checks can be applied to both the output of the plug-ins and the database itself to validate the information. A plug-in concept is also used to provide views of the information.

4. Implementation and Evaluation

The GSR is implemented using the Django Web application framework. A rendering of the GLUE 2.0 information model has been provided as a Django model and is used to initialise a MySQL database. Currently the GSR employs four plug-ins to ingest information and insert the result into the database using Django's Object Relational Mapping functionality. The first is the GLUE 2.0 snapshot, which queries the WLCG information system, and is available as a stand alone package along with the corresponding Django model. As the OSG does not provide information in the GLUE 2.0 format, a second plug-in was created that obtains the required information in the GLUE 1.3 format [1] and performs the necessary translation. A third plug-in obtains service registrations by querying the GOCDB and OSG OIM respectively. Similar plug-in obtains the service downtimes from both the GOCDB and OSG OIM.



Global Service Registry WLCG Home | Contact | GGUS | Admin

GSR: Service Pledges

Services

Home

Show 100 entries

service_id
a2ge.jinr-t1.ru
abaddon.hec.lancs.ac.uk
abisko-ce.hpc2n.umu.se
abitibi.sbgrid.org
acc.grid.umb.sk
accounting.dur.scotgrid.ac.uk
accounting.egi.eu
accounting.ndgf.org

Global Service Registry WLCG Home | Contact | GGUS | Admin

GSR: Service Records

Services

Home

Show 100 entries

Search:

id	admindomainforeignkey	updated
zeus60.cyf-kr.edu.pl_bdii-top_4259157489	CYFRONET-LCG2	2013-10-09 16:01:23
zeus60.cyf-kr.edu.pl_bdii-top_2372158615	CYFRONET-LCG2	2013-08-22 12:02:59
xcreamce.ge.infn.it_ComputingElement	INFN-GENOVA	2013-10-08 00:02:29
wormhole.westgrid.ca_bdii-site_3536672524	SFU-LCG2	2013-09-10 01:02:24
wormhole.westgrid.ca_bdii-site_3171828135	SFU-LCG2	2013-10-09 16:01:25
wmslb101.grid.ucy.ac.cy_wmproxy_333884185	CY-01-KIMON	2013-10-09 16:02:29
wmslb101.grid.ucy.ac.cy_wmproxy_1770776232	CY-01-KIMON	2013-10-09 16:01:14
wmslb101.grid.ucy.ac.cy_lbserver_201164012	CY-01-KIMON	2013-10-09 16:01:20

Figure 3. Example views from the Global Service Registry portal

Django views are used to disseminate this information to the Configuration Databases. Although the plug-ins obtain all information from the information sources, these views can present only information that is specifically required by the VOs. By default, each specific object type of the GLUE 2.0 information model is available as a Web page and the result can be seen in Figure 3. This is achieved using Django views and the jQuery datatables library. An icon on the table links to a persistent URL where the data shown in the table can be downloaded in either the JSON or CSV formats. In addition, custom views are provided to simplify the ingestion of information into the VO's own Configuration Database. Two such views have been provided; flat and nested JSON views. The flat view replicates the format used by AGIS and hence the view performs a translation from GLUE 2.0 into the custom AGIS model. The nested view provides a GLUE 2.0 based JSON view.

The AGIS usecase was expoited to evaluate the GSR. AGIS relies on a set of collectors to gather information from the multiple information sources. A new collector was created to ingest information from the flat JSON view provided by the GSR and the result was compared to the internal information within AGIS. This comparison initially helped to discover issues with the GSR, however as these were resolved inconsistencies due to missing services remained. On further investigation it was revealed that AGIS does not remove information on decommissioned services. This highlighted the need of a service registry, such as the GOCDB, to define which service *should* be there and conversely, which *should not*.

5. Conclusion

This paper investigated the problem whereby each VO's configuration database interfaces with multiple information systems and proposed the GSR as a potential solution. The GSR aggregates information from the multiple information systems and provides a single interface that can be used to populate the configuration databases with information about Grid services in the WLCG infrastructure. From the VOs perspective it transparently addresses the problem of heterogeneous information systems and generally can help to ensure information consistency and quality. As it is provided as a central service, new capabilities can be easily and quickly added.

During this investigation two questions were raised; why are there errors in the original information sources and why are there no information sources for information that is being communicated out-of-band? In addition, the comparison with AGIS revealed that it does not remove information from decommissioned services and hence there is a need for an authoritative information source that defines which service *should* be in the infrastructure and hence which *should not*.

However, each of the LHC VOs still maintains a configuration database that is the information source for their independent experiment framework. As such the requirements of the configuration database are driven by the experiment framework and hence they are closely coupled. Future work could investigate the feasibility of providing a common Configuration Database to eliminate of the duplication of effort in this area.

References

[1] Sergio Andreozzi, Stephen Burke, Laurence Field, and Maarten Litmaath. GLUE schema version 1.3.

[2] Alexey Anisenkov, Sergey Belov, Alessandro Di Girolamo, Stavro Gayazov, Alexei Klimentov, Danila Oleynik, and Alexander Senchenko. AGIS: the ATLAS grid information system. In *Journal of Physics: Conference Series*, volume 396, page 032006, December 2012.

[3] Ian Bird. Computing for the large hadron collider. *Annual Review of Nuclear and Particle Science*, 61(1):99–118, November 2011.

[4] L. Field, B. Konya, and S. Andreozzi. GLUE specification v. 2.0. Technical Report GFD.147, Open Grid Forum, March 2009.

[5] I. Foster and C. Kesselman. Globus: a metacomputing infrastructure toolkit. *International Journal of High Performance Computing Applications*, 11(2):115–128, June 1997.

[6] Laurence Field and Markus W. Schulz. Grid deployment experiences: The path to a production quality LDAP based grid information system. In *In the Proceedings of 2004 Conference for Computing in High-Energy and Nuclear Physics*, pages 723–726, Interlaken, Switzerland, October 2004.

[7] Gilles Mathieu, Dr Andrew Richards, Dr John Gordon, Cristina Del Cano Novales, Peter Colclough, and Matthew Viljoen. GOCDB, a topology repository for a worldwide grid infrastructure. *Journal of Physics: Conference Series*, 219(6):062021, April 2010.

[8] S Metson, D Bonacorsi, M Dias Ferreira, and R Egeland. SiteDB: marshalling people and resources available to CMS. *Journal of Physics: Conference Series*, 219(7):072044, April 2010.

[9] Ruth Pordes, Don Petravick, Bill Kramer, Doug Olson, Miron Livny, Alain Roy, Paul Avery, Kent Blackburn, Torre Wenaus, Frank Wrthwein, Ian Foster, Rob Gardner, Mike Wilde, Alan Blatecky, John McGee, and Rob Quick. The open science grid. *Journal of Physics: Conference Series*, 78, July 2007.

[10] P. Saiz, L. Aphecetche, P. Buni, R. Piska, J.-E. Revsbech, and V. ego. AliEnALICE environment on the GRID. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 502(2-3):437–440, April 2003.

[11] The Open Science Grid Executive Board. A science driven production cyberinfrastructurethe open science grid. *Journal of Grid Computing*, 9(2):201–218, December 2010.

[12] A Tsaregorodtsev, M Bargiotti, N Brook, A C Ramo, G Castellani, P Charpentier, C Cioffi, J Closier, R G Diaz, G Kuznetsov, Y Y Li, R Nandakumar, S Paterson, R Santinelli, A C Smith, M S Miguelez, and S G Jimenez. DIRAC: a community grid solution. *Journal of Physics: Conference Series*, 119(6):062048, July 2008.