

CRIC: a unified information system for WLCG and beyond

Alexey Anisenkov^{1,2,*}, Julia Andreeva³, Alessandro Di Girolamo³, Panos Paparrigopoulos³, and Aresh Vedae⁴

¹Budker Institute of Nuclear Physics, Novosibirsk, Russia

²Novosibirsk State University, Novosibirsk, Russia

³CERN, Geneva, Switzerland

⁴PIC, Barcelona, Spain

Abstract. The Worldwide LHC Computing Grid (WLCG) is an innovative distributed environment which is deployed through the use of grid computing technologies in order to provide computing and storage resources to the LHC experiments for data processing and physics analysis. Following increasing demands of LHC computing needs toward high luminosity era, the experiments are engaged in an ambitious program to extend the capability of WLCG distributed environment, for instance including opportunistically used resources such as High-Performance Computers (HPCs), cloud platforms and volunteer computers. In order to be effectively used by the LHC experiments, all these diverse distributed resources should be described in detail. This implies easy service discovery of shared physical resources, detailed description of service configurations and experiment-specific data structures is needed. In this contribution, we present a high-level information component of a distributed computing environment, the Computing Resource Information Catalogue (CRIC) which aims to facilitate distributed computing operations for the LHC experiments and consolidate WLCG topology information. In addition, CRIC performs data validation and provides coherent view and topology description to the LHC VOs for service discovery and configuration. CRIC represents the evolution of ATLAS Grid Information System (AGIS) into the common experiment independent high-level information framework. CRIC's mission is to serve not just ATLAS Collaboration needs for the description of the distributed environment but any other virtual organization relying on large scale distributed infrastructure as well as the WLCG on the global scope. The contribution describes CRIC architecture, implementation of data models, collectors, user interfaces, advanced authentication and access control components of the system.

1 Introduction

The Worldwide LHC Computing Grid (WLCG) [1] is a global collaboration of the computing centers, in more than 40 countries, with the main goal to provide a resource to store, distribute, process and analyze data generated by the Large Hadron Collider (LHC).

All physicists participating in the LHC experiments independently of the location of their institution should have access to the LHC data and necessary resources to perform their analysis. The solution for the LHC computing system should be able to store, distribute and

*Alexey.Anisenkov@cern.ch

process an unprecedented amount of LHC data, with transparent access to this data of the distributed community of more than 10 thousand researches. Most of the LHC institutions own large computing resources. Therefore, the computing system for LHC was designed by integrating resources of the computing centers of the scientific institutions participating in the LHC experiments in a single LHC computing service. This design was based on the concept of the computing grid, currently combining resources of more than 170 computing centers in more than 40 countries. It is the world's largest computing grid. It is based on two main grid infrastructures – the European Grid Infrastructure in Europe [2] and Open Science Grid in the US[3]. The WLCG distributed infrastructure successfully supported LHC scientific program for almost 10 years.

Next generation of the LHC experiments - High Luminosity LHC (HL LHC) brings new computing challenges. The system should be able to manage multi-Exabyte of data per year and would require about 20M cores for data processing. In contrast with the first years of LHC data taking, when WLCG hardware resources were more homogeneous, the infrastructure should be able to integrate very heterogeneous resources including HPC, specialized clusters, commercial and non-commercial clouds. Moreover, it implies usage of different types of computing platforms like CPU, GPU and FPGAs. More or less static computing infrastructure is moving towards dynamic one, where elasticity comes in place of static capacity. All those new trends have a strong impact on the WLCG Information System. The primary goal of the information system is to describe the topology and configuration of the WLCG infrastructure. Description of the dynamic and heterogeneous distributed computing environment is a complex task.

2 Overview of the WLCG information sources

There are several low-level information services, data providers and various resource description systems that should be considered by each experiment in order to get a full picture about physical resources description. Some of them are specific for a particular grid infrastructure like EGI or OSG and are being used by all the LHC experiments, while others are specific for a particular experiment.

Information provided by the WLCG information sources is consumed by various clients: among those are testing, monitoring and accounting systems (central as well as experiment-specific ones) workload management and data management systems of the experiments. Depending on the information consumer, various bits of data providers are required. By their nature, these data can be static like service endpoints and site topology description, semi-static, like service capacity or dynamic, (e.g. the number of jobs in the queue). A detailed description of a few information sources is provided in the table 1.

Every component of the WLCG information sources performs a specific function, for example, enables service discovery (GocDB and OIM) or keeps information about site or service capacity and pledges (REBUS). Sometimes data provided by various components are incomplete or contradictory. There is no central place where data can be checked and validated by all the experiments. In addition to the generic data provided by WLCG infrastructure, experiments require dedicated configuration information which can be specific for a given experiment or a particular task performed by the experiment computing system.

Apart from resources description, high-level VO-oriented middleware services and applications also require the diversity of common configurations to be centrally stored and shared. For now, every experiment has to solve all those problems on its' own. All these limitations are considered and covered in the CRIC project to enable enhanced operations for VOs and provide a proper overall description of their topology and computing resources. Also, CRIC

will help integrate configuration and status details about resources, services and the topology of involved computing infrastructures. CRIC is a high-level experiment-oriented but still experiment-independent information middleware for distributed computing.

3 Main concept of the CRIC system

Computing Resource Information Catalogue as a high-level information system is focused to describe the topology of the WLCG infrastructure as well as experiment-specific configuration required to exploit this infrastructure according to the experiments computing models.

As the evolution of the ATLAS Grid Information System (AGIS) [4], CRIC inherits from AGIS the main concept of resource declaration. It aims for clear separation of the description of the resources provided by the distributed sites vs information which is required to use these resources by the experiments. In simple words, it clearly distinguishes between resources ‘provided by’ vs ‘used by’. Why it is so important to ensure this separation? ‘Provided by’ information is mostly topology description of the distributed computing infrastructure which implies listing of the service endpoints, their location, implementation, software versions, etc. This is generic information which is consumed by all experiments and does not depend on their computing models. It is defined by the site administrators and service providers. A ‘Used by’ information defines how experiments use a particular resource and how it is integrated into the experiment’s computing system. This, experiment-specific, information can be defined by the experts of the workload or data management systems of the experiments, operations teams or experiment-support teams at the sites.

Therefore, the separation of ‘provided by’ vs ‘used by’ allows to ensure clean data structures with well-defined responsibilities corresponding to user roles which are translated in corresponding access/update privileges in CRIC. This separation makes possible to design a system which can effectively serve various experiments as well as WLCG central operations. CRIC is an experiment-oriented, but experiment-independent service.

4 Architecture

Design of CRIC has been driven by the experience gained so far in operating ATLAS Distributed Computing environment using AGIS as well as by future requests of WLCG operations and HL LHC challenges. Analysis of the infrastructure topology and configuration data flows as well as collaboration with various communities inside and outside LHC scope allowed to understand synergies and to properly construct CRIC data models.

“Provided by” type of information is described in the core part of CRIC, which is generic and serves all experiments. “Used by” type of info which is mostly experiment-oriented is implemented in the experiment-specific plugins. Such a structure allows customizing functionality required by various experiments in the experiment plugins while performing common tasks, like collecting data from the primary information sources, in CRIC core.

This modular architecture is based on the Django framework [5]. Modularity implies constructing shared building blocks which allow to optimize the development process and to ensure common look and feel of the user interfaces. Data collectors assemble generic information in the core part of CRIC. Many modern tools and technologies (such as Bootstrap, jQuery and Web services) are used for the development of the user interfaces, while for applications, data is exposed via REST APIs which are configurable by filters and different views. The overall architecture is summarized in Figure 1.

Flexibility and extensibility of the system are essential to enable the description of very dynamic and heterogeneous resources and to follow the technology evolution. The goal is

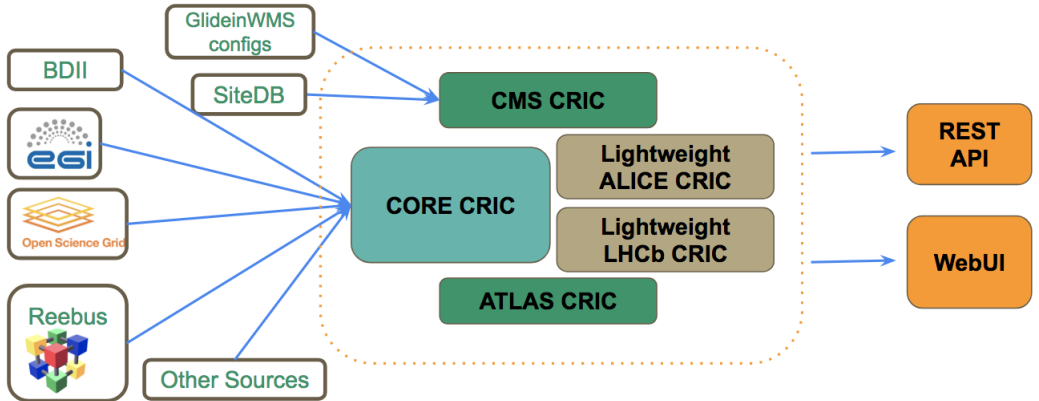


Figure 1. CRIC’s architecture

to provide a modular implementation that allows us to be able to re-use the various components, to extend existing functionality or introduce a new one. CRIC implementation of some generic and experiment-specific concepts will be described in the next sections.

5 Deployment model

CRIC experiment plugins, which include the full description of the experiment computing and storage configuration, will be provided for ATLAS and CMS. In addition to the experiment-specific plugins, WLCG plugin enables tasks required for central operations. Among those tasks is managing pledge information, which is currently performed in REBUS system [6]. REBUS can be retired as soon as WLCG CRIC instance is released into production.

For LHCb and ALICE which currently intend to rely on their internal systems for ‘used-by’ type of information, some limited set of experiment-specific configuration (only the part which is required for central operations) will be provided in the WLCG CRIC instance.

ATLAS and CMS CRIC instances can work autonomously without any dependency on the WLCG CRIC instance. However, in order to minimize operational effort for validation of information coming from various external information sources, ATLAS and CMS CRIC instances will use WLCG CRIC as a primary information source for their core CRIC components. In this scenario, all external data collectors related to core CRIC data structures will be enabled only for the WLCG CRIC instance. There, data will be centrally checked and modified if required and then propagated to the ATLAS and CMS instances.

6 Authentication and Authorization

An information system is a critical service for any kind of computing operations. Support for shared responsibilities between various groups, teams and individual users who are in charge of certain operations or services has to be enabled. Depending on their roles and tasks, various categories of users need to interact with particular CRIC data structures and correspondingly require different privileges for creating, deleting, modifying information in CRIC. Moreover, users must be able to authenticate via various methods (SSL, SSO etc). The system should be able to properly recognize a given user regardless applied authentication

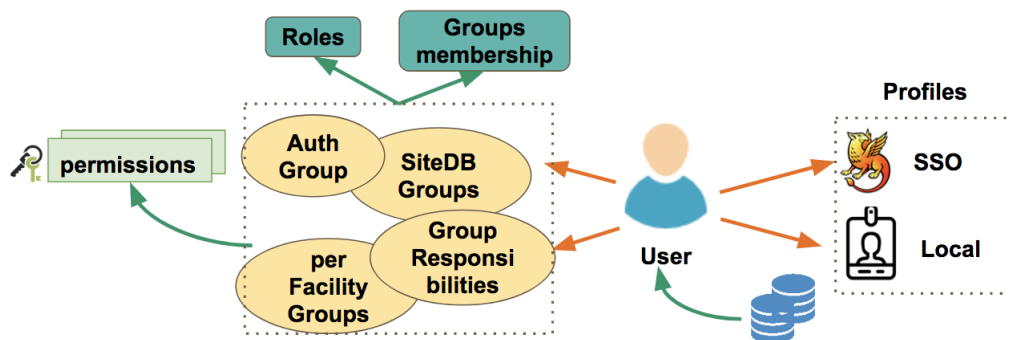


Figure 2. CRIC's Authentication and Authorization architecture

method and to grant him/her appropriate rights. User's and group's roles can be defined in different databases, these could be institutional databases or experiment-specific sources. Those sources are external to CRIC and CRIC should be able to aggregate info from all of them to translate it into a set of corresponding privileges of the relevant user or group. Obviously, managing user information in CRIC should be straightforward and enabled via a dedicated administrative user interface. The Authentication and Authorization architecture is illustrated in Figure 2.

The currently available authentication options in CRIC are SSL using X509 certificate, CERN Single Sign-On (SSO) and local authentication using name/password pair. More authentication methods can be added to CRIC, with minimum effort, using its modular architecture. Every mean of authentication implies creating of user profile(s) which should be associated with a given user. Association of various profiles related to a given user can be performed using different attributes (or their combination), for example, mail address, user DNs, user unique ID in CERN DB or other primary keys.

Based on the operational requirements, CRIC foresees fine-grain authorization, not just on the level of a data model (federation, site, service, user, etc.), but on the level of a particular object instance (storage service at a particular site, VO pledges for a given federation). For example, site admin should be able to edit information only for his/her site. While person responsible for experiment data management might require to modify info limited to storage or transfer services for all sites used by a given experiment.

Authentication and authorization settings are defined by every experiment or by WLCG central operations according to their policies and are configured for a particular CRIC instance (ATLAS, CMS, WLCG). Experiments decide which authentication methods should be used, which authorization rules to be configured and how this information should be exposed. Moreover, CRIC access policy functionality is not limited to interaction with CRIC information. CRIC can also act as a provider of authorization information for various experiment services and computing activities. For example, for CMS CRIC provides information about user roles and group membership in the CMS context.

7 Advanced logging functionality

As has been mentioned above, information in CRIC can originate from various external information sources. It can be also updated manually by the privileged users. In order to

investigate eventual inconsistencies or errors in the CRIC data, advanced logging is mandatory. CRIC provides advanced logging functionality on the level of a given object instance and any object instances related to the one in question. Whenever any object instance in CRIC is modified, full list of changes including old values is being recorded. Using log records it is possible to find out who, when and how interacted with a given object instance. Another important part of CRIC logging is information about user access. Logging data is accessible via the table view of the CRIC administrative user interface.

8 CRIC data models

CRIC data models can be divided into two categories: core and experiment-specific ones. Core data models describe WLCG infrastructure and its various components while experiment-specific models describe how different VOs are using these components. Core components could be: resource center sites, federations, storage services, experiment sites etc. while job submission or data transfer configuration are clearly concepts that heavily depend their definition on the various systems used by experiments.

Designing CRIC data models, the developers tried to identify commonalities between various experiments in order to address common problems, in a common way. Doing that they made sure that experiments can benefit of handling and validating their shared data structures centrally.

Below, some examples of implementation of core and experiment-specific data models are described.

8.1 Common concepts. Storage service example.

The complexity of storage services is explained by multiple aspects, which this definition should take into account. On the one hand, a particular storage resource is defined by the storage space capacity which it provides. Also, the organization of this capacity in terms of shares for various types of data or computing activities must be taken into account. On the other hand, a storage service is defined by protocols which are able to perform different data operations like reading, writing and deleting data files. Moreover, the efficiency of one or another protocol for a particular operation can depend on many factors. Linking of a protocol with a data operation (reading, writing, deletion, others) and particular storage share is implemented via 'capability' data model.

'Capability' preference is normally defined for a given computing activity e.g. staging out result files after the end of a job or remote reading/saving log files. This preference can also depend on the location of the computing resource where the job is running, its' closeness to a given storage service, network quality, etc. Some of the parameters which have to be considered for storage service definition are generic (how storage space is shared, which protocols are enabled) and belong to a core part. Some are coupled with experiment data management and workload management systems and have to be handled in the experiment plugins.

CRIC introduced a unified description of the storage services, in the core part, which allows describing multiple protocols and storage shares, and then linking them together through a 'capability' concept. Object store technology has been also implemented in CRIC as a new type of service resource data model.

8.2 Experiment-specific concepts. Example of the CMS ‘facility’ concept implementation.

The primary goal of the experiment-specific plugins is to provide configuration required by the workload and data management systems of the experiments in order to exploit the WLCG resources. Plugins should also describe the organization of the resources following the specific needs and internal policies of the experiments. There are many commonalities between the experiments and therefore, the majority of data models even in the experiment-specific plugins have quite a lot of common structures. At the same time, there are concepts which have been implemented for a particular experiment. ‘CMS facility’ is an example of such a concept.

CMS facility combines resources which belong to a single administrative domain. These resources can be hosted by different sites, however, from the experiment perspective, they are managed together. This concept introduces an additional level of hierarchy in the experiment infrastructure: facility-site-service-resource vs site-service-resource.

9 Status and plans

9.1 CMS CRIC

The first implementation of the CMS CRIC has been deployed in production early Autumn 2018. This version enables all functionality which has been provided by the CMS SiteDB service. After integration of all CMS applications with CMS CRIC, SiteDB will retire. Further extending of functionality aims to facilitate CMS computing operations both for data processing and data transfer. This work is performed in close collaboration with the CMS computing community.

9.2 ATLAS CRIC

ATLAS was the first experiment to implement an information system to host different static and dynamic data needed to configure and operate the ATLAS Distributed Computing (ADC) [7] systems and services. Gradual migration of AGIS into corresponding ATLAS CRIC plugin is foreseen during next year.

9.3 WLCG CRIC

WLCG CRIC is a central CRIC instance which satisfies the needs of the WLCG central operations. It should provide a complete description of the topology and generic configuration of the WLCG resources used by all four LHC experiments. The implementation of following functional components are considered as middle term plans for the WLCG CRIC plugin:

- generate the so-called VOfeed XML files required for WLCG test submission system for LHCb and ALICE experiments. VOfeeds for ATLAS and CMS will be generated from ATLAS and CMS CRIC instances.
- provide REBUS functionality including pledge management
- provide the description of the WLCG storage services including storage shares for WLCG Storage Space Accounting System

Table 1. Examples of information sources used by CRIC

Name	Description
GOCDDB	GOCDDB is a central registry to record information about the topology of an e-Infrastructure. This includes entities such as Operations Centers, Resource Centers, service endpoints and their downtimes, contact information and roles of users responsible for operations at different levels. The service enforces a number of business rules and defines different grouping mechanisms and object-tagging for the purposes of fine-grained resource filtering.
OIM	The OIM service is the topology management system for the OSG. It holds information about people and resources involved in the OSG.
REBUS	REBUS is the Resource, Balance and Usage website for the whole of WLCG, including topology information, resource pledges, and installed capacities.

9.4 CRIC beyond LHC

CRIC is a suitable framework to also describe the computing topology and infrastructure of non-WLCG experiments, like for instance COMPASS at CERN SPS. COMPASS computing model relies on high-level frameworks which are used also from some WLCG experiments, like the workload management system PanDA. Thus the similarity for what concern the topology description allowed us to easily bootstrap a prototype CRIC server to ease the COMPASS computing topology description. CRIC plugin for COMPASS is for now only compute based, it covers basic functionality required by COMPASS production infrastructure, but it might be extended as needed. The COMPASS CRIC plugin is currently ready for final migration and planned to be integrated into the production COMPASS Computing infrastructure once the validation phase will be finished.

References

- [1] I. Bird, Computing for the Large Hadron Collider, *Annual Review of Nuclear and Particle Science*, **61**, 99-118 (2011)
- [2] G. Mathieu, A. Richards, J. Gordon, C.D.C. Novales, P. Colclough, and M. Viljoen. GOCDDB, a topology repository for a worldwide grid infrastructure, *J. Phys.: Conf. Ser.* **219**, 062021 (2010)
- [3] OIM: <https://oim.opensciencegrid.org>
- [4] A. Anisenkov, A.D. Girolamo and M.A. Pradillo AGIS: Integration of new technologies used in ATLAS Distributed Computing *J. Phys.: Conf. Ser.* **898**, 92023 (2017)
- [5] Django Project: <https://www.djangoproject.com/>
- [6] REBUS: <https://gstat-wlcg.cern.ch/apps/topology/>
- [7] S. Campana, et al. ATLAS Distributed Computing in LHC Run2, *J. Phys.: Conf. Ser.* **663**, 032004 (2015)