

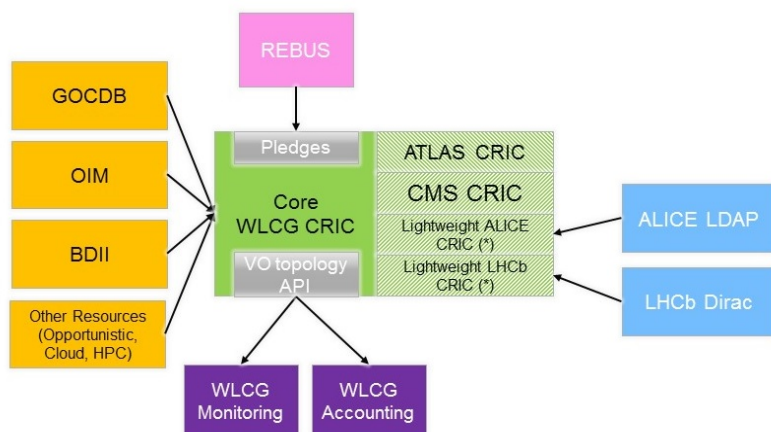
Table of Contents

Core CRIC.....	1
Introduction.....	1
Functionality.....	1
Information Catalogue.....	1
Information Validation.....	2
History of changes.....	2
Interfaces.....	2
Roles.....	3
Information Sources.....	3
Running CRIC as service at CERN.....	3
State of CRIC Instances.....	4

Core CRIC

Introduction

The Computing Resource Information Catalogue (CRIC) is the new WLCG Information System. The CRIC architecture for WLCG will rely on a core part on one side and on experiment specific parts on the other side. Experiment specific parts include experiment topology and specific data structures that are tightly coupled with experiment frameworks. The core part is basically an information catalogue for all WLCG resources and it gets information about existing resources from other existing tools like GOCDB and OIM. The functionalities of the core part are described below.



(*) Maintained by WLCG to store very simple experiment topology information (i.e. experiment names)

Functionality

Information Catalogue

- **Description:** CRIC should contain a list of sites which will contain a list of services with associated attributes that would allow for service discovery.
- **Implementation:**
 - ◆ List of Federations
 - ◇ Federation Name
 - ◇ List of Sites
 - ◇ Contact email
 - ◇ Pledges (per year)
 - Tape
 - Disk
 - CPU
 - ◆ List of sites:
 - ◇ Site Name
 - ◇ Experiment Site Name
 - ◇ Country
 - ◇ Longitude, Latitude
 - ◇ Time Zone
 - ◇ Contact email
 - ◇ Status (Enabled, Disabled)
 - ◇ State (Active, Downtime)
 - ◇ List of Services
 - ◆ List of services and associated specific attributes:

- ◇ Computing
 - Job Manager
 - Capacity
 - Queues
 - Max Wall Clock Time
 - Max memory
- ◇ Storage
 - Protocol
 - Implementation
 - Capacity
- ◇ PerfSonar
 - Flavour
- ◇ Squid
- ◆ List of comon attributes:
 - ◇ Service Name
 - ◇ Service Endpoint
 - ◇ Service Status (Enabled, Disabled)
 - ◇ Service State (Active, Downtime)
- **Status:**
 - ◆ Identify which of these services/attributes already exist in CRIC
 - ◆ Check with IS TF whether other services/attributes are missing
 - ◆ Whatever experiment CRICs need, may impact core CRIC as it is the one collecting the services/attributes from the information sources. Core CRIC should be flexible to modify these definitions at any time, extending them with new attributes if necessary.

Information Validation

- **Description:** Information should be validated in an automatic way before entering into core CRIC
- **Implementation:** The validation criteria needs to be defined and implemented into core CRIC. The information workflow needs to be defined.
- **Status:**
 - ◆ Identify which validation criteria already exists in CRIC
 - ◆ Discuss which other criteria is missing
 - ◆ Agree on the information validation workflow, deciding what CRIC will do if information fails the validation step

History of changes

- **Description:** Log information (when, how, by whom) should be entered
- **Implementation:** each associated change in the system done automatically or manually should be logged and the logs should be browseable from the webUI
- **Status:**
 - ◆ This feature already exists in CRIC but not for all objects. To be reviewed with developers and extend to cover core CRIC use cases when necessary.

Interfaces

- **Description:** Information from CRIC will be accessible through a webUI and also through an API
- **Implementation:** a REST API should be available to be able to get information from CRIC
- **Status:**
 - ◆ Available, however, writeable API only available for experiment CRIC. To be implemented also for core CRIC.

Roles

- **Description:** different roles can interact with core CRIC
 - ◆ Site admin: is allowed to manually modify information about its site and associated services.
 - ◆ Super admin: is allowed to manually modify any information in CRIC. Restricted to WLCG Operations team.
- **Implementation:** logging to CRIC is possible using user certificates
- **Status:**
 - ◆ Available but site admins have privileges per "object" and not per "site", this means that if they are allow to modify a CE or a SE, they can do it for all the sites. This logic needs to be revisited to be able to have privileges at site level.

Information Sources

- **Description:** CRIC will collect information about resources for WLCG from different sources:
 - ◆ Existing tools from associated infrastructure projects: GOCDB, OIM, BDII
 - ◆ Opportunistic resources or HPC resources through manual input or any automatic way to collect the necessary information required by CRIC
- **Implementation:** collectors extracting information from the required information source should be implemented and run within CRIC. It should be easy to add or remove collectors according to WLCG needs
- **Status:**
 - ◆ Available for REBUS, GOCDB, OIM and BDII. However, existing collectors will need to be modified to query the needed attributes and the envisaged changes in the information sources. See CRIC Information Sources Presentation for more details.

Running CRIC as service at CERN

The following requirements need to be met in order to run CRIC as a CERN service:

- CRIC software should be packaged by RPM and should be buildable on via Koji.
- CRIC service should be run as a load-balanced service in a multi-master way (ideally, rather than failover) - i.e. clients can talk to any node.
- CRIC service should be sessionless, or at least sessions should be shared across all nodes - otherwise load-balancing on SSL is hard.
- CRIC should not have state on local disk. All state information should be in a database, provided by CERN DBOD.
- CRIC should allow for termination of SSL at Apache layer, with identity passed to web-app via mod_ssl
- CRIC should provide logging information, in particular what operation called with which parameters is needed. Also IP address / SSL identity. If multiple components / classes, use a common request identifier to tie all related logs together. Reasonable use of log4* verbosity levels (DEBUG/ERROR etc).
- CRIC should provide operational documentation, including what processes need to be running and expected system utilisation.
- CRIC should run in CentOS7 as the operating system.
- CRIC application code should be in an open CERN version control system (ideally Gitlab), visible to operations team
- An escalation path to development team should be defined in case of operations problems
- CRIC should provide a structured configuration file, ideally of the broken-out cric.d/ directory style if multiple parts / multiple services.
- Puppet module will be provided by CERN IT.

-- MariaALANDESPRADILLO - 2016-09-09

State of CRIC Instances

Name	Plugin	Type	LoadBalanced	Puppet environment	Puppet hostgroup
atlas-cric-1.cern.ch					
atlas-cric-2.cern.ch					
cms-cric-0.cern.ch					
cms-cric-1.cern.ch					
cms-cric-2.cern.ch					
cms-cric-3.cern.ch					
cms-cric-4.cern.ch					
cms-cric-dev-1.cern.ch					
cms-cric-dev-2.cern.ch					
cms-cric-dev-3.cern.ch					
cms-cric-dev-4.cern.ch					
cms-cric-preprod-1.cern.ch					
cms-cric-preprod-2.cern.ch					
cms-cric-preprod-4.cern.ch					
wlcg-cric-1.cern.ch					
wlcg-cric-2.cern.ch					
wlcg-cric-3.cern.ch					
wlcg-cric-dev-1.cern.ch					
wlcg-cric-dev-2.cern.ch					
doma-cric-1.cern.ch					
escape-cric.cern.ch					
compass-cric-1.cern.ch					
compass-cric-2.cern.ch					

This topic: EGEE > CoreCRIC

Topic revision: r6 - 2020-04-06 - PanosPaparrigopoulos



Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Ask a support question or Send feedback