A unified topology system for a large scale, heterogeneous and dynamic computing infrastructure

Alexey Anisenkov (BINP)

on behalf of CRIC team

CHEP 2019, Australia, Adelaida, 7 Nov 2019
CRIC: a high-level information middleware to configure computing environment and describes resources as VOs need

Resources description

- GOCDB
- OIM
- OpenLDAP
- BDII
- REBUS
- Other sources
  - Low-level infosys
  - Data providers
  - Service discovery

Experiment applications

- Rucio
- PanDA
- GlideinWMS
- WLCG squids
- Phedex
- And many others…
CRIC mission: link Resources & VOs together

➢ Consolidate (WLCG)** topology information of a large scale computing infrastructure
➢ Facilitate distributed computing operations for (LHC)** Experiments

Key functional capabilities of CRIC information concept:

■ Built-in information Model(s) for Resource descriptions
■ Clear distinction between (physical) resources *provided by* (Sites) and how they are *used by* Experiment(s)
■ Built-in aggregation and validation of data collected from various low-level information providers (sources)
■ Ability to extend and complement Information Model(s) with Experiment(s) specific data structures. Plugin based approach allows VOs to inherit shared functionality and address custom requirements
■ Flexibility to address technology evolution and changes in the VO Computing models and applications
■ Experiment-oriented but still Experiment-independent IS framework

* CRIC is not coupled to WLCG, it could be applied beyond LHC experiments
CRIC Architecture: examples of shared features out of the box

- **Plugin based**: VO can configure default behaviour
- **Base implementation** for the Resource/Topology description
- **REST API** data export (filters, presets, various output formats)
- **Shared engine/widgets** for WebUI (downtime calendars, table view, tree view, inline editors, etc.)
- **Enhanced Authorization** (CERN SSO, SSL, passwd based; local accounts)
- **Enhanced Authentication** (instance specific permissions, groups, roles, etc., map permissions to e-groups, fetch info from ext sources)
- **Detailed History of Changes** (who, when, how interacted with object)

VOs can use shared WLCG CRIC instance as the master source or optionally fetch data directly from low-level info providers.
Few Implementation details: Web2.0 based

- Apache/WSGI + Python + Django framework as server backend
- Independent database backends (Oracle, MySQL, postgres, etc)
- Web Services technologies (REST API, WebUI, widgets)
- Bootstrap framework as HTML/CSS/JS client frontend (responsive, interactive, mobile-friendly)
- Client AJAX, JQuery plugins, own shared widgets (datatables, treeview, calendar views, inline editors ..)
- Plugin based approach (shareable applications in “core” re-used by many components)
In production for CMS Community since the end of 2018

- As a replacement of CMS SiteDB (currently in read-only mode, to be retired soon)
- Basic topology description (sites, processing unit names translation, etc)
- User Groups, roles and privileges used by CMS services (CRAB3, Phedex, WMAgent, cmsweb, etc)
- CMS User description required by CMS applications
- Job submission configuration (GlideinWMS)
  (see the talk “Exploiting CRIC to streamline the configuration management of glideinWMS factories for CMS support” by James Letts)
- Various REST API export, including integrated CORE functionality and CMS specific objects of Information Model (CMSSite, ComputeUnit, ComputeResources, GlideInEntry, ..)
  (ongoing, step by step integration into production

Alexey Anisenkov, CHEP-2019

http://cms-cric.cern.ch
WLCG CRIC

Dedicated CRIC instance for central WLCG operations

➢ Single entry point for complete WLCG topology description and service configurations for the all 4 LHC experiments

➢ Main info provider for cross-experiment tools: WLCG Accounting, Monitoring, Service Availability, Test submission systems,..

➢ Federation Pledges management and topology export (REBUS replacement)

➢ VOFeed XML generation (ALICE, LHCb)

➢ Management of VO Pledge Requirements

➢ Tracking of various Task Forces and Migration activities

➢ WLCG Accounting data validation (storage space and CPU capacity from WSSA)

➢ WLCG Accounting Report generation interfaces

➢ In production since Summer 2019. Site Admins started to validate and update information

http://wlcg-cric.cern.ch
ATLAS CRIC (ongoing)

In fact **CRIC** is the evolution of **AGIS** (ATLAS Grid Information system) - completely refactored code, inherits all AGIS features

Moving from AGIS to ATLAS CRIC has started

- All API export should be implemented in backward compatible way (no significant changes expected for exist **ATLAS** clients)
- **CRIC** will sync data from **AGIS** and advertise own API for the integration period (AGIS is still a master for given data block once “edit” functionality is migrated into CRIC)
- Redirect back to **AGIS** for not covered functionality
- Once a functional block is migrated to CRIC, disable modification in AGIS and redirect to CRIC for the migrated part

**Tentative Plan for the migration:**
- Basic topology, Blacklisting API, PanDA SW releases -- Oct/Nov 2019
- Moving slightly dependent data and settings -- Nov 2019
- Complete **PQ+RSE** management within CRIC -- Beginning of 2020

Alexey Anisenkov, CHEP-2019

http://atlas-cric.cern.ch/
DOMA CRIC

Dedicated CRIC instance for Rucio TPC tests and DOMA related activities

➢ Provides **Storage description** and related data structures for Experiment agnostic DOMA Rucio instance

➢ In close cooperation with Rucio experts to polish RSE related models and CRIC interfaces in order to provide appropriate API export for Rucio clients (probes)

➢ Rucio team has tested RSE configuration coming from DOMA CRIC with Rucio ESCAPE instance. All works well. Look forward for the next integration steps.

➢ Once CRIC and Rucio integration will be completely tested and evaluated, developed CRIC models and interfaces within DOMA CRIC will be shared with other plugins (ATLAS CRIC, CMS CRIC)

http://escape-cric.cern.ch

Alexey Anisenkov, CHEP-2019
COMPASS CRIC

Dedicated CRIC instance for COMPASS Experiment at CERN SPS

➢ COMPASS Distributed Computing Environment is very similar to one used by ATLAS (Computing Model, PanDA WMS, PanDA Pilot, ..)

➢ First COMPASS CRIC prototype has been evaluated, tested and initially integrated into COMPASS environment.

➢ Today COMPASS upgrades components of WMS (PanDA Pilot, Harvester migration) that in particular requires updates from COMPASS CRIC side

➢ Current status: integration step, to be released soon.

http://compass-cric.cern.ch

Alexey Anisenkov, CHEP-2019
Lightweight CRIC plugin (ongoing)

Universal topology description of generic distributed infrastructure

- Enables all CRIC features but with simplified Computing Model description
- Basic models for **Compute** and **Storage** Resources
  (StorageUnit+StorageResource, ComputeUnit+ComputeResource)
- Completely CERN-independent
- Standalone distribution (via images), not coupled to CERN Openstack deployment infrastructure
- Suitable for small VOs or Experiments beyond LHC

Requested by Experiments at JINR (NICA and beyond) as the Information component for the Unified Resource Management System
Collaborative support model

➢ CRIC project is a joint effort between CERN (IT-WLCG and the Experiments), Novosibirsk (BINP), DUBNA (JINR) and other institutions.

➢ CRIC core services are supported by the CRIC team

➢ For the CRIC plugins we expect active participation from the Experiments mainly for functional requirements gathering, light developments and polishing of affected experiment-specific features

➢ Collaborators from the Experiments are welcome!

  ○ Developments, use cases gathering, sustainable strategy planning...

  ○ Please join and get involved to deliver together powerful and sustainable experiment-oriented CRIC tools!

https://gitlab.cern.ch/cric/
CRIC offers a common framework describing LHC Computing infrastructure with also an advanced functionality enabled to describe all necessary Experiment-specific configurations.

➢ Ready to use now. Let’s play and/or contribute.
Thank you for your attention!

Backup slides

- [http://cms-cric.cern.ch](http://cms-cric.cern.ch) (CMS-CRIC)
- [http://cms-cric-docs.web.cern.ch](http://cms-cric-docs.web.cern.ch) (CMS-CRIC documentation)
- [http://wlcg-cric.cern.ch](http://wlcg-cric.cern.ch) (WLCG-CRIC)
- [http://atlas-cric.cern.ch](http://atlas-cric.cern.ch) (ATLAS-CRIC testbed)
- [http://escape-cric.cern.ch](http://escape-cric.cern.ch) (DOMA-CRIC)
- [http://compass-cric.cern.ch](http://compass-cric.cern.ch) (COMPASS-CRIC)
LHC Experiments rely on heterogeneous distributed computing

- variety of computing resources involved

- variety of infrastructures and middleware providers

Distributed Computing Environment (Resources)
Each **Community** uses and describes **Resources** in its own way

- **Computing Models** are similar but still have different implementation
- Various high level **VO-specific frameworks** & middleware services (e.g. for Data and Workflow management)
- **Cross experiments applications** (monitoring, accounting, testing frameworks, resource usage descriptors, etc)

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
Authorization and Authentication (A&A)

➢ CRIC supports enhanced **Access controls** and user Group management
➢ Several **Authentication** methods enabled (SSO, SSL, Proxy cert, passwd)
➢ Flexible utilisation of **Permissions**, **Roles** and **Groups** at various levels
➢ Fine grained Auth checks at various levels (object, model, restricted instances, global permissions)
➢ Ability to bootstrap DB (User info) from whatever external source (CERN DB, Experiment DBs, config files, e-groups, VOMS roles, etc)

Each Experiment could configure own **Data access policies!**
Example of A&A use-cases for different VOs

- **CMS** uses CRIC not only to define access rights within the system, but also to control user privileges for **CMS applications** (CRAB, WMAgent, Phedex, etc...). Relies on CERN SSO and local authentication.

- **CMS** enables instance specific permission checks (who is allowed to manage specific object(s), e.g. update CERN-PROD site, and/or all affected resources)

- **ATLAS** considers simplified Auth concept based on user’s DNs

---

Experiment decides what elements should be used out of the CRIC box to implement own policies and follow own workflow.
Data export availability

➢ Thanks to LB deployment (and appropriate DB backend) CRIC is able to sustain with high API load coming directly from worker nodes

➢ For highest reliability, VO can enable dedicated CacheCheckers application to automatically cache and dump required API JSONs to special location (e.g. CERN EOS box)

➢ Experts can subscribe to email notifications in case of JSON changes

| Cache name       | URL                                               | Cache                                                                 | Stable cache                     | Zip | Link | Email | Emails | Period |
|------------------|---------------------------------------------------|                                                                     |                                  |     |      |       |        |        |
| accounts.group   | https://cms-cric.cern.ch/api/accounts/group/query/list?json | /eos/user/c/cric/jsoncache /cms/accounts.group/accounts.group_list_%Y_%m_%d_%H%M%S_ETAG_VALUE.json | /eos/user/c/cric/jsoncache /cms/accounts.group_list.json |     | ✓    | ✓     | ✓      | 15     |
| accounts.user_roles | https://cms-cric.cern.ch/api/accounts/user/query/list?json&preset=roles | /eos/user/c/cric/jsoncache /cms/accounts/user/accounts.user_list_roles_%Y_%m_%d_%H%M%S_ETAG_VALUE.json | /eos/user/c/cric/jsoncache /cms/accounts.user_list_roles.json |     | ✓    | ✓     | ✓      | 15     |
| cms.site         | https://cms-cric.cern.ch/api/cms/site/query/list?json | /eos/user/c/cric/jsoncache /cms/cms/site/cms.site_list_%Y_%m_%d_%H%M%S_ETAG_VALUE.json | /eos/user/c/cric/jsoncache /cms/cms.site_list.json |     | ✓    | ✓     | ✓      | 15     |

Latest (stable) caches can be further mirrored to CVMFS (e.g. used by ATLAS) as (failover, cache) access to topology data from worker nodes

Alexey Anisenkov, CHEP-2019
Recent core developments (tableviews)

Inline editor for bulk updates
Flexible table view visualization

➢ Sometimes it’s helpful to provide required set of parameters as a dedicated view (with ability to dynamically expose on fly all affected/predefined fields)

➢ Activate inline modification for specific fields

Update required fields for several objects,
Bulk save

Customize visualization on fly, prepare custom data view for share and distribution
Recent core developments (request user privileges)

**ADMIN Privileges Request/approval wizard**

- VO can customize supported list of ADMIN Groups family (Global ADMIN, Site Admin, Federation Admin, etc..)

Clicky-Clicky wizard to request privileges by user

ADMIN decides/grants only required perms. Email notification sent on request/approval steps
Ongoing core developments

Moving to use-case oriented approach of updating data

➢ Classical approach assumes to update some set of parameters for specific model of Information schema (="configure these variables for these resources")

➢ In reality, typical use-case oriented (workflow) update
  o involves modification of some attributes of several affected models depending on user input
  o requires extra validation and conditional modification
  o user usually does not know which parameter is affected
    (for example in ATLAS, PandaQueue model has ~ 100 parameters)

➢ CRIC will provide a wizard-like workflow forms to process specific use-case for data modification

➢ Example: I want to enable remote-io mode for jobs:
  ■ at which site?
    ● for which type of jobs? (ANALY, PROD)
      o for which input storage?
    ■ for which type of access? (LAN/WAN), ..