Computing Resources Information Catalog

... the evolution of the ATLAS Grid Information System (AGIS) for other Collaborations

the CRIC dev team:
A. Anisenkov (BINP); A. Vedaee (pic); B. Konya (LUND);
S. Di Guida, J. Andreeva, A. Di Girolamo (CERN)
Worldwide LHC Computing Grid

- 200 Computing Centers
- 500 PB of storage (disk + tape)
- 500k job slots
- 5k+ physicists (users)
WLCG: a bit more...

4 Experiments
Different computing models
Different frameworks for Data and Workflow Management
... and more

Variety of grid middleware
Different infrastructures & middleware providers
... and more

Variety of computing resources

- “Standard” grid sites
- Cloud resources
- Opportunistic High-Performance Computing
Information component

Variety of grid computing resources

Different computing models

Different frameworks for Data and Workflow Management

Variety of grid middleware

Different infrastructures & middleware providers

Variety of computing resources

§ “Standard” grid sites

§ Cloud resources

§ Opportunistic High-Performance Computing

Information System:

link together the computing resources,
describe their topology and
automate services configs
Information Systems: a big world

N.b.: Information System → multiple interpretations

GOCDB

Grid Configuration Database

OIM

OSG Information Management System

Open LDAP BDII

Berkeley Database Information Index

REBUS

WLCG REsource, Balance & USage

Experiments (& WLCG) Resources Configuration systems

others

N.b.: Information System → multiple interpretations
Experiments frameworks: a big world

- ATLAS Workload Management System
- CMS Data Management System (Phedex)
- ATLAS Data Management System (Rucio)
- WLCG Squid monitoring
- Monitoring tools
- Testing systems (HammerCloud)
- Pilots, AutoPilot Factories
- BigPanda Monitors
- Software Installation systems
Resources & Services: Gluing the two “big worlds” together

- ATLAS Grid Information System (AGIS) is the central information system for ATLAS:
  - connects Resources and Experiments frameworks together for the ATLAS experiment.
  
AGIS integrates configuration and status information about resources, services and topology of the whole computing infrastructure used by ATLAS Distributed Computing

- CRIC is the AGIS evolution
  - Next generation system
  - Non-experiment specific, fitting the needs of multiple collaborations
Information system: a key component of Computing

It does not really matter how big or small an Experiment/Collaboration is

- An information component/service/system is needed in order to effectively operate and configure computing system:
  - Proper description of physical Computing resources
  - Proper description of Experiment’s Computing Model, its topology and implication to high level applications
  - Integration of configuration and settings of high-level applications and services involved into Distributed Computing
  - Central operation entry point (WebUI portal) for end-users
  - Central data provider (REST-full API) for applications
  - Information protection, authorization, input data validation, tracking history of changes, rollback functionality .. user-oriented views ..
CRIC is the middleware designed to describe the topology of the Computing models, providing unified description of resources and services used by Experiment applications.
CRIC: Resources & Experiments

- Clear distinction between resources provided by (Sites) and resources used by (Experiments)

- Establish relationship between resources to Experiment objects

Providing an abstraction layer from the physical resources the system allows the Experiment to define their own real organization of the resources, experiment specific topology and own services structures.
CRIC: implementation details 1/2

- Django based

- Module implementation:
  - plugin based approach
  - Shareable applications that could be reused in separate projects, so that we can make common “core” and attach various plugins into it

- MVC based Django implementation (Model-View-Controller)
  - Model: database layer mapped to python objects (using built-in Django ORM - Object Relational Mapping) -- no direct dependency to specific database backed
  - View: user interface layer split into business logic code (called views in Django) and HTML page templates: views render html templates
  - Controller: a set of custom middlewares and django cores enabled the processing of user request and finally evaluating attached view function to handle user response
CRIC: implementation details 2/2

- Client-server architecture, 2 independent services:
  - API service (REST-full GET export: JSON, etc; POST update) -- mainly used to export data, bulk updates and operate data programmatically
  - WebUI portal (AJAX support, JQuery widgets, based on Bootstrap, etc) -- mainly used to navigate, browse and declare objects

- Extensibility: the services could be hosted on different nodes, LB supports (same persistent DB instance)

- Various integrated collectors run by crons to populate/sync DB from ext. sources

- The system supports information protection (Federated Identity through SSO):
  - Authorization is required to modify data: Groups, Roles and list of specific permissions could be directly associated to users
Plugin based: Core and Experiments

**Core CRIC**
- *Resources “Provided by”*
- Single entry point for WLCG topology and service configuration
- Consumes information from available information sources

**Experiments CRICs**
- *Resources “Used by”*
- Describes experiment topology
- Uses core CRIC and adds extra info needed by experiment operations and workflows

**Lightweight CRIC**
- Experiment site names and WLCG site names mapping and information on which resources are used by the experiment

(*) Maintained by WLCG to store very simple experiment topology information (i.e. experiment names)
Storages - pre-CRIC attempts

... Either too little
Storages - pre-CRIC attempts

... Either too much!!
CRIC approach

- Describe only those system characteristics that are really needed
- Dynamic - static separation
- Keep simple things simple
- BUT Avoid oversimplification

LAGOM: Not too little, not too much. Just right. (Swedish)
Storages in CRIC: details of objects

- **StorageService**: represents the entire storage system that makes its Shares, defined within a DataStore, available via Endpoints. Static information.
  - Attributes: name, id, servicetype, implementation, capabilities, qualitylevel, storagecapacity ...

- **StorageShare**: represents a logical storage area, a part of the storage capacity, allocated on DataStore(s) for a specific user group or use case. May contain dynamic information (e.g. space usage).
  - Attributes: name, id, policyrules, path, assignedendpoints, servingstate, accessmode, maximumlatency, retentionpolicy, defaultlifetime, maximumlifetime, expirationmode, totalsize, usedsize, numberoffiles, message

- **StorageEndpoint**: represents the network interface to the storage system that may be contacted to manage and access data stored in the StorageShare(s) of DataStore(s). May contain dynamic information (e.g. healthstate or servingstate).
  - Attributes: name, id, endpointurl, assignedshares, interfacetype, interfaceversion, capabilities, qualitylevel, servingstate, healthstate, message

- **DataStore**: describes a homogeneous instance of a physical storage extent (e.g. a tape or a disk server). Static information.
  - Attributes: name, id, datastore-type, latency, totalsize, vendor, message
Distributed Data Management Endpoint diagram

Service world

Storage Service

StorageShare aka StorageArea (SpaceTokens, paths …)

Experiment world

Many more attributes:
- Disk/Tape
- path for HTTP, path for SRM, …
- Which protocol for which activity, e.g.:
  - HTTP for logs and user download
  - gridFTP for third party transfer
- Physics Groups
- Topology (Tier level, closeness)
- …

Storage Endpoint, aka Protocols

DDM Endpoint

http

xrootd

srm
Successful experience in ATLAS Computing with AGIS motivated and inspired WLCG community to evolve and consider CRIC as a base platform for WLCG Information configuration system

- **CRIC**: architecture and core functionalities focused to cover Experiments/collaborations needs.
  - common framework for the description of all WLCG resources and consistent interfaces for the clients from several collaborations.
  - possibility to extend the system and implement experiment specific CRIC plugins.
  - Easy and light to integrate by Collaborations thanks to well defined (REST) interfaces.

- Part of CRIC tools can be shared and centrally managed to minimize support efforts for smaller collaborations
  - Built-in lightweight version of experiment CRIC could be provided for small collaborations
We've already got one. A hidden microphone in the staff coffee area.

GET ALL THE INFORMATION YOU CAN, WE'LL THINK OF A USE FOR IT LATER.