CRIC

Computing Resource Information Catalog

The ATLAS Grid Information System (AGIS) evolution for other communities

Alexey Anisenkov *(BINP)*

on behalf of CRIC team

NEC 2017, Montenegro, 26 Sept 2017

WLCG
Worldwide LHC Computing Grid

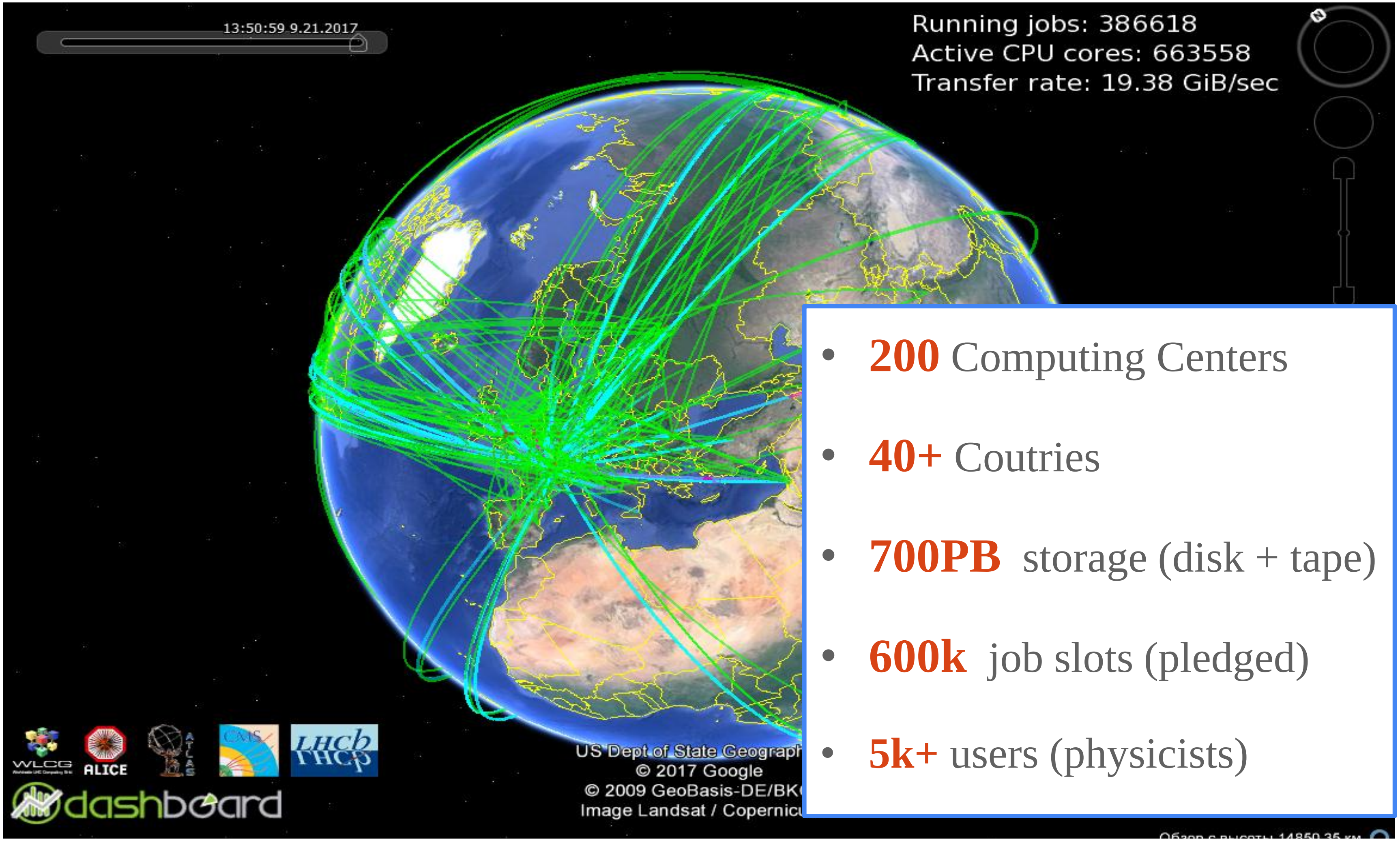
ATLAS EXPERIMENT

CERN

Outline

- The Role of Information system in Distributed Computing Environment (Why do we need IS?)
- What is AGIS/CRIC?
- CRIC main features
- CRIC plugins for other Collaborations

Distributed Computing Environment: Worldwide LHC Computing Grid (WLCG)



13:50:59 9.21.2017

Running jobs: 386618
Active CPU cores: 663558
Transfer rate: 19.38 GiB/sec

WLCG
ALICE
ATLAS
CMS
LHCb
dashboard
US Dept of State Geograph
© 2017 Google
© 2009 GeoBasis-DE/BK
Image Landsat / Copernicu
Обзор с высоты 14850.35 км

- **200** Computing Centers
- **40+** Coutries
- **700PB** storage (disk + tape)
- **600k** job slots (pledged)
- **5k+** users (physicists)

International Collaboration of Computing centers located across the world to distribute and analyse LHC data

WLCG: Challenges

4 (main) Experiments

- Different Computing models
- Different high level frameworks
for Data and Workflow management systems

ATLAS EXPERIMENT   CMS   LHCb   ALICE   ...

WLCG: Challenges

Variety of GRID Technologies

- Different infrastructures

European Grid Infrastructure

Open Science Grid

NORDUGRID

Grid Solution for Wide Area Computing and Data Handling

- Different middleware providers

UNICORE

HTCondor High Throughput Computing

ARC

gLite Lightweight Middleware for Grid Computing

globus toolkit

WLCG: Challenges

Variety of Computing Resources

"Standard" Grid

Owned by VOs

- **Pledged resources**

Cloud

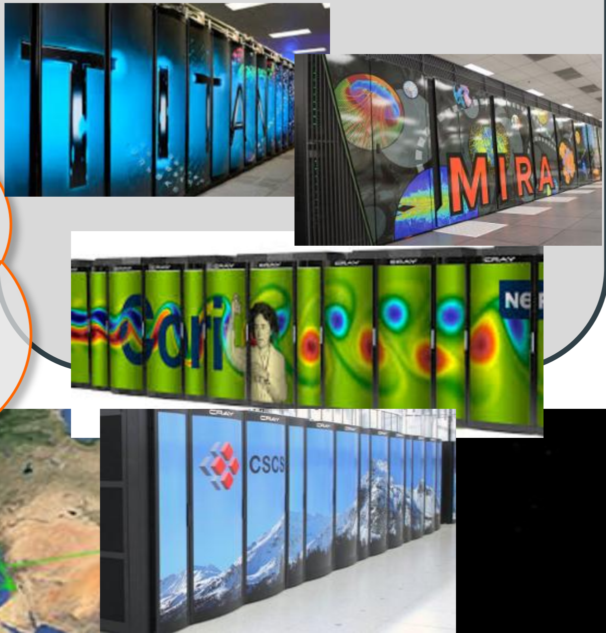
VOs Clouds and commercial Clouds

- **Rented resources, on-demand, opportunistic**

HPC

Research granted access

- **Opportunistic backfilling**



All these questions need to be addressed

4 Experiments

Different Computing models

Different frameworks for Data & Workflow management

Heterogeneous GRID technologies

Different infrastructures, middleware providers

Heterogeneous Compute Resources

“Standard Grid”, HPC, Cloud

egi

Open Science Grid

NORDUGRID

Grid Solution for Wide Area Computing and Data Handling

Information component as a middleware service of Distributed Computing

4 Experiments

Variety of GRID technologies

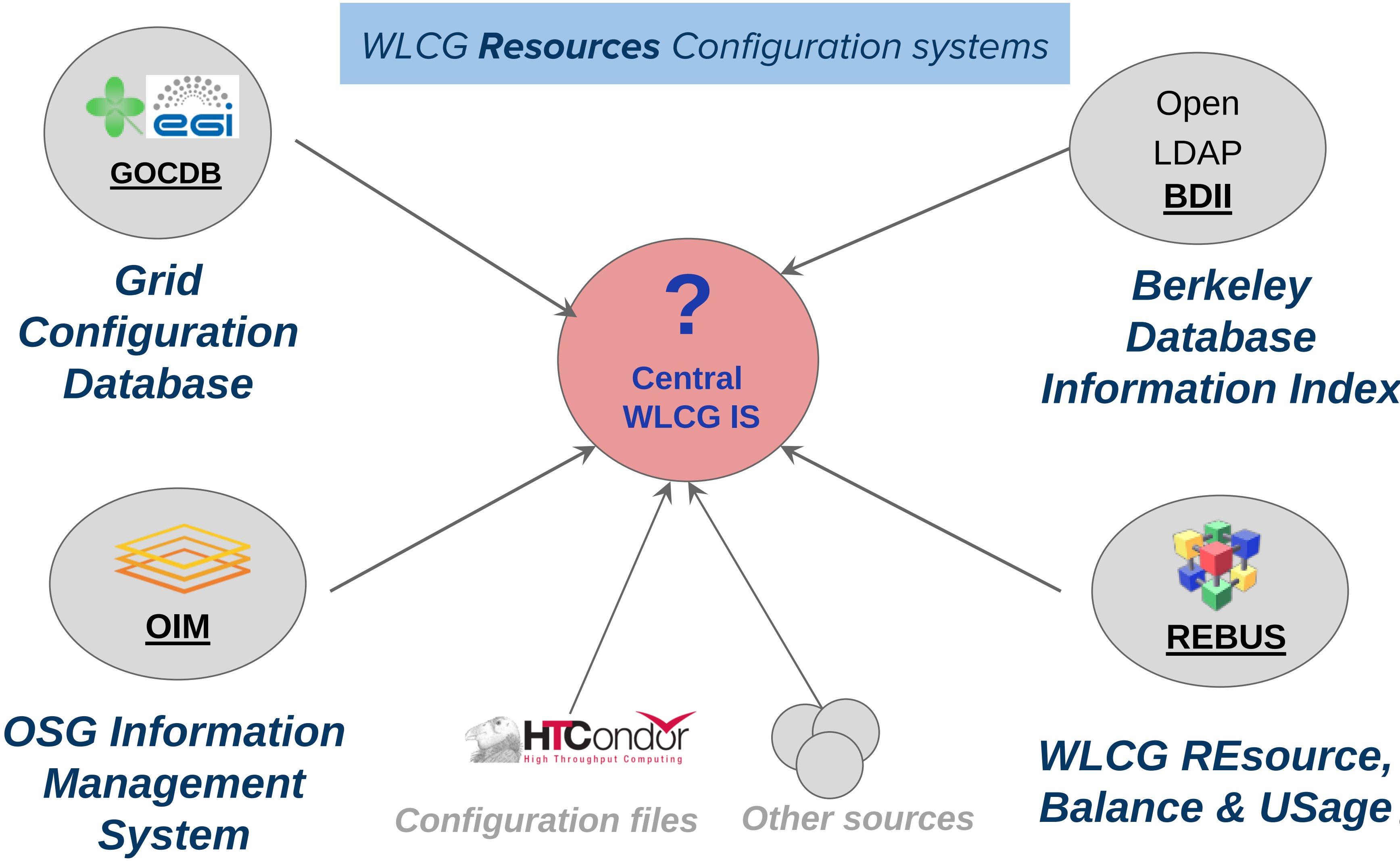
Heterogeneous Compute Resources

Need for an intermediate middleware system

Information component

to **describe** and **link** together all the Computing Resources, their **topology** and **services**

A big world of
Information systems (Resource configuration)



WLCG ***Resources*** *Configuration systems*

GOCDB

Grid Configuration Database

Open LDAP **BDII**

Berkeley Database Information Index

?

Central WLCG IS

OIM

OSG Information Management System

Configuration files

Other sources

REBUS

WLCG REsource, Balance & USage

A big world of
Experiments frameworks (Services)

*High level VO-oriented middleware services and applications require the diversity of **common** configurations as well.*

ATLAS Data Management System (Rucio)

PanDA

ATLAS Workload Management System

CMS Data Management System (Phedex)

ATLAS dashboard

Monitoring tools

sam

WLCG Squid monitoring

Pilots, AutoPilot Factories

BigPanda Monitors

Software Installation systems

DDM Accounting

Testing systems (HammerCloud)

Resources & Services:

Gluing them together via high-level Information component (AGIS/CRIC)

ATLAS Grid Information System (AGIS) is the central information system for ATLAS:

- connects **Resources** and **Experiment frameworks (services)** together for the ATLAS Collaboration
- integrates configuration and status information about resources, services and topology of the whole Computing infrastructure used by ATLAS Distributed Computing

Computing Resource Information Catalog (CRIC) is the AGIS evolution:

- Next-generation system
- Non experiment specific (but still experiment oriented), fitting the needs of multiple Collaborations

Information system:
a key component of Computing

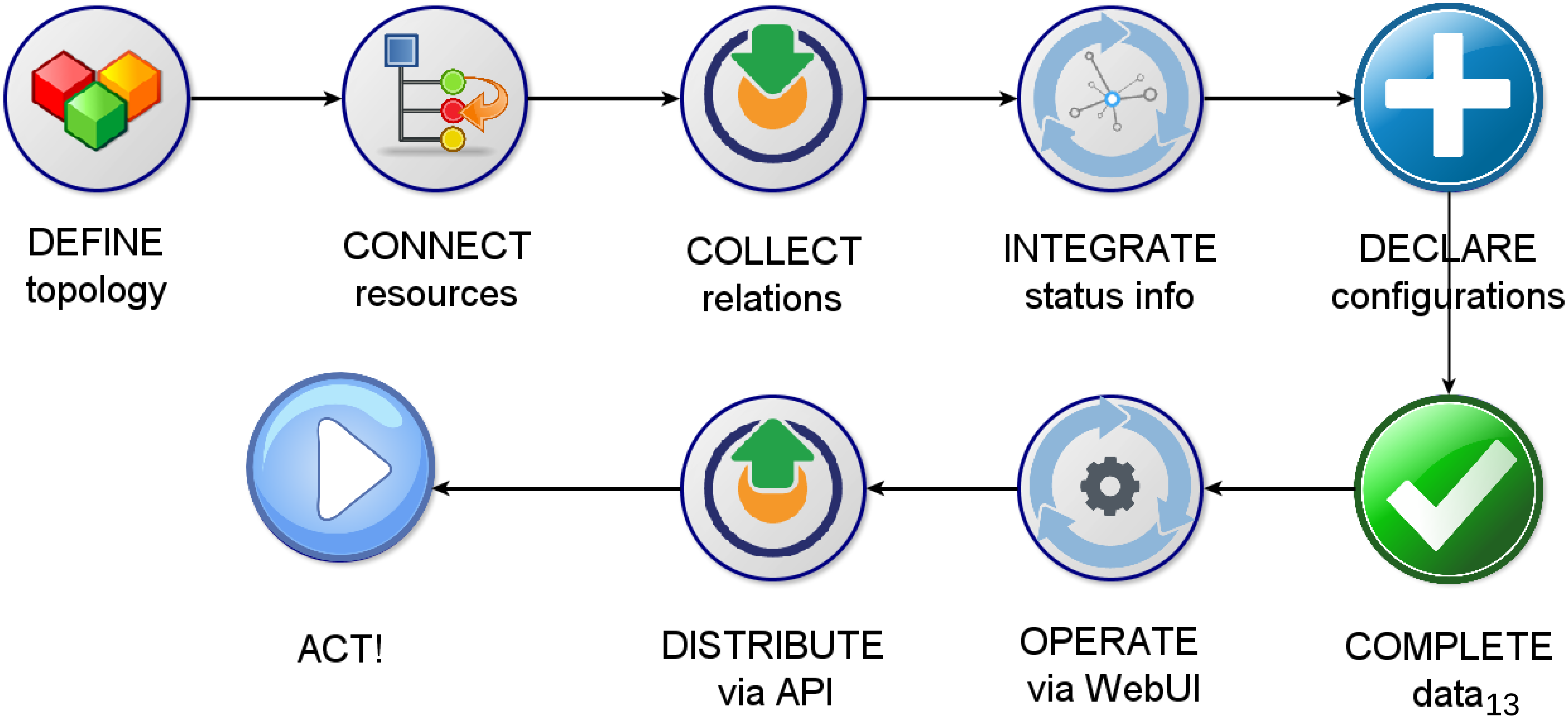
It does not really matter how big or small an Experiment/Collaboration is.

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 and more ..

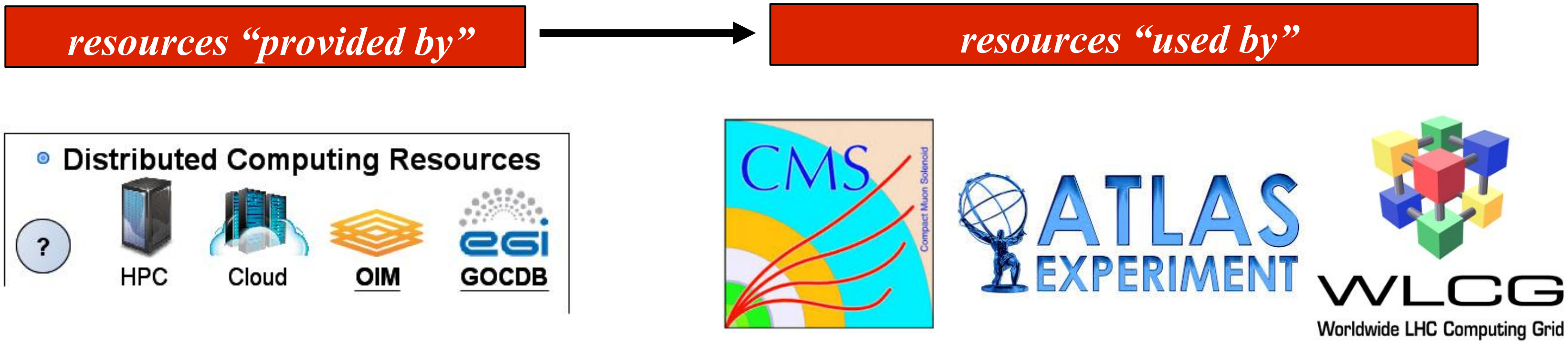
Computing Resource Information Catalog

CRIC is the **(high-level) middleware** designed to describe the topology of the Computing model (s), providing unified description of resources and services used by Experiment applications



CRIC Information Model: Resources & Experiment(s)

- Clear distinction between resources *provided by* (Sites) and resources *used by* (Experiments)
- Establish relationship between Resources to Experiment objects


resources "provided by"

resources "used by"

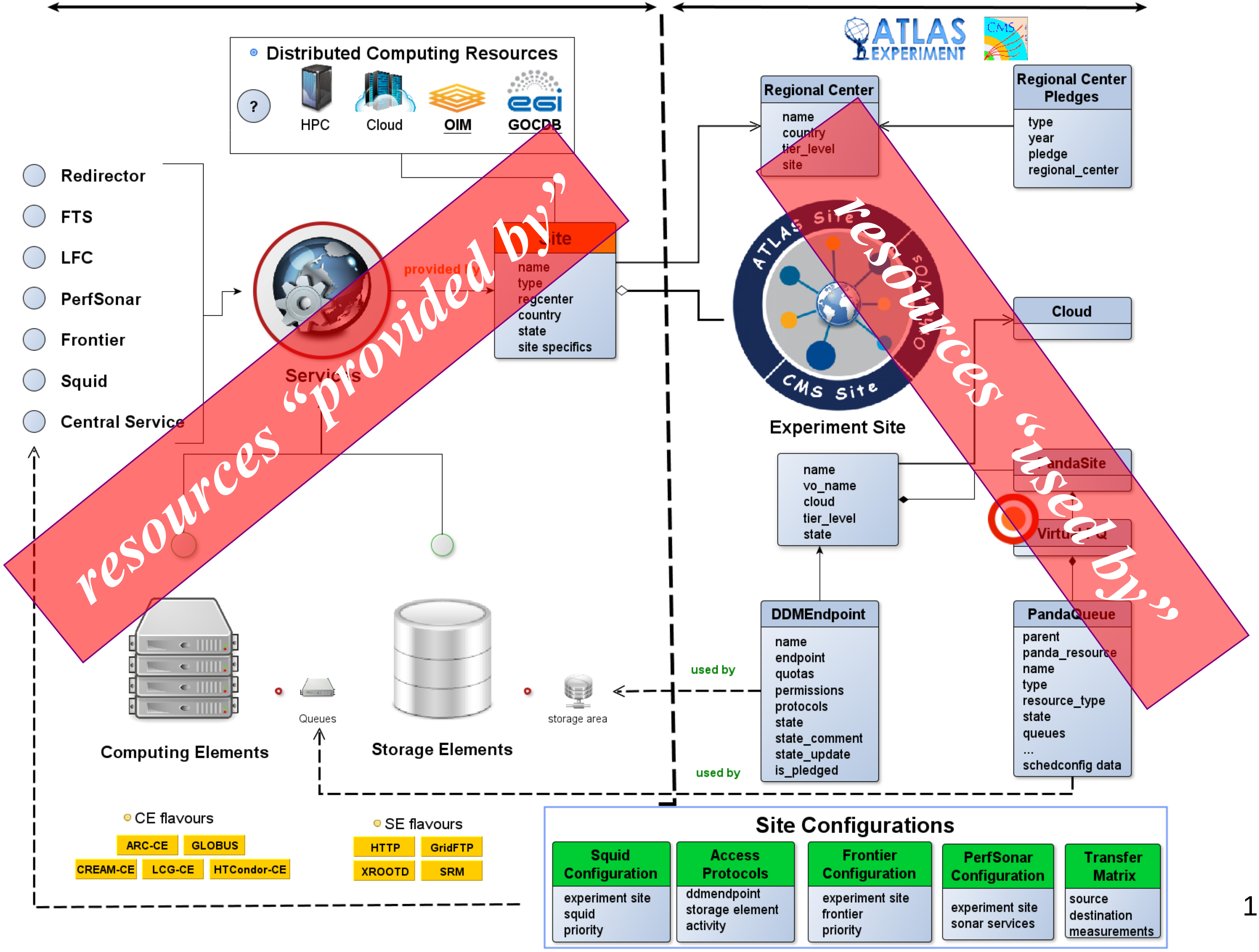
By Providing such abstraction layer from the physical Resources CRIC allows Experiments to define their own real organization of resources and required experiment specific structures.

"Physical"

HW/SW resources

"Logical"

Experiment usage



Distributed Computing Resources

?

HPC

Cloud

OIM

GOCDB

Redirector

FTS

LFC

PerfSonar

Frontier

Squid

Central Service

Services

provided by

Site

name
type
regcenter
country
state
site specifics

Computing Elements

Queues

Storage Elements

storage area

CE flavours

ARC-CE

GLOBUS

CREAM-CE

LCG-CE

HTCondor-CE

SE flavours

HTTP

GridFTP

XROOTD

SRM

resources "provided by"

ATLAS EXPERIMENT

CMS

Regional Center

name
country
tier_level
site

Regional Center Pledges

type
year
pledge
regional_center

ATLAS Site

CMS Site

Experiment Site

name
vo_name
cloud
tier_level
state

Cloud

PandaSite

VirtualQ

DDMEndpoint

name
endpoint
quotas
permissions
protocols
state
state_comment
state_update
is_pledged

PandaQueue

parent
panda_resource
name
type
resource_type
state
queues
...
schedconfig data

used by

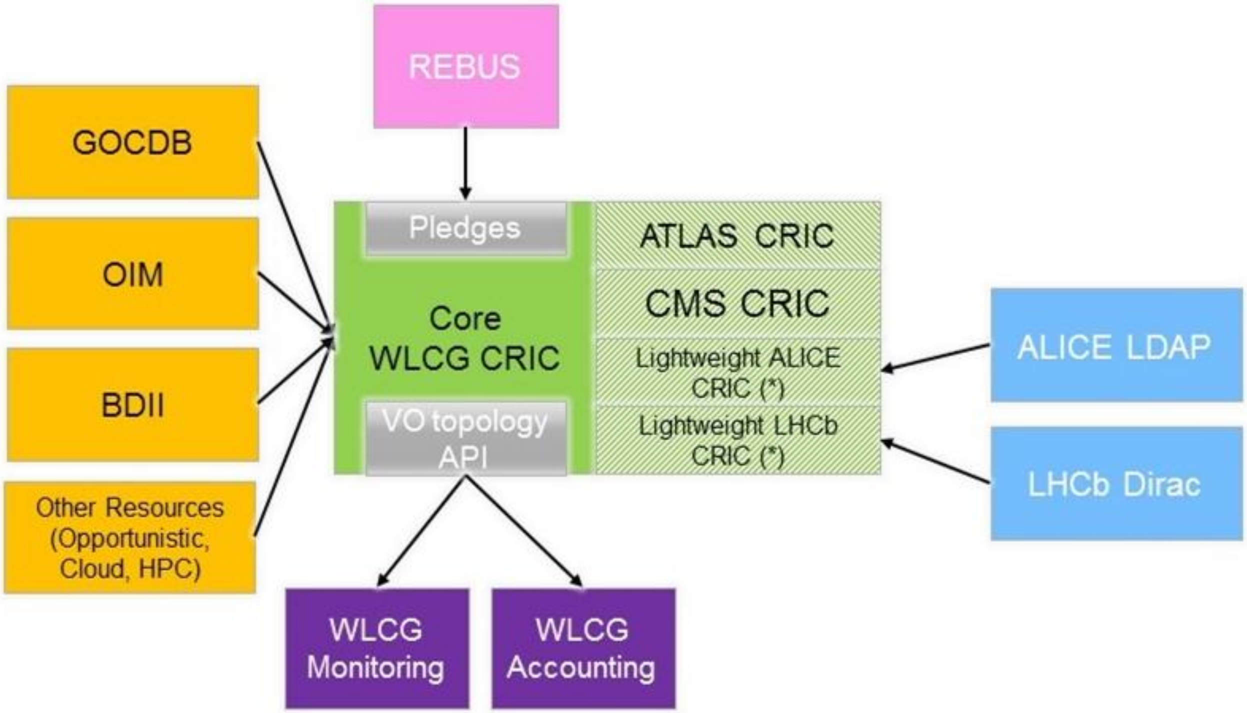
used by

resources "used by"

Site Configurations

Squid Configuration	Access Protocols	Frontier Configuration	PerfSonar Configuration	Transfer Matrix
experiment site squid priority	ddmendpoint storage element activity	experiment site frontier priority	experiment site sonar services	source destination measurements

Plugin based: Core and Experiment CRICs



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

Core CRIC

- ***resources "provided by"***
- Single entry point for WLCG topology and service configuration
- Consumes information from all 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

- Map site names (Experiment vs WLCG),
  details about which resources are used by the experiment
- Required for small VOs and WLCG monitoring applications

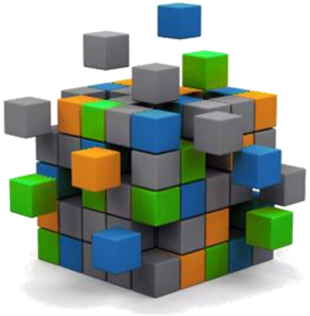
CRIC implementation: Involved technologies (Web2.0)

- Apache/WSGI + Python + Django framework as server backend
- Independent database backend
  (Oracle, MySQL, sqlite, etc)
- Web Services technologies
  (REST API, WebUI, widgets)
- Bootstrap framework as HTML/CSS/JS client frontend
  (responsive, interactive, mobile)
- client AJAX, JQuery library and plugins, own widgets
  (datatables, treeview, calendar..)
- Plugin based approach
  (shareable applications in “core” and re-used by many plugins)

CRIC Implementation features

Client-Server architecture, 2 independent services:

- REST API service (JSON, XML, etc) – mainly used to export data, bulk updates and operate data programmatically
- WebUI portal (interactive AJAX/Bootstrap) – mainly used to navigate, browse and declare objects (integrated form validation, suggestions, etc)



Module based implementation:

Scalable Django approach makes logic isolation into applications very effectively:

- Experiment CRICs inherit a lot of functionality from CORE, can customize and implement missing bits

Extensibility and scalability:

- CRIC services can be hosted on different nodes,
- LB support (same persistent DB instance)
- Experiment CRIC instances rely on central WLGC CRIC CORE

High level design and logic separation:

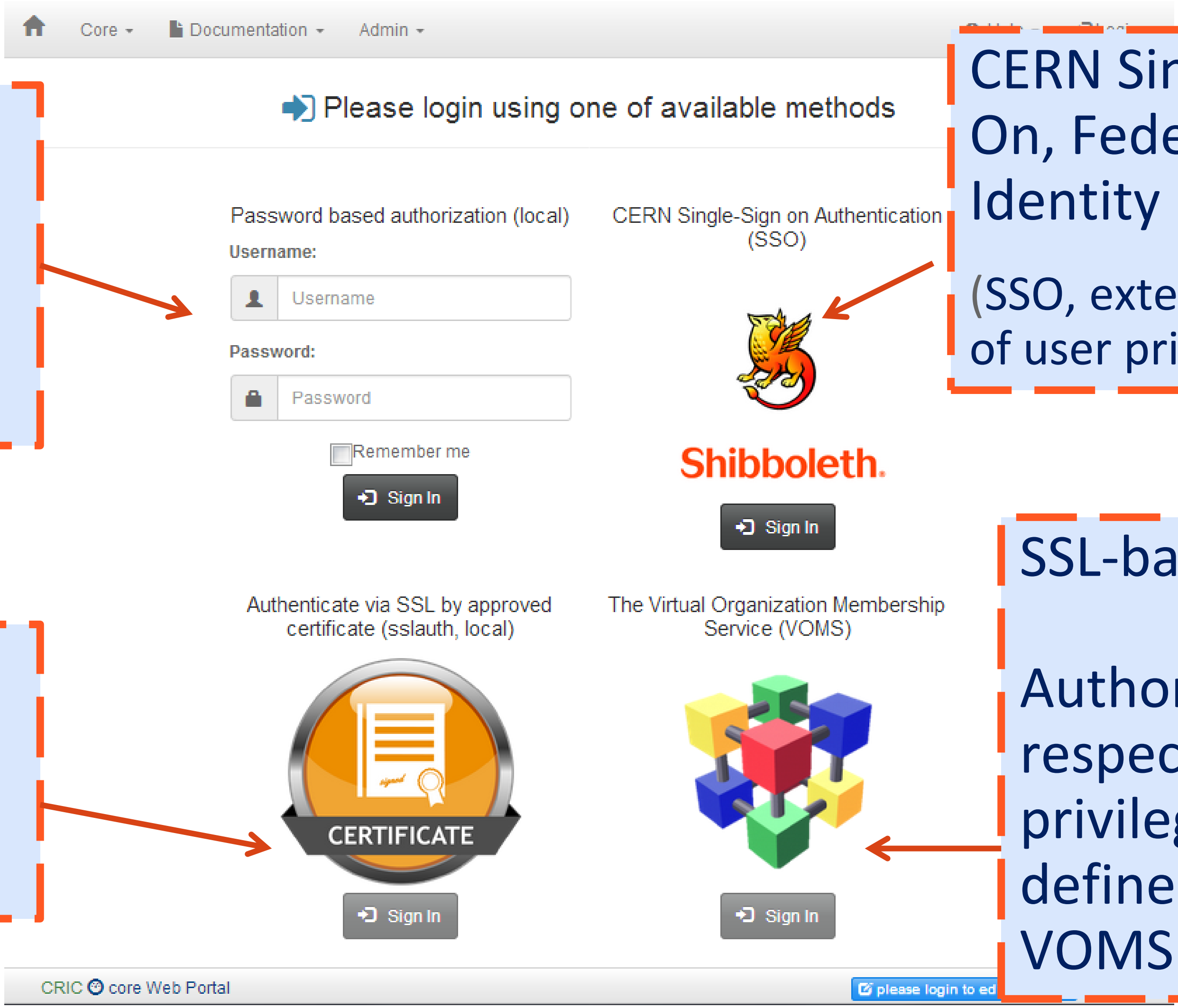
- all page “faces” are isolated into (html) templates that could be customized by VOs (e.g. CMS specific, ATLAS details, etc)

Access Control and Information protection

- ✓ CRIC supports information protection
- ✓ Authorization is required to modify data or access to restricted pages
- ✓ Group, Roles or list of specific Permissions could be directly associated to Users
- ✓ Each Experiment could configure own data access policies
- ✓ Several authentication methods
- ✓ Several types of permissions:
  - Model permissions (e.g. "can update all Site objects")
  - Instance specific permissions ("can update only given site CERN-PROD" or "all sites from CH country")
  - Global permissions (actions) – "can modify sensitive info in given form"

Authorization: Sources of permissions

The system supports several Authentication methods respect to Different Authorization sources to access WebUI pages or apply restricted actions



Please login using one of available methods

Password based authorization (local)

Username: Username

Password: Password

□ Remember me

CERN Single-Sign on Authentication (SSO)

Authenticate via SSL by approved certificate (sslauth, local)

The Virtual Organization Membership Service (VOMS)

password-based authentication

(Local permissions)

CERN Single Sign-On, Federated Identity

(SSO, external source of user privileges)

SSL certificate authentication

(Local permissions)

SSL-based

Authorization respect to user privileges defined in VOMS

CRIC features as a middleware for Experiments

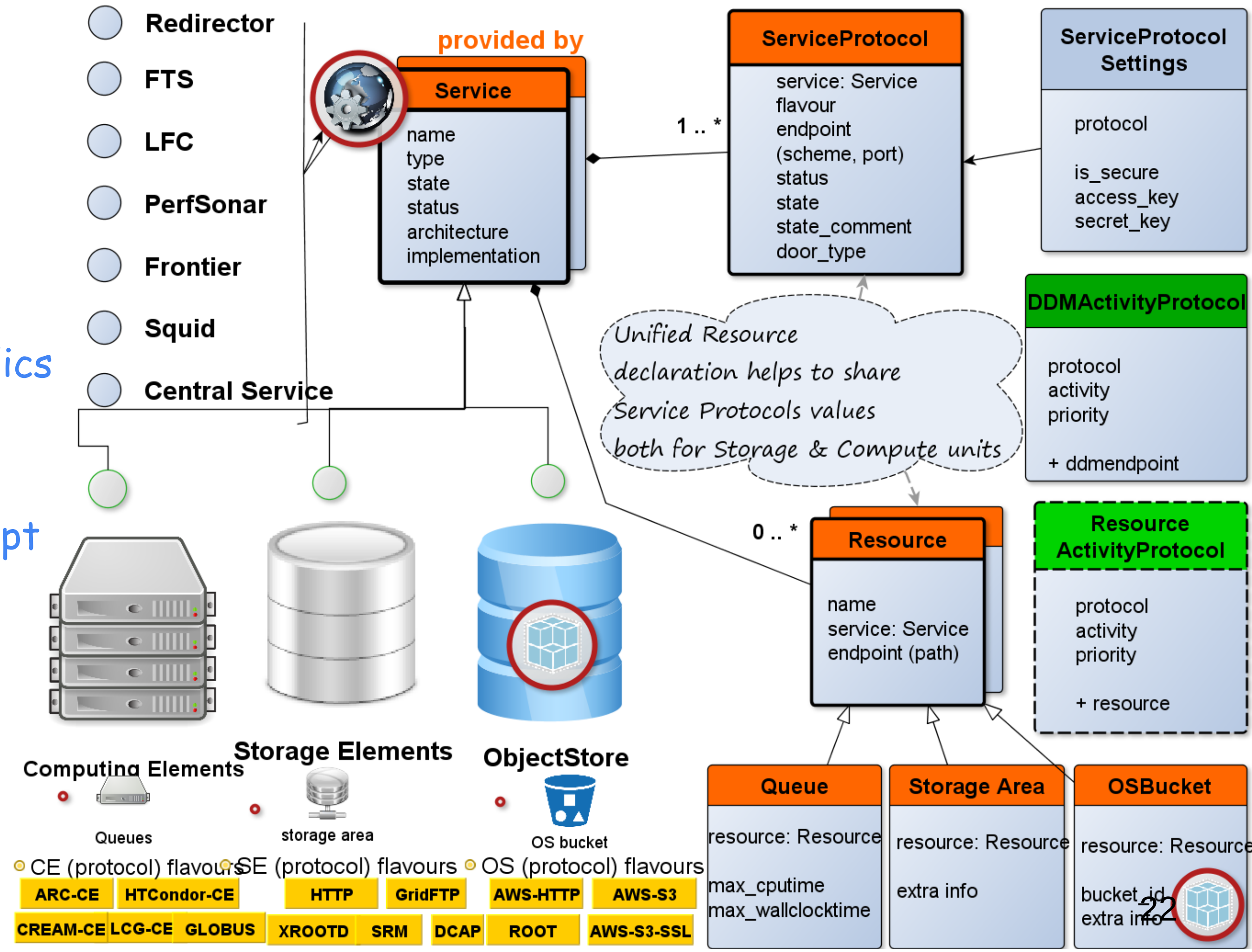
1. Helps to integrate new Computing technologies (for ATLAS) which do not yet appear in WLCG as GRID services or can not be part of WLCG in general, for example:
    - newer type of Storage based on ObjectStore technology
    - Federated Access to storage (FAX redirectors, direct access to remote files from Worker Nodes)
    - Description of opportunistic resources

2. Helps to minimize side effects for end-user applications of various internal migrations/changes/tests/evolution of Distributed Computing components:
    - Consolidation of protocols description that should be applied only for few sites, newer type of Storage based on OS technology
    - Keeps data export in several format for backward-compatibility reasons

3. Masks incompatible updates in external data providers or implement missing functionality/overwrite/fulfill data entries:
    - e.g. fix CE description
      (wrongly published number of cores, core-power)

Ongoing developments: Storage Object Consolidation (core CRIC)

- Global activity started within ATLAS to unify SE & CE declaration
- Link together all protocols, activities, closeness metrics, space tokens, other experiment specifics belong same Storage into unified Resource
- Multiple protocols concept
- Connect associated CE to default Storage for given activity
- Integration of new SE Objects with the experiments



Redirector
FTS
LFC
PerfSonar
Frontier
Squid
Central Service

provided by

Service
name
type
state
status
architecture
implementation

1 .. *

ServiceProtocol
service: Service
flavour
endpoint
(scheme, port)
status
state
state_comment
door_type

ServiceProtocol Settings
protocol

is_secure
access_key
secret_key

Unified Resource declaration helps to share Service Protocols values both for Storage & Compute units

DDMActivityProtocol
protocol
activity
priority

+ ddmendpoint

0 .. *

Resource
name
service: Service
endpoint (path)

Resource ActivityProtocol
protocol
activity
priority

+ resource

Computing Elements
Queues
CE (protocol) flavours
ARC-CE HTCondor-CE
CREAM-CE LCG-CE GLOBUS

Storage Elements
storage area
SE (protocol) flavours
HTTP GridFTP
XROOTD SRM DCAP

ObjectStore
OS bucket
OS (protocol) flavours
AWS-HTTP AWS-S3
ROOT AWS-S3-SSL

Queue
resource: Resource

max_cputime
max_wallclocktime

Storage Area
resource: Resource

extra info

OSBucket
resource: Resource

bucket_id
extra info

Ongoing developments: CRIC plugin for CMS

Computing Resource Information Catalog

- Implementation of CMS Computing Model (iterative process)
- Collecting Requirements and feature Requests from CMS experts

- Implemented Various collectors and Comparators fetching CMS specifics objects from external sources
  (SiteDB loader, GlideIn entires provider)

- Base WebUI forms to modify objects of CMS Computing Model
- Base REST API for data export, including integrated CORE functionality and CMS specific objects of Information Model
  (CMSSite, ComputeUnit, ComputeResources, GlideInEntry, etc)

- Implementation of data export in backward compatible format currently used by CMS applications (GlideInFactory XML entries)

- CMS CRIC part is already integrated into CRIC instance (getting from the box all core functionality), first prototype is in the process to be released

- Currently CMS CRIC is under active testing

Ongoing developments: CRIC plugin for COMPASS at CERN SPS

- LIT JINR at Dubna was inspired by AGIS/CRIC project and asked about CRIC evaluation for COMPASS Experiment at CERN SPS and then for possible application for NICA collider

- Currently CRIC team is developing a prototype for the Information system within CRIC infrastructure targeted to cover immediate needs of COMPASS Experiment.

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

- The implementation of COMPASS CRIC plugin overlaps with ATLAS CRIC so that CRIC modules for COMPASS will be also useful for ATLAS and help for further AGIS to CRIC migration

First prototype of COMPASS CRIC plugin is currently under testing and will be released soon

Some history

AGIS: mainly ATLAS oriented IS

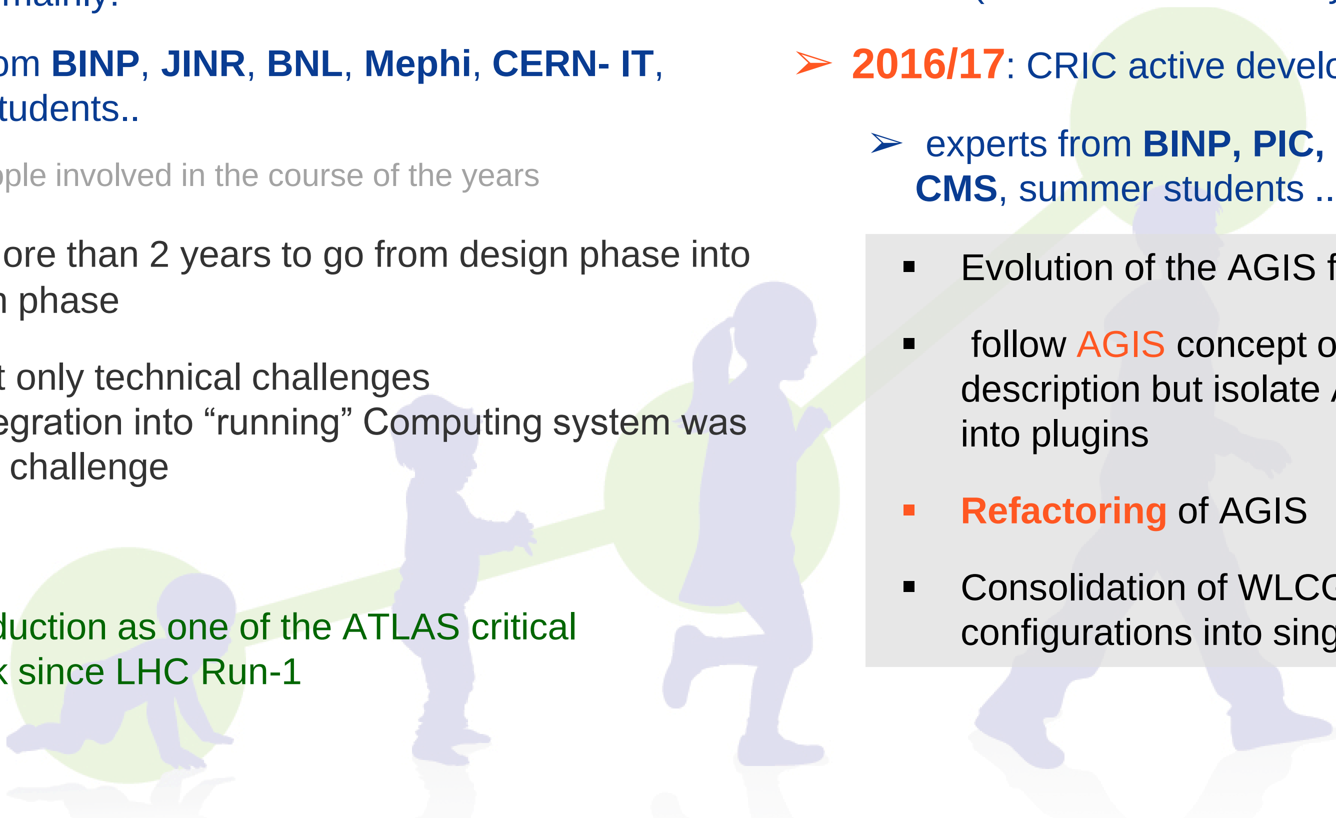
CRIC: experiment independent but still experiment oriented

- **2009**: First proposal of AGIS as a GRID Information system for ATLAS
- **2009**: A collaborative project involving several institutes, mainly:

  experts from **BINP**, **JINR**, **BNL**, **Mephi**, **CERN- IT**, summer students..

  Several people involved in the course of the years
- **2011**: More than 2 years to go from design phase into production phase
  - Not only technical challenges
  - Integration into "running" Computing system was the challenge
- **2011**:
  In full production as one of the ATLAS critical framework since LHC Run-1

- **2015**: CMS evaluation: several prototypes of AGIS for CMS Experiment
- **2015/16**: established new TaskForce within WLCG (WLCG Information System Evolution TF)
- **2016/17**: CRIC active developments
  - experts from **BINP, PIC, LUND, CERN-IT**, **CMS**, summer students ..
    - Evolution of the AGIS framework
    - follow AGIS concept of Resource description but isolate ATLAS specifics into plugins
    - **Refactoring** of AGIS
    - Consolidation of WLCG topology and configurations into single IS



Conclusion

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 needs
  - Fundamental Concept of Resource description ("provided by" vs "used by") allows to effectively cope with requirements of several Collaborations with maximum flexibility
  - 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 actually shared and centrally managed to minimize support efforts for several Collaborations.
  - Built-in **lightweight CRIC** version of experiment CRIC will be provided for small Collaborations, not even need to host your own full CRIC service.
  - **CMS CRIC** and **COMPASS CRIC** plugins are under active developments (testing)

Thank you for your attention!

Examples of Information stored in CRIC

Topology of Computing model:

- List of sites, services and its descriptions
- Site specifics
- CE, SE service details
- Local (batch) queue specifications

Declatation of high level experiment data models

- Storage Element in DDM system
- Compute Unit in PanDA
- Configuration of Frontier/Squid infrastructure

Quasi-static info:

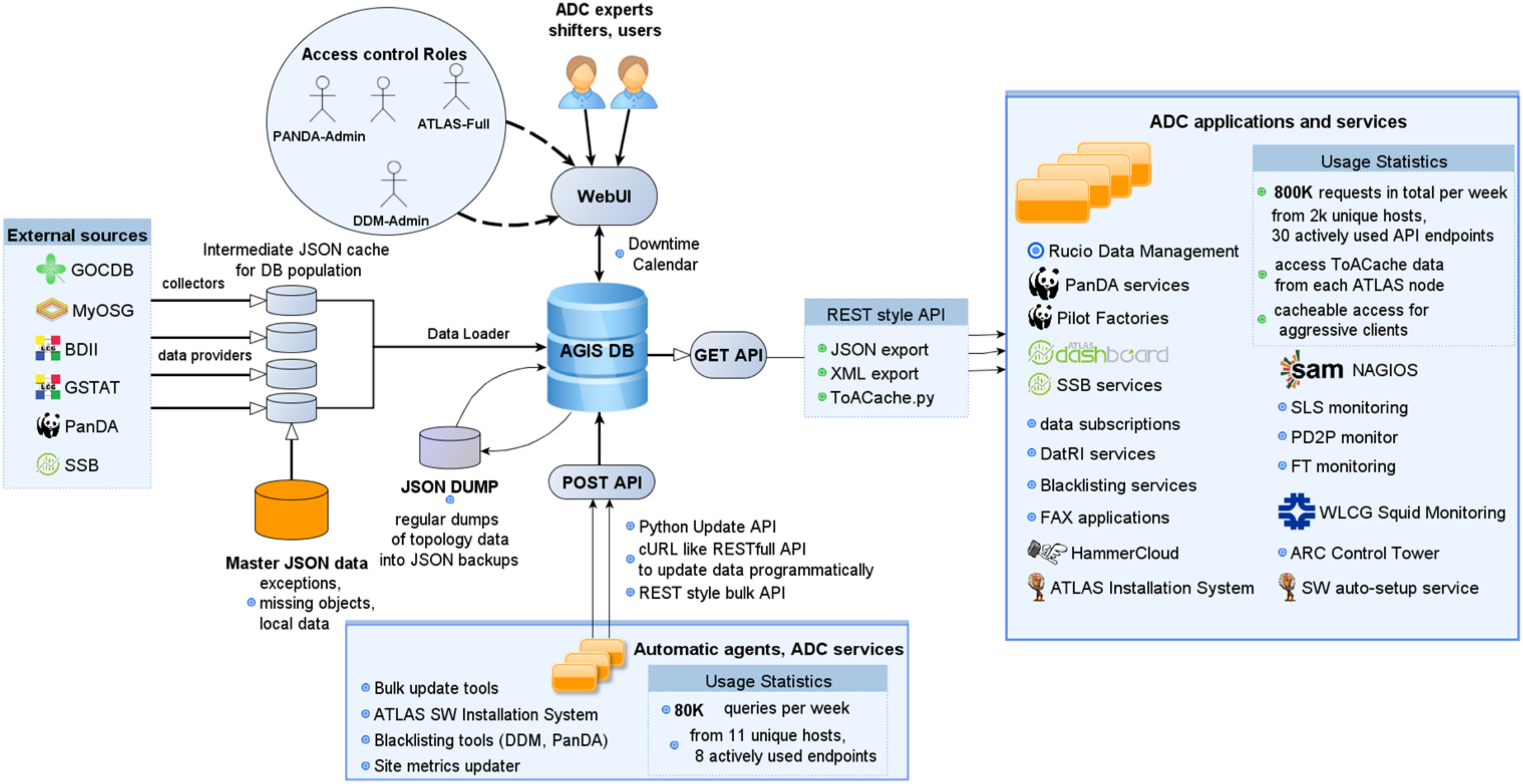
- Site dowmtines details
- Site blacklisting details

Two types of data:

- external (cached and collected from ext sources)
- managed within the system (CRIC is the master for such data)

Architecture of the system: Example from ATLAS

General overview of system services and data workflow



Access control Roles

PANDA-Admin

ATLAS-Full

DDM-Admin

ADC experts
shifters, users

WebUI

Downtime
Calendar

External sources

- GOCDB
- MyOSG
- BDII
- GSTAT
- PanDA
- SSB

Intermediate JSON cache
for DB population

collectors

data providers

Data Loader

AGIS DB

GET API

JSON DUMP

- regular dumps
of topology data
into JSON backups

Master JSON data

- exceptions,
missing objects,
local data

POST API

- Python Update API
- cURL like RESTfull API
to update data programmatically
- REST style bulk API

REST style API

- JSON export
- XML export
- ToACache.py

ADC applications and services

- Rucio Data Management
- PanDA services
- Pilot Factories
- ATLAS dashboard
- SSB services
- data subscriptions
- DatRI services
- Blacklisting services
- FAX applications
- HammerCloud
- ATLAS Installation System

Usage Statistics

- **800K** requests in total per week
from 2k unique hosts,
30 actively used API endpoints
- access ToACache data
from each ATLAS node
- cacheable access for
aggressive clients

- sam NAGIOS
- SLS monitoring
- PD2P monitor
- FT monitoring
- WLCG Squid Monitoring
- ARC Control Tower
- SW auto-setup service

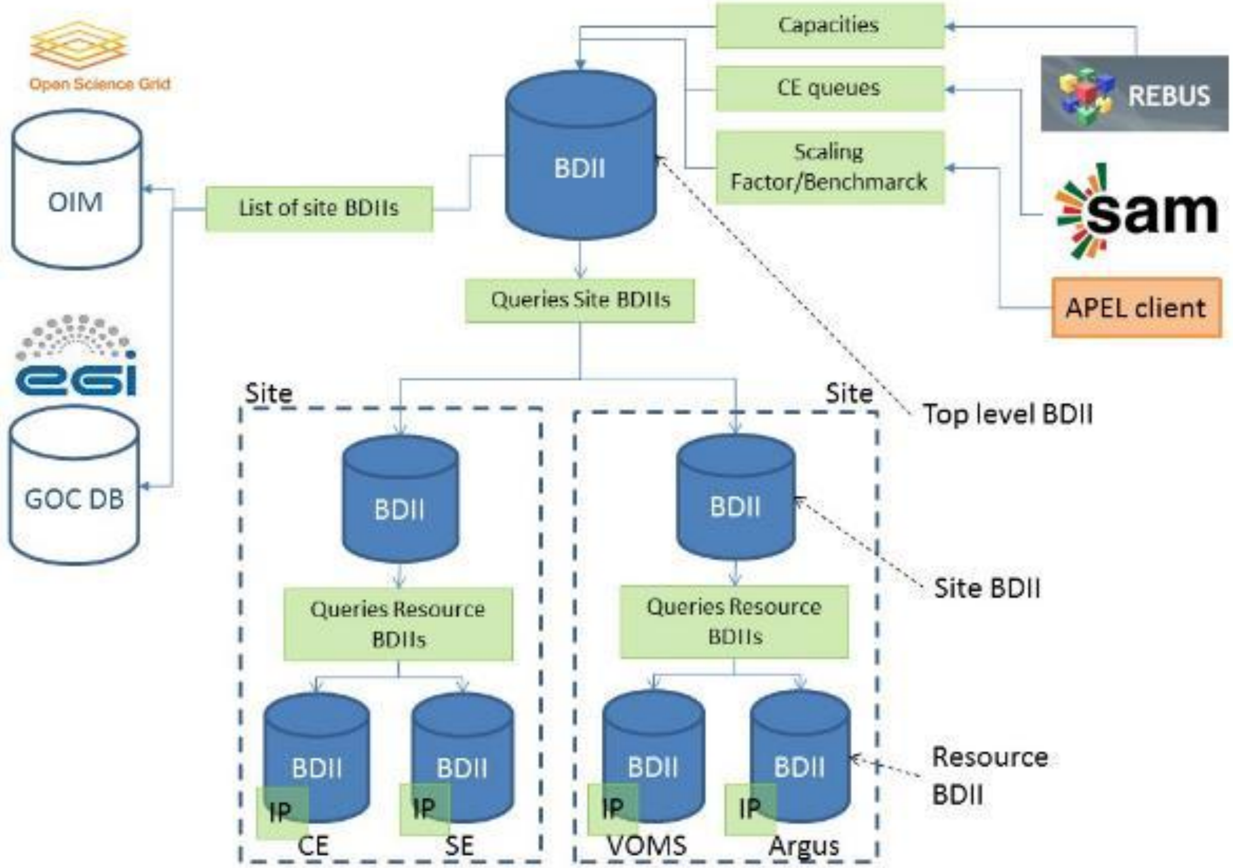
Automatic agents, ADC services

- Bulk update tools
- ATLAS SW Installation System
- Blacklisting tools (DDM, PanDA)
- Site metrics updater

Usage Statistics

- **80K** queries per week
- from 11 unique hosts,
8 actively used endpoints

Current WLCG Information System



Distributed BDII approach

- LDAP based
- no WebUI to manage data
- require installation of the service at each WN
- Powerful but complicated "language" of Computing topology description (GLUE schema)
- General trend to simplification of operations (OSG is dropping support of BDII)

Does not fit well Experiments needs (ATLAS)