



# WLCG Information System Evolution

**2019 Joint HSF/OSG/WLCG Workshop HOW2109**

Julia Andreeva , CERN

# Motivation

- Being the core service of the WLCG infrastructure IS needs to evolve to follow infrastructure growth, technology evolution and changes in the computing models of the experiments
- Current IS has been designed for the EGI fully distributed operational model.
- WLCG operational model is centralized. While we are missing central place where topology and configuration data can be aggregated, validated, corrected and then served to all interested clients
- System is not flexible. For example, integration of new types of resources (HPC, clouds) is not straightforward.

# Follow up on implementation

- **Primary data sources** for service level information. Currently propagated via BDII
- **Central topology and configuration service** which collects and validates data from all primary sources. Provides possibility to correct data by authorized users. Provides common set of UIs and APIs for all interested clients. Sends notifications in case of spotted inconsistencies, etc...
- Both should be compatible in terms of data structures for service description

# Primary data sources

## Current

For static and dynamic information

Service BDII -> Site BDII -> Top level BDII

GLUE schema and ldap protocol

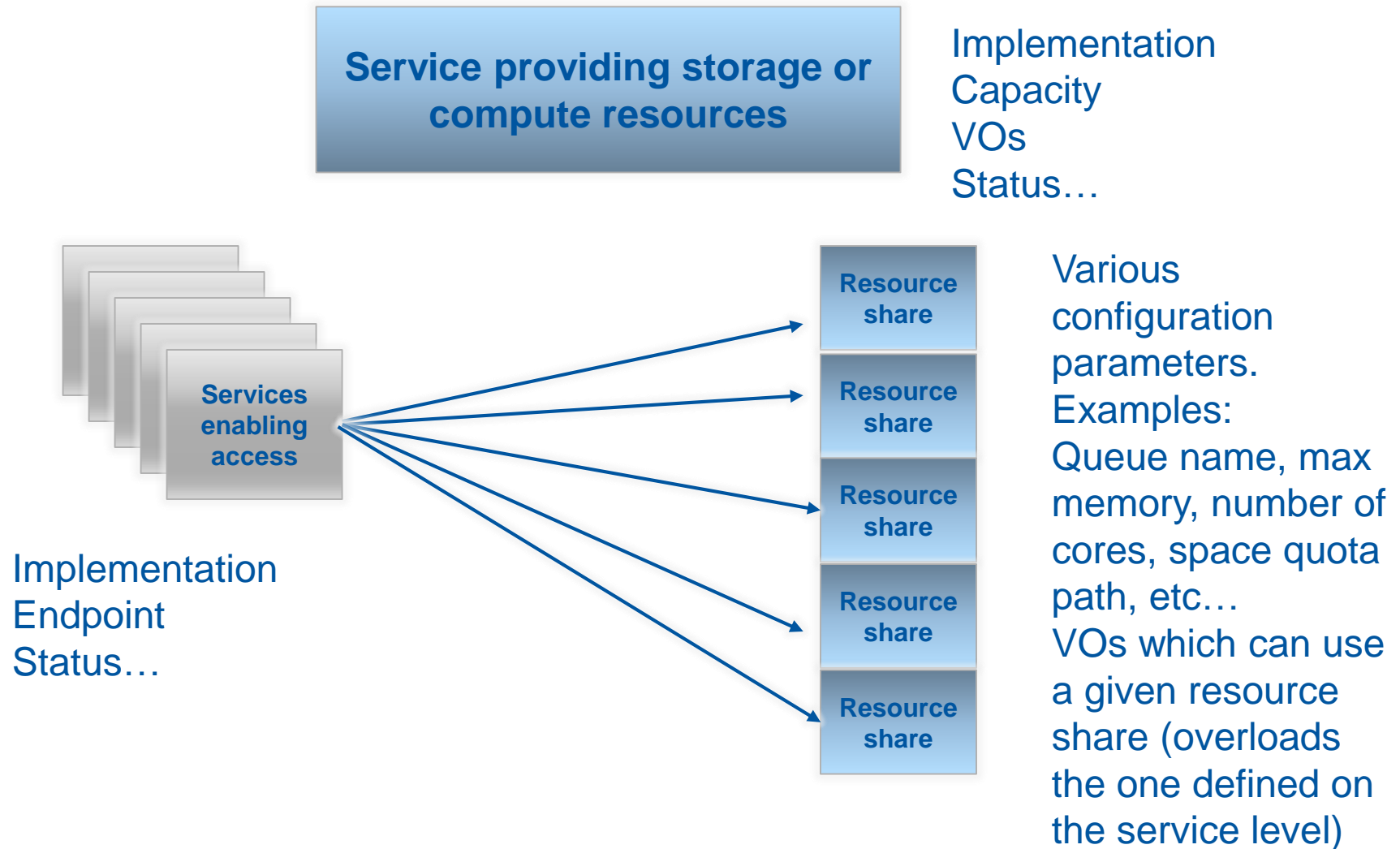
## Future

For static and dynamic information

Storage Resource Reporting (SRR) json topology description and storage accounting data for storage services and Computing Resource Reporting (CRR) topology description for computing services. Accessible via http protocol

Limited set of attributes compared to BDII, published in a well defined format, following an agreed schema

# Data structures and their attributes



# Compute resource example

**condor cluster at AGLT2**

```
"jobmanager": "condor",  
"jobmanager_version": "xxx",  
"status": "production",  
"number_logical_cpus": xxx,  
"capacity_HS06":xxx,  
"OS" : xxx,  
"assigned_vos" : [ATLAS, CMS]
```

## Computing elements

<https://gate01.aglt2.org:8888>

<https://gate12.aglt2.org:1122>

## Queues

**Long**

**Short**

```
"max_walltime_minutes": xxx,  
"max_cputime_minutes": xxx,  
"memory_GB":xxx,  
"assigned_vos": ["ATLAS", "CMS"],
```

```
"flavour": "HTCondor-CE",  
"version": "2.12",  
"status": "production"
```

More details can be found following the link:

[https://docs.google.com/document/d/1pg\\_5Kibc\\_Z4JF4\\_HJyW5xL6GVYKwXxOU7DXf2QP9Ag](https://docs.google.com/document/d/1pg_5Kibc_Z4JF4_HJyW5xL6GVYKwXxOU7DXf2QP9Ag)

# Storage Resource Example

**goliath100.farm.particle.cz**  
(DPM at Prague)

“implementation”：“DPM”,  
“status”：“production”

## Protocols

root

srm

https

gsiftp

## Storage shares or space quotas

ATLASDATADISK

DUNEDATADISK

DTEAM

ATLASLOCALGR  
OUPDISK

“interfacetype”：“gsiftp”,  
“status”：“production”,  
“endpoint”：“  
gsiftp://goliath100.farm.particle.cz/

“servingstate”：“open”,  
“vos”：“atlas”,  
“path”：“/dpm/farm.particle  
.cz/home/atlas”,  
“totalsize”：“208907209277  
4400,  
“usedsize”：“182119395051  
4079

**More details  
can be found  
following the  
link:**

<https://docs.google.com/document/d/1yzCvKpxsbcQC5K9MyvXc-vBF1HGPPBk4vhjw3MEXoXf8/>

# What else?

- Relationship between access services/protocols and resource shares. Example: CE X provides access to the queues A and B, not C and D (implemented)
- Prioritization of usage of access protocols for particular operations. Example: protocol A has higher priority compared to protocol B for local reading (currently not foreseen in SRR)
- Format should be flexible and extendable to be able to handle additional attributes. Example: QoS parameters of the storage



# SRR and CRR implementation

- Work with all storage middleware providers in order to enable SRR generated by storage services. All of them put in place first SRR prototype. More details.
- DPM is the most advanced one. DPM already entered deployment phase. DPM Upgrade Task Force has been setup. Pioneer sites have been upgraded.
- CRR format has been agreed by the IS Evolution Task Force. Implementation of CRR by the pioneer sites is ongoing. Was prototyped not only for classical batch resources, but also for cloud resources by Andrew McNab (vac & vcycle)

# Central Topology and Configuration Service

- Computing Resources Information Catalogue (CRIC) is a high level information system aiming to describe the topology of the WLCG infrastructure and other resources used by the LHC experiments (HPC, clouds, etc...) and experiment-specific configuration required to exploit these resources according to the experiments Computing models.
- Inspired by ATLAS Grid Information System (AGIS). Evolving AGIS towards common global solution.
- CRIC should become a central entry point for all kinds of WLCG topology and configuration information

# CRIC development principles

- Plugin based -> straightforward customization to address various experiment requirements and implementation of the dedicated experiment instances
- Shared building blocks -> common look and feel and optimized development process
- Flexible -> possible to follow technology evolution and changes in the experiment computing models and applications.

**Lego bricks like approach**

# CRIC structure

## WLCG CRIC

Extension of CORE CRIC to support LHCb and ALICE and WLCG central operations tasks

## CMS CRIC

To describe CMS - specific configuration required to use provided resources

## CORE CRIC

To describe resources provided to the LHC experiments

## DOMA CRIC

To provide additional configuration required for TPC tests

**Every CRIC instance contains CORE part and one of the blue plugins**

## ATLAS CRIC

To describe ATLAS-specific configuration required to use provided resources

# Functionality

- Collecting data from variety of sources (GocDB, OIM, REBUS, CMS Glideins config XML, BDII, CRR/SRR...)
- Providing UI and APIs for all kinds of topology and configuration info
- Authentication & Authorization for data viewing and update
  - Several Authentication methods are enabled
  - Flexible utilization of Permissions, Roles and Groups at various levels
  - Fine grain A&A on the level of a single CRIC object
- Advanced logging functionality. Logging is performed on the CRIC object level. One can check **who, when and how** interacted with a given object
- Notification. One can subscribe to be notified in case of modification of a particular object

# Status

- CMS CRIC deployed in production in September 2018. Provides functionality which allows to retire SiteDB.
  - [Production instance](#)  
<http://cms-cric.cern.ch/>
- WLCG CRIC – stable prototype. Will be deployed in production after validation of ALICE and LHCb VO feeds (work in progress).
  - [Production instance, no editing via UI is currently allowed.](#)  
<http://wlcg-cric.cern.ch/>
  - [Validation and playground instance. Anyone can edit information](#)  
<http://wlcg-cric-dev-1.cern.ch>
- DOMA CRIC – prototyped, work in progress
- ATLAS CRIC – the plan is to have a workable prototype by the end of Q2 2019. Migration from AGIS should be agreed with ATLAS. If thing goes well migration could be done by the end of 2019.

# WLCG CRIC functionality

- View & edit: federations, pledges, resource center sites, experiment sites, services, storage protocols, queues
- Table view and detailed instance view is provided for all data models
- All changes are being logged
- Downtime calendar ( two types of visualization)
- A&A is work in progress
- UI to add/modify pledges.

# WLCG CRIC clients

- CRIC will generate VOfeeds for all experiments. VOfeeds are used by central, testing, monitoring and accounting applications
- WLCG Storage Space Accounting system already uses CRIC for storage topology description
- Gradual migration of REBUS clients to WLCG CRIC will start with central monitoring applications



# DOMA CRIC

- TPC DOMA activity required configuration for the tests. The tests are performed using RUCIO. Number of participating sites is growing. Various configurations are required for every protocol depending on the tested activity (LAN/WAN reading, LAN/WAN writing)
- CRIC team volunteered to help in providing necessary configuration. It is VO-independent. Decided to implement a dedicated plugin to host all the models describing DOMA's configurations
- Working in a close loop with Rucio developers CRIC development team setup DOMA CRIC instance (<http://doma-cric.cern.ch>) which provides necessary APIs (full storage description enchanted with some configuration variables) for DOMA tests along with the UI to inject/modify input data
- Next step is to provide transfer matrix (work in progress)
- Useful experience to prototype RUCIO-related functionality for CMS and ATLAS. Proved that the system is flexible and can easily satisfy new requirements

# Conclusions

- WLCG IS is evolving towards new implementation which should be able to respond the needs of changing computing infrastructure and WLCG operational model
- Two directions of work : primary information sources (site services describing themselves) and central topology and configuration service
- We aim to provide a flexible and extensible system which can be quickly adopted for ever-changing environment

# Backup slide. Screenshot examples (1)

## Federation table

Export  
data in  
json, csv  
or pdf

Switch  
on  
filter

Sort  
columns

Export

Columns

Filter

Reload

Pledge View

Show

100

entries

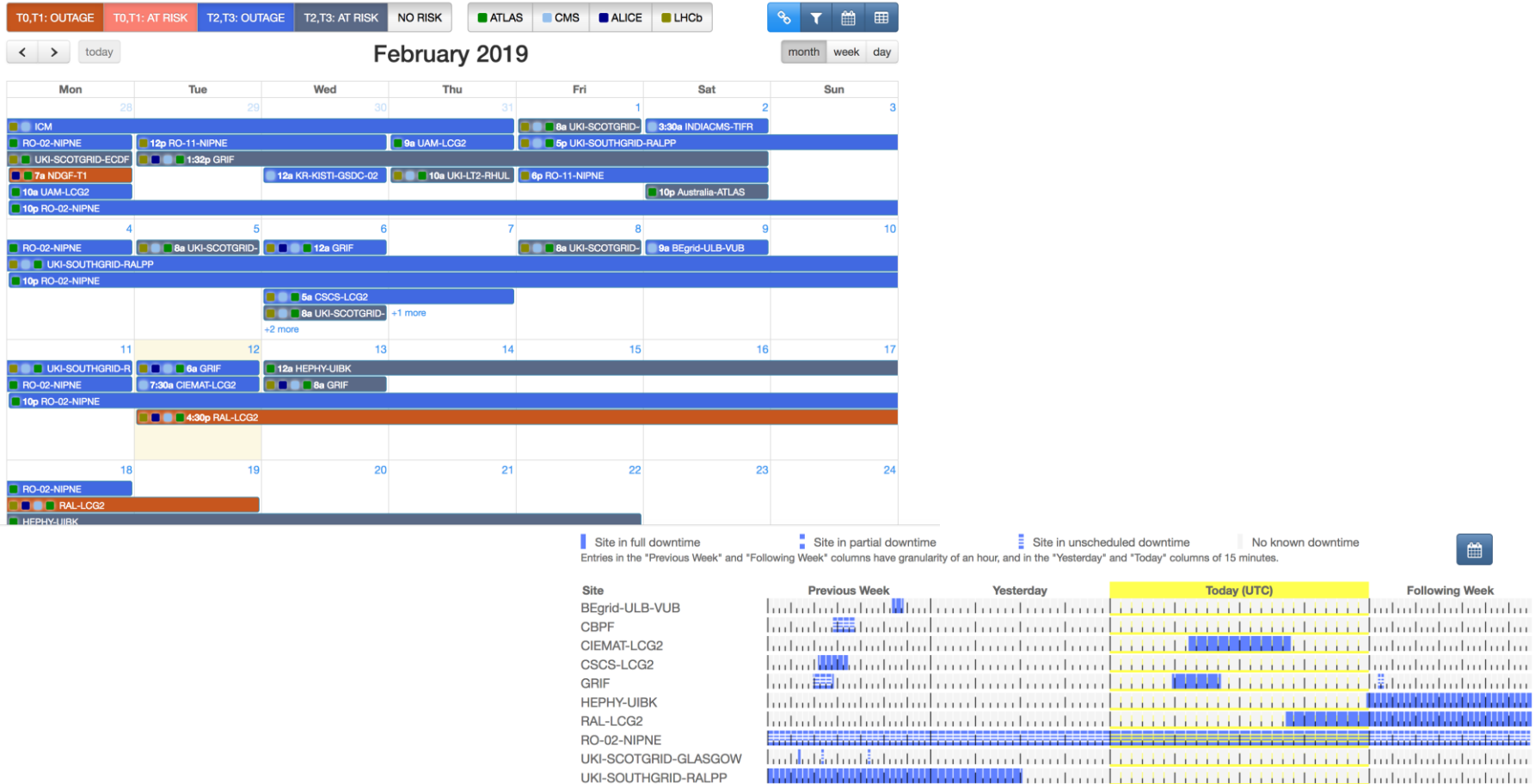
columns

Search:


Federation	Accounting name	Tier	Country	DISK 2018 ALICE	CPU 2018 ALICE	TAPE 2018 ALICE	DISK 2018 ATLAS	CPU 2018 ATLAS	TAPE 2018 ATLAS	DISK 2018 CMS	CPU 2018 CMS	TAPE 2018 CMS	DISK 2018 LHCb	CPU 2018 LHCb	TAPE 2018 LHCb
<a href="#">AT-HEPHY-VIENNA-UIBK</a>	Austrian Tier-2 Federation	2	Austria	250	5000	0	120	1875	0	500	10000	0	0	0	0
<a href="#">AU-ATLAS</a>	University of Melbourne	2	Australia	0	0	0	1390	18318	0	0	0	0	0	0	0
<a href="#">BE-TIER2</a>	Belgian Tier-2 Federation	2	Belgium	0	0	0	0	0	0	5570	56200	0	0	0	0
<a href="#">BR-SP-SPRACE</a>	SPRACE, Sao Paulo	2	Brazil	0	0	0	0	0	0	1900	25200	0	0	0	0
<a href="#">CA-EAST-T2</a>	Canada-East Federation	2	Canada	0	0	0	2275	29000	0	0	0	0	0	0	0
<a href="#">CA-TRIUMF</a>	TRIUMF-LCG2	1	Canada	0	0	0	7400	94900	21100	0	0	0	0	0	0
<a href="#">CA-WEST-T2</a>	Canada-West Federation	2	Canada	0	0	0	2275	29000	0	0	0	0	0	0	0
<a href="#">CH-CERN</a>	CERN-PROD	0	Switzerland	26200	350000	49100	27000	411000	105000	26100	423000	97000	11400	88000	33600
<a href="#">CH-CHIPP-CSCS</a>	CHIPP	2	Switzerland	0	0	0	2100	54000	0	1600	36000	0	800	24000	0
<a href="#">CN-IHEP</a>	IHEP, Beijing	2	China	0	0	0	400	5780	0	540	5780	0	0	0	0
<a href="#">CZ-Prague-T2</a>	FZU AS,	2	Czechia	1400	6000	0	1800	16000	0	0	0	0	0	0	0

# Backup slide. Screenshots examples (4)

## Two different representations of the downtime calendar



# Backup slide. APIs

 Core ▾ Core API ▾ WLCG ▾ Docs ▾ Admin ▾ Logs ▾

RC site (GO

Adm

C

Last modifi

Status Info

OL

State

API index

Federation JSON export

Resource Centre Site JSON export

Service JSON export

Compute Element JSON export

Site JSON export

Downtime JSON export

User JSON export

User Roles JSON export

Group JSON export

Edit

base URI: [/api/core/](#)

The list of accepted functions and its descriptions

operation	API endpoint	available JSON presets	description of accepted input keys			required_keys
			accepted_keys	accepted_list_keys	accepted_flags	
base URI= <a href="/api/accounts/group/query/">/api/accounts/group/query/</a>						
help	/api/accounts/group/query/help/					
list	/api/accounts/group/query/list/	<a href="#">default</a>   <a href="#">full</a>   <a href="#">merged</a>	columns, draw, egroups, email, firstname, id, last_modified, lastname, length, name, order, preset, source, start, username, users	name, source, user_id, username	json, json_pretty	
base URI= <a href="/api/accounts/user/query/">/api/accounts/user/query/</a>						
help	/api/accounts/user/query/help/					
list	/api/accounts/user/query/list/	<a href="#">default</a>   <a href="#">group-responsibilities</a>   <a href="#">groups</a>   <a href="#">people</a>   <a href="#">roles</a>   <a href="#">site-responsibilities</a>   <a href="#">whoami</a>	columns, date_joined, dn, draw, email, firstname, group, group_source, groups, id, is_active, is_staff, is_superuser, lastname, length, order, preset, profiles, start, username	group, group_id, group_source, username	json, json_pretty	
base URI= <a href="/api/core/ce/query/">/api/core/ce/query/</a>						
help	/api/core/ce/query/help/					
list	/api/core/ce/query/list/	<a href="#">default</a>	columns, country, country_code, description, draw, endpoint, federation, flavour, id, is_monitored, jobmanager, length, name, order, preset, queue, queues, rcsite, rcsite_state, start, state, status, version	country, country_code, federation, flavour, id, name, queue, rcsite, rcsite_state, state, status	json	
base URI= <a href="/api/core/downtime/query/">/api/core/downtime/query/</a>						
help	/api/core/downtime/query/help/					