



GOADB Status Update

Gridpp37 – 31/08/2016

George Ryall

David Meredith

Outline

- Introduction:
 - Who are we
 - What is GOCDB
- Information systems evolution
- Custom properties
- Write API
 - Background
 - Description
 - Example
- Other recent developments
 - Forced paging
- Contacts and links to further information



Who are we

George Ryall (ME):

- 0.6 FTE
- Based at RAL
- Started working on GOCDB at the start of the FY (having previously spent 6 months as part of a graduate scheme placement).

Dave Meredith:

- 0.4 FTE
- Based at DL
- Lead developer

Where does our funding come from?

- Operations: egi.eu, match funded by GridPP
- New developments: EGI Engage



What is GOCDDB?

The Short answer:

- Grid Operations Centre DataBase: Goc.egi.eu – a grid topology database.

The Long answer:

- a central registry to record information about the topology of an e-Infrastructure.
- Currently used by EGI and WLCG, with EUDAT running their own instance.
- Information includes entities such as operations centres, sites, services, service endpoints and their downtimes, users, and the relationships between the above.
- The service enforces a number of business rules and defines different grouping mechanisms and object-tagging for the purposes of fine-grained resource filtering.
- A role-based permissions model allows resource owners to manage their own resources user roles.
- Both a user facing web interface and a read programmatic interface are provided.



Information Systems evolution

All LHC VOs rely on the BDII to discover CE queues and related information. However, the amount of information that is actually used by the VOs is very little compared to what the BDII publishes. Moreover, OSG is moving out of the BDII. For this reason, there is a general agreement that the static information needed by LHC VOs could be moved to GOCDB and OIM instead. This would allow WLCG to rely only on GOCDB/OIM to get information from EGI and OSG resources. This would also allow sites not belonging to EGI or OSG and not running a BDII to provide the set of needed attributes in an easy way, avoiding to run a service like the BDII.

The following set of attributes are considered to be sufficient to cover experiment use cases:

| Attribute generic name | GLUE 1 | GLUE 2 | Example |
|------------------------|------------------------------|---------------------------------|--------------------|
| CE endpoint | Included In GlueCEUniqueID | GLUE2EndpointID | ce503.cern.ch:9619 |
| Queue name | Included In GlueCEUniqueID | GLUE2ComputingShareMappingQueue | g=id_2nh_atlas |
| CE type | GlueCEImplementationName | GLUE2EndpointImplementationName | HTCondo=CE |
| CE LRMS type | GlueCEInfoLRMSType | GLUE2ManagerProductName | Condo= |
| Max CPU time | GlueCEPolicyMaxCPUTime | GLUE2ComputingShareMaxCPUTime | 2880 |
| Max Wallclock time | GlueCEPolicyMaxWallClockTime | GLUE2ComputingShareMaxWallTime | 2880 |

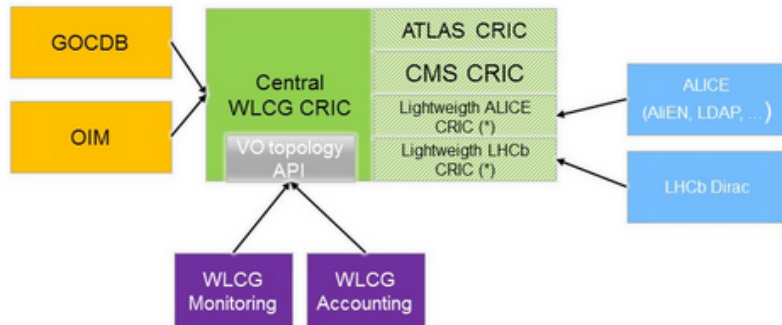
The IS TF will investigate how to move this information from the BDII to GOCDB/OIM taking the following actions:

- The GLUE2 name of the attributes will be used
- The GLUE Enumerated types of values will be used for the types
- Identify a set of volunteer sites to define the new attributes in GOCDB and OIM
- Using the web UI, use extension properties to define key/value pairs matching the list of attributes
 - GOCDB: a writable API has been requested and is under discussion
- Ask AGIS to consume this information from GOCDB/OIM instead of the BDII

After this exercise is carried out, IS TF will be in a position to evaluate whether it's a feasible choice and plan for a wide deployment if necessary.

Monitoring

WLCG monitoring relies on the BDII to be able to get the queue names for the CEs that need to be tested according to the experiments VO feeds. There is a proposal to improve the current VO feed so that it directly includes the queue name on it. However, with the new CRIC, it seems reasonable to get this information directly from CRIC, as shown in the picture below.



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

- Will require more information to be stored in the GOCDB, some of which may need updating more frequently.
- The additional information can be stored using the recently developed custom properties feature.

- Andrew McNab will be presenting an update on this tomorrow morning.
- General aim is to move WLCG away from reliance on the BDII.

Custom Properties

Service: svr009.gla.scotgrid.ac.uk - ARC-CE Delete Edit

ARC CE

System

| | |
|------------------|--|
| Host name | svr009.gla.scotgrid.ac.uk |
| IP Address | 130.209.239.9 |
| IP v6 Address | |
| Operating System | |
| Architecture | |
| Contact E-Mail | uki-scotgrid-glasgow@physics.gla.ac.uk |

Grid Information

| | |
|-------------|--|
| Host | /C=UK/O=eScience/OU=Glasgow |
| DN | /L=Compserv/CN=svr009.gla.scotgrid.ac.uk |
| URL | |
| Parent Site | UKI-SCOTGRID-GLASGOW |
| Scope Tags | atlas, cms, EGI, lhcb, tier2, wlog |

Project Data

| | |
|------------------|---|
| Production Level | ✓ |
| Beta | ✗ |
| Monitored | ✓ |

Service Groups this Service Belongs To

Service Endpoints (endpoints?)

| Name | URL | Interface Name | Edit | Remove |
|--------------------------------|-----|----------------|------|--------|
| + Add Endpoint | | | | |

Extension Properties Export all properties

| Name | Value | Edit | Select All |
|---------------------------------|---------------------------|------|--------------------------|
| GLUE2EndpointID | svr009.gla.scotgrid.ac.uk | | <input type="checkbox"/> |
| GLUE2ComputingShareMappingQueue | condor_q2d | | <input type="checkbox"/> |
| GLUE2EndpointImplementationName | ARC-CE | | <input type="checkbox"/> |
| GLUE2ManagerProductName | Condor | | <input type="checkbox"/> |
| GLUE2ComputingShareMaxCPUTime | 2880 | | <input type="checkbox"/> |
| GLUE2ComputingShareMaxWallTime | 2880 | | <input type="checkbox"/> |

[+ Add Properties](#) Select action... Submit

Add Service Property

Property Name

Property Value

```
GLUE2EndpointID = svr009.gla.scotgrid.ac.uk
GLUE2ComputingShareMappingQueue = condor_q2d
GLUE2EndpointImplementationName = ARC-CE
GLUE2ManagerProductName = Condor
GLUE2ComputingShareMaxCPUTime = 2880
GLUE2ComputingShareMaxWallTime = 2880
```

No file selected.

- The custom properties feature allows multiple key-value pairs to be associated with several types of GOCDB entity.
- These have been used to prototype WLCG requirements for new attributes.
- Multiple properties can be added through the web interface at once.

Write API Background

- GOCDDB's current API provides a variety of views of the information within it (e.g. https://goc.egi.eu/gocdbpi/public/?method=get_site).
- It is extensively documented here:
https://wiki.egi.eu/wiki/GOCDDB/PI/Technical_Documentation.
- Reducing the reliance on the BDII is likely to involve increasing the frequency of updates to the properties of some entities in GOCDDB.
- It may be more appropriate to make these updates through a write API.
- A write API is under development.
- That development work is documented here:
https://wiki.egi.eu/wiki/GOCDDB/Write_API_Development
- Whilst the current development code for it can be found on GitHub:
<https://github.com/GRyall/gocdb/tree/write-api>
- An alpha release for initial user testing is expected W/C September 19th.



Write API

- Initially only supporting changes to extension properties for sites, services, and service endpoints. Though it is written to be easily extended to allow changes to other properties if required.
- URL of the form .../v5/<entityType>/<entityID>/ExtensionProperties/, with the request contents and HTTP method determining the action (PUT, POST and DELETE supported, with GET a likely future development).
- Currently all requests must be authenticated using a x509 certificate
- It is envisaged that site admins will maintain an access control list of authorised client DNs through the web portal.
- Output and input is in JSON format, with appropriate HTTP response codes for different errors.
- A list of actions that are planned to be supported in the initial release can be found here:
https://wiki.egi.eu/wiki/GOCDB/Write_API_Development#Initially_supported_urls.



Write API - example

- A typical request might be a POST request to “*../v5/Service/5682/ExtensionProperties*” with the contents of the request being:
 - ```
{"GLUE2EndpointID":"arc-ce04.gridpp.rl.ac.uk",
 "GLUE2EndpointImplementationName":"ARC-CE"}
```
- This would be made using a certificate with a DN in the list of authorised certificates for the sites that the service with 5682 belongs to.
- This would Add the extension properties in the request to the service with id 5682, and fail if there were already extension properties defined with either of the keys (PUT has a different behaviour).
- A PUT request would have overwritten the existing value if a pair with the requested key already existed.



# Write API

A list of actions that are planned to be supported in the initial release can be found here: [https://wiki.egi.eu/wiki/GOCDB/Write\\_API\\_Development#Initially\\_supported\\_urls](https://wiki.egi.eu/wiki/GOCDB/Write_API_Development#Initially_supported_urls)

## Initially supported urls

The following table shows the methods and URLs that we plan to implement

### Site methods

| Method | URL                                                         | Request Body                               | Function                                                                                                                                                                               |
|--------|-------------------------------------------------------------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| POST   | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Adds the extension properties defined in the request to the Service with the given ID (fails if any of the extension properties are already defined)                                   |
| PUT    | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Replaces the extension properties for the service with the given ID with those in the request (If none of them are currently defined, the it is functionally equivalent to POST)       |
| DELETE | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Removes all the extension properties specified in the request body for the identified service (fails if any of the listed properties are not currently defined)                        |
| DELETE | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | None or empty                              | Removes all the extension properties for the identified service                                                                                                                        |
| POST   | <apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name> | Value in JSON Format e.g Note 2            | Adds an extension property of the named service with the named name and value in the request (fails if property with that name is already defined)                                     |
| PUT    | <apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name> | Value in JSON Format e.g Note 2            | Updates the extension property of the named service with the named name and value in the request (If none of that name is already defined, then it is functionally equivalent to POST) |
| DELETE | <apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name> | None or empty                              | Removes the named extension property from the named service (fails if request body is present, or the named property is not defined for the identified site)                           |

### Service methods

| Method | URL                                                         | Request Body                               | Function                                                                                                                                                                               |
|--------|-------------------------------------------------------------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| POST   | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Adds the extension properties defined in the request to the service with the given ID (fails if any of the extension properties are already defined)                                   |
| PUT    | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Replaces the extension properties for the service with the given ID with those in the request (If none of them are currently defined, the it is functionally equivalent to POST)       |
| DELETE | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Removes all the extension properties specified in the request body for the identified service (fails if any of the listed properties are not currently defined)                        |
| DELETE | <apiurl>/v5/Service/<Service ID>/ExtensionProperties        | None or empty                              | Removes all the extension properties for the identified service                                                                                                                        |
| POST   | <apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name> | Value in JSON Format e.g Note 2            | Adds an extension property of the named service with the named name and value in the request (fails if property with that name is already defined)                                     |
| PUT    | <apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name> | Value in JSON Format e.g Note 2            | Updates the extension property of the named service with the named name and value in the request (If none of that name is already defined, then it is functionally equivalent to POST) |
| DELETE | <apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name> | None or empty                              | Removes the named extension property from the named service (fails if request body is present, or the named property is not defined for the identified site)                           |

### Service Endpoint methods

| Method | URL                                                            | Request Body                               | Function                                                                                                                                                                          |
|--------|----------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| POST   | <apiurl>/v5/Endpoint/<End point ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Adds the extension properties defined in the request to the SE with the given ID (fails if any of the extension properties are already defined)                                   |
| PUT    | <apiurl>/v5/Endpoint/<End point ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Replaces the extension properties for the SE with the given ID with those in the request (If none of them are currently defined, the it is functionally equivalent to POST)       |
| DELETE | <apiurl>/v5/Endpoint/<End point ID>/ExtensionProperties        | Key value pairs in JSON Format e.g. Note 1 | Removes all the extension properties specified in the request body for the identified SE (fails if any of the listed properties are not currently defined)                        |
| DELETE | <apiurl>/v5/Endpoint/<End point ID>/ExtensionProperties        | None or empty                              | Removes all the extension properties for the identified SE                                                                                                                        |
| POST   | <apiurl>/v5/Endpoint/<End point ID>/ExtensionProperties/<name> | Value in JSON Format e.g Note 2            | Adds an extension property of the named SE with the named name and value in the request (fails if property with that name is already defined)                                     |
| PUT    | <apiurl>/v5/Endpoint/<End point ID>/ExtensionProperties/<name> | Value in JSON Format e.g Note 2            | Updates the extension property of the named SE with the named name and value in the request (If none of that name is already defined, then it is functionally equivalent to POST) |
| DELETE | <apiurl>/v5/Endpoint/<End point ID>/ExtensionProperties/<name> | None or empty                              | Removes the named extension property from the named SE (fails if request body is present, or the named property is not defined for the identified SE)                             |

Note 1: [{"PROPERTY1NAME": "PROPERTY1VALUE", "PROPERTY2NAME": "PROPERTY2VALUE", "PROPERTY3NAME": "PROPERTY3VALUE"}]

Note 2: [{"value": "PROPERTYVALUE"}]

# Other Recent Developments

Whilst the Write API is currently a major part of recent work, we have also recently completed a number of other developments:

- Forced of existing API queries – see the next slide.
- Dual stack IPv6 test instance (<https://ip6tb-goc.gridpp.rl.ac.uk/portal/>).
- Downtime calendar.
- Ability to declare downtimes in local timezone.



# Forced Paging

- The next release of GOCDDB will enforce paging of results on most existing API queries.
- This is cursor, not offset, paging.
- More details, including links to a test server with this code on, can be found here:

<https://wiki.egi.eu/wiki/GOCDB/notifications#v5.7>

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <results>
3 <meta>
4 <link rel="self" href="https://gocdb-test.esc.rl.ac.uk/gocdbpi/public/?method=get_downtime_nested_services"/>
5 <link rel="next" href="https://gocdb-test.esc.rl.ac.uk/gocdbpi/public/?method=get_downtime_nested_services&next_cursor=447"/>
6 <link rel="prev" href="https://gocdb-test.esc.rl.ac.uk/gocdbpi/public/?method=get_downtime_nested_services&prev_cursor=1"/>
7 <link rel="start" href="https://gocdb-test.esc.rl.ac.uk/gocdbpi/public/?method=get_downtime_nested_services&next_cursor=0"/>
8 <count>400</count>
9 <max_page_size>400</max_page_size>
10 </meta>
11 <DOWNTIME ID="1" PRIMARY_KEY="38707G0" CLASSIFICATION="UNSCHEDULED">
12 <SEVERITY>OUTAGE</SEVERITY>
13 <DESCRIPTION>Service maintenance</DESCRIPTION>
14 <INSERT_DATE>1291626720</INSERT_DATE>
15 <START_DATE>1291593600</START_DATE>
16 <END_DATE>1292112000</END_DATE>
17 <FORMATED_START_DATE>2010-12-06 00:00</FORMATED_START_DATE>
18 <FORMATED_END_DATE>2010-12-12 00:00</FORMATED_END_DATE>
19 <GOCDB_PORTAL_URL>https://gocdb-test.esc.rl.ac.uk/portal/index.php?Page_Type=Downtime&id=1</GOCDB_PORTAL_URL>
20 <SERVICES>
21 <SERVICE>
22 <PRIMARY_KEY>2831G0</PRIMARY_KEY>
23 <HOSTNAME>glite-ce.grid.itb.ac.id</HOSTNAME>
24 <SERVICE_TYPE>CREAM-CE</SERVICE_TYPE>
25 <ENDPOINT>glite-ce.grid.itb.ac.idCREAM-CE</ENDPOINT>
26 <HOSTED_BY>ID-ITB</HOSTED_BY>
27 <AFFECTED_ENDPOINTS/>
28 </SERVICE>
29 </SERVICES>
30 </DOWNTIME>
31 <DOWNTIME ID="2" PRIMARY_KEY="36939G0" CLASSIFICATION="UNSCHEDULED">
32 <SEVERITY>OUTAGE</SEVERITY>
33 <DESCRIPTION>Service maintenance and update new host certificates </DESCRIPTION>
34 <INSERT_DATE>1289540220</INSERT_DATE>
35 <START_DATE>1289520000</START_DATE>
```



# Questions?

## Information and contacts:

- Contact:  
[gocdb-admins@mailman.egi.eu](mailto:gocdb-admins@mailman.egi.eu)
- Discussion list (mainly tool developers using the PI):  
[gocdb-discuss@mailmain.egi.eu](mailto:gocdb-discuss@mailmain.egi.eu)
- Support:  
<https://ggus.eu/>
- Source code:  
<https://github.com/GOCDB/gocdb>
- Requirements tracking:  
<https://rt.egi.eu/rt/Dashboards/5541/GOCDB-Requirements>
- Documentation:  
[https://wiki.egi.eu/wiki/GOCDB/Documentation\\_Index](https://wiki.egi.eu/wiki/GOCDB/Documentation_Index)

