

# CRR json

---

A. Forti

WLCG IS TF

22 November 2018



# General

- 4 sites:
  - RAL, Manchester, Liverpool, Glasgow
  - Freedom to set things up as each site preferred
    - As many ways as sites
  - Tried 2 formats
    - Flat and Hierarchical
    - Some confusion on the attributes
- One of the aims to move to json is to give us the ability describe non standard grid resources.
  - Asked to represent VAC resources as well



# GOCDDB

- Different views
  - Glasgow and Liverpool and RAL global json added the GOCDDB link at site level minimal effort one link for everything
    - Glasgow: 1 batch system, 1 queue, and multiple CEs
      - [https://goc.egi.eu/portal/index.php?Page\\_Type=Site&id=627](https://goc.egi.eu/portal/index.php?Page_Type=Site&id=627)
    - Liverpool 2 batch systems each with 1 queue and 2 CEs
      - [https://goc.egi.eu/portal/index.php?Page\\_Type=Site&id=232](https://goc.egi.eu/portal/index.php?Page_Type=Site&id=232)
    - RAL: 1 batch system, 1 queue, 4 CEs
      - [https://goc.egi.eu/portal/index.php?Page\\_Type=Site&id=570](https://goc.egi.eu/portal/index.php?Page_Type=Site&id=570)
  - Manchester 1 json per cluster attached to each CE
    - 2 batch systems, 3 queues each, 2 CEs
      - [https://goc.egi.eu/portal/index.php?Page\\_Type=Service&id=2695](https://goc.egi.eu/portal/index.php?Page_Type=Service&id=2695)
      - [https://goc.egi.eu/portal/index.php?Page\\_Type=Service&id=2696](https://goc.egi.eu/portal/index.php?Page_Type=Service&id=2696)



# Attributes

- Not much comment about the attributes themselves
- Some misunderstanding on which set of attributes
  - Glasgow had an old schema
  - RAL, Manchester and Liverpool followed the google doc one
- Google doc better defined
  - “resourcename” vs “id”
  - Could add “resourcetype”: batch\_system, vac, cloud, boinc,.....
  - Mechanism to extend attributes
    - Mandatory vs optional: “gpu”, “MPI”, “OS”,.....
    - In the hierarchical model: “queues”, “computingelements”
  - Could add prefixes to better identify what things are for
    - res\_, ce\_, qu\_, gl\_
    - Nobody really liked it, preferred hierarchical structure



# Flat vs Hierarchical

- Flat json were provided but considered “broken” with the same issues as Glue and REBUS.
  - Even if we can make it work
- Consensus that 1 global file with a hierarchical structure
  - Pros
    - Minimises repetition and inconsistencies
    - Better describes different type of resources
      - See Liverpool example with VAC
    - Decouples resources attributes from CE services
    - Minimises links in GOCDB
  - Cons
    - We cannot get the CEs to write them
    - Dedicated CEs pointing to the same batch system add some complication with the ACLs: queues and CEs have to be matched
      - Is it a real use case?



# Misc

- Language
  - We keep on talking about json but over the years has seen the advent of YAML
    - This would further simplify writing these files
    - Glasgow wrote a hierarchical example in YAML
- Physical location of the files
  - Interesting to note nobody even tried to put the jsons on the CEs.
    - All used either an http server or gitlab



# Conclusions

- This is would make publishing much easier once some rules are established
  - Writing a json takes about 15-20 minutes if one wants to check things
- Favourite structure: hierachical single file that can be added at site level in GOCDB
- Attributes should be extendable
  - Mandatory vs optional
- Language: YAML would be even simpler than json
- Need an information consumer
- Will have more dedicated meetings with other sites in the future



# Links

- RAL:
  - Flat json
- Glasgow:
  - Flat json
  - Hierarchical json
  - Hierarchical YAML
- Liverpool:
  - Flat json
  - Hierarchical json
    - Includes VAC resources
- Manchester
  - ce01 flat json
  - ce02 flat json
  - Hierarchical json

