



# LHCb infosystem plans

**Andrew McNab**  
University of Manchester  
LHCb and GridPP

# Quick overview

- LHCb uses DIRAC for all compute and storage tasks
  - Production runs, data management, user analysis jobs etc etc are all managed through DIRAC and layers above FTS, CREAM etc.
- The DIRAC services and agents get their view of the world from the DIRAC Configuration Service (CS)
  - CS info originally comes from multiple programmatic sources and manual edits
- So the LHCb information system problem becomes
  - What is needed in the DIRAC CS?
  - How to populate the DIRAC CS?
- This talk explains what we do now (GLUE1 Top BDII, GOCDDB)
- And how we are planning to extend it (GLUE2.0, direct resource BDII queries, more GOCDDB, direct GLUE2.0/JSON/HTTPS queries?)
- A lot of details taken from Andrea Valassi's contribution to the InfoSys TF Use Cases, which also enumerates the GLUE1 values we use.

# Bdii2CSAgent

- Bdii2CSAgent uses a Top BDII to collect GLUE1 details of all sites and compares what is already in the DIRAC CS
  - For new sites supporting LHCb, an email prompt is sent to suggest adding the site
  - For existing sites, the CS information is updated
  - Both CE level (queue lengths etc) and site level (ops emails etc) information is collected
- Evolution to be discussed at LHCb Computing Workshop next week
  - Decision already taken to add GLUE2.0 support to DIRAC
  - May begin using functions that get storage usage from BDII rather than SRM as now
  - What about information duplicated in the GOCDB?

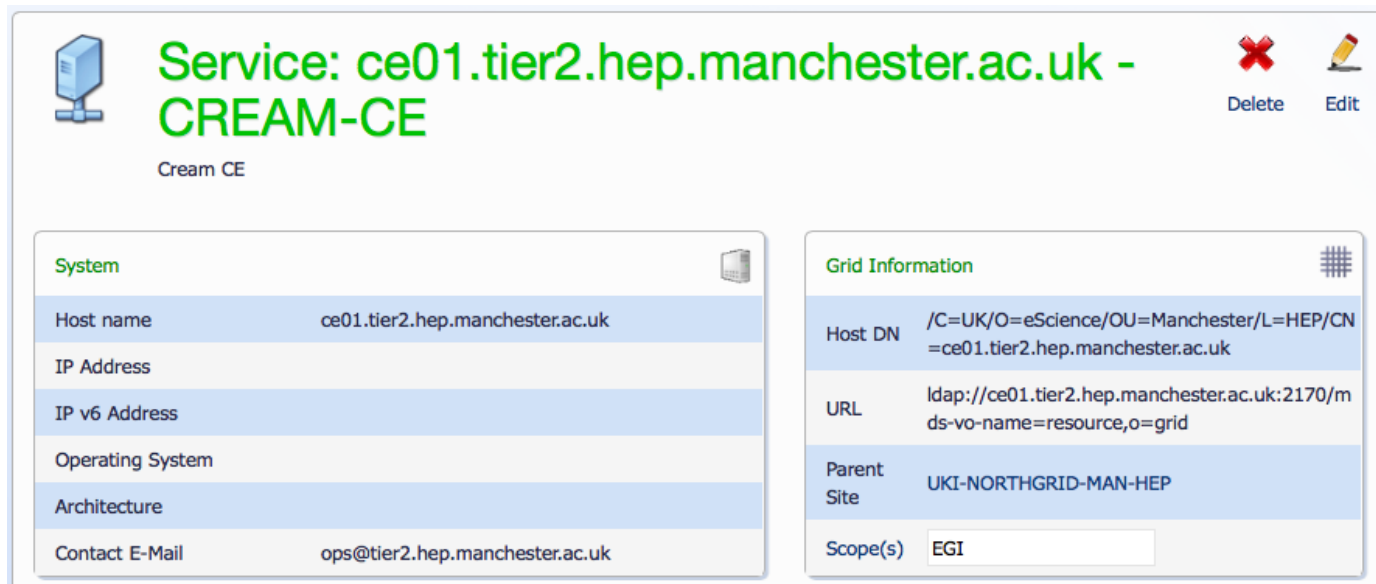


# GOCDDB2CSAgent

- Parallels Bdi2CSAgent
  - Uses GOCDDB to populate some DIRAC CS values
- Currently just PerfSonar endpoints
- Framework can be extended in the future
  - Site ops emails, lat/lon, description?
- How to merge with Bdi2CSAgent information?
  - Which version wins if there are duplicates?
- What about other similar sources of information?
  - For example OSG's OIM
- Again, to be discussed during LHCb Computing Workshop

# Direct queries to resource BDIIs

- The BDII (slapd) running on a resource (eg CREAM CE) can be queried directly
  - Results are equivalent to records in the Top or Site BDII
- GOCDB already has entries for individual resources so that fine-grained downtimes can be declared
- So put the LDAP URL of the resource BDII into the GOCDB service record, using the “Grid Information : URL” field



The screenshot shows a GOCDB service record for 'ce01.tier2.hep.manchester.ac.uk - CREAM-CE'. The record is titled 'Service: ce01.tier2.hep.manchester.ac.uk - CREAM-CE' and is categorized as 'Cream CE'. It features two main sections: 'System' and 'Grid Information'. The 'System' section includes fields for Host name, IP Address, IP v6 Address, Operating System, Architecture, and Contact E-Mail. The 'Grid Information' section includes fields for Host DN, URL, Parent Site, and Scope(s). The URL field is highlighted, showing the LDAP URL: 'ldap://ce01.tier2.hep.manchester.ac.uk:2170/m ds-vo-name=resource,o=grid'. The Parent Site is 'UKI-NORTHGRID-MAN-HEP' and the Scope(s) is 'EGI'. There are 'Delete' and 'Edit' icons in the top right corner.

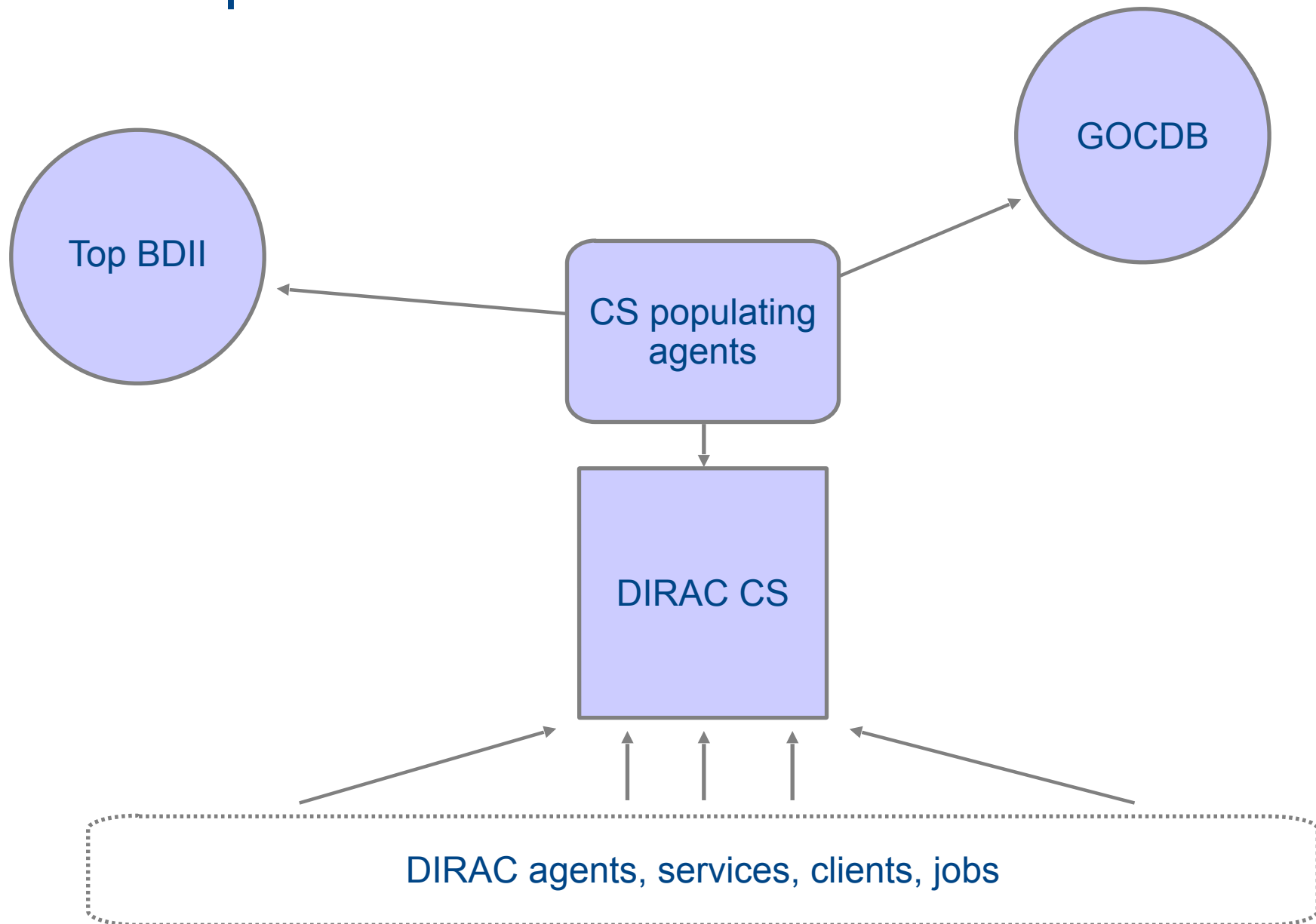
System	
Host name	ce01.tier2.hep.manchester.ac.uk
IP Address	
IP v6 Address	
Operating System	
Architecture	
Contact E-Mail	ops@tier2.hep.manchester.ac.uk

Grid Information	
Host DN	/C=UK/O=eScience/OU=Manchester/L=HEP/CN=ce01.tier2.hep.manchester.ac.uk
URL	ldap://ce01.tier2.hep.manchester.ac.uk:2170/m ds-vo-name=resource,o=grid
Parent Site	UKI-NORTHGRID-MAN-HEP
Scope(s)	EGI

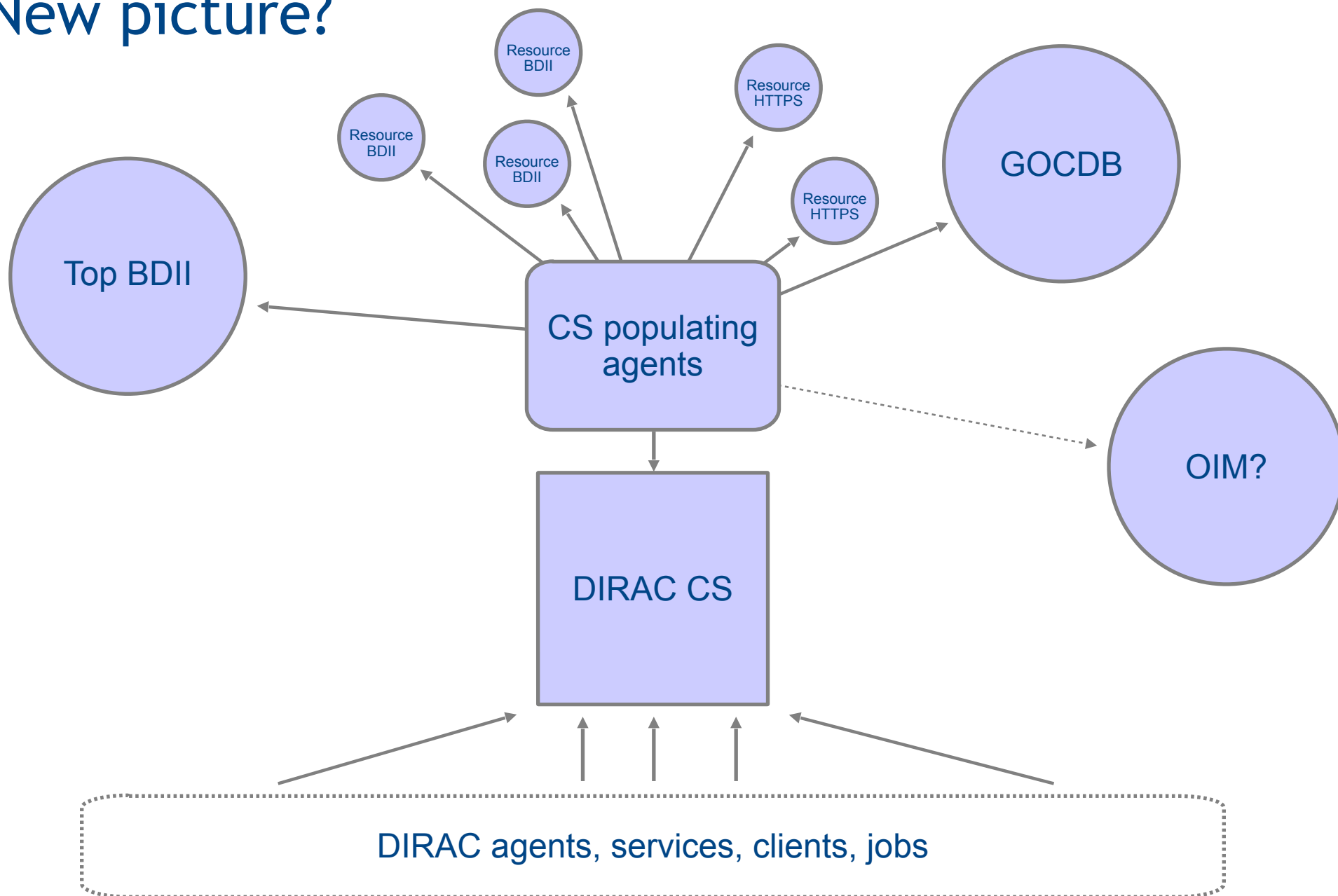
# LDAP URL aware Bdi2CSAgent

- LHCb has done a proof of concept of this approach in the Python modules called by Bdi2CSAgent
  - Go through all the CREAM and ARC CEs services in GOCDB, looking for ones with the URL field set
  - For each of these, if URL is like ldap://...o=grid then use DIRAC ldapCEState function for querying and parsing
  - Add the results to the same big dictionary of Top BDII results that Bdi2CSAgent processes
  - If the parent site is not already there, add the ops emails, lat/lon, and description from GOCDB too
  - Job done!
- Production quality implementation of this method will be part of the GLUE 2.0 support rewrite of how DIRAC CS is populated
- URL field could be something other than LDAP: eg GLUE2/JSON/HTTPS

# Current picture



# New picture?







# Summary

- For LHCb, it's all about how the DIRAC CS is populated
  - The rest of DIRAC just sees the CS
- LHCb planning GLUE 2.0 BDII support
  - Also extending use of GOCDB
  - And whether to use storage usage from BDII rather than SRM
- Proof of concept has been produced that also populates DIRAC CS with GLUE1 information directly from resource BDIIs
  - Using LDAP URLs in service's details in GOCDB
  - Won't go into production using GLUE1 though
- This approach can readily be extended to services like Vac/Vcycle which will publish in GLUE2/JSON/HTTPS
  - See next talk for GridPP work on this