Information System usage by the EGI Operational Tools

D.Cesini (INFN/IGI)

On behalf of the EGI-JRA1 activity

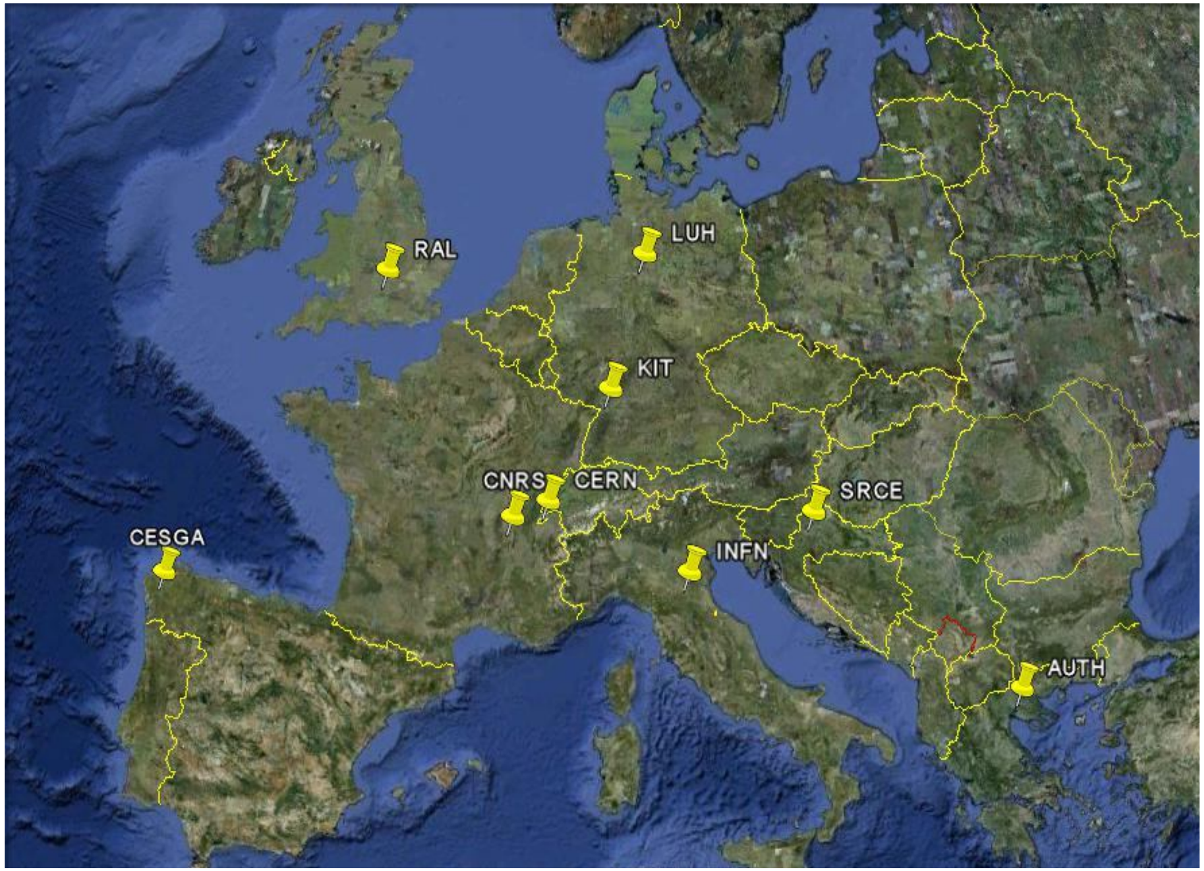
Towards an Integrated Information System Workshop

Amsterdam 01/12/2011

Outline

- The EGI-JRA1 Activity
  - Developed tools and other tasks
- The GOCDB
  - Tool presentation
  - Tool usage statistics
- The ops current usage of the Information System and GOCDB
- Conclusion

Tools developed by EGI-JRA1



- Operation Portal (CNRS)
- EGI Helpdesk (KIT)
- GOCDB (RAL)
- Accounting Repository (RAL)
- Accounting Portal (CESGA)
- SAM/MyEGI (CERN/SRCE)
- Metrics Portal (CESGA)

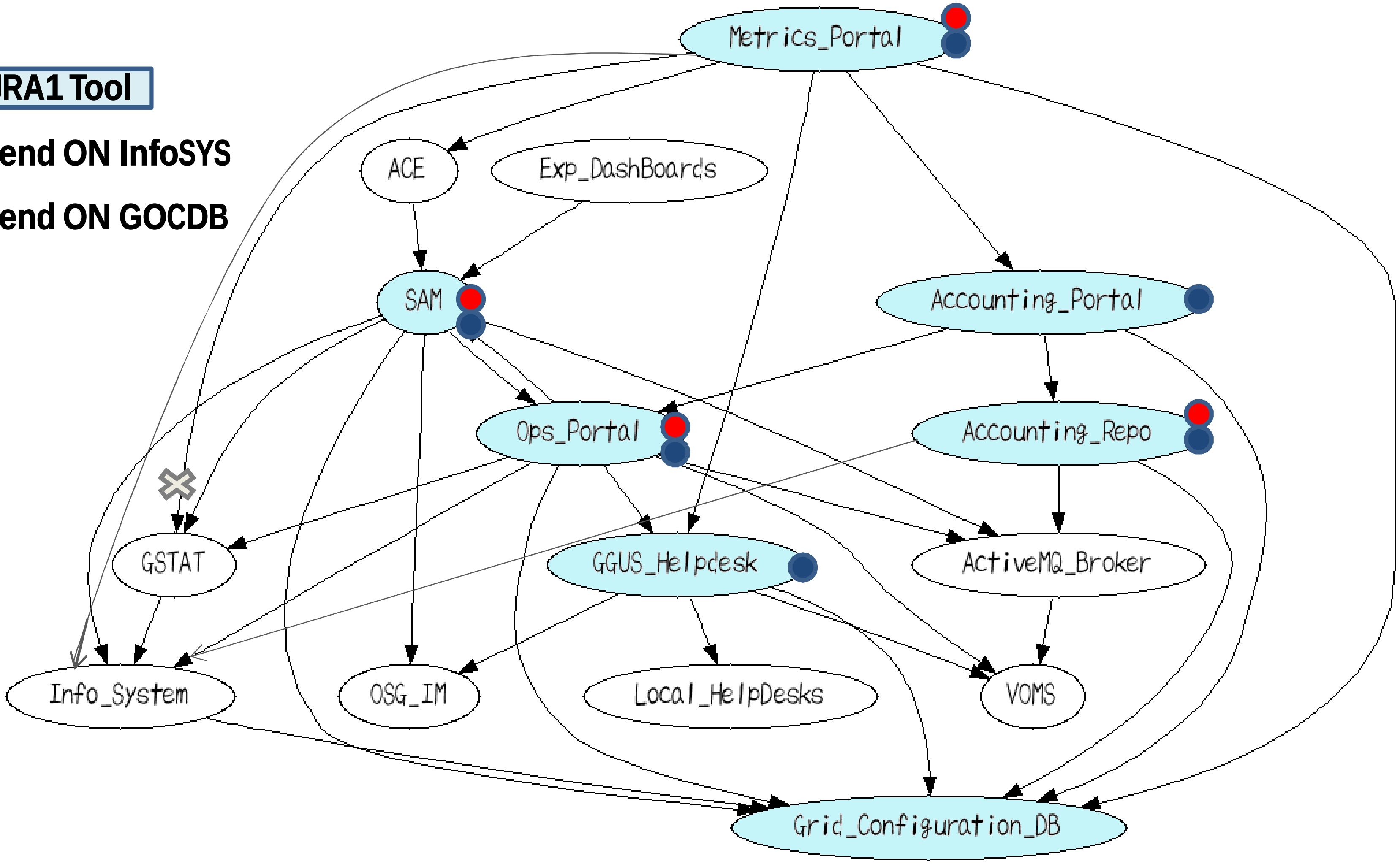
Other activity tasks

- Message Broker Configuration to support tools and operation (AUTH)
- Accounting for different resource types (LUH/INFN/RAL)
  - Billing
  - Accounting of application usage
  - Accounting of data usage
  - Accounting of capacity and cloud computing usage

JRA1 Tools & Dependencies

EGI-JRA1 Tool

- 🔴 Depend ON InfoSYS
- 🔵 Depend ON GOCDB



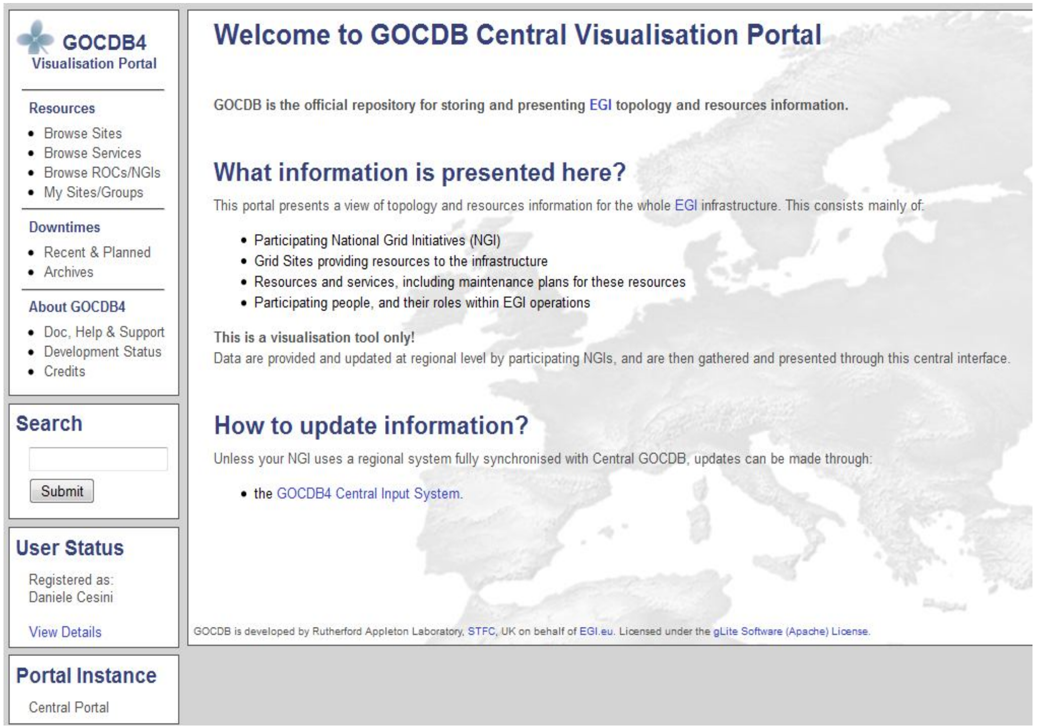
Source MS704 + fixes

GOCDB

GOCDB

- EGI relies on a central configuration database to record “static” information

- Contributed by the resource providers
  - service instances that they are running
  - the individual contact, role and status information for those responsible for particular services



Welcome to GOCDB Central Visualisation Portal

GOCDB is the official repository for storing and presenting EGI topology and resources information.

What information is presented here?

This portal presents a view of topology and resources information for the whole EGI infrastructure. This consists mainly of:

- Participating National Grid Initiatives (NGI)
- Grid Sites providing resources to the infrastructure
- Resources and services, including maintenance plans for these resources
- Participating people, and their roles within EGI operations

This is a visualisation tool only!
Data are provided and updated at regional level by participating NGIs, and are then gathered and presented through this central interface.

How to update information?

Unless your NGI uses a regional system fully synchronised with Central GOCDB, updates can be made through:

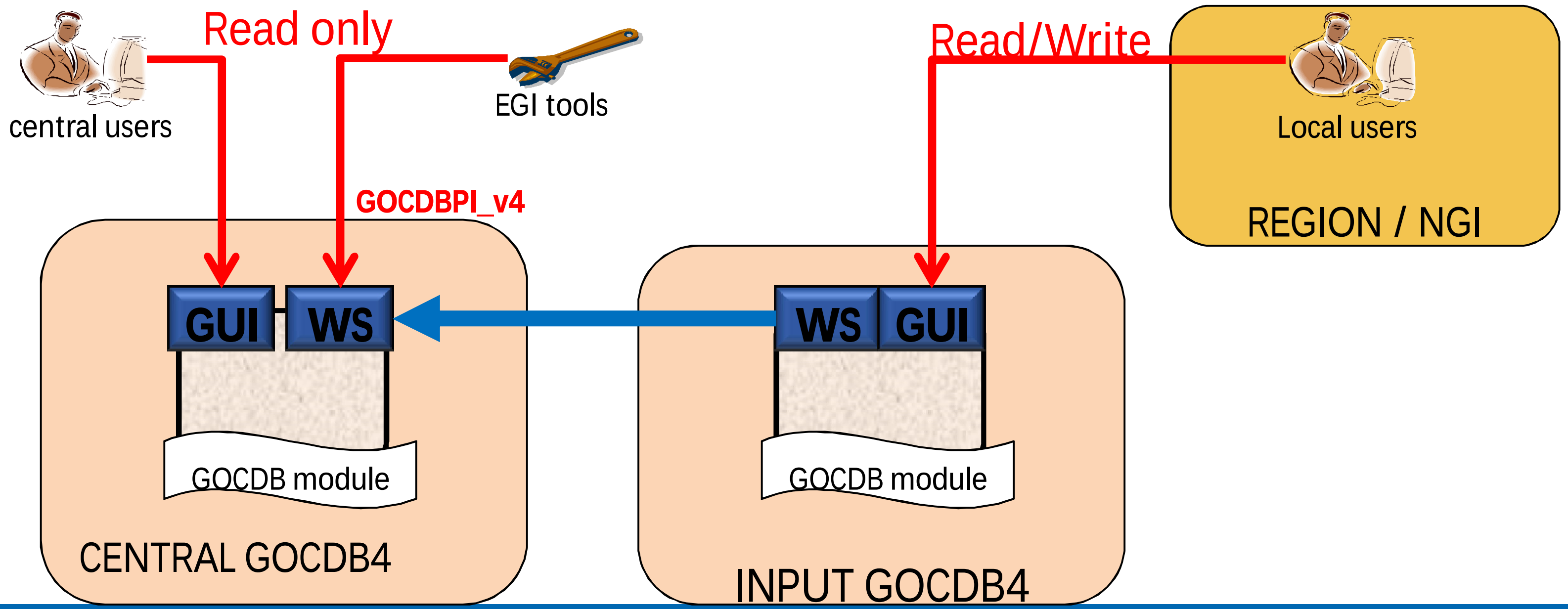
- the GOCDB4 Central Input System.

GOCDB is developed by Rutherford Appleton Laboratory, STFC, UK on behalf of EGI.eu. Licensed under the gLite Software (Apache) License.

GOCDB

- GOCDB is currently hosted centrally and can store both EGI data and Locally scoped data not intended to be visible to EGI
  - Regional Standalone GOCDB available/maintained for NGI willing to install it
  - Regional Publishing GOCDB not available - is a long term development
- GOCDB comes along in two flavors, a *read-only* "Central Visualisation Portal" and a *writeable* "Input System"

https://wiki.egi.eu/wiki/GOCDB/Release4/Architecture



GOCDB Architecture

- A three tiered web application
  - a web GUI for manually recording Grid topology information
  - a REST style API for querying that data in XML
  - Data stored in an Oracle database

- GOCDB does not provide a writable programmatic interface for create/update/delete
  - Rather, the main CRUD interface is the web portal which provides control to users in defining their topology

GOCDB Data Model

- GOCDB records topology information and also its associated state
  - i.e. recording what sites/services/downtimes are meant to be functioning at a particular snapshot in time
- The information is largely hierarchical;
  - a parent NGI (group object) aggregates many child Site objects which in turn aggregate child Service Endpoints.
- Downtime objects are linked to individual service endpoints (not sites)
  - If all of the site's services are in downtime, then the whole site is effectively in downtime.
  - Each of the different entity types have relevant attributes, including people memberships, contact information, service types, downtime start/end dates etc.

GOCDB Permission & Role Model

- The permissions model is hierarchical
  - users with higher level permissions on parent objects (e.g. NGIs) can grant permissions to other users over sibling and child objects (e.g. Sites)
  - following the creation of a user with a top level 'NGI manager role,' NGIs can manage and propagate their own users, Sites, and Service Endpoints without involving the GOCDB administrators
  - the roles/permissions model is currently being re-developed to facilitate finer grained permissions
    - In the newly evolving AAA model, the hierarchical permission model is less obvious
      - https://wiki.egi.eu/w/images/b/b7/FinerGrainedGOCDB_rolesVeraProposal2.xls

GOCDB Stateful Data and State Transition Rules

- GOCDB has predefined state transitions rules
  - sites have an associated ‘Production’ and ‘Certification’ status.
  - A Site would normally moves from ‘Candidate -> Uncertified -> Certified’ during the certification process
  - other transitions are strictly forbidden.
    - Similar state transition rules also exist when declaring and editing downtimes and when granting/revoking roles

- https://wiki.egi.eu/wiki/GOCDB/Input_System_User_Documentation

GOCDB Historical Data and Auditing

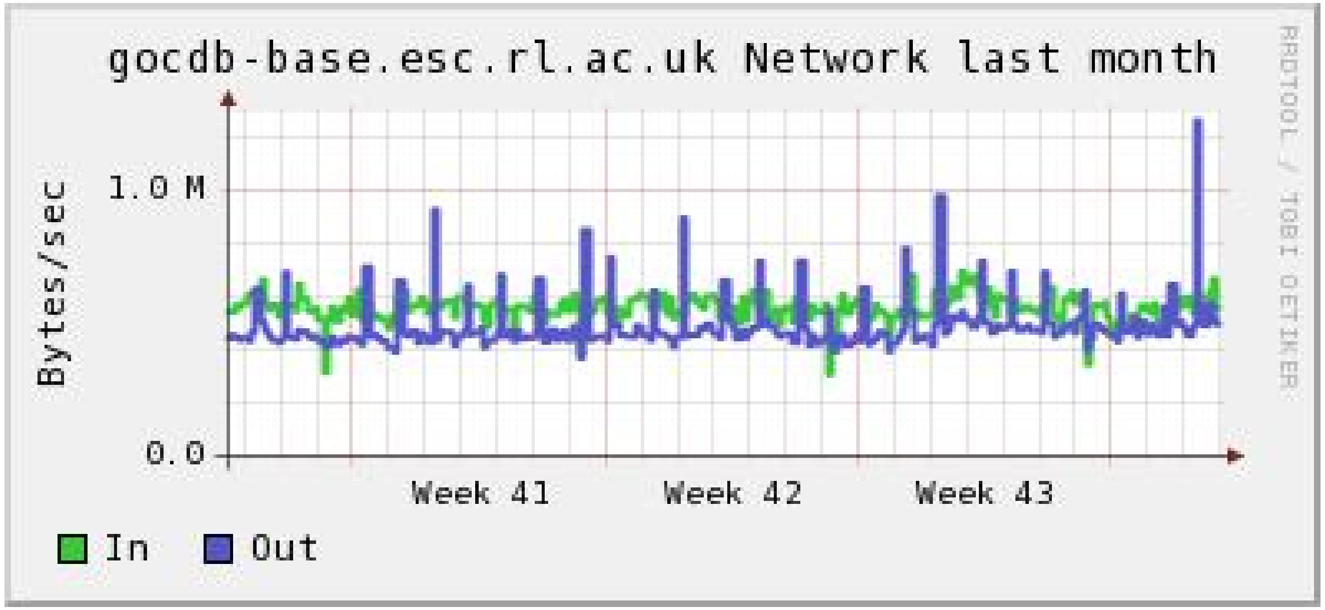
- Data are never actually deleted from the DB
  - Instead, objects have an associated timestamp which indicates whether that object is currently ‘live’ or is deleted
  - By providing timestamp parameters to the PI methods, the PI can be used to query for historical/audit data
    - e.g. list a site’s certification status history
    - who, why and when those changes were applied

GOCDB Usage Statistics (1/3)

PI Statistics on Visualization Portal

- In October 2011 there were nearly 6 million separate queries for the PI:
  - which averages ~2.3 queries per second
  - The query rate has a background steady state averaging 0.4 MB/sec
  - cyclic daily peaks
    - reflecting requests from automated scripts
- Given this high usage, caching the results of popular and expensive queries is essential
  - Selected queries are executed every 30mins or so and the XML results are cached and served by Apache.

GOCDB Usage Statistics (2/3)



- The number of valid PI queries starting with URL fragment '/gocdbpi':
  - Sep=4.3M
  - Oct=5.9M

GOCDB Usage Statistics (3/3)

User interactions on the read/write input portal

- Usage of the read/write input portal can also be considered as high
  - The total number of separate page requests, excluding noise such as css and image file requests is:
    - Sep=8566
    - Oct=8974 (for Oct, this averages 289 user interactions per day).
  - A partial selection of different page requests for October and September are:

# TOTAL GET/POST requests (excluding noise) (Sep=8566, Oct=8974)

# POST new downtimes (Sep=391, Oct=457)

# POST edit downtimes (Oct=175, Sep=148)

# View object (Sep=3854, Oct=3975)

GOCDB Documentation

- **Main Entry Point: https://goc.egi.eu**

- https://wiki.egi.eu/wiki/GOCDB
- https://wiki.egi.eu/wiki/GOCDB/Documentation_Index
- https://wiki.egi.eu/wiki/GOCDB/Release4

GOCDB BACKUP

GOCDB PROM (1/3)

- PROM is a proprietary Object Relational Mapping (ORM) style solution for persisting data as objects in a relational database, and importantly, for recording parent/child relationships (links) between those objects *without* defining DDL schema constraints at the database-schema level (relationships are traditionally enforced in a relational database using PK/FK constraints).

- By excluding DDL schema constraints, a PROM database can accommodate schema changes *without* affecting existing software or data.

- For GOCDB, the intention is to allow NGIs to add their own custom data objects to a local installation whilst leaving the core GOCDB objects intact.

- While the PROM methodology certainly does provide flexibility, it also increases complexity. This is especially true when querying for object relationships. This is because queries must repeatedly join across the core ‘admin’ PROM tables in order to determine those relationships before those objects can be returned as linked object-graphs.

GOCDB PROM (2/3)

- A PROM database requires a standard set of core ‘admin’ tables that are specifically used to record information about the different types of data objects, and which of those objects are related (linked). For example, for every new data object, a new entry is recorded in the PROM ‘object list’ table (recording the object id and other meta-data such as the object’s table name).

- Similarly, parent/child relationships between objects are stored as individual entries in the PROM ‘object link’ table (each link entry associates a parent object of a particular type to a child object of a particular type).

- The rules for linking objects are defined in a separate ‘link type’ table. The core PROM admin tables include TOBJECTS, TOBJECTTYPES, TOBJECTLINKS, and TLINK_TYPES. For GOCDB, the custom ‘data’ tables include USER, SITE, SERVICE_ENDPOINT, DOWNTIME GROUP etc (see the logical and physical entity model diagrams).

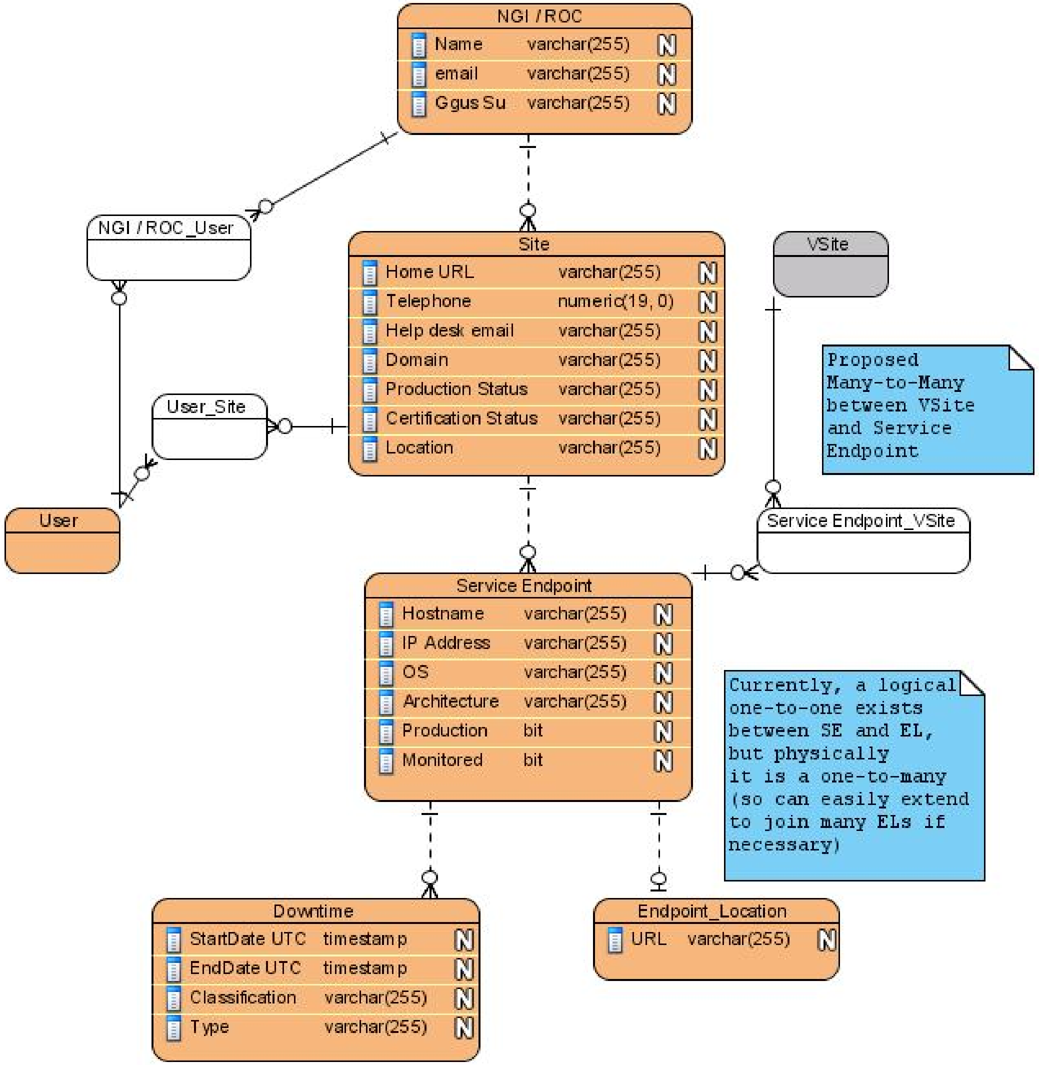
GOCDB PROM (3/3)

- To ease working with a PROM database, a database agnostic ‘IPromAPI’ interface has been defined (introduced in GOCDB v4.2) which can be implemented for different databases as required (currently only implemented in PHP for Oracle).

- The IPromAPI interface is a CRUD style API and simplifies insertion/deletion/linking/updating of objects. Importantly, it also enforces the relationship rules as defined in the ‘link type’ table.

- Since object relationships are not defined in DDL, the ‘Physical PROM DDL’ is distinctly different from the ‘Logical Object Model.’ For more details on PROM see: https://wiki.egi.eu/wiki/GOCDB/PROM

Logical GOCDB Object Model (simplified)



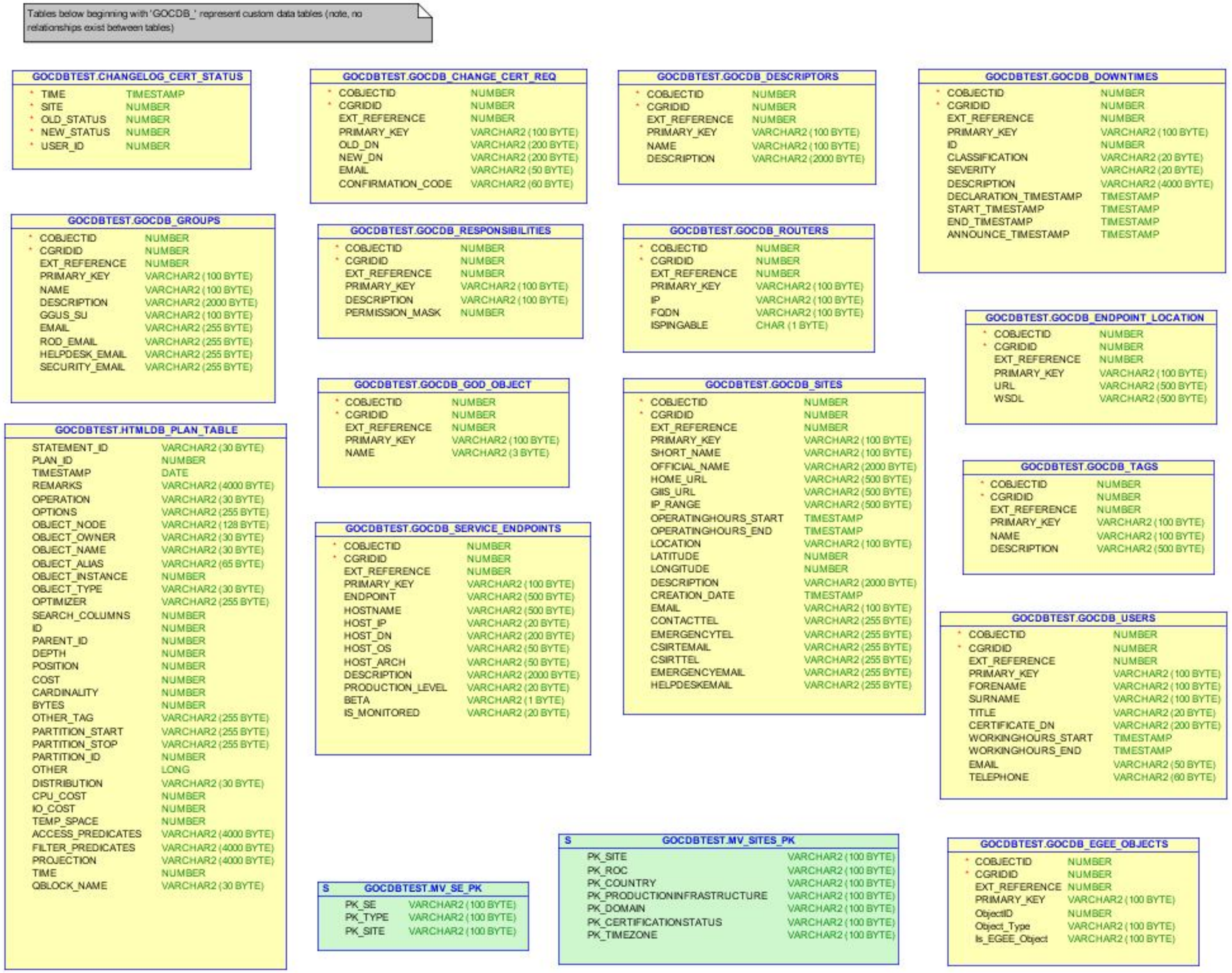
Proposed Many-to-Many between VSite and Service Endpoint

Currently, a logical one-to-one exists between SE and EL, but physically it is a one-to-many (so can easily extend to join many ELs if necessary)

The relationships shown here are *logical* – they are not actually defined in a DDL schema with PK/FK constraints.

In a PROM database, relationships are enforced by the PROM API and the available link types recorded in the PROM admin tables (admin tables not shown, see the GOCDB physical entity model).

Physical GOCDB Entity Model (1/2)



Tables below beginning with 'GOCDB_' represent custom data tables (note, no relationships exist between tables)

GOCDBTEST.CHANGELOG_CERT_STATUS	
* TIME	TIMESTAMP
* SITE	NUMBER
* OLD_STATUS	NUMBER
* NEW_STATUS	NUMBER
* USER_ID	NUMBER

GOCDBTEST.GOCDB_CHANGE_CERT_REQ	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
OLD_DN	VARCHAR2 (200 BYTE)
NEW_DN	VARCHAR2 (200 BYTE)
EMAIL	VARCHAR2 (50 BYTE)
CONFIRMATION_CODE	VARCHAR2 (60 BYTE)

GOCDBTEST.GOCDB_DESCRIPTORS	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
NAME	VARCHAR2 (100 BYTE)
DESCRIPTION	VARCHAR2 (2000 BYTE)

GOCDBTEST.GOCDB_DOWNTIMES	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
ID	NUMBER
CLASSIFICATION	VARCHAR2 (20 BYTE)
SEVERITY	VARCHAR2 (20 BYTE)
DESCRIPTION	VARCHAR2 (4000 BYTE)
DECLARATION_TIMESTAMP	TIMESTAMP
START_TIMESTAMP	TIMESTAMP
END_TIMESTAMP	TIMESTAMP
ANNOUNCE_TIMESTAMP	TIMESTAMP

GOCDBTEST.GOCDB_GROUPS	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
NAME	VARCHAR2 (100 BYTE)
DESCRIPTION	VARCHAR2 (2000 BYTE)
GGUS_SU	VARCHAR2 (100 BYTE)
EMAIL	VARCHAR2 (255 BYTE)
ROD_EMAIL	VARCHAR2 (255 BYTE)
HELPDESK_EMAIL	VARCHAR2 (255 BYTE)
SECURITY_EMAIL	VARCHAR2 (255 BYTE)

GOCDBTEST.GOCDB_RESPONSIBILITIES	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
DESCRIPTION	VARCHAR2 (100 BYTE)
PERMISSION_MASK	NUMBER

GOCDBTEST.GOCDB_ROUTERS	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
IP	VARCHAR2 (100 BYTE)
FQDN	VARCHAR2 (100 BYTE)
ISPINGABLE	CHAR (1 BYTE)

GOCDBTEST.GOCDB_ENDPOINT_LOCATION	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
URL	VARCHAR2 (500 BYTE)
WSDL	VARCHAR2 (500 BYTE)

GOCDBTEST.HTMLDB_PLAN_TABLE	
STATEMENT_ID	VARCHAR2 (30 BYTE)
PLAN_ID	NUMBER
TIMESTAMP	DATE
REMARKS	VARCHAR2 (4000 BYTE)
OPERATION	VARCHAR2 (30 BYTE)
OPTIONS	VARCHAR2 (255 BYTE)
OBJECT_NODE	VARCHAR2 (128 BYTE)
OBJECT_OWNER	VARCHAR2 (30 BYTE)
OBJECT_NAME	VARCHAR2 (30 BYTE)
OBJECT_ALIAS	VARCHAR2 (65 BYTE)
OBJECT_INSTANCE	NUMBER
OBJECT_TYPE	VARCHAR2 (30 BYTE)
OPTIMIZER	VARCHAR2 (255 BYTE)
SEARCH_COLUMNS	NUMBER
ID	NUMBER
PARENT_ID	NUMBER
DEPTH	NUMBER
POSITION	NUMBER
COST	NUMBER
CARDINALITY	NUMBER
BYTES	NUMBER
OTHER_TAG	VARCHAR2 (255 BYTE)
PARTITION_START	VARCHAR2 (255 BYTE)
PARTITION_STOP	VARCHAR2 (255 BYTE)
PARTITION_ID	NUMBER
OTHER	LONG
DISTRIBUTION	VARCHAR2 (30 BYTE)
CPU_COST	NUMBER
IO_COST	NUMBER
TEMP_SPACE	NUMBER
ACCESS_PREDICATES	VARCHAR2 (4000 BYTE)
FILTER_PREDICATES	VARCHAR2 (4000 BYTE)
PROJECTION	VARCHAR2 (4000 BYTE)
TIME	NUMBER
QBLOCK_NAME	VARCHAR2 (30 BYTE)

GOCDBTEST.GOCDB_GOD_OBJECT	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
NAME	VARCHAR2 (3 BYTE)

GOCDBTEST.GOCDB_SITES	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
SHORT_NAME	VARCHAR2 (100 BYTE)
OFFICIAL_NAME	VARCHAR2 (2000 BYTE)
HOME_URL	VARCHAR2 (500 BYTE)
GIIS_URL	VARCHAR2 (500 BYTE)
IP_RANGE	VARCHAR2 (500 BYTE)
OPERATINGHOURS_START	TIMESTAMP
OPERATINGHOURS_END	TIMESTAMP
LOCATION	VARCHAR2 (100 BYTE)
LATITUDE	NUMBER
LONGITUDE	NUMBER
DESCRIPTION	VARCHAR2 (2000 BYTE)
CREATION_DATE	TIMESTAMP
EMAIL	VARCHAR2 (100 BYTE)
CONTACTTEL	VARCHAR2 (255 BYTE)
EMERGENCYTEL	VARCHAR2 (255 BYTE)
CSIRTEMAIL	VARCHAR2 (255 BYTE)
CSIRTTEL	VARCHAR2 (255 BYTE)
EMERGENCYEMAIL	VARCHAR2 (255 BYTE)
HELPDESKEMAIL	VARCHAR2 (255 BYTE)

GOCDBTEST.GOCDB_TAGS	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
NAME	VARCHAR2 (100 BYTE)
DESCRIPTION	VARCHAR2 (500 BYTE)

GOCDBTEST.GOCDB_SERVICE_ENDPOINTS	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
ENDPOINT	VARCHAR2 (500 BYTE)
HOSTNAME	VARCHAR2 (500 BYTE)
HOST_IP	VARCHAR2 (20 BYTE)
HOST_DN	VARCHAR2 (200 BYTE)
HOST_OS	VARCHAR2 (50 BYTE)
HOST_ARCH	VARCHAR2 (50 BYTE)
DESCRIPTION	VARCHAR2 (2000 BYTE)
PRODUCTION_LEVEL	VARCHAR2 (20 BYTE)
BETA	VARCHAR2 (1 BYTE)
IS_MONITORED	VARCHAR2 (20 BYTE)

GOCDBTEST.GOCDB_USERS	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
FORENAME	VARCHAR2 (100 BYTE)
SURNAME	VARCHAR2 (100 BYTE)
TITLE	VARCHAR2 (20 BYTE)
CERTIFICATE_DN	VARCHAR2 (200 BYTE)
WORKINGHOURS_START	TIMESTAMP
WORKINGHOURS_END	TIMESTAMP
EMAIL	VARCHAR2 (50 BYTE)
TELEPHONE	VARCHAR2 (60 BYTE)

S	GOCDBTEST.MV_SE_PK
PK_SE	VARCHAR2 (100 BYTE)
PK_TYPE	VARCHAR2 (100 BYTE)
PK_SITE	VARCHAR2 (100 BYTE)

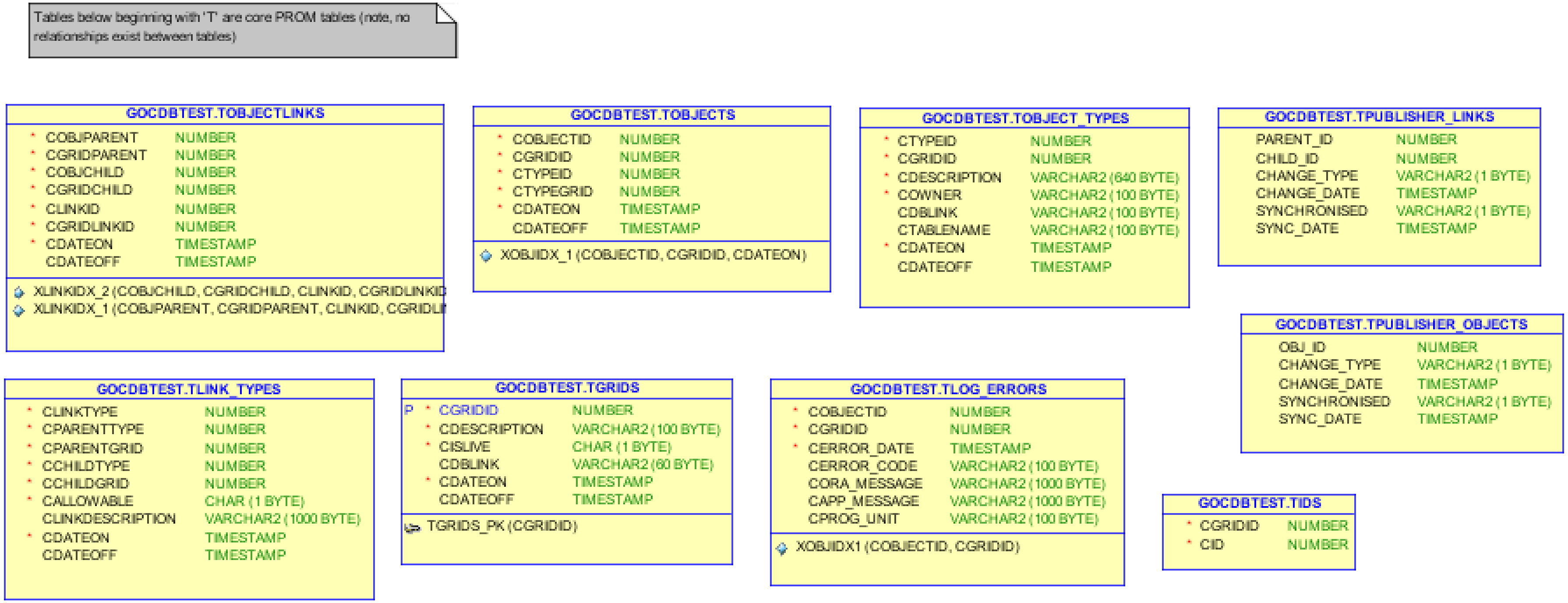
S	GOCDBTEST.MV_SITES_PK
PK_SITE	VARCHAR2 (100 BYTE)
PK_ROC	VARCHAR2 (100 BYTE)
PK_COUNTRY	VARCHAR2 (100 BYTE)
PK_PRODUCTIONINFRASTRUCTURE	VARCHAR2 (100 BYTE)
PK_DOMAIN	VARCHAR2 (100 BYTE)
PK_CERTIFICATIONSTATUS	VARCHAR2 (100 BYTE)
PK_TIMEZONE	VARCHAR2 (100 BYTE)

GOCDBTEST.GOCDB_EGEE_OBJECTS	
* COBJECTID	NUMBER
* CGRIDID	NUMBER
EXT_REFERENCE	NUMBER
PRIMARY_KEY	VARCHAR2 (100 BYTE)
ObjectID	NUMBER
Object_Type	VARCHAR2 (100 BYTE)
Is_EGEE_Object	VARCHAR2 (100 BYTE)

Note, PK/FK relationships do not exist between tables in a PROM database. Instead, they are enforced by the application using the PROM API.

The tables in the upper half of the diagram are the GOCDB specific data tables, while the tables in the lower half are the PROM admin tables.

Physical GOCDB Entity Model (2/2)



Tables below beginning with 'T' are core PROM tables (note, no relationships exist between tables)

GOCDBTEST.TOBJECTLINKS

*	COBJPARENT	NUMBER
*	CGRIDPARENT	NUMBER
*	COBJCHILD	NUMBER
*	CGRIDCHILD	NUMBER
*	CLINKID	NUMBER
*	CGRIDLINKID	NUMBER
*	CDATEON	TIMESTAMP
	CDATEOFF	TIMESTAMP

XLINKIDX_2 (COBJCHILD, CGRIDCHILD, CLINKID, CGRIDLINKID

XLINKIDX_1 (COBJPARENT, CGRIDPARENT, CLINKID, CGRIDLI

GOCDBTEST.TOBJECTS

*	COBJECTID	NUMBER
*	CGRIDID	NUMBER
*	CTYPEID	NUMBER
*	CTYPEGRID	NUMBER
*	CDATEON	TIMESTAMP
	CDATEOFF	TIMESTAMP

XOBJIDX_1 (COBJECTID, CGRIDID, CDATEON)

GOCDBTEST.TOBJECT_TYPES

*	CTYPEID	NUMBER
*	CGRIDID	NUMBER
*	CDESCRIPTION	VARCHAR2 (640 BYTE)
*	COWNER	VARCHAR2 (100 BYTE)
	CDBLINK	VARCHAR2 (100 BYTE)
	CTABLENAME	VARCHAR2 (100 BYTE)
*	CDATEON	TIMESTAMP
	CDATEOFF	TIMESTAMP

GOCDBTEST.TPUBLISHER_LINKS

PARENT_ID	NUMBER
CHILD_ID	NUMBER
CHANGE_TYPE	VARCHAR2 (1 BYTE)
CHANGE_DATE	TIMESTAMP
SYNCHRONISED	VARCHAR2 (1 BYTE)
SYNC_DATE	TIMESTAMP

GOCDBTEST.TPUBLISHER_OBJECTS

OBJ_ID	NUMBER
CHANGE_TYPE	VARCHAR2 (1 BYTE)
CHANGE_DATE	TIMESTAMP
SYNCHRONISED	VARCHAR2 (1 BYTE)
SYNC_DATE	TIMESTAMP

GOCDBTEST.TLINK_TYPES

*	CLINKTYPE	NUMBER
*	CPARENTTYPE	NUMBER
*	CPARENTGRID	NUMBER
*	CCHILDTYPE	NUMBER
*	CCHILDGRID	NUMBER
*	CALLOWABLE	CHAR (1 BYTE)
	CLINKDESCRIPTION	VARCHAR2 (1000 BYTE)
*	CDATEON	TIMESTAMP
	CDATEOFF	TIMESTAMP

GOCDBTEST.TGRIDS

P *	CGRIDID	NUMBER
*	CDESCRIPTION	VARCHAR2 (100 BYTE)
*	CISLIVE	CHAR (1 BYTE)
	CDBLINK	VARCHAR2 (60 BYTE)
*	CDATEON	TIMESTAMP
	CDATEOFF	TIMESTAMP

TGRIDS_PK (CGRIDID)

GOCDBTEST.TLOG_ERRORS

*	COBJECTID	NUMBER
*	CGRIDID	NUMBER
*	CERROR_DATE	TIMESTAMP
	CERROR_CODE	VARCHAR2 (100 BYTE)
	CORA_MESSAGE	VARCHAR2 (1000 BYTE)
	CAPP_MESSAGE	VARCHAR2 (1000 BYTE)
	CPROG_UNIT	VARCHAR2 (100 BYTE)

XOBJIDX1 (COBJECTID, CGRIDID)

GOCDBTEST.TIDS

*	CGRIDID	NUMBER
*	CID	NUMBER

GOCDB PI

- The PI is a set of RESTful GET request methods for querying GOCDB data formatted in XML over HTTPS
- Each method can be parameterized in order to narrow the search results, e.g.:
  - select all SEs belonging to Site X
  - show all sites belonging to NGI X
- Some popular methods can be scoped, i.e.
  - https://goc.egi.eu/gocdbpi/private/?method=get_site&scope=Local (returns all Local scoped sites)
- https://wiki.egi.eu/wiki/GOCDB/PI/Technical_Documentation

GOCDB PI Methods

get_site	Returns site information including contacts, grouped by site
get_site_list	Returns a list of sites with minimal associated information
get_site_contacts	Returns a list of persons (and associated info) having a role at site level, grouped per site
get_site_security_info	Returns security contact information for sites
get_roc_list	Returns a list of NGIs with minimal associated information
get_subgrid_list	Returns a list of Subgrids (i.e. registered sub-parts of an NGI) with minimal associated information
get_roc_contacts	Returns NGI contact details, including NGI contact mail address and list of NGI staff
get_egee_contacts	Returns a list of contacts for staff that have a role a EGI level
get_downtime	Returns a list of EGI downtimes for sites and nodes
get_service_endpoint	Returns a list of service endpoints (single node x single service) and associated information
get_service_types	Returns a list of valid service types and associated description
get_user	Returns a user or a list of users with associated details and roles
get_downtime_to_broadcast	Returns the list of downtimes recently declared with notification settings for CIC portal downtime notification service
get_cert_status_changes	Returns a list of changes to certification statuses
get_cert_status_date	Returns a list of current certification statuses for Production sites and the date they entered that status

Status of Regionalisation Developments

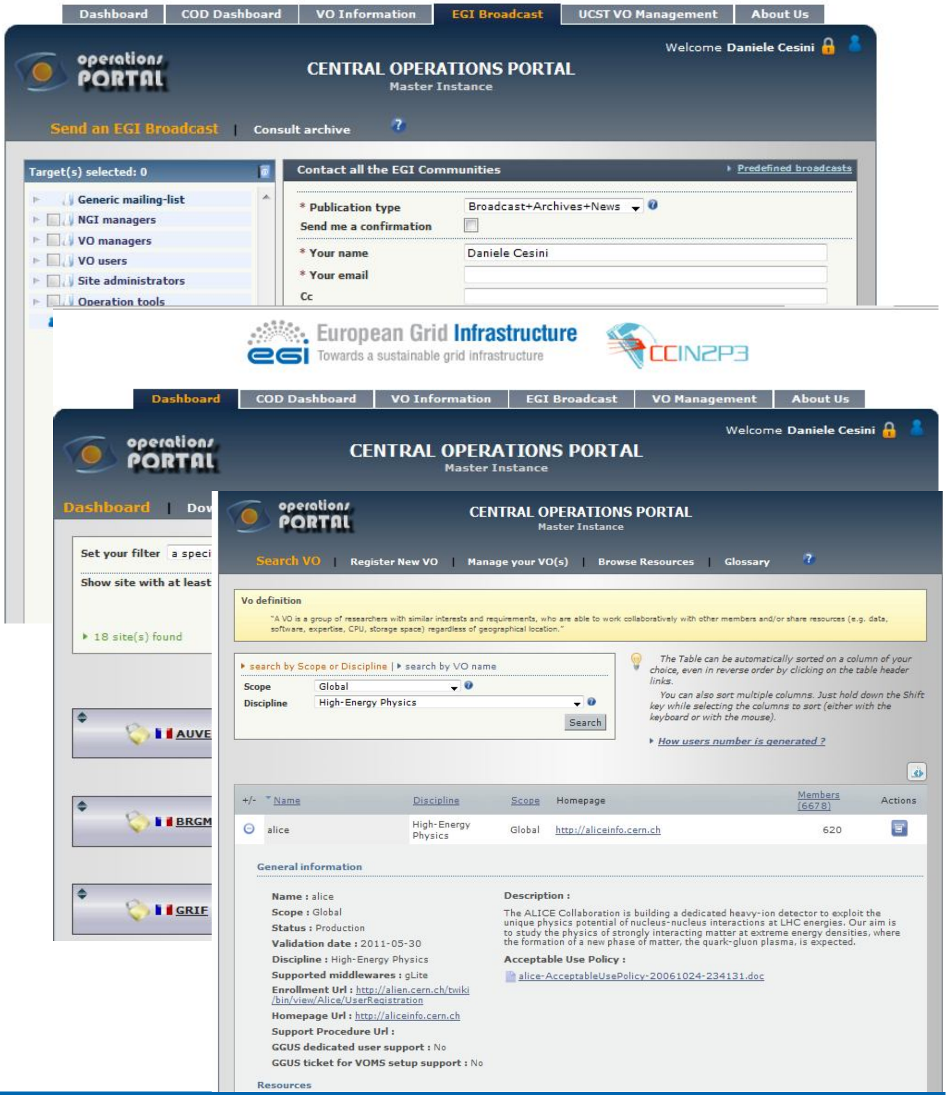
- GOCDB developments in 2011 have focused on
  - a significant amount of re-development/refactoring
  - addressing new requirements for the central instance that have emerged throughout the year.
  - (more info on https://wiki.egi.eu/wiki/GOCDB/Release4/Development)
- Fixing the Atomic PROM API and addition of data-scoping/tagging are essential for the regional-publishing model described at: https://wiki.egi.eu/wiki/GOCDB/Release4/Regionalisation
- There are still three outstanding developments to be tackled before we can really start in earnest on regionalization:
  - Virtual Sites
  - a finer grained permission/authorization model
  - replacing the current XML output module as it cannot create nested XML collections (is necessary)
  - These are detailed at: https://wiki.egi.eu/wiki/GOCDB/Release4/Development#Development_Roadmap

Operations Portal

Operations Portal

Single access point to almost all operational information. Widely used in the day-by-day run of the GRID.

- Broadcast tool
- Operational Dashboard
- VO Information
  - IDCard
  - User Traking
  - Resources
- USCT Management Dashboard
- Security Dashboard
- COD Dashboard
- VO Oriented Dashboard (in progress...)



* Publication type: Broadcast+Archives+News
Send me a confirmation ☐
* Your name: Daniele Cesini
* Your email:
Cc:

Set your filter: a speci
Show site with at least
18 site(s) found

Vo definition

"A VO is a group of researchers with similar interests and requirements, who are able to work collaboratively with other members and/or share resources (e.g. data, software, expertise, CPU, storage space) regardless of geographical location."

Scope: Global
Discipline: High-Energy Physics

The Table can be automatically sorted on a column of your choice, even in reverse order by clicking on the table header links.

You can also sort multiple columns. Just hold down the Shift key while selecting the columns to sort (either with the keyboard or with the mouse).

+/-	Name	Discipline	Scope	Homepage	Members (6678)	Actions
⊖	alice	High-Energy Physics	Global	http://aliceinfo.cern.ch	620	

Name : alice
Scope : Global
Status : Production
Validation date : 2011-05-30
Discipline : High-Energy Physics
Supported middlewares : gLite
Enrollment Url : http://alien.cern.ch/twiki/bin/view/Alice/UserRegistration
Homepage Url : http://aliceinfo.cern.ch
Support Procedure Url :
GGUS dedicated user support : No
GGUS ticket for VOMS setup support : No

Description :

The ALICE Collaboration is building a dedicated heavy-ion detector to exploit the unique physics potential of nucleus-nucleus interactions at LHC energies. Our aim is to study the physics of strongly interacting matter at extreme energy densities, where the formation of a new phase of matter, the quark-gluon plasma, is expected.

Acceptable Use Policy :

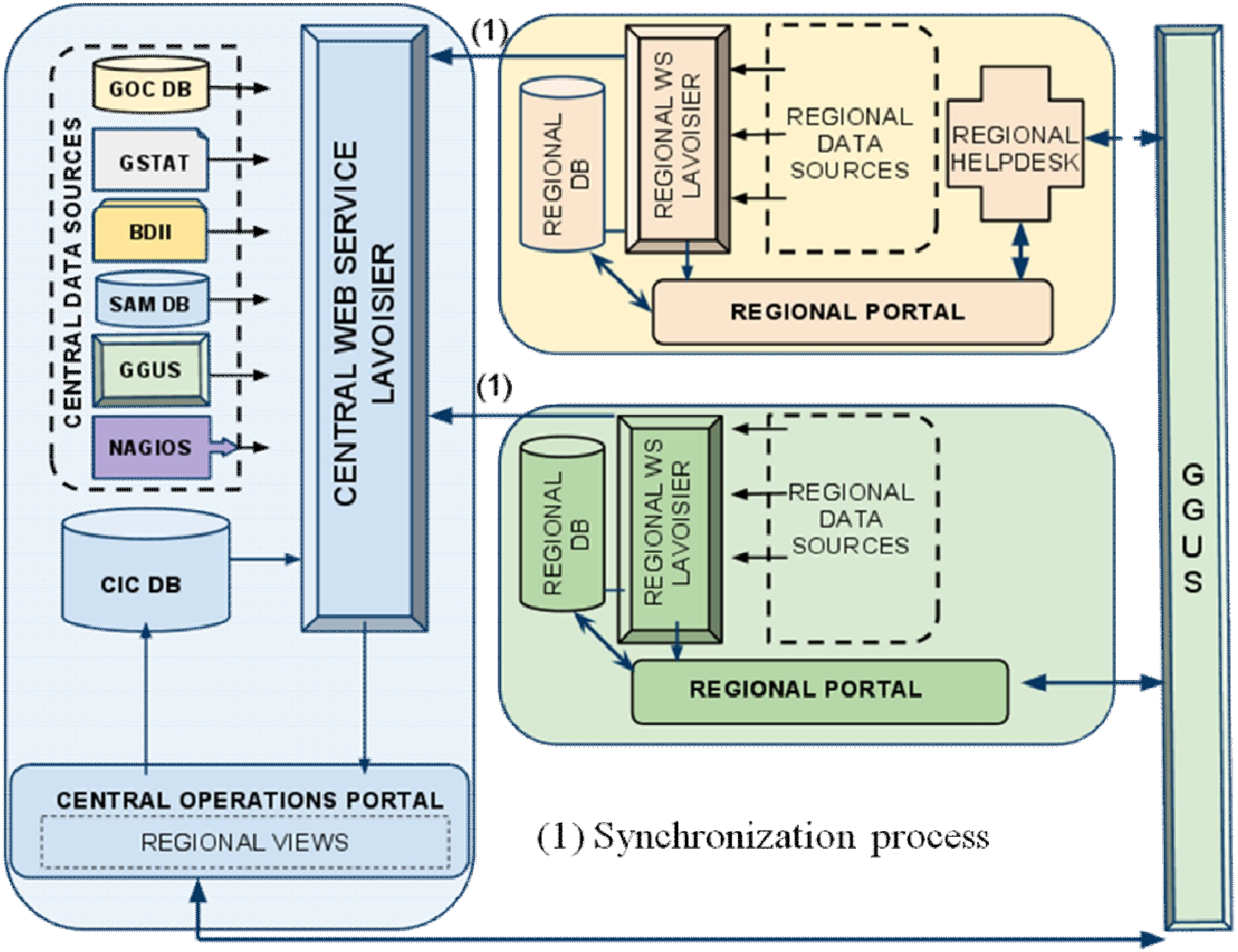
alice-AcceptableUsePolicy-20061024-234131.doc

Operations Portal Architecture

Regional and Central Instances synchronized though Lavoisier

More info:
MS701 :
https://documents.egi.eu/document/39

MS705:
https://documents.egi.eu/document/27



Ops Portal and Infosys

- Direct ldap queries to one Top-BDII
  - Service topology: where ? What ? And which VO ?
  - Every 2 hours
  - Cache mechanism: not replaced in case of failure
  - GOC DB information insufficient (VO Information)

- Query to GSTAT (per site and per VO)
  - status of the sBDII - CPU (site, vo) - Jobs (site, vo) - Storage (site, vo)
  - Every 6 min
  - Cache mechanism: not replaced in case of failure

Ops Portal and Infosys

- Information coming from BDII is used
  - By Downtime Notification System
  - Into the Resource Distribution browser
    - site and VO view

- Information coming from GSTAT is used
  - By the VO Module
  - By the Dashboard

Ops Portal and GOCDB

- GOCDB as official source of the following information:
  - services/sites/ngi topology
  - contact info
  - production status

- This information is enhanced by GSTAT or BDII queries

Ops Portal Requirement to InfoSys

- Continue to offer the same level of information available today by the three tools
- Enhance this information if possible
  - more information at the service level
    - storage distribution on specific SE
- More APIs
  - To query the BDII ldap commands are not enough flexible and easy usable
  - in GSTAT not possible to have a global view (egi_ngi , egee_roc, wlcg_tier)

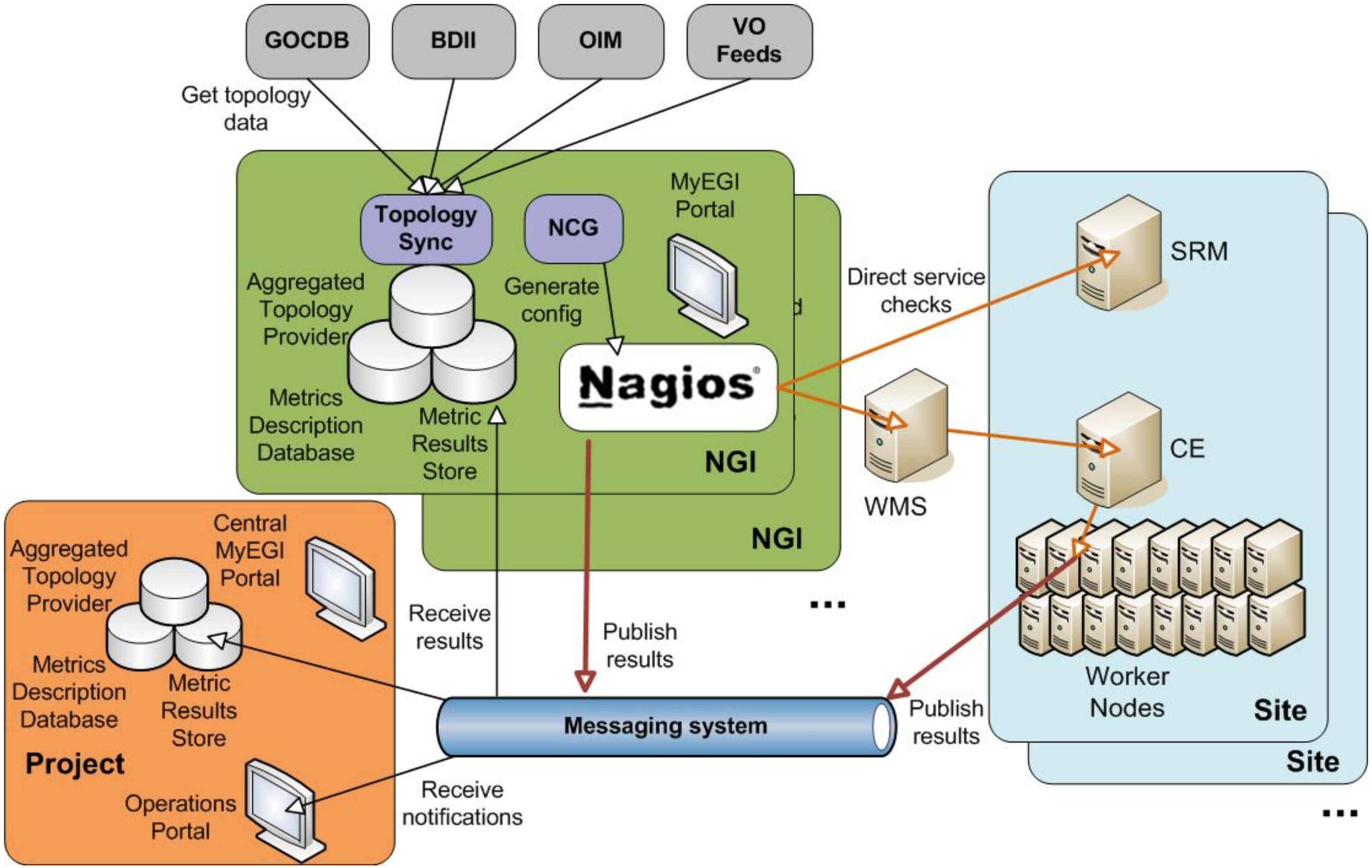
SAM Framework

Service Availability Monitoring (SAM)

SAM monitoring framework for RCs and services

- one of the main data sources for the Operations Portal
- data source to create Availability/Reliability statistics
- components:
  1. test submission framework: based on the NAGIOS system set up and customized by the NAGIOS Configurator (NCG)
  2. databases for storage of information about topology, metrics and results
  3. visualization tool GUI: MyEGI

SAM architecture



SAM and The Information System

- BDII, GSTAT and REBUS are used by SAM
- From BDII (direct ldap queries):
  - service URIs
  - supported VOs for each service
  - services supporting different MPI implementations;
- From GSTAT:
  - CPU information (PhyCPU, LogCPU, ksi2k, HEPSPEC06);
- From REBUS:
  - Tier and federations (Tier, Federation, FederationAccountingName, Site, Infrastructure, Country).

SAM and GOCDB

- SAM uses the GOCDB by querying the following feeds:
  - https://goc.egi.eu/gocdbpi/private/?method=get_site
  - https://goc.egi.eu/gocdbpi/private/?method=get_service_endpoint
  - https://goc.egi.eu/gocdbpi/private/?method=get_downtime
  - https://goc.egi.eu/gocdbpi/private/?method=get_service_types
- The Information System is needed to get extra information not provided by GOCDB
  - which VOs support each service and services supporting MPI implementations

SAM and Configuration

- Frequency of information retrieval = 30mins (configurable)

- A caching mechanism is used internally
  - Information is stored and updated every time it is modified.

- Infosys endpoints are in the configuration files (there is no discovery)

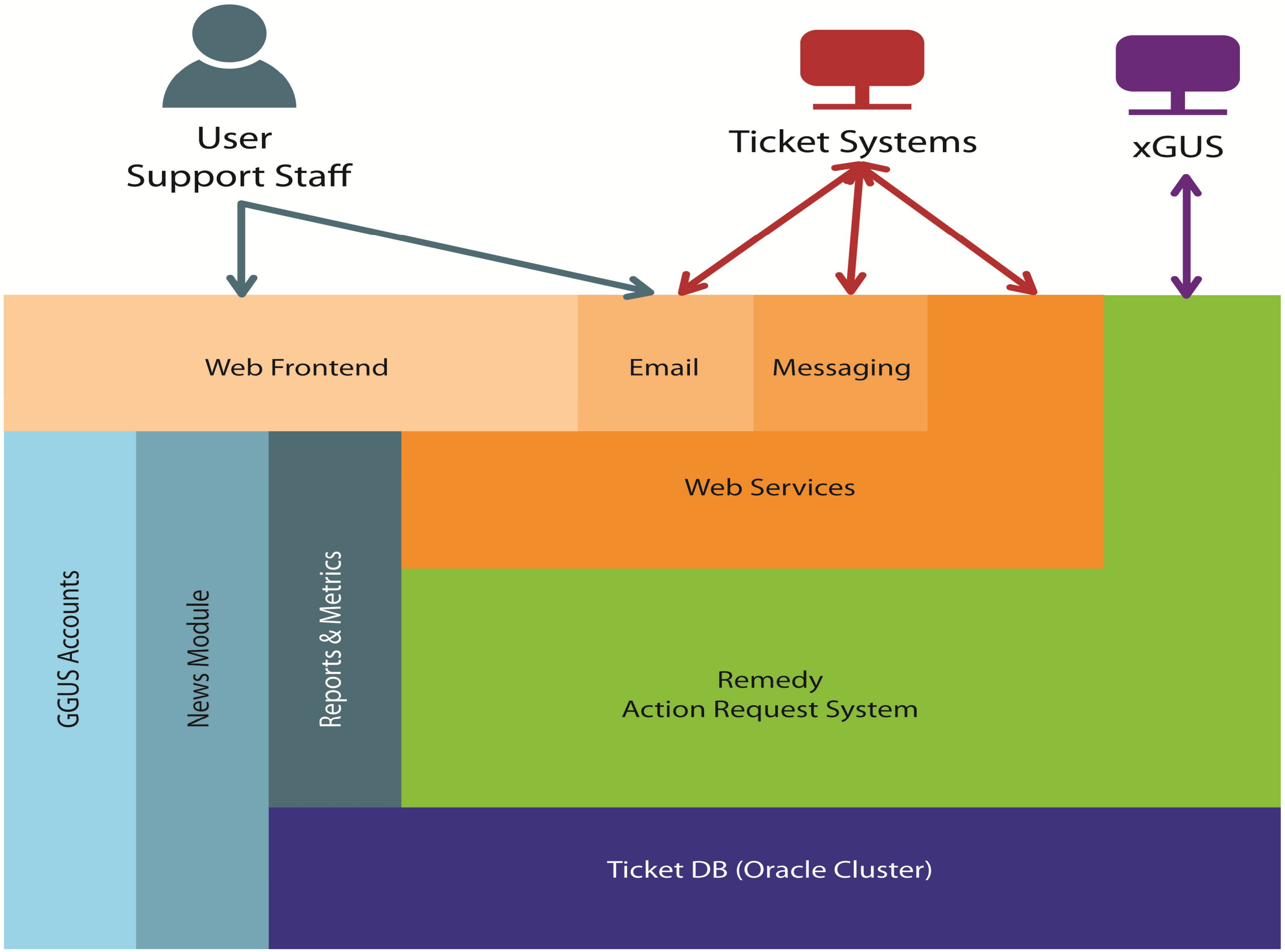
SAM Requirements to the InfoSys

- Current InfoSys provides sufficient information and we have no new requirements
- All the information provided is needed by SAM
  - Replacing one of the InfoSys would require getting the respective information from elsewhere
- It would be convenient to have a single place for all the information
- Also we would benefit if each InfoSys tool has a well defined scope (avoiding overlaps)

EGI Helpdesk (GGUS)

- EGI Helpdesk
  - distributed helpdesk with central coordination: Global Grid User Support (GGUS)

GGUS Architecture



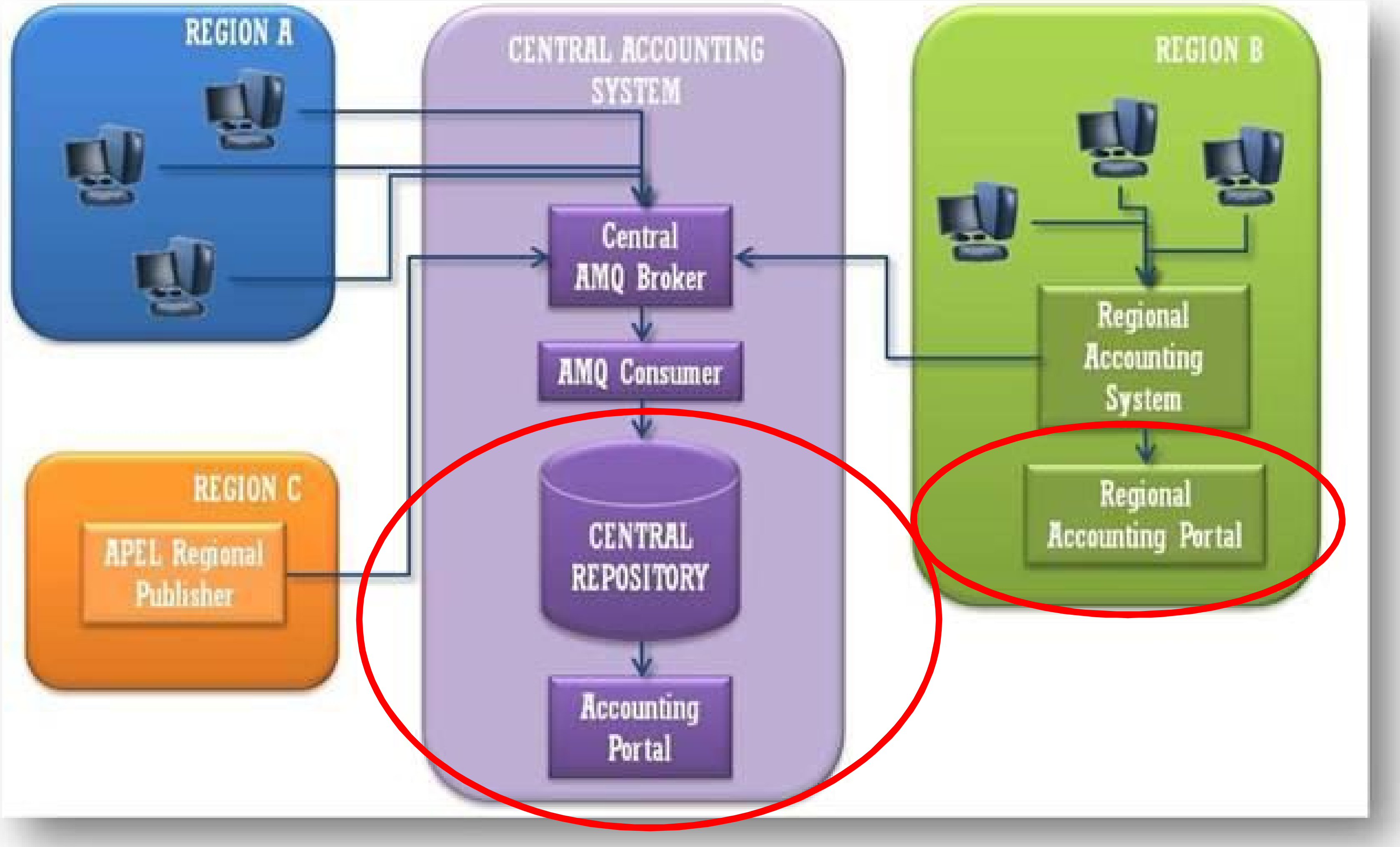
GGUS and GOCDB

- GGUS has NO dependencies on the Information System

- Has dependencies on GOCDB
  - Information retrieved every night through a Cronjob:
    - site data
    - contact mail address
    - site names

Accounting in EGI-JRA1

Accounting


REGION A
CENTRAL ACCOUNTING SYSTEM
REGION B
Central AMQ Broker
AMQ Consumer
Regional Accounting System
REGION C
APEL Regional Publisher
CENTRAL REPOSITORY
Regional Accounting Portal
Accounting Portal

Accounting Repository and Portal

Accounting Repository (STFC)

- usage of compute resources within the production infrastructure
- based on gLite-APEL

Accounting Portal (FCTSG)

- GUI for access to data from the Accounting Repository

Accounting Repository and Infosys

- APEL retrieves batch scaling or benchmark values for a cluster from the BDII
- Uses them to normalize CPU accounting data

- Two ldap queries:
  - First search is the highest priority search.
  - If both searches return a SpecInt value for a cluster, the value from the first search is used; the second is disgarded.

Accounting Repository and Infosys

First search

Find a GIIS entry where "objectclass=GlueCE" (GlueCEUniqueID=lcgce06.gridpp.rl.ac.uk:2119/jobmanager-lcgpbs-grid1000M)

Get the "GlueForeignKey" attribute (GlueClusterUniqueID=sl5-1000.gridpp.rl.ac.uk)

Remove "GlueClusterUniqueID=" from the start
(sl5-1000.gridpp.rl.ac.uk)

Extract "SpecInt" from "GlueCECapability"
(CPUScalingReferenceSI00=1000)

Second search

Find a GIIS entry where objectclass=GlueSubCluster
(GlueSubClusterUniqueID=lcgce06.gridpp.rl.ac.uk)

Get the GlueChunkKey
(GlueClusterUniqueID=lcgce06.gridpp.rl.ac.uk)

Get the GlueSubClusterUniqueID
(lcgce06.gridpp.rl.ac.uk) Get SpecInt from

GlueHostBenchmarkSI00
(1000)

Accounting Repository and Infosys

- At most sites the accounting records are built locally
  - Normalization information could probably be stored elsewhere by APEL configuration

- At a few sites (eg France) accounting is published by one site for many
  - In this case the normalization values are retrieved remotely using the BDII.
  - Any alternative to this would either have to gather normalization data during the parsing phase on the CE, or establish some other remote publishing technique.

APEL and GOCDB

- APEL’s uses GOCDB to retrieve information necessary to build
  - the EGI topology information for summaries
  - the ACLs for the APEL AMQ broker.

Accounting Portal and Infosys/GOCDB

- BDII is not used by the Accounting Portal

- GOCDB is used to retrieve topological information:
  - site information
  - tier status
  - country/ngi mappings, etc..
  - This information is gathered every 3 hours.

Metrics Portal

Metrics Portal

- Collects a set of metrics from different resources to help in measuring project performance
- Keeps track of the project evolution by displaying historical values of the metrics in a single place.
- It also provides web interfaces to inject the metrics into the database

Metrics Portal and Infosys

- The metrics portal was supposed to get information from GSTAT, but resulted to be not flexible
  - the first problem was separating EGI+EGEE nodes
  - there are metrics not covered by GSTAT
  - unnecessarily strict with the publication
  - and many sites disappear
- Now use BDII directly
- Information is collected for all EGI sites and then aggregated by NGI based on GOCDB
  - some NGIs are further partitioned into countries

Metrics Portal and Infosys

- For each site it gets weekly:
  - Sites using MPI
  - Disk and Tape Storage
  - Logical CPUs
  - HEPSPEC06 capacity
    - get SI2K, since HEPSPEC06 is reported with issues and inconsistencies

- No caching is used
  - If the infosys is not reached, the script sends an email to the maintainer
  - The update can be then done manually later

Metrics Portal and GOCDB

- Many of the above metrics are not usually covered by GOCDB, but if GOCDB carried all the needed information sufficiently updated the Metrics Portal would probably use it instead of BDII

- From GOCDB retrieves data for mapping sites to NGIs
  - for some NGIs the mapping is refined to countries
    - e.g. Ibergrid -> Spain+Portugal
    - NGI_NGDF -> Finland, Denmark, Sweden and Norway

Metrics Portal future Usage of the InfoSys and Requirements

- The future uses depend on the metrics evolution
- If NGI wide BDIIs are available it would be possible to use them

Requirements:

- Better coverage for HEPSPEC06 information
- Info on Cloud resources should be made as seamless as possible to have a unified view of the total composition the infrastructure (Grid+Cloud)

Summary Table

	BDII	GSTAT	GOCDB	BDII query Frequency	Cached	IS Clients
SAM	Yes	Yes	Yes	30 mins	Yes	BDII: ldapsearch (will use ARC Lib and Unicore CLI) GOCDB PI
Ops Portal	Yes	Yes	Yes	BDII: 2hrs GSTAT: 6mins	Yes	BDII: ldapsearch GOCDB PI
Acc Repo	No	No	Yes	-	-	GOCDB PI
Acc Portal	No	No	Yes	GOCDB: 3hrs	-	GOCDB PI
Metrics Portal	Yes	Yes	Yes	weekly	-	BDII: ldapsearch GOCDB PI
GGUS	No	No	Yes	Every night	-	GOCDB PI
GOCDB	No dependecies					

Conclusion

- From an EGI-JRA1 perspective BDII, GSTAT and GOCDB are all needed as source of information
  - BDII and GSTAT are mainly used to complement GOCDB info for VO mapping
- No big requirements on the Information System
  - Improvement of some features, i.e.:
    - better coverage for HEPSPEC06 information
    - More information at service level (SE)
  - APIs to query the BDII would be great
  - Stability of information could be improved but is obtained with internal caching mechanisms
  - Accounting data punctual normalization can provide new requirements depending on how it is implemented
- It would be convenient to have a single place for all the information