GOCDB
A repository for a worldwide grid infrastructure

G. Mathieu, A. Richards, J. Gordon, C. Del Cano Novales,
P. Colclough, M. Viljoen

CHEP09, Prague, March 2009

Outline

- What is GOCDB
  - Context and generalities
  - Purpose and history
  - GOCDB interactions with other EGEE tools
- GOCDB architecture
  - Components, services and interfaces
  - Development and procedures
  - Availability and failover
- Evolution
  - GOCDB in a distributed environment

What is GOCDB? (1)
Context and generalities

- What's in a name?
  - GOCDB = **G**rid **O**perations **C**entre **D**ata**B**ase

- A central (static) information repository

- Key component of EGEE and WLCG, used as an authoritative data source

What is GOCDB? (2)
Context and generalities

- Stores information about (and links together):
  - Regions and countries
  - Sites, nodes and services
  - Users

- Consists of 2 parts:
  - A Database
  - A web portal (https://goc.gridops.org)

What is GOCDB? (3) Handled data

- Administrative info
  - Contacts, names, groupings (region, country…)
  - *Used for official procedures, reports, listings*

- Resources and services
  - service endpoints and types
  - *Used for monitoring and operations*

- Maintenance plans
  - *Used for monitoring and availability reports*

What is GOCDB? (4)
History

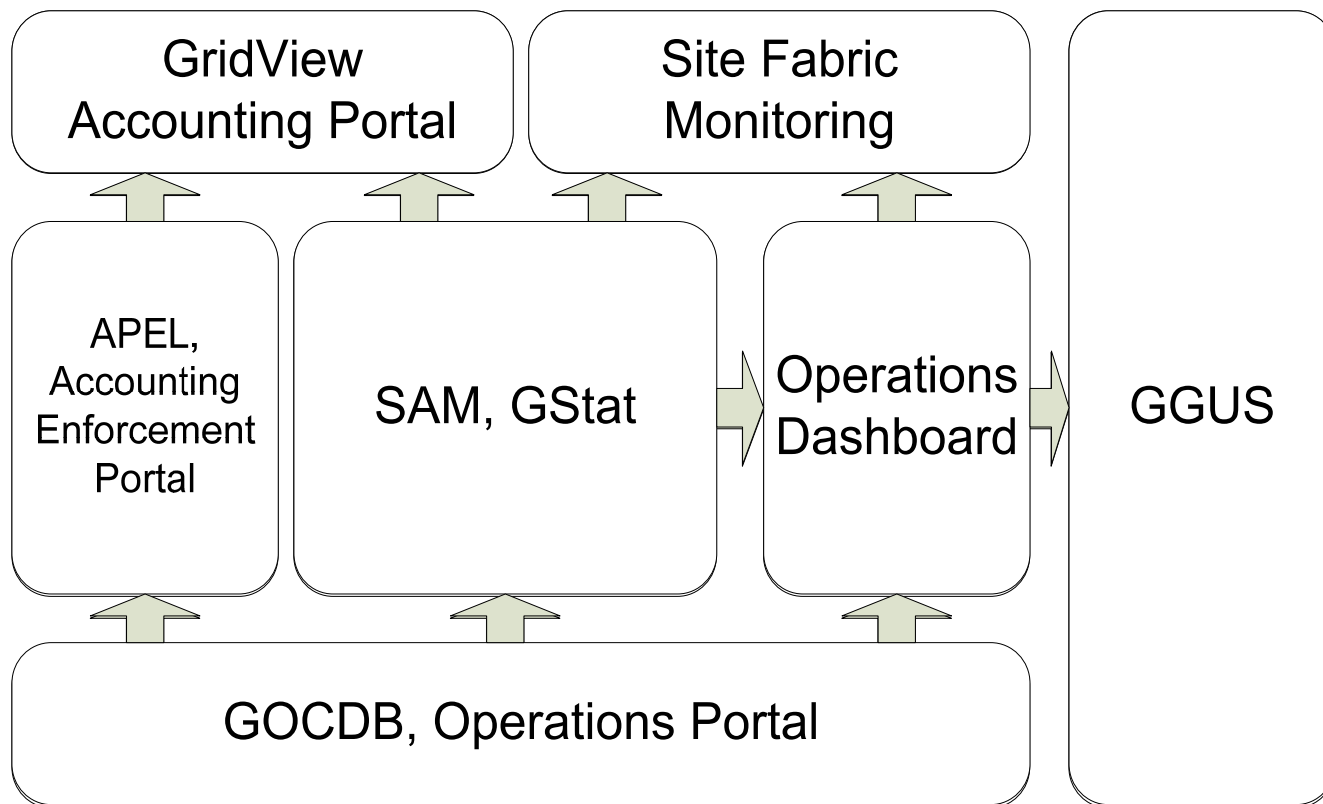| Year	Era	Description
2003		
2004		
	Stone Age	Static list of sites and their contacts
2005		
	Iron Age	simple MySQL DB + scripts
2006		
	Medieval times	MySQL DB + proper web interface
2007		
	Industrial Revolution	Oracle DB + user friendly web portal
2008		
	Modern times	Oracle DB + Web portal + web services
2009		

Architecture overview

- Web portal
  - Based on a Model-view-Controller (MVC) principle
  - Coded in PHP, object fashion
- Database
  - hosted on Oracle 11g cluster
  - 3 schemas: Production, test, development
- Authentication
  - Based on X509 certificates
  - Role management
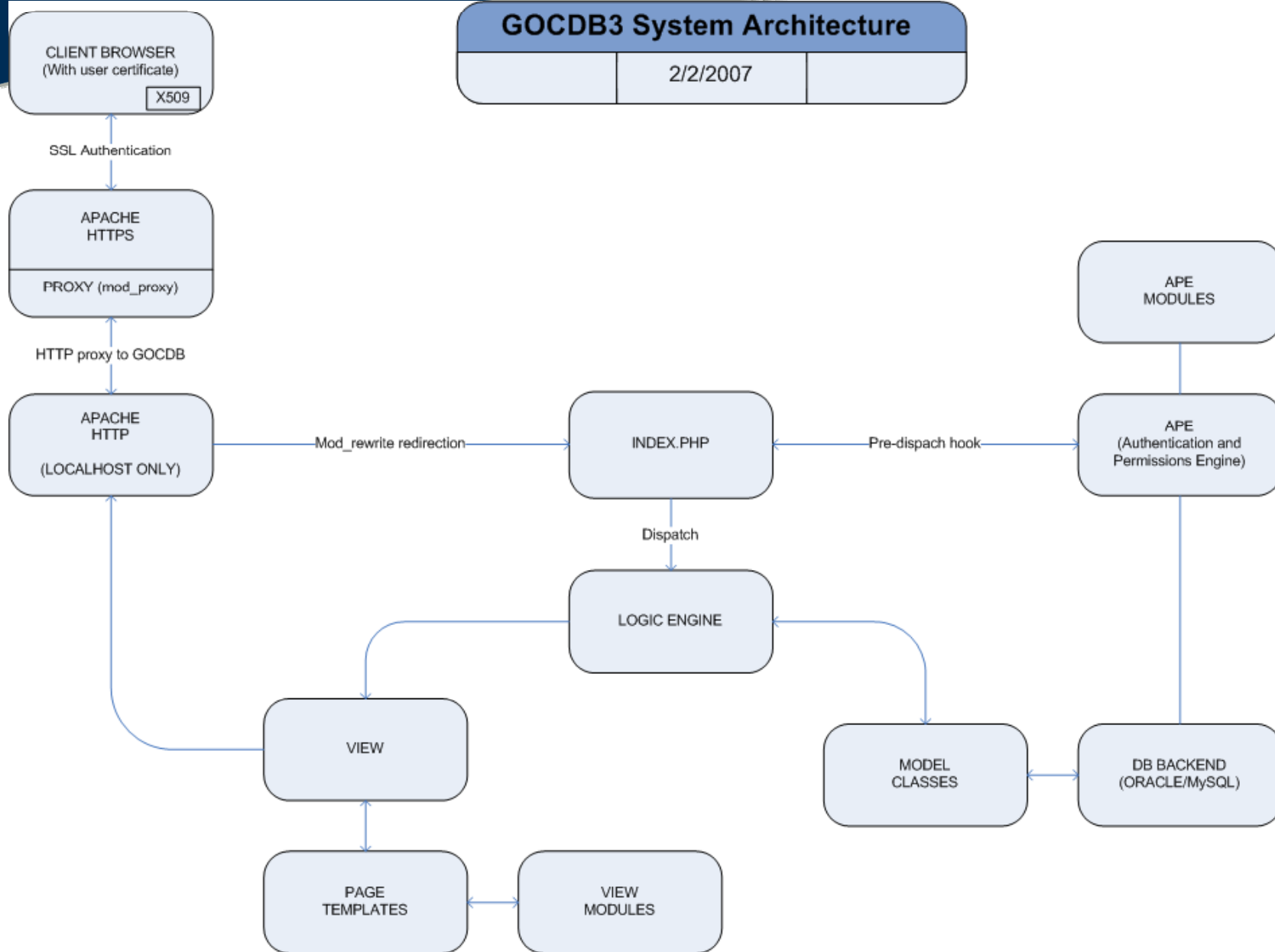
Services and interfaces

- GOCDB Programmatic interface (GOCDB-PI)
  - REST based interface over https
  - Business logic put in stored functions at DB level
  - Use of Oracle XML DB
  - Different methods with different security levels

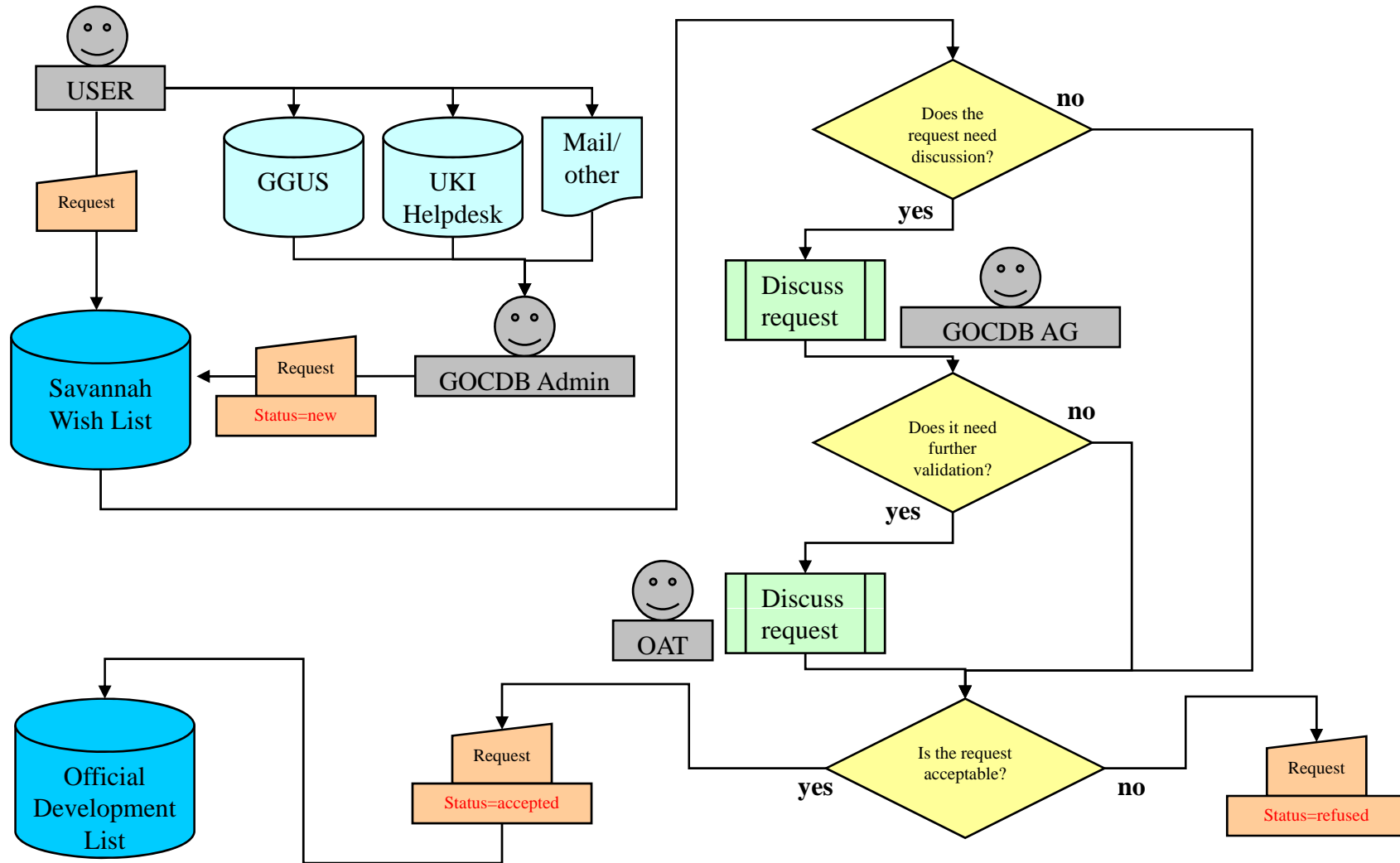- SOAP web services under study

Processes and procedures

- Code synchronisation
  - GOCDB repository in SVN
- Release procedure
  - RPM deployment
  - IP switch between production and test server
- Development cycle
  - Savannah task/bug tracker
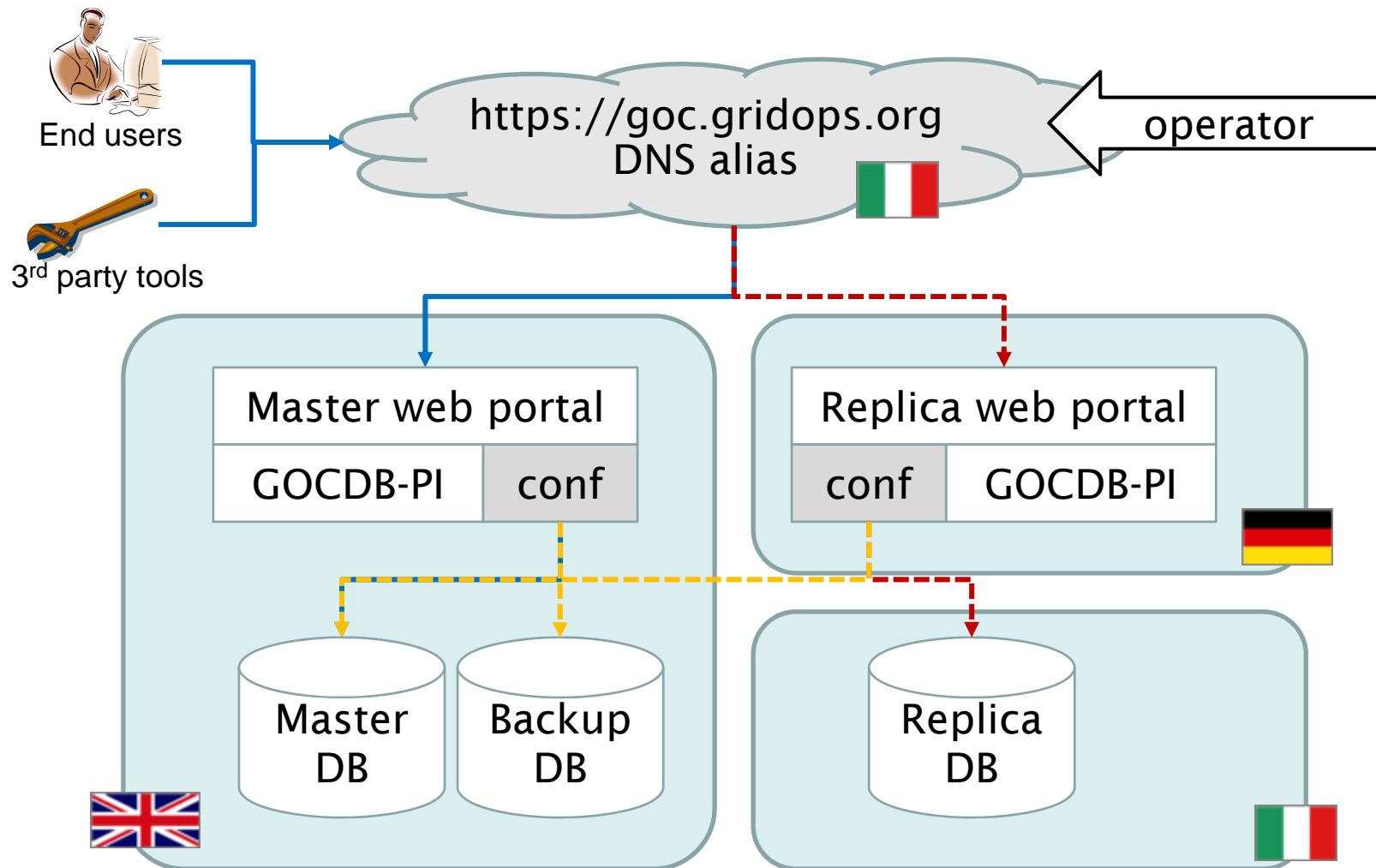 - Well established validation process

Development workflow

GOCDB Failover (1)

- Web portal
  - Master instance at RAL, UK
  - Replica instance at ITWM, Germany
- Database
  - Master instance at RAL, UK   Oracle 11g
  - Replica instance at CNAF, Italy   Oracle 10g
  - Local backup instance at RAL, UK   Oracle Xpress

GOCDB Failover (2)

GOCDB evolution

- Model evolution in EGEE
  - From central to distributed
  - Automate, simplify, standardize

- Distributing "the beast"
  - New architecture under development
  - Challenging work and short deadline

Main ideas

- Keep a central service, not necessarily a central DB
  - There is a need for a central access point, but:
  - the fact that regional DB are distributed or not must not be an issue

- Build a sustainable architecture that allows regionalisation but doesn't force it
  - Not all regions are at the same level

- Propose an implementation where nothing exists, work with existing solutions otherwise
  - Some regions have their own solution and don't want to be forced to use another one

High level architecture

One level down

Region 3 local portal

Region 3 homemade local DB

Publisher

Region 3

3rd party tools

End users

WS interface

Central portal

Query interface

Region 1 local portal

Query interface

Data collector

GOCDB internal model (regions 1, 2 and 3)

GOCDB Region 1 model

GOCDB internal queries

Region 1

RAL
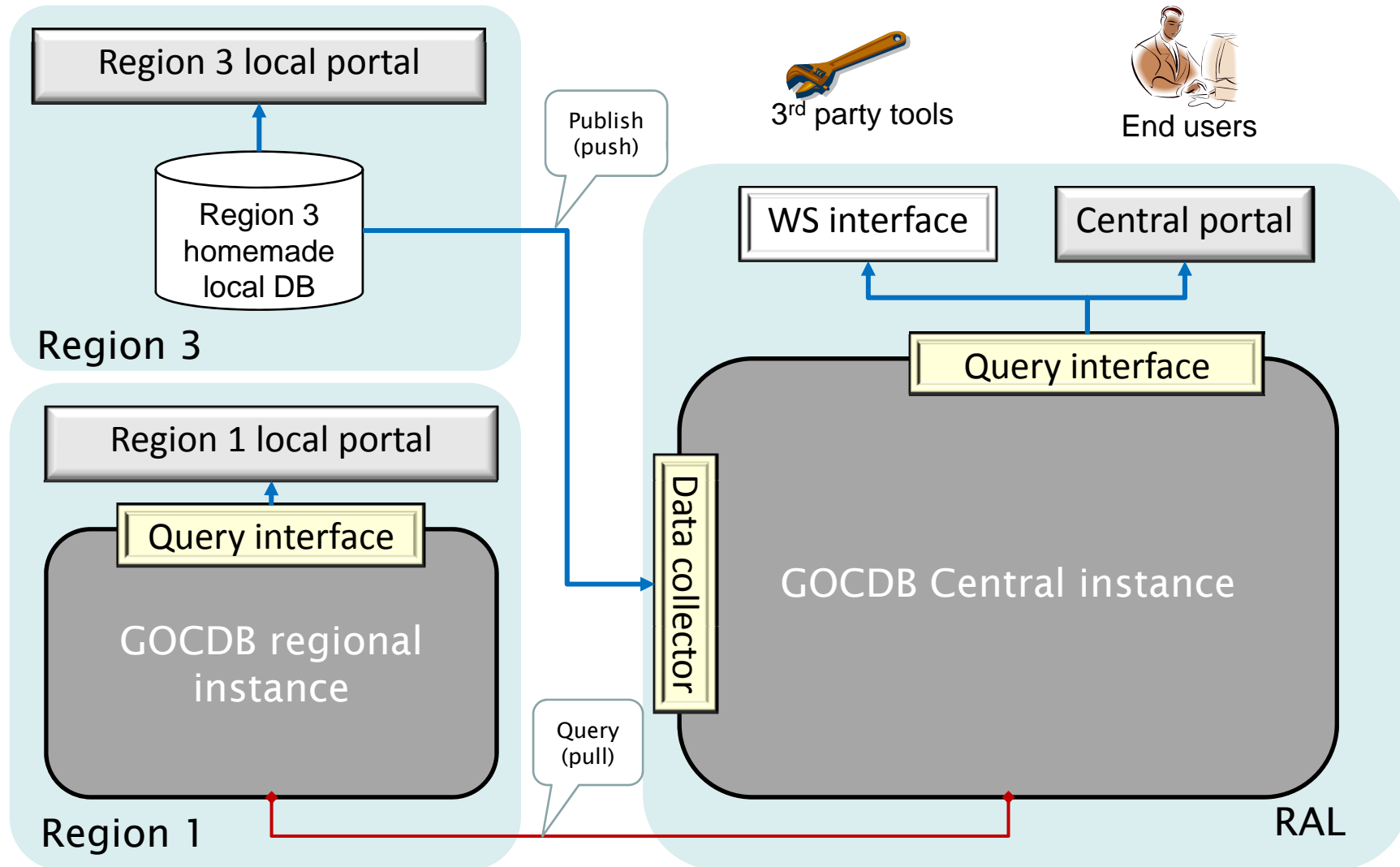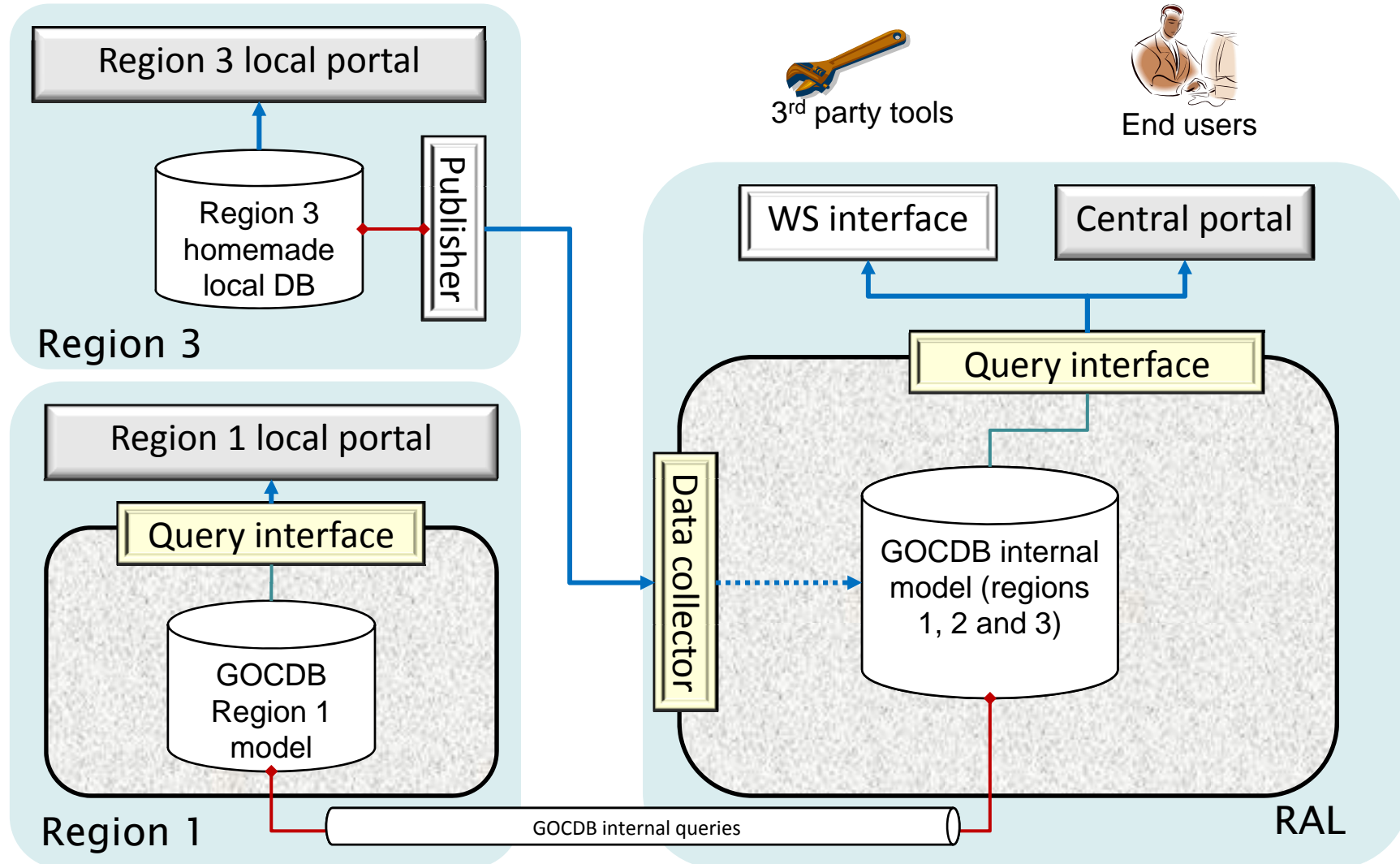
Details and components
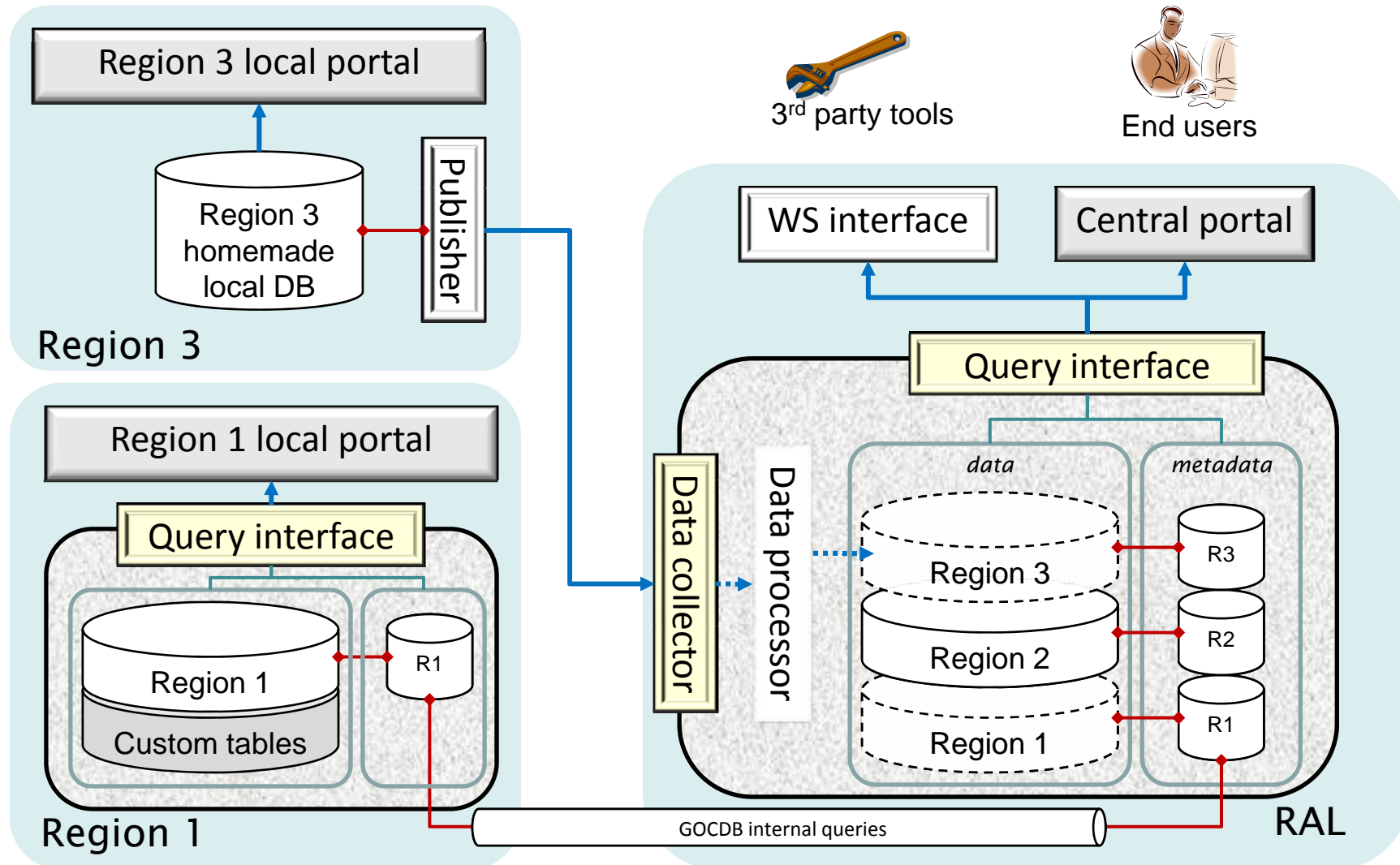
internal DB schema

Current relational model

SITES
- ID
- Sitename
- email

NODES
-ID
-hostname
-IP_addr

PATHS
- path
-SiteID
-NodeID
-ServiceID

SERVICES
- ID
- name
- desc

DOWNTIMES
- ID
- path
- start_date
- end_date
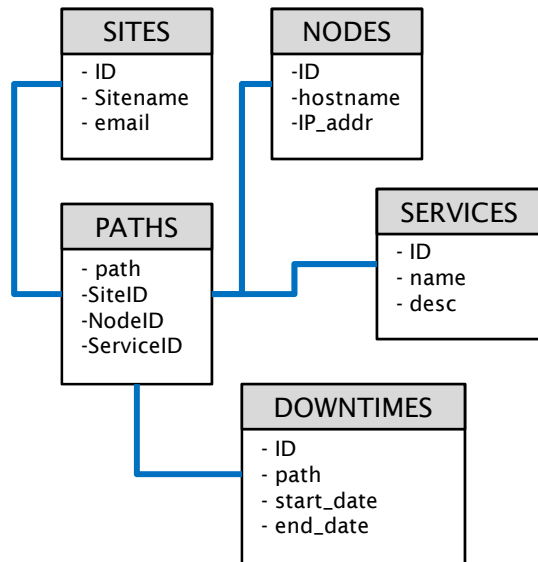
- Physical Data Tables
- Hard coded relationships and constraints

Proposed object model

objectTypes
-Type
- Name
-Database
-Grid

objects
-Type
-objectID

objectLinkTypes
- parent_type
-Child_type
-Allowed

objectLinks
-ID
-Link_type
-Parent_objectID
-Child_objectID
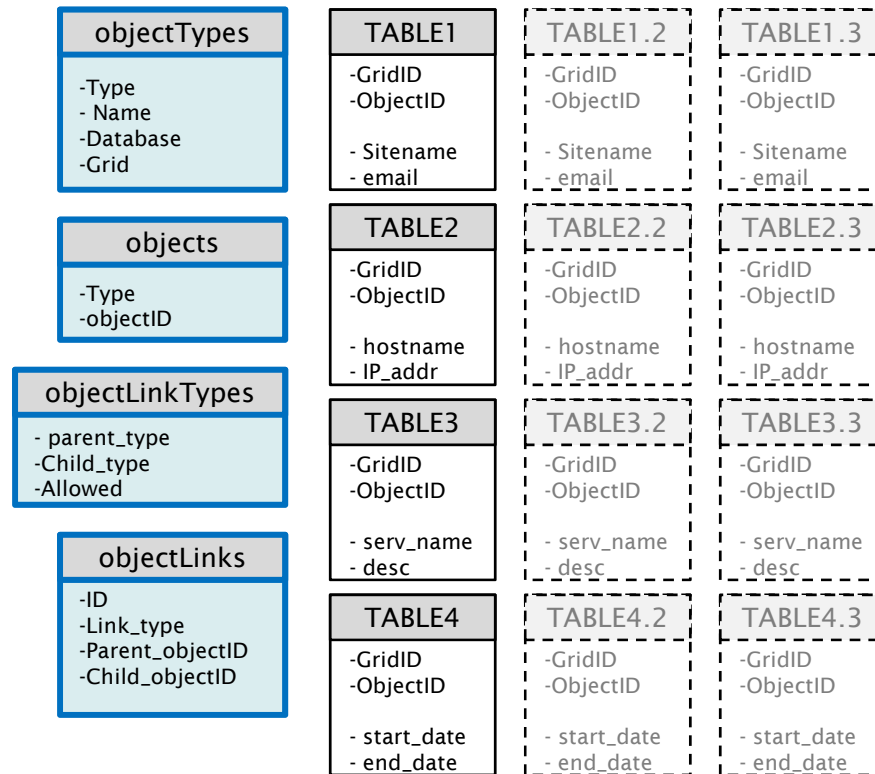
Core tables
(relationships)

TABLE1	TABLE1.2	TABLE1.3
-GridID -ObjectID - Sitename - email	-GridID -ObjectID - Sitename - email	-GridID -ObjectID - Sitename - email
TABLE2	TABLE2.2	TABLE2.3
-GridID -ObjectID - hostname - IP_addr	-GridID -ObjectID - hostname - IP_addr	-GridID -ObjectID - hostname - IP_addr
TABLE3	TABLE3.2	TABLE3.3
-GridID -ObjectID - serv_name - desc	-GridID -ObjectID - serv_name - desc	-GridID -ObjectID - serv_name - desc
TABLE4	TABLE4.2	TABLE4.3
-GridID -ObjectID - start_date - end_date	-GridID -ObjectID - start_date - end_date	-GridID -ObjectID - start_date - end_date

Data tables

Collection1   Collection2   Collection3

For more details…

- GOCDB homepage
  - http://www.grid-support.ac.uk/content/view/406/290/

- GOCDB4 development wiki
  - http://goc.grid.sinica.edu.tw/gocwiki/GOCDB4_development

- "A pseudo object database model and its applications on a highly complex distributed architecture"
  - IARA/IEEE Conference on Advances in Databases (DBKDA 2009)
    March 1-6, 2009 - Cancun, Mexico

Thank you

- Questions ?