

GOCDDB (Regionalisation), GLUE2 and Info Discovery

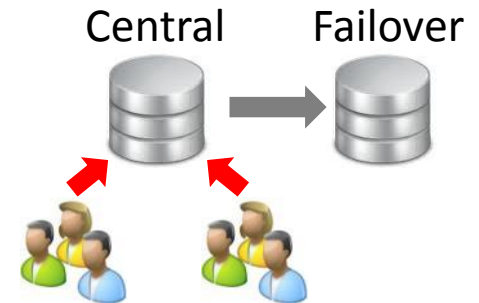
David Meredith, John Casson

EGI TF Prague
September 20th 2012

Deployment Scenarios

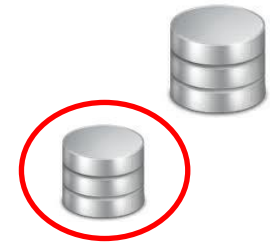
1. Central GOCDB

- Used by all NGIs. Data in one location. Failover.
- Scoping allows hosting of non-EGI data.
- Centrally supported.
- Custom Service Types (eg 'CUSTOM.UserPortal')
- Custom Property Bags (TODO)
- Can't accommodate schema customisations



2. Regional-Standalone GOCDB

- Freely customisable (e.g. EUDAT GOCDB instance).
- Does not synch data with the central GOCDB.



3. Regional-Publishing GOCDB (?)

- Synchronise EGI scope data to parent GOCDB.
- Major undertaking. To be decided.



Regional-Publishing GOCDDB: Use Cases

1. Enables Customisation Beyond Custom SE Types + Property Bags

- Schema changes, new tables, GUI refinements
- Beware Restriction: Synchronised data is 'Open for extension, but closed for modification'



- Strict data validation = modifications would fail validation when publishing.
- Extensions ignored when publishing (e.g. in place of `<xsd:any/>`, new tables).

2. Data Visibility

- Private data is not published to central GOCDDB
 - Sensitive sites and services ?


3. Resilience

- If central instance fails, Regional-Publishing GOCDDB is still available
 - Has resilience really been an issue (esp. considering failover instance)?

Regional-Publishing GOCDDB: Data History

- A GOCDDB maintains a history of data changes:
 - Site Certification-Status change history
 - Historical Downtimes
 - Site-to-NGI affiliation history (TODO)
- History + any offline changes require synchronisation:
 - EGI scoped data must match across instances else ‘out-of-synch’ issues.
 - Is more complex than synching ‘current time snapshot’ where the same data is always overwritten/updated

Regional-Publishing GOCDDB: Change Log

- A Change Log would be maintained by Regional GOCDDB
- For each and every single CRUD Action, a new row would be inserted into a change-log table.
 - Create/Update: Current object state is recorded (e.g. serialized XML)
 - Restriction: Customisations must not break the change log. 

id	Tx_ID	ObjectID	Tstamp	UserID	Operation	Synched	Object state (XML/FK to archive table)
1	5	30	1203..	4	Create	True	<site>...site data elided...</site>
2	6	30	2336..	5	Update	True	<site>...changed site data...</site>
3	7	40	2335..	8	Create	False	<ServiceEndpoint>...<ServiceEndpoint>
4	7	41	2135..	6	Create	False	<link>parent = 30, child = 40</link>



Build an XML Synch Doc for publishing

Regional-Publishing: Sync Doc

- Chronological record of all Object Actions (insertion, linking, update, delete).
- Changes published to the central GOCDB.
 - Batch job, event-driven, manually via admin interface
 - Published Atomically in single Transaction (Tx₁)
 - Split change log into chunks
 - User role checking required to assert permissions.

(Tx₁)

```
<?xml version="1.0" encoding="UTF-8"?>
<Entities>
  <service_endpoint operation="update">
    <primary_key>449G3</primary_key>
    <hostname>test.host.com</hostname>
    <service_type>APEL</service_type>
    <ip>10.0.0.1</ip>
    <site>RAL-LCG2</site>
    <scope><tag>EGI</tag></scope>
  </service_endpoint>

  <user operation="delete">
    <primary_key>392G3</primary_key>
    <forename>Fred</forename>
    <surname>Flintstone</surname>
    <email>fred@egi.eu</email>
    <certificate_dn>/some/dn</certificate
    <scope><tag>EGI</tag></scope>
  </user>

  <downtime operation="add">
    <primary_key>371G3</primary_key>
    <hostname>test.host.com</hostname>
    <service_type>APEL</service_type>
    <endpoint>test.host.comAPEL</endpoint>
    <severity>OUTAGE</severity>
    <insert_date>1261486602</insert_date>
    <start_date>1261486500</start_date>
    <end_date>1261497600</end_date>
    <scope><tag>EGI</tag></scope>
  </downtime>
</Entities>
```

1st

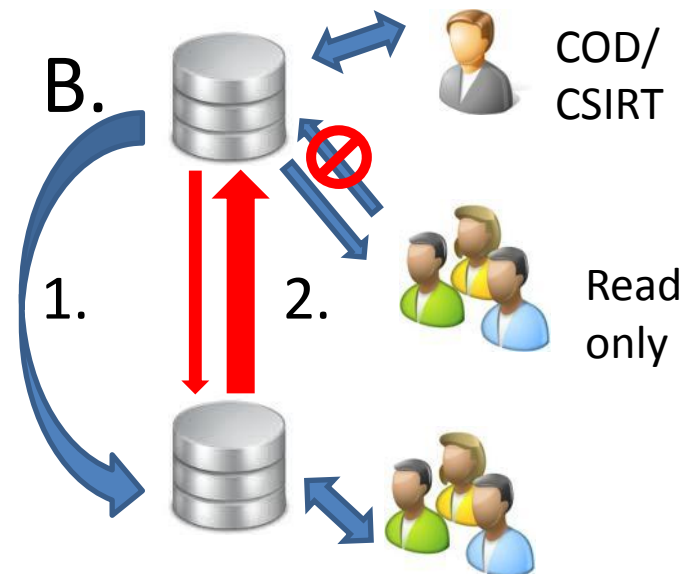
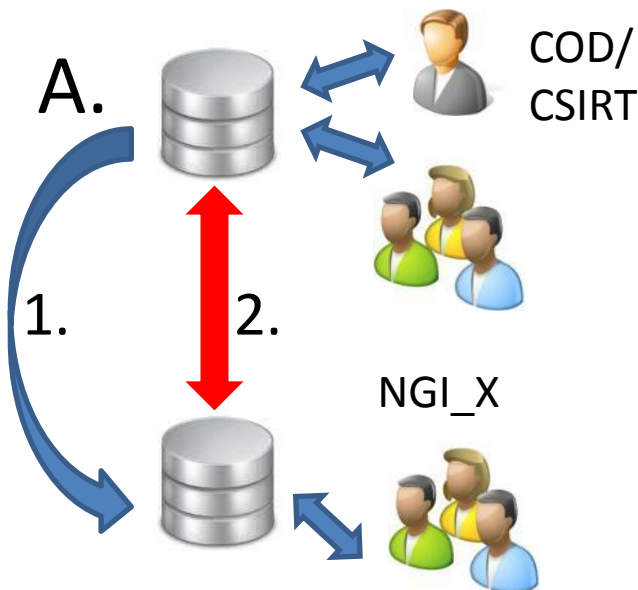
Chronological Object Actions

last

(simplified sample only)

Regional-Publishing GOCDDB: 2-Way Syncing

- A. Updates should be allowed from either instance (COD/CSIRT).
- Requires a 2 way synch (complex, potential for out-of-sync errors is greatly increased). Full sync probably not possible.
- B. Block central edits for NGIs with a regional instance
- Still allow central updates (COD/CSIRT teams) = requires 2 way sync of a sub-set of data.



Regional-Publishing GOCDB: Summary

- GOCDB Regionalisation is complex
 - Data history, 2 way synching, central updates/overwrites e.g. by COD/CSIRT = out of sync issues likely = greater support.
- Recent responses from NGIs suggest that scoping/regional views in central GOCDB seem to accommodate most regionalisation requirements
 - (Except customisation beyond custom SE types and Property bags).
- Feasible (2 x 0.5 FTEs)? / Required?

Info Discovery: Property Bags

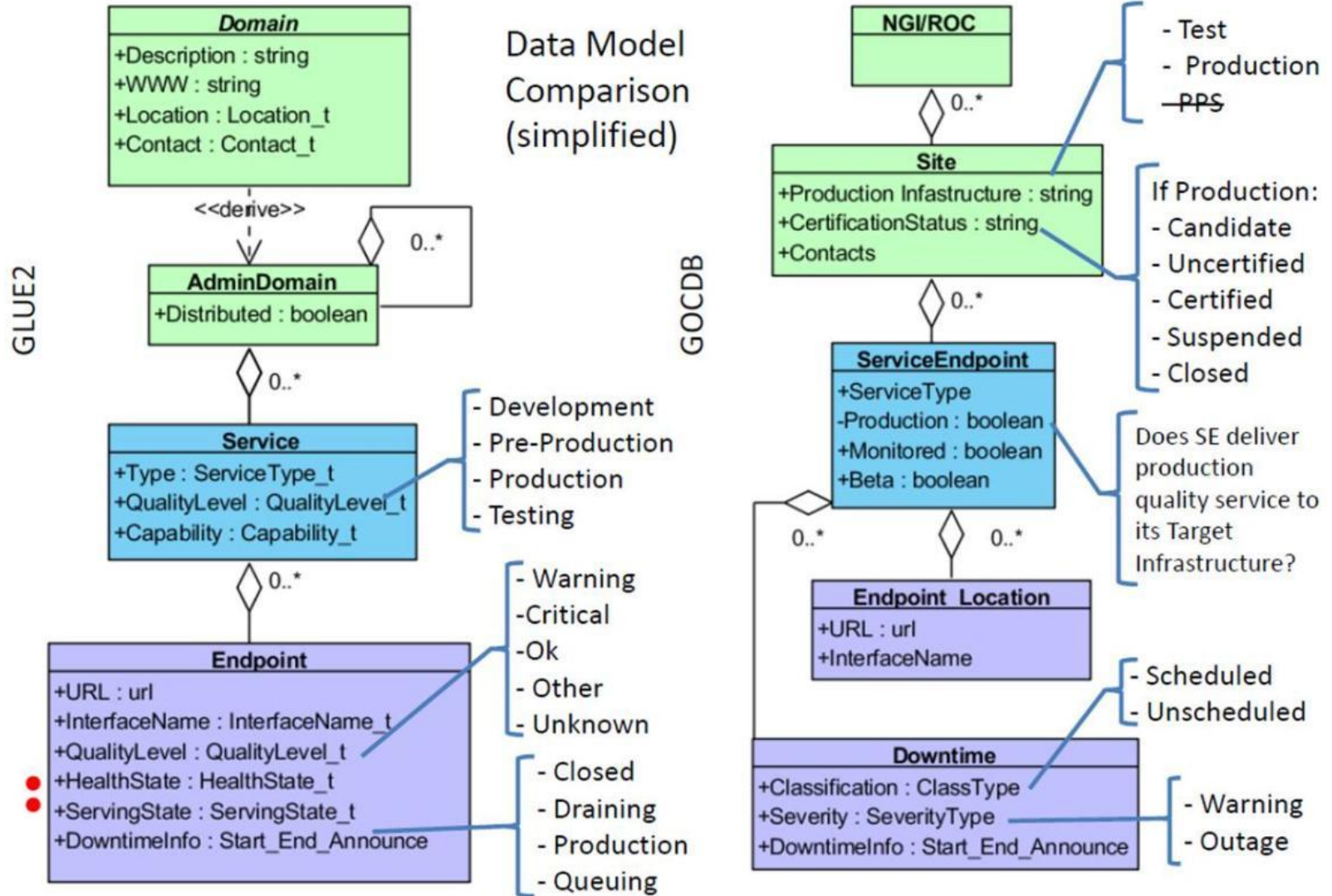
```
<SERVICE_ENDPOINT PRIMARY_KEY="50257G0">
  <PRIMARY_KEY>123456G0</PRIMARY_KEY>
  <HOSTNAME>some.host.ac.uk</HOSTNAME>
  <GOCDB_PORTAL_URL>https://goc.egi.eu/p
  <HOST_OS>SL5</HOST_OS>
  <BETA>N</BETA>
  <SERVICE_TYPE>SomeComputeService</SERV
  <IN_PRODUCTION>Y</IN_PRODUCTION>
  <NODE_MONITORED>Y</NODE_MONITORED>
  <SITENAME>DGIREF</SITENAME>
  <COUNTRY_NAME>Germany</COUNTRY_NAME>
  <COUNTRY_CODE>DE</COUNTRY_CODE>
  <ROC_NAME>NGI_DE</ROC_NAME>
  <ENDPOINT>
    <URL>ldap://ce-cms.vinca.rs:2170/md
    <InterfaceName>RIS</InterfaceName>
  </ENDPOINT>
  <Extensions>
    <Extension>
      <LocalID>1234</LocalID>
      <Key>MPI</Key>
      <Value>True</Value>
    </Extension>
    <Extension>
      <LocalID>12345</LocalID>
      <Key>somekey</Key>
      <Value>valueX</Value>
    </Extension>
  </Extensions>
</SERVICE_ENDPOINT>
```

- A means to customise PI queries
- User cases reported in RT + EUDAT (<https://rt.egi.eu/rt/Ticket/Display.html?id=3764>)
- Core GOCDB entities extended to define an optional set of custom key-value pairs
 - Users, Sites, SEs, ServiceGroups
 - Follows GLUE2 <Extensions>
- Query support in PI using ‘properties’ GET parameter with an LDAP query string value:
- [get service endpoint&properties=\(MPI=True\)](#)
- [get service endpoint&properties=\(MPI=*\)](#)
- [get service endpoint&properties=\(&\(MPI=*\)\(somekey=valueX\)\)](#)

[https://goc.egi.eu/gocdbpi/private/?method=get_service_endpoint&properties=\(&\(MPI=*\)\(somekey=valueX\)\)](https://goc.egi.eu/gocdbpi/private/?method=get_service_endpoint&properties=(&(MPI=*)(somekey=valueX)))

GLUE2: Data Model Comparison


- GOCDB data corresponds to a sub-set of GLUE2.



- Put selected Endpoints into DT

- Put all Service Endpoints into DT
- Downtime history

GLUE2: GOCDDB PI

- Add new PI methods to render GOCDDB in GLUE2 XML.
- PI is a set of Projection style queries
 - Projection queries simply specify the entities you need to render when building a SELECT query (for SQL, you would normally specify fields/cols).
- GOCDDB provides 18 projection style methods:
 - get_service_endpoint
 - get_ngi
 - get_site
 - get_contact
 - get_downtime
 - get_site_contacts ...

All support different GET request parameters to refine searches

GOcdb PI Renderings

```
<results>
<ROC ROC_NAME="NGI_UK">
  <ROCNAME>NGI_UK</ROCNAME>
  <MAIL_CONTACT>UKNGI-OPERATIONS@jiscma
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT USER_ID="5955046" PRIMARY_KEY="44479G0">
    <FORENAME>John</FORENAME>
    <SURNAME>Kewley</SURNAME>
    <TITLE>Mr</TITLE>
    <DESCRIPTION/>
    <EMAIL>john.kewley@stfc.ac.uk</EMAIL>
    <TEL>+44 1925 603513</TEL>
    <WORKING_HOURS_START/>
    <WORKING_HOURS_END/>
    <CERTDN>/C=UK/O=eScience/OU=CLRC/
    <ROLE_NAME>NGI Operations Deputy
  </CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
  <CONTACT>
```

1.

```
<results>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME ID="25994" PRIMARY_KEY="44479G0" CI="44479G0">
  <PRIMARY_KEY>44479G0</PRIMARY_KEY>
  <HOSTNAME>goc.egi.eu</HOSTNAME>
  <SERVICE_TYPE>egi.GOCDB</SERVICE_TYPE>
  <ENDPOINT>goc.egi.eu/egi.GOCDB</ENDPOINT>
  <HOSTED_BY>GRIDOPS-GOCDB</HOSTED_BY>
  <GOCDB_PORTAL_URL>https://goc.egi.eu/portal
  <SEVERITY>WARNING</SEVERITY>
  <DESCRIPTION>Network disruptions to allow
  <INSERT_DATE>1299751740</INSERT_DATE>
  <START_DATE>1300176000</START_DATE>
  <END_DATE>1300190400</END_DATE>
  <FORMATED_START_DATE>2011-03-15 08:00</FORMATED_START_DATE>
  <FORMATED_END_DATE>2011-03-15 12:00</FORMATED_END_DATE>
</DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
<DOWNTIME>
```

2.

~ Consider 1000's of records = can produce large XML documents.

- 1) https://goc.egi.eu/gocdbpi/private/?method=get_roc_contacts&roc=NGI_UK
- 2) https://goc.egi.eu/gocdbpi/public/?method=get_downtime&topentity=GOcdb

GLUE2: Flat XML Rendering

```
<glue:Entities
  xmlns:xsi='http://www.w3.org
  xmlns:glue='http://schemas.
  xsi:schemaLocation='http://
  <glue:Location>
  <glue:Contact>
  <glue:Contact>
  <glue:UserDomain>
  <glue:AdminDomain>
  <glue:AdminDomain>
  <glue:Service>
  <glue:ComputingService>
  <glue:StorageService>
  <glue:Endpoint>
  <glue:ComputingEndpoint>
  <glue:StorageEndpoint>
  <glue:ComputingShare>
  <glue:StorageShare>
  <glue:ComputingManager>
  <glue:StorageManager>
  <glue:DataStore>
  <glue:ExecutionEnvironment>
  <glue:Activity>
  <glue:ComputingActivity>
  <glue:AccessPolicy>
  <glue:MappingPolicy>
</glue:Entities>
```

- OGF 35 voted for a flat XML style
- Relationships modelled using element ID references (rather than nesting parent/child: many parents possible).
- Traversing associations requires sub-queries (cross referencing element IDs) but is flexible.

```
<glue:ComputingService BaseType="Service">
  <glue:ID>computingService1</glue:ID>
  <glue:Type></glue:Type>
  <glue:QualityLevel>production</glue:QualityLevel>
  <glue:Associations>
    <glue:ComputingEndpointID>computingEndpoint1</
  </glue:Associations>
</glue:ComputingService>

<glue:ComputingEndpoint BaseType="Endpoint">
  <glue:ID>computingEndpoint1</glue:ID>
  <glue:URL>uri://some.url.ac.uk/service</glue:URL>
  <glue:InterfaceName></glue:InterfaceName>
  <glue:QualityLevel>development</glue:QualityLevel>
  <glue:HealthState>ok</glue:HealthState>
  <glue:ServingState>production</glue:ServingState>
  <glue:Associations>
    <glue:ComputingServiceID>computingService1</gl
  </glue:Associations>
</glue:ComputingEndpoint>
```

(elements are collapsed)

```

<?xml version="1.0" encoding="UTF-8"?>
<glue:Entities
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:glue='http://schemas.org/glue'
  xsi:schemaLocation='http://schemas.org/glue'
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:StorageService>
  <glue:Endpoint BaseType="Endpoint">
    <ID>endpoint1</ID>
    <Extensions>
    <URL>uri://some.url.ac.uk/service</URL>
    <InterfaceName></InterfaceName>
    <QualityLevel>development</QualityLevel>
    <HealthState>ok</HealthState>
    <ServingState>production</ServingState>
    <DowntimeAnnounce>2012-03-29T12:30:00Z</DowntimeAnnounce>
    <DowntimeStart>2012-04-29T12:30:00Z</DowntimeStart>
    <DowntimeEnd>2012-05-29T12:30:00Z</DowntimeEnd>
    <DowntimeInfo>We had a power outage!</DowntimeInfo>
    <Associations>
  </glue:Endpoint>
  <glue:ComputingEndpoint>
  <glue:StorageEndpoint>
  <glue:StorageEndpoint>
  <glue:StorageEndpoint>
</glue:Entities>

```

Sample Flat Rendering

'get_service_endpoint&roc=NGI_X'

- Entity elements:
minOccurs=0, maxOccurs=unbounded
- Can select/render (project) just the required entities under the same Doc root element.
- Efficient: No redundant data (consider 1000s of records).

Summary

- GOCDB Regionalisation is complex (data history, central updates, 2-way synching).
- Scoping/regional views in central GOCDB seem to accommodate most regionalisation requirements.
- Property Bags with PI query support, and Custom Service types should cater for most info discovery requirements.
- GLUE2: Projected data is easily rendered using Flat XSD.
 - GLUE2 session for more details (Thurs 16:00).
- OTAG for other future GOCDB directions (Fri).