THE UNIVERSITY OF MELBOURNE

Tom Fifield
fifieldt@unimelb.edu.au

Prototyping a Distributed Computing Environment for SuperBelle

Motivation

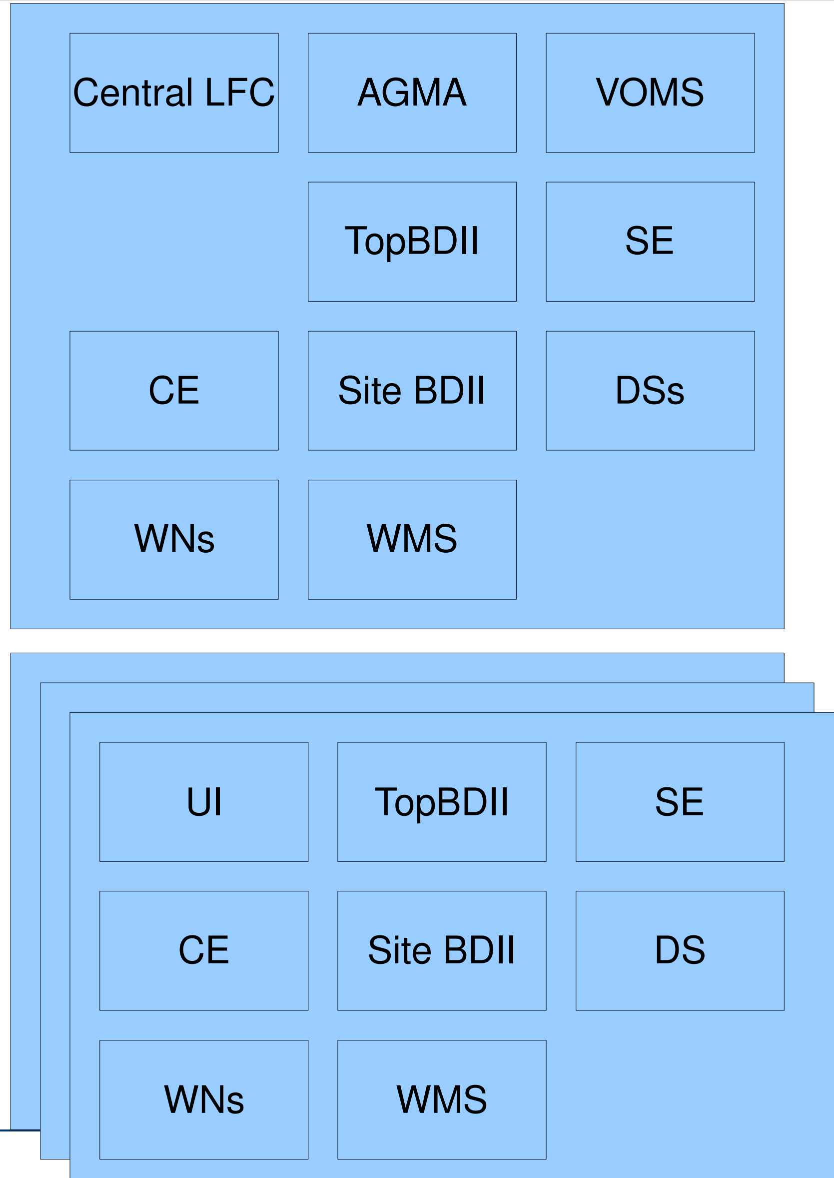
- We need a distributed computing solution for SuperBelle
- Thomas Kuhr presented *Data Handling at CDF*
- Soonwook Hwang presented *AMGA Metadata Catalogue*
- Idea: Start work on a gLite-based solution encompassing these ideas immediately

THE UNIVERSITY OF MELBOURNE

The architecture

"Tier 0"
KEK

"Tier 1"
Germany,
Korea,
Aus etc



That looks really complicated

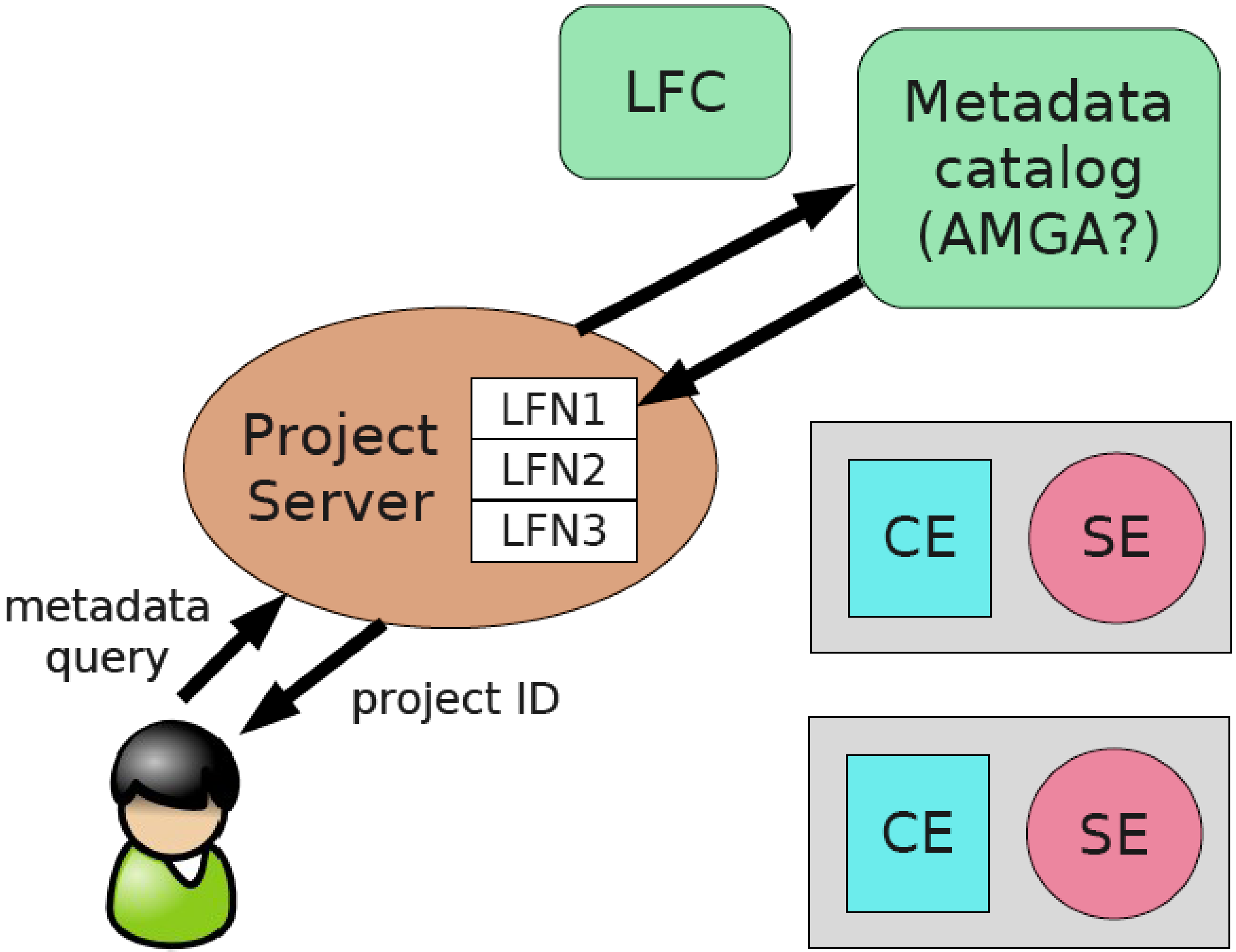
Let's divide into two stages

- Stage 1: Focus on storage
- Stage 2: Complete computing tasks

Recall

- These will be explained in detail later:
  - CE = Compute Element, coordinates access to computing resources
  - SE = Storage Element, coordinates access to storage resources
  - LFC = LCG File Catalogue, keeps track of file locations on the grid
  - AMGA = ARDA Metadata Grid Application, keeps track of data about files

End of Stage 1



Thomas Kuhr, 2009-02-17

Stage 1

- Get to the point where a user is returned a project ID
  - For now, we ignore submission to Compute Element and running the job
  - This abstraction makes it simpler, and easier to prototype storage ideas

A Test Environment

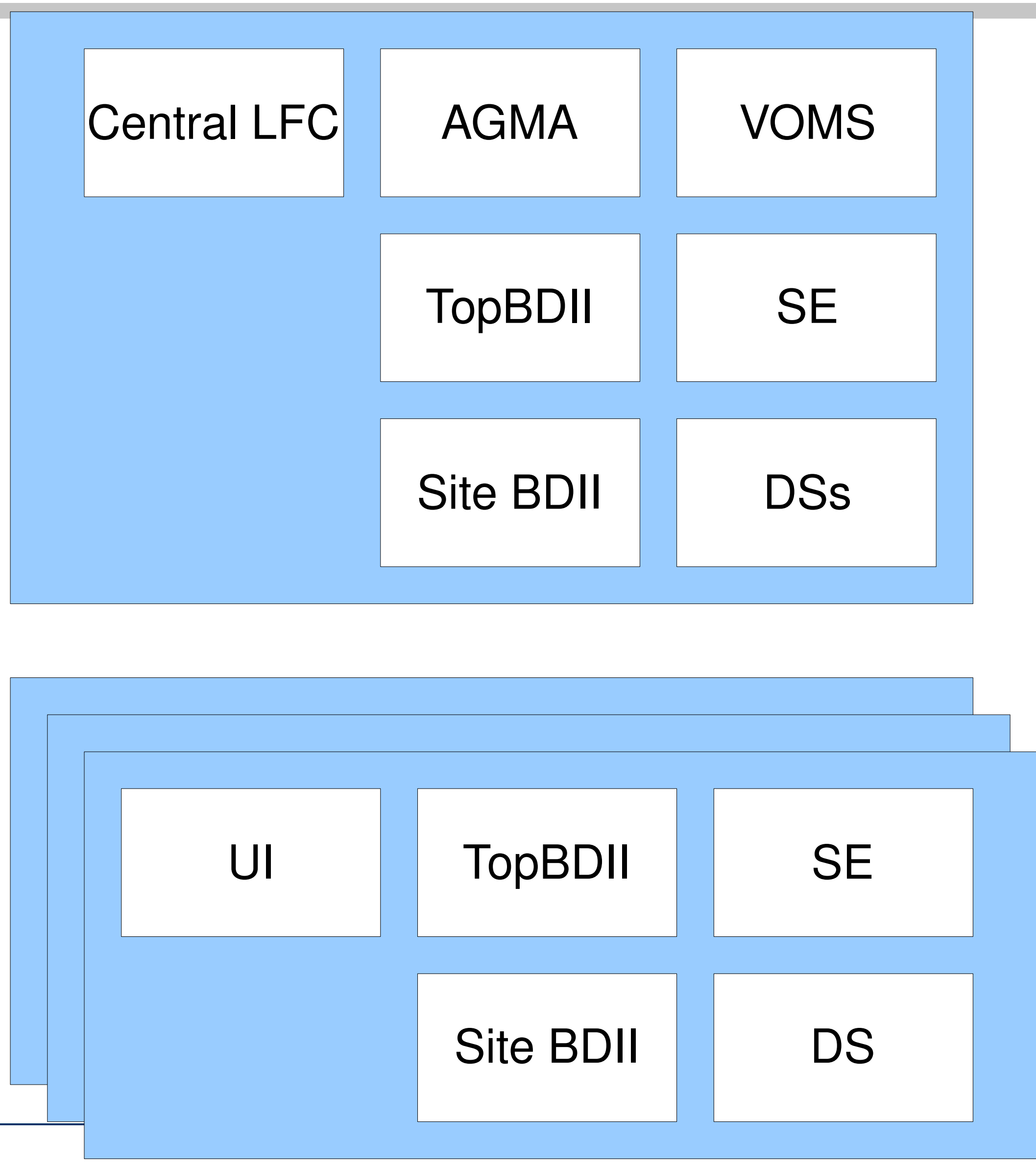
- Tiered architecture
  - KEK has much of storage/processing, but this allows other sites to give access to their users locally
- Possible to simulate in ~12 **gLite** services across 2 virtual sites (12 CPUs, 12GB Ram, 200Gb disk) [production needs a lot more!]
  - Use virtual machines = low to zero cost

THE UNIVERSITY OF MELBOURNE

gLite Components – Stage 1

"Tier 0"
KEK

"Tier 1"
Germany,
Korea,
Aus etc



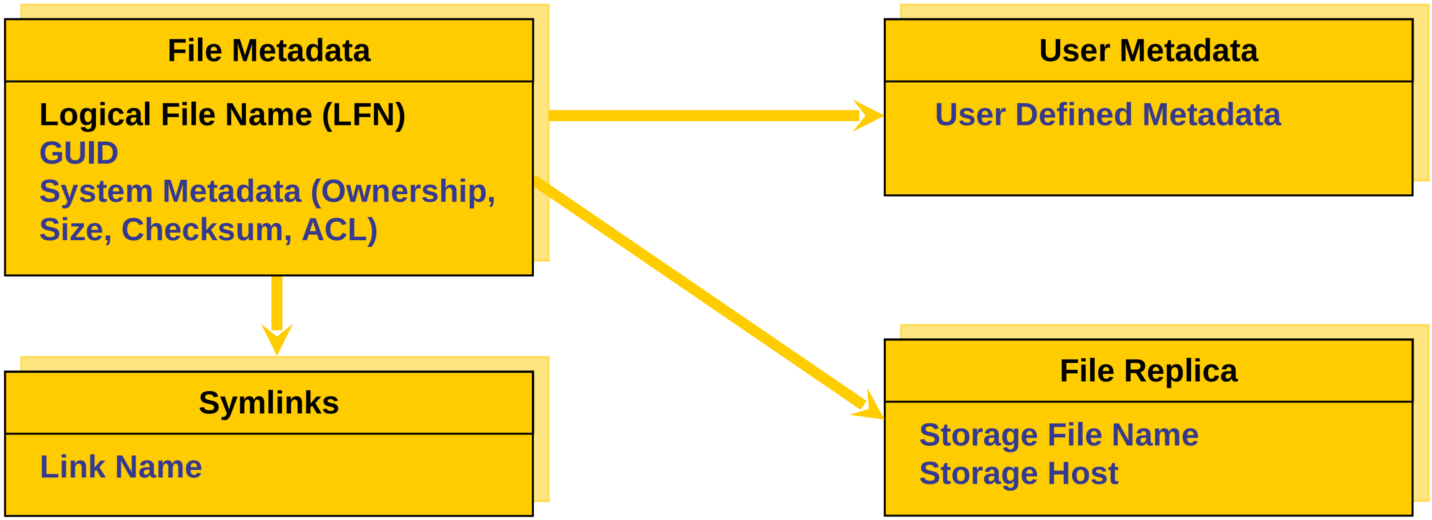
LFC?

- LCG File Catalogue
  - Designed for performance and scalability
  - Oracle or MySQL backend
  - Maps between a logical file name and its physical location(s)

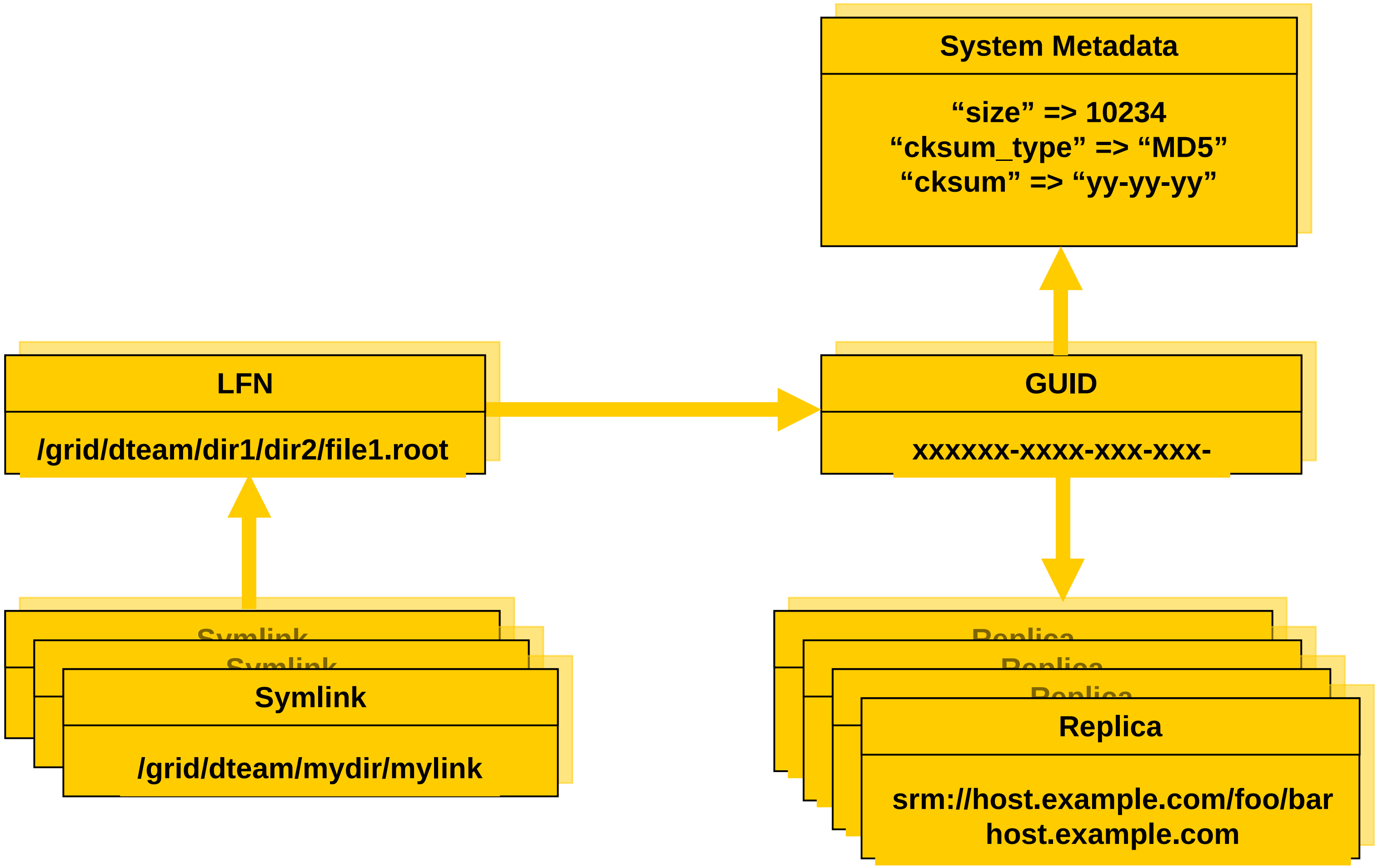
LFC?

- LCG File Catalogue
  - Data in Datasets
  - Datasets comprised of files
  - Can have multiple copies – either partial or complete datasets – stored anywhere in the grid

LFN


File Metadata
Logical File Name (LFN)
GUID
System Metadata (Ownership, Size, Checksum, ACL)
User Metadata
User Defined Metadata
Symlinks
Link Name
File Replica
Storage File Name
Storage Host

Example

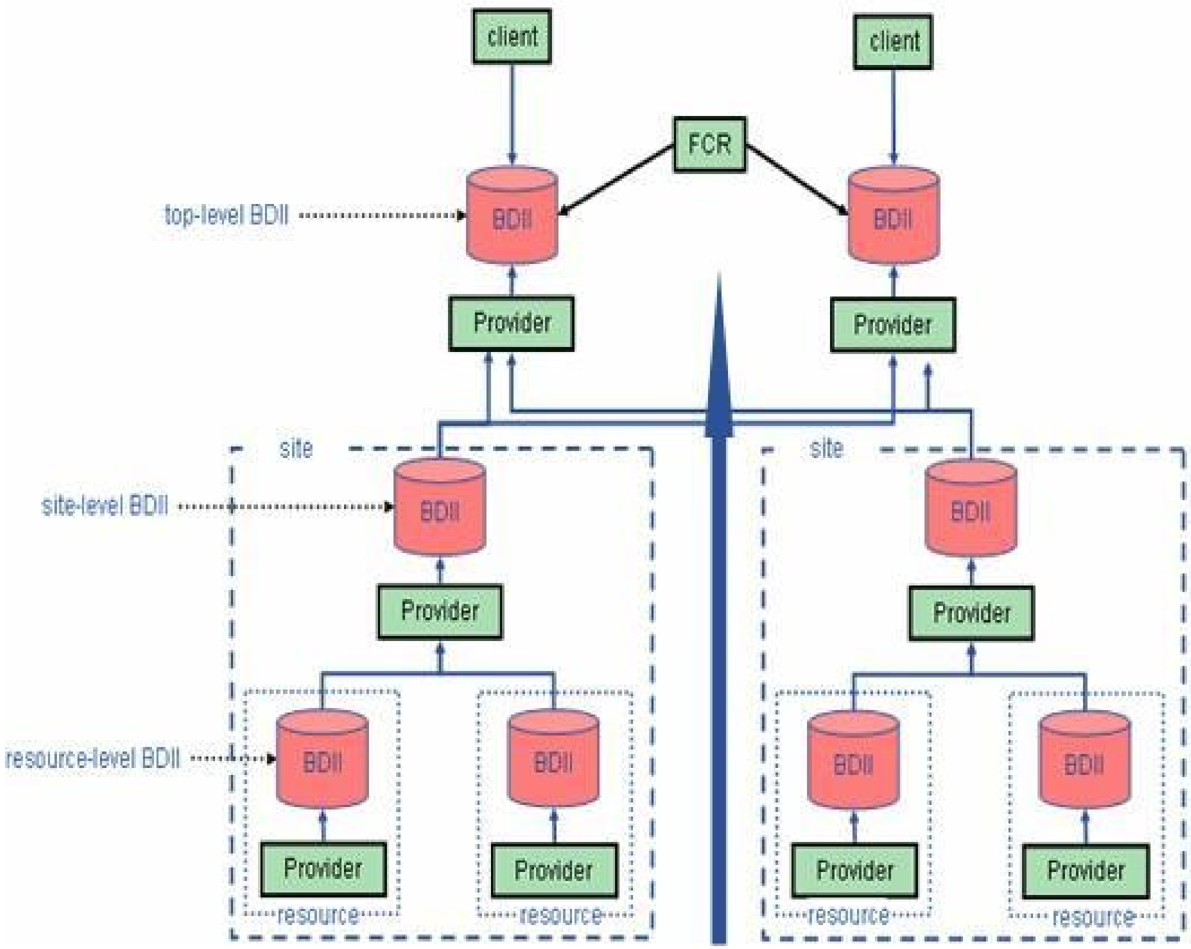

System Metadata
“size” => 10234
“cksum_type” => “MD5”
“cksum” => “yy-yy-yy”
LFN
/grid/dteam/dir1/dir2/file1.root
GUID
xxxxxx-xxxx-xxx-xxx-
Symlink
/grid/dteam/mydir/mylink
Replica
srm://host.example.com/foo/bar
host.example.com

- ARGA Metadata Grid Application
  - Metadata catalogue
  - A structured way of storing information
  - A powerful query interface

Information System

- BDII = Berkeley Database Information Index
- GLUE Schema – common data model for Grid resources
  - List of services at a site
  - Available storage
  - Number of cpus free/in use
  - Queue information
- Resource level → Site level → Top level

Information System



Information Flow

THE UNIVERSITY OF MELBOURNE

Example

```
siteName:            Australia-ATLAS
Web:                 http://epp.ph.unimelb.edu.au/
Location:            Melbourne, Australia
Latitude:            -37.48141
Longitude:           144.57351
...
agh2.atlas.unimelb.edu.au:2119/jobmanager-lcgpbs-belle
   GlueCEStateStatus:                    Production
   GlueCEPolicyMaxRunningJobs:           0
   GlueCEPolicyMaxWallClockTime:         4320
...
SubClusters:
agh2.atlas.unimelb.edu.au
   GlueHostOperatingSystemName:          ScientificSL
   GlueHostOperatingSystemRelease:       4.6
   GlueHostOperatingSystemVersion:       Beryllium
   GlueSubClusterPhysicalCPUs:           80
   GlueSubClusterLogicalCPUs: 88
...
GlueSEUniqueID:      agh3.atlas.unimelb.edu.au
   GlueSEName:       Australia-ATLAS:srm_v1
```

User Interface

- Machine has
  - user accounts
  - installed tools for working with grid
  - access to submit jobs to grid

Storage

- SE = Storage Element
  - Provides uniform access to storage
  - Abstraction allows for different hardware (Disk Servers) to run behind different SEs: tape, arrays, disk
  - PFN/SURL → SE → TURL

Architecture - Stability

- ✓ Virtual Organisation Management (VOMS)
- Storage Element (SE)
  - ✓ dpm
  - ✓ dCache
  - ✗ Castor
- Information System (BDII)
- ✓ LCG File Catalog (LFC)
- ✓ User Interface (UI)

A Test Environment

- We have much of the build and configuration process automated and ready to go, using cfengine
- What is cfengine?

- “*Cfengine is an automated suite of programs for configuring and maintaining Unix-like computers*”
- We've turned what a grid node needs into a cfengine configuration.
  - Packages
  - Security: Firewall, user accounts, permissions etc
  - Configuration of services

What cfengine means

- Example: We need a new worker node
  1) Rack the hardware
  2) Assign an IP address to the hardware
  3) Tell cfengine that this machine is a worker node
  4) Turn the machine on
- We have this working for CE, WN, SE, DS, BDII, Mon, UI
  - No LFC, AGMA or VOMS

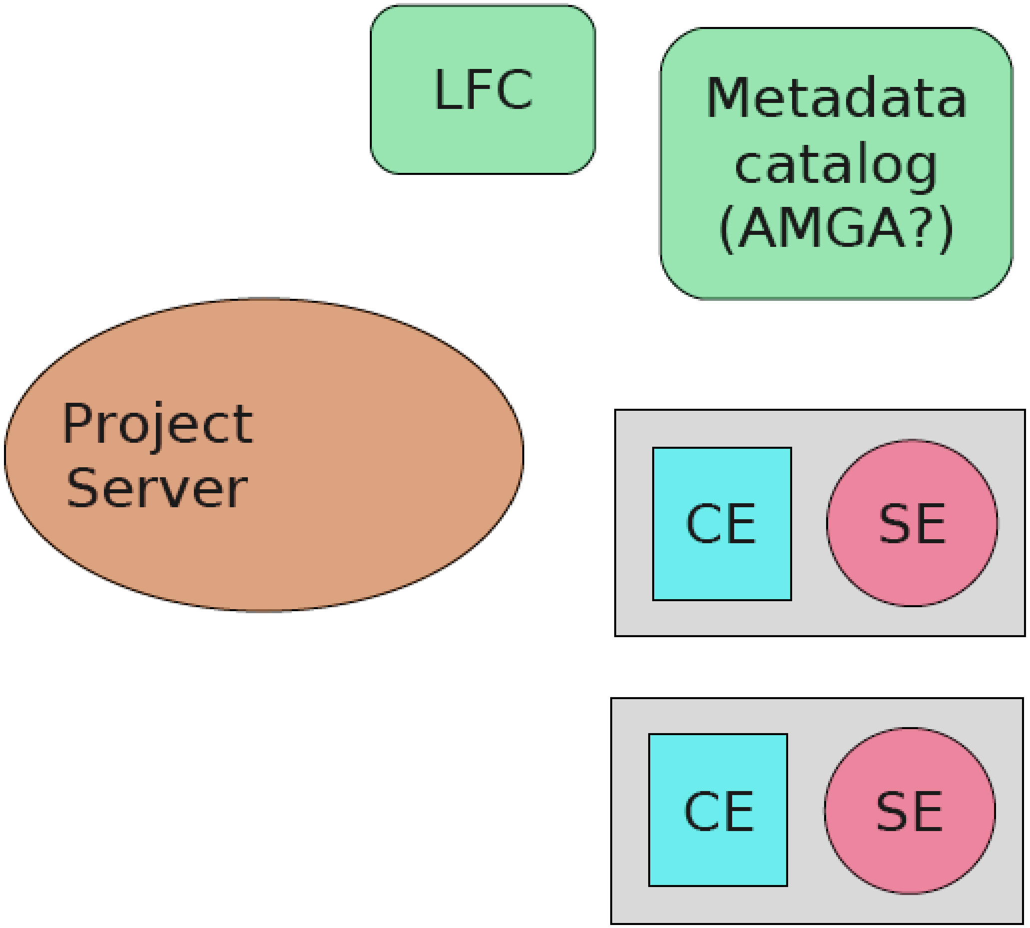
The Unknown Areas

- LCG tools alone don't deal with these:
- Data registration
  - What happens to data from the detector?
- Data distribution
  - What should be where, and how does it get there?
- User tools
  - How do users want to use the data?

Potential Solutions

- Data registration
  - Integration with AGMA (as by Hwang), LFC
- Data distribution
  - Proposal by Thomas Kuhr
- User tools
  - Write them.

Recall



The Plan – Stage 1

1) Install all grid services in virtual servers, using existing cfengine scripts for configuration as much as possible
2) Work on simulation:
   1) Registration of datasets and metadata
   2) Site administrator data management tools
   3) User query

The Plan – Stage 1 – Environment Setup

- Very minimum specs needs 12 cores:
  - Central LFC
  - VOMS
  - AGMA
  - UI
  - 2 × SE
  - 2 × Disk Server
  - 2 × Top BDII
  - 2 × Site BDII

The Plan – Stage 1 – Environment Setup

- SL4 with gLite can be done in 15GB disk
- Many services will run within 1GB of RAM
  - SE is an exception

```
24421 dpmmgr    16   0    0  39:03.57 25.5 1310m 1.0g 3952 S dpnsdaemon
24312 dpmmgr    16   0    0  39:46.75  6.9  779m 272m 4056 S dpm
 4394 mysql     16   0    0 484:53.40  0.7  174m  27m 3784 S mysqld
 6677 root      16   0    0   6:42.70  0.4  224m  14m 2832 S dsm_om_connsvc3
24464 dpmmgr    16   0    8 128:14.61  0.3  280m  13m 3972 S srmv1
24552 dpmmgr    16   0    0 322:38.70  0.3  280m  12m 4280 S srmv2.2
```

- This is going to be slow
- More concerned with software feasibility
  - The gLite components have already proven their performance

The Plan – Stage 1 – Environment Setup

- For example, 2 × 8-core machines with 16Gb of RAM each would be safe
  - – eg Dell PowerEdge 1950

The Plan – Stage 1 – Simulation

- We need to:
  - Emulate the creation of datasets, including:
    - Copying physical files to disk servers via SE
    - Registering physical files as datasets in LFC
    - Registering metadata in AGMA
  - Allow the distribution datasets, by creating tools for site administrators that:
    - Facilitate the transfer of files from other SEs
    - Register the new replicated physical files in the LFC

The Plan – Stage 1 – Simulation

- We need to:
  - Work on the Project Server to ensure
    - It can communicate with LFC and AGMA to determine dataset locations based on metadata
    - This is the link we need to provide – core of the project

The Plan – Stage 1 – Measuring Success

- If the system works end-to-end, we've done our job:
  1) A dataset is created and registered in LFC and AGMA at “Tier 0”
  2) The dataset is copied to the “Tier 1” site
  3) A user puts in a metadata request and receives back the project ID and locations (“Tier 0”, “Tier 1”) hosting the dataset
- This will prove the first part of the model suggested by Kuhr, and leave us with software to move to completion

Progress

- Working on defining requirements
- Use cases from Katayama san
- Data Handling
  - The system must allow individual users access to raw data (Use Case #4)
  - The system must support returning histograms and other information (Use Case #4, Use Case #9)
  - The system must facilitate the access of another users data, within permission restrictions (Use Case #7)
  - The system must interface with the QAM System (Use Case #13)
- Job Handling
  - The system must support masses of parallel Jobs to a fine grain (Use Case #3)
  - The system must support different run lengths for jobs (Use Case #3)
  - The system must minimize idle CPU time between jobs (Use Case #3)
  - The system should support the ability to re-run with certain modules only (Use Case #3)
  - The system must allow for priority to be given to certain short-running jobs (Use Case #4)
  - The system must allow for automatic determination of CPU and Storage resources to use, where the user does not specify them (Use Case #4)
  - The system must allow similar jobs to be run with minimal changes (i.e. parameterised jobs) (Use Case #5)
  - The system must provide an appropriate level of debugging information (Use Case #5)
  - The system must provide an interface that allows users to manage their jobs (Use Case #7)
  - The system must allow computation to be run on a selected number of events (Use Case #7)
  - The system must run on an entire data set with the shortest elapsed time possible (Use Case #7)
  - The system must allow the use of geant4 (Use Case #8)

Next steps

- Determine format of user metadata query
- Determine heirachy schema for storage
- Site manager tools to assist with Data Distribution - what do we need above lcg-rep?
- What can we use from SAM?

THE UNIVERSITY OF MELBOURNE

© Copyright The University of Melbourne 2008