# Running the Grid on lite "gLite"


Hamza Mehammed
National e-Science Centre, UK

06.05.2009

Edinburgh e-Science TOE
Training Outreach and Education

www.eu-egee.org

e-infrastructure

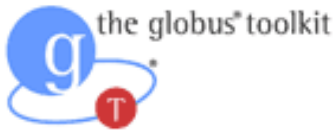
# gLite

- Grid middleware (production level)
- Giving the impression of having coherent virtual computing centre
- Integrates sites which provide batch system via a getkeeper
- Uses distributed WMS to accept  and manage jobs
- Sites provide CE service as common interface for WMS

# gLite

- **Platform dependent: Scientific Linux 4 or 3**
- **Glite 3.1 comes from**
  - DataGrid
  - DataTag
  - Globus
  - GriPhyN
  - iVDGL
  - EGEE and LCG






# gLite Components

- User Interface (UI)
- Workload Management System (WMS)
- Computing Element (CE)
- Workernode (WN)
- Storage Element (SE)

# User Interface (UI)

- Job submission machine
  - Must provide proxy certificate
- Provide CLI tools
  - List of resources
  - Job: execute, cancel, status, output, …
  - Logging and bookkeeping information
  - Files: copy, replicate and delete
  - Status of resources
- Use of WLCG/EGEE API

# Workload Management System (WMS)

- **WM maps user requirement to resources**
- **Implemented as distributed set of resource broker (e.g.: tens in EGEE)**
- **Resource broker gets information from GIS (not on every site)**
- **Uses Logging and Bookkeeping**
  - Track jobs in term of events:
    - Submitted, pending, allocated, …
- **WMS sends job execution requests to the CE using**
  - Condor-G

# WMS (Cont.)

- **Collect information from WMs and CEs**
- **Match-making**
  - Availability
  - Closeness
  - Rank (running/queued) : default # CPUs
  - BDII provides status of resources
- **RB locate files using Data Location Interface (DLI) to File Catalogue**
  - To support other File Catalogues than LFC

# Computing Element (CE)

- **Generic interface to the cluster**
- **Local Resource Management System (LRMS): batch system**
- **LRMS:**
  - PBS(open or pro), LSF, Maui/Torque, BQS and Condor
- **Grid Gate (GG): accept and dispatch jobs**
  - LCG CE (EDG) and gLite CE (EGEE)
- **Worker Node (WNs)**
  - Cluster Nodes, same commands as UI
  - VO specific SW in shared file system

# CE (Cont.)

- **CE = a single *queue* in the LRMS**
  - <gg_hostname>:<port>/<gg_type>-\
    <LRMS_type>-<batch_queue_name>
  - E.g.: lcg02.sinp.msu.ru:2119/blah-pbs-atlas
  - ⇒Different queues = different CE
- CE logs:
  - When it receive a job (scheduled)
  - When it's running
  - When it's done

# Storage Element (SE)

- **Uniform access to data storage resources**
  - Simple disc server
  - Large disc array
  - Tape-based (MSS)
- **Managed by Storage Resource Manager (SRM)**
  - Middleware service
  - Transparent file migration from disk to tape
  - Space reservation

# SE (Cont.)

- **Classic SE: GridFTP and RFIO (old)**
- ***Disk Pool Manager (DPM)***
  - Small SEs with disk-based storage only
- **CASTOR:**
  - disc buffer & MSS, virtual file system,
  - use RFIO for disc/tape file migration
- **dCache:**
  - Server frontend to pool nodes (dynamically)
  - virtual file system used as disk buffer to MSS
- **LCG Disk Pool Manager**
  - Lightweight disk pool manager (10TB) : sRFIO

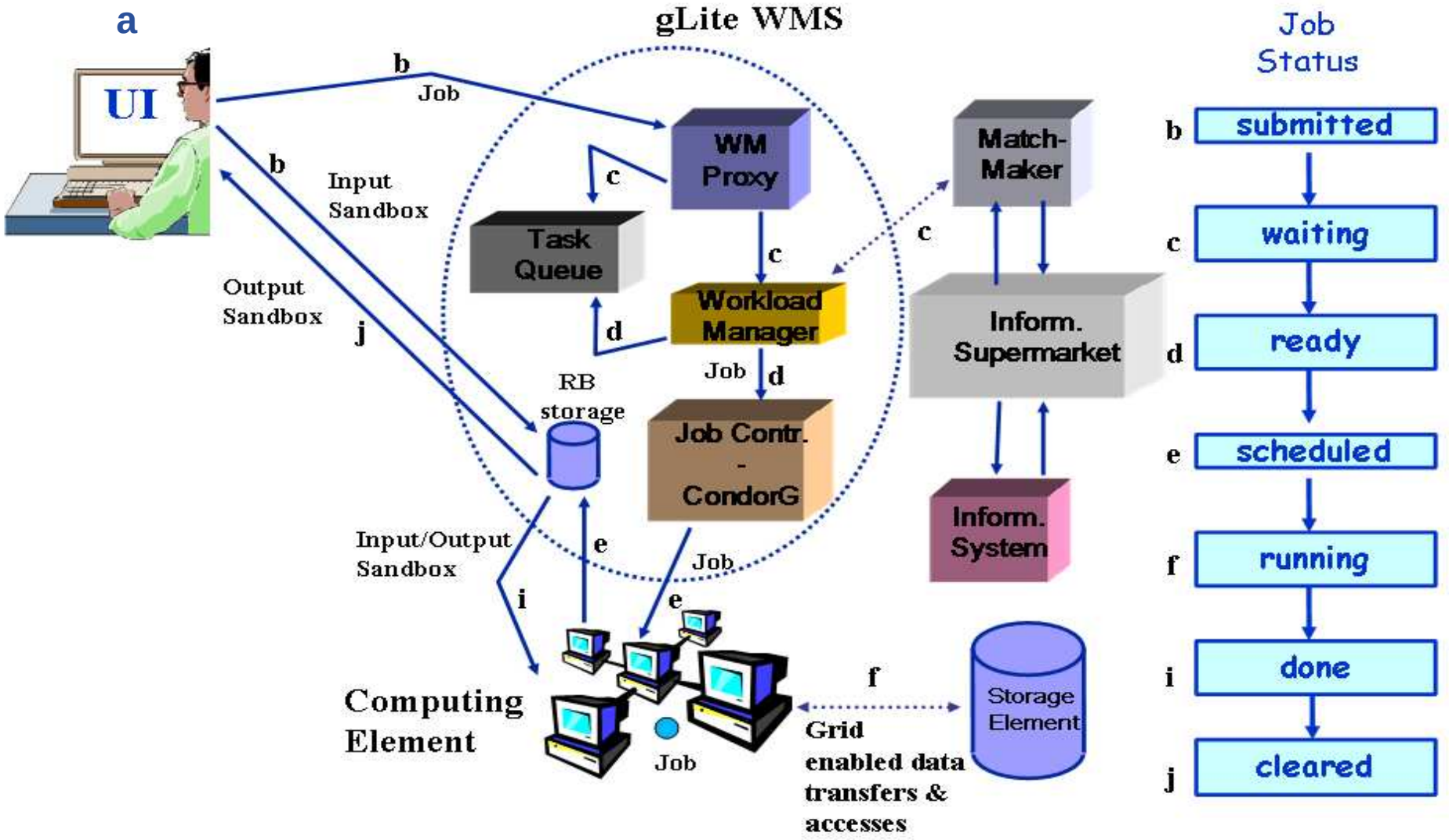
# Job Submission

- UI submit job
  - A job identifier will be returned
- Resource Broker (RB)
  - Responsible for the coordination of sandboxes
  - Takes care about proxy propagation
  - Performs matchmaking using Information Index
    - CE and SE
- Matchmaking can be modified by user via
    - Rank expression
    - Requirements field

# Job Submission (Cont.)

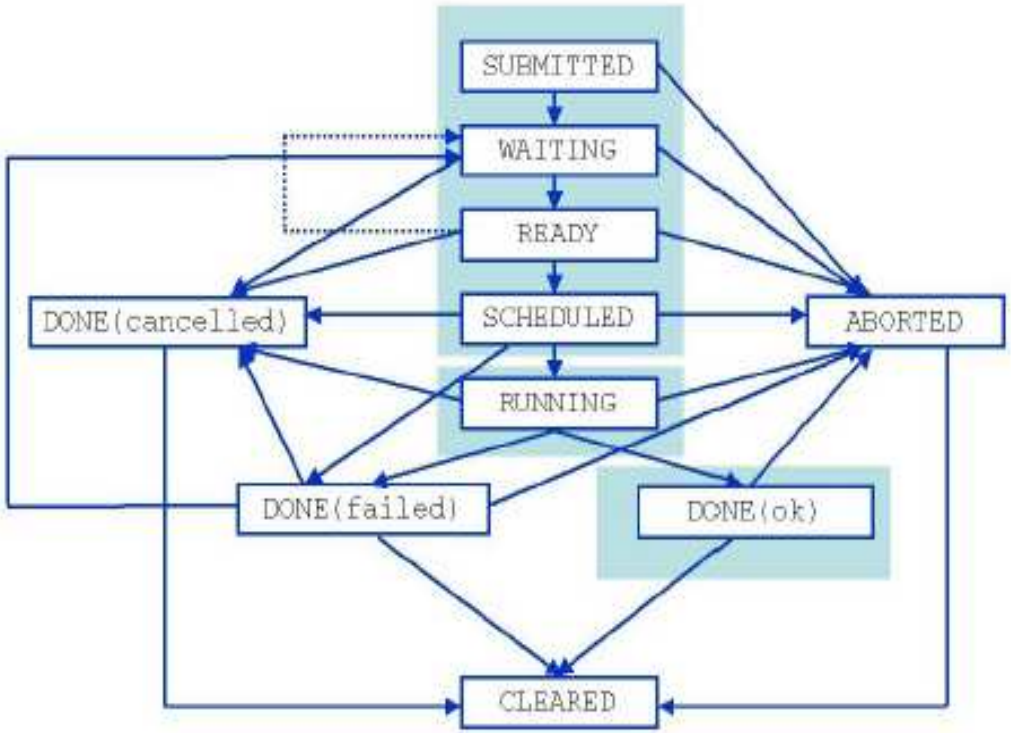
- When a CE receives a job, this is moved on a queue
  - Then the job will be executed on the first available among its WN (where the batch system clients run)
  - When execution is complete, output files are copied to the CE using scp
  - If the job is successfully executed, output files are copied back to the WMS using GridFTP

# Job Submission (Cont.)



# Jobs State Machine

| Flag | Meaning |
|---|---|
| SUBMITTED | submission logged in the Logging & Bookkeeping service |
| WAIT | job match making for resources |
| READY | job being sent to executing CE |
| SCHEDULED | job scheduled in the CE queue manager |
| RUNNING | job executing on a Worker Node of the selected CE queue |
| DONE | job terminated without grid errors |
| CLEARED | job output retrieved |
| ABORT | job aborted by middleware, check *reason* |



# Information Service (IS)

- ***Information Service***
  - Resource discovery
  - Monitoring
  - Accounting
- ***GLUE Schema***
  - Common conceptual data model to be used for Grid resource monitoring and discovery

# JOB DESCRIPTION LANGUAGE (JDL)

- **High-level language**
- **Describe jobs and aggregates of jobs**
  - Executable, input and output files, myproxy server, requirements, resubmission, …
- **A file containing: attribute = expression;**
- **To be used by the WMS**

# Simple JDL Example

- ***E.g.:***

```
Executable = "test.sh";
Arguments = "fileA fileB";
StdOutput = "std.out";
StdError = "std.err";
InputSandbox = {"test.sh", "fileA", "fileB"};
OutputSandbox = {"std.out", "std.err"};
```

where test.sh could be, for example:

```
#!/bin/sh
echo "First file:"
cat $1
echo "Second file:"
cat $2
```

# JDL

- **Conditions: !, && and ||**
- **Conditional Statements: x == y ? p : q**
- **Boolean expression**
  - Requirements = other.GlueCEInfoLRMSType == "PBS" && other.GlueCEInfoTotalCPUs > 1;
- **Particular queue**
  - Requirements = other.GlueCEUniqueID == "lxshare0286.cern.ch:2119/jobmanager-pbs-short";
- **Using wildcards**
  - Requirements = (!RegExp("cern.ch", other.GlueCEUniqueID));

# JDL (Cont.)

- **Environment variable:**
  - Environment = {"CMS_PATH=$HOME/cms",
    "CMS_DB=$CMS_PATH/cmdb"};
- **Resubmission by the WMS:**
  - RetryCount and ShallowRetryCount
- **Operating system and version:**
  - other.GlueHostOperatingSystemName
  - other.GlueHostOperatingSystemRelease
- **Disabling default**
  - attribute = “”;

# Goodness of a CE and Requirements

- **Default Rank =**
  - other.GlueCEStateEstimatedResponseTime
- **Rank = other.GlueCEStateFreeCPUs;**
- **Rank = (other.GlueCEStateWaitingJobs == 0 ? other.GlueCEStateFreeCPUs : \ other.GlueCEStateWaitingJobs);**
- **Requirements = other.GlueHostNetworkAdapterOutboundIP==true && Member("VO-alice-AliEn", other.GlueHostApplicationSoftwareRunTimeEnvironment) && (other.GlueCEPolicyMaxWallClockTime > 1440 );**

# Commands

- **glite-job-<XY> is deprecated**
- **glite-wms-job-<XY> is recommended**
- **glite-wms-job-submit [-d delegID] [-a] [-o joblist] jdlfile**
  - Submit JDL to WMS
- **glite-wms-job-submit -a test.jdl**
  - https://lb102.cern.ch:9000/Bla6RySximq_vQ
  - https://<LB_hostname>[:<port>]/<string>

# Commands

- **Listing matching CE**
  - Test JDL
  - Lists information about available CEs and SEs
  - Filter :rank and requirements expression

  # glite-wms-job-list-match -a --rank <jdl file>
- **glite-wms-job-status**
  - Fetch information about the job's current state
  - Contacts LB
- **glite-job-output**
  - Fetching the results from WMS to the UI

# Command line Interface

- **Job Management Service: WMproxy**
  - submission of job collections
  - faster authentication
  - faster match-making
  - faster response time for users
  - higher job throughput
- **Commands for:**
  - submit, status, logging, output, cancel, and delegate

# Proxy Delegation

- **Automatically**
- **Explicitly creating a delegated credential**
- **glite-wms-job-delegate-proxy -d <delegID>**
  - Random with –a
  - glite-wms-job-submit -d mydelegID test.jdl

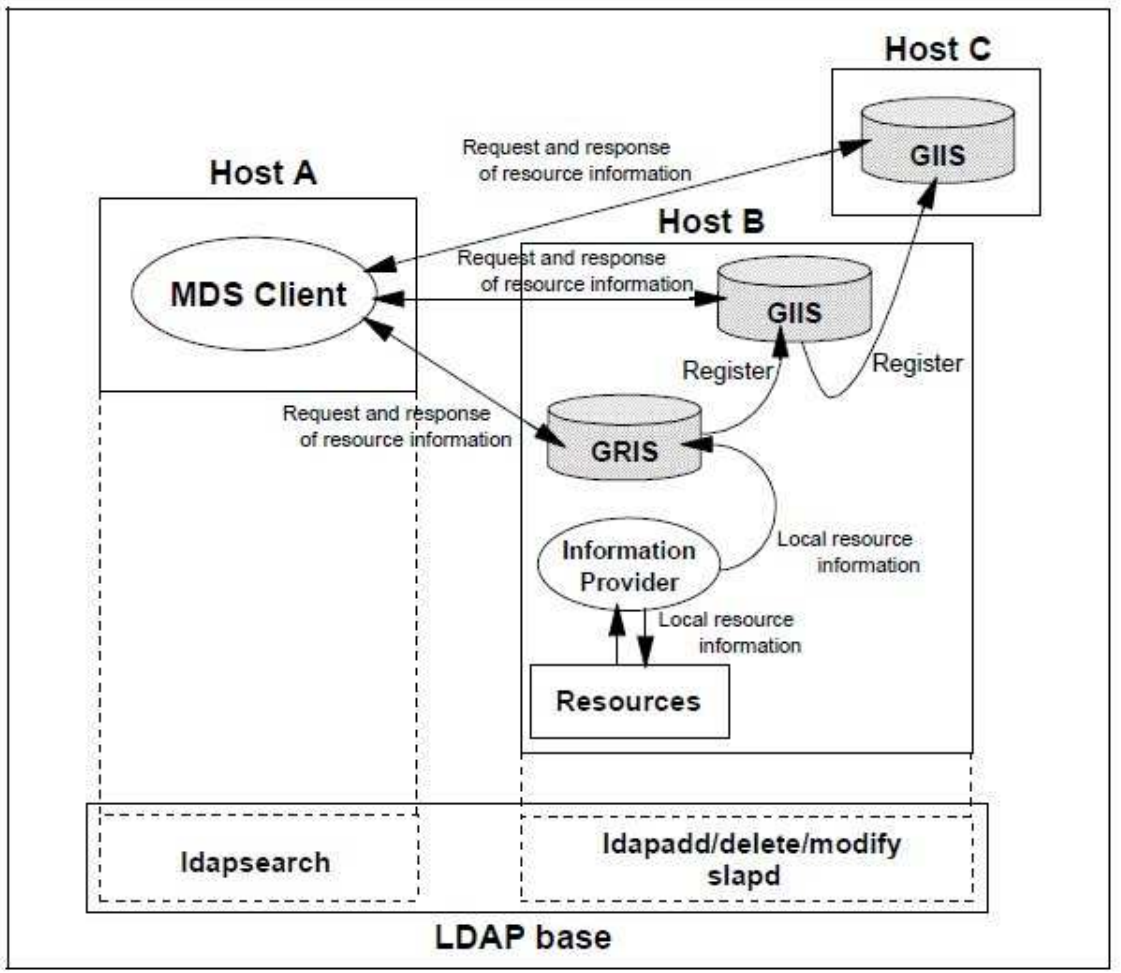
# MDS2

- **MDS provides access to static and dynamic information of resources**
- **Components:**
  - Grid Resource Information Service (GRIS)
  - Grid Index Information Service (GIIS)
  - Information Provider
  - MDS client

# GRIS and GIIS

- **GRIS:**
  - Repository of local resource information
  - GRIS registers its information with a GIIS
  - Information is updated when requested
  - Information is cached (time-to-live TTL)
- **GIIS**
  - Contains indexes of resource information
    - From GRIS and other GIISs
  - Can be seen as a grid wide information server
  - Has a hierarchical mechanism (like DNS)

# MDS2

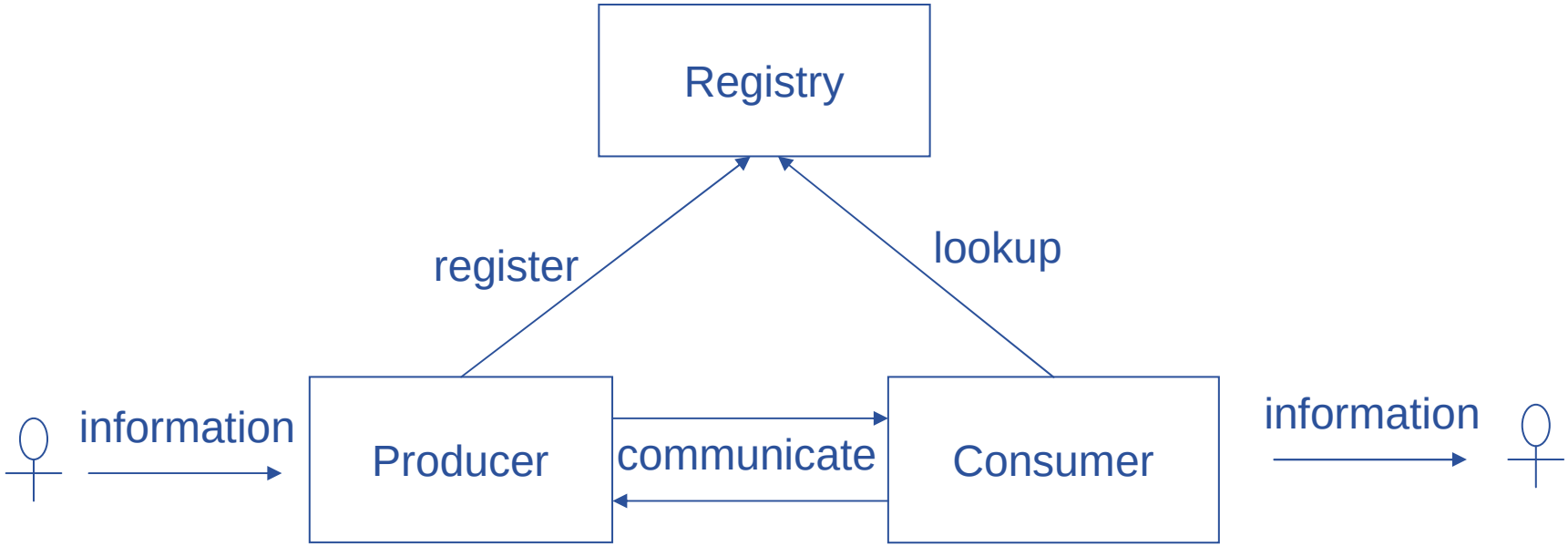



http://www.redbooks.ibm.com/

# MDS2 in gLite

- **Insecure Information provider on sites**
  - Published by LDAP server (GRIS)
- **BDII information from GRIS**
- **Site Admin => GOC**
- **Top level BDII from GOC**
- **Uses OpenLDAP to implement GLUE**
  - Specialized DB for reading, browsing and searching
- **DNs entries building Directory Information Tree (DIT)**

# Grid Monitoring Architecture (GMA)



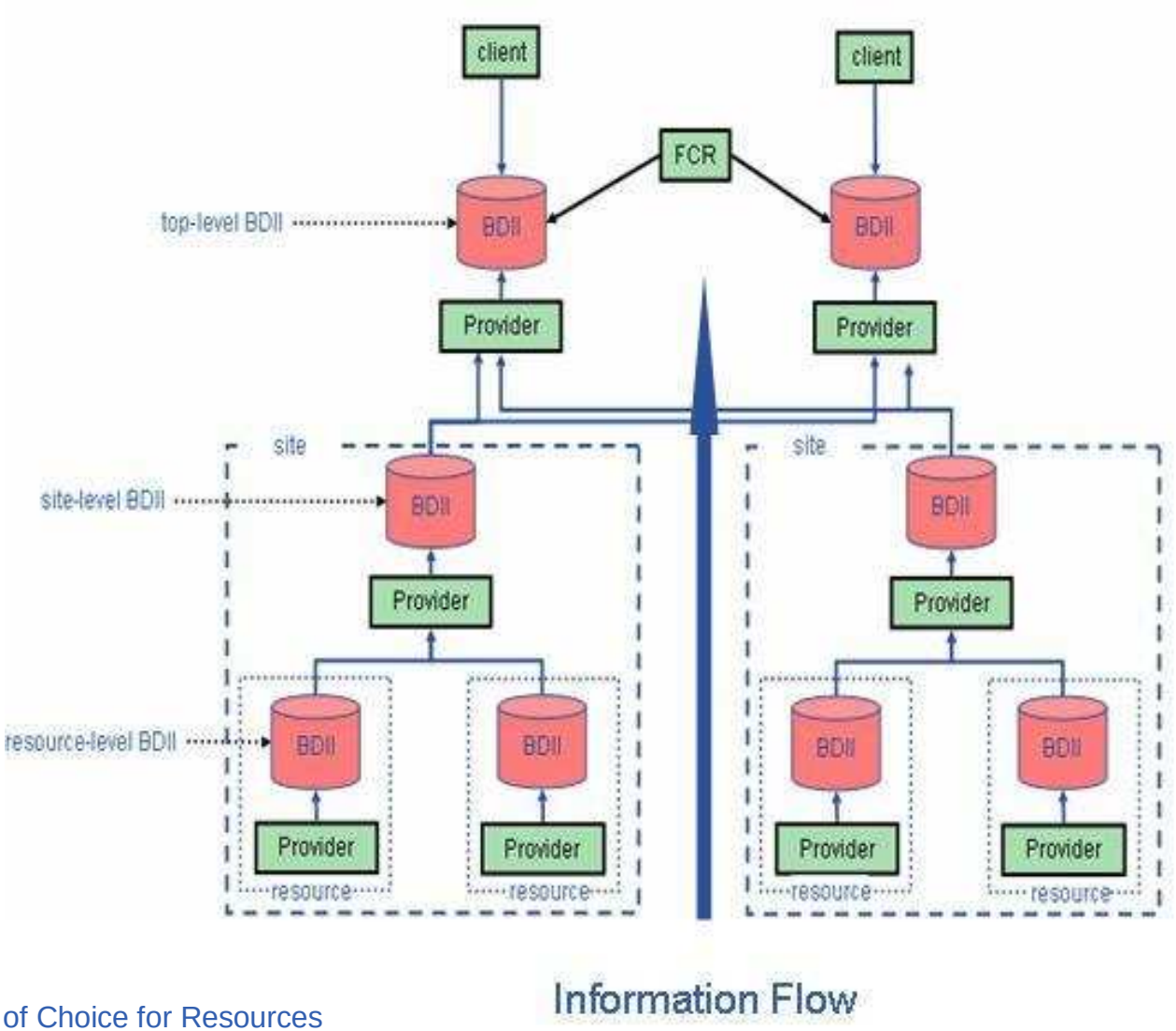
# Retrieving Information

- **lcg-infosite: VO specific info**
  - lcg-infosites --vo <vo> <option> -v <info> -f <site> --is <bdii>
  - Option: vo, ce, se, sitenames, lfc, …
  - E.g.: lcg-infosites --vo alice ce
  - Output: #CPU, Free, Total Jobs, Running, Waiting, CE
- lcg-info: CEs or SEs specific info
  - lcg-info [--list-ce | --list-se] [--query <query>] [--attrs <attrs>]
  - E.g. for all available attributes: lcg-info --list-attrs
  - lcg-info --vo cms --list-ce --query 'Processor=*thlon*, \ OS=*Scientific*' –attrs 'RunningJobs,FreeCPUs'
  - Output: Results per CE

# GRIS

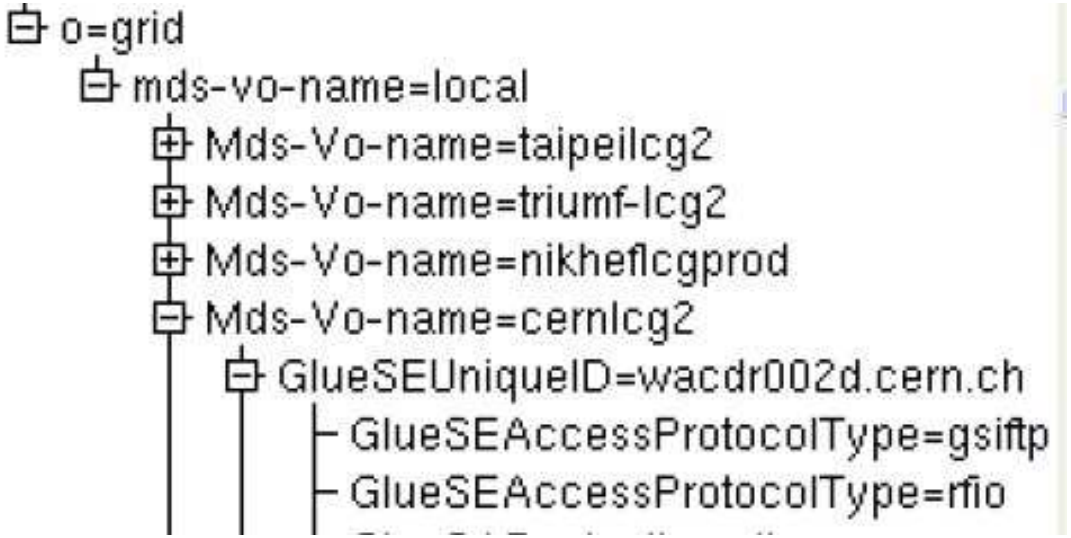
- **First level of MDS information**
- **No need to query a GRIS directly (firewall)**
- **LDAP Server**
  - ldapsearch -x -h <hostname> -p 2135 –b \ "mds-vo-name=local, o=grid“ <filtering>
- **Filtering: GLUE entries (e.g.: GlueSite…)**
  - ( "&" or "|" or "!" (filter1) [(filter2) ...] ) (…)
  - E.g.: (& (Name= Joe) (Add=NeSC))
- **Further filtering  using grep, …**

# The Information Service in WLCG/EGEE




FCR: Freedom of Choice for Resources

# Directory Information Tree (DIT)




- DN is formed from a sequence of attribute/value pairs
- A BDII can be interrogated using the base name mds-vo-name=local,o=grid

# Service Discovery (SD)

- **Access service details**
- **Questions like:**

  *I am at Z, in the X VO. Where is a MyProxy server?*
- ***Supports:***
  - *MDS, R-GMA, MDS4, and XML file*
- ***Usage e.g.:***
  - glite-sd-query -t myproxy -s CERN-PROD

# Proxy Delegation

- **Automatically**
- **explicitly creating a delegated credential**
- **glite-wms-job-delegate-proxy -d <delegID>**
  - Rqandom with –a
  - glite-wms-job-submit -d mydelegID test.jdl

# Status and Cancelling

- **glite-wms-job-status [-v verbosity] [-i joblist] jobIDs**
  - --from /--to [MM:DD:]hh:mm, -s or -e <status>

  E.g.:

  glite-wms-job-status --all -e CLEARED --to 17:35
- **glite-wms-job-cancel [-i joblist] jobID**
- **glite-wms-job-output [-dir outdir] [-i joblist] jobIDs**
- **glite-wms-job-logging-info [-v verbosity] [-i joblist] jobIDs**

# Advanced Sandboxes

- Files in  GridFTP server
- InputSandbox = {"gsiftp://lxb0707.cern.ch/cms/doe/data/fileA", …};
- OutputSandbox = {"fileA", "fileB“, "file:///tmp/C"};
- OutputSandboxBaseDestURI = \ "gsiftp://lxb0707.cern.ch/cms/doe/";
- Note: File /tmp/C will be copied using glite-wms-job-output command

# Real Time Output

- **PerusalFileEnable to true**
- **PerusalTimeInterval**
- **WMS or GridFTP (PerusalFilesDestURI)**
- **E.g.:**
  - glite-wms-job-perusal --set –f \ stdout.log -f stderr.log -f testfile.txt <ID>
  - glite-wms-job-perusal --get -f testfile.txt <ID>
- **glite-brokerinfo [-v] [-f filename] function [par] [par] ...**

# Advanced Job Types

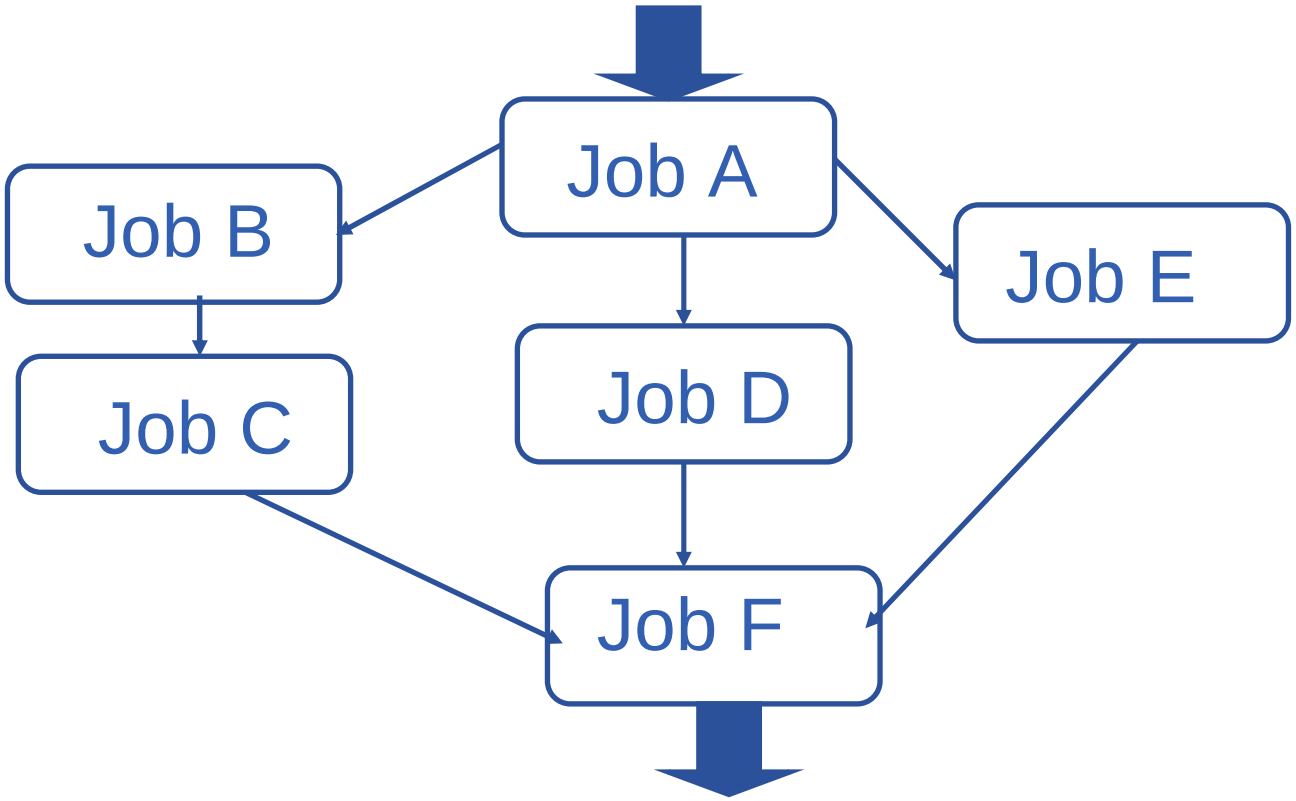
- Using compound jobs in one shot submission
  - possibly very large, up to thousands
  - Submission time reduction and single AA process
- Single call to WMProxy server
- Availability of different Job Ids
- Job Collection
  - glite-wms-job-submit -a --collection <dir>
  - Returns JobID for the collection
- Parametric jobs

  Parameters = 100; ParameterStart = 1;

  ParameterStep = 1;

# Advanced Job Types (Cont.)

- DAG jobs
  - sets of jobs linked by relative dependencies
    - input, output, or execution



# Advanced Job Types (Cont.)

- **Interactive Jobs using X window**

  JobType = "Interactive" ;

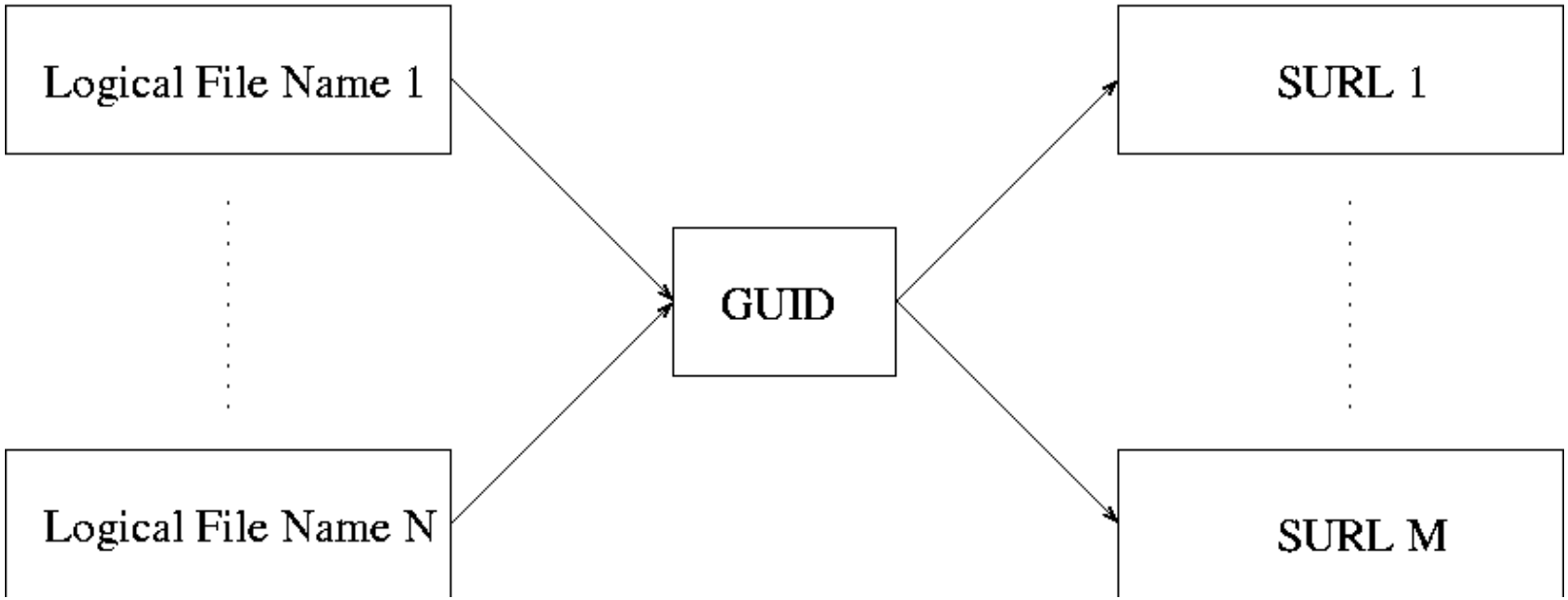
  Executable = "interactive.sh" ;

  InputSandbox = {"interactive.sh"} ;
- **Message Passing Interface (MPI) Jobs**
- **Other Features**
  - Using several WMS
  - Separate LB server

# Grid Files

- **Transparent file location**
  - Grid Unique IDentifier (GUID) : guid:<MAC+timestamp>
  - Logical File Name (LFN) : lfn:<any_string>
- **Physical Address**
  - S (Storage) URL: info about location (replica):
    - <sfn | srm>://<SE_hostname>/<string>
  - T(Transport) URL info how to access:
    - <protocol>://<SE host>:<port>/<path>
- **Using different names (File Catalog)**
  - hierarchical directory like structure: L(LCG)FC

# Grid files




- The mappings between LFNs, GUIDs and SURLs are kept in a service called a ***File Catalogue***
- Officially supported (in WLCG/EGEE) catalogues
  - ***LCG File Catalogue (LFC)***

# FILE CATALOGUE IN GLITE

- **To be considerd as Grid file:**
  - In SE and in File catalogue
- **The Catallogue publishes service URL in the IS**
  - Local File Catalogue:
    - only replicas stored at a given group of site
  - Global File Catalogue:
    - information about all files in the Grid
- **LFC commands**
  - lfc-mkdir, lfc-rm, lfc-chmod, lfc-ln, …
- **Directory structure: /grid/<VO>/<path>**
- **High level tools (lcg-utils): Consistancy**
- **ACL used by VOMS**

- Detailed information: http://glite.web.cern.ch/glite/documentation/

The END