Sun Grid Engine, a new scheduler for EGEE

G. Borges, M. David, J. Gomes, J. Lopez, P. Rey, A. Simon, C. Fernandez, D. Kant, K. M. Sephton

IBERGRID Conference
Santiago de Compostela, Spain
14, 15, 16 May 2007

L I P

Information Society

Outline

- **EGEE back to basics**
  - The EGEE project
    - The Infrastructure and the gLite Middleware
- **EGEE Local Resource Management Systems (LRMS)**
  - LSF, Torque/Maui, Condor and Sun Grid Engine
- **Sun Grid Engine gLite integration (for the lcg-CE)**
  - JobManager
  - Accounting Information
  - Information plug-in
  - YAIM Integration
- **Conclusions and Future Work**

EGEE Project

- **Enabling Grids for E-Science fundamental goal**
  - Deployment of a **Grid Infrastructure for all fields of science**
- **EGEE infrastructure**
  - Resources are “glued” together by a set of agreed services provided and supported by the EGEE comunity
  - EGEE proposes gLite as the appropriate middleware to support the necessary grid services for multi-science aplications
- **EGEE Services are divided in two different sets:**
  - Core Services: Only installed in some RCs but used by all users
    - Resource Broker, Top-BDII, File Catalogues, VOMS servers,...
  - Local Services: Deployed and Maintained by each participating site
    - **Computing Element**, Storage Element, MonBox, User Interface,...

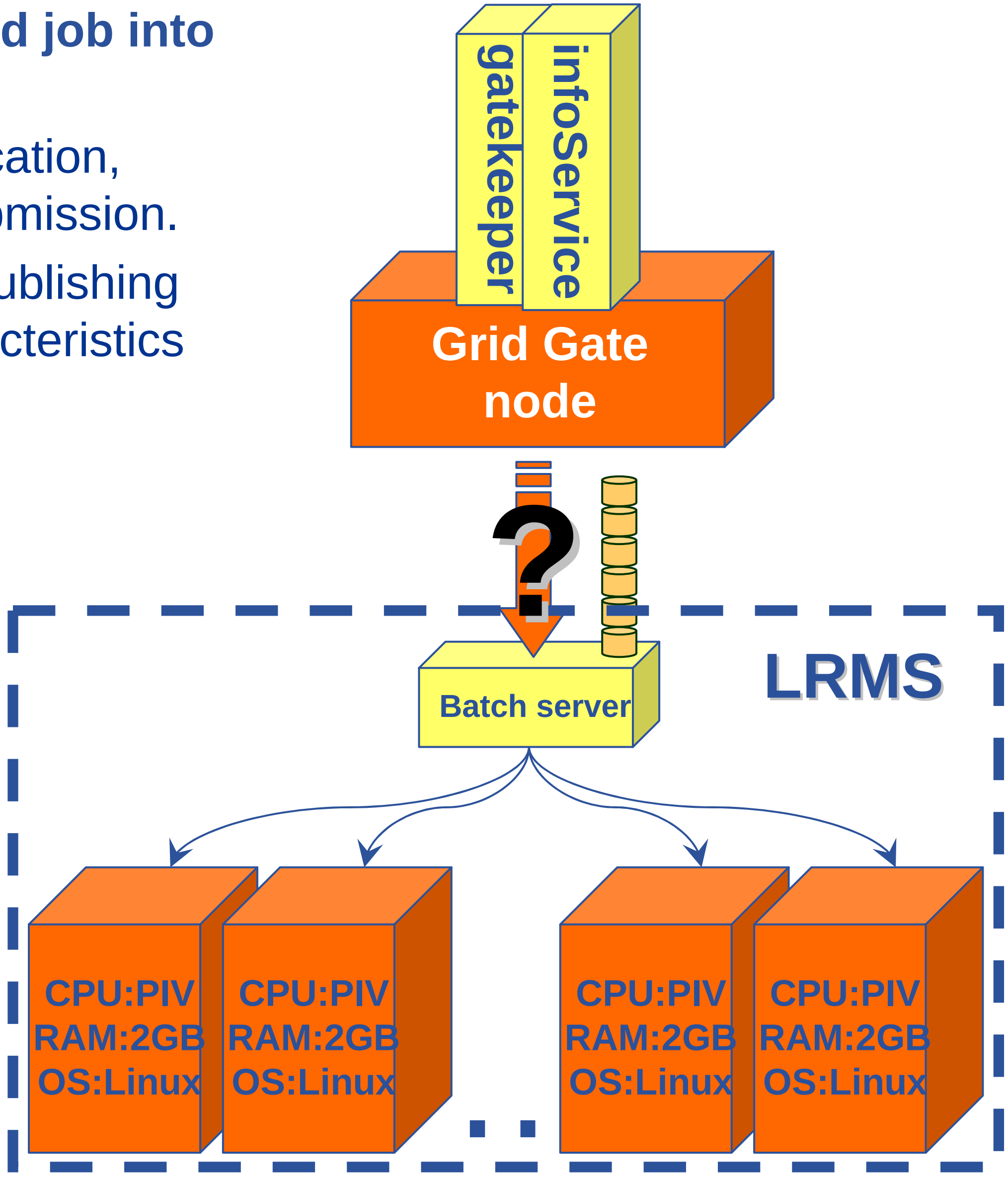
The Computing Element

- **The CE may be used by a generic client**
  - An end-user which interacts directly with it
  - The Workload Manager (RB) which submits a given job to it after going through by all the matchmaking process
- **It is THE SERVICE representing the computing resources**
  - Authentication and Autorization
  - Has to interact with the **Local Resource Management System**
    - **Job management (Job submission, Job control, Job canceling,...)**
    - **Provide information describing itself**
      - *This information is published in the Information Service*
      - *Used by the match making engine which matches available resources to queued jobs.*

Computing Element and the LRMS

- The CE is the **entry point** from a Grid job into the LRMS
  - **Gatekeeper Service** for Authentication, Authorization and Globus Job Submission.
  - **GRIS Service (InfoService)** for publishing Local Resource Usage and Characteristics

- gLite must implement proper tools (Virtual Layers) to
  - **Use LRMS specific cmds for**
    - Job Management (translate RSL requests; feed the L&B Service)
    - Query Resource Usage (feed the CE GRIS Service)
  - Process the **Accounting Information** generated by the LRMS and feed it to the central Accounting Registry



LRMS

- **The Local Resource Management System (LRMS) is the Cluster Component which**
  - Manages the execution of Users Applications
  - Allows to optimize the Cluster Resource Usage
  - Enables to fullfil a broadrage of Usage Policies
  - Easies the Cluster Administration Tasks

- **Each EGEE Cluster Admin should be allowed to choose the LRMS he thinks its best for their needs**
  - Most of the times, EGEE clusters are shared with Local Farms
  - However, only Torque/Maui and LSF are fully supported in EGEE

- **gLite should be able to cope with a much wider set of LRMS**
  - Easies the integration of clusters already in operation
  - Better inter-operability
  - The wider the gLite offer, more appealing it becomes…

Basic LRMS Comparison

LRMS	Pros	Cons
LSF	- Flexible Job Scheduling Policies - Advance Resource Management ○ Checkpointing & Job Migration, Load Balacing - Good Graphical Interfaces to monitor Cluster functionalities - Integrable with Grids	- Expensive comercial product - Not suitable for small computing clusters
Torque/ Maui	- Good integration of parallel libraries ○ Able to start parallel jobs using LRMS services ○ Full control of parallel processes - Flexible Job Scheduling Policies ○ Fair Share Policies, Backfilling, Resource Reservations	- Configurations done through the command line - A non user friendly GUI - Software development uncertain - Bad documentation
Condor	- CPU harvesting - Special ClassAds language - Dynamic check-pointing and migration - Mechanisms for Globus Interface - Coherent with gLite MD	- Not optimal to parallel aplications - Check-pointing only works for batch jobs - Complex configuration

Sun Grid Engine

- **SGE, an open source job management system supported by Sun**
  - Queues are located in server nodes and have attributes which caracterize the properties of the different servers
    - A user may request at submission time certain **execution features**
      - *Memory, execution speed, available software licences, etc*
    - Submitted jobs wait in a holding area where its requirements/priorities are determined
      - *It only runs if there are queues (servers) matching the job requests*

- **Some Important Features**
  - Supports **Check-pointing and Migration...**
    - Although some additional programming could be needed
  - **Tight integration of parallel libraries**
    - Supported through a SGE specific version of “rsh”, called “qrsh”
  - **Flexible Scheduling Polices**
  - Implements **Calendars**
    - Fluctuating Resources
  - **Intuitive Graphic Interface**
    - Used by users to manage jobs and by admins to configure and monitor their cluster
  - **Good Documentation**
  - Still Work in Progress
    - Observed flaws maybe addressed to dedicated teams and support is assureb by dedicated staff

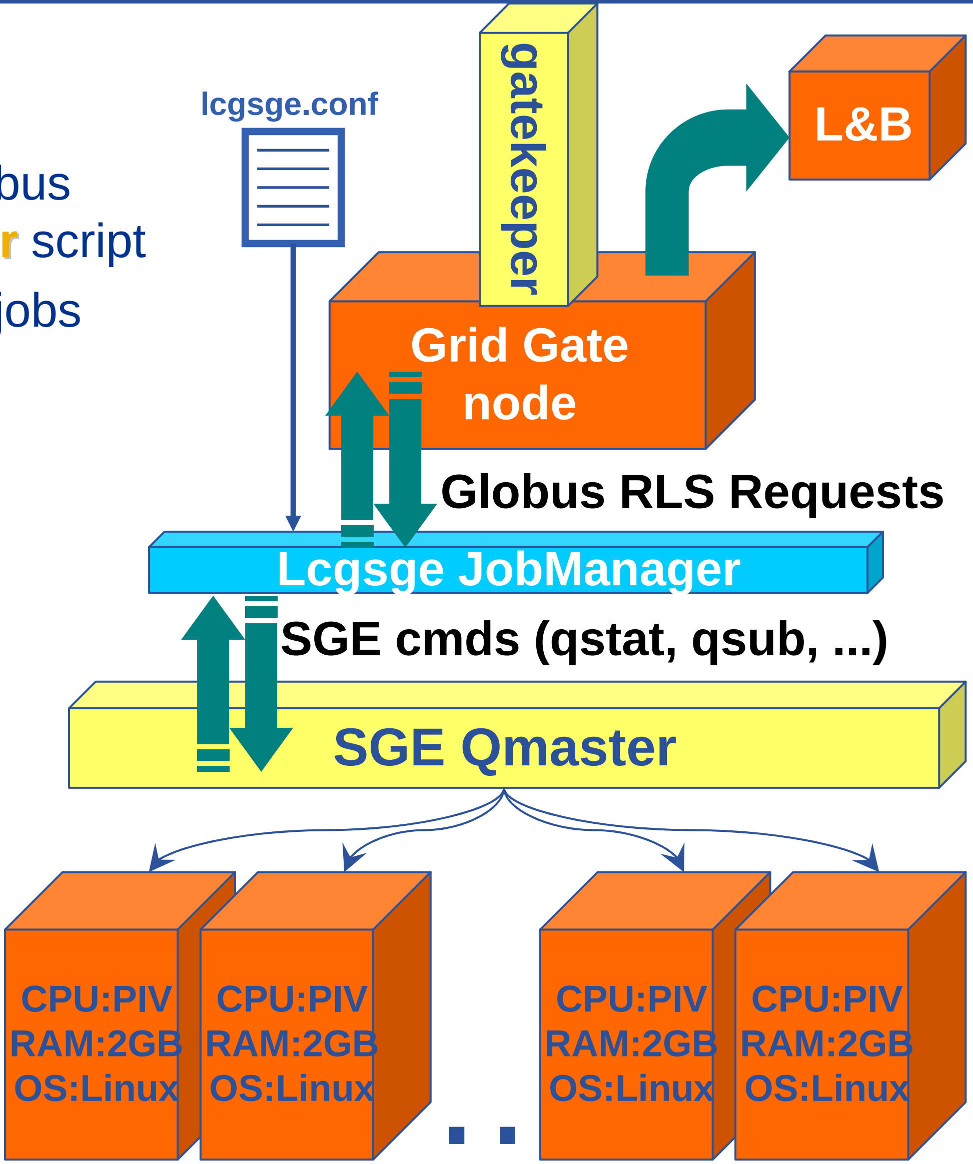
LCGSGE JobManager (1)

- **The JM is the core service of the Globus GRAM Service**
  - Submits jobs to SGE based on Globus requests and through a **jobwrapper** script
  - Intermediary to query the status of jobs and to cancel them

- **SGE command client tools (qstat, qsub, qdel) have to be available in the CE**
  - Even if the Qmaster machine is installed in another machine

- **Doesn’t require shared homes**
  - But home dirs must have the same path on the CE and WNs

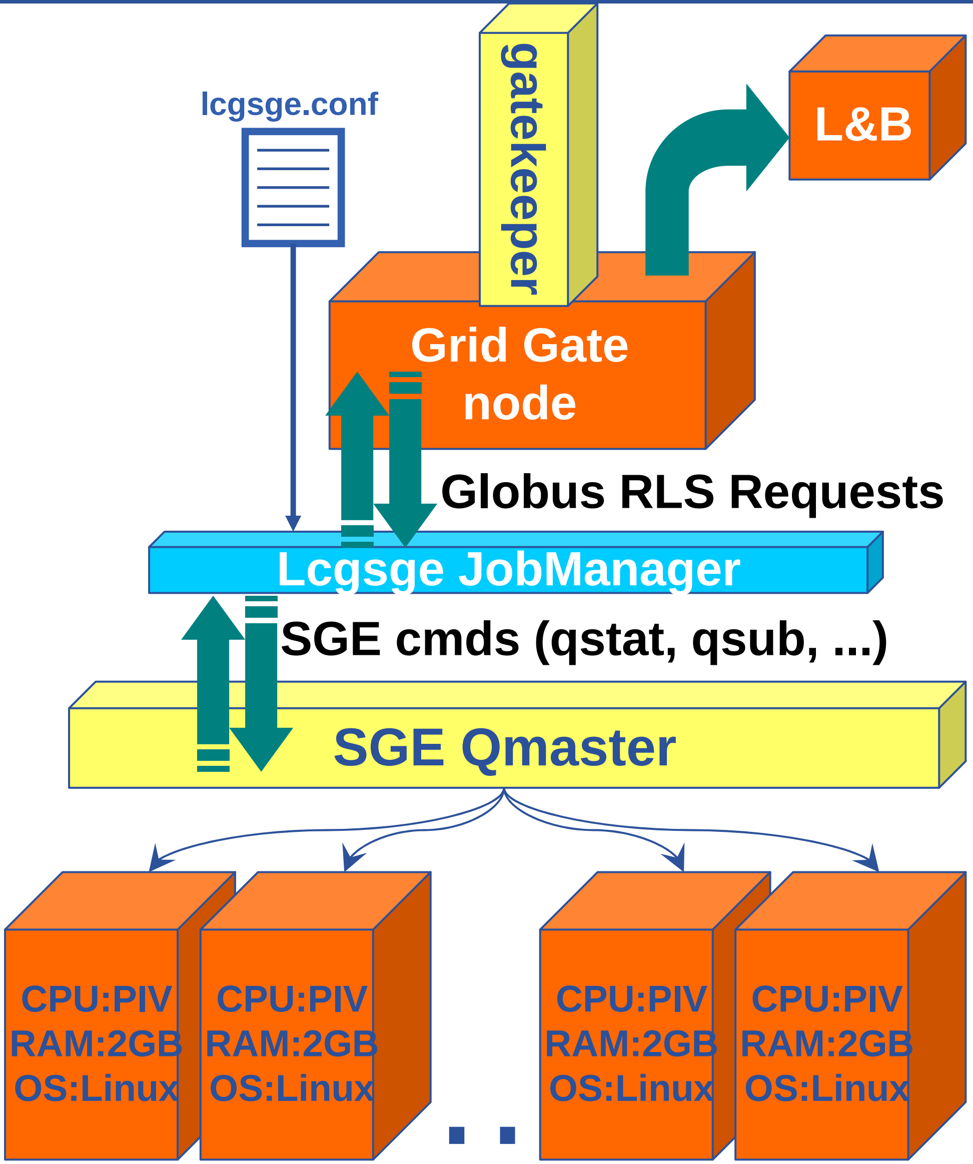
- **The SGE JM is based on the LCGPBS JM**
  - Requires XML::Simple.pm



LCGSGE JobManager (2)

SGE JM re-implements the following functions:

- **Submit**: Checks Globus RSL arguments returning a Globus error if the arguments are not valid or if there are no resources
- **Submit_to_batch_system**: Submits jobs to SGE, after building the **jobwrapper** script, by getting the necessary information from the RSL variables
- **Poll**: Links the present status of jobs running in SGE with the Globus appropriate message
- **Poll_batch_system**: Allows to know the status of running jobs parsing the **qstat** SGE output.
- **Cancel_in_batch_system**: Cancels jobs running in SGE using **qdel**

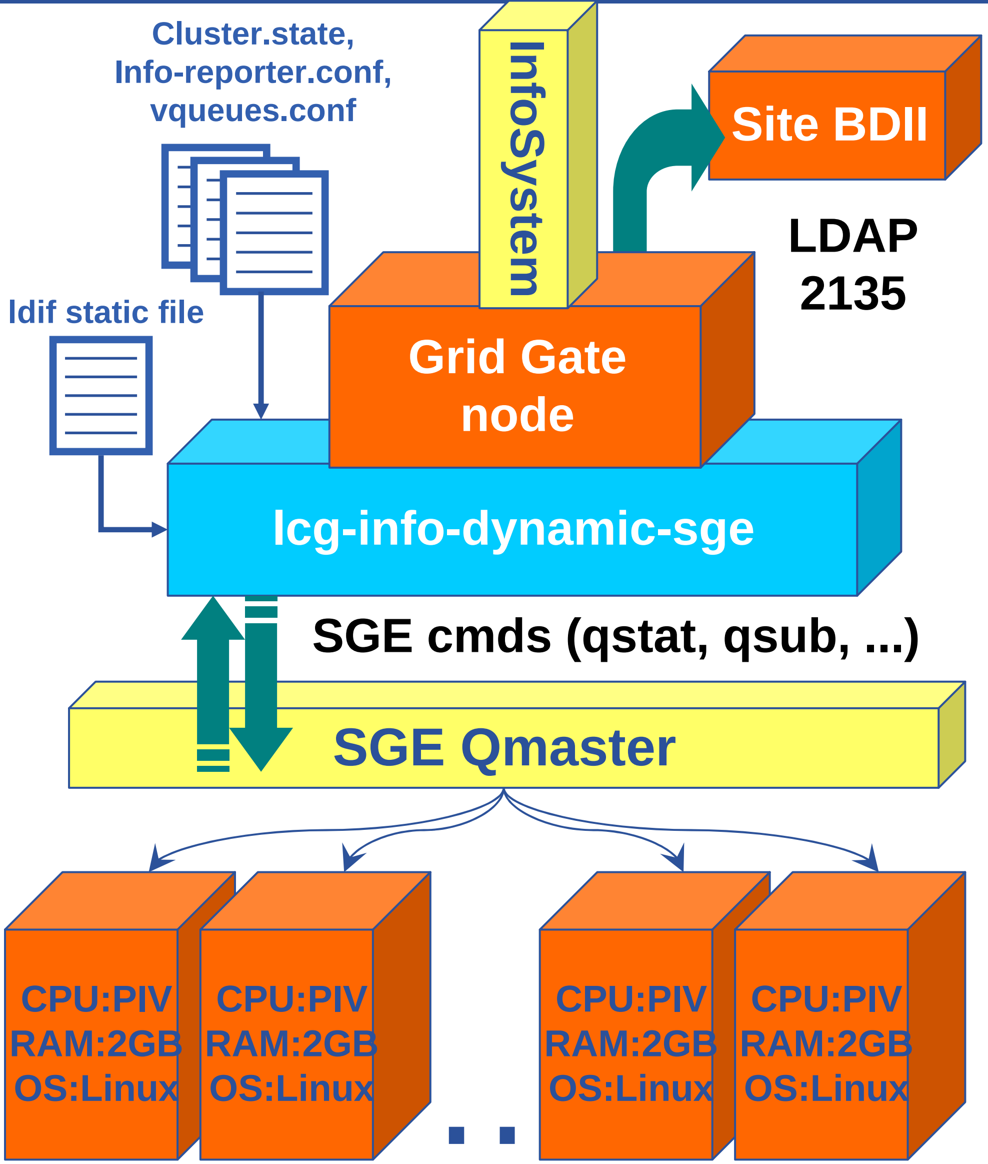


SGE Information Plugin (1)

- The solution implemented for SGE does not currently use the generic EGEE scripts
  - lcg-info-dynamic-sge”
    - A standalone Information plugin script that examines SGE queuing system state

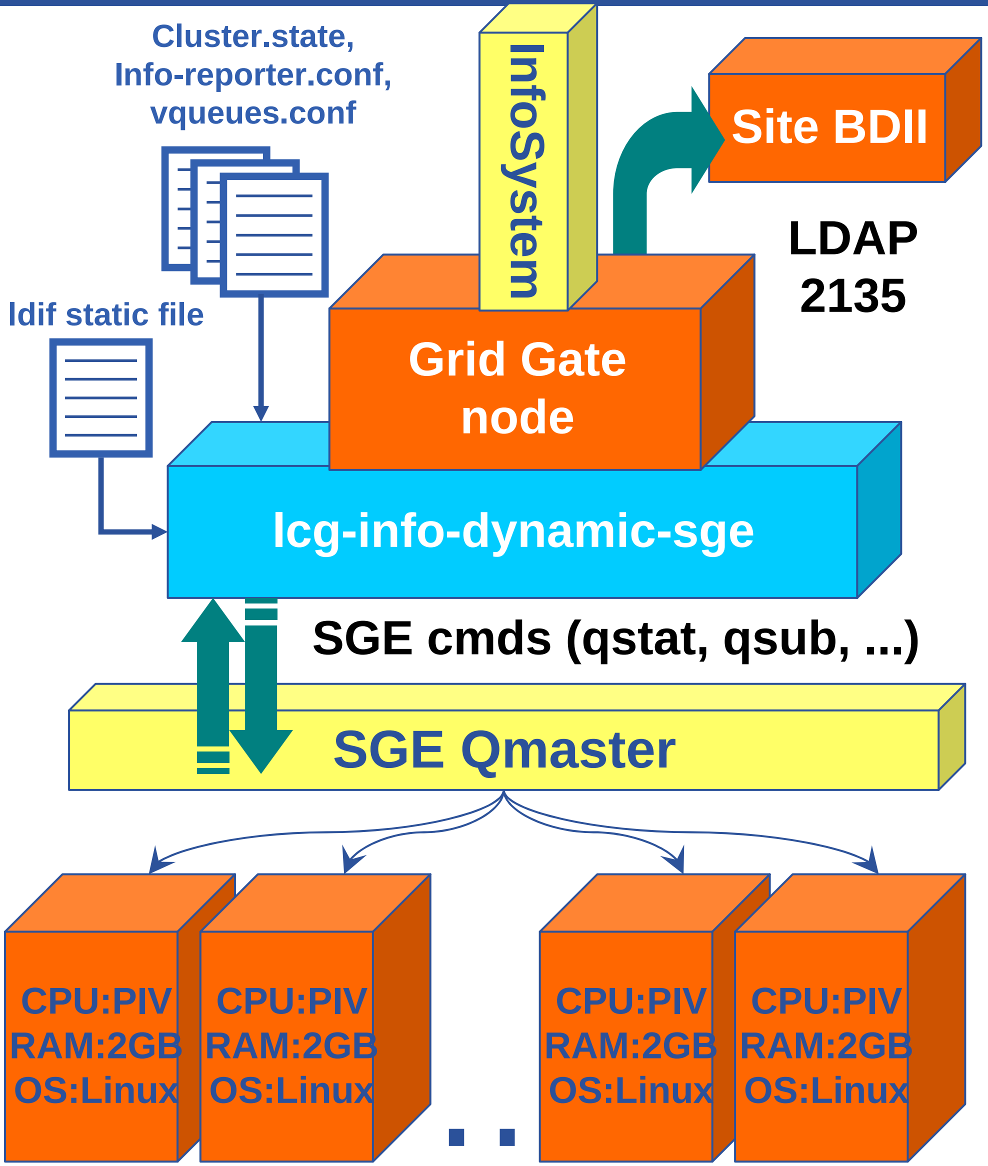
- Information expected to be reported is based on queues
  - SGE does not assign a job to a queue until execution time.
  - ``virtual queues" are used

- The info reporter reads…
  - A copy of a static ldif file with details of all ``virtual queues“
  - Config files specifying how virtual queues map into a list of resource requirements



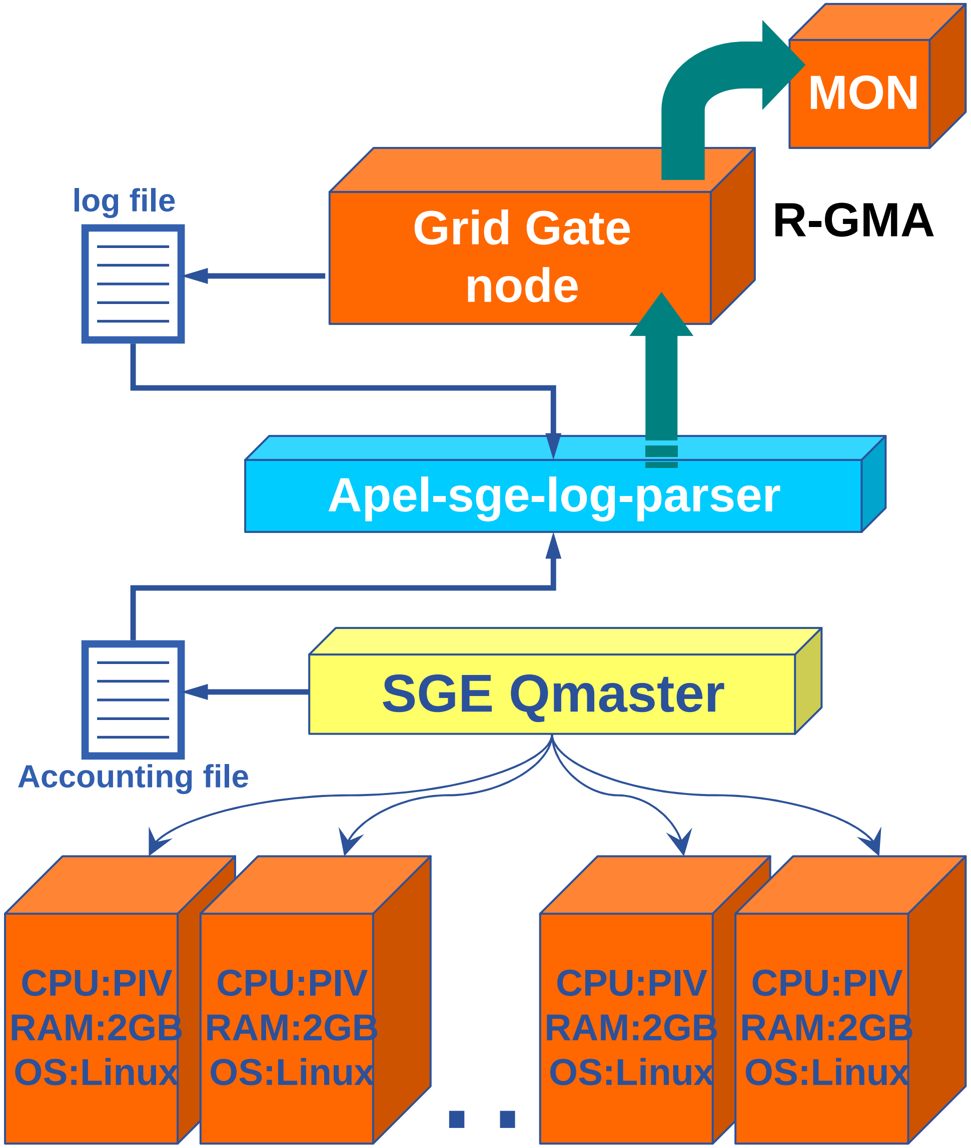
SGE Information Plugin (2)

- The dynamic information
  - single call to SGE's ``qstat"
- The system determines which virtual queues the job should be associated with
- Each virtual queue is considered to count up
  - Nb of job slots, Nb of pending/running jobs
  - Total amount of runtime left on all of the jobs assuming that they will run for their max duration
- The state of the batch queues can change quite fast …
  - Option to capture a copy of all information provider input data, which can be replayed to the information provider



SGE Accounting Plugin

- APEL SGE plug-in is a log processing application
  - Used to produce CPU job accounting records
  - Interprets gatekeeper & batch system logs
- Requires the JM to add ``gridinfo" records in the log file
  - Standard Globus JMs do not log them but LCG JMs do it
- apel-sge-log-parser parses the SGE accounting log file
  - This information, together with the gridinfo mappings from the JobManager are joined together to form accounting records
  - Published using R-GMA to an accounting database.



SGE YAIM Integration (1)

- **YAIM (Yet Another Installation Method)**
  - Separates the instalation process from the configuration one
  - Based on a library of bash functions called by a configuration script
    - Functions needed by each node are defined in node-info.def file
    - The grid site topology is totally encapsulated on the site-info.def file

- **Development of two integration rpms**
  - lcgCE-yaimtosge-0.0.0-2.i386.rpm
  - gliteWN-yaimtosge-0.0.0-2.i386.rpm
  - Requirements
    - SGE installed (we presently made SGE rpms to install it)
    - lcg-CE and glite-WN
    - glite-yaim (>=3.0.0-34), perl-XML-Simple (>= 2.14-2.2), openmotif (>=2.2.3-5) and xorg-x11-xauth (>= 6.8.2-1)

SGE YAIM Integration (2)

- **$SGE_ROOT software dir must be set to /usr/local/sge/pro**
  - May be changed by the site admin in a future release
- **The SGE Qmaster can only be installed in the CE**
  - May be installed in another machine in a future release
- **Three new variables must be set in the site-info.def**
  - **SGE_QMASTER, DEFAULT_DOMAIN, ADMIN_EMAIL**
- **The integration rpms do...**
  - Change the **node-info.def** file to include two new node types
    - CE_sge and WN_sge
    - Run the same functions as the CE and WN nodes, plus at the end
      - *Config_sge_server and Config_sge_client*

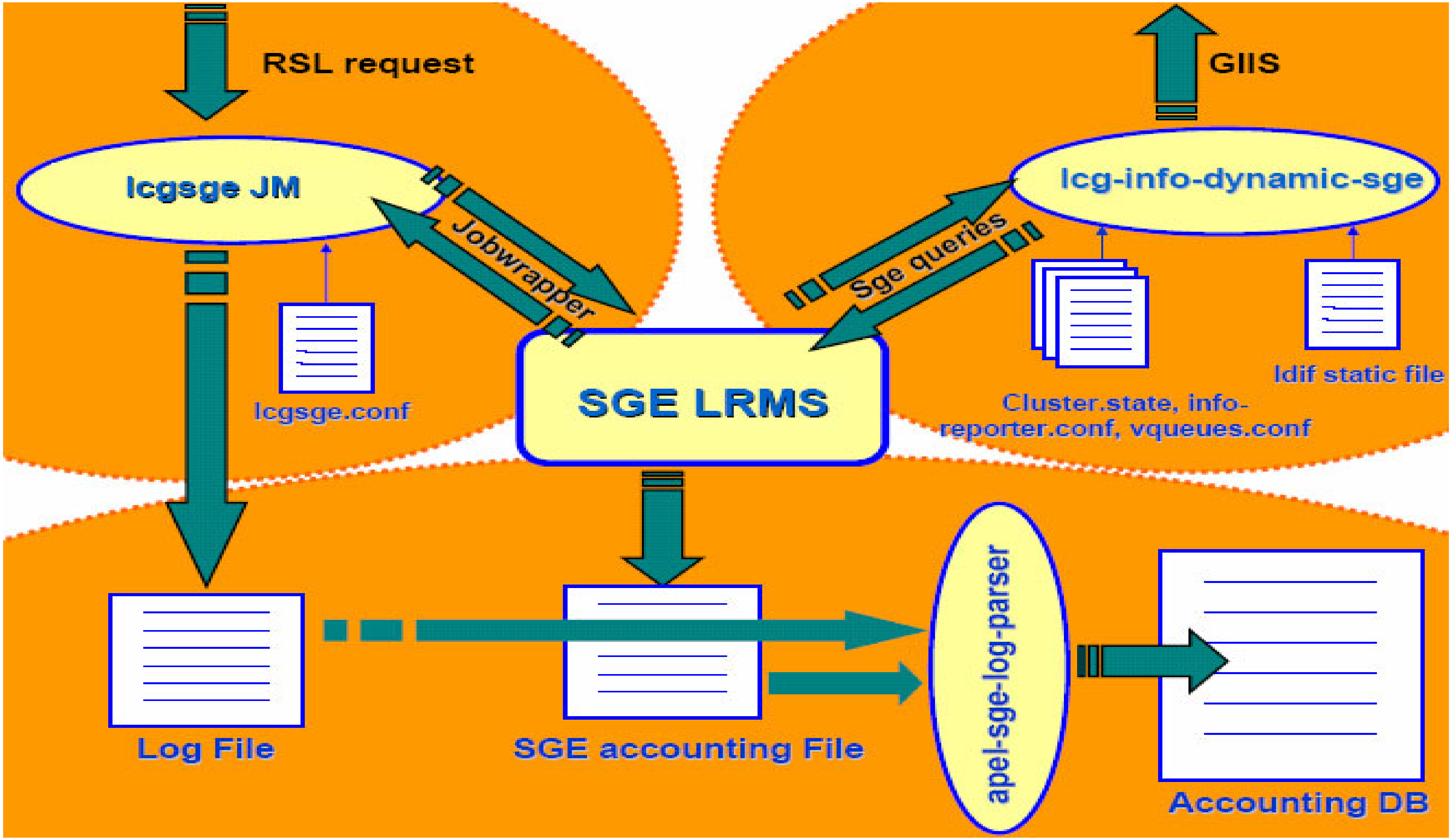
SGE YAIM Integration (3)

- **The Config_sge_server**
  - Uses an auxiliary perl script (**configure_sge_server.pm**)
    - Builds all the default SGE directory structure
    - Configures environment setting files, sets the global SGE configuration file, the SGE scheduler configuration file and SGE complex attributes
  - Defines one cluster queue for each VO
  - Deploys the **lcgsge JM** and builds its configuration files
  - Deploys **SGE Information plug-in** and builts its configuration files
  - Accounting is not properly integrated but will be soon...

- **The Config_sge_client**
  - Uses an auxiliary perl scrip (**configure_sge_client.pm**)
    - Builds all the default SGE directory structure in the client

SGE YAIM Integration (4)

- **/opt/glite/yaim/bin/yaim –c –s site-info.def –n CE_sge**



Future Work

- **SGE is working on a lcg-CE although additional work is required**
  - **YAIM SGE integration**
    - More flexible allowing site admins to dynamically set a broader range of options
    - Separate Qmaster from the CE
    - Fully integrate the SGE Accounting
  - **SGE Information Provider** needs to improve its flexibility and take into account overlapping cluster queues / virtual queues definitions

- **Started on integrating support for BLAHP, running on glite-CE**
  - Will be used within glite-CE and CREAM to interface with the LRMS
  - Expected to share the configuration files and concept of virtual queues with the information provider.
  - Other local middleware elements (GIIS, YAIM) basically remain unchanged for this glite-CE flavour.

- **Still missing**
  - GridICE sensors for SGE