



Enabling Grids for E-scienceE

gLite Application Developers Course

www.eu-egee.org





Enabling Grids for E-science

Application Development and Aspects of gLite 3.0

Mike Mineter

*Training Outreach and Education
University of Edinburgh, UK*

www.eu-egee.org



- **GILDA team – for GILDA and associated Wiki etc.**
- **Many discussions with EGEE colleagues, in particular**
 - Emidio Giorgio, University of Catania and INFN, Italy
 - Richard Hopkins & Guy Warner, TOE, Scotland
 - Gergely Sipos, SZTAKI, Hungary

What is a grid application?

Software that interacts with grid services to achieve requirements that are specific to a particular VO or user.

- **This talk maps the landscape – a high-level view of application development in Grids**
- **Practicals will explore specific features in that landscape**

- **Review of concepts: grids and grid applications**
- **Types of Grid applications**
- **Challenges to researchers who write applications**
- **More about gLite Services**
- **Overview of the rest of the course**

- Enabling a whole-system approach
- Collaborative research / engineering / public service ...

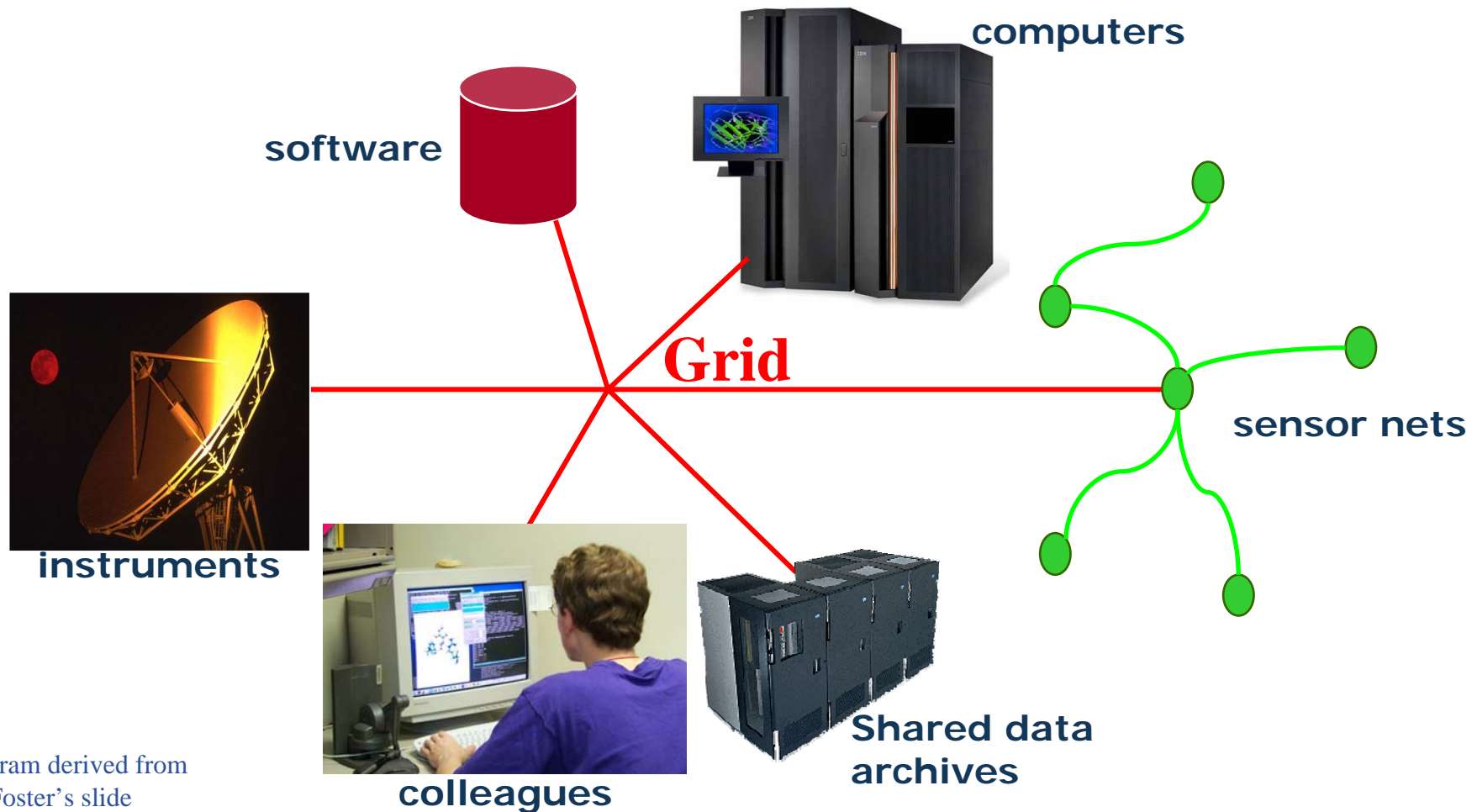
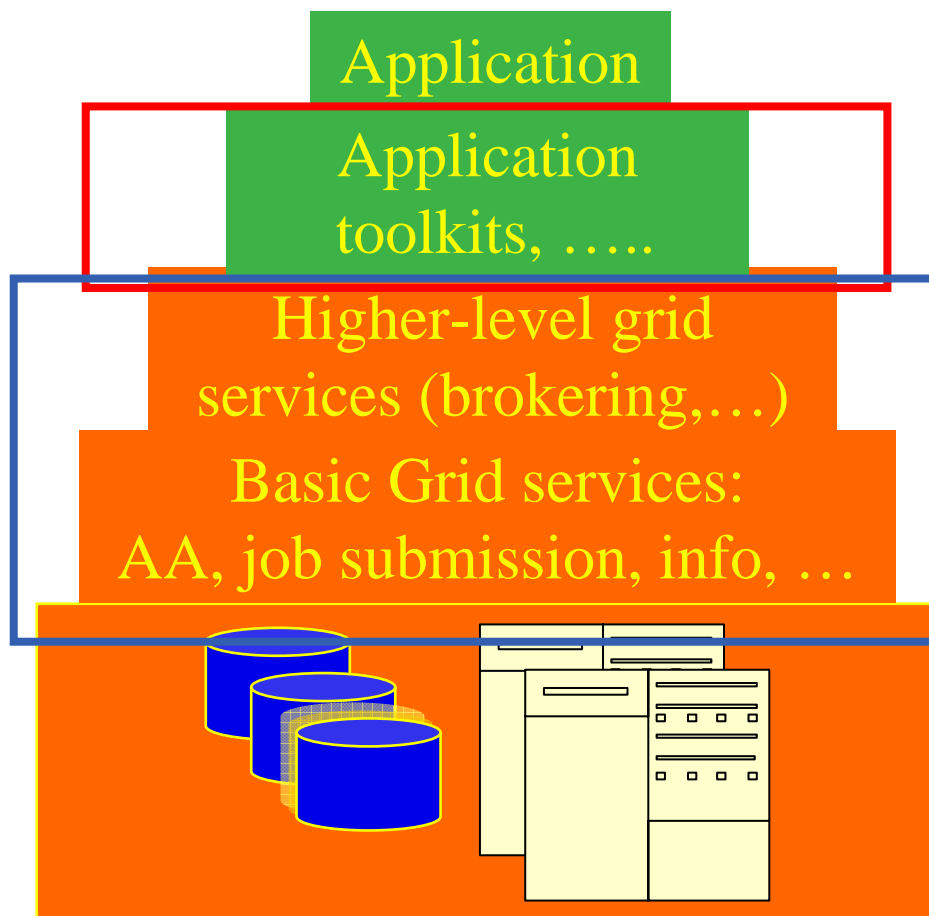


Diagram derived from Ian Foster's slide



Where computer science meets the application communities!
VO-specific developments built on higher-level tools and core services

Makes Grid services useable by non-specialists

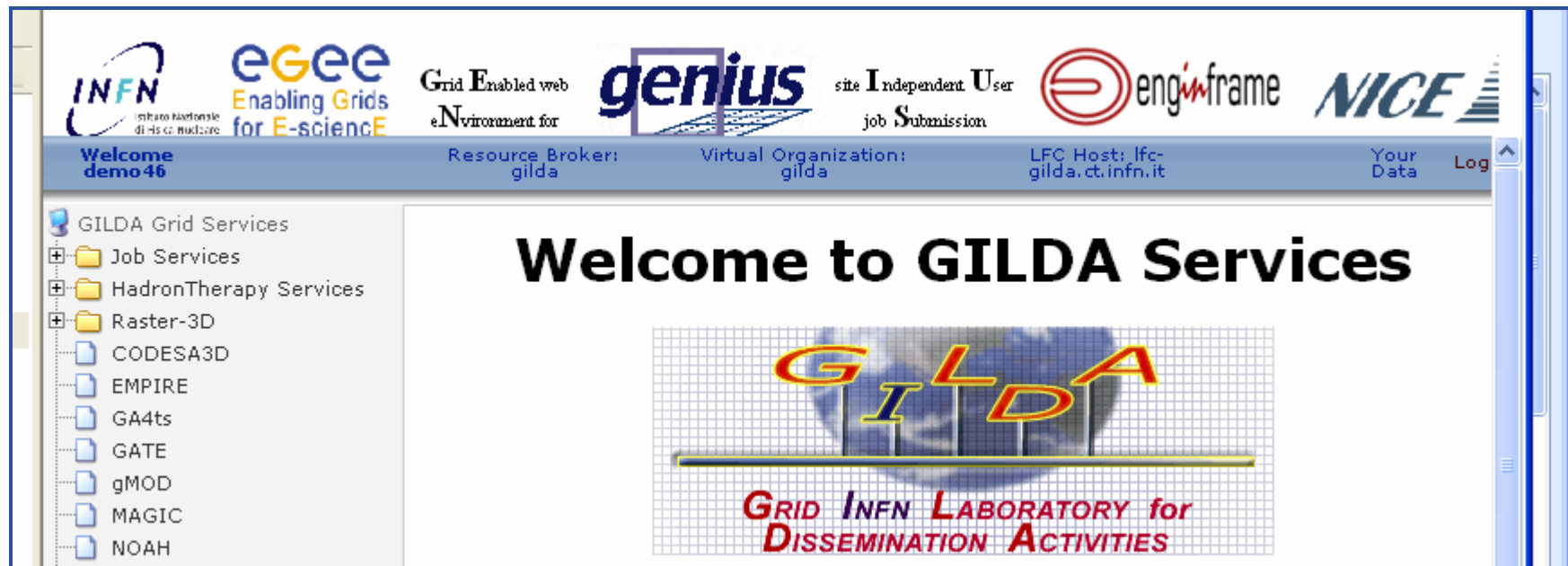
Grids provide the compute and data storage resources

Production grids provide these core services.

Types of grid applications

- 1. Simple jobs – submitted to WMS to run in batch mode**
- 2. Job invokes grid services**
 - To read & write files on SE
 - Monitoring
 - For outbound connectivity (interactive jobs)
 - To manage metadata
 - ...
- 3. Complex jobs**
 - An environment controls multiple jobs on users' behalf
 - High-level services
 - Portals with workflow
 - Software written for the VO (or by the user)
 - ...

- **From the UI**
 - Command Line Interfaces / Scripts
 - APIs
 - Higher level tools
- **From desktop Windows applications**
 - Use Grids without awareness of them!
 - But gLite not (yet) supporting Windows
- **From portals**
 - For recurring tasks: “core grid services” as well as application layer
 - Accessible from any browser
 - Tailored to applications
 - In EGEE: P-GRADE and GENIUS

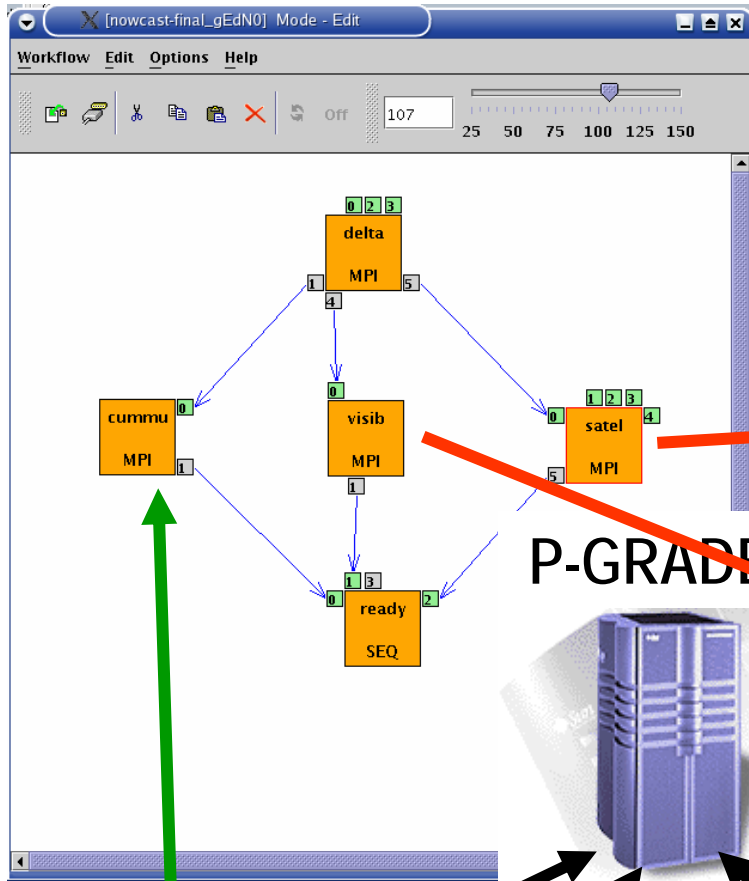


The screenshot shows the GENIUS web interface. At the top, there are logos for INFN, EGEE (Enabling Grids for E-science), genius, and NICE. Below the logos, there is a navigation bar with the text "Grid Enabled web eNvironment for" and "site Independent User job Submission". The main content area displays "Welcome to GILDA Services" in large bold letters. Below this, there is a graphic with the text "GILDA" in large letters, and "GRID INFN LABORATORY for DISSEMINATION ACTIVITIES" in smaller letters. On the left side, there is a sidebar menu with a tree view showing "GILDA Grid Services" and several sub-items: Job Services, HadronTherapy Services, Raster-3D, CODESA3D, EMPIRE, GA4ts, GATE, gMOD, MAGIC, and NOAH. At the top right of the main content area, there is a "Log" button and a "Your Data" link.

- **For many application communities**
 - Interface can be tailored for specific requirements
- **For demonstration purposes**
 - <https://glite-demo.ct.infn.it/>
 - Available for anyone to use
 - <https://glite-tutor.ct.infn.it/>
 - Fuller functionality for users who have stored long-lived proxy in MyProxy server

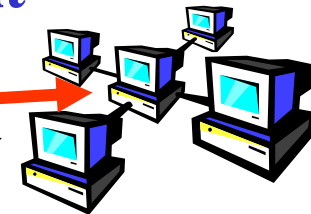


Multi-Grid P-GRADE Portal



Different jobs of a workflow can be executed in different grids

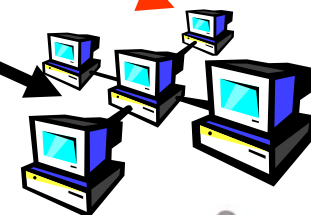
EGEE Grid
e.g. VOCE



P-GRADE-Portal

The portal can be connected to multiple grids

UK NGS



London

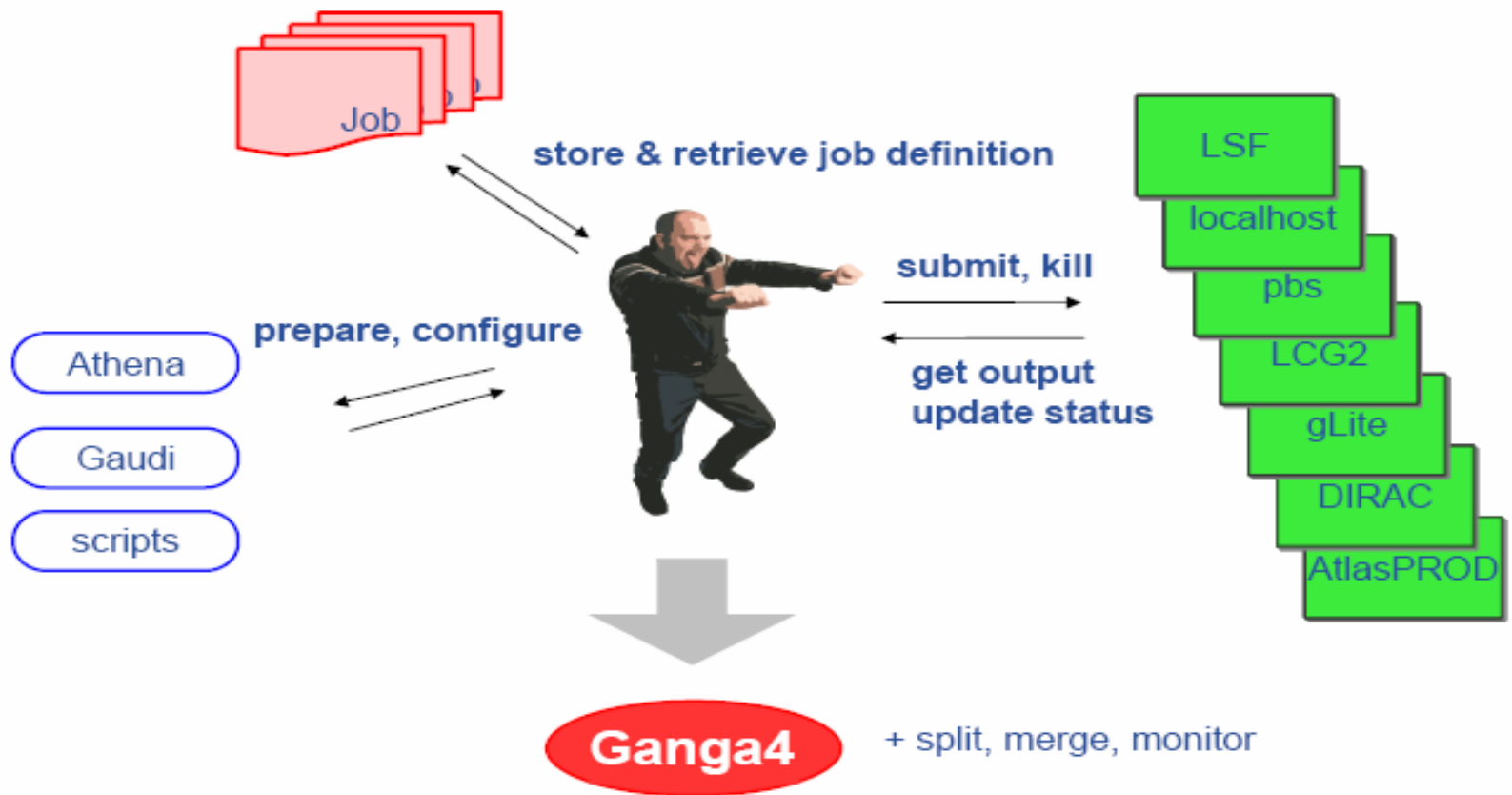


Rome



Athens





- **Ganga is a lightweight user tool**
ganga.web.cern.ch/
- **But also: Ganga is a developer framework**

- **What is being shared?**
 - resources of storage and/or compute cycles
 - software and/or data
- **Distinct groups of developers and of users?**
 - Some VOs have distinct groups of developers and users...
 - Biomedical applications used by clinicians,....
 - Some don't
 - Physics application developers who share data but write own analyses
 - Effect: need to
 - hide complexity from the 1st type of VOs.... E.g. AA
 - expose functionality to 2nd type of VOs

- **I need resources for my research**
 - I need richer functionality
 - MPI, parametric sweeps,...
 - Data and compute services together...

- **I provide an application for (y)our research**
 - How!?
 - Pre-install executables ?
 - Hosting environment?
 - Share data
 - Use it via portal?

- **We provide applications for (y)our research**
 - Also need:
 - Coordination of development
 - Standards
 - ...



Engineering challenges increasing

Challenges to researchers who write grid applications

- **Research software is often**

- Created for one user: the developer
- Familiarity makes it useable
- Short-term goals: Used until papers are written and then discarded

- **Grid applications are often used**

- by a VO
- Without support from developer
- In new contexts and workflows

- **Grid application developers are**

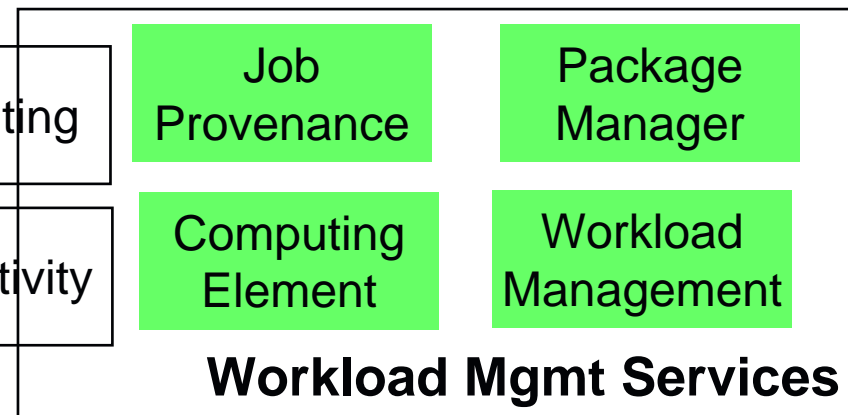
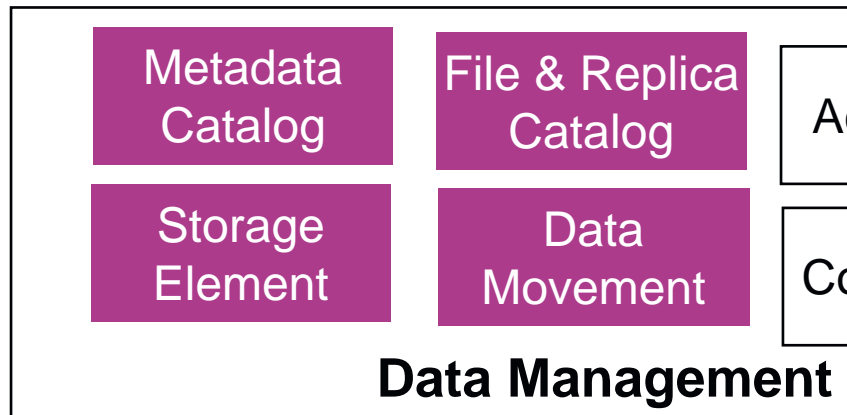
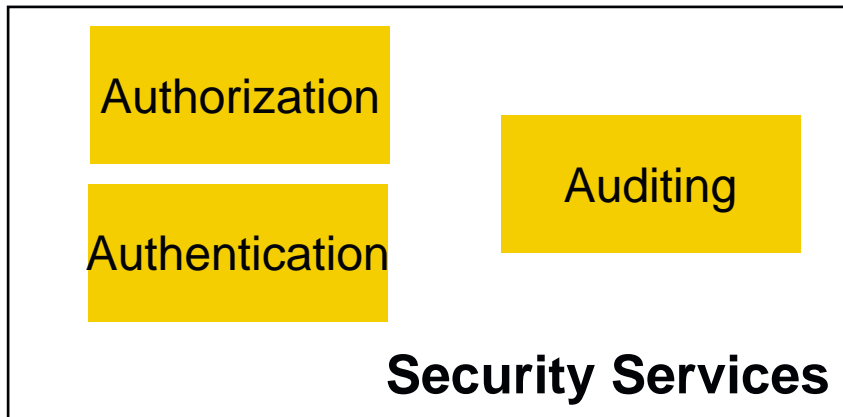
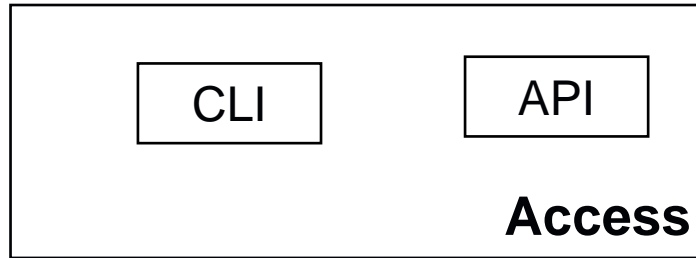
- In a research environment
- Yet their s/w must have:
 - Stability
 - Documentation
 - Useability
 - Extendability
- i.e. Production quality

Need expertise in:

- **software engineering**
- **application domain**
- **grid computing**

- **Team work!**
- **Engaged in world-wide initiatives – reuse, don't make your own! Cross disciplines for solutions.**
- **From research to production software: ~5 times the effort.**
 - “80% of the time for last 10% of the functionality & reliability”
- **Standardisation is key**
 - For re-use, for dynamic configuration of services,..
 - Both for middleware and domain specific (e.g. GEON)
- **Need to follow a deliberate development process**
 - Waterfall? Rapid prototyping?
 - Requirements engineering, design, implementation, validation, deployment
 - Engaged with the user community

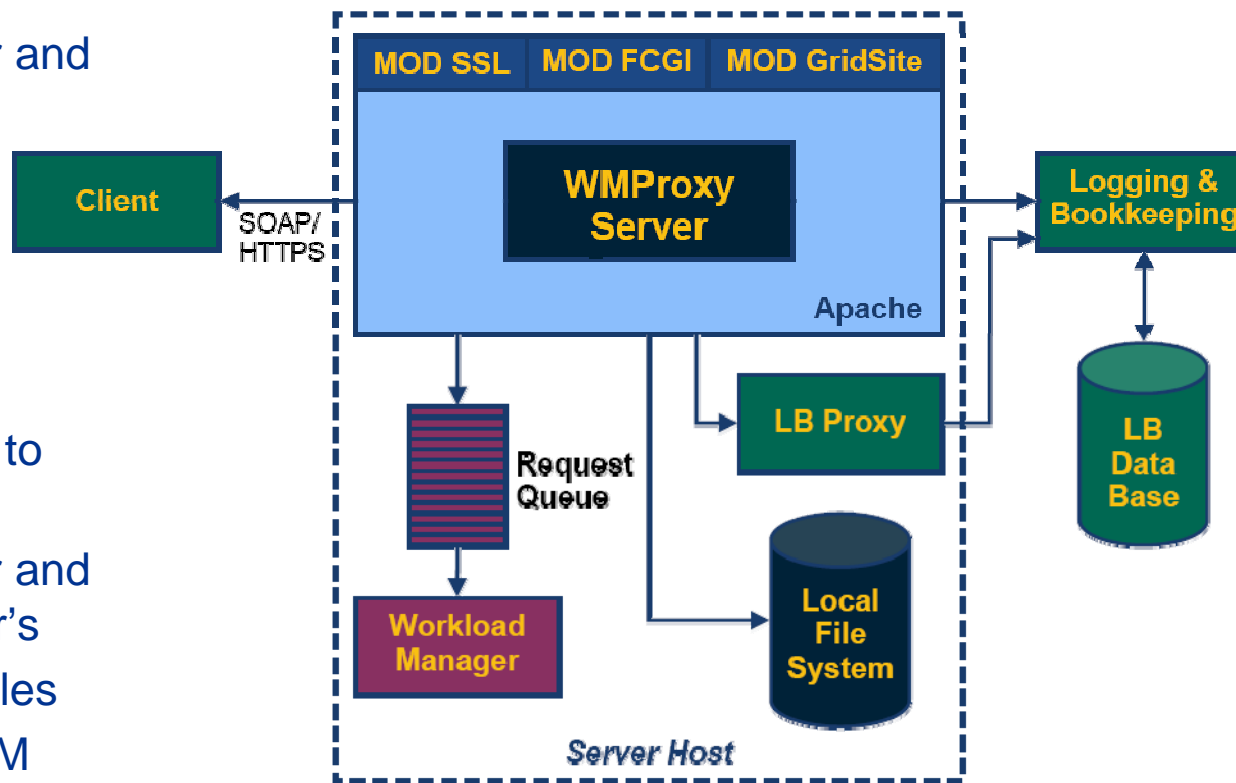
More about gLite services



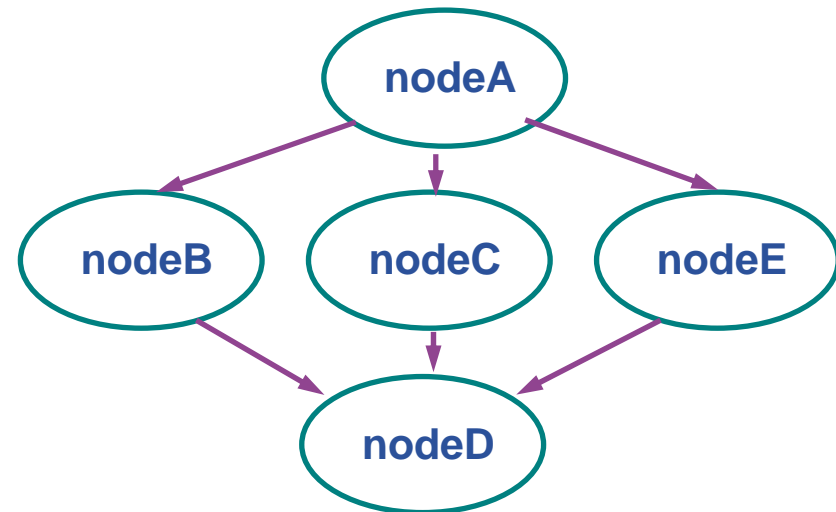
- **During today the focus is on:**
 - New functionality in gLite 3.0 Workload Management
 - Accessing data on SEs
 - Can have massive files, too big to copy
 - How to access these?
 - Management of metadata
 - May have many thousands of files
 - Need to access and re-use based on characteristics... more than by their logical file names.
 - Monitoring of applications
 - May be running many long jobs
 - What's happening?!

- **Helps the user accessing computing resources**
 - resource brokering
 - management of input and output
 - management of complex workflows
- **Support for MPI job even if the file system is not shared between CE and Worker Nodes (WN) – easy JDL extensions**
- **Web Service interface via WMPProxy**

- WMPProxy is a SOAP Web service providing access to the Workload Management System (WMS)
- Job characteristics specified via JDL
 - jobRegister
 - create id
 - map to local user and create job dir
 - register to L&B
 - return id to user
 - input files transfer
 - jobStart
 - register sub-jobs to L&B
 - map to local user and create sub-job dir's
 - unpack sub-job files
 - deliver jobs to WM



- **Direct Acyclic Graph (DAG)** is a set of jobs where the input, output, or execution of one or more jobs depends on one or more other jobs
- **A Collection** is a group of jobs with no dependencies
 - basically a collection of JDL's
- **A Parametric job** is a job having one or more attributes in the JDL that vary their values according to parameters
- **Using compound jobs** it is possible to have one shot submission of a (possibly very large, up to thousands) group of jobs
 - Submission time reduction
 - Single call to WMPProxy server
 - Single Authentication and Authorization process
 - Sharing of files between jobs
 - Availability of both a single Job Id to manage the group as a whole and an Id for each single job in the group



- **glite-wms-job-submit will supercede glite-job-submit (which is superceding edg-job-submit)**
- **Its support for compound jobs will simplify application software**
 - WMPProxy manages sub-jobs
 - Shared Input and Output “sandboxes”
- **MUST establish proxy delegation before this can be used!**

Overview of practicals

- **Induction courses introduce basic services.**
- **Today: building more complex applications**
- **Use of WMS and WMPProxy**
 - Parallelism using MPI
 - Within a CE – not across domains
 - Use of APIs
- **Use of GFAL functions to access files on SEs**
- **Use of AMGA: metadata service**
- **Use of R-GMA: monitoring service**