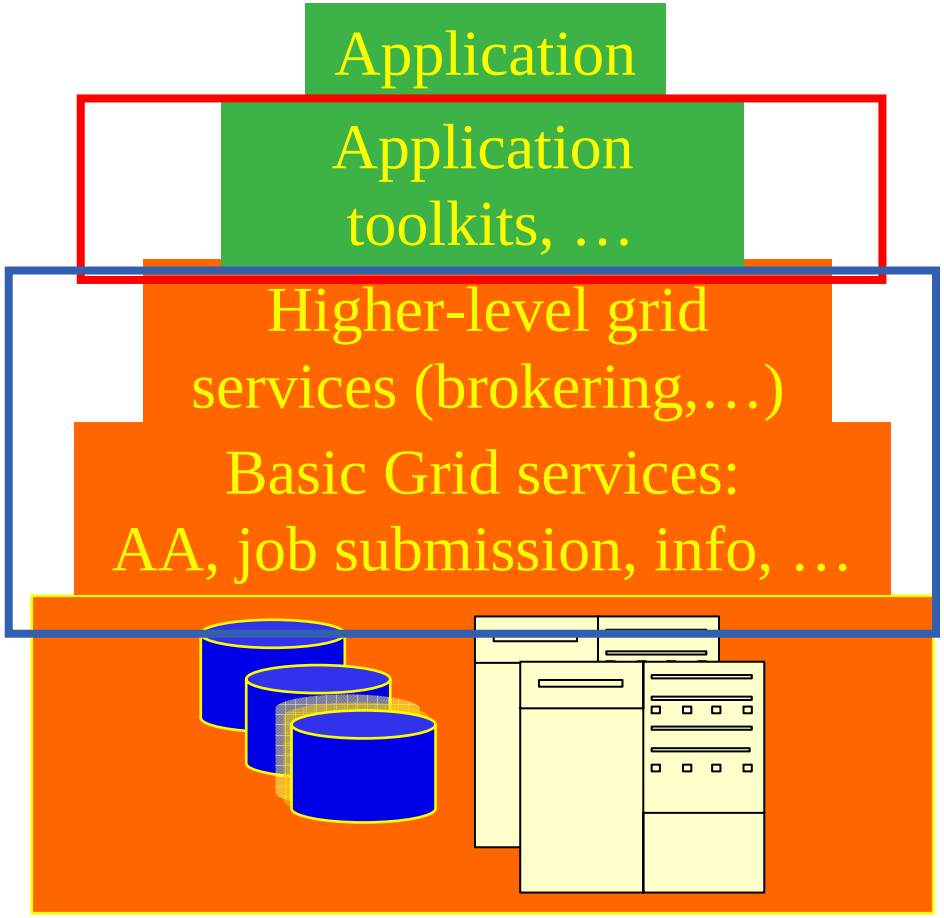
More about gLite services

…but first….

P-GRADE and GEMLCA

Empowering VO's



Where computer science meets the application communities!

VO-specific developments:

- Portals
- Virtual Research Environments
- Semantics, ontologies
- Workflow
- Registries of VO services

Production grids provide these services.

P-GRADE Portal and GEMLCA

Grid Execution Management for Legacy Code Applications

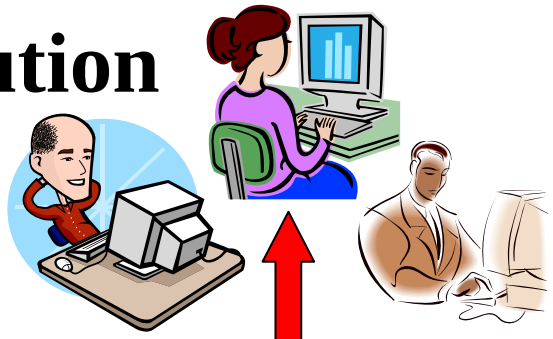
P-GRADE portal

MTA SZTAKI

Tamas Kiss, Gabor Terstyanszky
Centre for Parallel Computing
University of Westminster
kisst@wmin.ac.uk

Peter Kacsuk
SZTAKI Hungary
University of Westminster
kacsuk@sztaki.hu

The P-GRADE solution



User-friendly
Web interface

NGS National Grid Service
core production computational and data grid

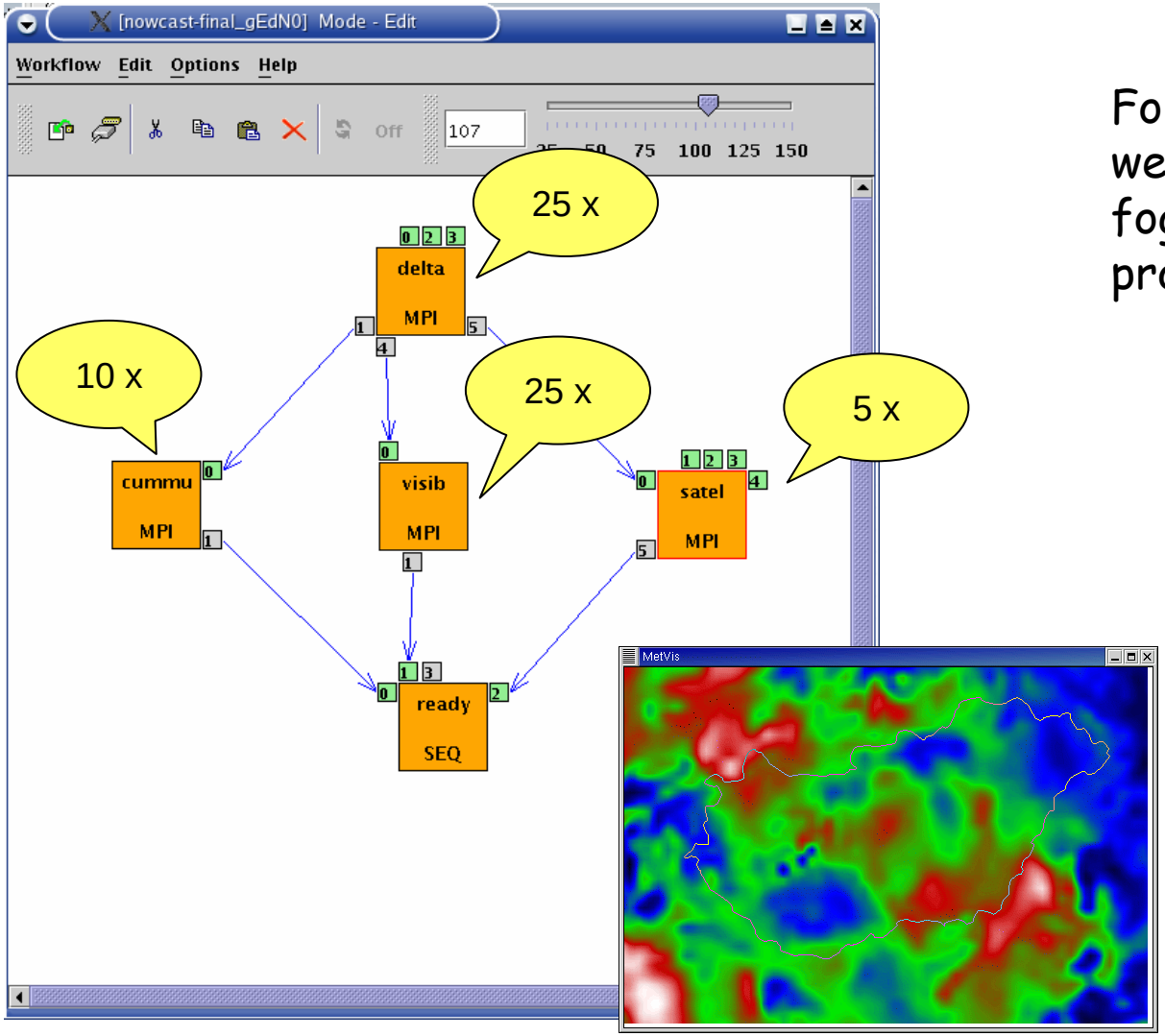
Leeds
Oxford
GT2
Manchester
Rutherford

P-GRADE portal

Portal server at UoW

Ultra-short range weather forecast

(Hungarian Meteorology Service)



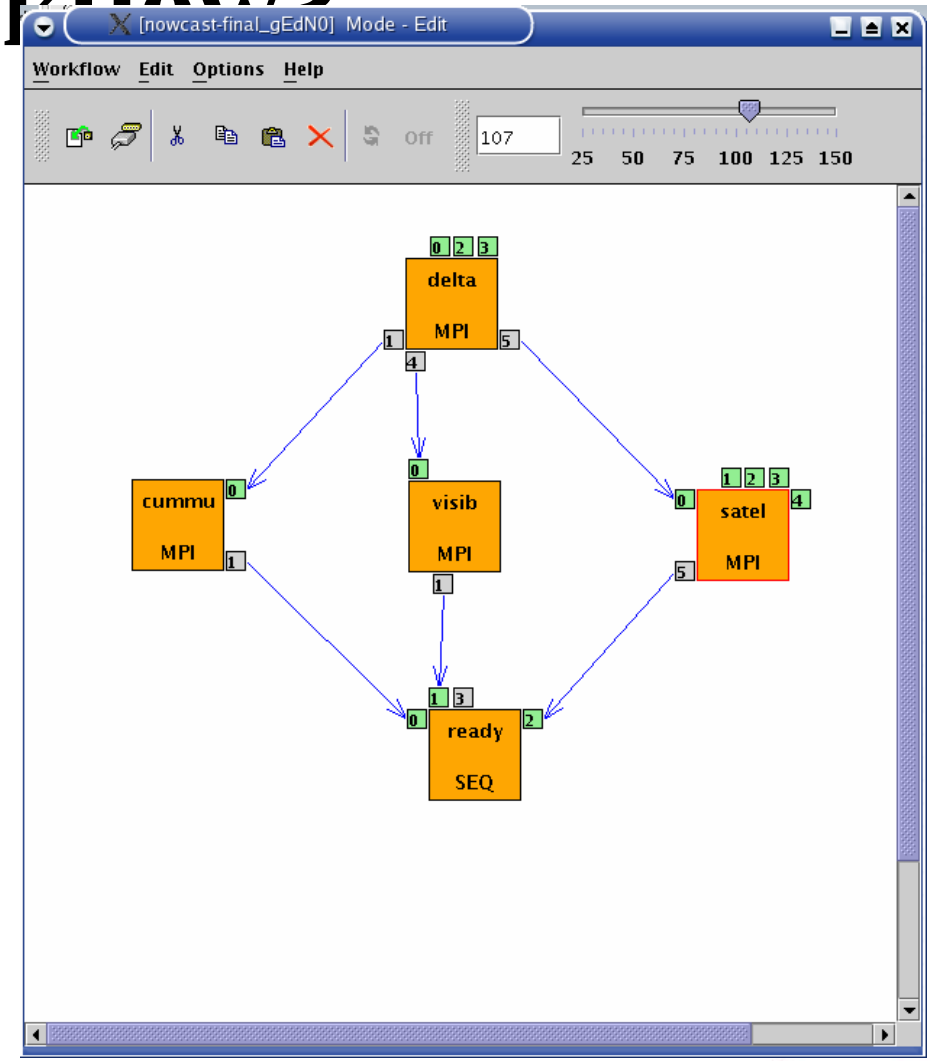
Forecasting dangerous weather situations (storms, fog, etc.), crucial task in the protection of life and property

Processed information: surface level measurements, high-altitude measurements, radar, satellite, lightning, results of previous computed models

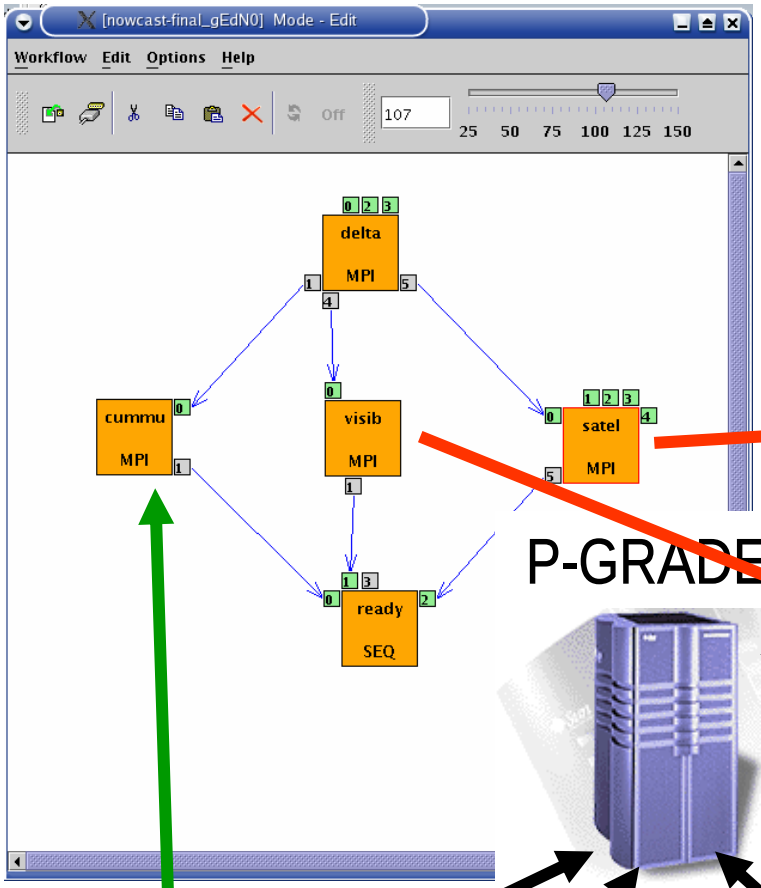
Requirements:
- Execution time < 10 min
- High resolution (1km)

What is a P-GRADE Portal workflow?

- **a directed acyclic graph (DAG) where**
  - Nodes represent jobs (executable batch programs)
  - Ports represent input/output files the jobs expect/ produce
  - Arcs represent file transfer between the jobs
- **semantics of the workflow:**
  - A job can be executed if all of its input files are available
    - **local input files**: on the portal server
    - **remote input files**: at Grid storage service providers



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

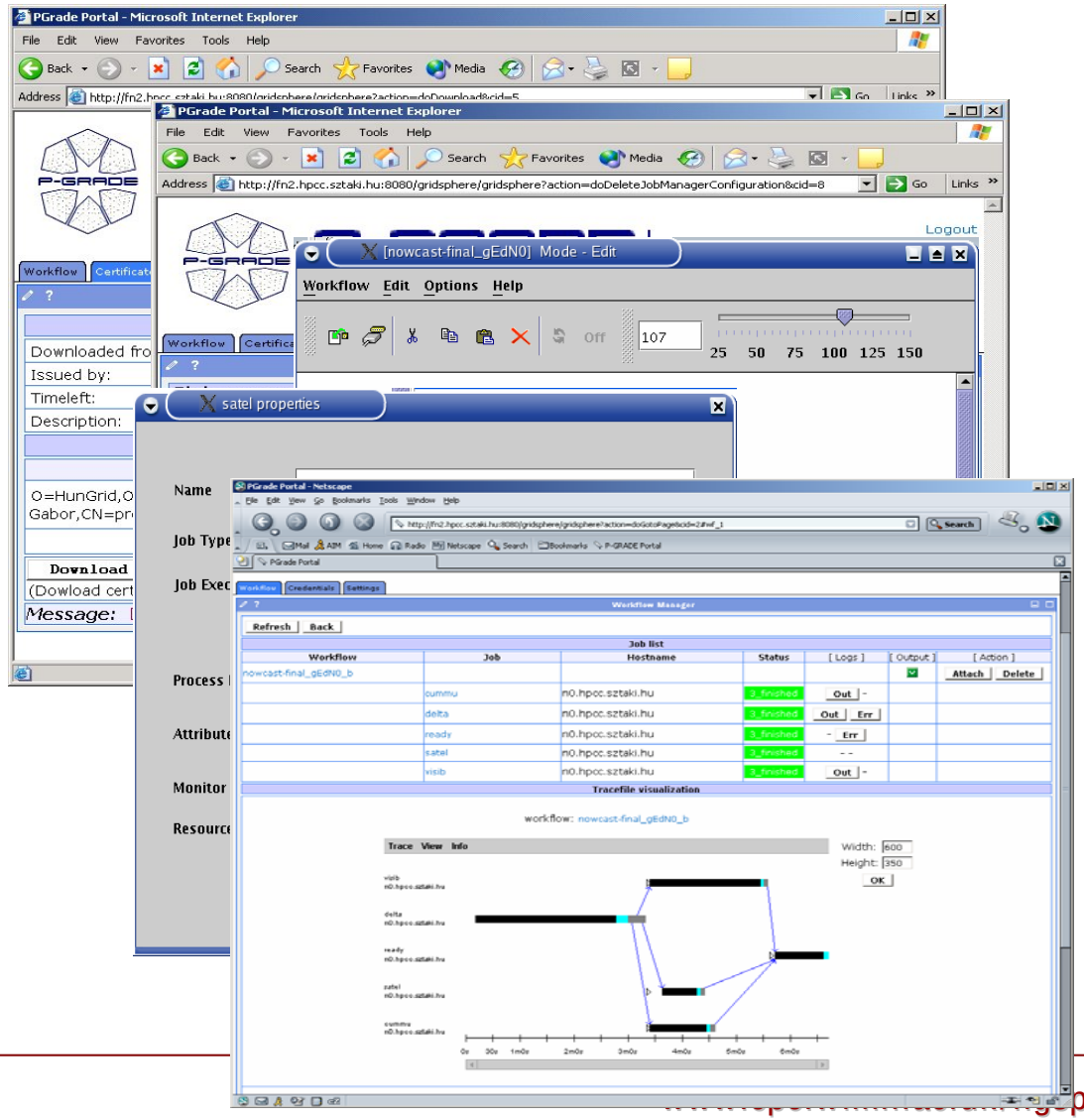
UK NGS

London

Rome

Athens

P-GRADE portal in a nutshell



Proxy management

Grid resources management

Workflow creation

Job mapping to Grid resources

Workflow management and

execution visualization

GEMLCA objectives

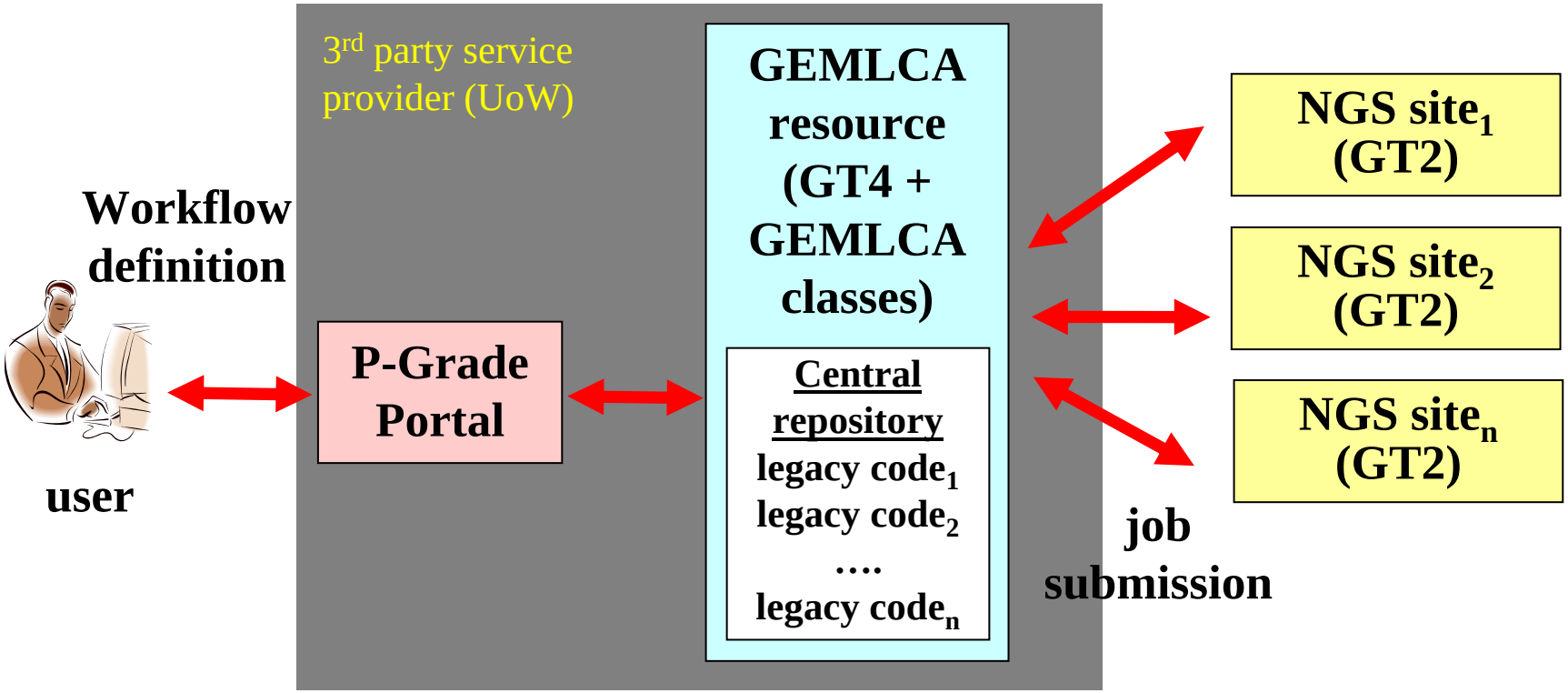
- **To deploy legacy code applications as Grid services without reengineering the original code and minimal user effort**

GEMLCA

- **To create Grid workflows where components can also be legacy code applications**
- **To make these functions available from a Grid Portal**

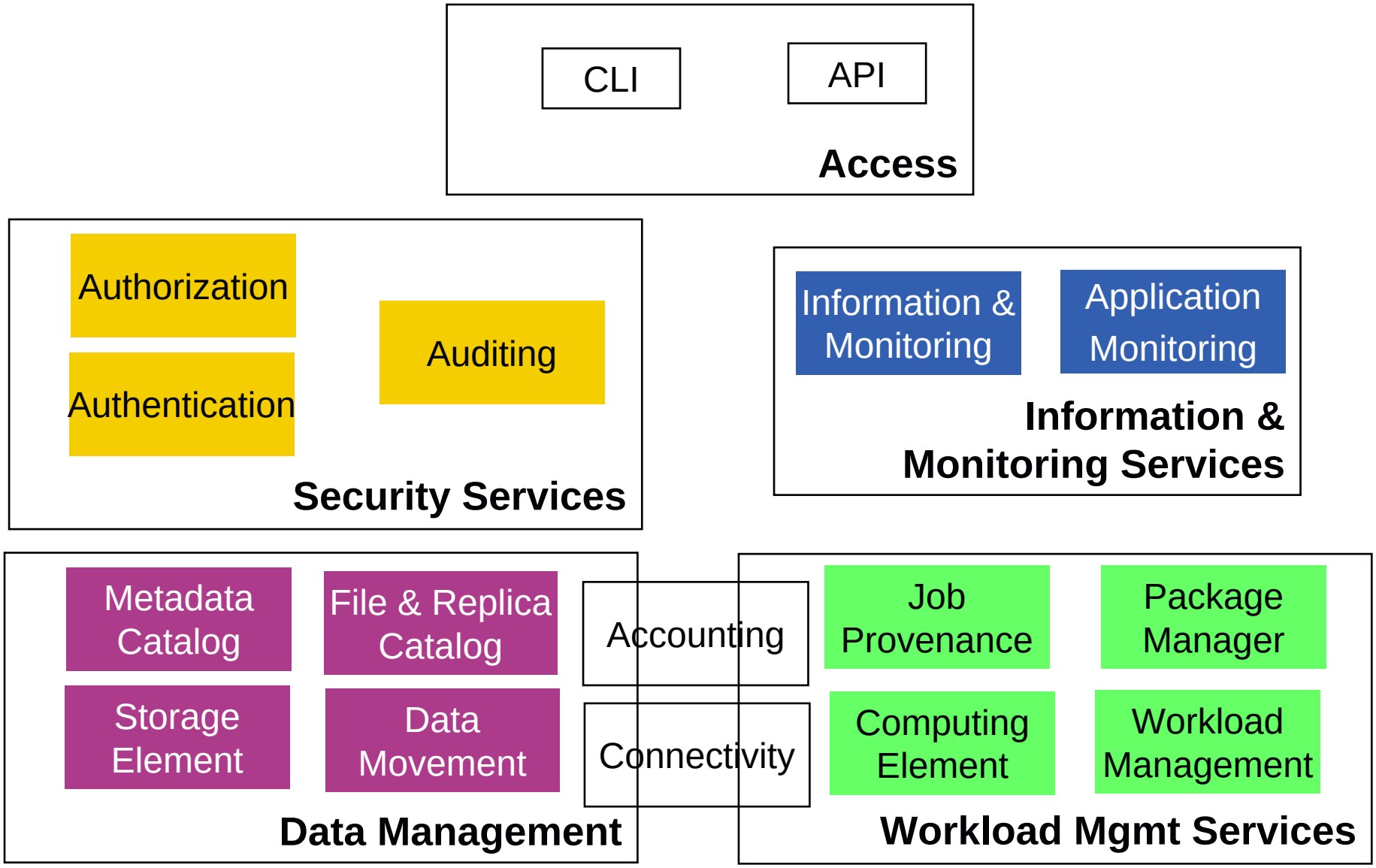
GEMLCA & P-GRADE Portal Integration

GEMLCA repository



More about gLite services

gLite Grid Middleware Services



More about gLite services

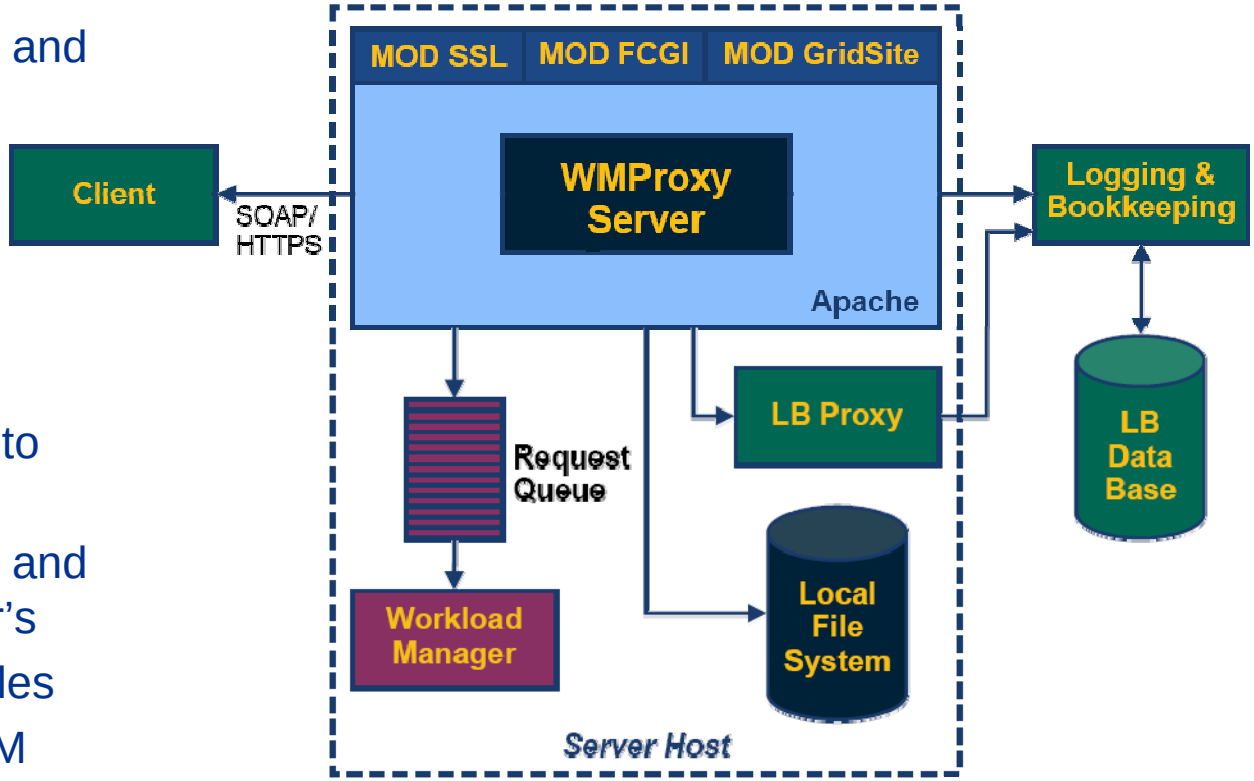
- **During today the focus is on:**
  - Description of new functionality in gLite 3.0 Workload Management
  - AMGA - Management of metadata
    - May have many thousands of files
    - Need to access and re-use based on characteristics… more than by their logical file names.
  - R-GMA - Monitoring of applications
    - May be running many long jobs
    - What's happening?!
  - APIs for some gLite services

Workload Management System

- **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 WMProxy**

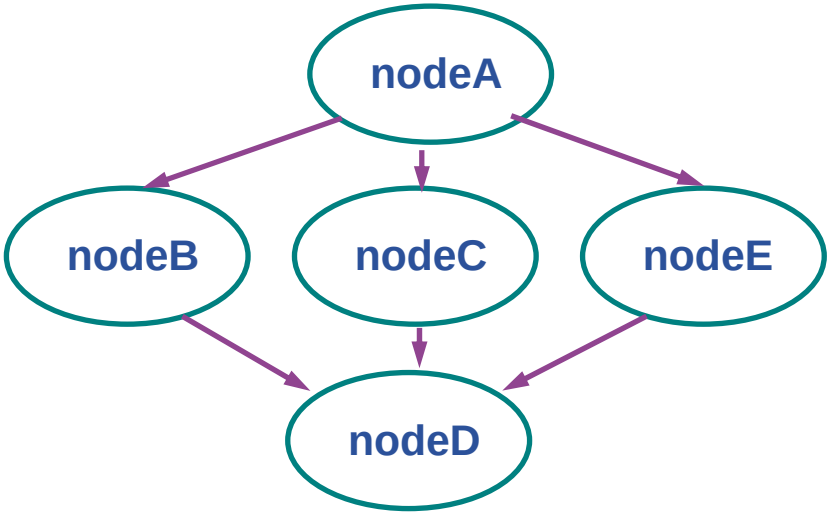
WMProxy

- **WMProxy 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



Complex Workflows

- **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 WMProxy 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

WMProxy: users' view

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

Overview of practicals

Overview of the practicals

- **Induction courses introduce basic services.**
- **Today: building more complex applications**
- **Use of WMS and WMProxy**
  - 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**