Introduction to Grid Application Development

Vangelis Floros
(efloros@cern.ch)
Application Support Team
NCSR “Demokritos”, Institute of Nuclear Physics

www.eu-egee.org

Information Society

Acknowledgments

- **Portion of slides (derived from those) prepared by:**
  - Mike Mineter, NESC
  - Charles Loomis, LAL-Orsay
  - Roberto Barbera and his GILDA team
    University of Catania and INFN
  - EGEE-II NA4 Activity Member’s

Contents

- Basic Concepts
- Types of Grid Applications
- Challenges
- Crash introduction to gLite services
- Application Families

- Working from the command line

- Introduction to the basic APIs

Definition, Ian Foster

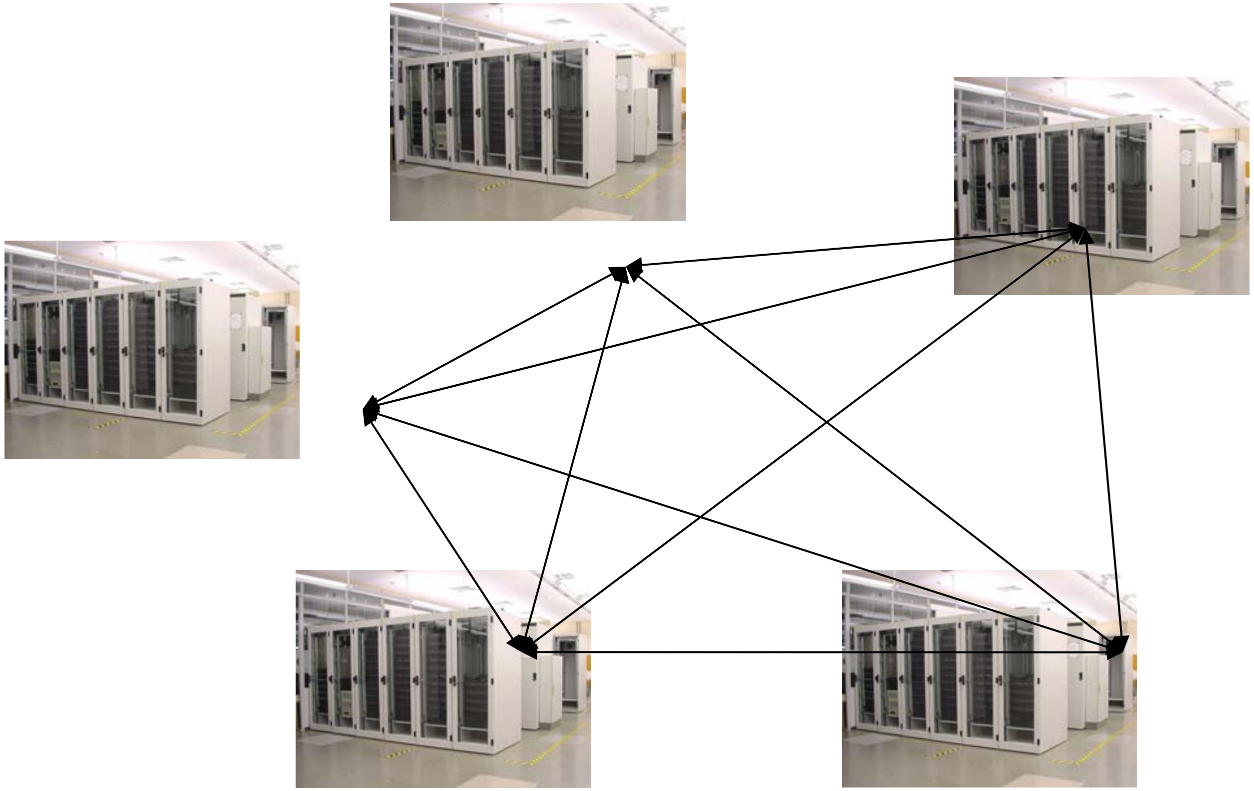
“Grid computing is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations” (I.Foster)

A *Virtual Organisation* is:

- **People from different institutions working to solve a common goal**
- **Sharing distributed processing and data resources**

Focus: Wide area, collaboration, virtual organisations

Practical definition in some areas

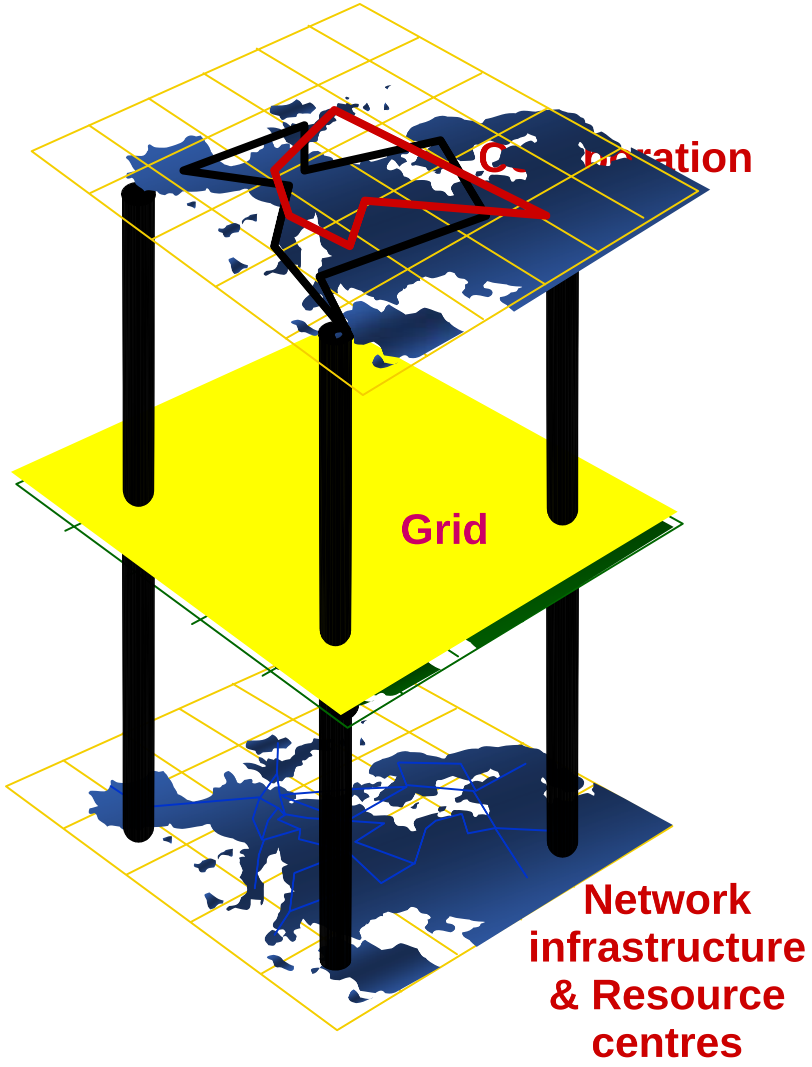


Grid Computing ==

Clustering Clusters; Building a global batch submission system ...

e-Science and the Grid

- *What is e-Science?*
  Collaborative science that is made possible by the sharing across the Internet of resources (data, instruments, computation, people's expertise...)
  - ***Often very compute intensive***
  - ***Often very data intensive (both creating new data and accessing very large data collections) – data deluges from new technologies***
  - ***Always: crosses administrative boundaries***
- *More precisely "e-Research": arts, humanities, social science,… are engaging with distributed environments.*
- *A Grid can orchestrate these resources*



Grids: a foundation for e-Research

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

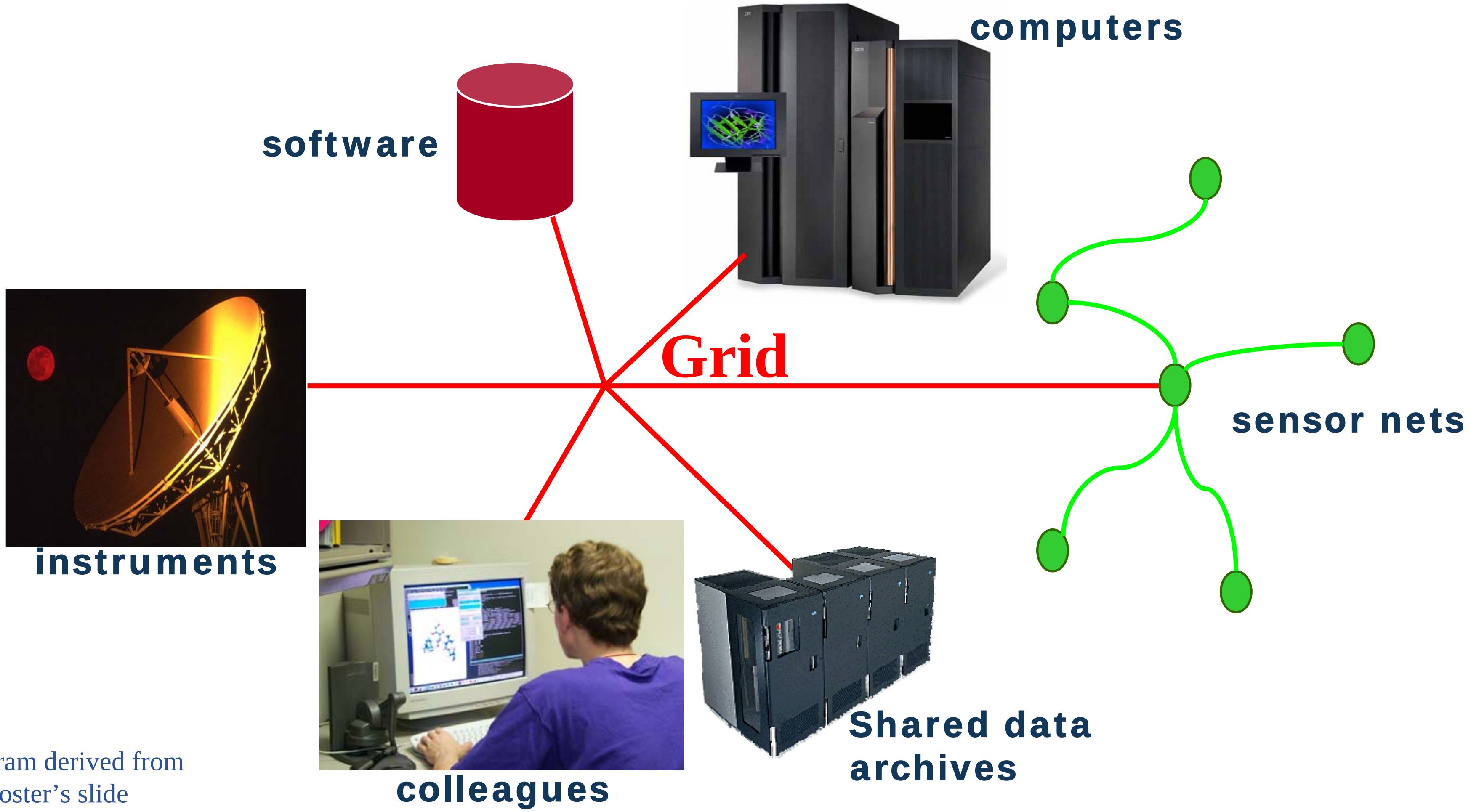
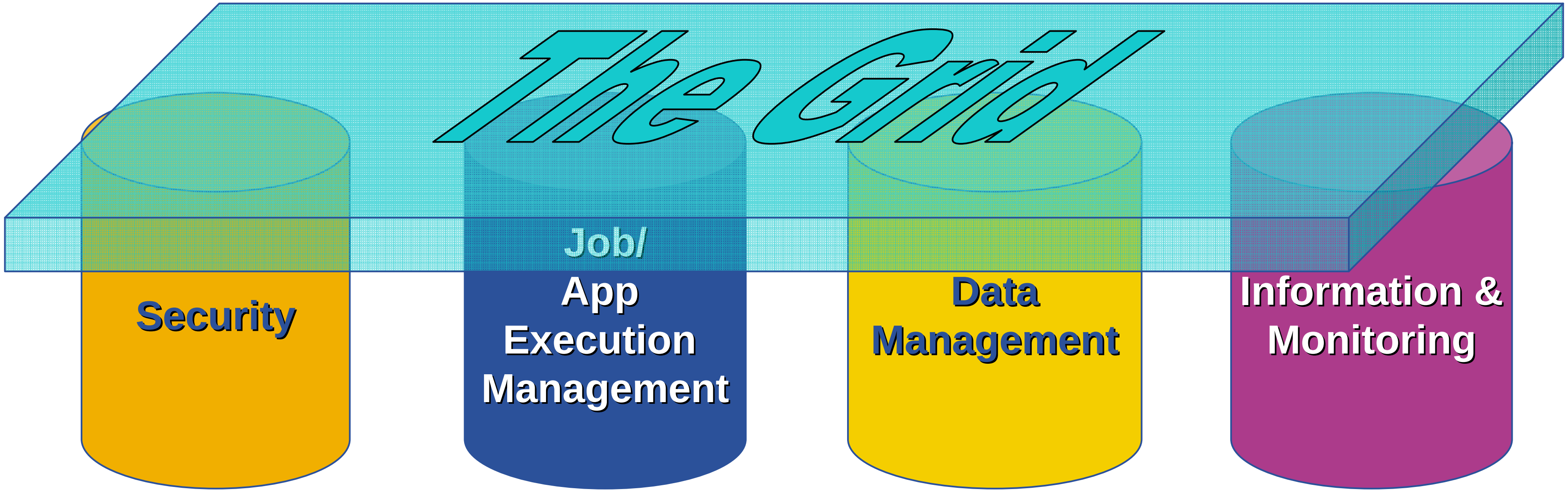


Diagram derived from
Ian Foster’s slide

Basic Concepts

The four pillars of Grid Computing



- Authentication
- Authorization
- Confidentiality
- Integrity
- VO management

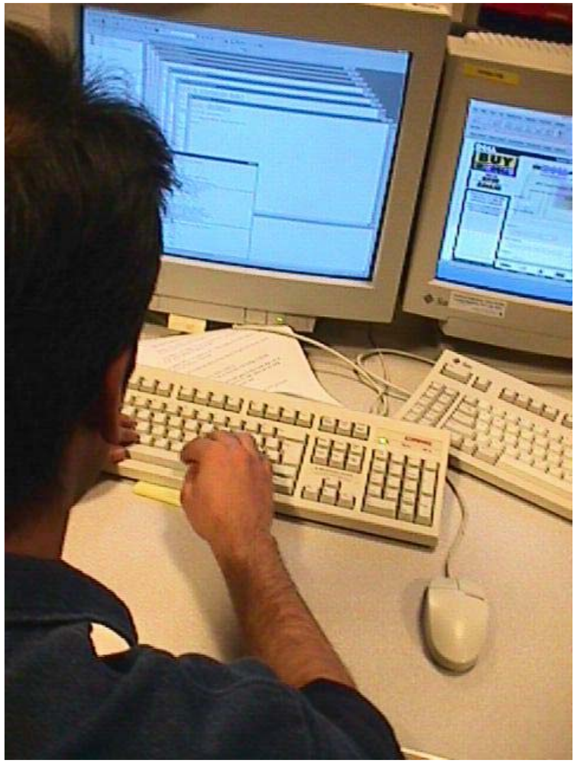
- Remote execution
- Load balancing
- Interactivity
- Parallelism
- Workflows

- Data staging
- Bulk data transfers
- Replication
- Metadata

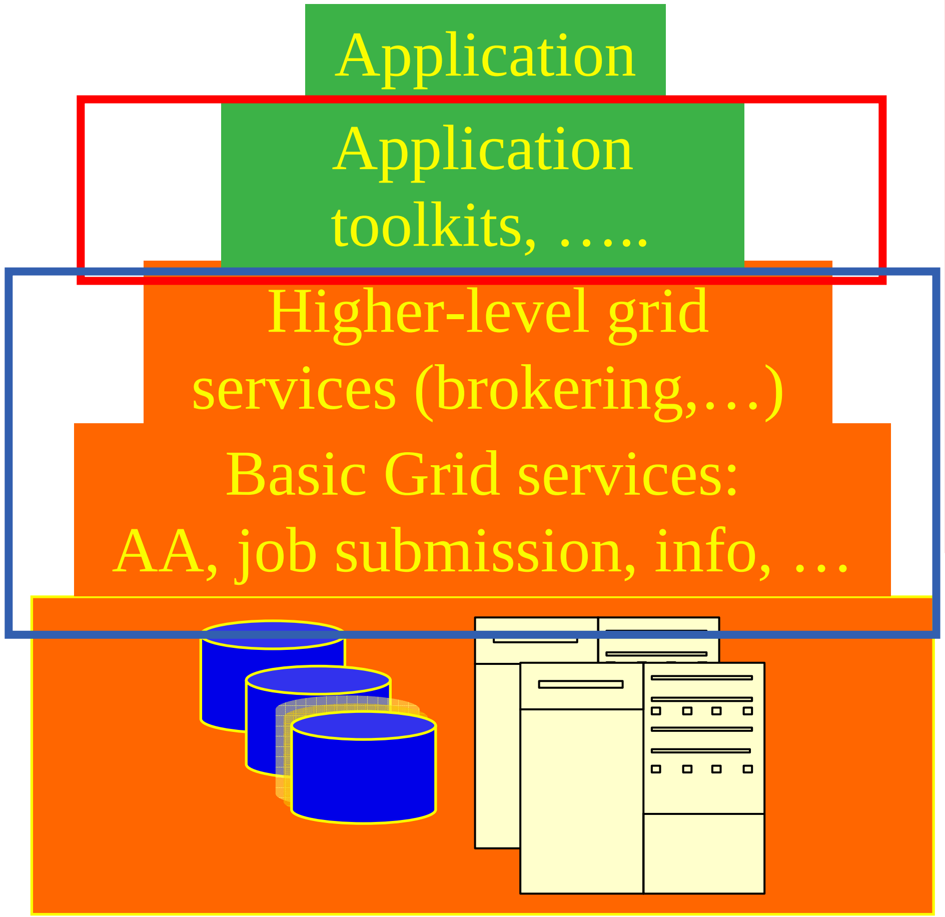
- Resource discovery
- Events & Notifications
- Resource status & monitoring

Grid Application Development

- Application development in the Grid implies the exploitation of APIs, tools and environments that provide the four basic Grid capabilities order to perform complex tasks and achieve diverse goals.
- The extend and approach that the four basic Grid concepts are materialized depends on the specific capabilities of the Grid enabling technologies (in our case the gLite middleware suite)



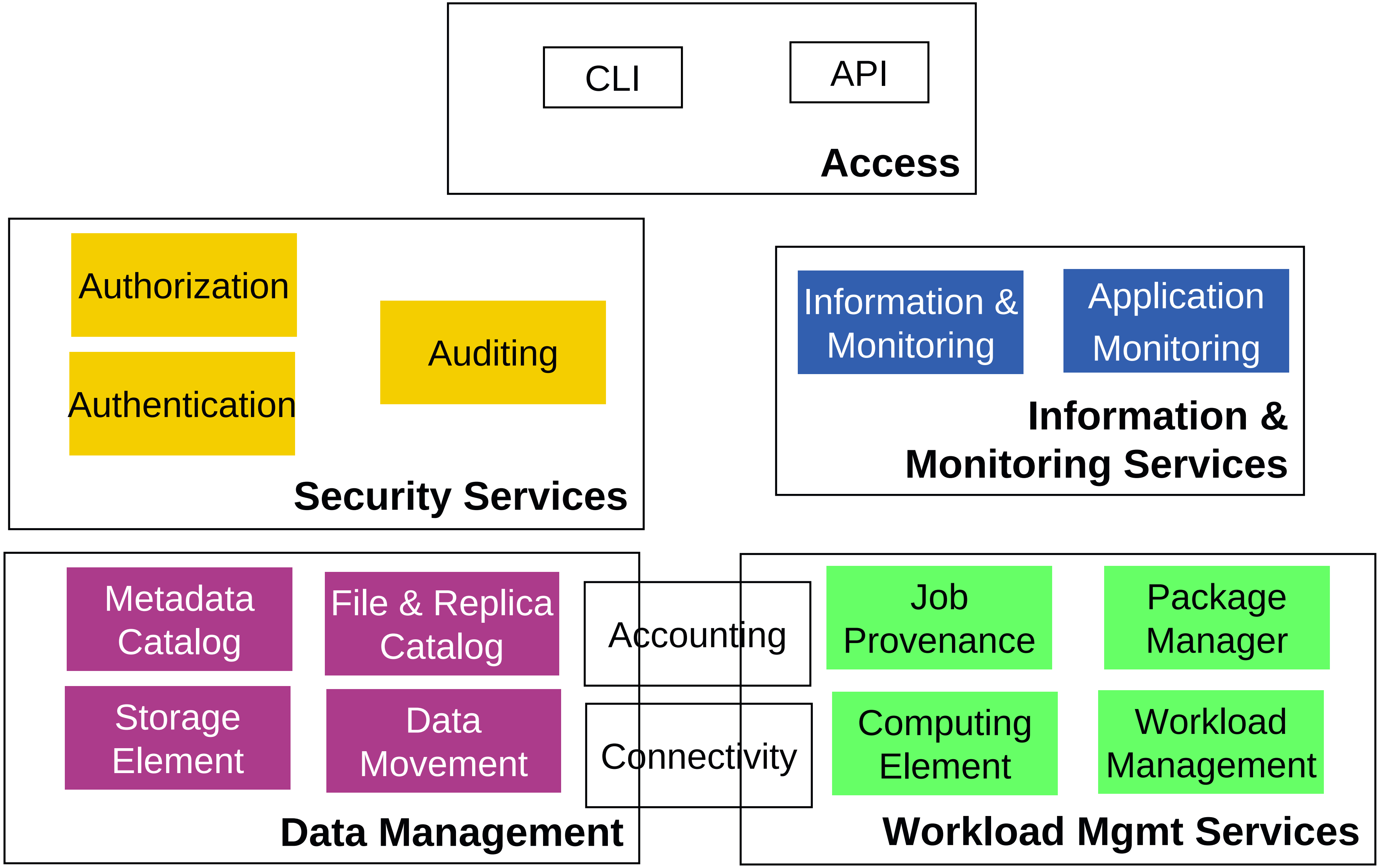
The vital layer


Application
Application toolkits, …..
Higher-level grid services (brokering,…)
Basic Grid services: AA, job submission, info, …

- 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.

gLite Grid Middleware Services



The Job concept

- **gLite follows the job submission concept for application execution and resource sharing**

- **A job is a self contained entity that packages and conveys all the required information and artifacts for the successful remote execution of an application.**
  - Executable files
  - Input/Output data
  - Parameters
  - Environment
  - Infrastructure Requirements
  - Workflows

Pragmatic approach to app dev

- Benefits and Restrictions.
- Potential compromises

- Resource sharing (no dedicated resources)
- Explicit and implicit collaboration (working in shared environment)
- Security risks
- Performance compromises (wrt system responsiveness, some times too much middleware!)
- Application Models (the application may have to adapt to the grid and not vice versa)

Types of grid applications

Complexities 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)
     - …

Gridification Levels

- **No development**. Wrap existing applications as jobs. No source code modification is required
- **Minor modifications**. The application exposes minimal interaction with the grid services (e.g. Data Managements)
- **Major modifications**. A wide portion of the code is re-written to adopt to the new environment (e.g. parallelization, metadata, information)
- **Pure grid applications**. Developed from scratch. Extensively exploit existing grid services to provide new capabilities customized for a specific domain (e.g. metadata, job management, credential management)

APIs

- Service Oriented vs Classic APIs

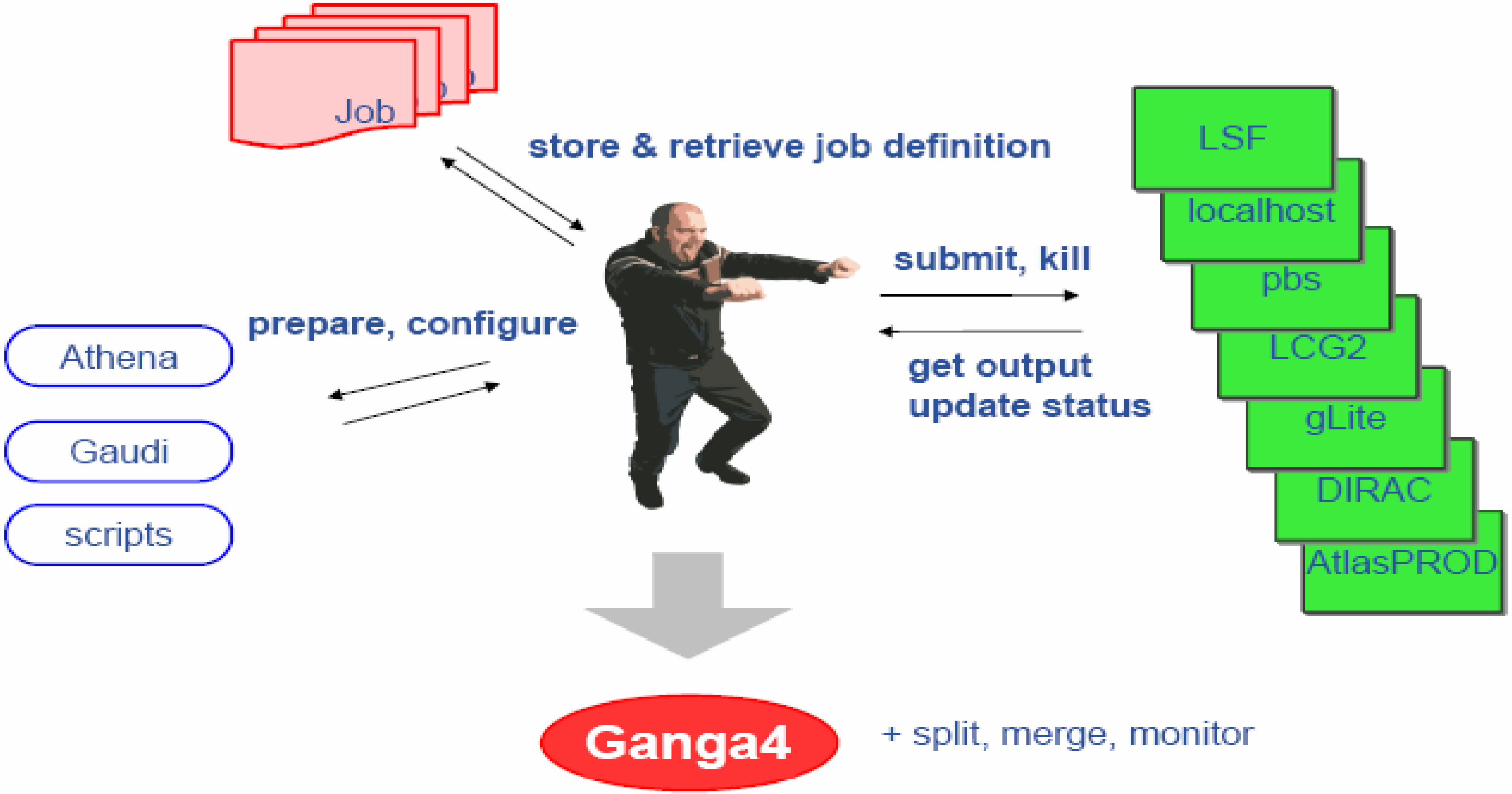
- Static compilation
- Shared libraries

- Libraries are transferred to precompiled Service clients
- May consume Web Service stubs and develop new clients from scratch

Invocation of applications

- **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

Example of higher-level tools: GANGA

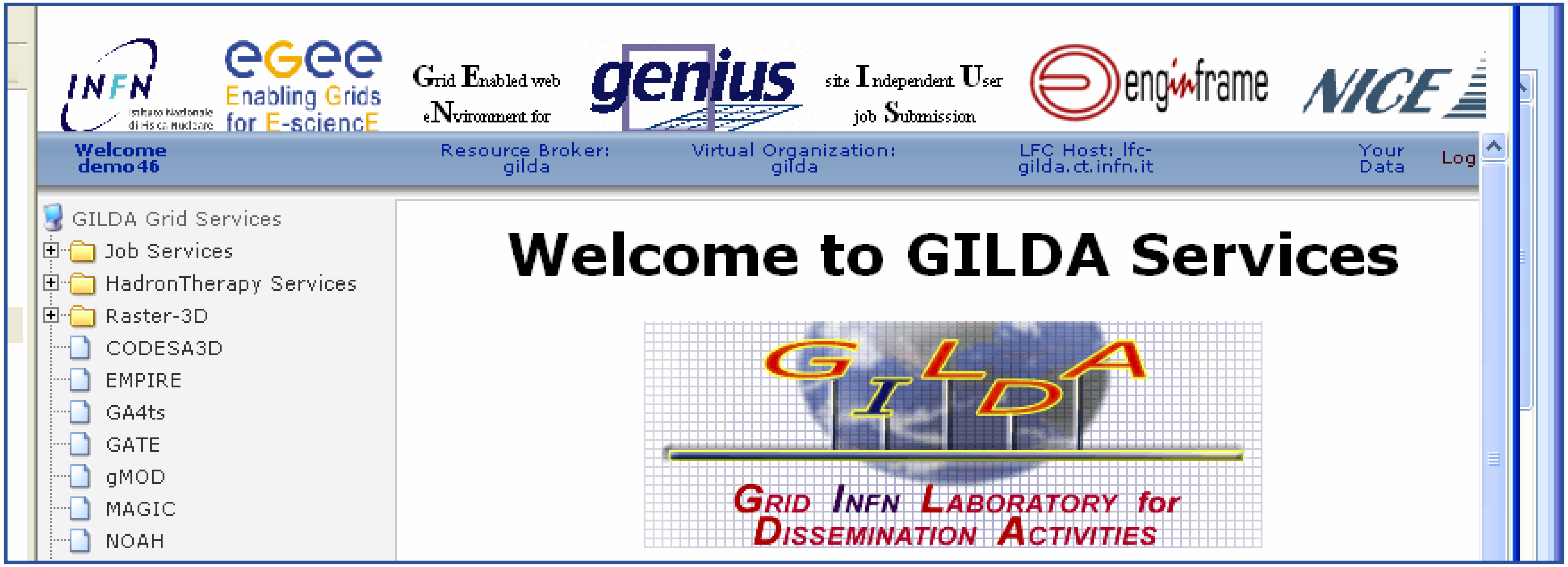


- **Ganga is a lightweight user tool**

ganga.web.cern.ch/

- **But also: Ganga is a developer framework**

GENIUS


INFN
Istituto Nazionale di Fisica Nucleare
eGee
Enabling Grids for E-sciencE
Grid Enabled web eNvironment for
genius
site Independent User job Submission
enginframe
NICE
Log
GILDA Grid Services
Job Services
HadronTherapy Services
Raster-3D
CODESA3D
EMPIRE
GA4ts
GATE
gMOD
MAGIC
NOAH
GILDA
GRID INFN LABORATORY for DISSEMINATION ACTIVITIES

Welcome demo46

Resource Broker: gilda

Virtual Organization: gilda

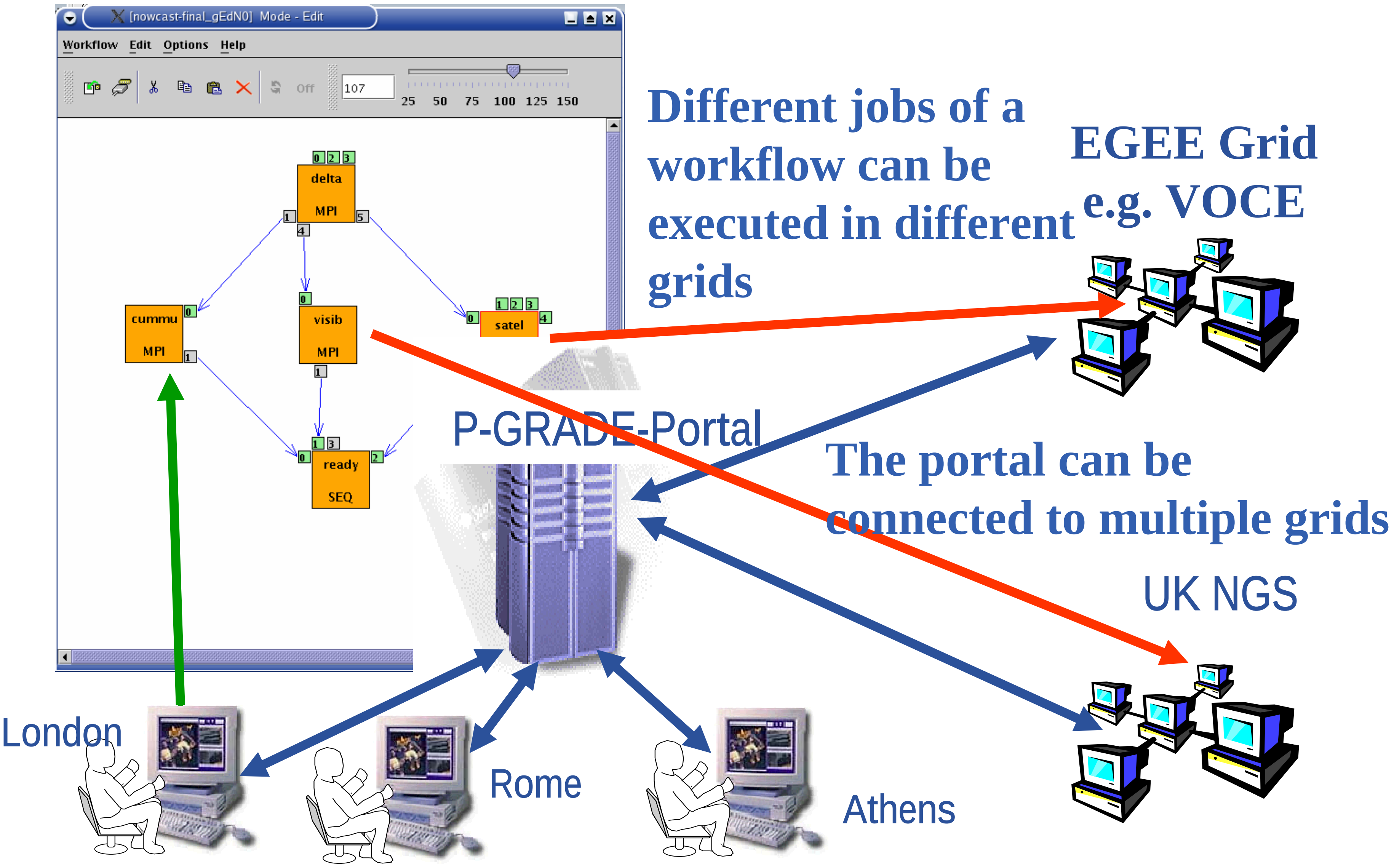
LFC Host: lfc-gilda.ct.infn.it

Your Data

Welcome to GILDA Services

- **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

Characteristics of VOs

- **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

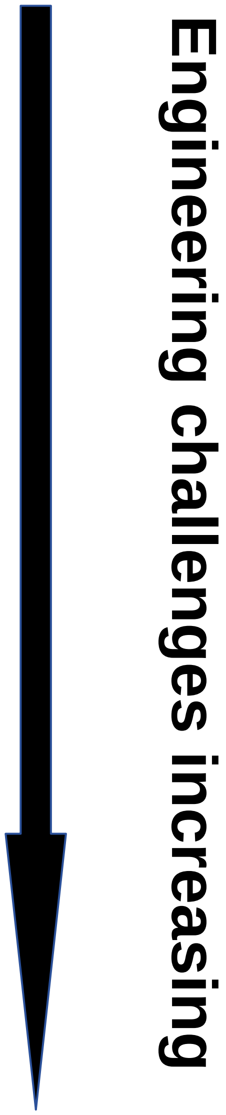
Challenges to researchers who write grid applications

Different Goals for App. Development

- **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

- **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**

Consequences

- **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

More about gLite services

- **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?!

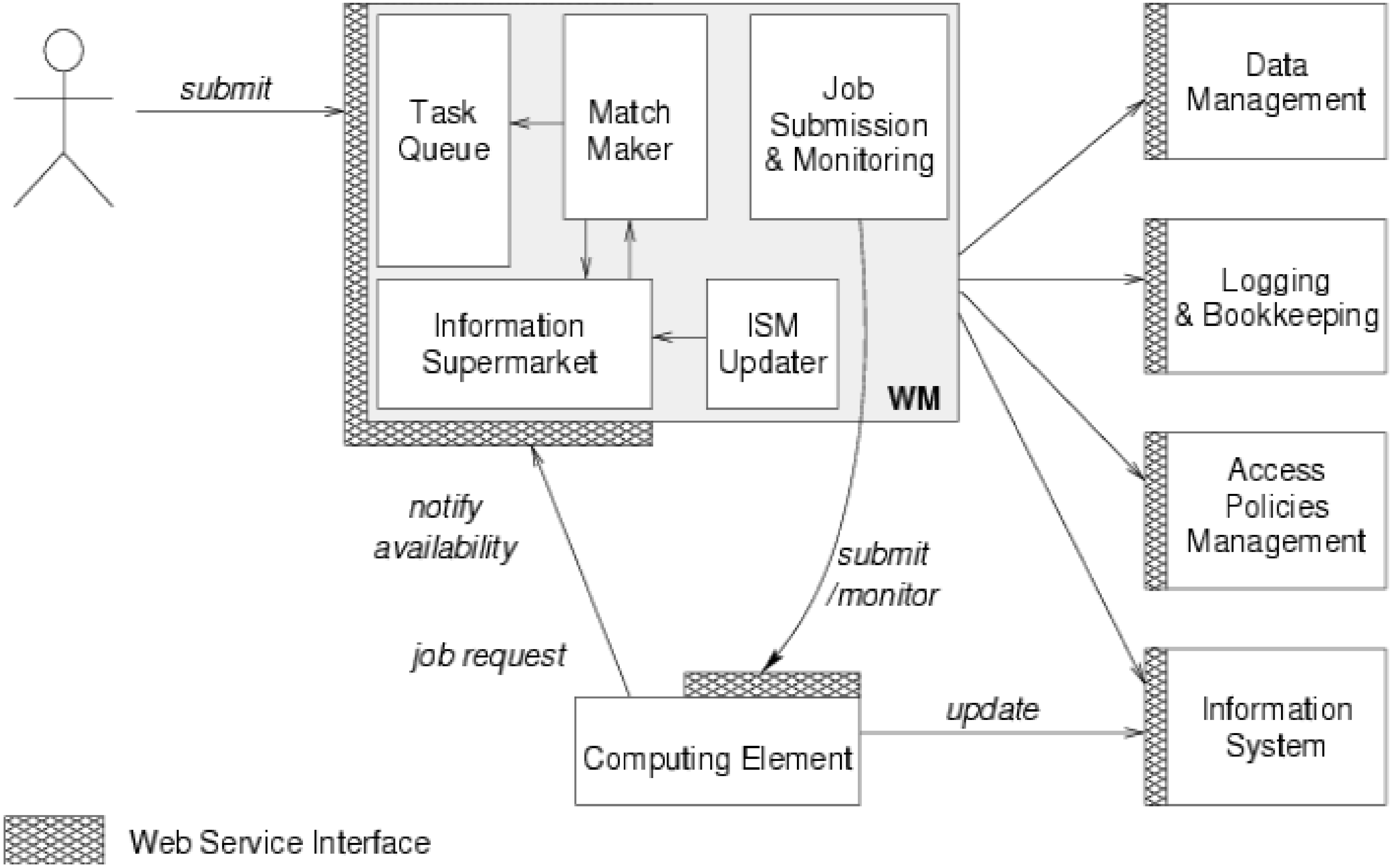
Workload Management System

- **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**

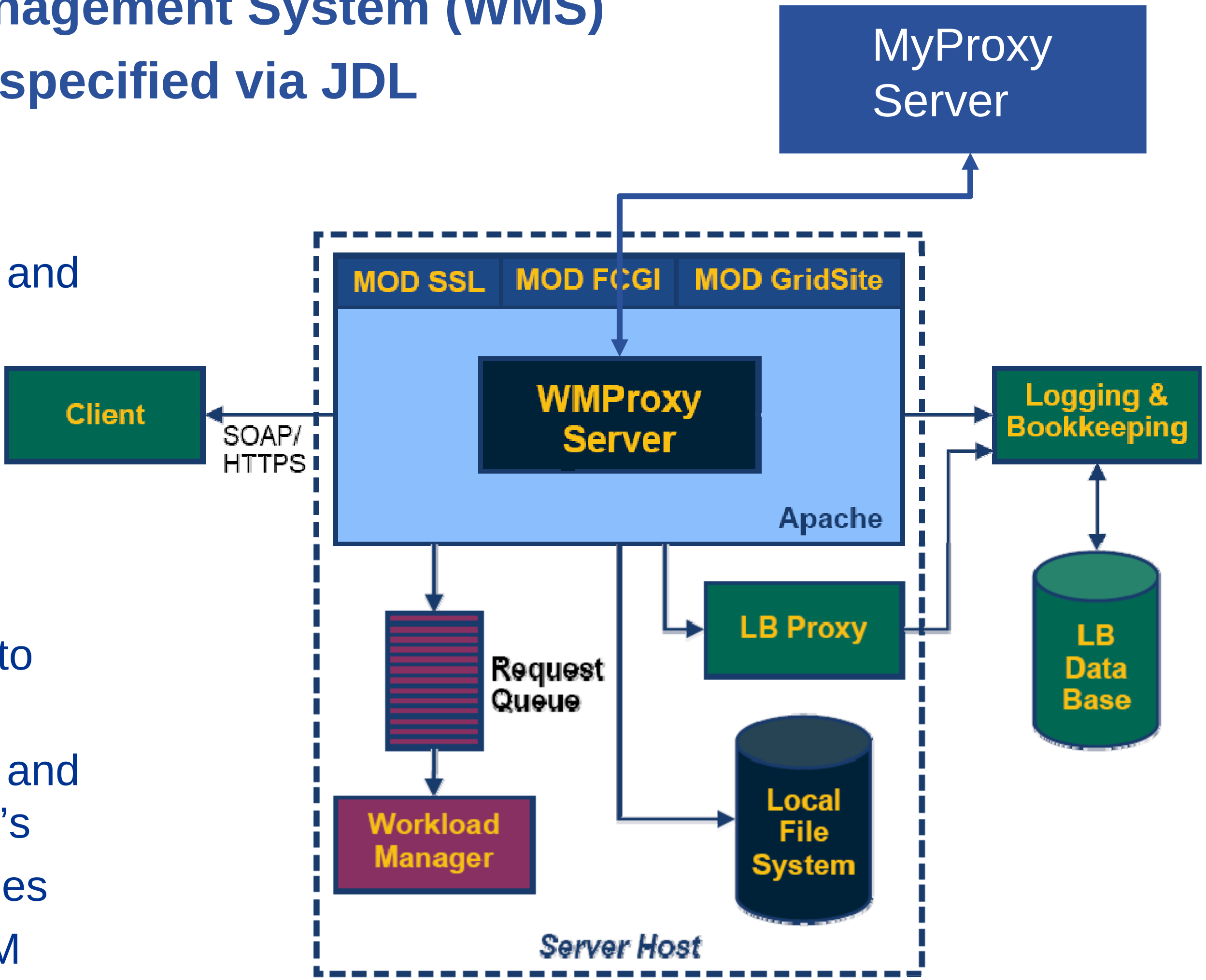
- **Web Service interface via WMProxy**

Workload Management System



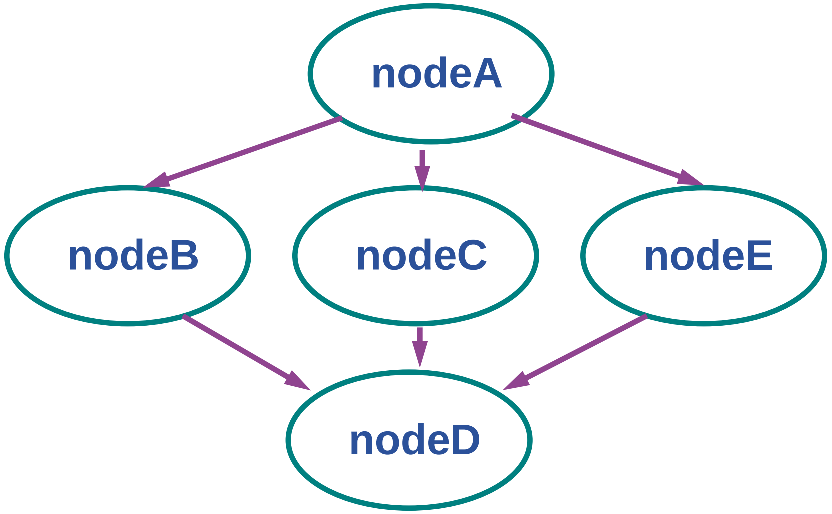
WMProxy

- **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

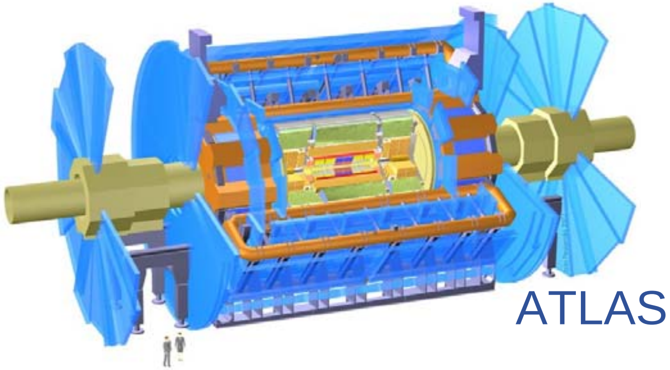
- **glite-wms-job-submit will supercede glite-job-submit (which is superceding edg-job-submit)**
- **Its support for compound jobs will simplify application software**
  - WMProxy manages sub-jobs
  - Shared Input and Output "sandboxes"
- **MUST establish proxy delegation before this can be used!**

Application Families

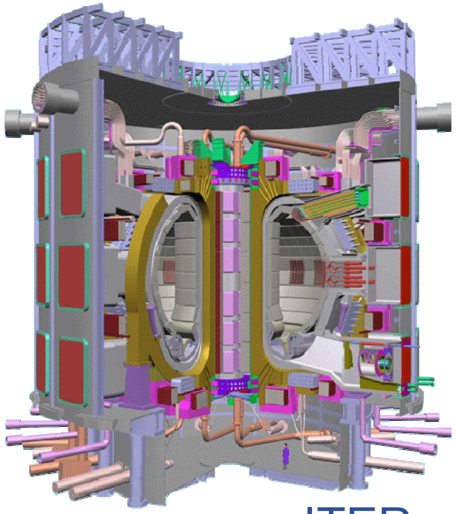
- Simulation
- Bulk Processing
- Responsive Apps.
- Workflow
- Parallel Jobs
- Legacy Applications

Simulation

- **Examples**
  - LHC Monte Carlo simulation
  - Fusion
  - WISDOM—malaria/avian flu
- **Characteristics**
  - Jobs are CPU-intensive
  - Large number of independent jobs
  - Run by few (expert) users
  - Small input; large output
- **Needs**
  - Batch-system services
  - Minimal data management for storage of results



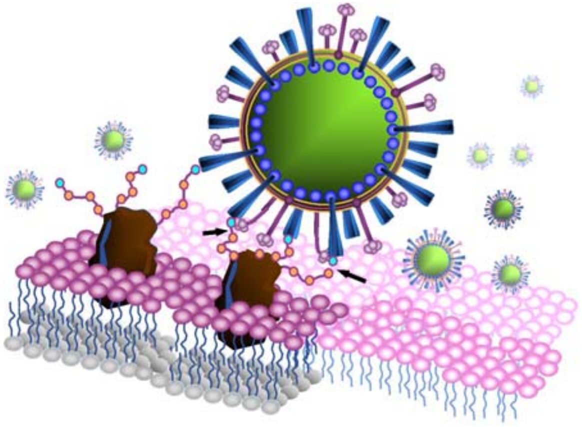
ATLAS



ITER

Drug Discovery

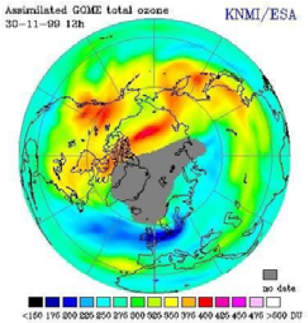
- **WISDOM focuses on in silico drug discovery for neglected and emerging diseases.**



- **Malaria — Summer 2005**
  - 46 million ligands docked
  - 1 million selected
  - 1TB data produced; 80 CPU-years used in 6 weeks
- **Avian Flu — Spring 2006**
  - H5N1 neuraminidase
  - Impact of selected point mutations on eff. of existing drugs
  - Identification of new potential drugs acting on mutated N1
- **Fall 2006**
  - Extension to other neglected diseases

Bulk Processing

- **Examples**
  - HEP processing of raw data, analysis
  - Earth observation data processing
- **Characteristics**
  - Widely-distributed input data
  - Significant amount of input and output data
- **Needs**
  - Job management tools (workload management)
  - Meta-data services
  - More sophisticated data management

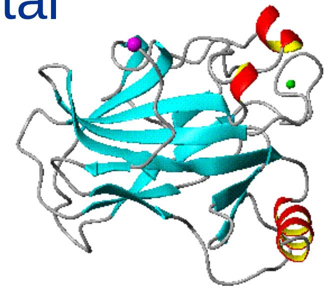

Assimilated GOME total ozone
30-11-99 12h
KNMI/ESA
no data
<150 175 200 225 250 275 300 325 350 375 400 425 450 475 >500 DU

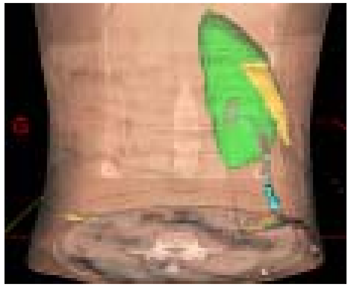
Responsive Apps. (I)

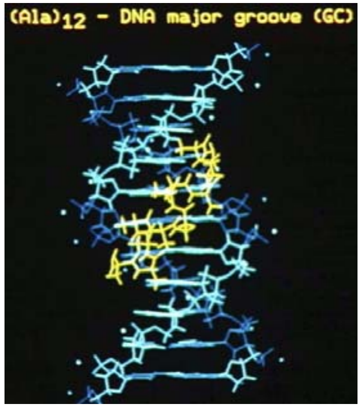
- **Examples**
  - Prototyping new applications
  - Monitoring grid operations
  - Direct interactivity
- **Characteristics**
  - Small amounts of input and output data
  - Not CPU-intensive
  - Short response time (few minutes)
- **Needs**
  - Configuration which allows “immediate” execution (QoS)
  - Services must treat jobs with minimum latency

Responsive Apps. (II)

- **Grid as a backend infrastructure:**
  - gPTM3D: interactive analysis of medical images
  - GPS@: bioinformatics via web portal
  - GATE: radiotherapy planning
  - DILIGENT: digital libraries
  - Volcano sonification



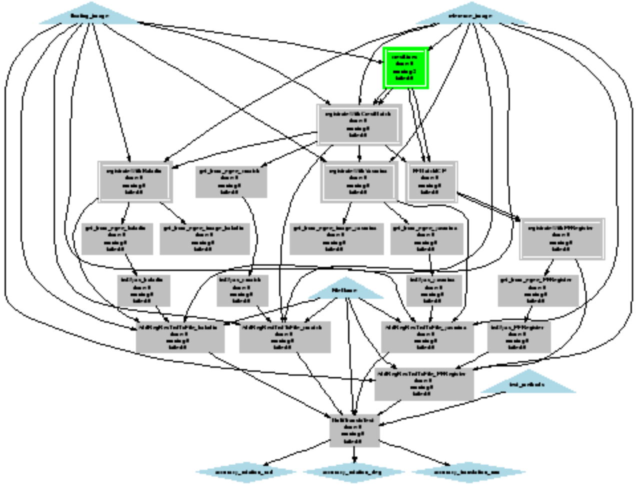




- **Characteristics**
  - Rapid response: a human waiting for the result!
  - Many small but CPU-intensive tasks
  - User is not aware of “grid”!
- **Needs**
  - Interfacing (data & computing) with non-grid application or portal
  - User and rights management between front-end and grid

Workflow

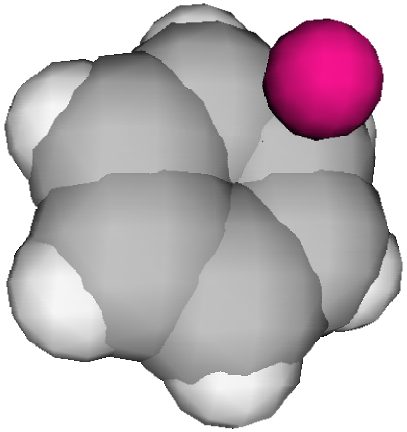
- **Examples**
  - “Bronze Standard”: image registratio
  - Flood prediction



- **Characteristics**
  - Use of grid and non-grid services
  - Complex set of algorithms for the analysis
  - Complex dependencies between individual tasks

- **Needs**
  - Tools for managing the workflow itself
  - Standard interfaces for services (I.e. web-services)

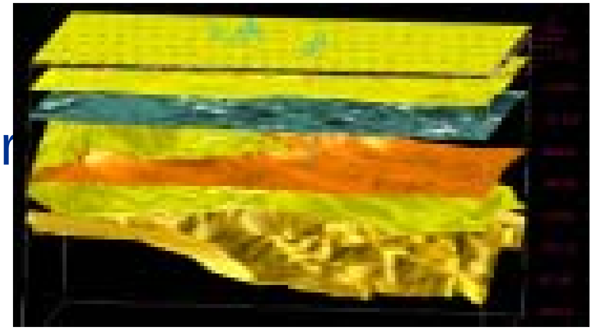
Parallel Jobs



- **Examples**
  - Climate modeling
  - Earthquake analysis
  - Computational chemistry
- **Characteristics**
  - Many interdependent, communicating tasks
  - Many CPUs needed simultaneously
  - Use of MPI libraries
- **Needs**
  - Configuration of resources for flexible use of MPI
  - Pre-installation of optimized MPI libraries

Legacy Applications

- **Examples**
  - Commercial or closed source binaries
  - Geocluster: geophysical analysis softwar
  - FlexX: molecular docking software
  - Matlab, Mathematics, …



- **Characteristics**
  - Licenses: control access to software on the grid
  - No recompilation ⇒ no direct use of grid APIs!
- **Needs**
  - License server and grid deployment model
  - Transparent access to data on the grid

Questions?

www.eu-egee.org

Information Society