Enabling Grids for E-sciencE

gLite Overview

Mike Mineter
National e-Science Centre, Edinburgh

Taipei, 22 August 2005

www.eu-egee.org

Information Society

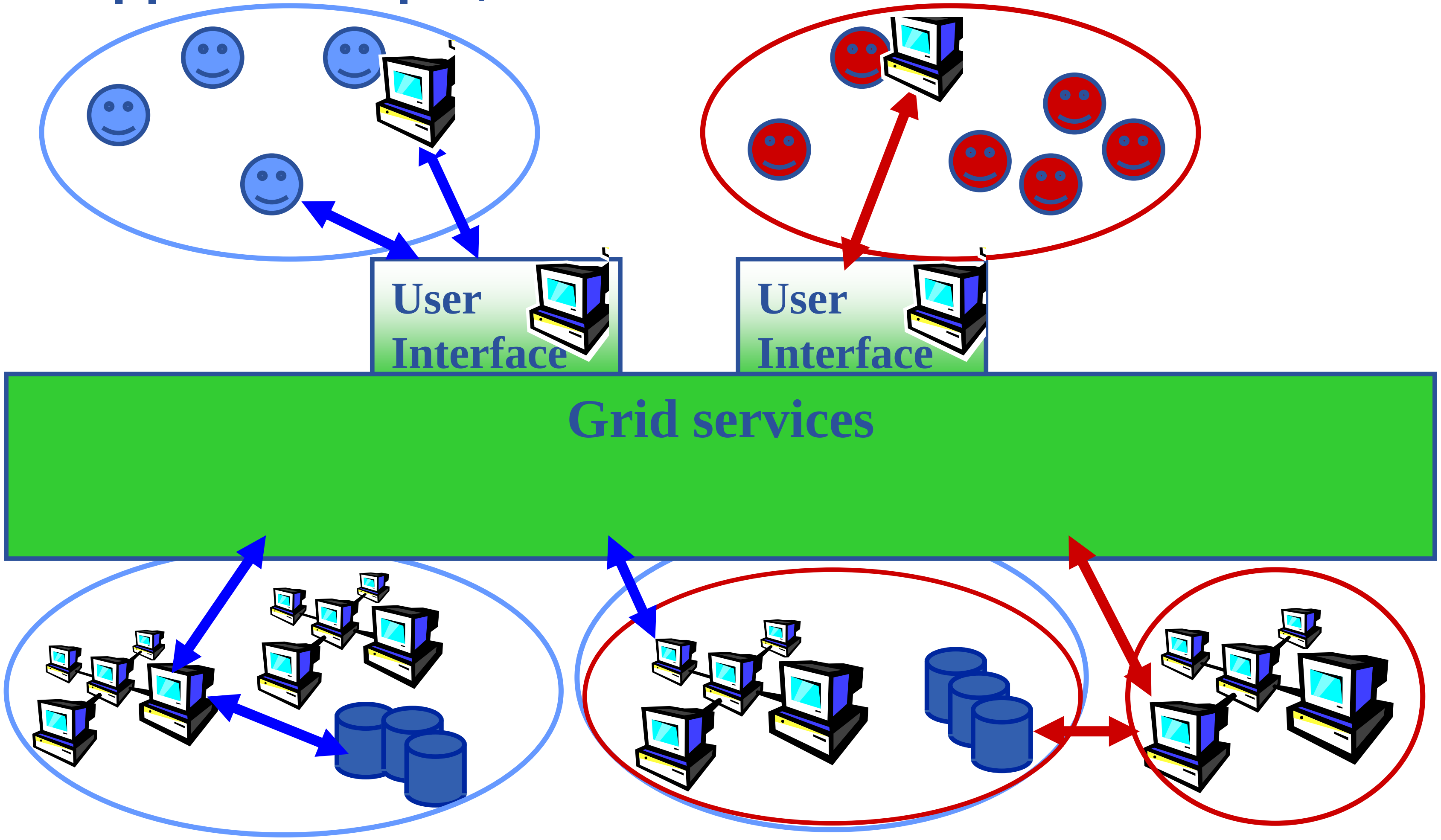
Outline

- Grid concepts
- Background to gLite
- gLite services
- gLite status
- Summary and conclusions

Grid concepts

A multi-VO Grid

- EGEE is establishing a production grid infrastructure to support multiple, diverse VO’s



User Interface

User Interface

Grid services

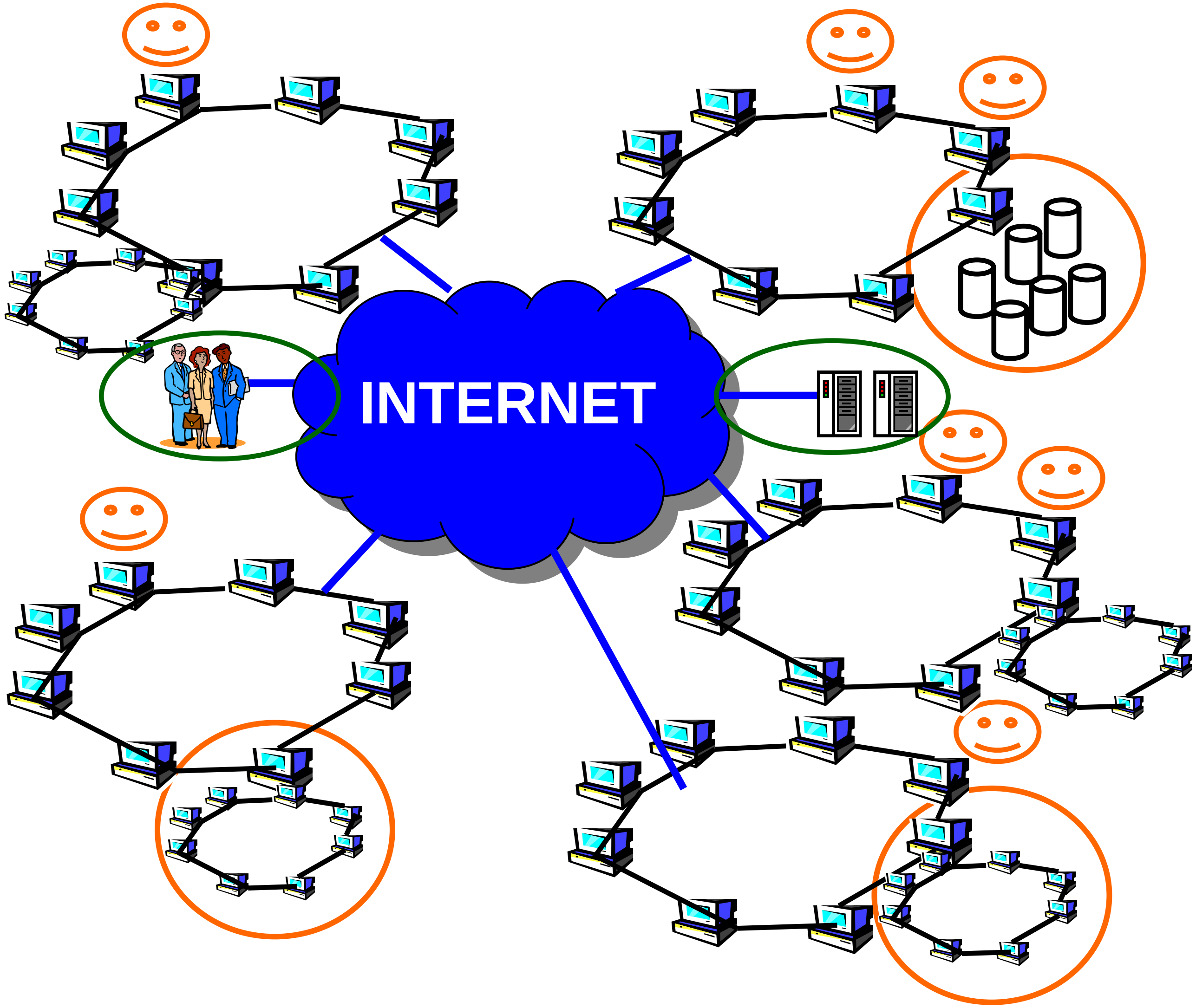
A multi-VO grid

- Users join VO
- Virtual organisation contributes resources & negotiates access

- Grid middleware runs on each resource
  - Data storage
  - (Usually) batch jobs on pools of processors

- Additional services (both people and middleware) enable the grid

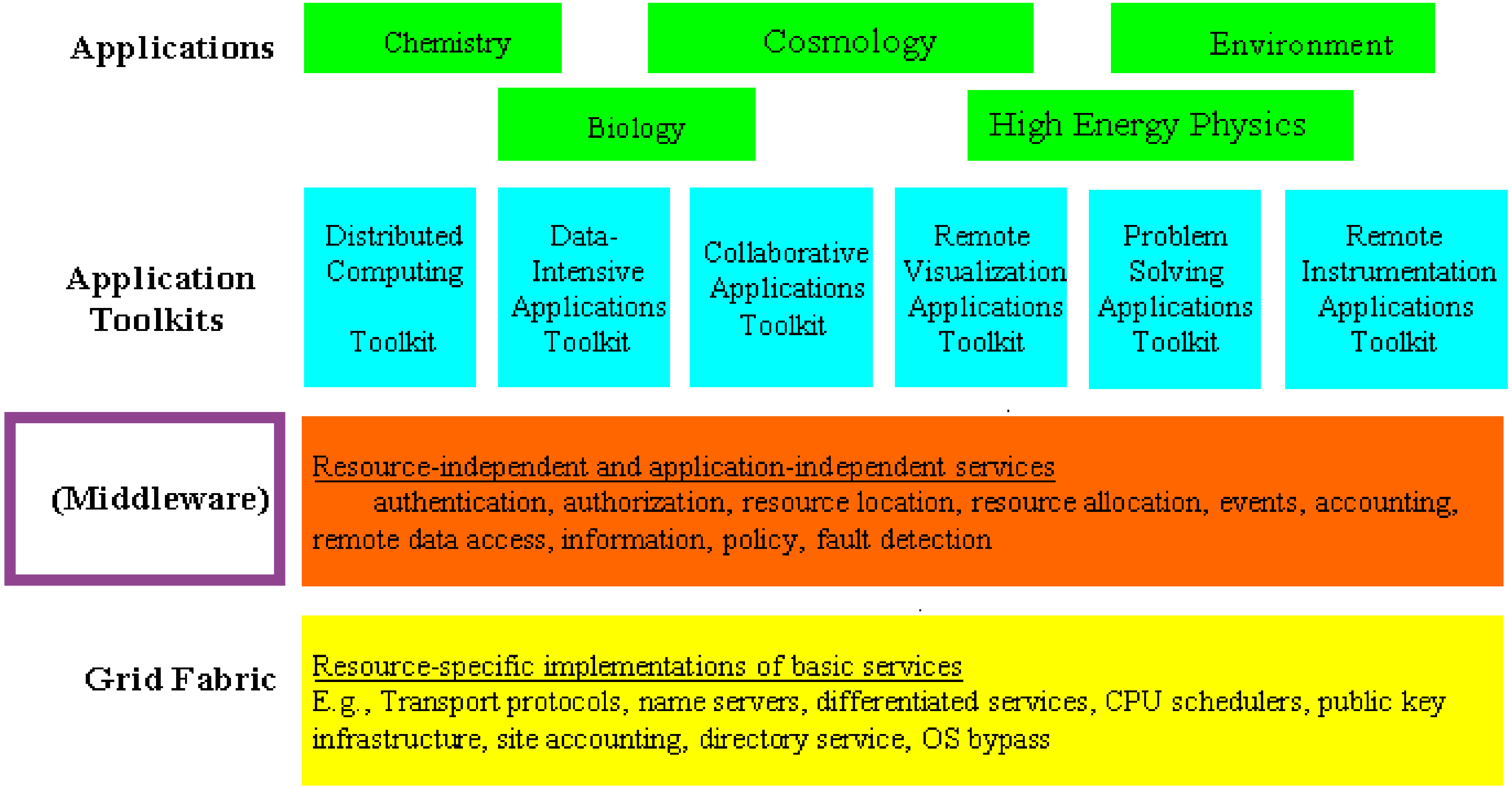
- Effect: “virtual computing” across administrative domains



Key concepts

- **Virtual organisation: people who collaborate by sharing resources – e.g. data, storage, CPU's, programs - across administrative and organisational boundaries**
- **Single sign-on**
  - I connect to one machine – some sort of "digital credential" is passed on to any other resource I use, basis of:
    - *Authentication*: How do I identify myself to a resource without username/password for each resource I use?
    - *Authorisation*: what can I do? Determined by
      - *My membership of a VO*
      - *VO negotiations with resource providers*
- **Grid middleware**
  - on each resource
  - services that enable the grid
- **User just perceives "shared resources" with no concern for location or owning organisation**

The Grid from a Services View



Applications

Chemistry

Cosmology

Environment

Biology

High Energy Physics

Application Toolkits

Distributed Computing Toolkit

Data-Intensive Applications Toolkit

Collaborative Applications Toolkit

Remote Visualization Applications Toolkit

Problem Solving Applications Toolkit

Remote Instrumentation Applications Toolkit

(Middleware)

Resource-independent and application-independent services

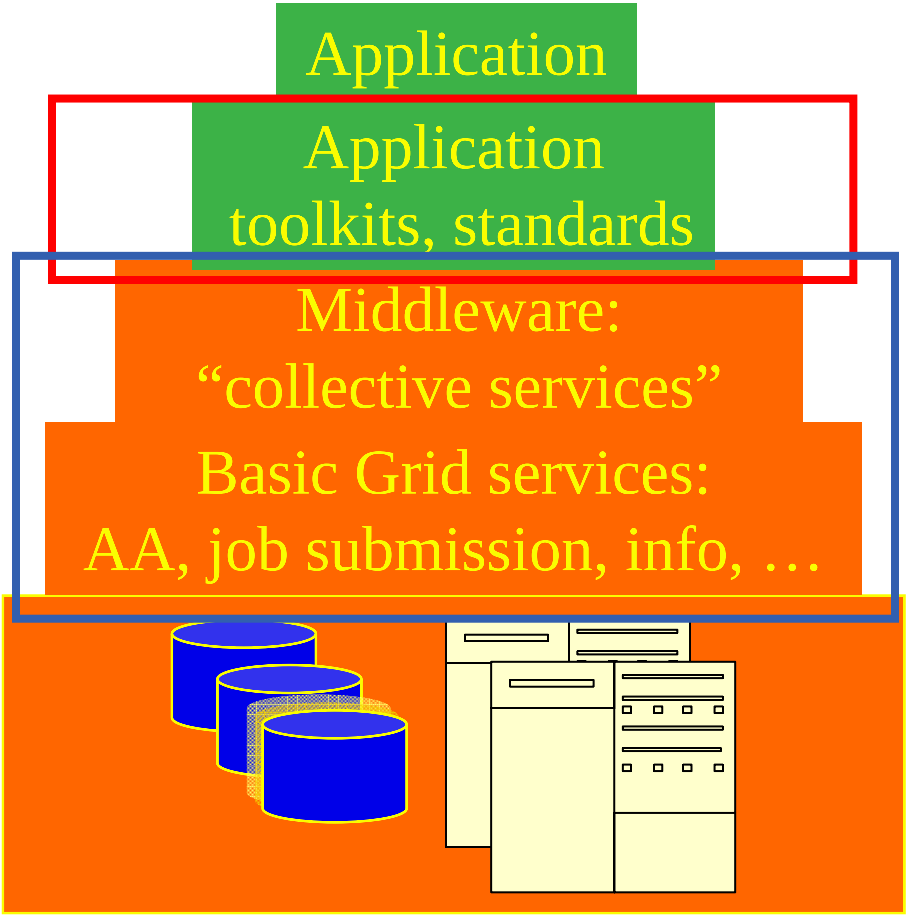
authentication, authorization, resource location, resource allocation, events, accounting, remote data access, information, policy, fault detection

Grid Fabric

Resource-specific implementations of basic services

E.g., Transport protocols, name servers, differentiated services, CPU schedulers, public key infrastructure, site accounting, directory service, OS bypass

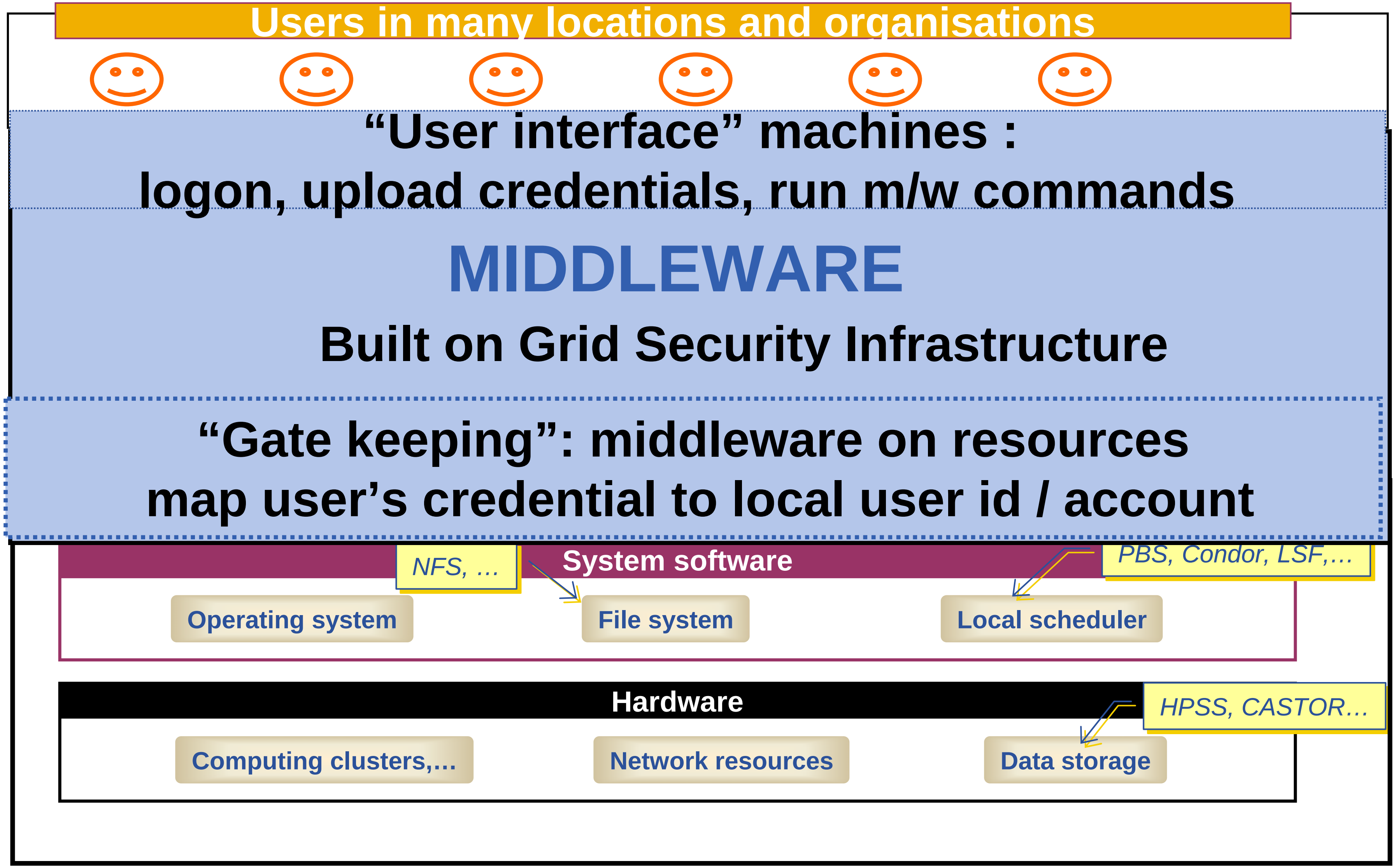
VO’s and middleware


Application
Application toolkits, standards
Middleware: “collective services”
Basic Grid services: AA, job submission, info, …

- The tools, services used by the VO’s applications
- Community-specific standards
- Application development environment, portals, semantics, workflow
- In EGEE-1: Mainly VO-specific

- Emphasis of EGEE -1 middleware

Authorisation, Authentication (AA)



Users in many locations and organisations

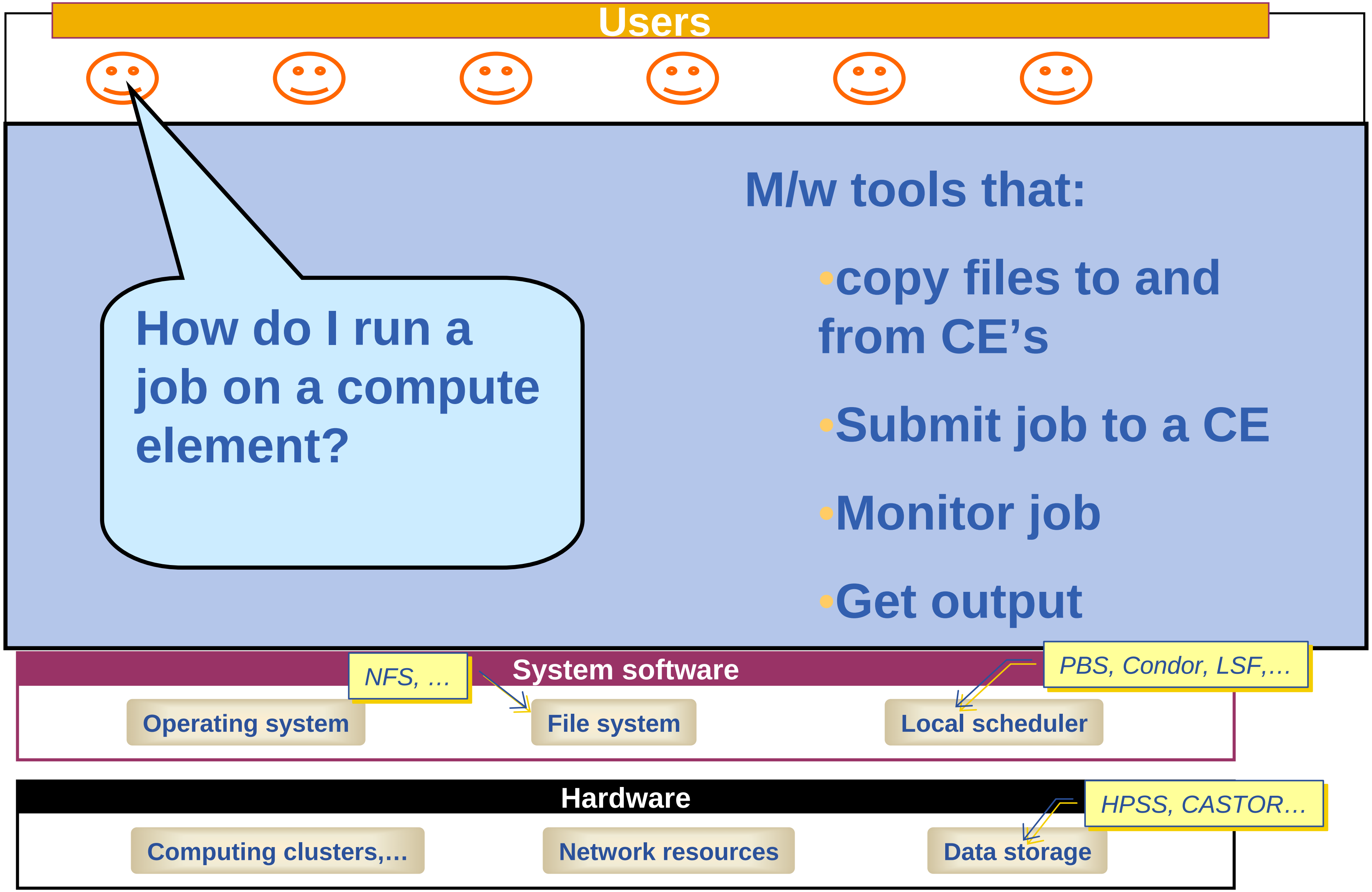
“User interface” machines :
logon, upload credentials, run m/w commands

MIDDLEWARE

Built on Grid Security Infrastructure

“Gate keeping”: middleware on resources
map user’s credential to local user id / account

Basic job submission



Users

How do I run a job on a compute element?

M/w tools that:

- copy files to and from CE's
- Submit job to a CE
- Monitor job
- Get output

System software

NFS, …

PBS, Condor, LSF,…

Operating system

File system

Local scheduler

Hardware

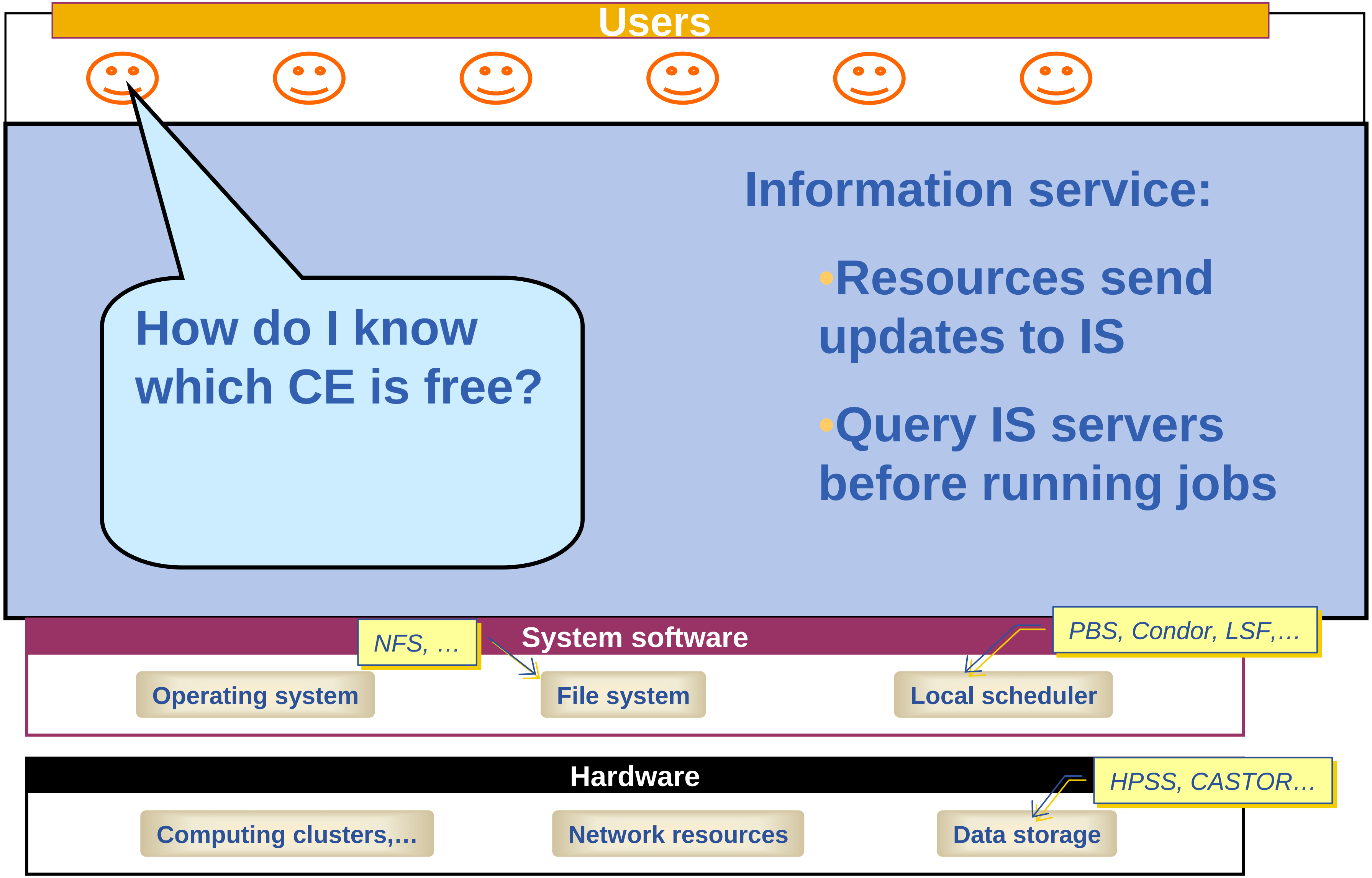
HPSS, CASTOR…

Computing clusters,…

Network resources

Data storage

Information service



Users

How do I know which CE is free?

Information service:

- Resources send updates to IS
- Query IS servers before running jobs

System software

NFS, …

PBS, Condor, LSF,…

Operating system

File system

Local scheduler

Hardware

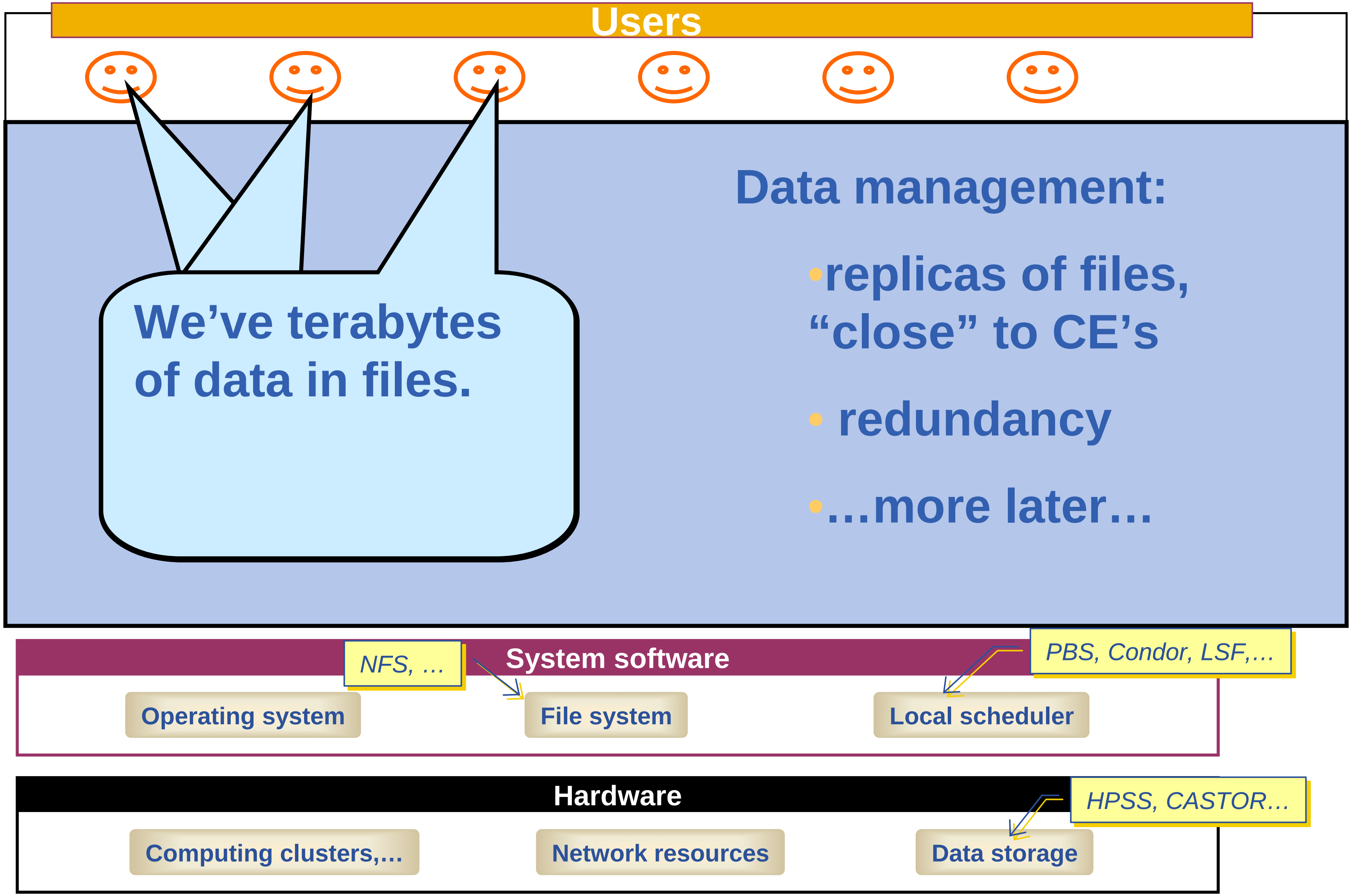
HPSS, CASTOR…

Computing clusters,…

Network resources

Data storage

File management



Users

We've terabytes of data in files.

Data management:

- replicas of files, "close" to CE's
- redundancy
- …more later…

System software

Operating system

File system

Local scheduler

NFS, …

PBS, Condor, LSF,…

Hardware

Computing clusters,…

Network resources

Data storage

HPSS, CASTOR…

Globus Toolkit

- **A software toolkit: a modular “bag of technologies”**
  - Made available under liberal open source license
- ***Not* turnkey solutions, but *building blocks* and *tools* for application developers and system integrators**
- **Tools built on Grid Security Infrastructure to include:**
  - Job submission: run a job on a remote computer
  - Information services: So I know which computer to use
  - File transfer: so large data files can be transferred
  - Replica management: so I can have multiple versions of a file “close” to the computers where I want to run jobs
- **Production grids are (currently) based on the Globus Toolkit release 2**
- **Globus Alliance: http://www.globus.org/**

Running a job with GT2

- An example of the command line interface to the *tool* for Job submission – need to know name of a CE to use

```
globus-job-submit grid-data.rl.ac.uk/jobmanager-pbs /bin/hostname -f

https://grid-data.rl.ac.uk:64001/1415/1110129853/

globus-job-status https://grid-data.rl.ac.uk:64001/1415/1110129853/

DONE

globus-job-get-output https://grid-data.rl.ac.uk:64001/1415/1110129853/

grid-data12.rl.ac.uk
```

- Build on this to support job submission to the grid, not to a CE – and to make more friendly interfaces for users

Background to gLite

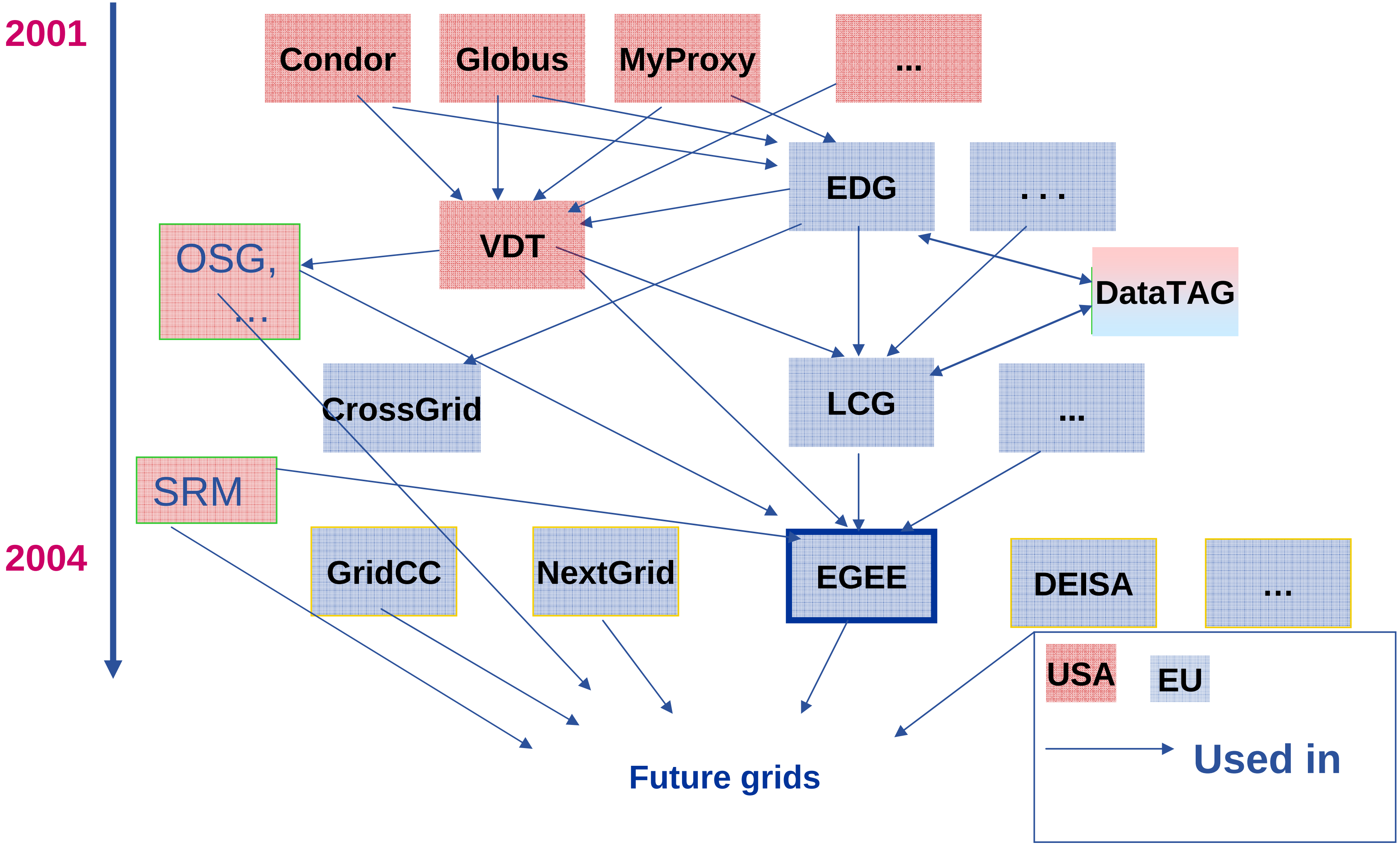
If “The Grid” vision leads us here…

… then where are we now?

Grids: where are we now?

- **Many key concepts identified and known**
- **Many grid projects have tested, and benefit from, these**
- **Major efforts now on establishing:**
  - Standards (a slow process)
    (e.g. Global Grid Forum, http://www.gridforum.org/ ,
    OASIS, W3C, IETF)
  - Production Grids *for multiple VO's*
    - "Production" = Reliable, sustainable, with commitments to quality of service
    - One stack of middleware that serves many communities
    - Operational procedures and services (people, policy,..)
  - New user communities
- **… whilst research & development continues**

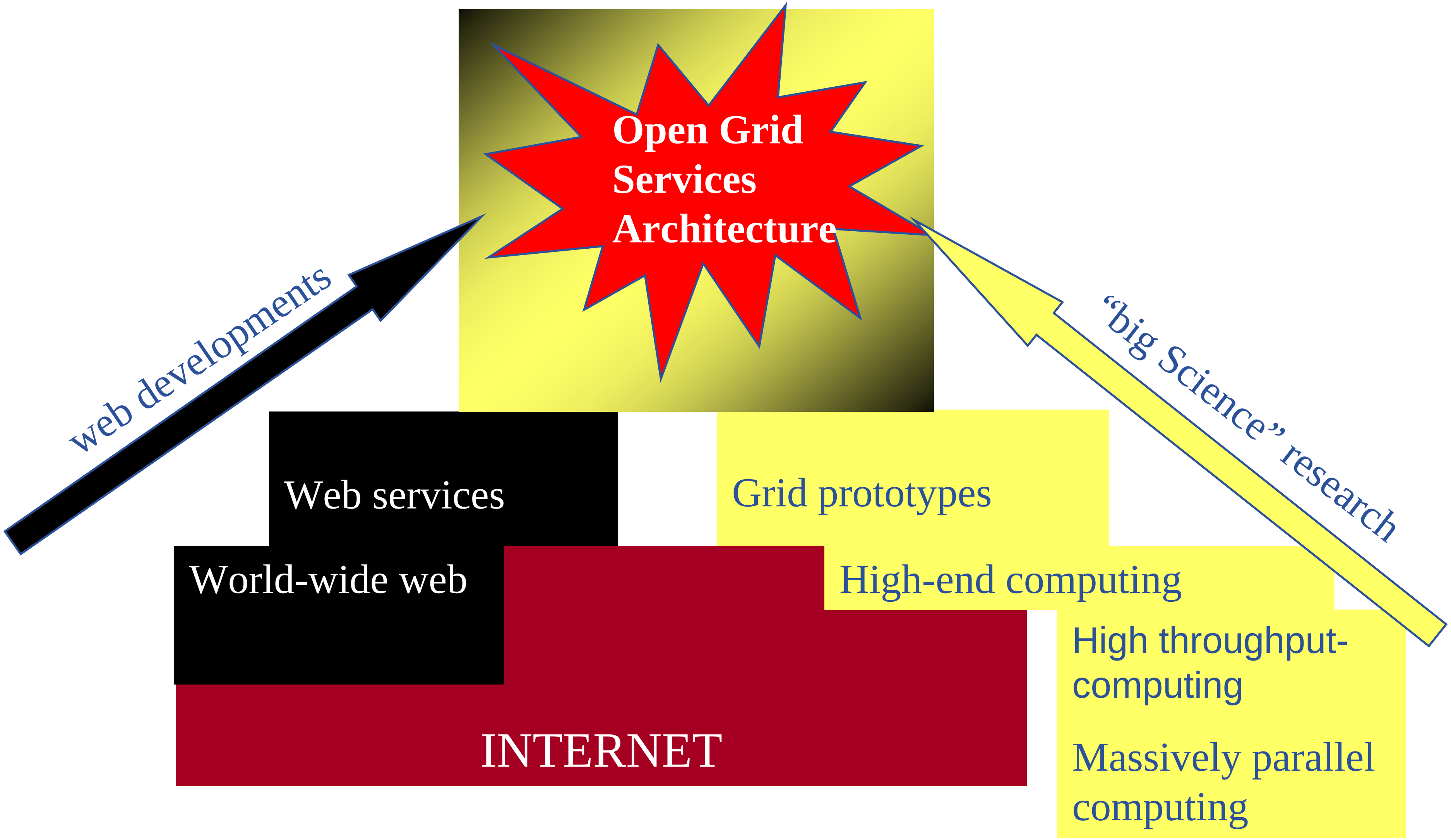
Parts of the Grid "ecosystem"



EGEE Middleware - gLite

- **gLite: is a Service Oriented Architecture**
  - Components are loosely coupled by messages
  - Facilitates interoperability
  - Allows easier compliance with upcoming standards, hosting environments, toolkits
    - Follow WSRF standardization
    - Start with plain WS (WS-I)
  - Architecture is not bound to specific implementations
  - Heterogeneous resources (storage, computation…)
- **Flexibility in configuration**
  - services can be deployed and used independently
- **The gLite service decomposition has been largely influenced by the work performed in the LCG project**

The vision of 2001: convergence of Web Services and Grids



Service orientation

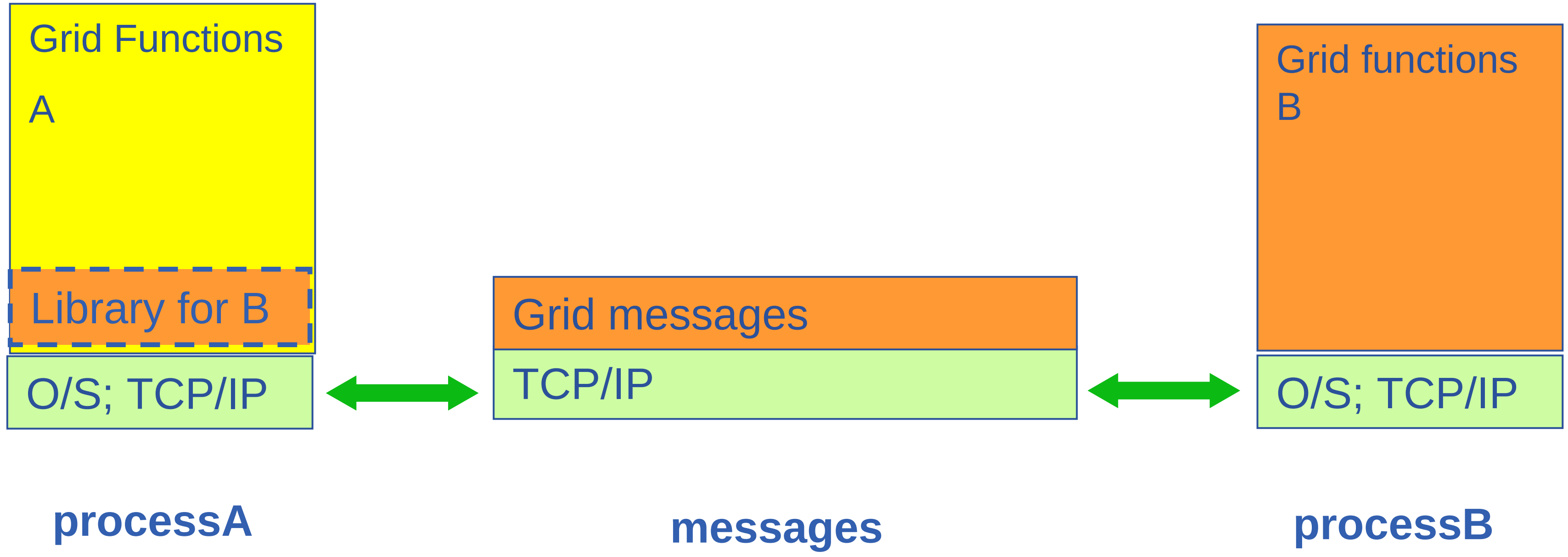
- **Components that are loosely coupled by messages**
  - Accessible across network; modular and self-contained; clean modes of failure
  - So can change implementation without changing interfaces
  - Can be developed in anticipation of new uses
- **… and are based on standards.**

- **Opens EGEE to:**
  - New middleware (plethora of tools now available)
  - Heterogeneous resources (storage, computation…)
  - Interact with other Grids (international, regional and national)

Grids before service orientation

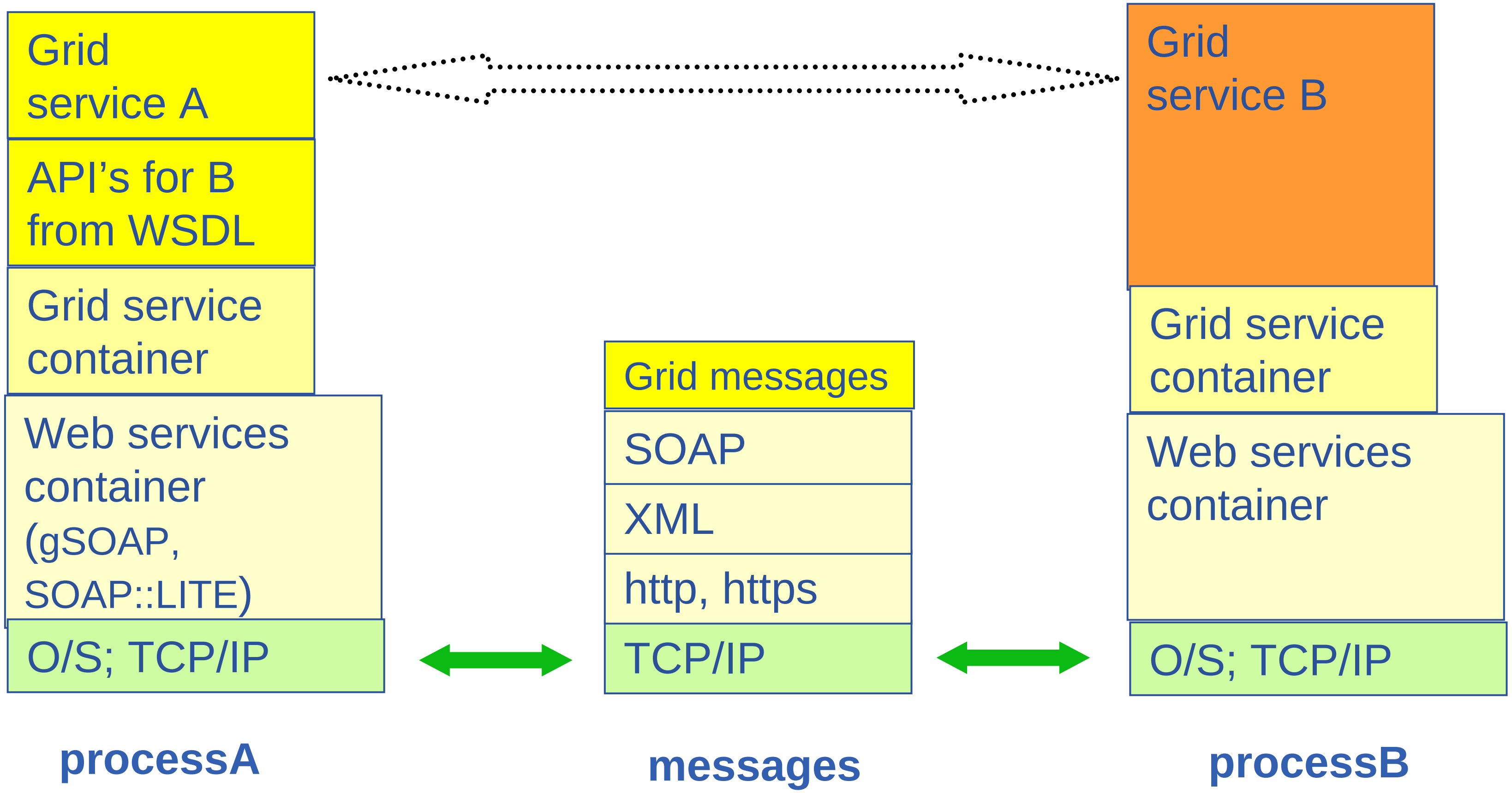
For A to use B, build process with library for B

Tight-coupling: to program A need code and detailed information about B.

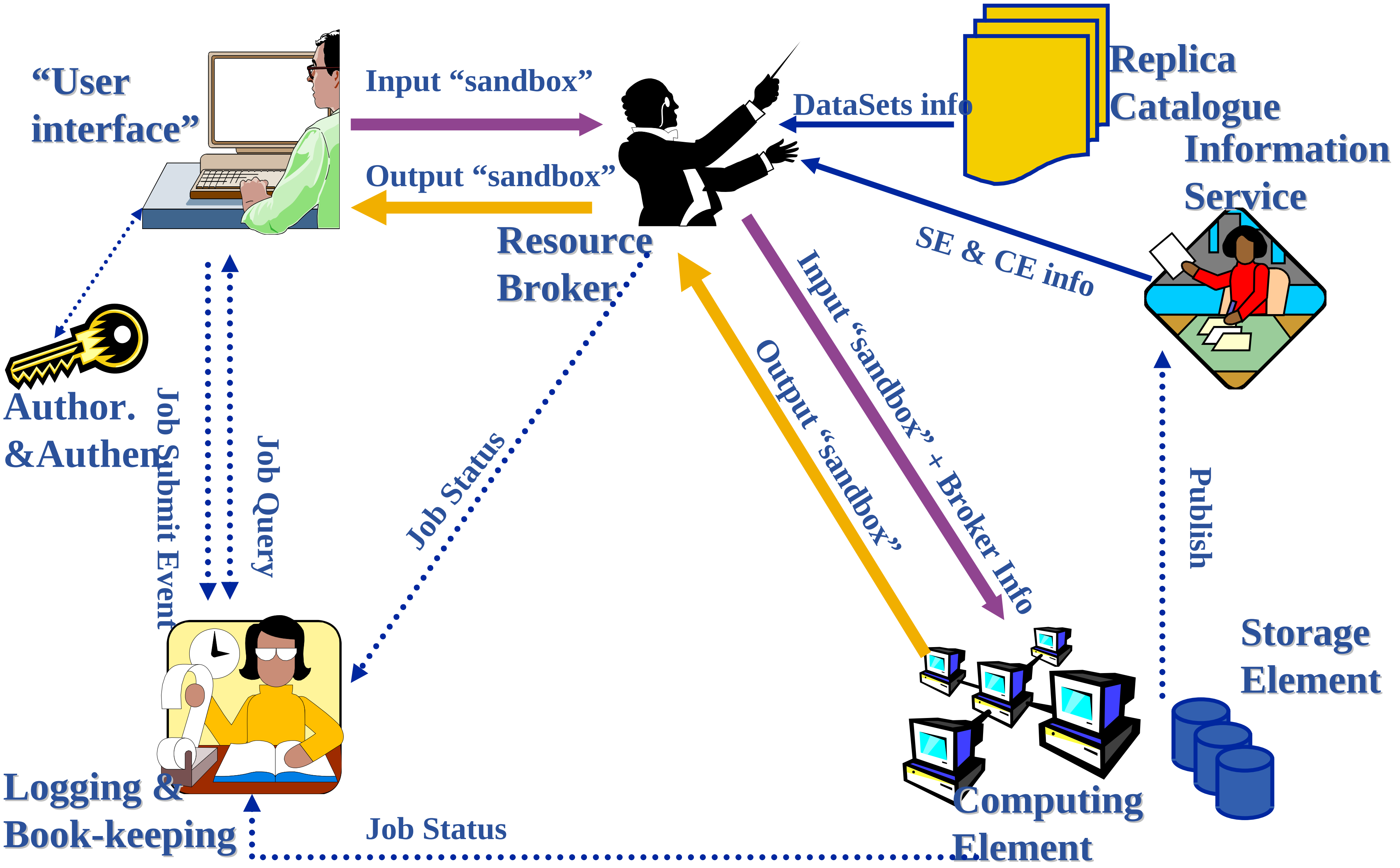


Grids with Service Orientation

- **Using service B from service A:**
  - From WSDL build APIs to use service
  - (Usually) use SOAP to access service

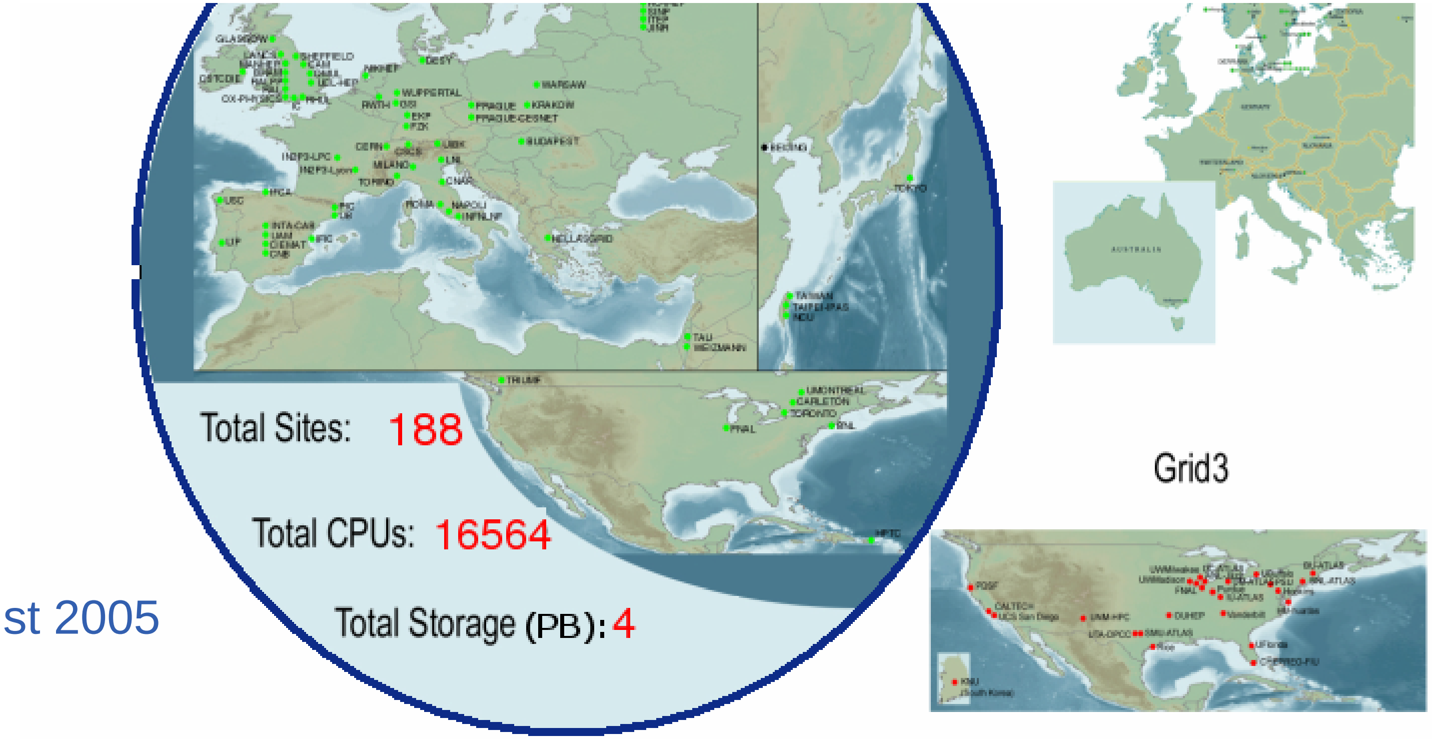


Current production m'ware: LCG-2



LCG-2

- **Real-time monitor**
  - http://www.hep.ph.ic.ac.uk/e-science/projects/demo/index.html
- **Current status**
  - http://goc.grid-support.ac.uk/gridsite/monitoring/



12 August 2005

gLite

gLite and computation

- **Jobs are:**
  - run from batch queues, termed “computing elements” CE’s
  - Described in “Job Description Language”
    - Slight modification from LCG
- **gLite also supports**
  - Interactive jobs
    - Jobs still run in batch mode – “listener” receives messages from CE
  - Parallelism using MPI
    - *run on CE’s that support MPI – not across administrative domains*
  - Workflow
  - Checkpointing

gLite and data

Simple data

- Files

- **Requires**
  - **Replica files**
    - Move data to computation
  - **Virtual filesystems**
  - **Metadata for files**
- **These services are provided in gLite**

Structured data

- RDBMS, XML databases

- Require ***extendable*** middleware tools to support
  - *computation near to data*
  - easy access, controlled by AA
  - integration and federation
- Hence **OGSA-DAI**
  DAI: Data Access and Integration
- **OGSA-DAI is NOT currently being ported to gLite**

EGEE middlewares face to face

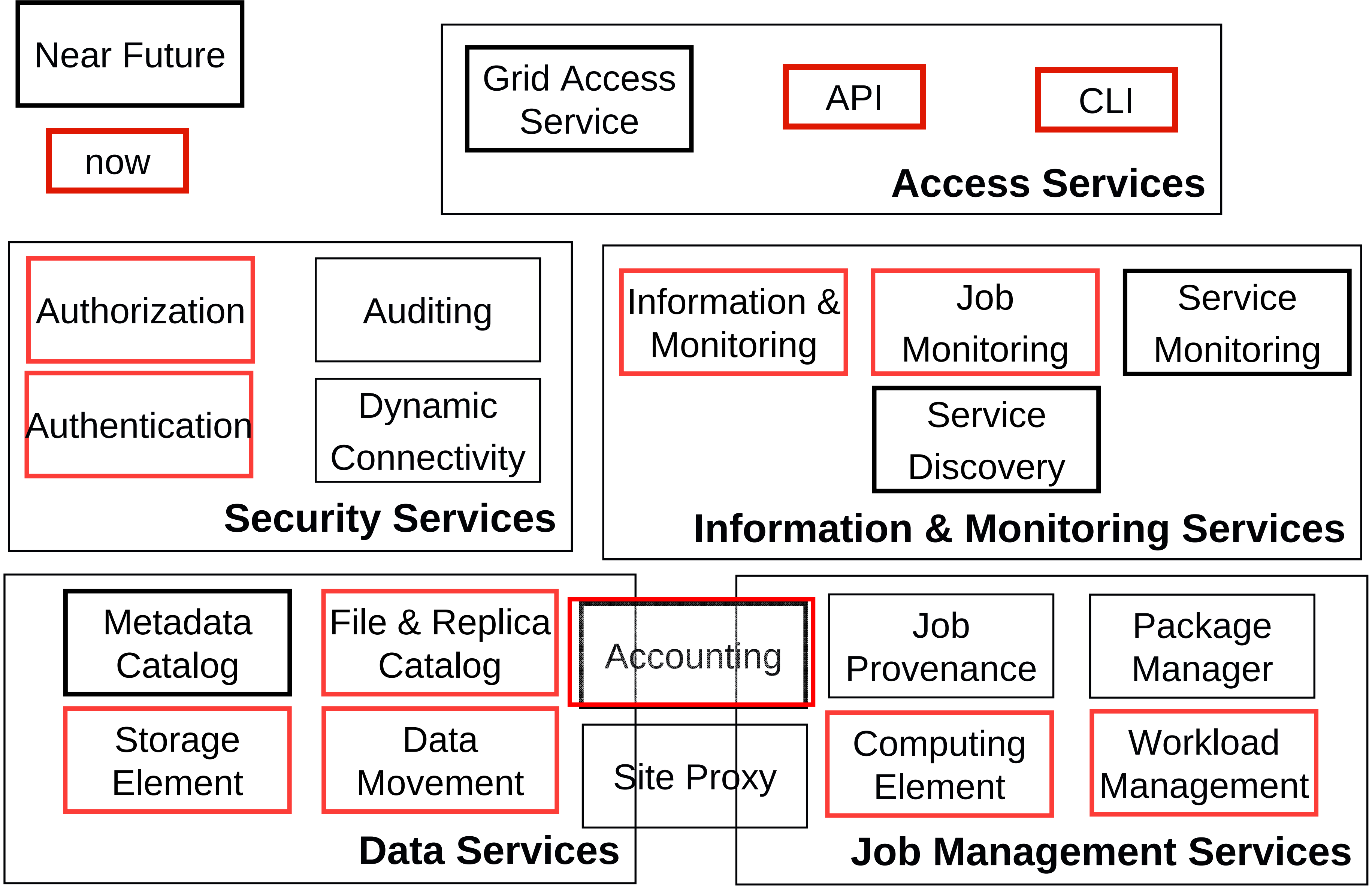
LCG

- **Security**
  - GSI
- **Job Management**
  - Condor + Globus
  - CE, WN
  - Logging & Bookkeeping
- **Data Management**
  - LCG services
- **Information & Monitoring**
  - BDII (evolution of MDS)
- **Grid Access**
  - CLI + API

gLite

- **Security**
  - GSI and VOMS
- **Job Management**
  - Condor + Globus + blahp
  - CE, WN
  - Logging & Bookkeeping
  - Job Provenance
  - Package management
- **Data Management**
  - LFC
  - gLite-I/O + FiReMan
- **Information & Monitoring**
  - BDII
  - R-GMA + Service Discovery
- **Grid Access**
  - CLI + API + Web Services

gLite components overview



Near Future

now

Grid Access Service

API

CLI

Access Services

Authorization

Auditing

Authentication

Dynamic Connectivity

Security Services

Information & Monitoring

Job Monitoring

Service Monitoring

Service Discovery

Information & Monitoring Services

Metadata Catalog

File & Replica Catalog

Accounting

Job Provenance

Package Manager

Storage Element

Data Movement

Site Proxy

Computing Element

Workload Management

Data Services

Job Management Services

Some definitions

- **Computing element**
  - A cluster of processors
  - Runs a Local Resource Management System
    - Batch jobs
- **Storage element**
  - Implements SRM interfaces (+... See later)
  - Mass Storage Systems –gLite has been tested with
    - CASTOR, using rfio
    - dCache, using dcap
    - DPM, using rfio

gLite status

- **Used in “pre-production” mode**
- **gLite v1.3 released 05/08/2005**
  - File Placement Service, File Placement Service clients added to UI and WNs modules
  - new data transfer agents including architecture refactoring to allow proper inter-VO scheduling
- **gLite v1.2 released 22/07/2005**
  - File Transfer Service and the File Transfer Agents
  - improvements in all modules.
- **gLite v. 1.1 released 13/05/2005**
  - File Transfer Service and the Metadata Catalog
- **gLite v. 1.0 released 05/04/2005**

- **http://www.glite.org/**

Summary

- **gLite, the EGEE middleware:**
  - Is exiting prototyping phase and entering real production phase
    - LHC first real data are only 2 years away from now!
  - Implements a full and complete stack of grid services

- **Service orientation allows**
  - Use gLite services all together or separately
  - Can migrate from LCG to gLite incrementally

- **Is seeking to balance**
  - Conforming to (emerging) standards
  - Need to deliver a production service that demands efficiency, speed

Further information

- EGEE http://public.eu-egee.org/
- gLite http://www.glite.org/
- EGEE Middleware Architecture https://edms.cern.ch/document/594698/