gLite Overview

Mike Mineter
National e-Science Centre, Edinburgh

Tokyo, 25 August 2005

Outline

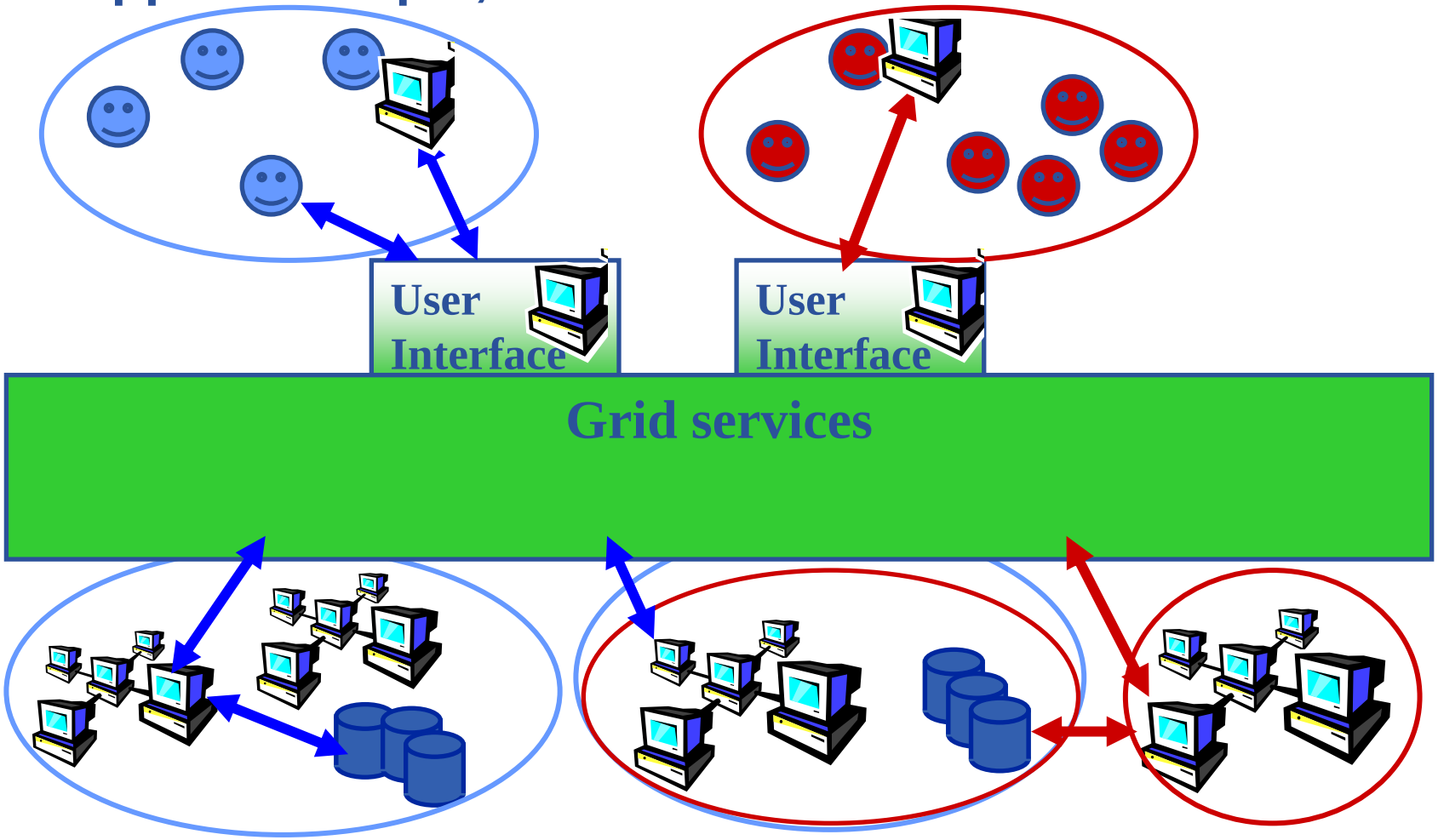
- **Grid concepts**
- **Background to gLite**
- **gLite services**
- **gLite status**

Additional information is found in hidden slides in the file that is available from the agenda page, http://agenda.cern.ch/fullAgenda.php?ida=a054533

Grid concepts

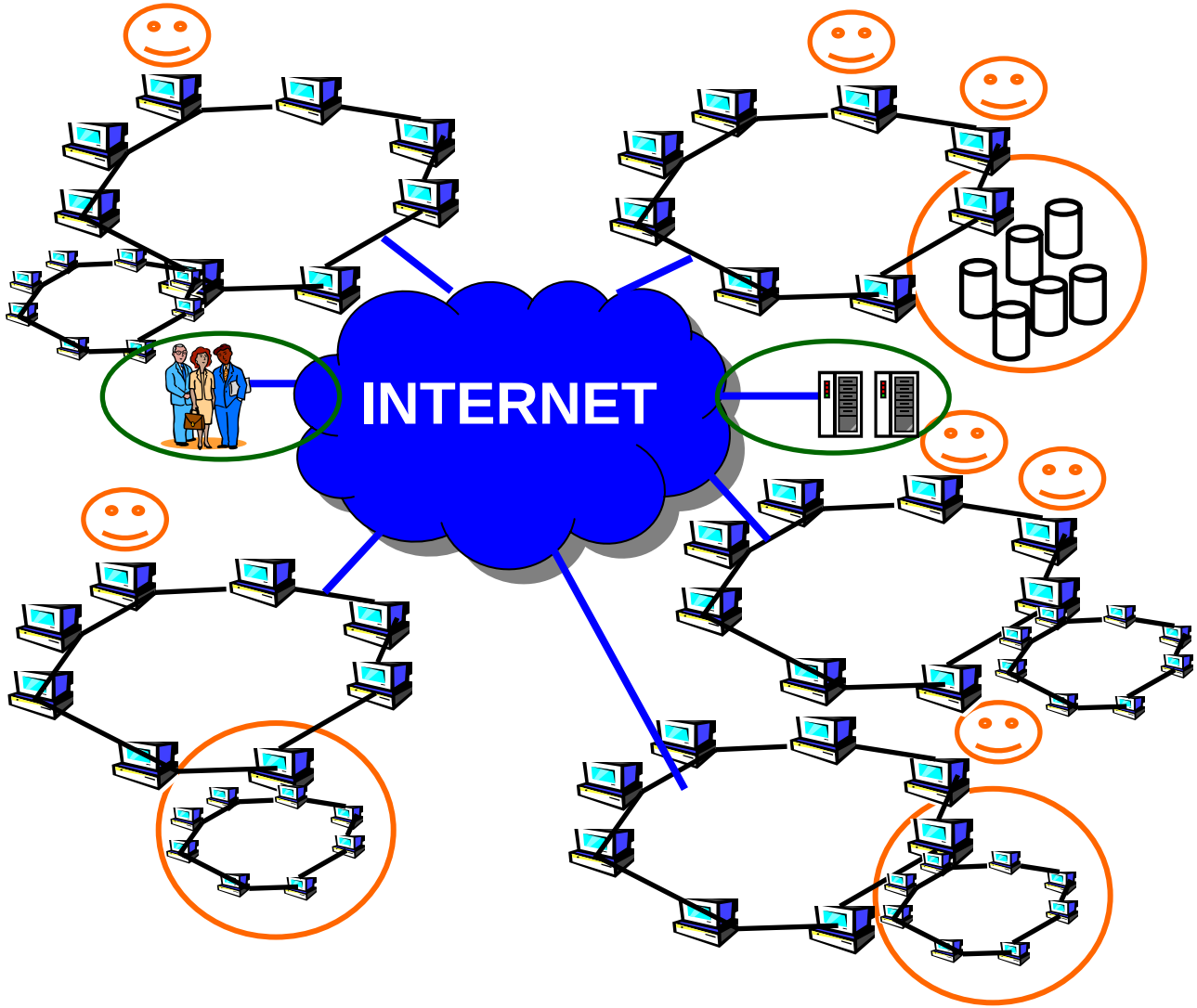
A multi-VO Grid

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



A multi-VO grid

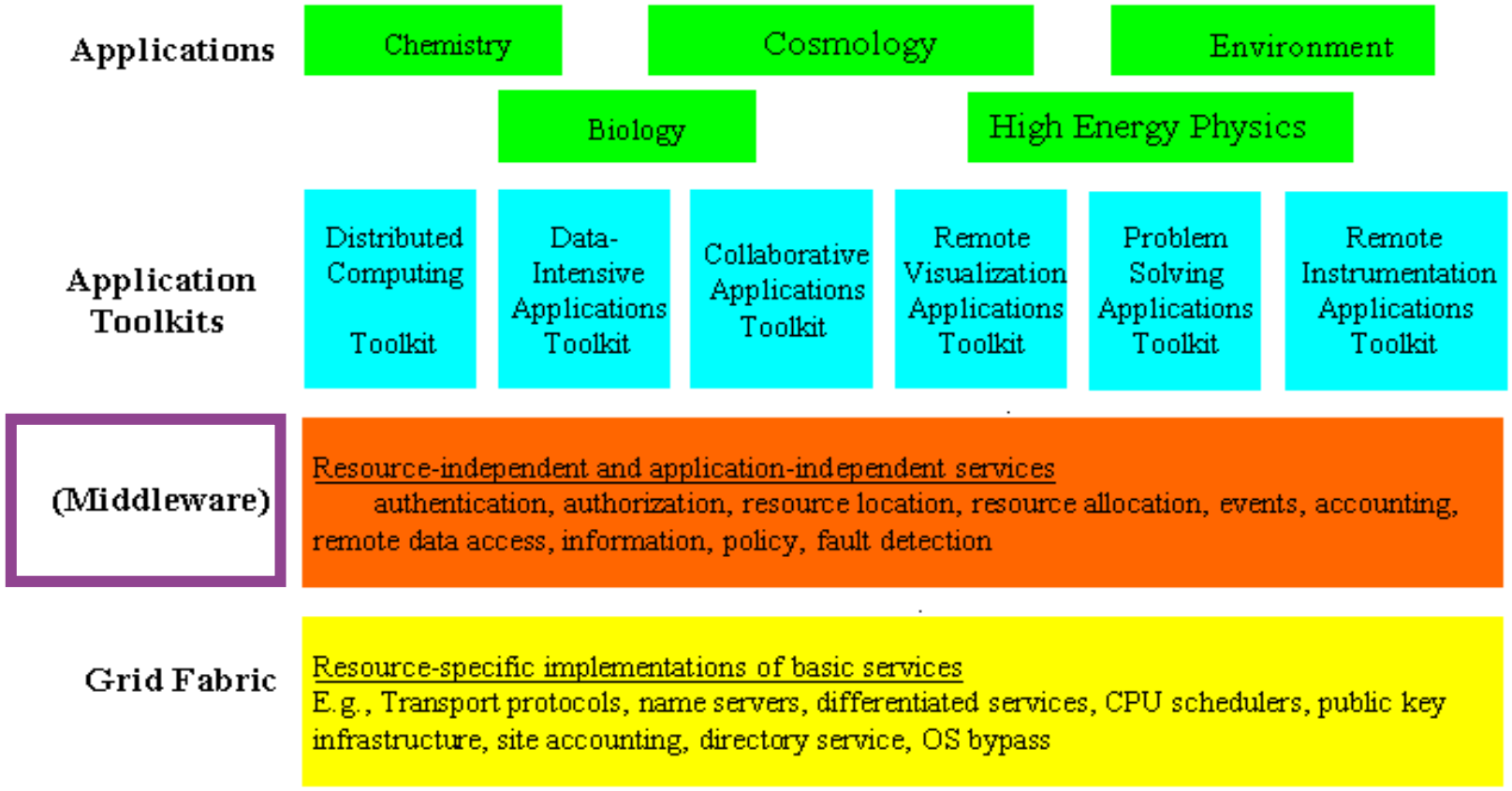
- Users join VO
- Virtual organisation contributes resources & negotiates access
- Grid middleware runs on each resource
  - "Storage elements"
  - "Compute elements"
- Additional services (both people and grid middleware) enable the grid
- Effect: "virtual computing" across administrative domains empowering collaboration



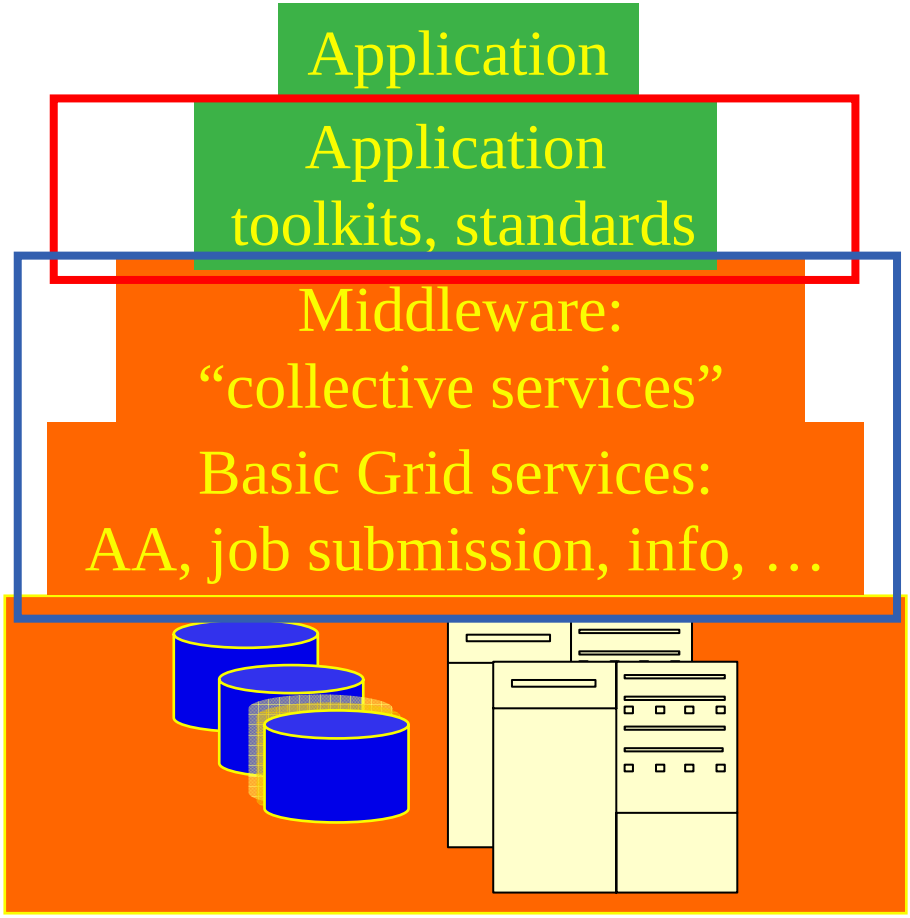
Key concepts

- **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 – "the operating system of a grid"**
  - 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



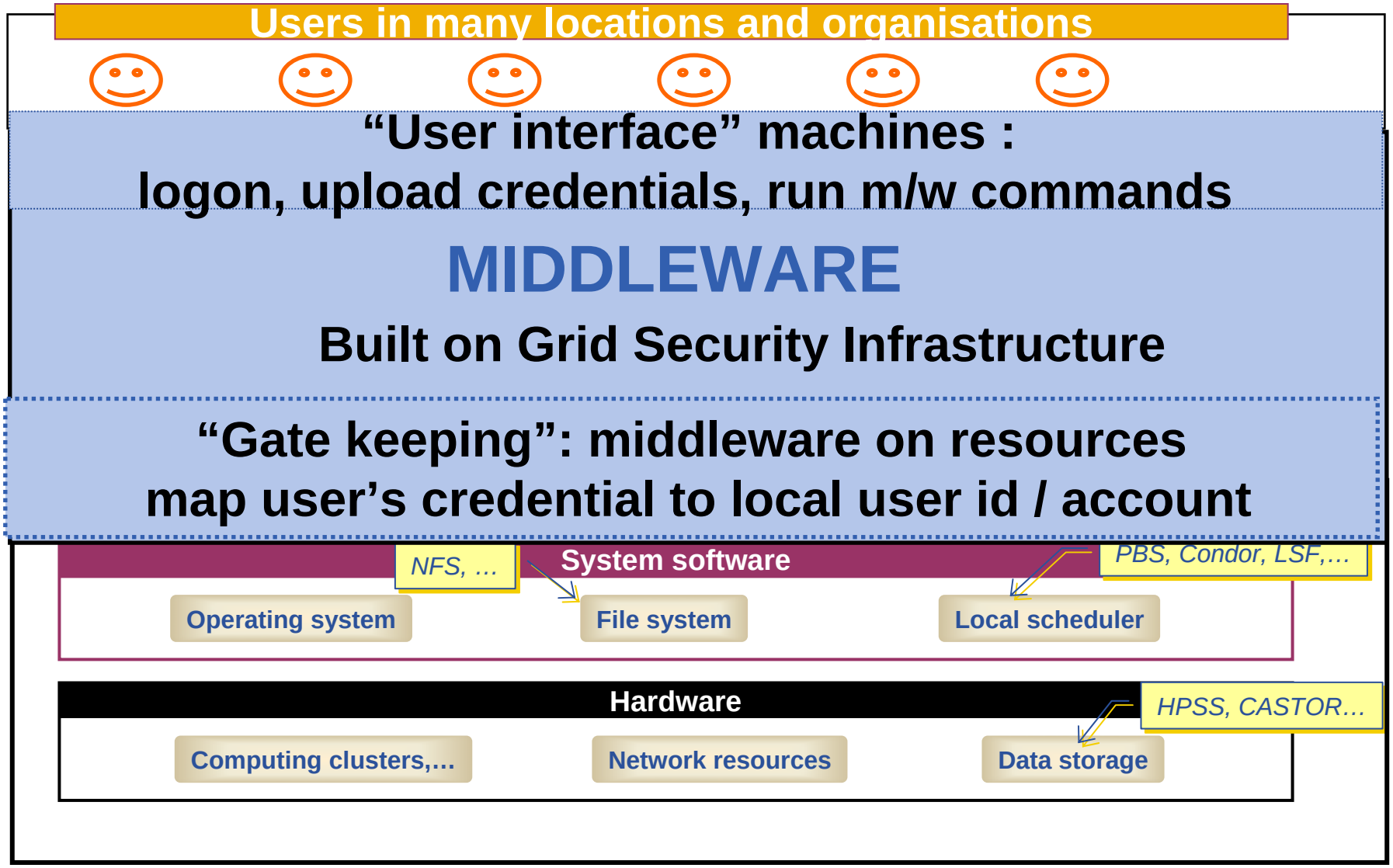
VO's and middleware



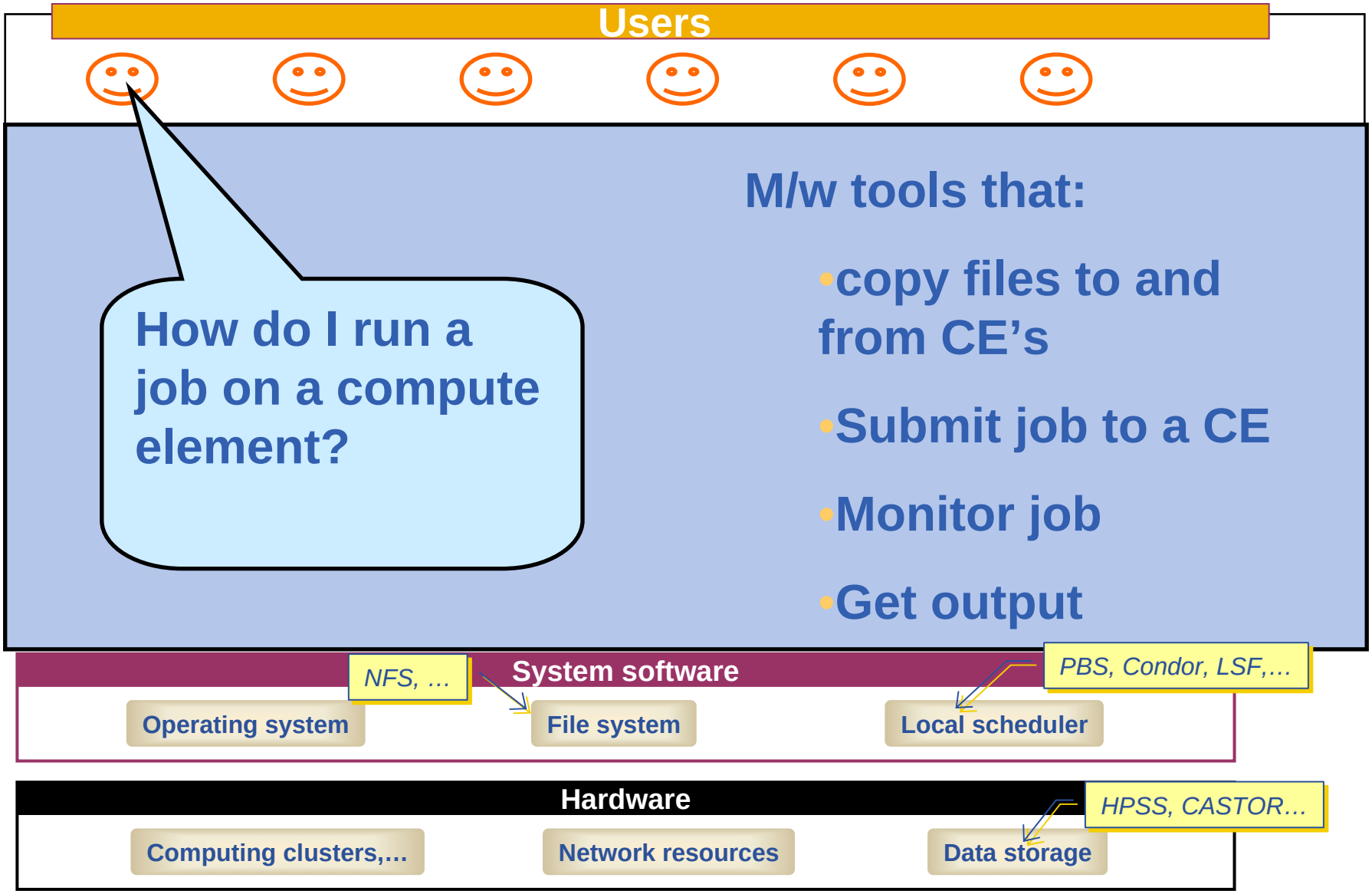
- The tools, services used by the VO's applications
- Community-specific standards
- Application development environment, portals, semantics, workflow
- In EGEE-1: Mainly VO-specific

- Emphasis of EGEE -1 middleware

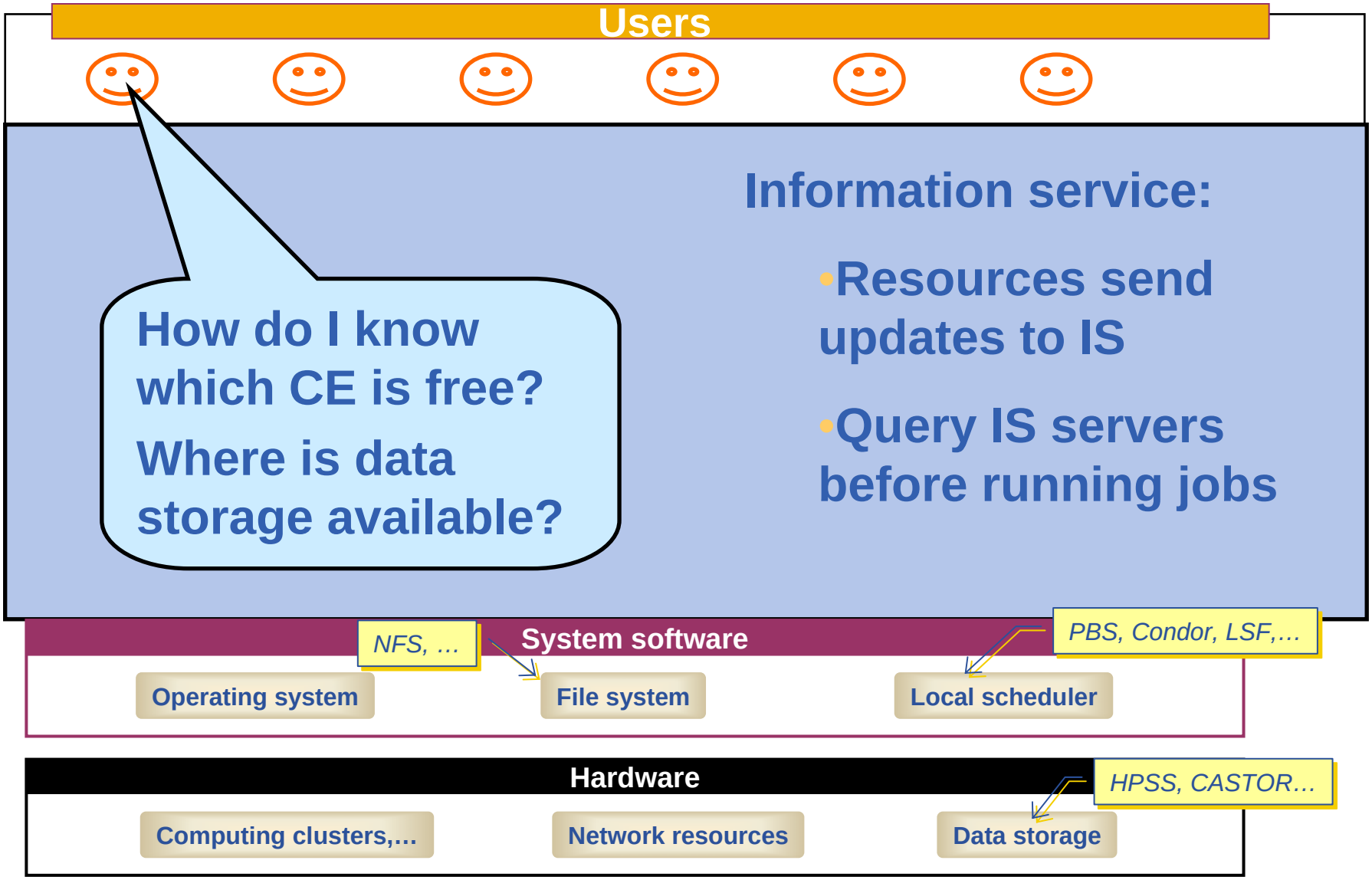
Authorisation, Authentication (AA)



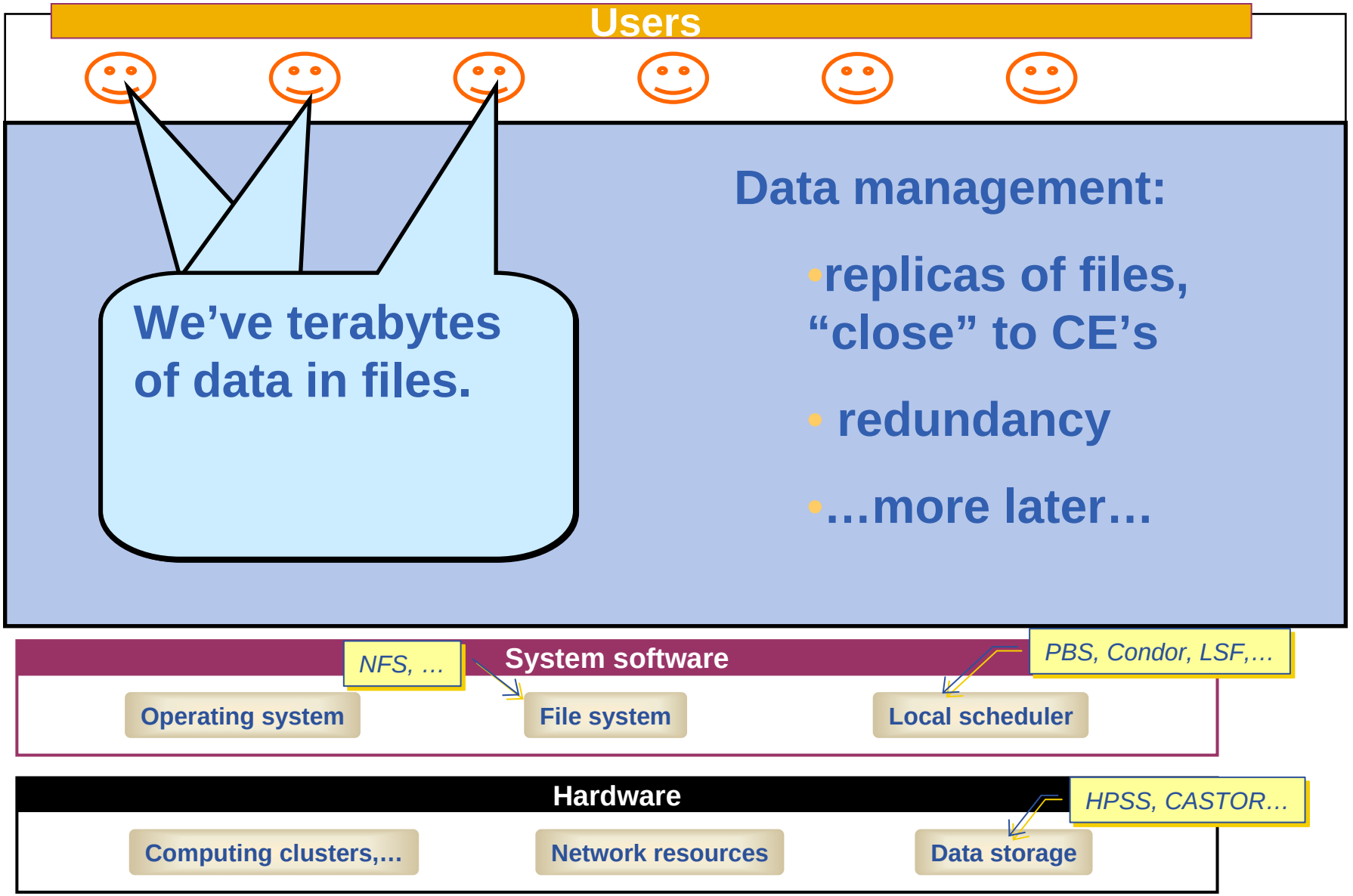
Basic job submission



Information service



File management



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 … so is gLite**
- **Globus Alliance: http://www.globus.org/**

Toolkit illustration

- **Command line interface to the tool for job submission – need to know name of a Compute Element**

```
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 tool to support job submission *to the grid*, not just to a named CE – and to make more friendly interfaces for users**

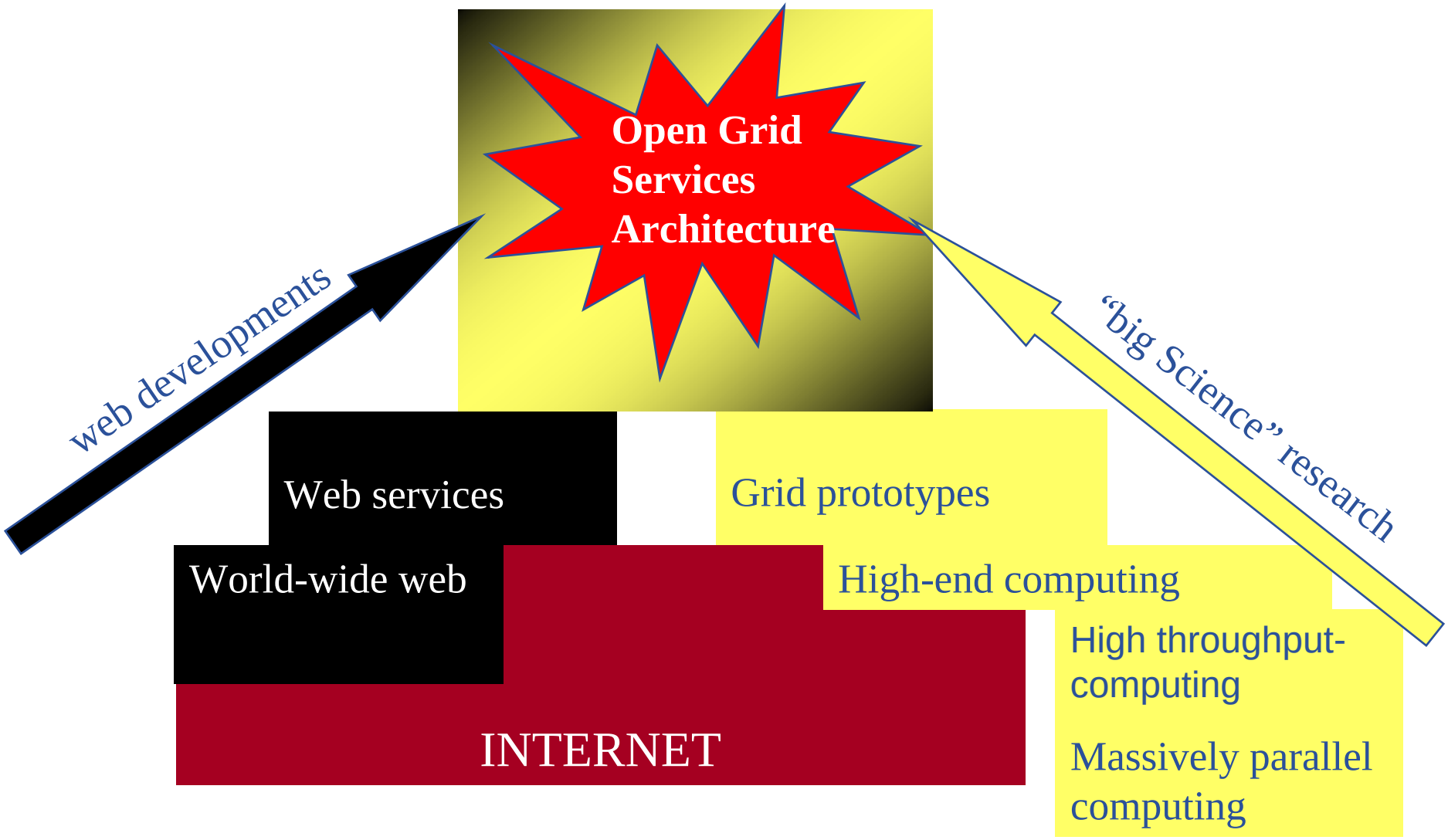
Background to gLite


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**
- **"Service orientation" seen as *the* way to build grids**

The vision of 2001: convergence of Web Services and Grids



Service Orientation

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

Service orientation

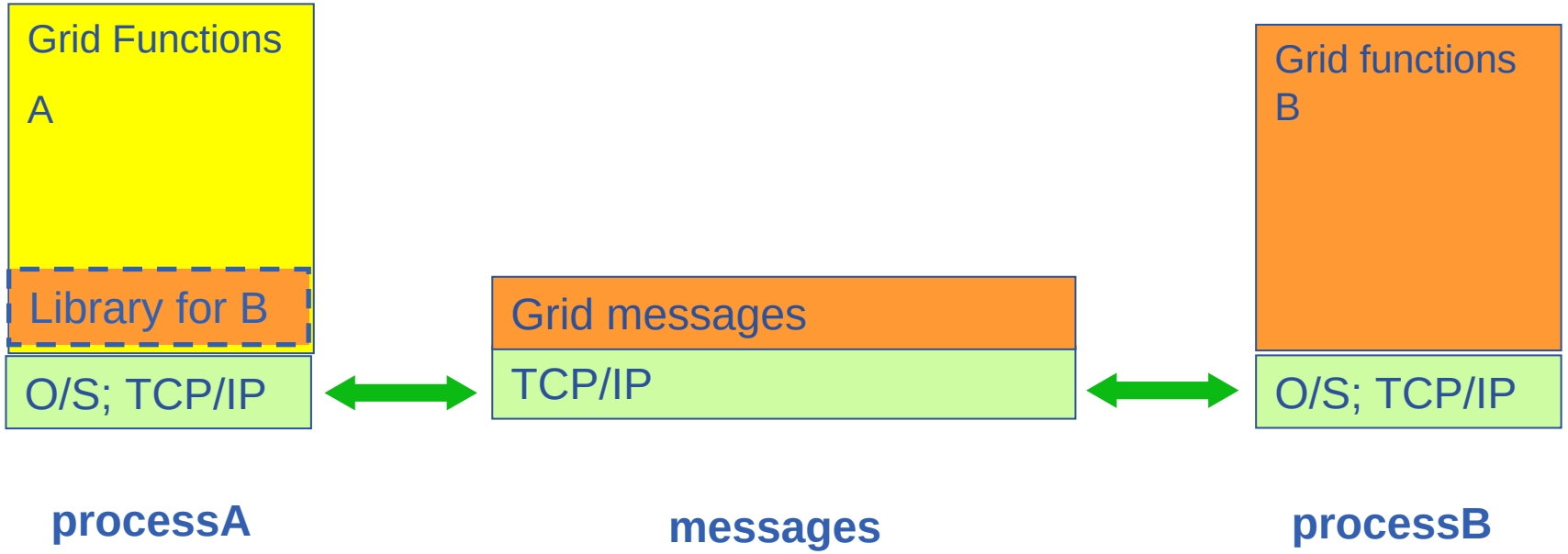
- **Components that are loosely coupled by messages**
  - Accessible across network; modular and self-contained; clean modes of failure
  - So can change implementation without changing interfaces
  - Can be developed in anticipation of new uses
- **… and are based on standards.**

- **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

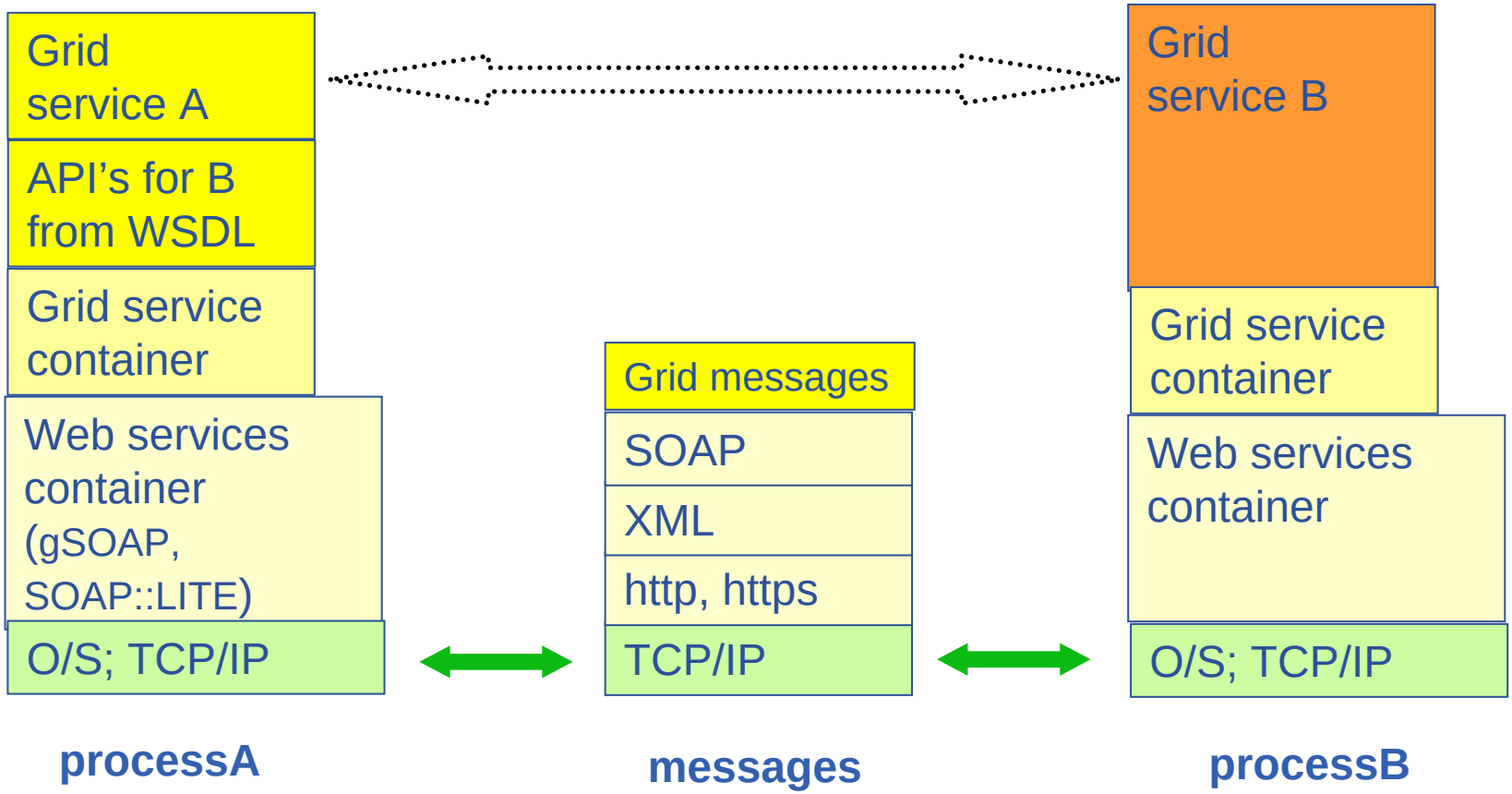
For A to use B, build process with library for B

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

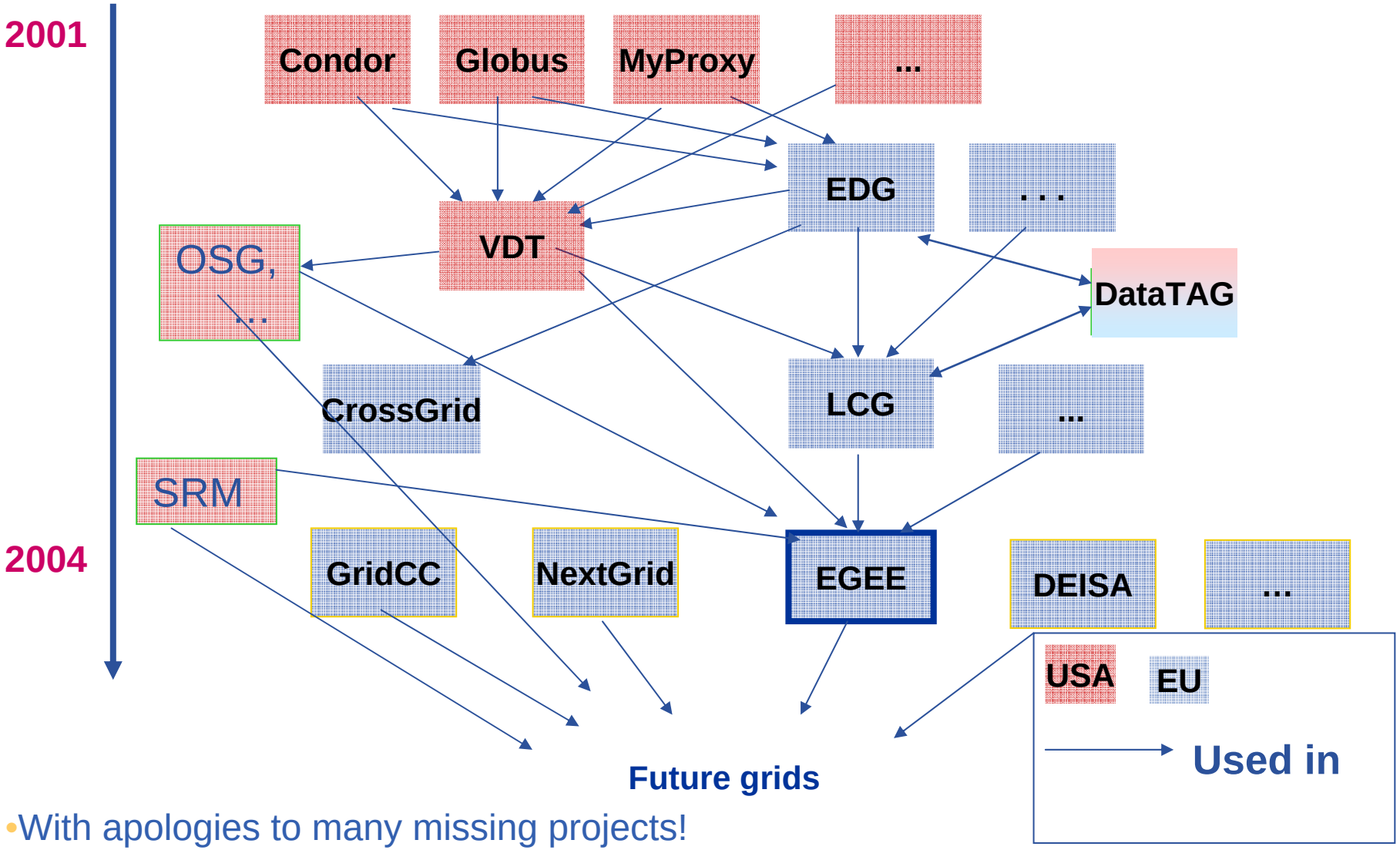


Grids with Service Orientation

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

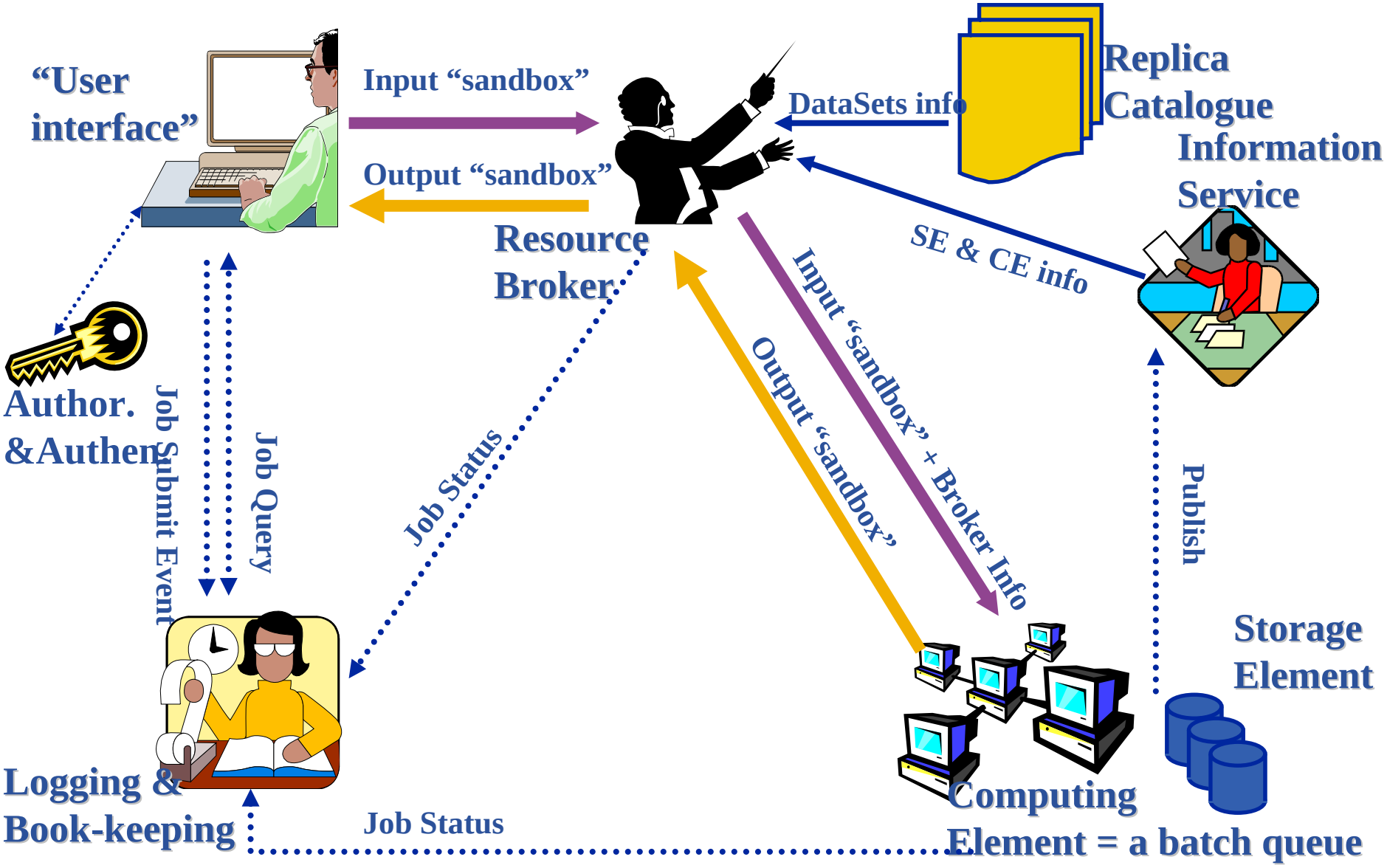


An EGEE-centric view of the Grid "ecosystem"



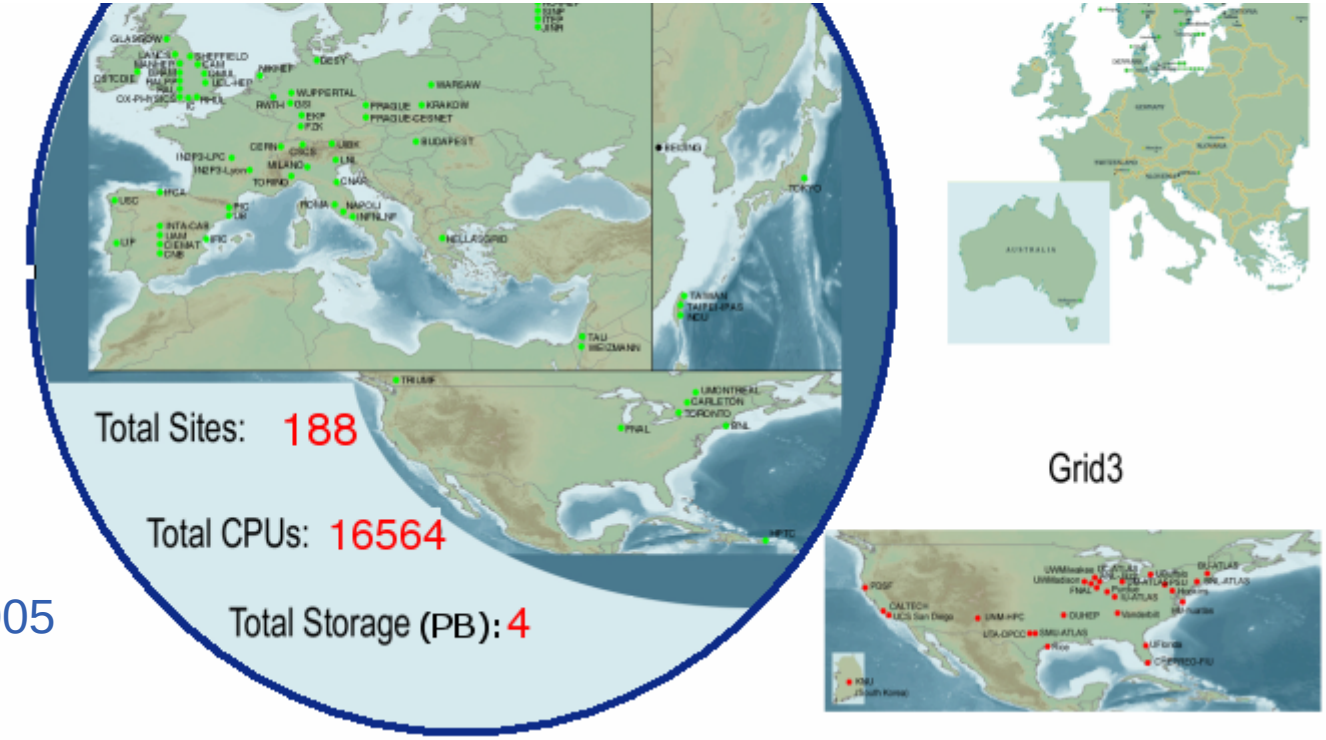
- With apologies to many missing projects!

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:**
  - (as in LCG) run from batch queues, termed “computing elements” CE’s
  - Described in “Job Description Language”
    - Slight modification from LCG
- **gLite also supports**
  - Interactive jobs
    - Jobs run in batch mode – “listener” receives messages from CE
  - Parallelism using MPI
    - *MPI jobs can run on CE’s that support MPI*
      not across administrative domains (not MPICH-G)
  - Workflow (DAGs, from Condor)
  - Checkpointing
  - Partitioned jobs (soon) – e.g. Monte-Carlo

gLite and data

Simple data

- Files

- **Requires**
  - **Replica files**
    - Move data to computation
  - **Virtual filesystems**
  - **Metadata for files**
  - **File transfer**

- **These services are amongst those 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

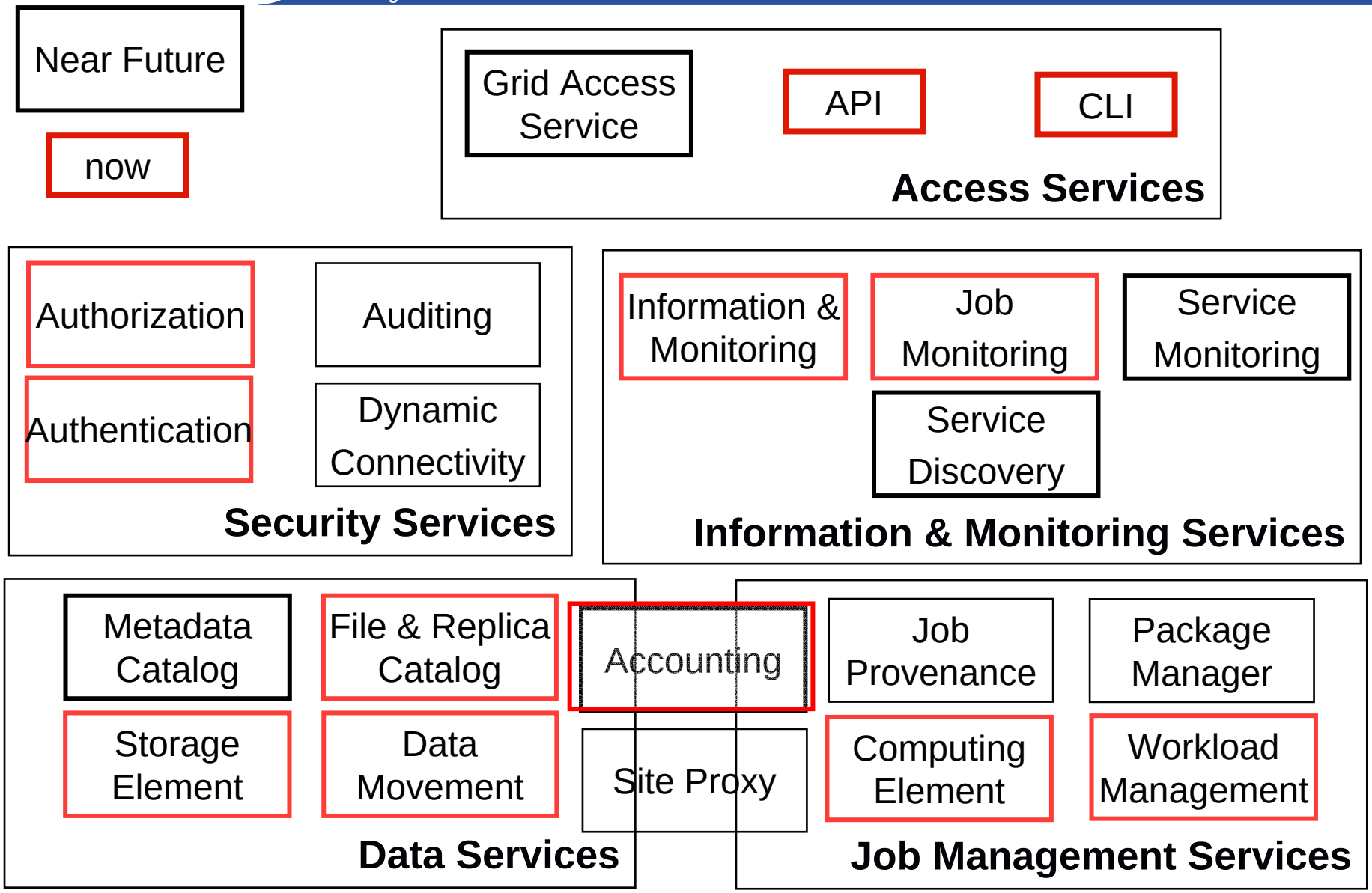
- **Security**
  - GSI and VOMS
- **Job Management**
  - Condor + Globus + blahp
  - CE, WN
  - Logging & Bookkeeping
  - Job Provenance
  - Package management
- **Data Management**
  - LFC
  - gLite-I/O + FiReMan
- **Information & Monitoring**
  - BDII
  - R-GMA + Service Discovery
- **Grid Access**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

BLAHP

- http://hepix.fzk.de/upload/lectures/BLAH_batch_system_intface.pdf

- The protocol
  - The BLAHP (Batch Local ASCII Helper Protocol) provides a set of plain ASCII commands used by Condor-C (and CREAM) to manage jobs on the batch systems.
- The daemon
  - BLAHPD implements the helper daemon responsible for converting BLAHP commands into batch system actions, interpreting their results and reporting them in BLAHP format.

gLite components overview



Some gLite definitions

- **Computing element**
  - A queue in a Local Resource Management System
    - Batch jobs that run on site's cluster
- **Storage element**
  - Implements SRM interfaces (+... See later)
  - gLite has been tested with
    - CASTOR
    - dCache
  - Being tested for DPM

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/