Grid in Alice – Status and Perspective

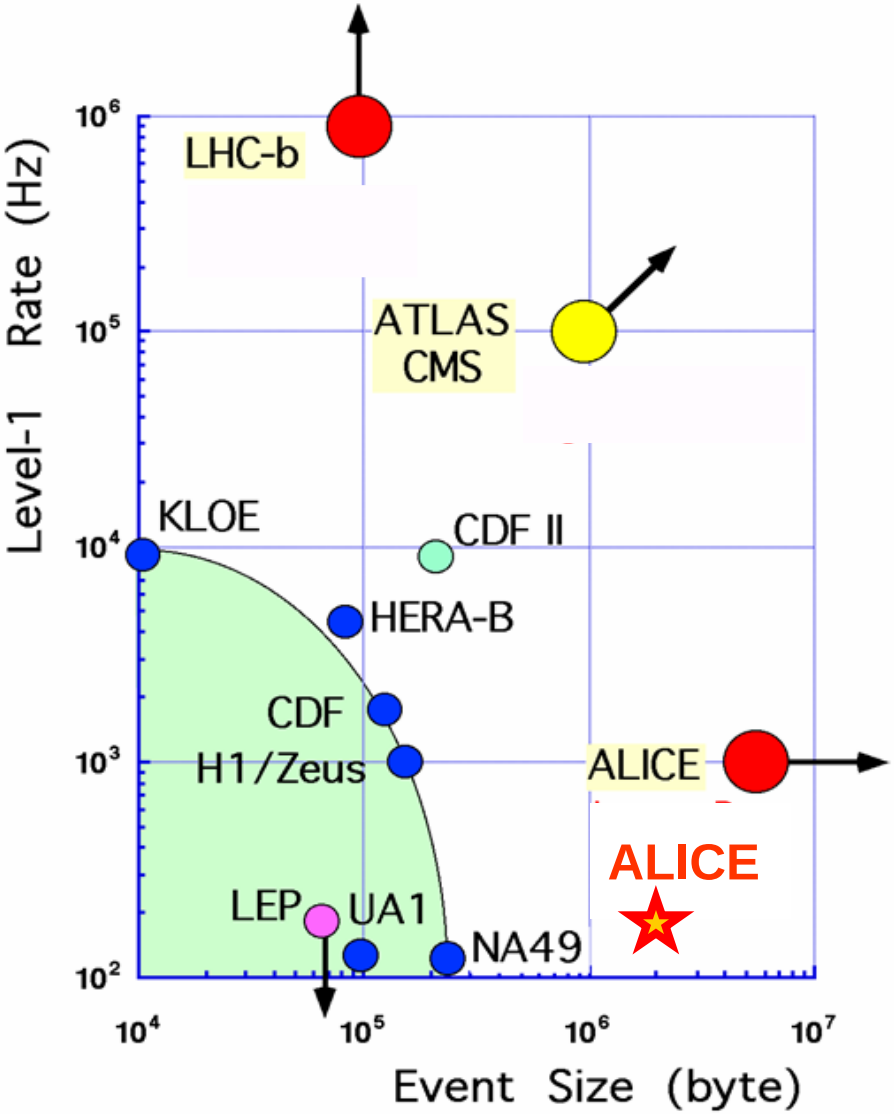
Predrag Buncic

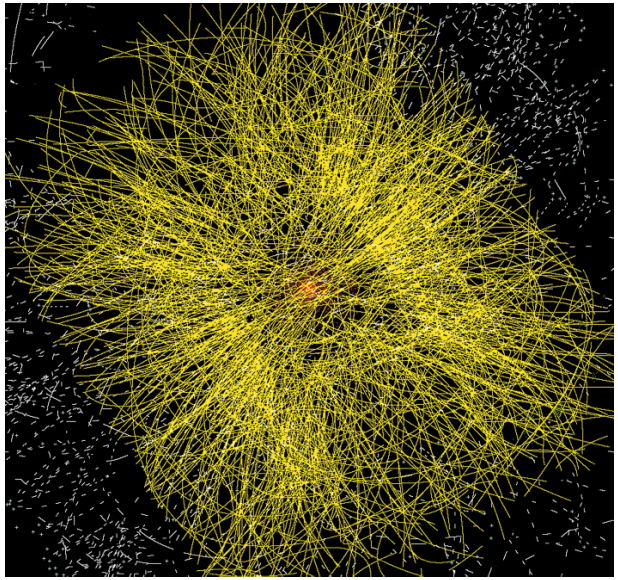
P.Saiz, A. Peters
C. Cristiou, J-F. Grosse-Oetringhaus
A. Harutyunyan, A. Hayrapetyan

Overview

- Grid in Alice
  - Dreaming about Grid (2001 – 2005)
  - Waking up (2005 – 2006)
  - Grid Future (2007+)
- Conclusions

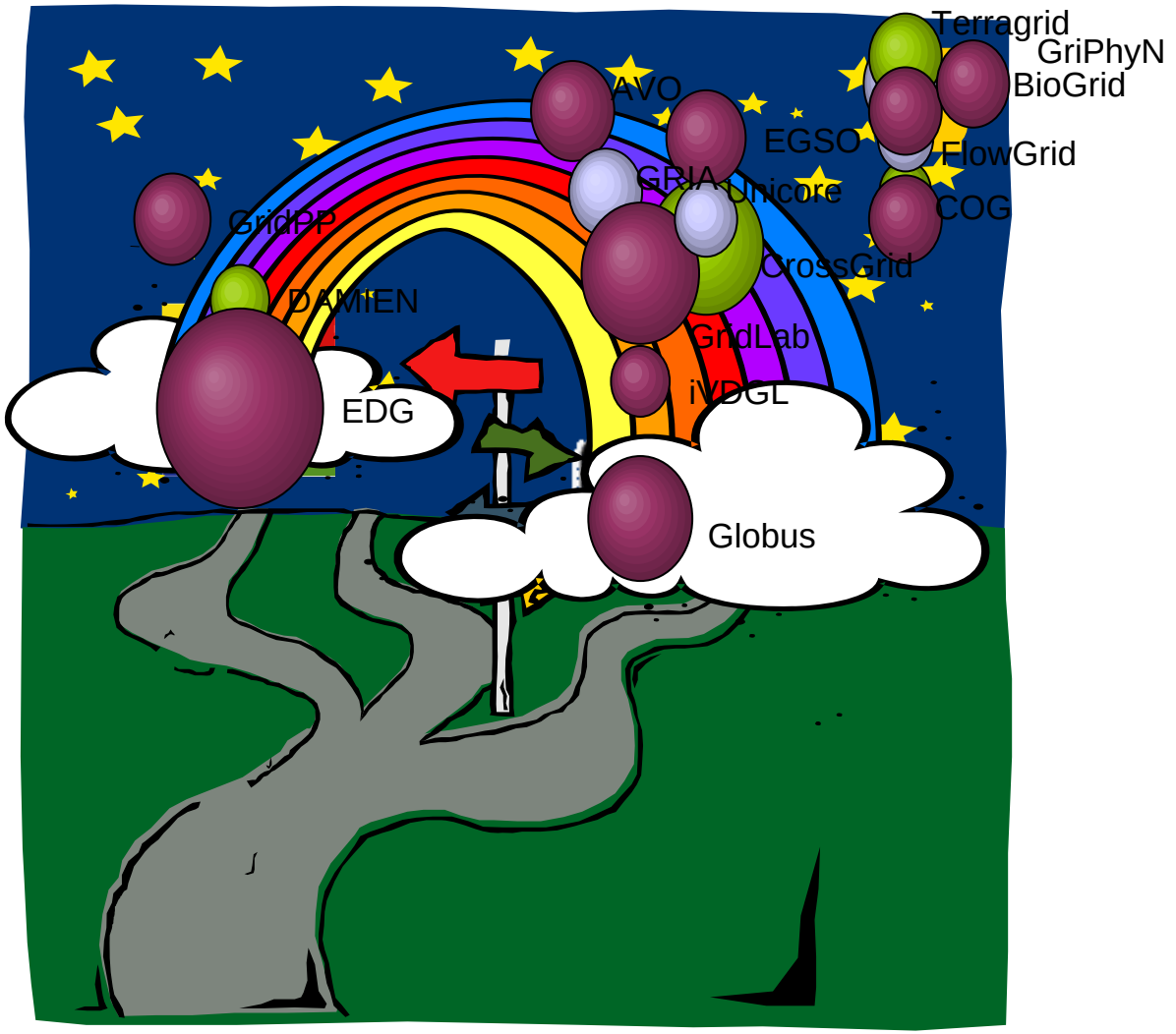
Need for Grid





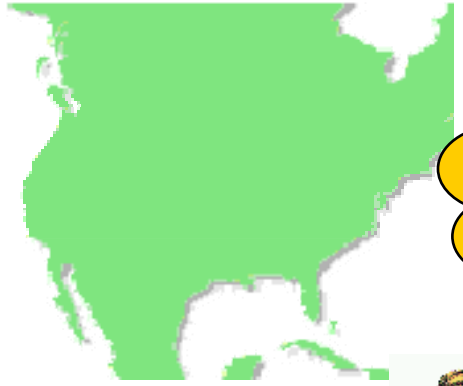
- 1.25 GB/sec
- 2 PB/year
- 8 h/event

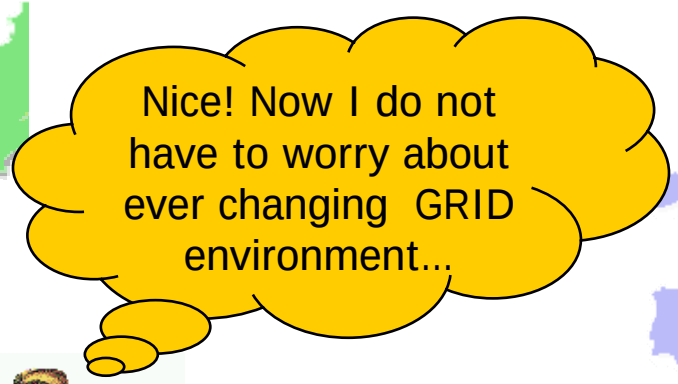
Five years ago…



Alice Environment @ Grid

User Interface		
VTD/OSG stack	AliEn stack	EDG stack





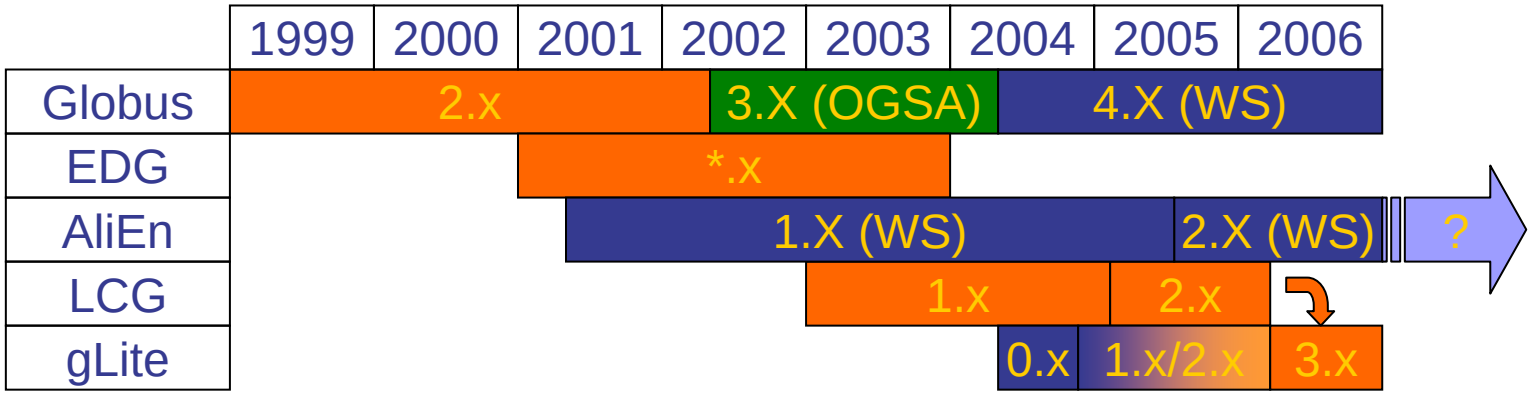
AliEn v1.0 (2001)

- New approach
  - Using standard protocols and widely used Open Source components
  - Interface to many Grids

- End-to-end solution
  - SOA (Service Oriented Architecture)
    - SOAP/Web Services (18)
      - Core Services (Brokers, Optimizers, etc
      - Site Services
      - Package Manager
      - Other (non Web) Services (ldap, database proxy, posix I/O)
  - Distributed file and metadata catalogue
  - API and a set of user interfaces

- Used as production system for Alice since end of 2001
  - Survived 5 software years

Technology matrix

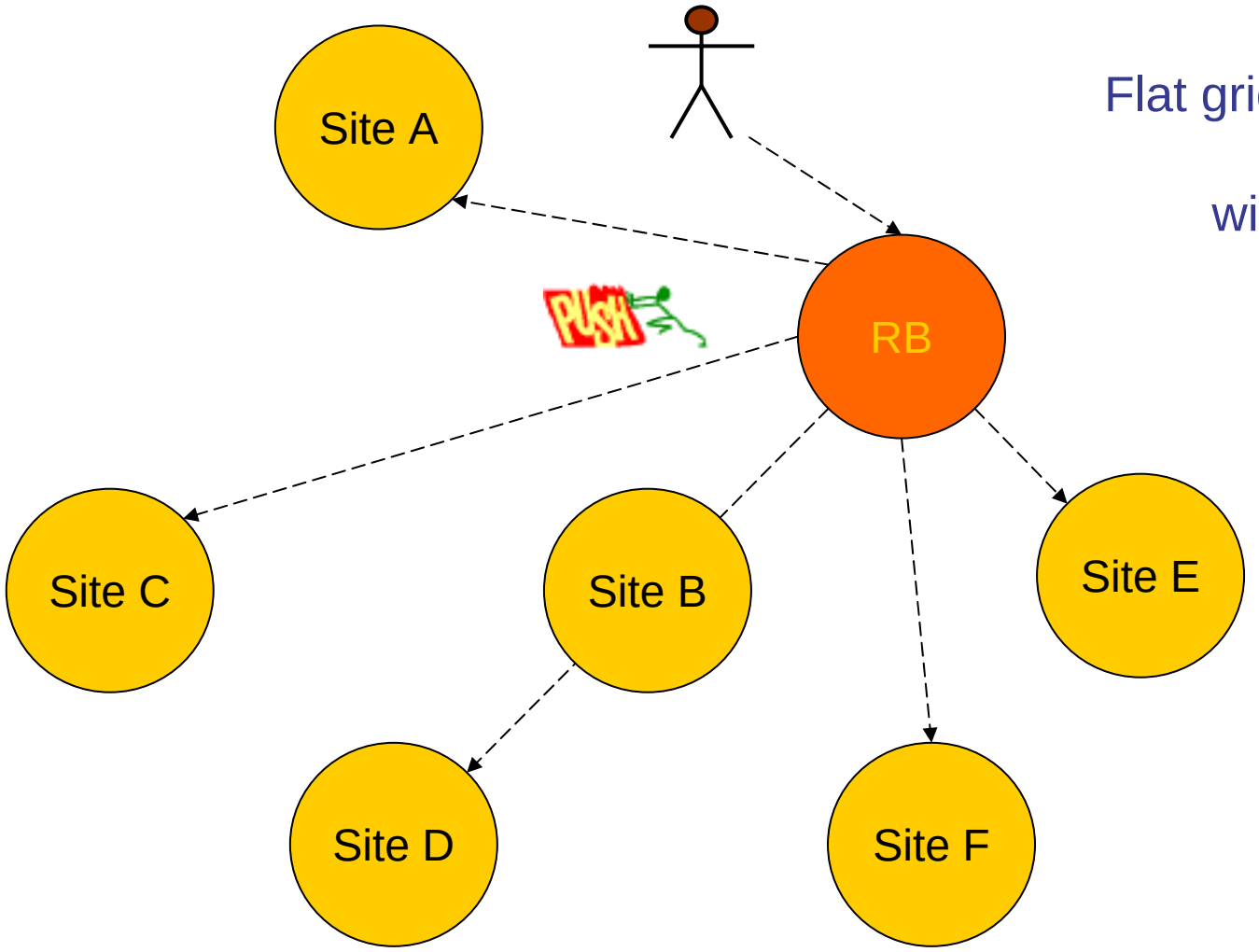


Proprietary protocol (Globus)

OGSA — Open Grid Services Architecture (Globus)

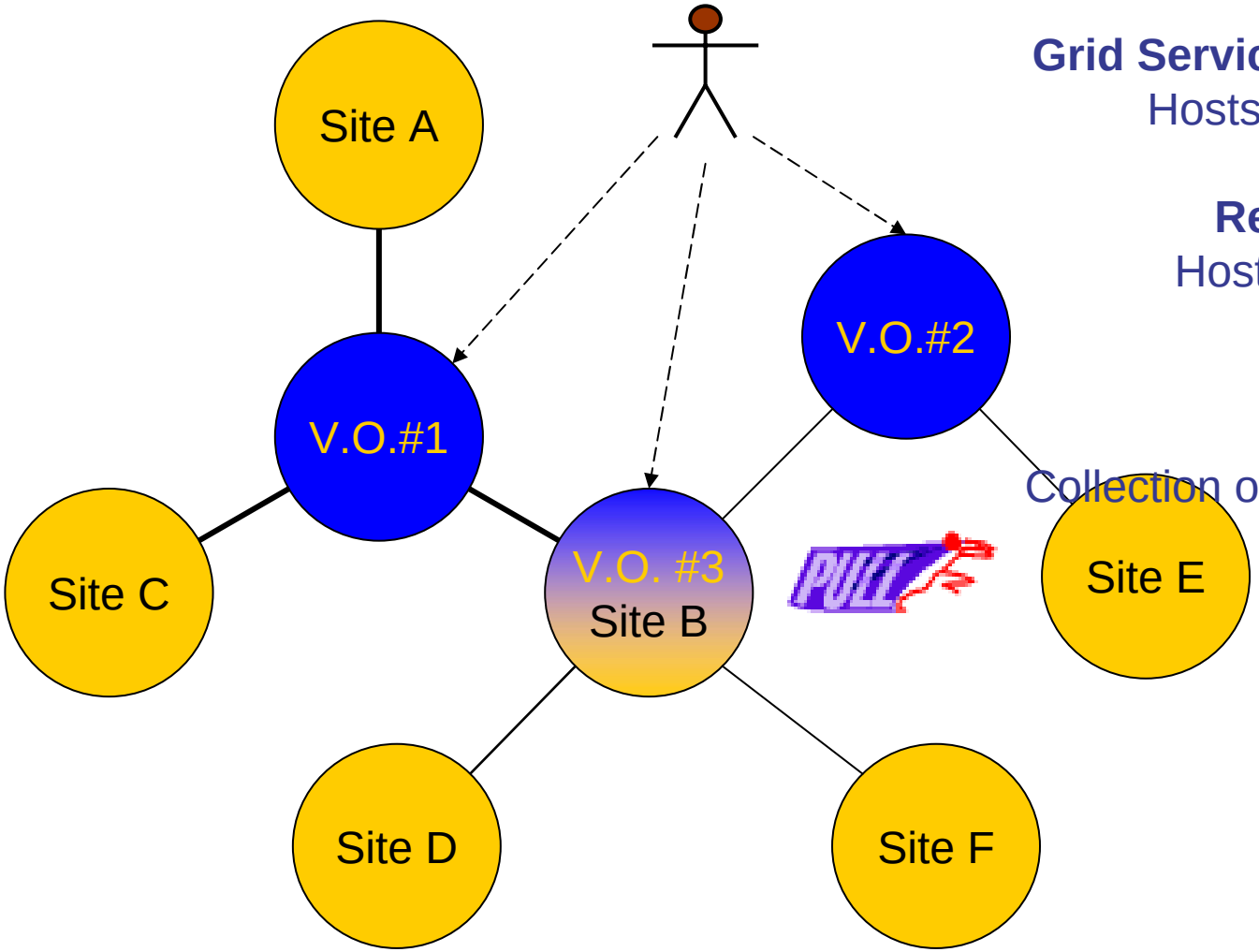
WS — Web Services (W3)

Before: Globus model



Flat grid, each user interacts directly
with site resources

AliEn model

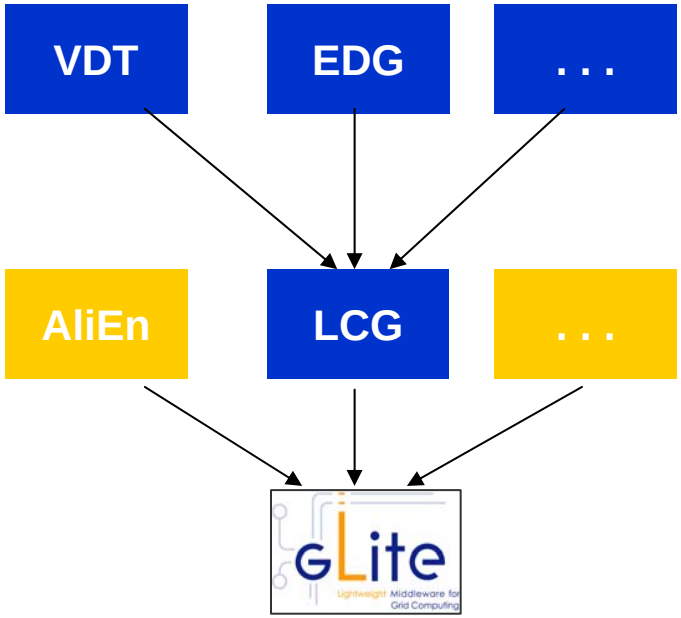


Grid Service Provider (Supersite):
Hosts Core Services (per V.O)

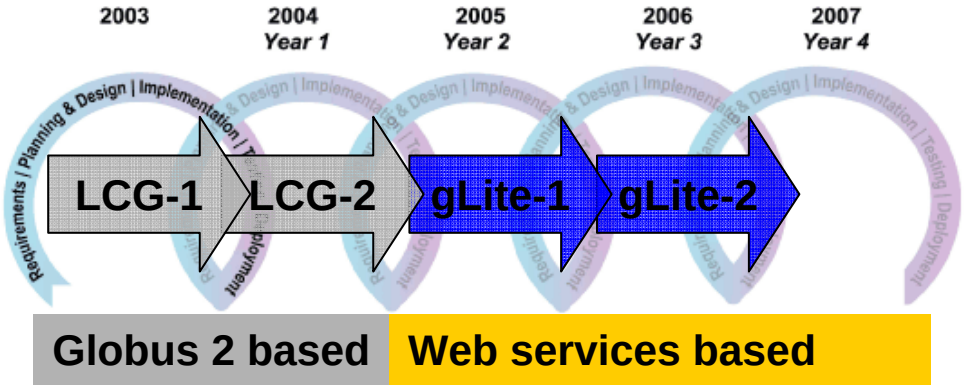
Resource Provider (Site):
Hosts an instance of CE, SE Services (per V.O.)

Virtual Organisation:
Collection of Sites, Users & Services

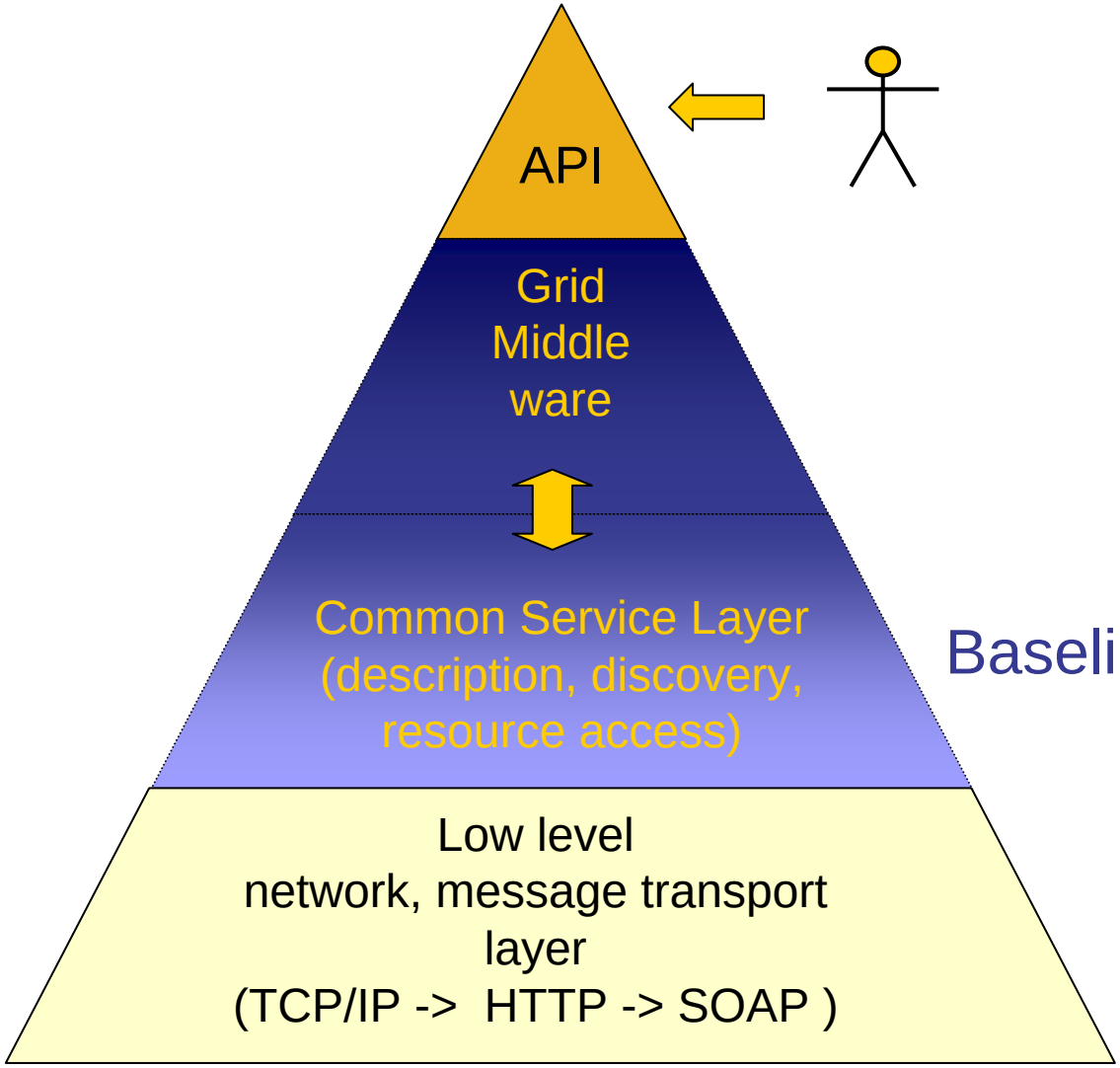
gLite: Initial goals



Re-engineer and harden Grid middleware (AliEn, EDG, VDT and others)
Provide production quality middleware



Reducing M/W Scope



Baseline Services

Alien v2.0

- Implementation of gLite architecture
  - gLite architecture was derived from AliEn

- New API Service and ROOT API
  - Shell, C++, perl, java bindings

- Analysis support
  - Batch and interactive
  - ROOT/PROOF interfaces
  - Complex XML datasets and tag files for event level metadata
  - Handling of complex workflows

- New (tactical) SE and POSIX I/O
  - Using xrootd protocol in place of aiod (glite I/O)
- Job Agent model
  - Improved job execution efficiency (late binding)

Distributed Analysis

File Catalogue Query

User Job (many events)

Data set (ESD's, AOD's)

Job Output

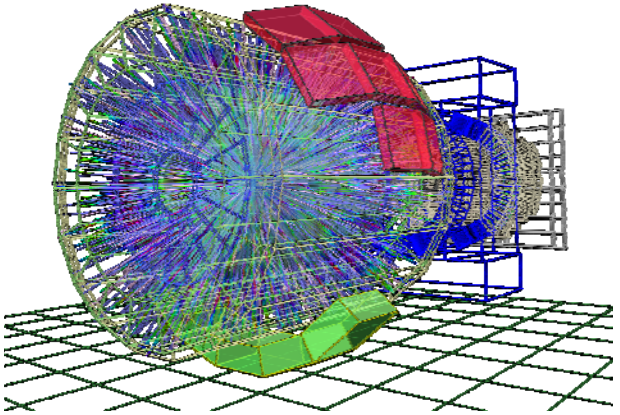
Job Optimizer

Sub-job 1

Sub-job 2

Sub-job n

Grouped by SE files location



Job Broker

Submit to CE with closest SE

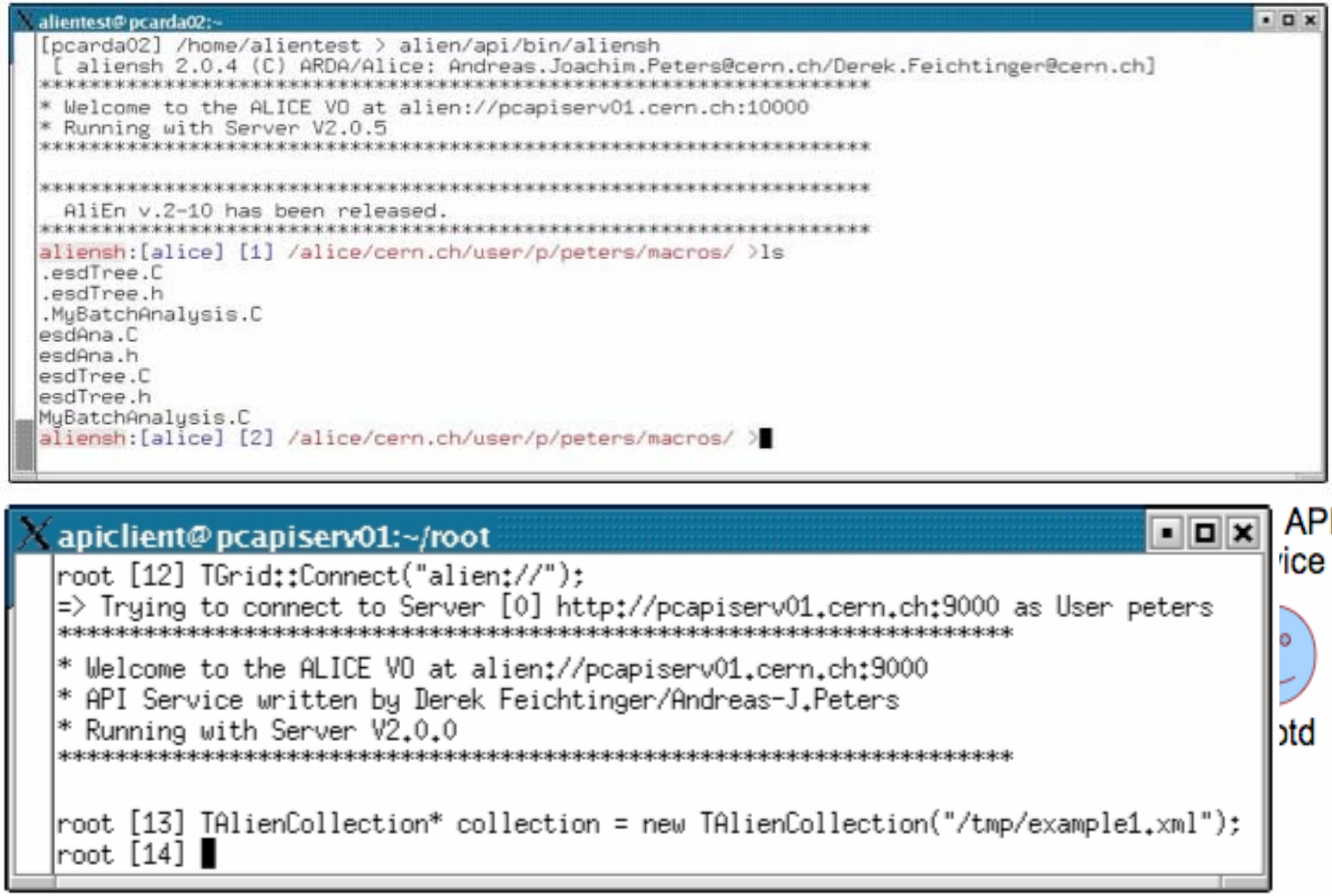
CE and SE Processing Output file 1

CE and SE Processing Output file 2

CE and SE Processing Output file n

File-merging Job

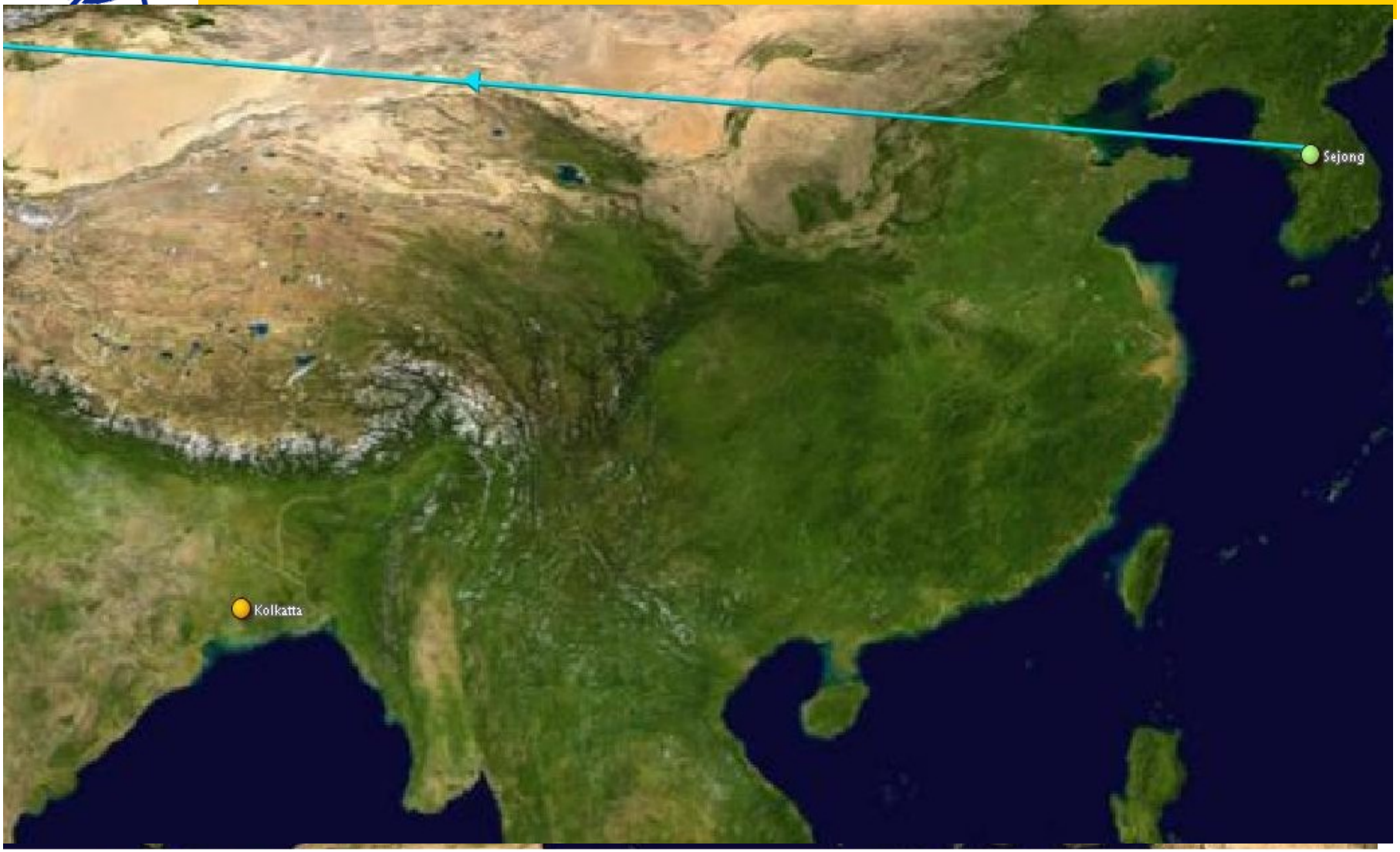
ROOT / AliEn UI



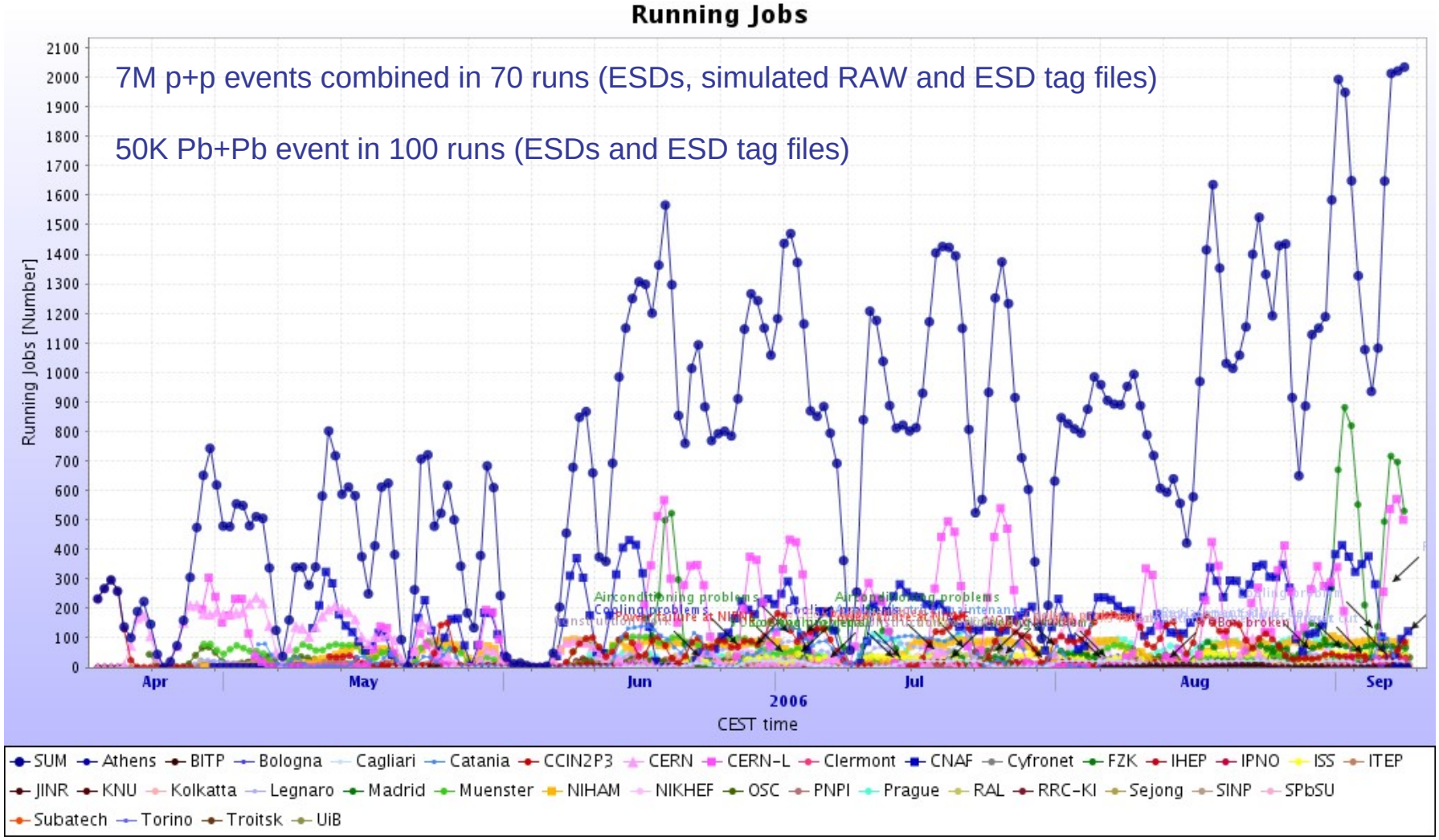
Reality check: PDC'06

1. Production of MC events for detector and software performance studies
2. Verification of the ALICE distributed computing model
   - Integration and debugging of the GRID components into a stable system
     - LCG Resource broker, LCG file catalogue, File transfer system, Vo-boxes
     - AliEn central services – catalogue, job submission and control, task queue, monitoring
   - Distributed calibration and alignment framework
   - Full data chain
     - RAW data from DAQ, registration in the AliEn FC, first pass reconstruction at T0, replication at T1
   - Computing resources
     - verification of scalability and stability of the on-site services and building of expert support
   - End-user analysis on the GRID

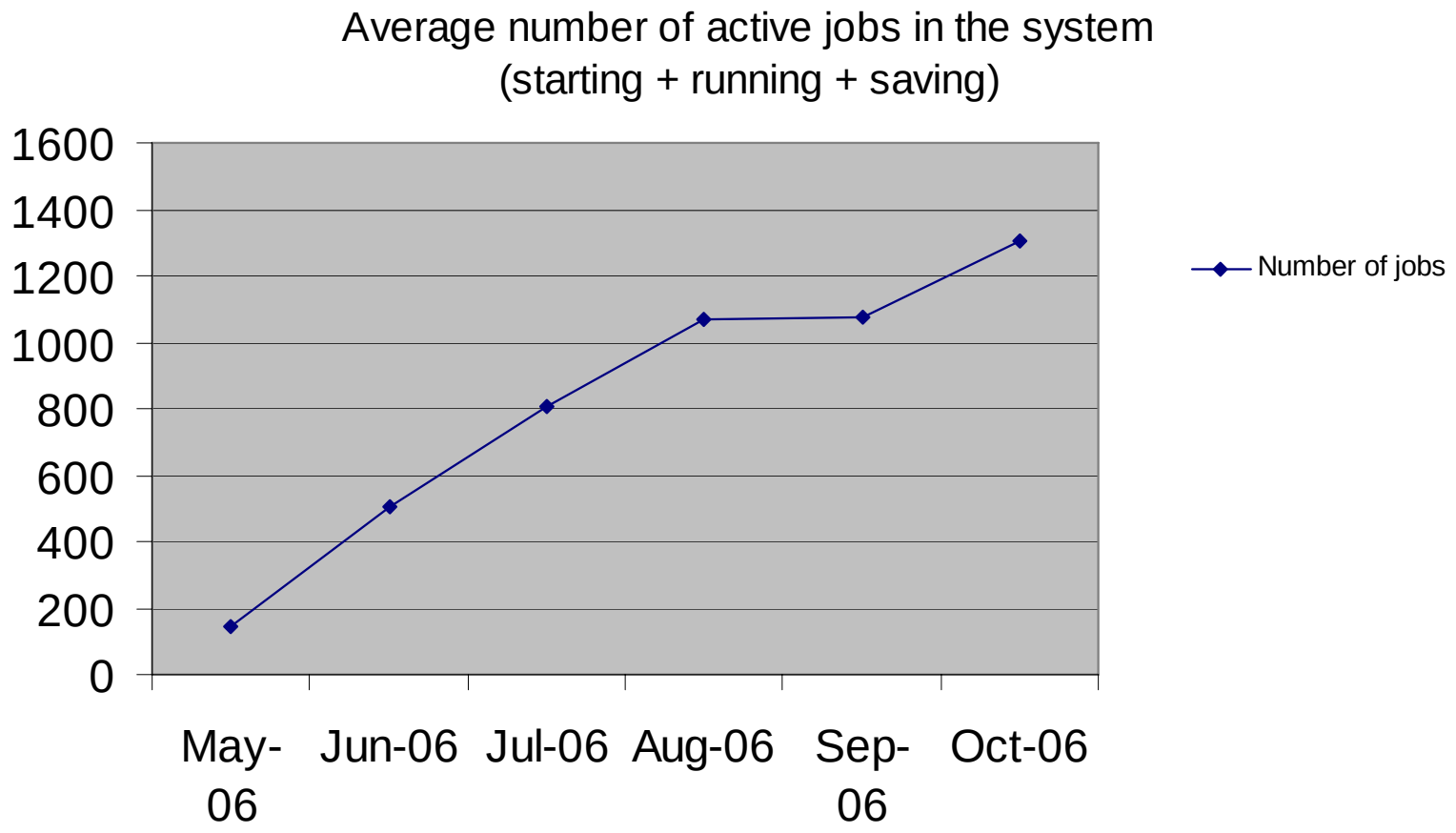
ALICE sites on the world map



History of running jobs

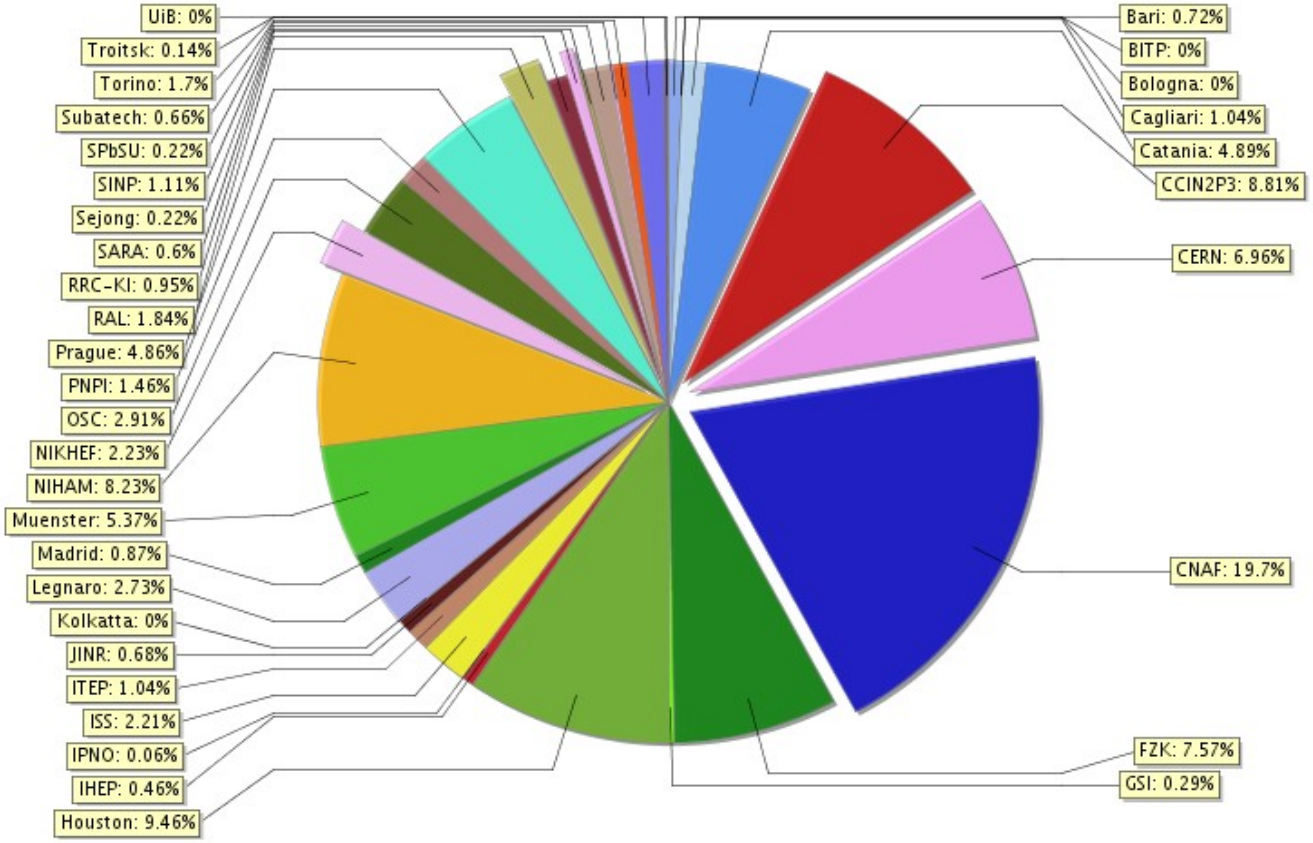


Concurrently running jobs



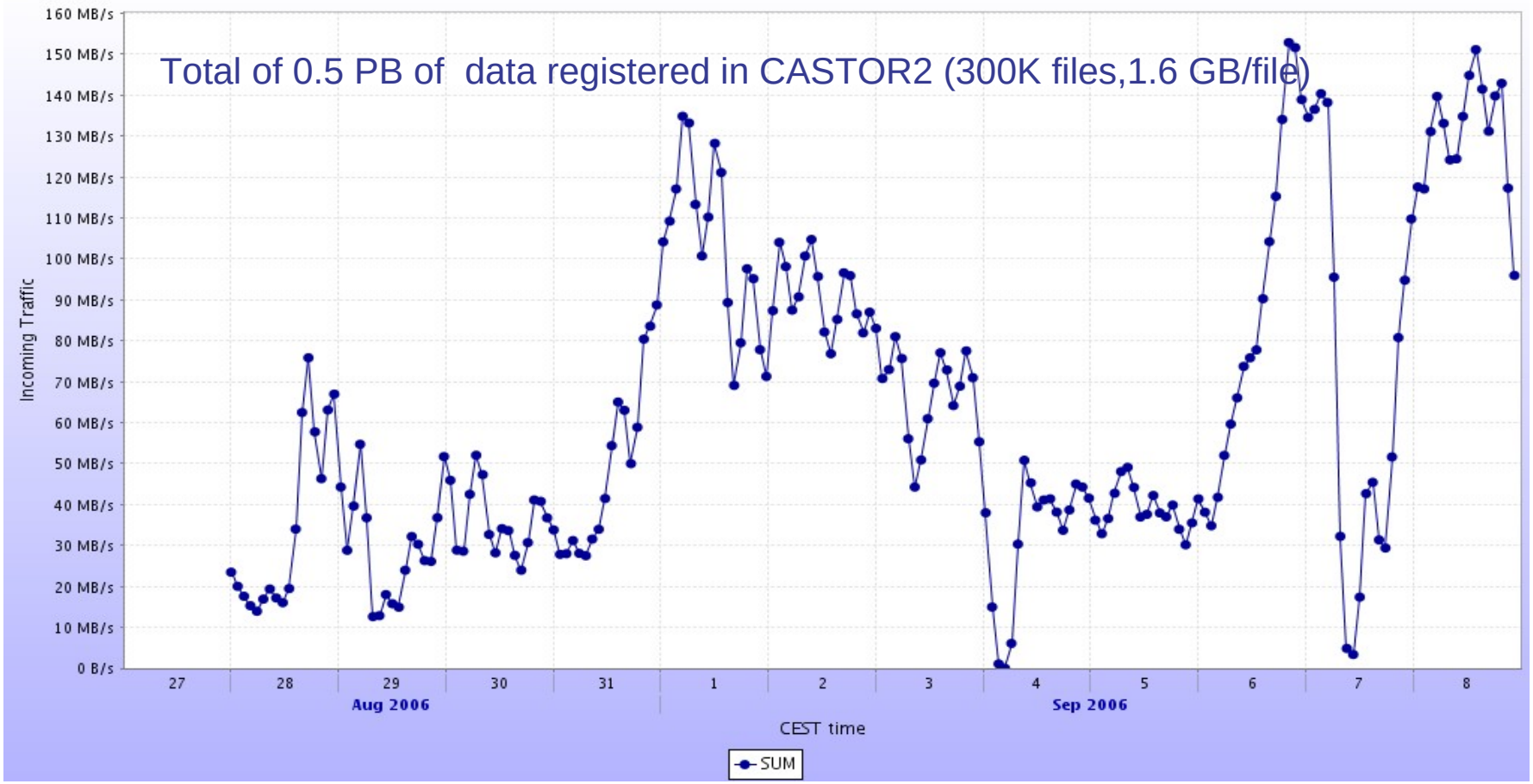
Resources statistics

- Resources contribution (normalized Si2K units): 50% from T1s, 50% from T2s
  - The role of the T2 remains very important!



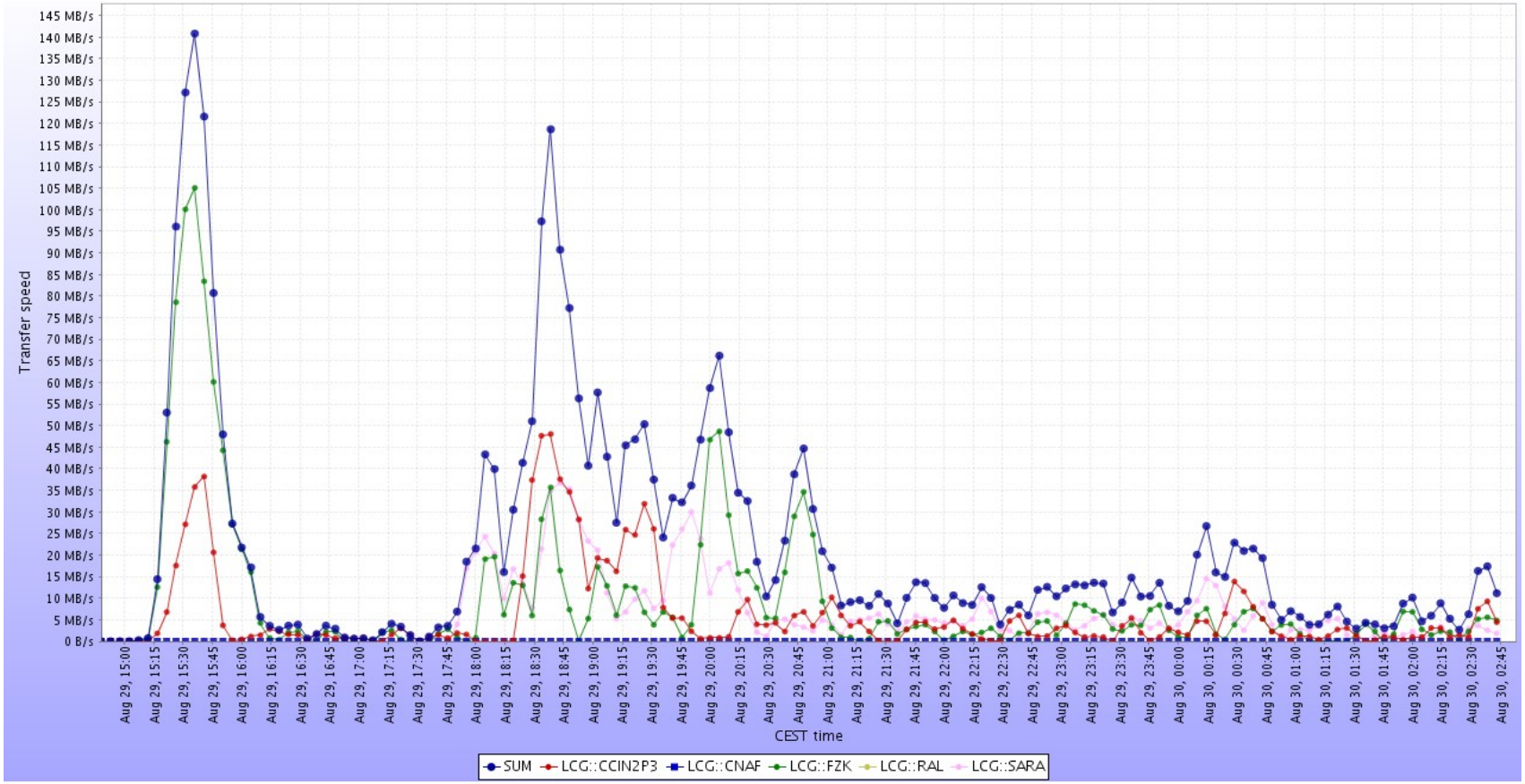
Data movement (xrootd)

- Step 1: produced data is sent to CERN
  - Up to 150 MB/sec data rate (limited by the amount of available CPUs) – ½ of the rate during Pb+Pb data export



Data movement (FTS)

- Step 2: data is replicated from CERN to the T1s
  - Test of LCG File Transfer Service
  - Goal is 300 MB/sec – exercise is still ongoing



Next Step: Alien v3.0

- Addressing the issues and problems encountered so far and trying to guess the technology trends

  - Scalability and complexity
    - We would like Grid to grow but how to manage complexity of such system?
  - Intra-VO scheduling
    - How to manage priorities with VO in particular for analysis?
  - Security
    - How to reconcile (Web) Service Oriented Architecture with growing security paranoia aimed at closing all network ports?
    - How to fulfil the legal requirements for process and file traceability?
  - Grid collaborative environment
    - How to work *together* on the Grid?

Intra-VO Scheduling

- Simple economy concept on top of existing fair share model
  - Users pay (virtual) money for utilizing Grid resources
  - Sites earn money by providing resources
  - The more user is prepared to ‘pay’ for job execution, sooner it is likely to be executed
- Rationale
  - To motivate sites to provide more resources with better QOS
  - To make users aware of the cost of their work
- Implementation (in AliEn v2-12)
  - Lightweight Banking Service for Grid (LBSG)
    - Account creation/deletion
    - Funds addition
    - Funds transaction
    - Retrieval of transactions’ list and balance

Overlay Messaging Networks

- Due to security concerns, any service that listens on open network port is seen as very risky
- Solution
  - We can use Instant Messaging protocols to create overlay network to avoid opening ports
    - IM can be used to route SOAP messages between central and site services
      - No need for incoming connectivity on site head node
    - It provides presence information for free
      - simplifies configuration and discovery
  - XMPP (Jabber)
    - A set of open technologies for streaming XML between two clients
      - Many open-source implementation
      - Distributed architecture
      - Clients connect to servers
      - Direct connections between servers
  - Jabber is used by Google IM/Talk
    - This channel could be used to connect grid users with Google collaborative tools

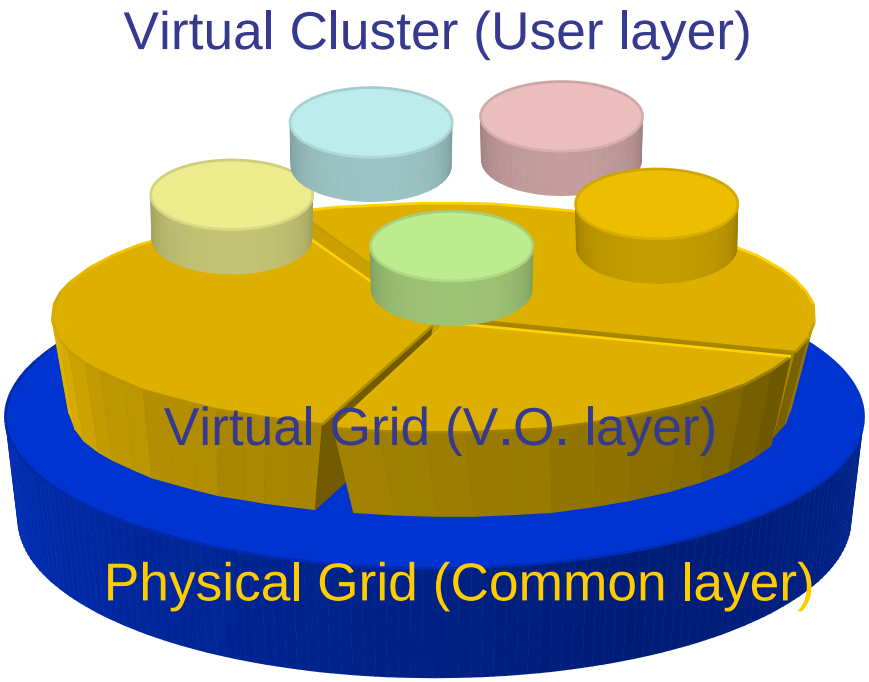
Sandboxing and Auditing

- The concept of VO identity gaining recognition
  - Model is accepted by OSG and exploited by LHC experiments
  - VO acts as an intermediary on behalf of its users
    - Task Queue – repository for user requests
      - AliEn, Dirac, Panda
    - Computing Element
      - Requests jobs and submits them to local batch system
  - Recently this model was extended to the worker node
    - Job Agent (pilot job) running under the VO identity on the worker node serves many real users

- The next big step in enhancing Grid security would be to run the Job Agents (pilot jobs) within a Virtual Machine
    - This can provide a perfect process and file sandboxing
    - Software which is run inside a VM can not negatively affect the execution of another VM

The Big Picture

- **Large "physical grid"**
  - Reliably execute jobs, store, retrieve and move files
- **Individual V.O. will have at given point in time access to a subset of these resources**
  - Using standard tools to submit the job (Job Agents as well as other required components of VO grid infrastructure) to physical grid sites
  - This way V.O. 'upper' middleware layer will create an overlay, a grid tailored to V.O needs but on smaller scale
  - At this scale, depending on the size of the VO, some of the existing solutions might be applicable
- **Individual users interacting with V.O middleware will typically see a subset of the resources available to the entire VO**
  - Each session will have certain number of resources allocated
  - In the most complicated case, users will want to interactively steer a number of jobs running concurrently on a many of Grid sites
  - Once again an overlay (Virtual Cluster) valid for duration of user session



AliEn/PROOF demo at SC05

Conclusions

- Alice is using Grid resources to carry out production (and analysis) since 2001
- At present, common Grid software does not provide sufficient and complete solution
  - Experiments, including Alice, have developed their own (sometimes heavy) complementary software stack
- In Alice, we are reaching a 'near production' level of service based on AliEn components combined with baseline LCG services
  - Testing of the ALICE computing model with ever increasing complexity of tasks
  - Seamless integration of interactive and batch processing models
    - Strategic alliance with ROOT
  - Gradual build up of the distributed infrastructure in preparation for data taking in 2007
  - Improvements of the AliEn software
    - hidden thresholds are only uncovered under high load
    - storage still requires a lot of work and attention
- Possible directions
  - Convergence of P2P and Web technologies
  - Complete virtualization of distributed computational resources,
  - By layering experiment software stack on top of basic physical Grid infrastructure we can reduce the scale of the problem and make Grid really work