



Enabling Grids for
E-science in Europe

www.eu-egee.org



CHEP 2004

The architecture of the AliEn system

Predrag Buncic

A. J. Peters, P. Saiz, J-F. Grosse-Oetringhaus



Talk Outline

- This talk presents the architecture of the original AliEn system, describes its evolution including a critical review of the major technology choices
 1. Introduction
 2. AliEn timeline
 3. AliEn inside EGEE
 4. More on a few selected services
 1. Resource Brokers
 2. File Catalogue and Metadata Catalogues
 3. Grid Access Service
 5. What did we learn?

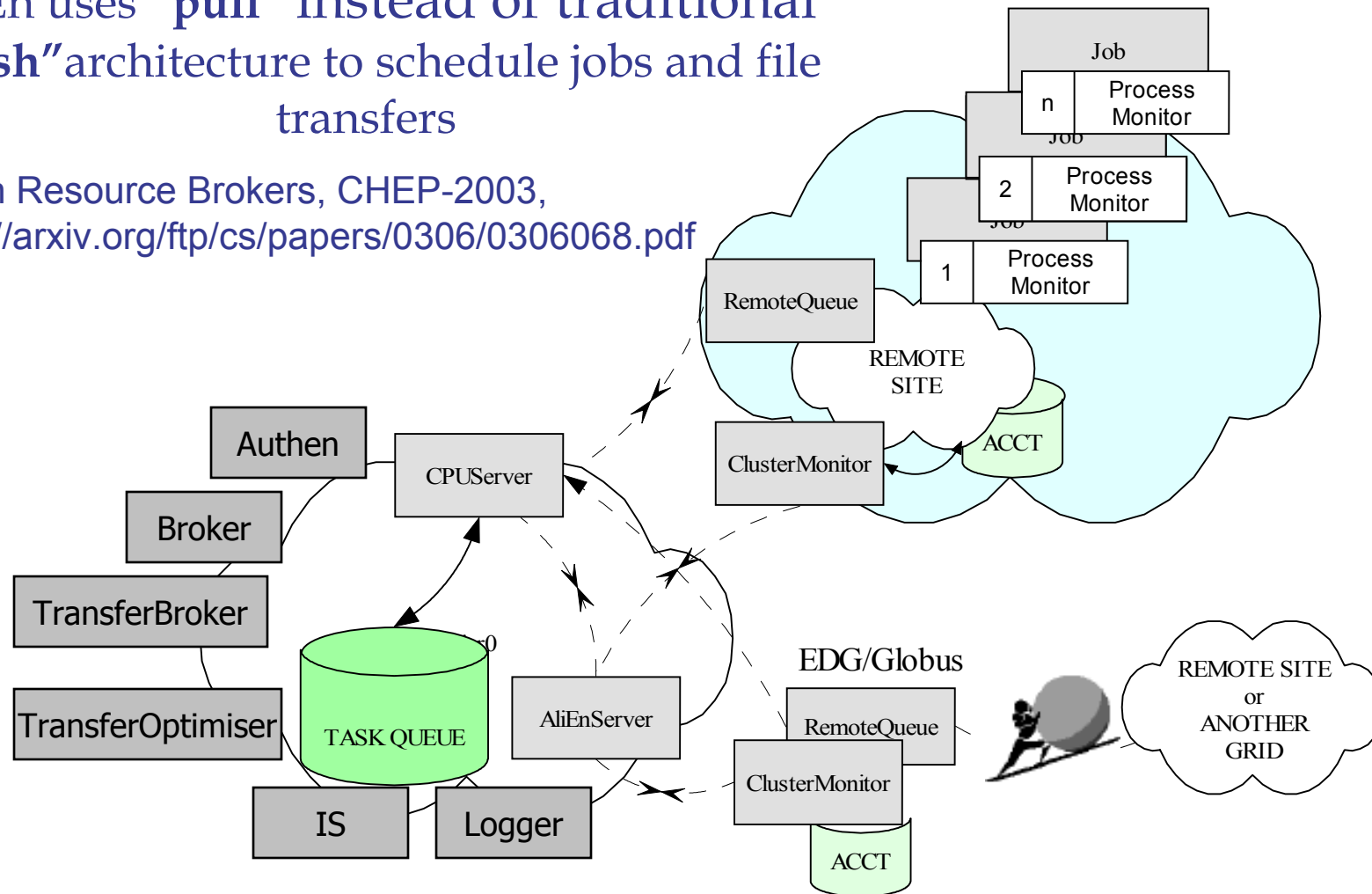
What is AliEn?

- A set of services (21 at the moment)
 - SOAP/Web Services (18)
 - Core Services
 - Resource Brokers, Optimizers, etc
 - Site Services
 - Abstract interfaces to resources (SE,CE, FTD,CM) with several backend implementation
 - Package Manager
 - Other (non Web) Services
 - ldap, database proxy, posix I/O
- Distributed file and metadata catalogue built on top of RDBMS
- User interfaces and API
 - command line, GUI, Web portal
 - C/C++/perl/java API, ROOT interface (TGrid/AliEn)
- JDL (Job description language) based on CONDOR ClassAds and Globus/GSI authentication

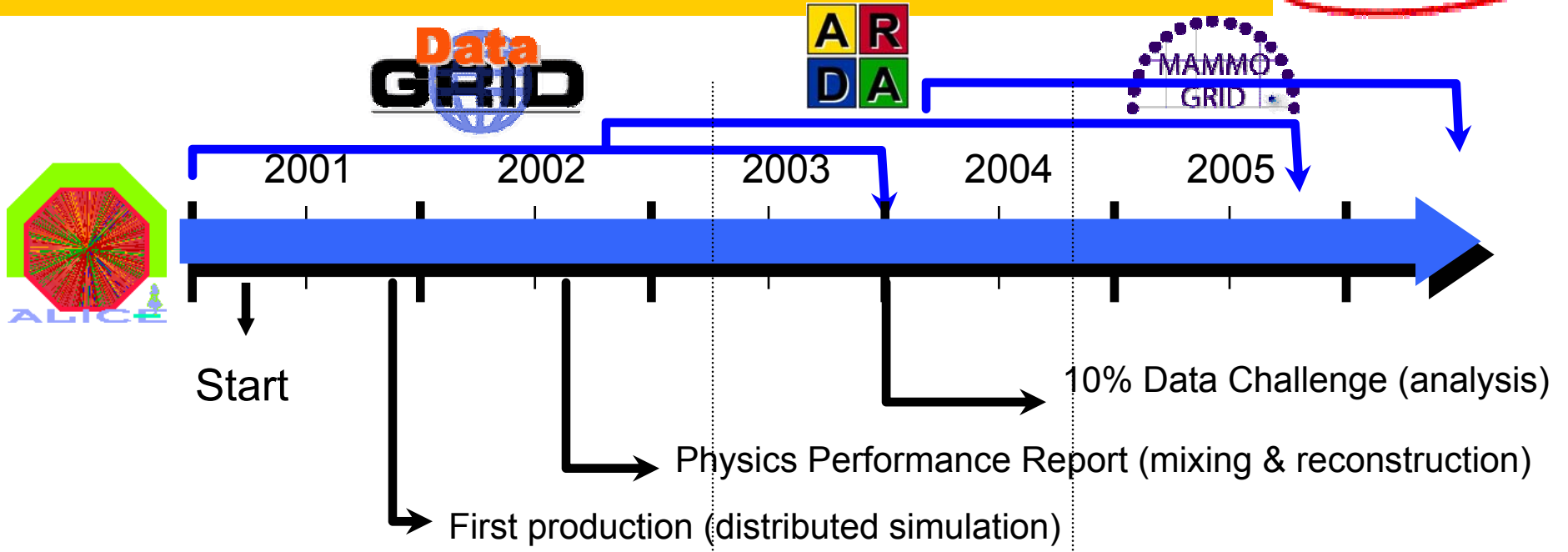
Push vs pull dilemma

AliEn uses “pull” instead of traditional “push” architecture to schedule jobs and file transfers

AliEn Resource Brokers, CHEP-2003,
<http://arxiv.org/ftp/cs/papers/0306/0306068.pdf>



AliEn Timeline



Functionality

Interoperability

Performance, Scalability, Standards

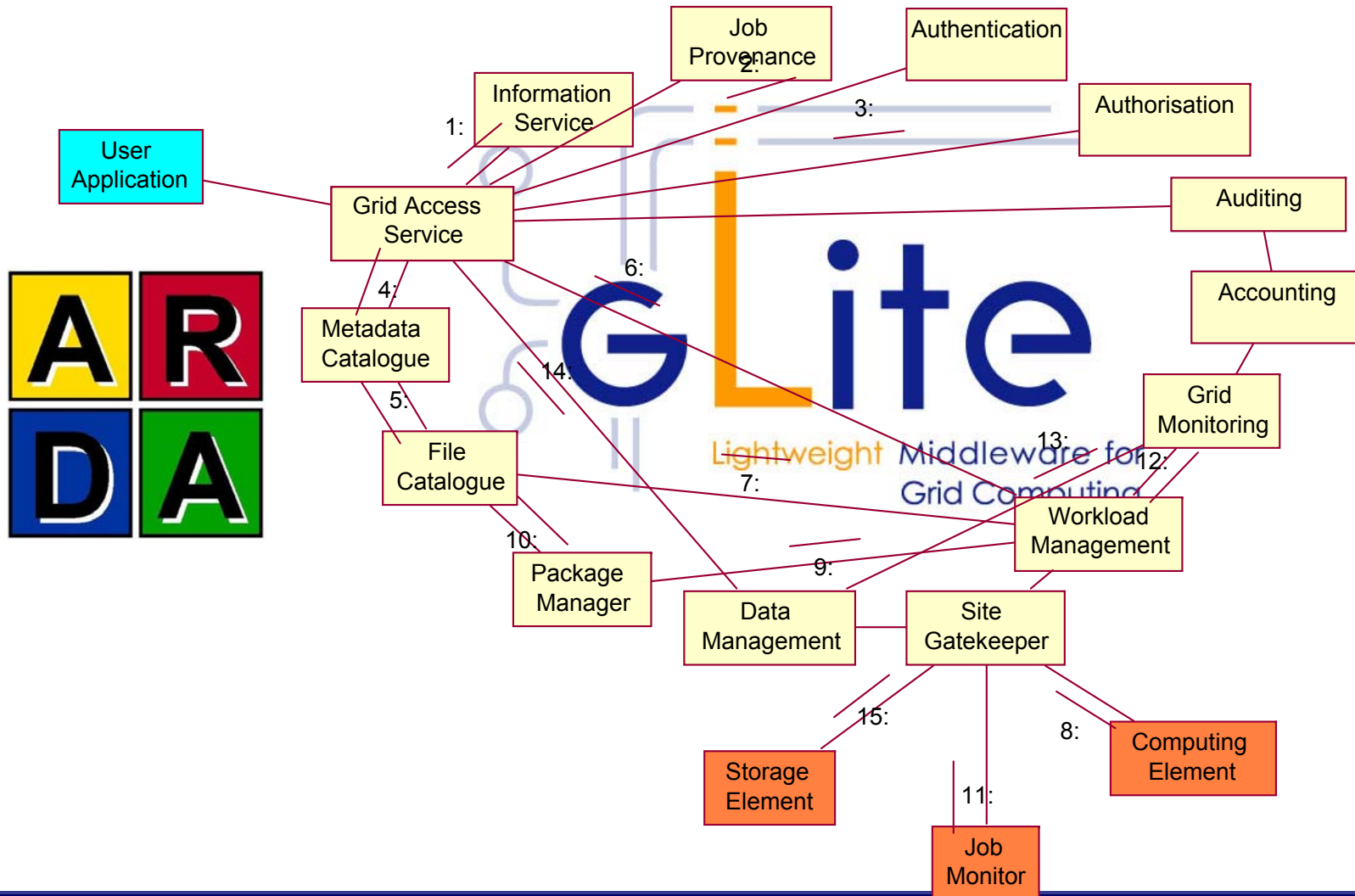


What happened to AliEn?

- Since April 1st, all developers are hired by the EGEE project
 - JRA1 (Joint Research Activity #1)
 - a group mandated to carry out re-engineering of Grid s/w and make deployable product out of the existing software components delivered by several Grid research projects (AliEn, EDG, VDT and others) taking into account input from all EGEE partners (including non HEP).
 - ARDA (A Realisation of Distributed Analysis for LHC)
 - a project to coordinate the activities to prototype distributed analysis systems for the LHC experiments using a grid
- Consequently, the development effort has been shifted to meet the needs of the EGEE project to
 - follow the EGEE architecture
 - fulfill the requirements of wider community of users
 - conform to widely accepted standards
 - comply with stringent security requirements
 - comply with EGEE software process
 - enable end-to-end analysis for HEP application gLite middleware



ARDA and gLite



AliEn in EGEE



gLite Lightweight Middleware for Grid Computing

What is gLite?

gLite (pronounced "gee-lite") is the next generation middleware for grid computing. Born from the collaborative efforts of more than 80 people in 10 different academic and industrial research centres as part of the [EGEE Project](#), gLite provides a bleeding-edge, best-of-breed framework for building grid applications tapping into the power of distributed computing and storage resources across the Internet.

Want to know more about gLite? Read the following [presentation](#).

gLite News

New gLite web site unveiled (13/09/2004)

The new gLite web site has officially gone online on Monday 13 September. The web site offers a single point of access to public documentation, installation packages and guides and loads of other useful information. The web site has been developed by the gLite [Integration Team](#) with the collaboration of all project members using original web templates from TERENA.

gLite People

The gLite software is produced as part of the EU EGEE Project funded by the European Communities. The following academic and industrial research centres are collaborating to the development of the software organized in three different Activities: [JRA1](#) (data management, workload management, monitoring, accounting, computing element, logging and bookkeeping), [JRA3](#) (security) and [JRA4](#) (network monitoring and provisioning).



The European Organization for Nuclear Research (CERN)



Istituto Nazionale di Fisica Nucleare (INFN), Italy

GLITE SUBSYSTEMS

ALIEN

DATA MANAGEMENT

ACCOUNTING

LOGGING AND BOOKKEEPING

MONITORING

SECURITY

WORKLOAD MANAGEMENT

DOWNLOAD

DOCUMENTATION

PACKAGES

ABOUT GLITE

EGEE JRA1

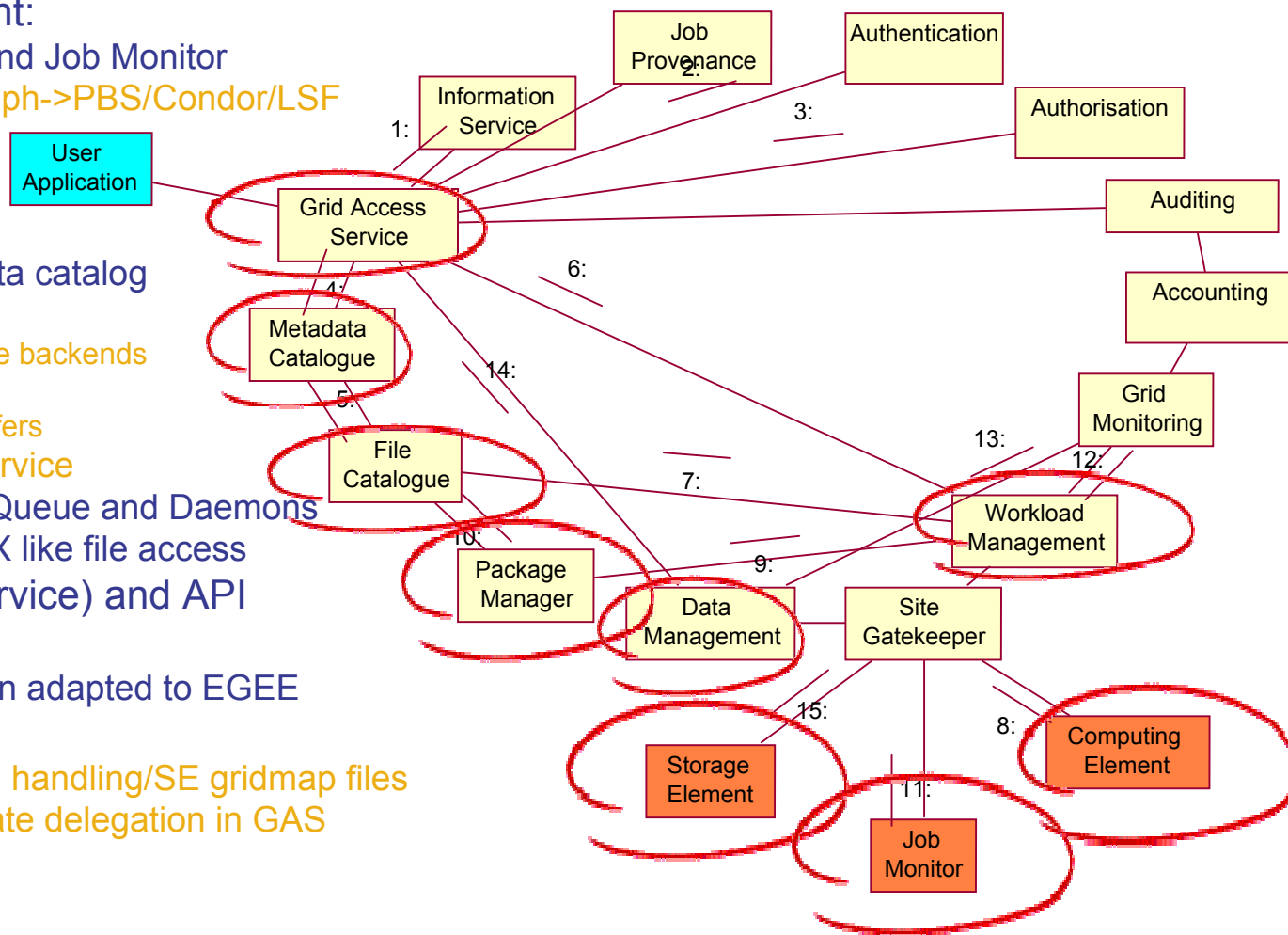
EGEE JRA3

EGEE JRA4

ABOUT EGEE

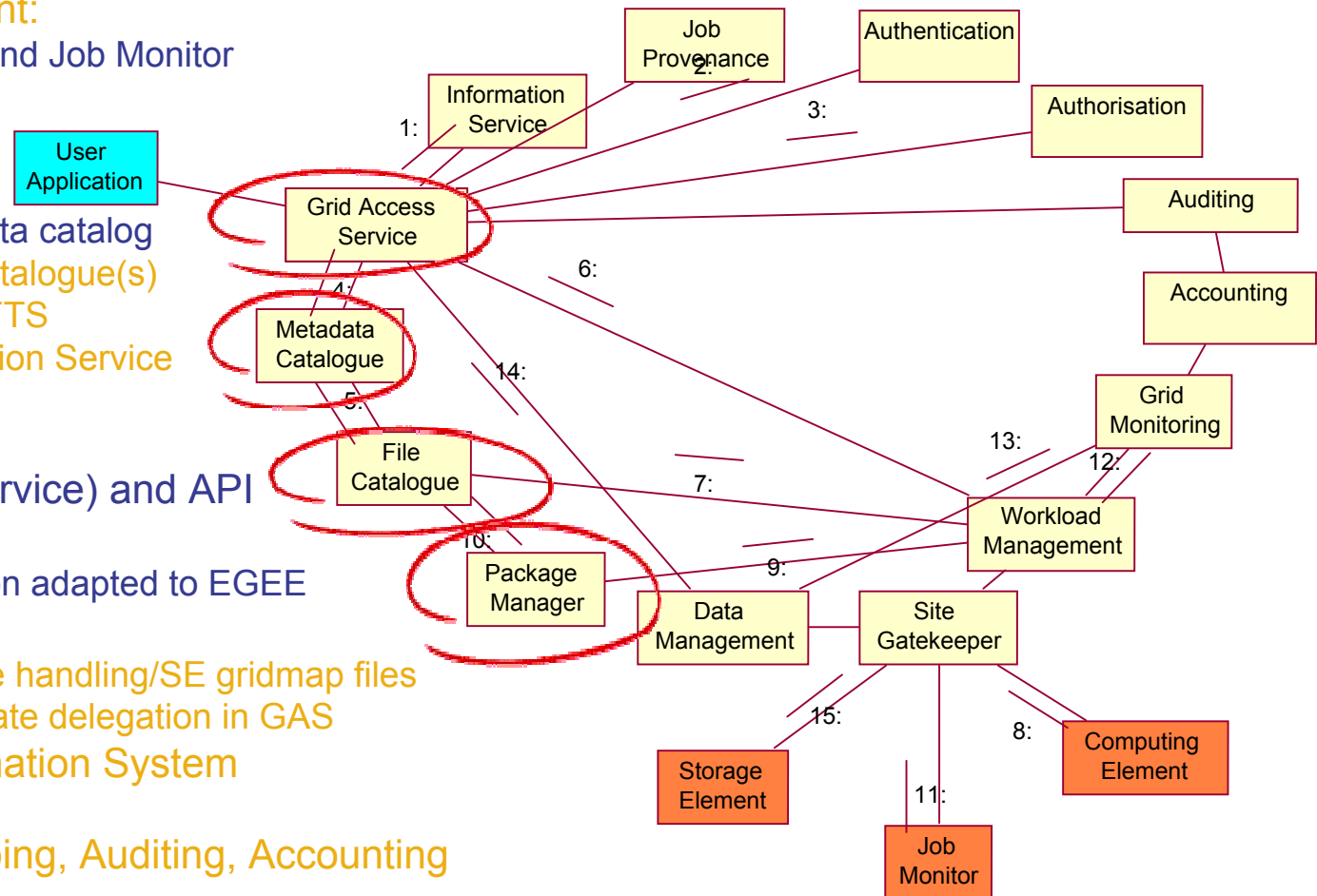
Current gLite Prototype

- AliEn “shell” as UI
- Workload Management:
 - Alien Task Queue and Job Monitor
 - CE->Condor-G->blaph->PBS/Condor/LSF
- Data Management
 - AliEn File & Metadata catalog
 - AliEn SE
 - Castor & D-Cache backends
 - SRM interface
 - gridFTP for transfers
 - Replica Location Service
 - AliEn File Transfer Queue and Daemons
 - Aiod/GFal for POSIX like file access
- GAS (Grid Access Service) and API
- Package Manager
 - AliEn implementation adapted to EGEE
- Security
 - VOMS for certificate handling/SE gridmap files
 - MyProxy for certificate delegation in GAS



First gLite Release

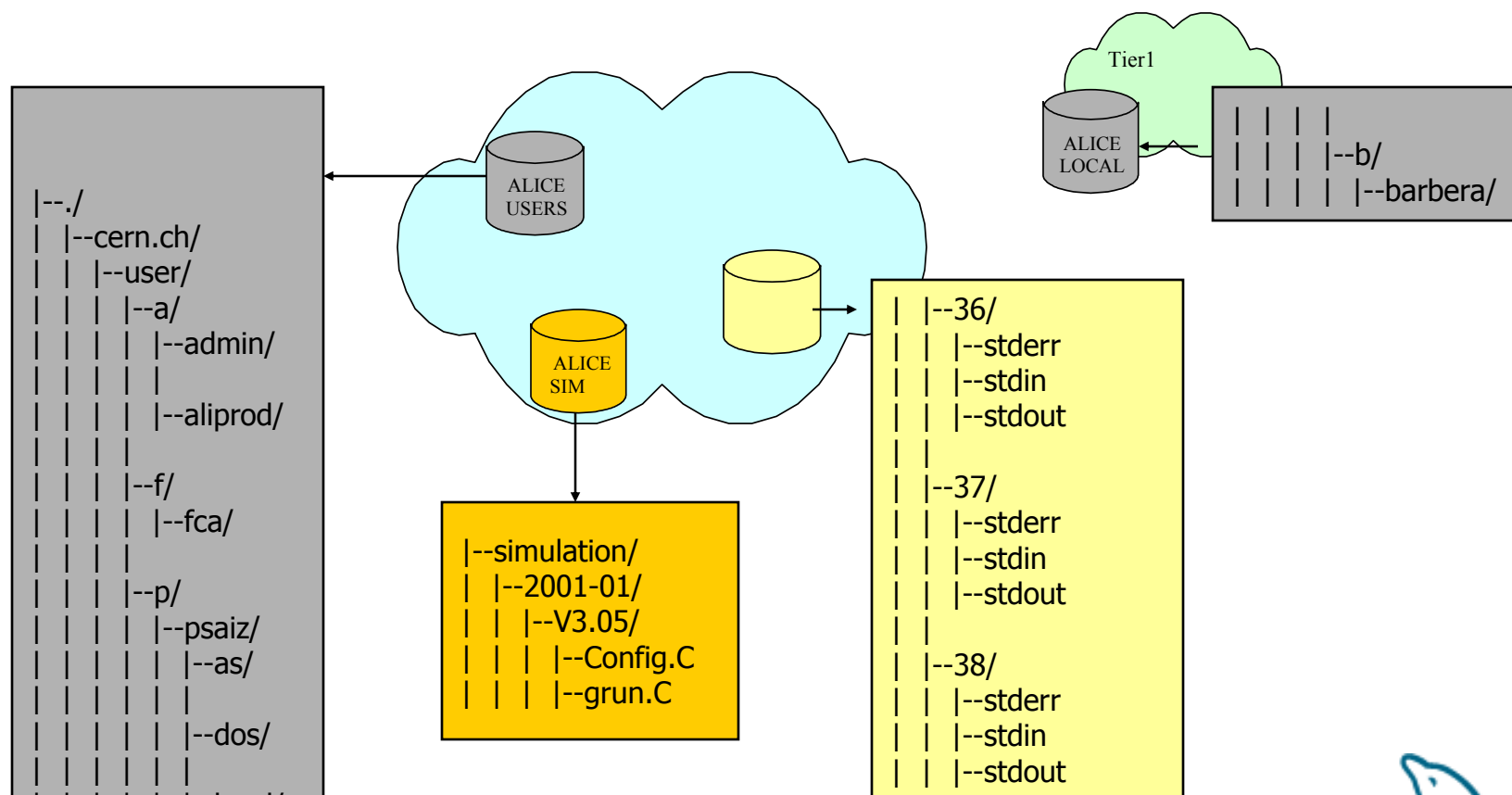
- AliEn “shell” as UI
- Workload Management:
 - Alien Task Queue and Job Monitor
 - EGEE CE & WMS
- Data Management
 - AliEn File & Metadata catalog
 - EGEE Metadata Catalogue(s)
 - New gLite SE and FTS
 - Local Replica Location Service
 - gLite-I/O
- GAS (Grid Access Service) and API
- Package Manager
 - AliEn implementation adapted to EGEE
- Security
 - VOMS for certificate handling/SE gridmap files
 - MyProxy for certificate delegation in GAS
- Monitoring and Information System
 - R-GMA
- Logging and Bookeeping, Auditing, Accounting
- Gatekeeper



More on selected AliEn Services

- File Catalogue
- Grid Access Service
- Workload Management (Task Queue)

File Catalogue and User Interface

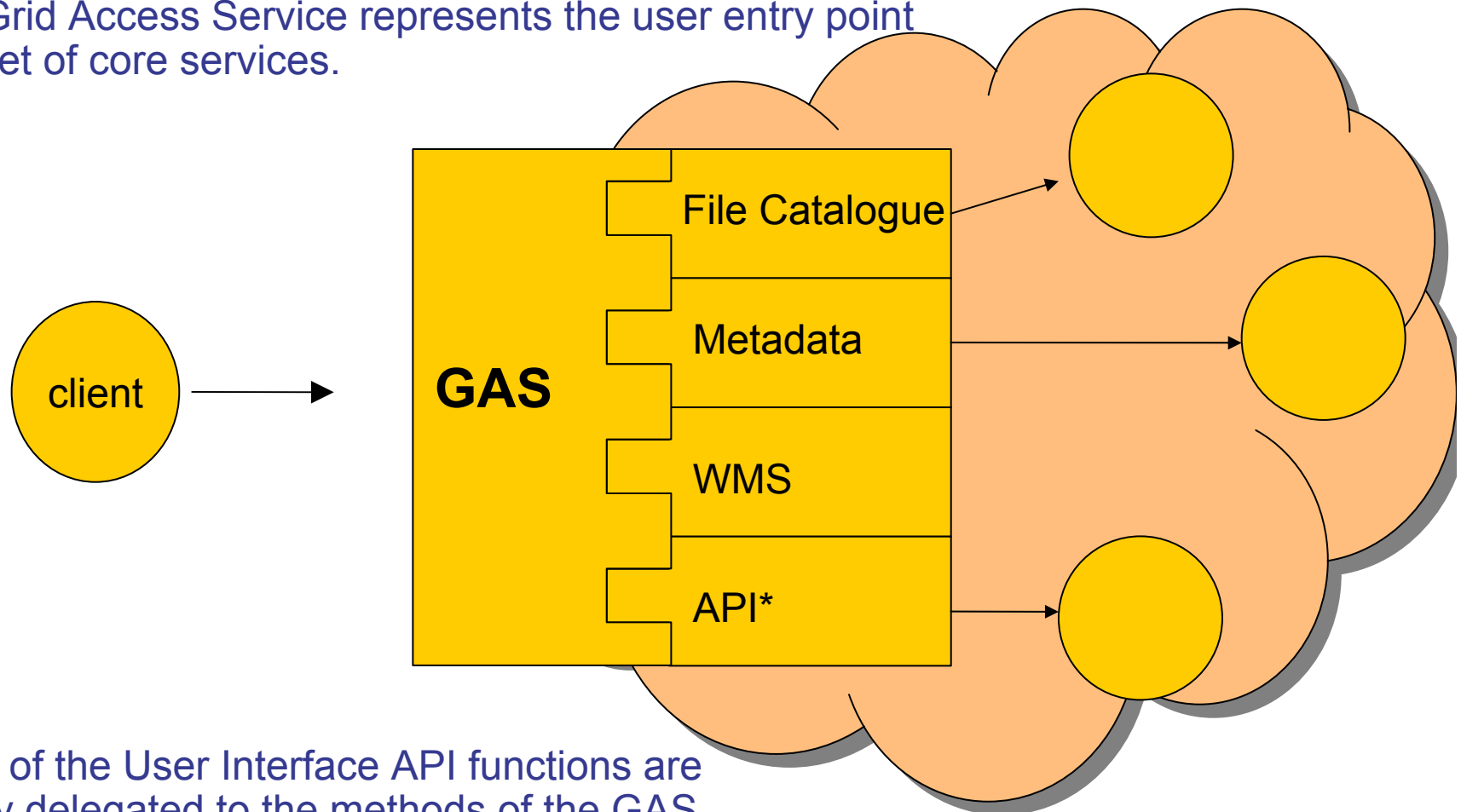


“One of the great things about Linux and Unix is that everything is a file -- or at least acts like one. From devices to sockets, the "everything is a file" paradigm has served Unix well for a long, long time.”



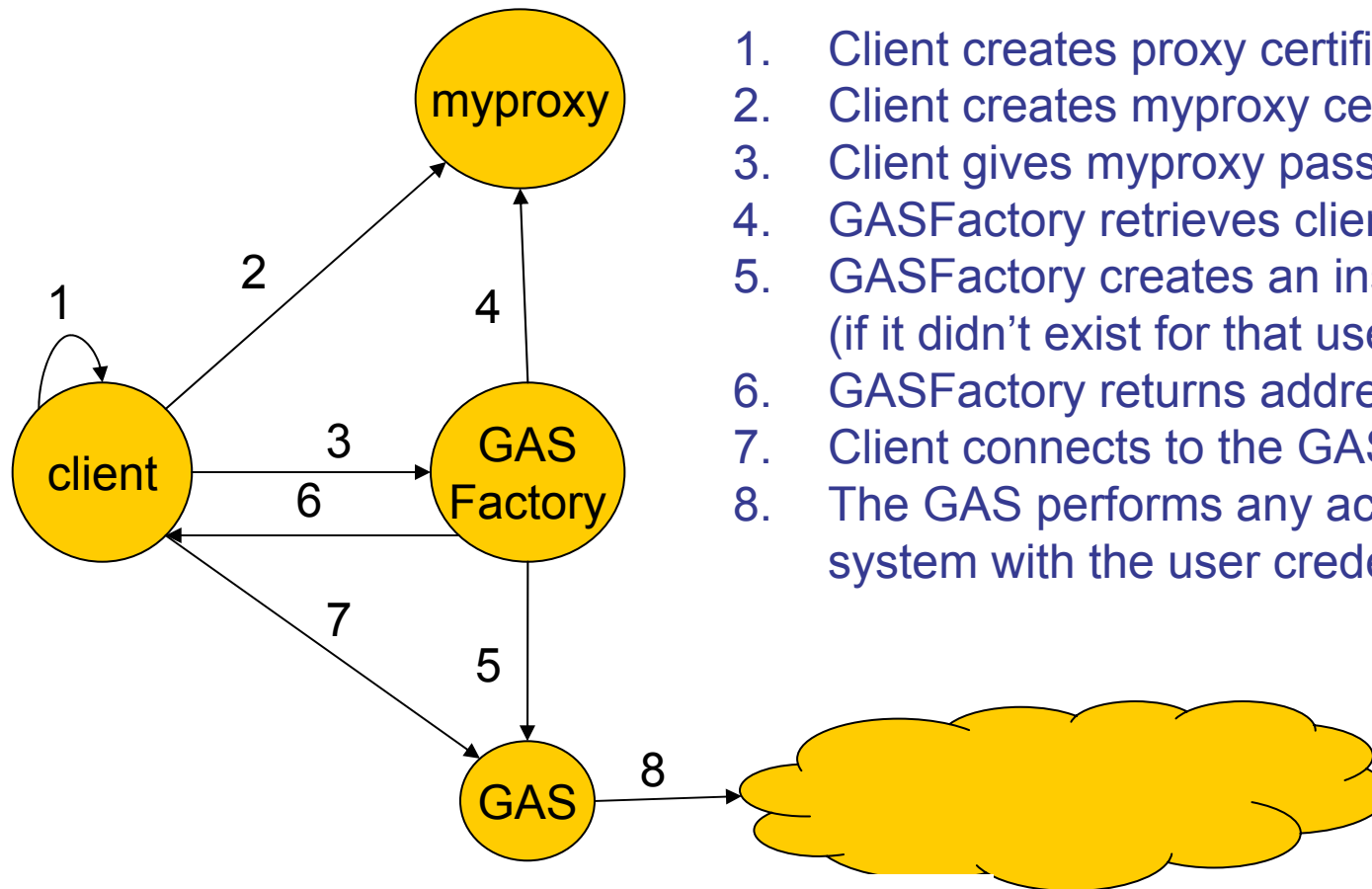
Grid Access Service (GAS)

- The Grid Access Service represents the user entry point to a set of core services.



- Many of the User Interface API functions are simply delegated to the methods of the GAS. In turn many of the GAS functions are delegated to the appropriate service.

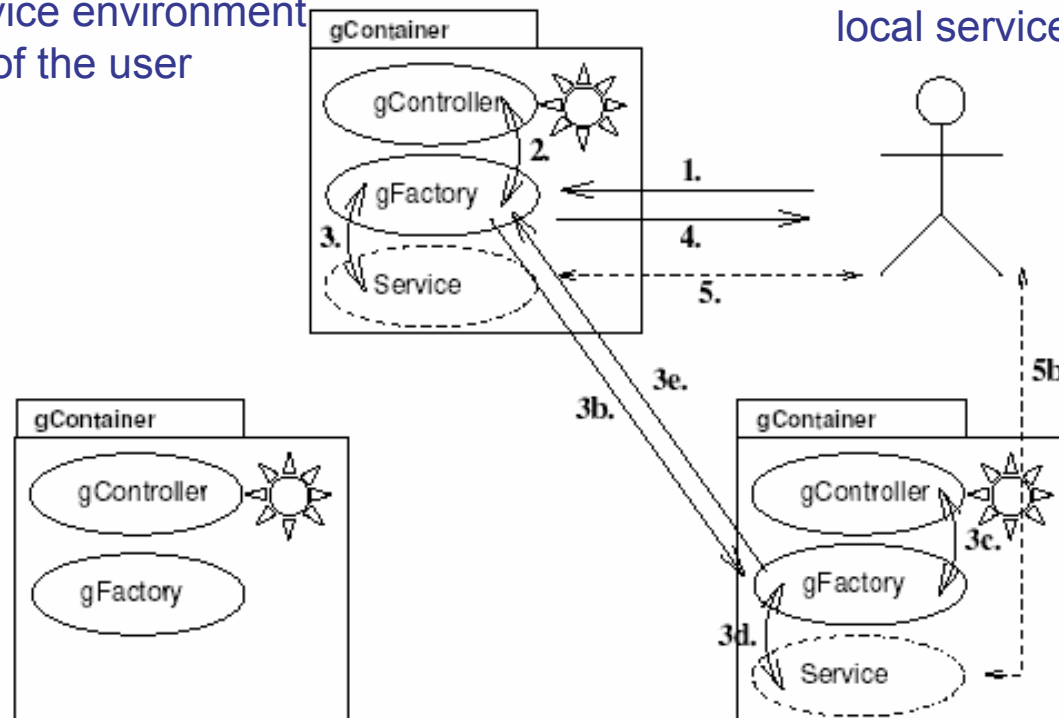
Use Case: Login



GAS Controller

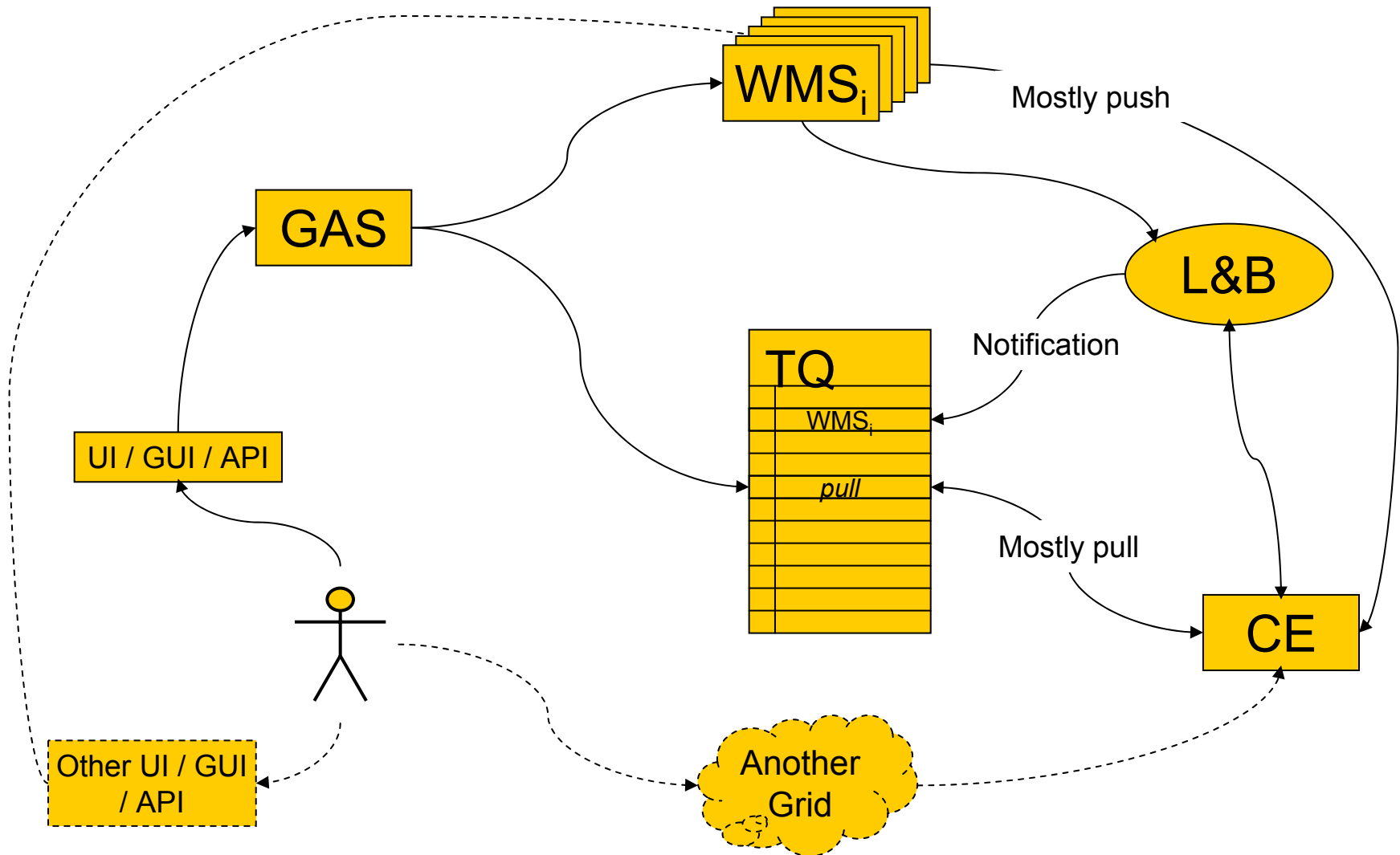
An instance of GAS should be created in a service environment in the proximity of the user (local site)

The GAS factory will ask Controller Service for appropriate service endpoint
The Controller can decide to create local service or can contact another Controller



GAS lifetime will be restricted to the lifetime of delegated proxy credentials and will be managed by the Controller Service and user who will be able to destroy his own GAS instance.

Task Queue and WMS proposal



Developing AliEn...

- Small team of developers
 - Between 2 and 4 + temporary students
- XP style rapid development cycle
 - Perl as main programming language
- Extensive use of Open Source components
 - More than 180, mostly perl modules
- What You See Is What You Get
 - Work with directly with users, always running prototype
 - No surprises, no big bang releases
- Continuous testing and deployment
 - About 40+ Alice sites
 - Reality check: Alice Data and Physics Challenges
- Collaboration with external partners
 - India, Ericsson research institute in Croatia, Hewlett-Packard

What did we learn?

- Perl as programming language
 - Extremely good for fast prototyping , easy extension and integration of foreign software components via reusable modules
- SOAP as communication protocol
 - Easy to get started, can be difficult when it comes to interoperability between services implemented in different languages, not good if one really needs performance
- WDSL, UDDI
 - Life is much easier if we do not have to use them
- Open Source components
 - Great if used unmodified but beware of license issues
- Abstract interfaces
 - Absolute must, even if it means an abstract interface to “standard interface”.

What did we learn ?

- Software process
 - Too much if it can kill the process but the discipline is needed
- Extreme programming
 - Worked incredibly well for me 😊
- Deployment
 - Easy if all external dependencies are distributed with the middleware, otherwise one needs an extra effort of entire integration team
- MySQL
 - Worked great for us as reliable and fast database backend
- License
 - It took almost one year to find rightful owner of AliEn software and define license. All EGEE software is under Open Source license.

More info...

- ✘ 443 [The AliEn Web Portal](#)
- ✘ 506-[The ALICE Data Challenge 2004 and the ALICE distributed analysis prototype](#)
- ✘ 436-[The LCG-AliEn interface, a realization of a MetaGrid system](#)
- ✘ 247-[Middleware for the next generation Grid infrastructure](#)
- ✘ 414-[Data Management in EGEE](#)
- ✘ 430-[Global Distributed Parallel Analysis using PROOF and AliEn](#)
- ✘ 414-[Data Management in EGEE](#)

- ✘ <http://www.eu-egee.org>
- ✘ <http://www.org.glite.org>