Enabling Grids for E-science in Europe

www.eu-egee.org

Grid Computing for Complex Problems, 29.11.2005

The EGEE Project and Grid Applications

Miroslav Ruda
Masaryk University and CESNET

EGEE is a project funded by the European Union under contract IST-2003-508833

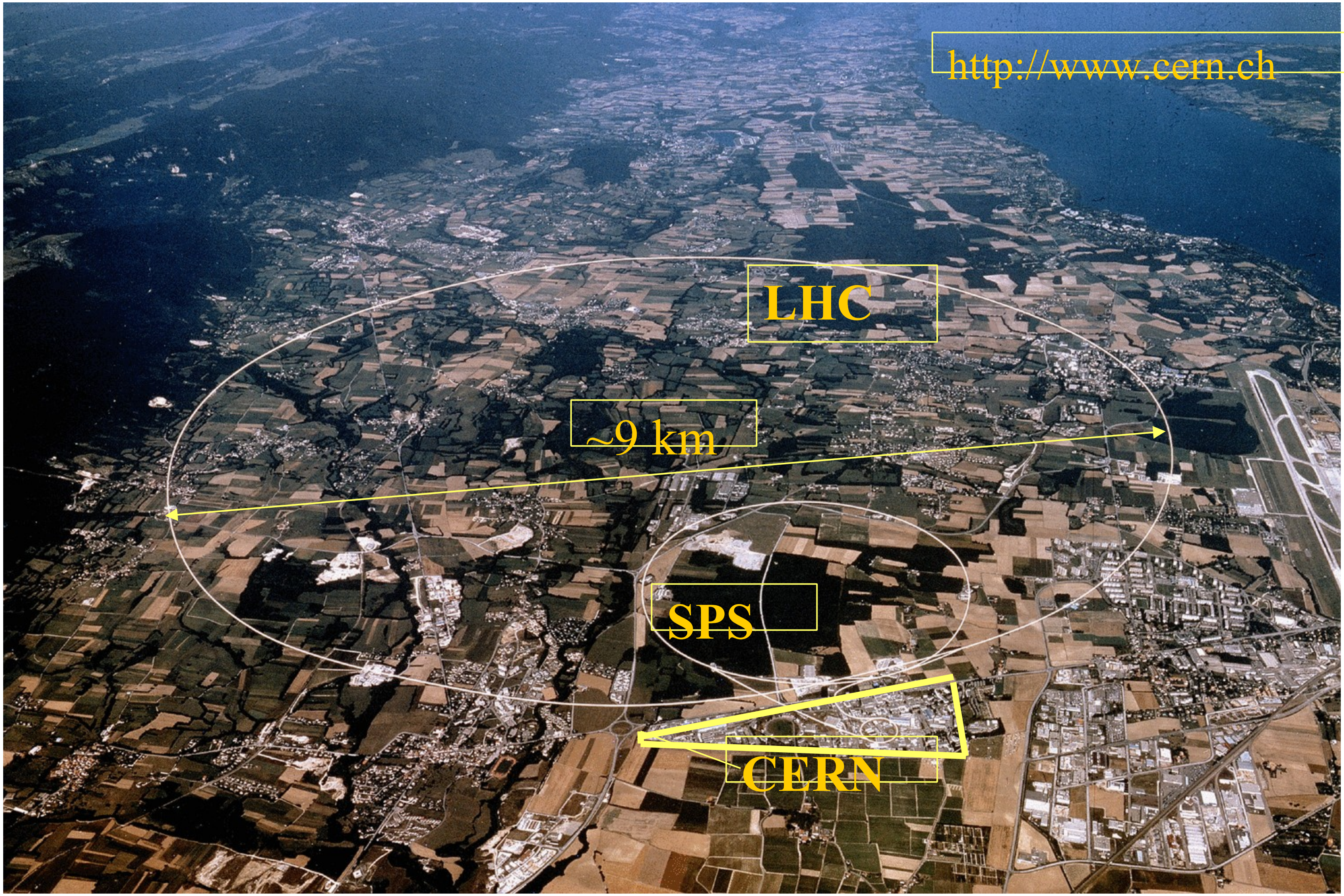
Outline

- EGEE introduction
  - Motivation
  - EDG/LCG/EGEE components
  - Virtual organisations, Security introduction
- EGEE middleware
  - Information Services
  - Data Management
  - Workload Management
    - L&B and Job Provenance
- VOCE - Virtual Organization for Central Europe
- Grid Application Toolkit (Gridlab)
  - Motivation
  - API, Implementation status

Acknowledgement

- This presentation is based on the work of many people:
  - Fabrizio Gagliardi, Flavia Donno and Peter Kunszt (CERN)
  - the EDG developer team
  - the EGEE training team
  - the NeSC training team
  - the SZTAKI training team

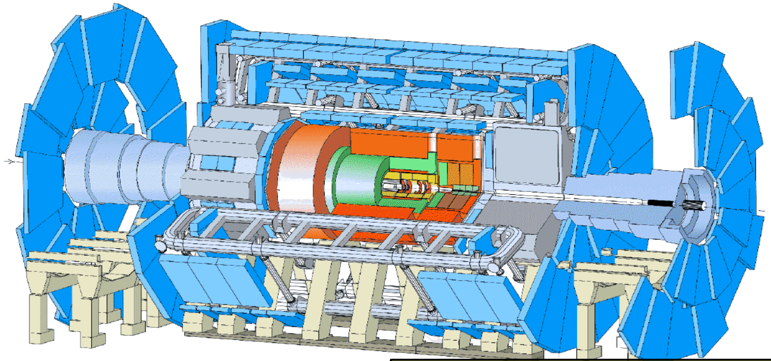
The CERN Large Hadron Collider



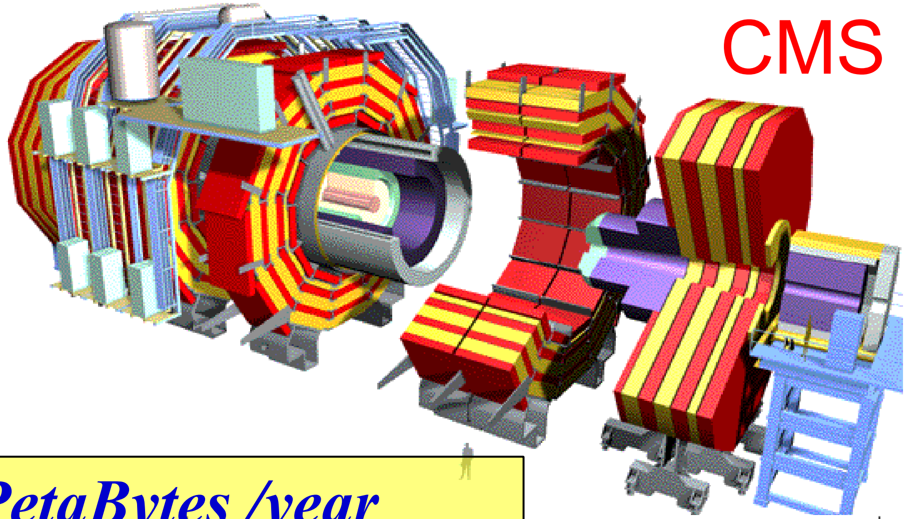
The LHC Experiments

Enabling Grids for E-science in Europe

ATLAS



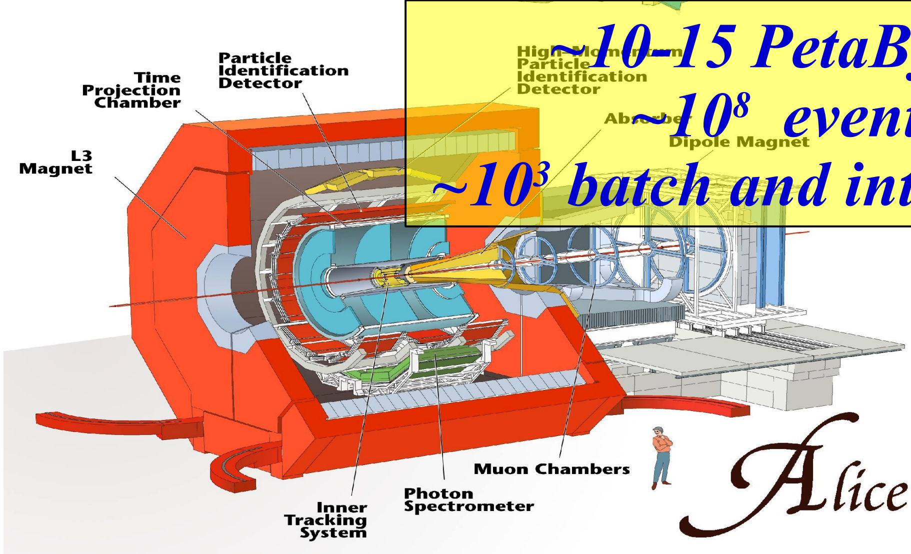
CMS

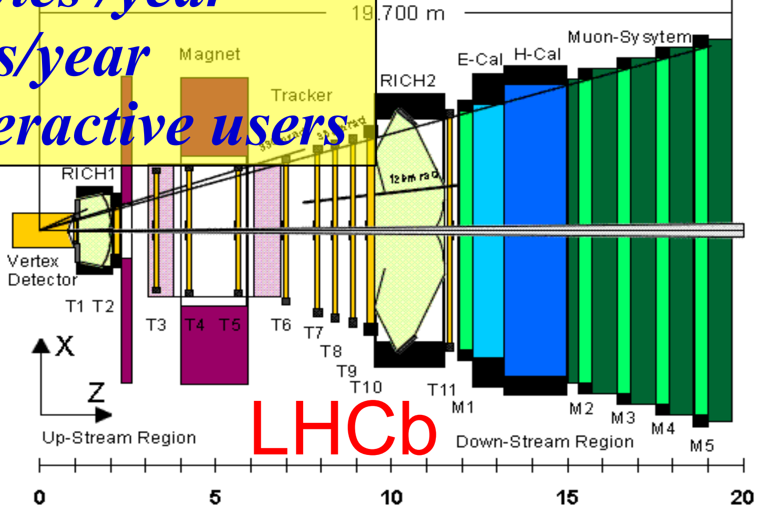


~10-15 PetaBytes /year

~10^8 events/year

~10^3 batch and interactive users

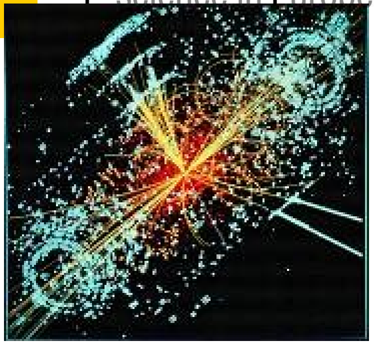


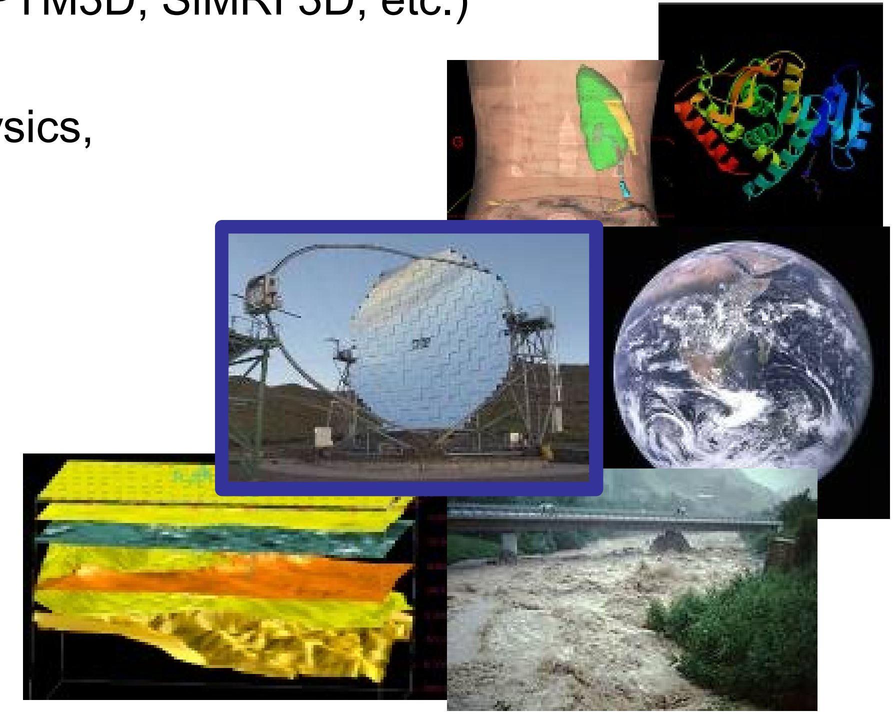


LHCb

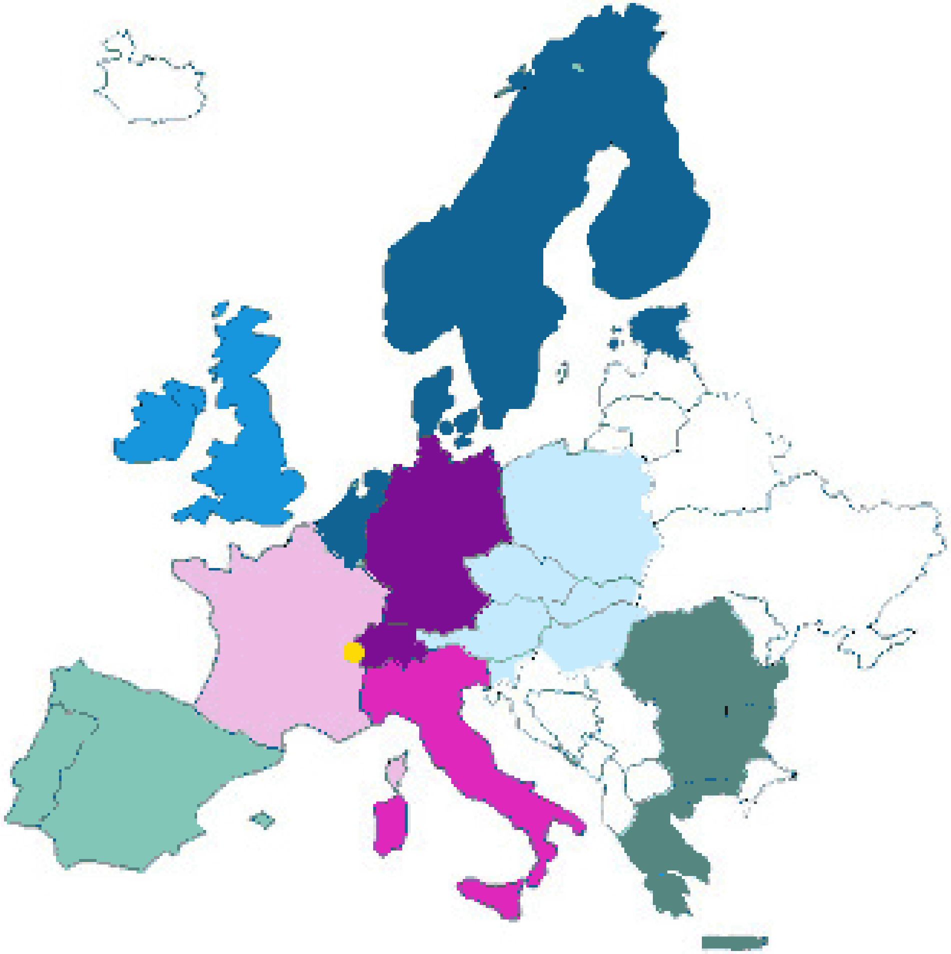
Application Support

- More than 20 applications from 7 domains
  - High Energy Physics
    - 4 LHC experiments (ALICE, ATLAS, CMS, LHCb)
    - BaBar, CDF, DØ, ZEUS
  - Biomedicine
    - Bioinformatics (Drug Discovery, GPS@, Xmipp_MLrefine, etc.)
    - Medical imaging (GATE, CDSS, gPTM3D, SiMRI 3D, etc.)
  - Earth Sciences
    - Earth Observation, Solid Earth Physics, Hydrology, Climate
  - Computational Chemistry
  - Astronomy
    - MAGIC
    - Planck
  - Geo-Physics
    - EGEODE
  - Financial Simulation
    - E-GRID





The EGEE Project (2004-'06) (http://www.eu-egee.org)



- 🟨 CERN
- ⬜ Central Europe (Austria, Czech Republic, Hungary, Poland, Slovakia, Slovenia)
- 🟪 France
- 🟪 Germany and Switzerland
- 🟦 Ireland and UK
- 🟪 Italy
- 🟦 Northern Europe (Belgium, Denmark, Estonia, Finland, The Netherlands, Norway, Sweden)
- 🟩 South-East Europe (Bulgaria, Cyprus, Greece, Israel, Romania)
- 🟩 South-West Europe (Portugal, Spain)

All work in EGEE will be organised on the basis of regionally based federations.

EGEE Threefold Mission

1. **Deliver production level Grid services**
   - essential elements: manageability, robustness, resilience to failure, consistent security model
   - scalability needed to absorb new resources rapidly as these become available
   - these elements will ensure the long-term viability of the infrastructure.

2. **Carry out a professional Grid middleware re-engineering activity**
   - in support of production level Grid services
   - continuously upgrade a suite of software tools capable of providing production level Grid services to a user base expected to grow and diversify rapidly.

3. **Undertake an outreach and training effort**
   - proactively market Grid services to new research communities in academia and industry
   - capture new e-Science requirements for the middleware and service activities
   - provide the necessary training to enable new users to benefit from the Grid infrastructure.

The EGEE Activities

32 Million Euros of EU funding over 2 years starting 1st April 2004

24% Joint Research

JRA1: Middleware Engineering and Integration

JRA2: Quality Assurance

JRA3: Security

JRA4: Network Services Development

28% Networking

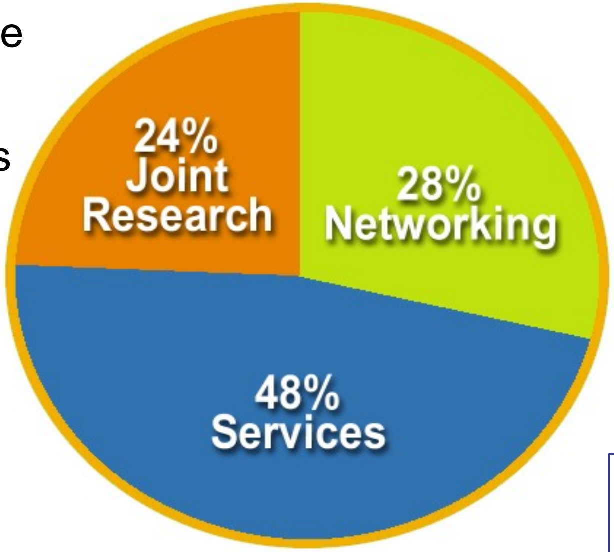
NA1: Management

NA2: Dissemination and Outreach

NA3: User Training and Education

NA4: Application Identification and Support

NA5: Policy and International Cooperation



48% Services

SA1: Grid Operations, Support and Management

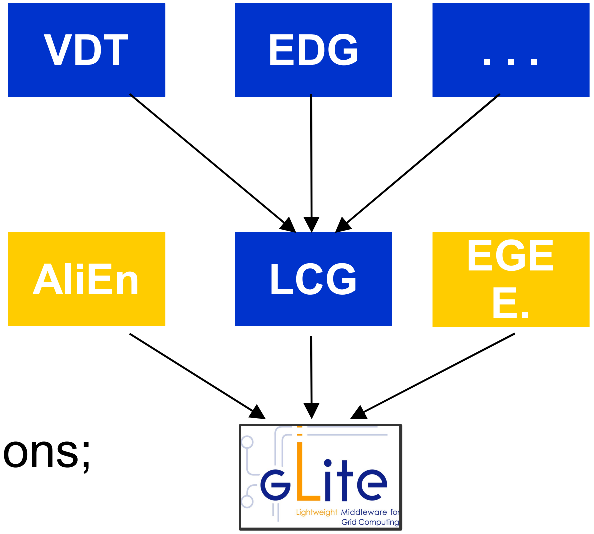
SA2: Network Resource Provision

Emphasis in EGEE is on operating a *production Grid* and on supporting the end-users.

Architecture Guiding Principles

- Lightweight (existing) services
  - Easily and quickly deployable
  - Use existing services where possible as basi
- Interoperability
  - Allow for multiple implementations
- Resilience and Fault Tolerance
- Co-existence with deployed infrastructure
  - Reduce requirements on site components
  - Co-existence (and convergence) with LCG-2 and Grid3 are essential for the EGEE Grid service
- Service oriented approach
  - Follow WSRF standardization
  - No mature WSRF implementations exist to date so start with plain WS-I
  - Provide framework to others so higher-level services can be developed quickly

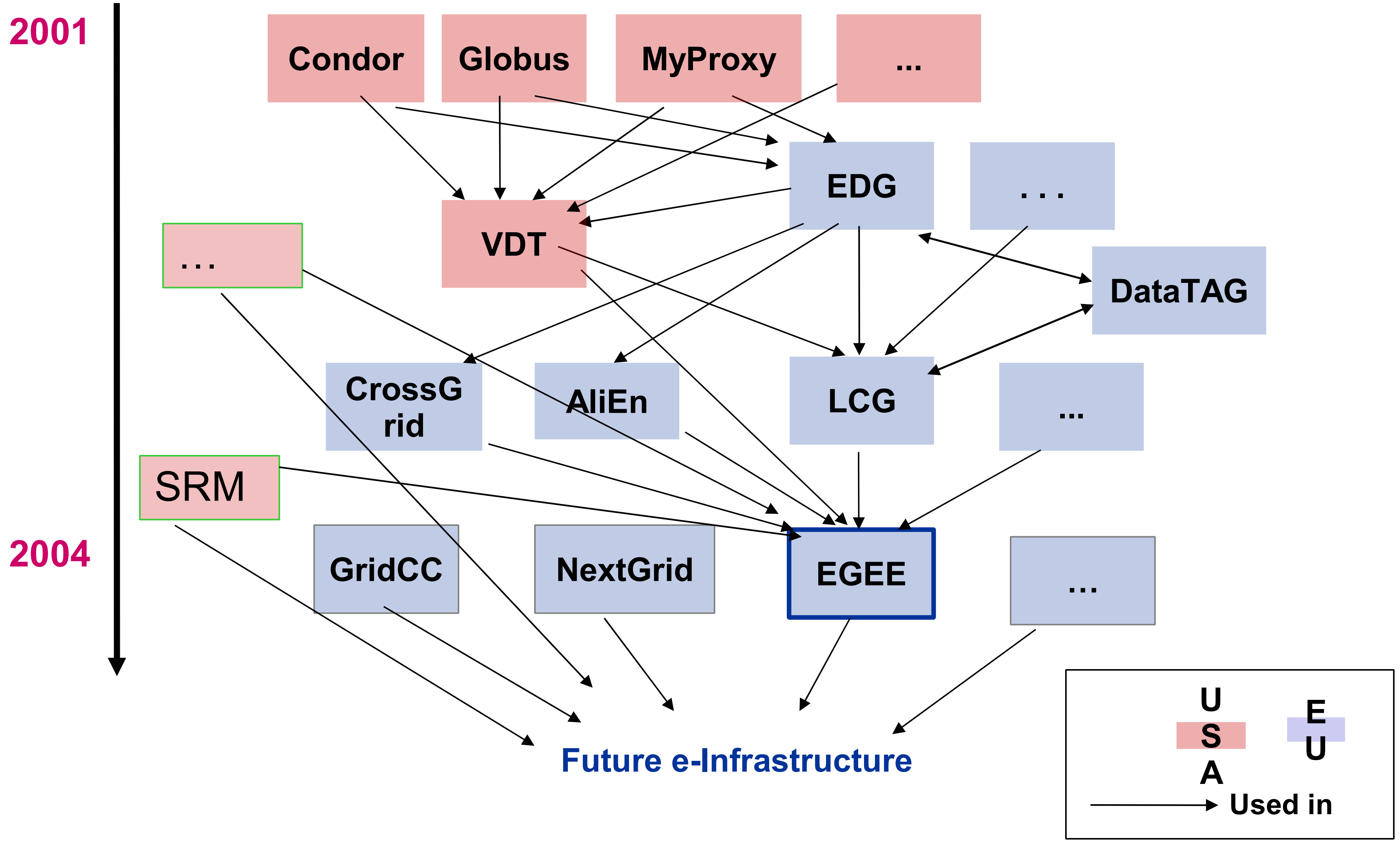
Approach



- Exploit experience and components from existing projects
  - AliEn, VDT, EDG, LCG, and others
- Design team works out architecture and design
  - Feedback and guidance from EGEE PTF & applications; Operations, LCG GAG & ARDA
- Components are initially deployed on a prototype infrastructure
  - Small scale (CERN & Univ. Wisconsin)
  - Get user feedback on service semantics and interfaces
- After internal integration and testing, components are delivered to grid operations group and deployed on the pre-production service

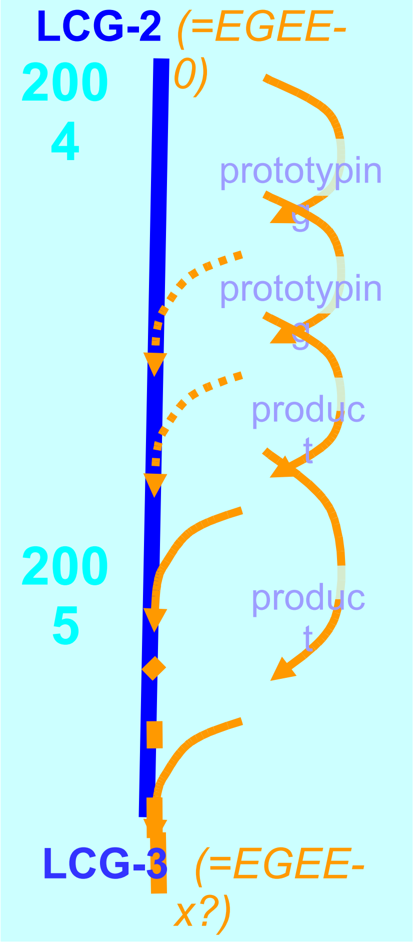
Draft Design - https://edms.cern.ch/document/487871/
PTF – Project Technical Forum (http://egee-ptf.web.cern.ch/egee-ptf/default.htm)
GAG – Grid Application Group (http://project-lcg-gag.web.cern.ch/project-lcg-gag/)
ARDA - A Realisation of Distributed Analysis for LHC (http://lcg.web.cern.ch/LCG/peb/arda/Default.htm)

EGEE view of history



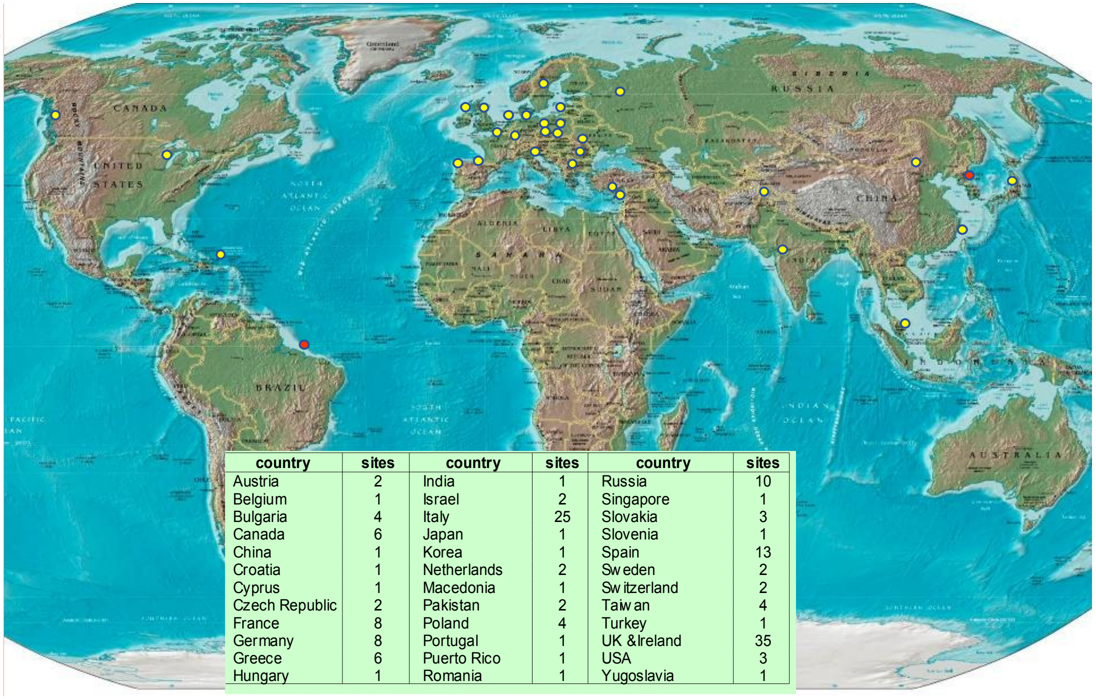
EGEE Middleware Migration

- **LCG-2**
  - Current base for ***production*** services
  - Evolves with certified new or improved services from the preproduction
- **Pre-production Service**
  - Early application access for new developments
  - Certification of selected components from gLite
  - Starts with LCG-2
- **Migrate new mware in 2005**
  - Organising smooth/gradual transition from LCG-2 to gLite for production operations



Where we are?

- Production
  - Sustained rate of ~**10.000 jobs/day**
  - **84 VOs** supported on the production infrastructure
    - **22 VOs** >1000 CPUh/m in average
    - Number of **users doubled** over the past 9 months
  - ~16.000 CPUs/170 sites
- Pre-Production Service
  - 14 sites, 4 virtual sites
- Interoperability efforts
  - OSG demonstrated
  - NorduGrid intensified

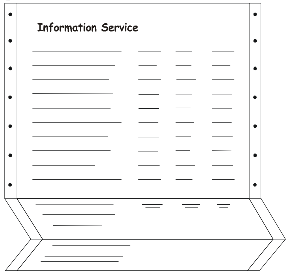


country	sites	country	sites	country	sites
Austria	2	India	1	Russia	10
Belgium	1	Israel	2	Singapore	1
Bulgaria	4	Italy	25	Slovakia	3
Canada	6	Japan	1	Slovenia	1
China	1	Korea	1	Spain	13
Croatia	1	Netherlands	2	Sweden	2
Cyprus	1	Macedonia	1	Switzerland	2
Czech Republic	2	Pakistan	2	Taiwan	4
France	8	Poland	4	Turkey	1
Germany	8	Portugal	1	UK &Ireland	35
Greece	6	Puerto Rico	1	USA	3
Hungary	1	Romania	1	Yugoslavia	1

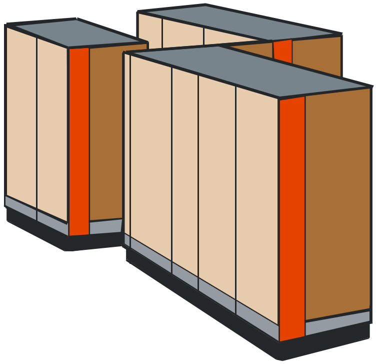
Main Logical Machine Types

- User Interface (UI)

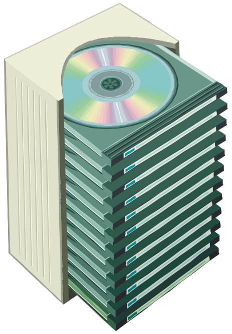
- Information Service (IS)



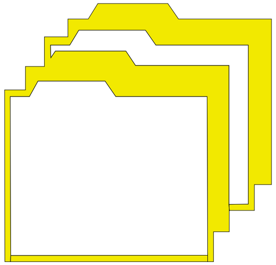
- Computing Element (CE)
  - Frontend Node
  - Worker Nodes (WN)



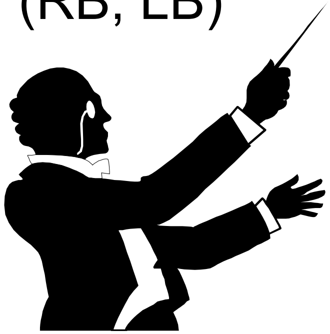
- Storage Element (SE)



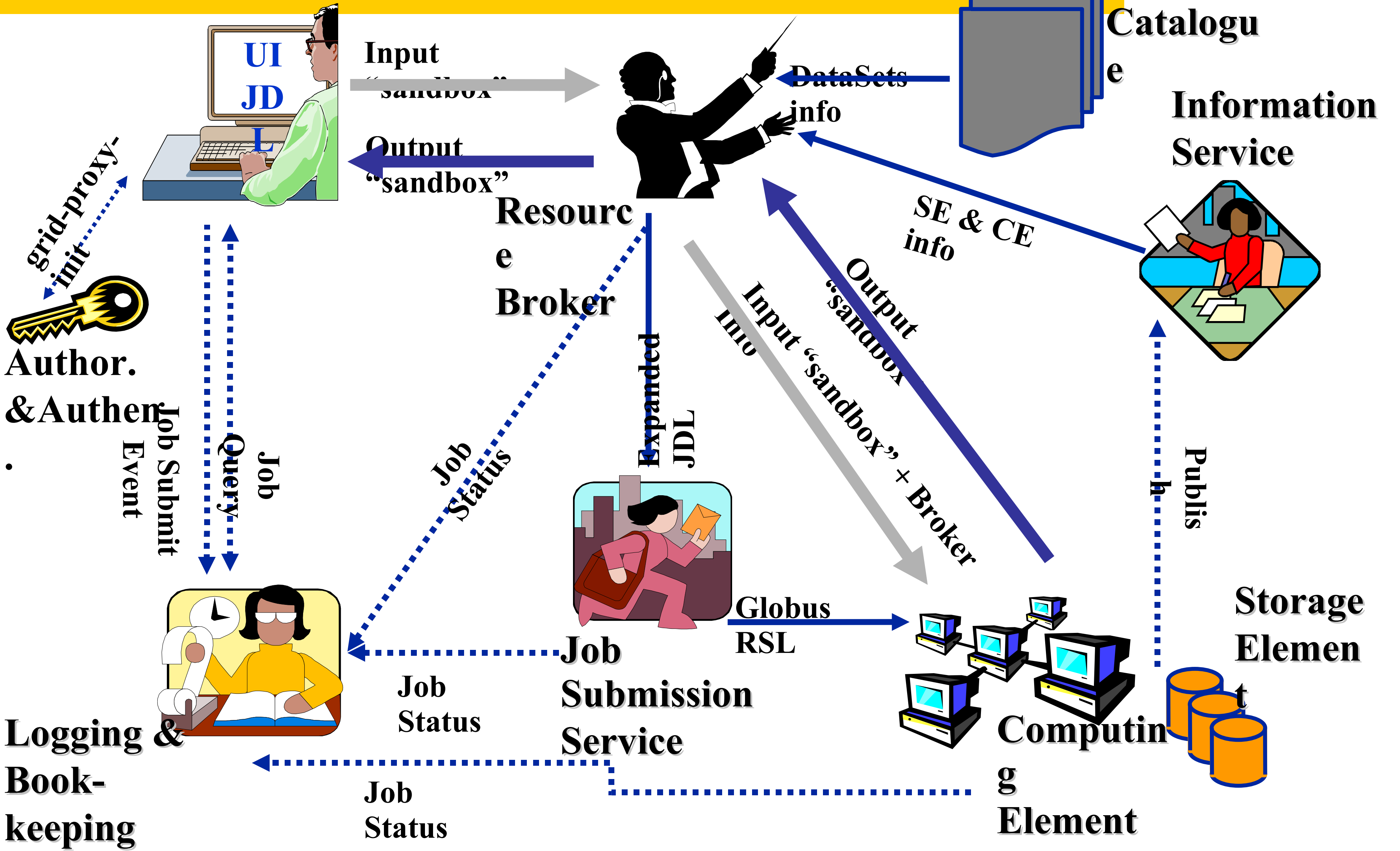
- Replica Catalog (RC,RLS)



- Resource Broker (RB, LB)



The lifecycle of an EGEE job



UI JDL

Input "sandbox"

Output "sandbox"

Resource Broker

DataSets info

Replica Catalogue

Information Service

SE & CE info

grid-proxy-init

Author. &Authen.

Job Submit Event

Job Query

Job Status

Expanded JDL

Input "sandbox" + Broker Info

Output "sandbox"

Publish

Job Submission Service

Globus RSL

Job Status

Storage Element

Computing Element

Logging & Book-keeping

Job Status

Security in the Grid

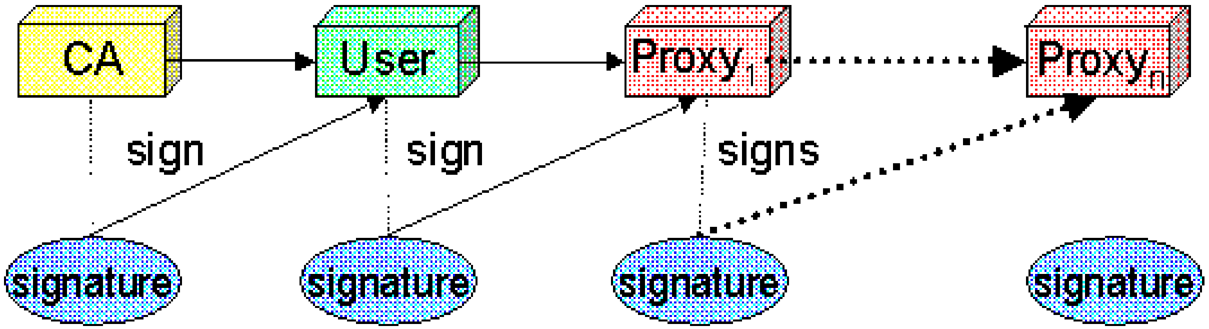
- Grid community mainly uses X.509 PKI
  - Well established and widely used (also for www, e-mail, etc.)
- X.509 certificate includes:
  - User identification (someone’s subject name)
  - Public key
  - A “signature” from a Certificate Authority (CA) that:
    - Proves that the certificate came from the CA.
    - Vouches for the subject name
    - Vouches for the binding of the public key to the subject
- Certification Authorities
  - EUGridPMA
  - IGTF launched
    - Coordinating European, Asian, and American GridPMAs

Grid Security Infrastructure (GSI)

- Globus Toolkit™ proposed and implements the Grid Security Infrastructure (GSI)
  - Protocols and APIs to address Grid security needs
- GSI protocols extend standard public key protocols
  - Standards: X.509 & SSL/TLS
  - Extensions: X.509 Proxy Certificates (single sign-on) & Delegation
- GSI extends standard GSS-API (Generic Security Service)
  - The GSS-API is the IETF standard for adding authentication, delegation, message integrity, and message confidentiality to applications.
- Proxy Certificate:
  - Short term, restricted certificate that is derived form a long-term X.509 certificate
  - Signed by the normal end entity cert, or by another proxy
  - Allows a process to act on behalf of a user
  - Not encrypted and thus needs to be securely managed by file system

Delegation

- Proxy creation can be recursive
  - each time a new private key and new X.509 proxy certificate, signed by the original key
- Allows remote process to act on behalf of the user
- Avoids sending passwords or private keys across the network
- The proxy may be a "Restricted Proxy": a proxy with a *reduced* set of privileges (e.g. cannot submit jobs).



Virtual Organisations

- A group of people sharing networked resources
- Cross organisational
- Shared authorisation/authentication
- A VO
  - Controls access to specified CE, SE
  - Usually comprises geographically distributed people
  - Requires the ability to know who has done what, and who will not be allowed to do it again…. Security.
- Current VO's:
  - Application oriented - HEP communities, biology, astronomy,…
  - Regional – VOCE, HunGridVO
- VOMS: enhanced flexibility in VO management

VOs and authorization

- Users belong to a Virtual Organization
  - Sets of users belonging to a collaboration
  - Each VO user has the same access privileges to Grid resources
  - List of supported VOs:
    - https://lcg-registrar.cern.ch/virtual_organization.html
- VOs maintain a list of their members
  - The list is downloaded by Grid machines to map user certificate subjects to local “pool” accounts: only mapped users are authorized in LCG

```
...
"/C=CH/O=CERN/OU=GRID/CN=Simone Campana 7461" .dteam
"/C=CH/O=CERN/OU=GRID/CN=Andrea Sciaba 8968" .cms
"/C=CH/O=CERN/OU=GRID/CN=Patricia Mendez Lorenzo-ALICE" .alice
...
```

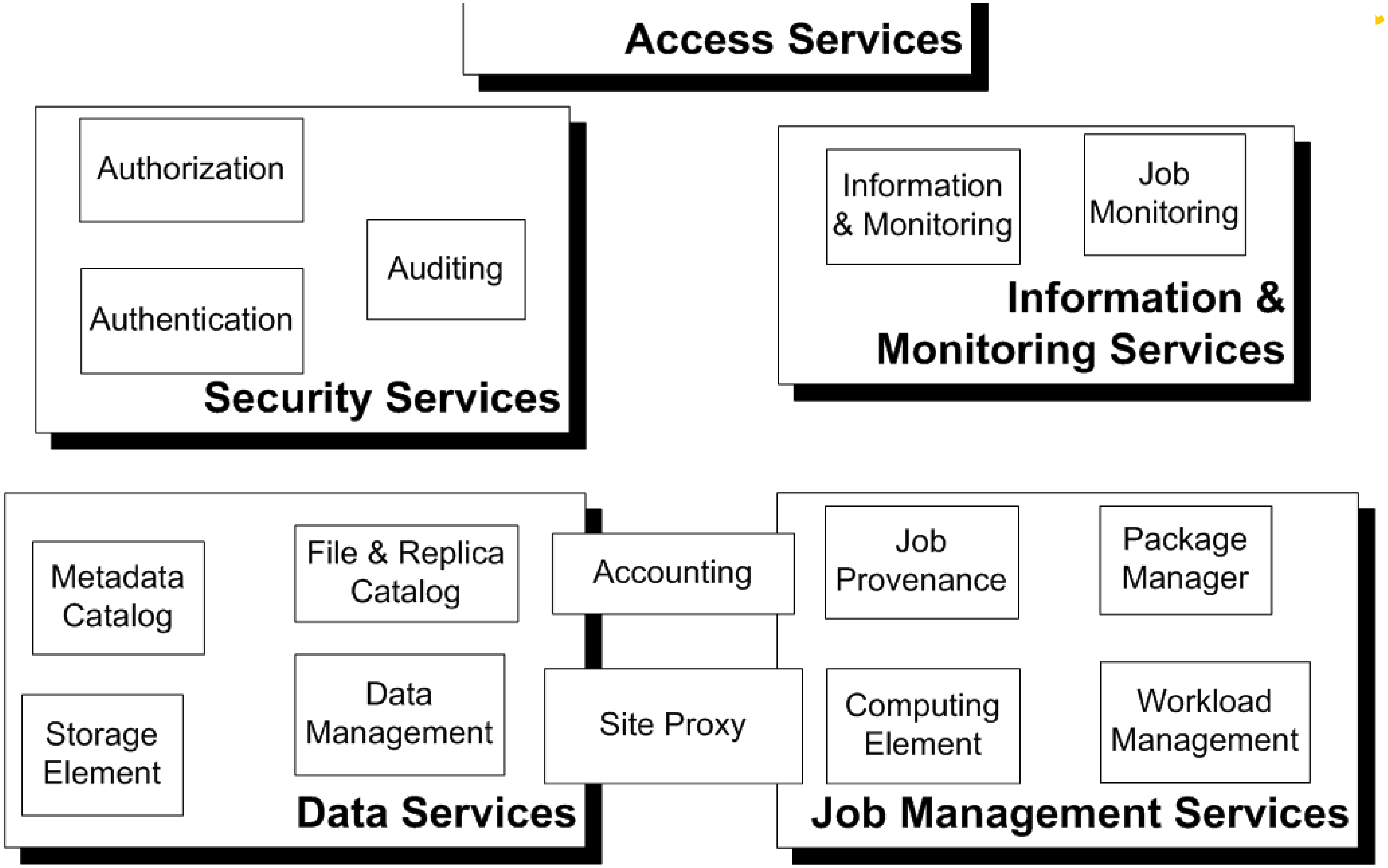
grid-mapfile

  - Sites decide which VOs to accept

EGEE middleware

- EGEE middleware provides generic Grid services:
  - Information
  - Job submission
  - Data management
  - Security
  - Logging
  - Monitoring

gLite Services



VOMS

- Virtual Organization Membership Service
- Central user database managed by VO
  - VOMS admin interface
- Aim
  - User gets attributes (groups, roles) from central DB
    - VOMS core service
  - User forwards these attributes to the services
    - VOMS proxy
  - Sites trust these attributes
    - No need for updates from the central DB
    - No need to care about the user identity

Grid information system

- Provides information on both:
  - The Grid itself
    - Mainly for the middleware packages, administrators
    - The user may query it to understand the status of the Grid
  - Grid applications
    - For users
- Flexible infrastructure
  - Able to cope with nodes in a distributed environment with an unreliable network
  - Dynamic addition and deletion of information producers
  - Security system able to address the access to information at a fine level of granularity
  - Allow new data types to be defined
  - Scalable
  - Good performance
  - Standards based

The Information System

- Two main Information System technologies are used
  - LDAP based from Globus, used in LCG and EDG
  - R-GMA, developed by the European DataGrid Project
- The **Information System** (IS) provides information about the Grid resources and their status
- LCG-2 - LDAP based Globus Monitoring and Discovery Service (MDS) architecture
  together with Berkley Database Information Indexes (BDII)
  - MDS is Part of Globus Toolkit, compatible with other elements
    - Used most often for resource selection
    - aid user/agent to identify host(s) on which to run an application
    - Standard mechanism for publishing and discovery
    - Decentralized, hierarchical structure
    - Soft-state protocols
    - Caching, Grid Security Infrastructure credentials

The MDS-2 Architecture and BDII

- Computing and storage resources at site report their static and dynamic status via the **GRISes** (Grid Resources Information Servers) to the **GIIS** (Grid Index Information server)
- The role of the GIIS is to collect info from all the GRISes and other GIISes information sources, but it has shown his scalability limits, growing and growing the number of sites
- Because of this, the **BDII** (Berkely DB Information Index) was introduced.
- The GIIS has been kept at site level, to collect info from the site GRISes.

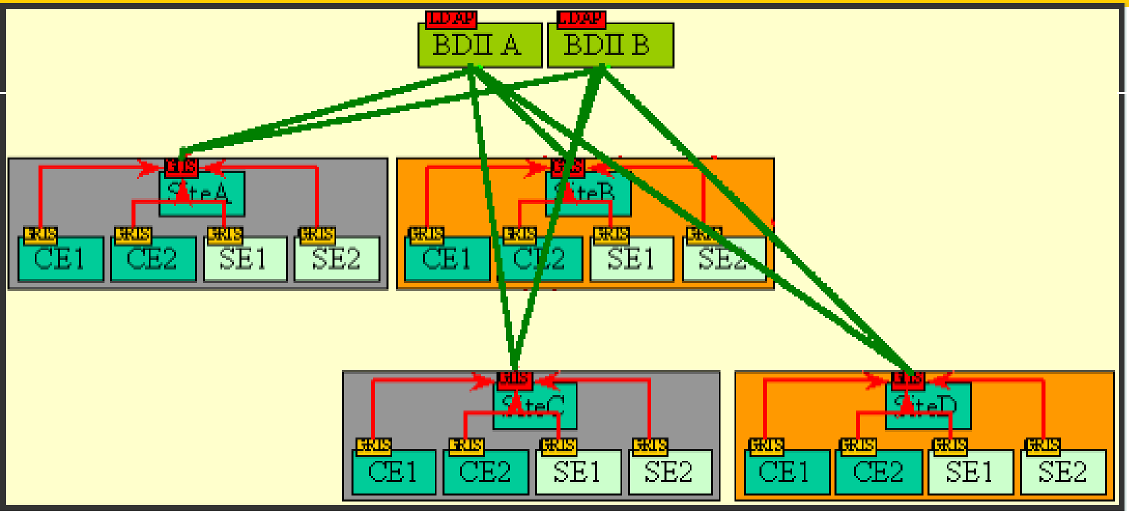
The responsible services

- **Lower level: GRIS**
  - Scripts and configuration files generate ldif files containing the information (for example, general information of the nodes)
  - Other tools responsible of the dynamic information (for example, available and/or used space into a SE) – the so called information providers
- **Medium level: local GIIS**
  - Same procedure taking the information from the registered GRISes
  - The GRIS/GIIS system can answer 1query/15min
- **Top level: BDII**
  - Publish the information of the site GIISs making a refresh every 2 minutes

The BDII

- The BDII queries the GIISes and acts as a cache storing information about the Grid status in its database.
- Each BDII contains information from the site GIISes defined by a **configuration file**, which it accesses through a web interfaces.
- Users and other Grid services (such as the RB) can **interrogate BDIIs** to get information about the Grid status
- Very up-to-date information can be found by directly interrogating the site GIISes or the local GRISes that run on the specific resources.

The complete hierarchy



- Local GRISes run on CEs and SEs at each site and report dynamic and static information regarding the status and availability of the services

```
ldapsearch –x –h <hostname>
-p 2135 –b “mds-vo-
name=local,o=grid”
```

- At each site, GIIS collects the information of all resources given by the GRISs

```
ldapsearch –x –h <hostname> -p 2135 –b “mds-vo-name=<name>,o=grid”
```

- Each site can run a BDII

It collects the information coming from the GIISs and collects it in a data base

```
ldapsearch –x –h <hostname> -p 2170 –b “o=grid”
```

The LDAP

- The information system is built on the **Light-weight Directory Access Protocol**
- It offers a hierarchical view of information
- The entries are arranged in a Directory Information Tree (DIT)
- Resources (computers, storage, …) each publish their part in this tree
- Queries can be posed to the current Information and Monitoring Service using LDAP search commands
- It establishes the transport and format of the messages used by a client to access a directory
- LDAP can be used as access protocol for a large number of databases
- It is the internal protocol used by the EGEE/LCG services to share information

The Glue Schema

- A Schema describes the attributes and the types of the attributes associated with data objects
- The offered data conforms to the **GLUE Schema**
- Grid Laboratory for a Uniform Environment
- The GLUE Schema activity aims to define a common conceptual data model to be used for Grid resources monitoring and discovery
- There are three main components of the GLUE Schema, they describe the attributes and the value of Site information
  - The Computing Element
  - The Storage Element
  - Network Monitoring

R-GMA: New System

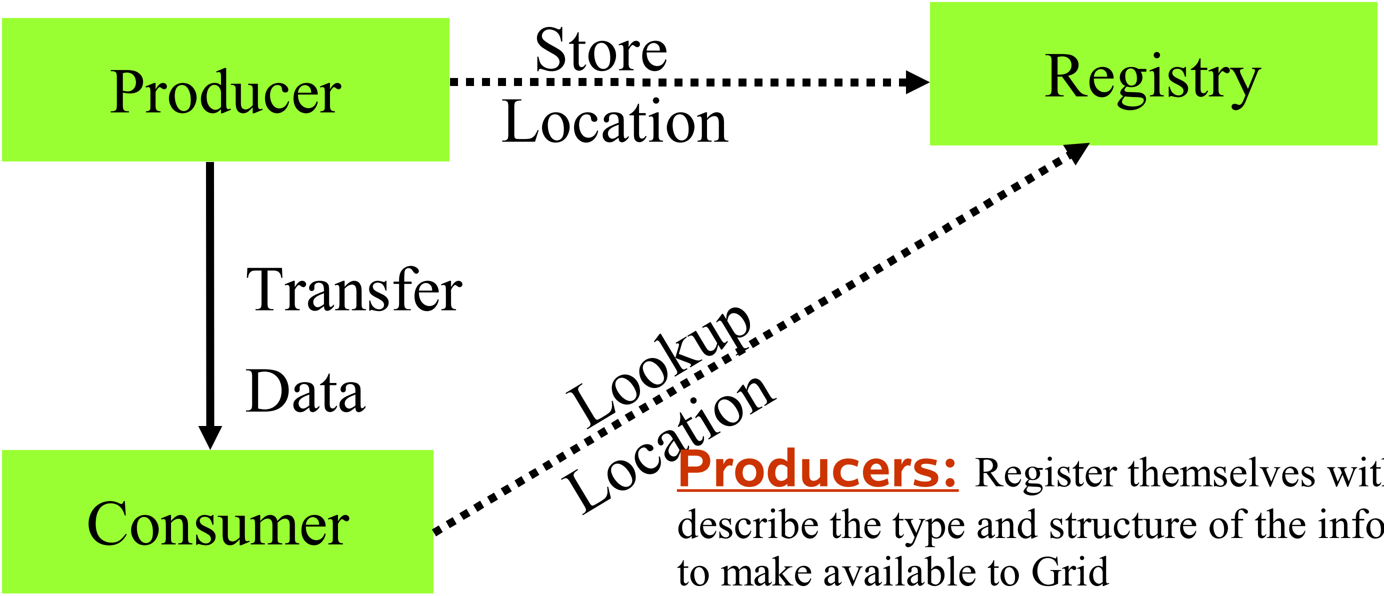
- Disadvantages of the old system:
  - LDAP does not allow to query information from different entries
  - MDS is not flexible enough to allow for dynamic publication of data from user applications

- Advantages of the new system:
  - R-GMA is quite flexible and allows cross queries between different entries
  - Anyone can introduce new information in the system in a easy way
  - It is quite dynamic with new Producers of information being notified by existing Consumers

R-GMA: Characteristics

- GMA (Grid Monitoring Architecture)
  - From GGF (Global Grid Forum)
  - Very simple; it does not define:
    - Data model
    - Data transfer mechanism
    - Registry implementation
- R-GMA (Relational GMA): Relational implementation
  - Powerful data model and query language
  - All data modelled as tables
  - SQL as query language. It can express most queries in one expression
  - You have a Relational DB for each VO

R-GMA Architecture



Producers: Register themselves with the Registry and describe the type and structure of the information they want to make available to Grid

Consumers: Query the Registry to find out the information available and locate Producers which provide such information. They can connect directly the Producers

Registry: General collector, its arrow line represents the main flow of data

Data managements : general concepts

- What does “Data Management” mean ?
  - Users and applications produce and require data
  - Data may be stored in Grid files
  - Granularity is at the “file” level (no data “structures”)
  - Users and applications need to handle files on the Grid
- Files are stored in appropriate permanent resources called “Storage Elements” (SE)
  - Present almost at every site together with computing resources
  - We will treat a storage element as a “black box” where we can store data
    - Appropriate data management utilities/services hide internal structure of SE
    - Appropriate data management utilities/services hide details on transfer protocols

Data Management Services

Enabling Grids for E-science in Europe

- Storage Element
  - Storage Resource Manager not provided by gLite
  - POSIX-I/O gLite-I/O
  - Access protocols gsiftp, https, rfio, …

rely on existing implementations

- Catalogs
  - File Catalog
  - Replica Catalog
  - File Authorization Service
  - Metadata Catalog

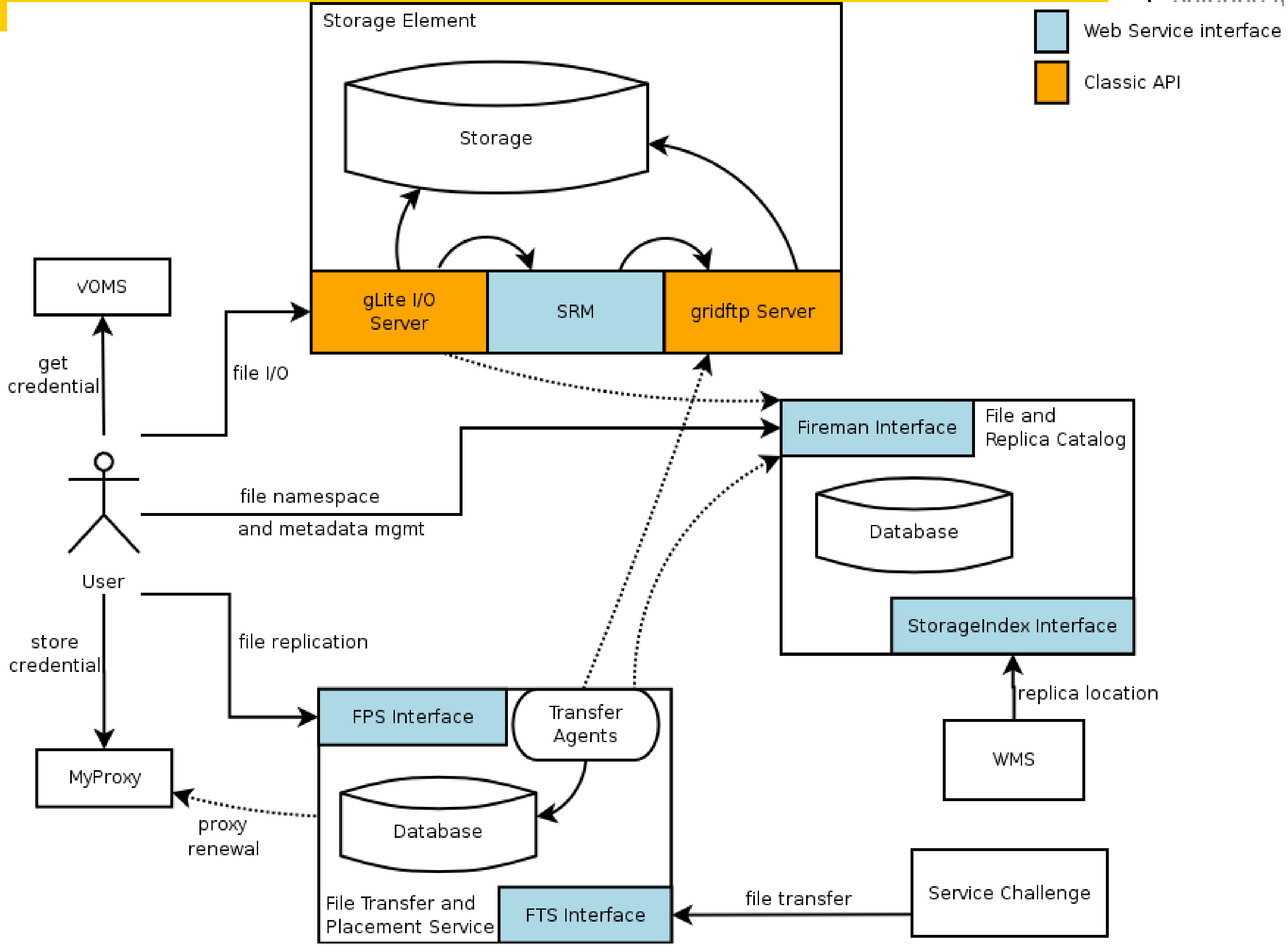
gLite FiReMan Catalog

(MySQL and Oracle)

gLite Standalone Metadata Catalog

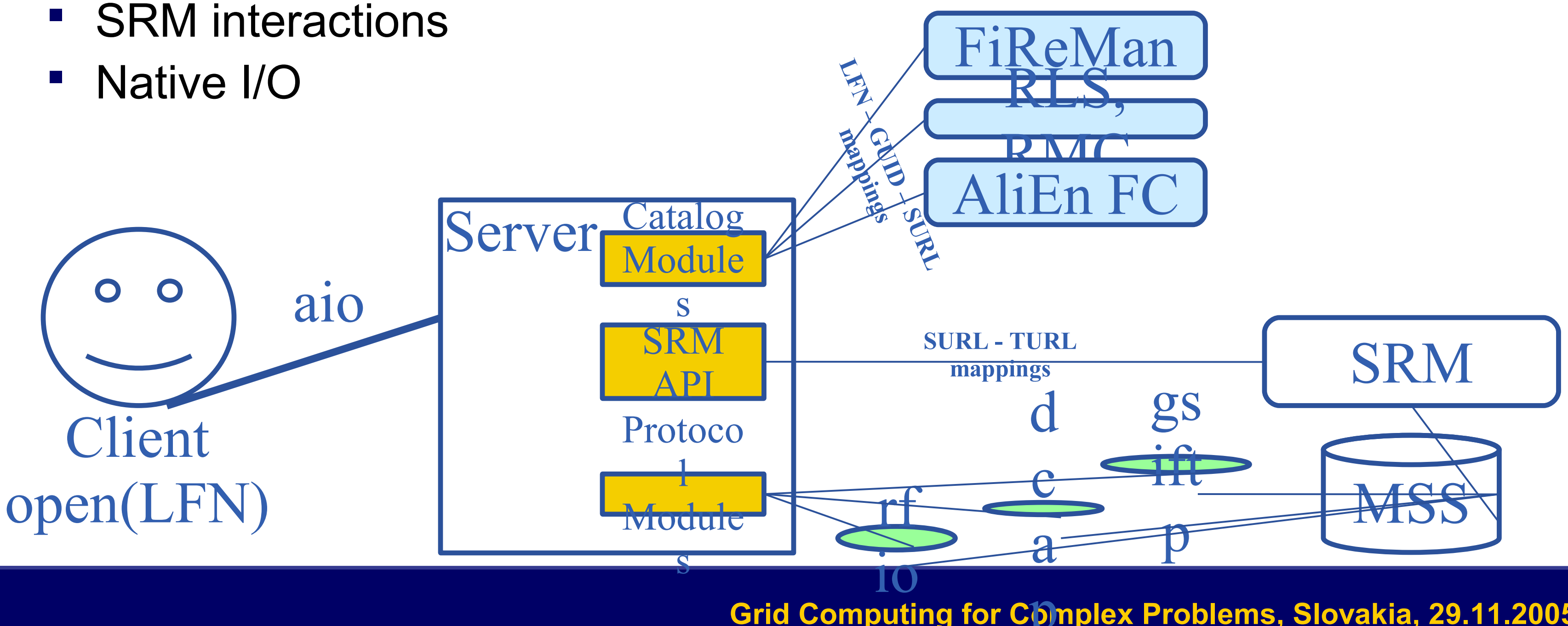
- File Transfer
  - Data Scheduler planned for Release 2
  - File Transfer Service gLite FTS and glite-url-copy
  - File Placement Service gLite FPS

Interaction Overview

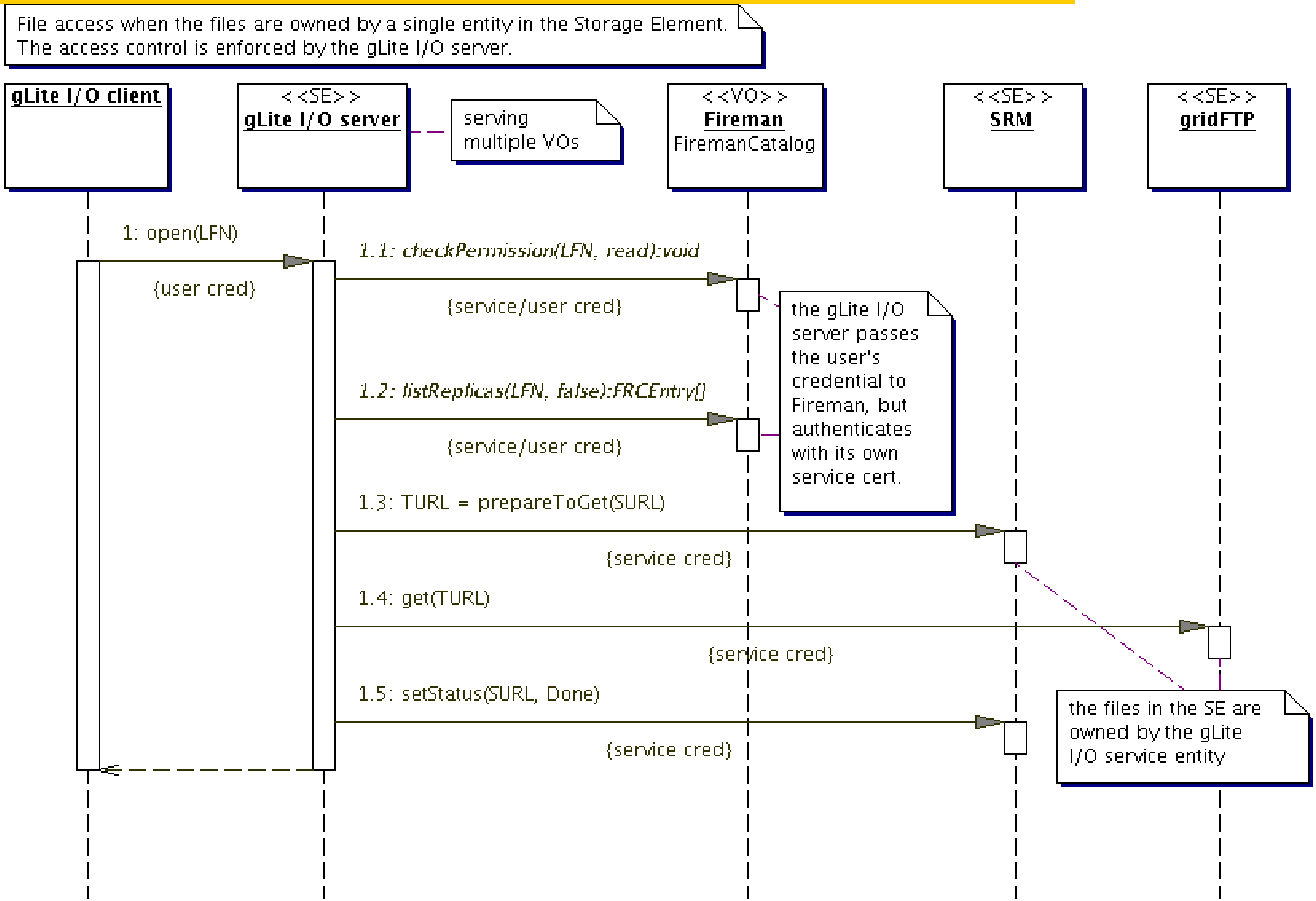


File Access Overview

- Client only sees a simple API library and a Command Line Interface
  - GUID or LFN can be used, i.e. open("/grid/myFile")
- GSI Delegation to gLite I/O Server
- Server performs all operations on User's behalf
  - Resolve LFN/GUID into SURL and TURL
- Operations are pluggable
  - Catalog interactions
  - SRM interactions
  - Native I/O



File Open



File access when the files are owned by a single entity in the Storage Element.
The access control is enforced by the gLite I/O server.
gLite I/O client
<<SE>>
gLite I/O server
serving multiple VOs
<<VO>>
Fireman
FiremanCatalog
<<SE>>
SRM
<<SE>>
gridFTP
1: open(LFN)
{user cred}
1.1: checkPermission(LFN, read):void
{service/user cred}
the gLite I/O server passes the user's credential to Fireman, but authenticates with its own service cert.
1.2: listReplicas(LFN, false):FRCEntry[]
{service/user cred}
1.3: TURL = prepareToGet(SURL)
{service cred}
1.4: get(TURL)
{service cred}
1.5: setStatus(SURL, Done)
{service cred}
the files in the SE are owned by the gLite I/O service entity

MSS and SRM

- gLite IO server relies against a Mass Storage System implementing SRM interface
- gLite IO server communicates with MSS through SRM
- SRM is not provided by gLite !
- Tested MSS are, till now, CASTOR and dCache
- Full support to functionalities depending also from MSS
- Installing and configuring MSS is apart from gLite issues
- How to and guides to do so

http://egee-na4.ct.infn.it/wiki/out_pages/dCache-SRM.html
http://storage.esc.rl.ac.uk/documentation/html/D-Cache-Howto

Basic IO commands

Copy a local file to Storage Element

- glite-put *local-file lfn:///lfn-name*

Copy a file from Storage element

- glite-get *lfn:///lfn-name localfile-path*

Remove a file from Storage element

- glite-rm *lfn:///lfn-name*

  if the lfn is the last replica, file entry is removed from the catalog

Before of executing glite-put or glite-rm, tools check that user has rights to perform requested operation.

Some details on protocols

- Data channel protocol: mostly gridFTP (**gsiftp**)
  - secure and efficient data movement
  - extends the standard FTP protocol
  - Public-key-based Grid Security Infrastructure (GSI) support
  - Third-party control of data transfer
  - Parallel data transfer
- Other protocols are available, especially for File I/O
  - rfio protocol:
    - for CASTOR SE (and classic SE)
  - gsidcap protocol:
    - for secure access to dCache SE
  - file protocol:
    - for local file access

Cataloguing Requirements

- Need to keep track of the location of copies (replicas) of Grid files
- Replicas might be described by attributes
  - Support for METADATA
  - Could be “system” metadata or “user” metadata
- Potentially, millions of files need to be registered and located
  - Requirement for performance
- Distributed architecture might be desirable
  - scalability
  - prevent single-point of failure
  - Site managers need to change autonomously file locations

File Catalogs in EGEE/LCG

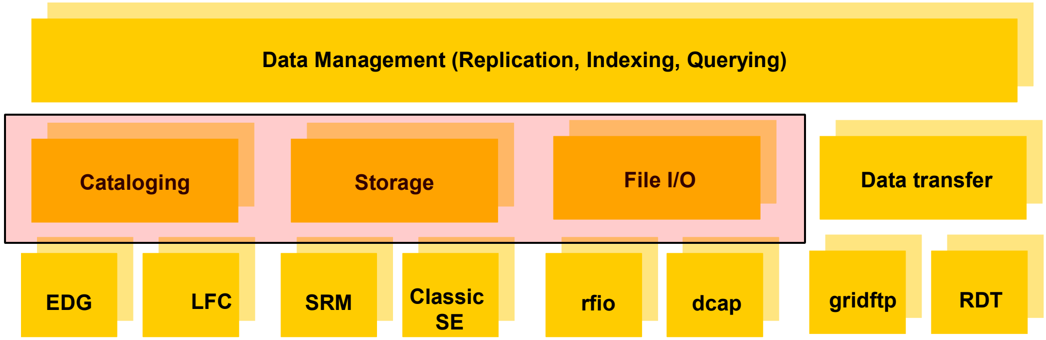
- Access to the file catalog
  - The DM tools and APIs and the WMS interact with the catalog
    - Hide catalogue implementation details
  - Lower level tools allow direct catalogue access
- EDG's Replica Location Service (RLS)
  - Catalogs in use in LCG-2
  - Replica Metadata Catalog (RMC) + Local Replica Catalog (LRC)
  - Some performance problems detected during LCG Data Challenges
- File and Replica Catalog (AliEn)
  - Better performance and scalability
- EGEE FiReMan

EGEE File Catalog

- Fixes performance and scalability problems seen in EDG Catalogs
  - Cursors for large queries
  - Timeouts and retries from the client
- Provides more features than the EDG Catalogs
  - User exposed transaction API (+ auto rollback on failure of mutating method call)
  - Hierarchical namespace and namespace operations (for LFNs)
  - Integrated GSI Authentication + Authorization
  - Access Control Lists (Unix Permissions and POSIX ACLs)
  - Checksums
- Interaction with other components
  - Supports Oracle and MySQL database backends
  - Integration with GFAL and lcg_util APIs complete
  - New specific API provided

Grid File Access Library

- GFAL is a library to provide access to Grid files
  - File I/O, Catalog Interaction, Storage Interaction
- Abstraction from specific implementations
- Transparent interaction with the information service, the file catalogs…
- Single shared library in threaded and unthreaded versions
  - libgfal.so, libgfal_pthr.so
- Single header file
  - gfal_api.h



Data transfer and replication

- Data movements capability (should be…) provided by
  - Data scheduler (DS) (top-level)
  - File Placements Services (FPS)
  - File Transfer Service (FTS)
- DS keeps track of data movement request submitted by clients
- FPS pools DS fetching transfers with local site as destination, updating catalog
- FTS maintains state of transfers
- Data scheduler has not been released with gLite 1.x
- So actually no replica can be performed with gLite DMS

Distribution Mechanism

- Data Scheduler (global and local schedulers)
  - Global scheduler (VO-specific) takes requests like
    - Copy set of files from A to B
    - Make set of files available at C
    - Upload files from GSIFTP server to D
    - Delete files
    - *Maybe also metadata operations*
  - Local scheduler fetches tasks from known global schedulers
    - Coupled tightly to a local transfer service
    - Manage transfer where the local site is a target
    - Assure atomicity of transfer and catalog operations
- Transfer Service
  - Queue data transfers to/from a given Storage Element (SRM)
  - Receives jobs from local scheduler
  - Manages transfers through a set of states

Workload Management

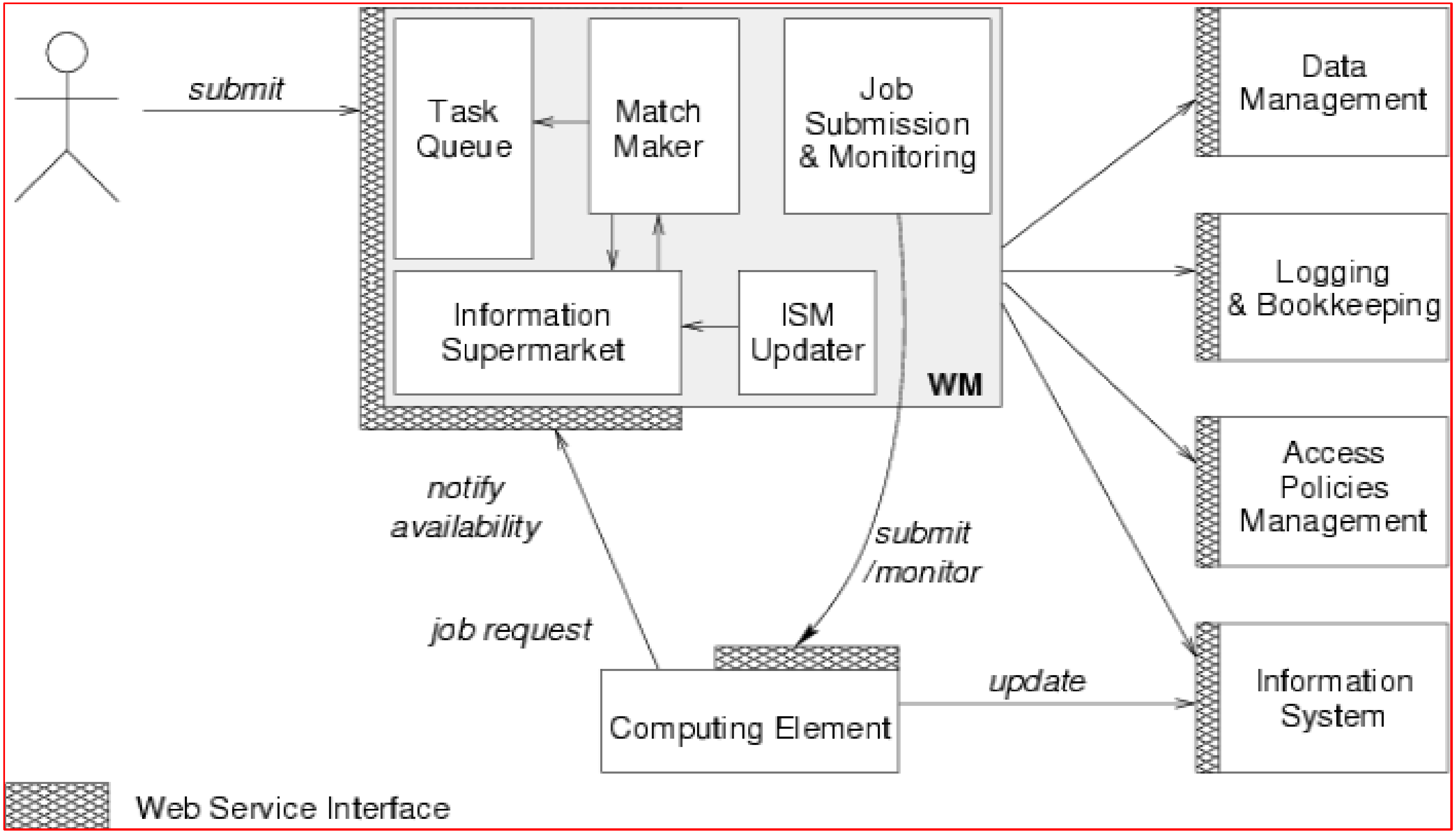
- The user submits jobs via the **Workload Management System**
- The Goal of WMS is the **distributed scheduling and resource management in a Grid environment**.
- What does it allow Grid users to do?
  - To submit their jobs
  - To execute them
  - To get information about their status
  - To retrieve their output
- The WMS tries to optimize the usage of resources
- As well as execute user jobs as fast as possible

Job Management Services

- main services related to job management/execution are
  - computing element
    - job management (job submission, job control, etc.), but it must also provide
    - provision of information about its characteristics and status
  - workload management
    - core component discussed in details
  - accounting
    - special case as it will eventually take into account
      - computing, storage and network resources
  - job provenance
    - keep track of the definition of submitted jobs, execution conditions and environment, and important points of the job life cycle for a long period
      - debugging, post-mortem analysis, comparison of job execution
  - package manager
    - automates the process of installing, upgrading, configuring, and removing software packages from a shared area on a grid site.
      - extension of a traditional package management system to a Grid

- Services communicate with each other as the job request progresses through the system
  - a consistent view of the status of the job is maintained by Logging and Bookkeeping service

WMS in gLite


submit
Task Queue
Match Maker
Job Submission & Monitoring
Information Supermarket
ISM Updater
WM
Data Management
Logging & Bookkeeping
Access Policies Management
Information System
notify availability
job request
submit /monitor
update
Computing Element
Web Service Interface

WMS Match Making

- The Match Maker (MM) is the core component of WMS.
- It has to find the best suitable computing resource (CE) where the job will be executed
- It interacts with Data Management service and Information System
    - They supply MM with all the information required for the resolution of the matches
- The CE chosen by MM has to match the job requirements (e.g. runtime environment, data access requirements, and so on)
- If 2 or more CEs satisfy all the requirements, the one with the best Rank is chosen

Direct Job submission

- The WMS has to deal with three possible scenarios.

Scenario 1: Direct Job Submission

- Job is scheduled on a given CE (specified in the edg-job-submit command via –r option)
- MM doesn’t perform any matchmaking algorithm

Brokered Job Submission, No InputData

Scenario 2: Job Submission without data-access Requirements

- Neither CE nor input data are specified.
- MM starts the matchmaking algorithm, which consists of two phases:
  - Requirements check (MM contacts the IS to check which CEs satisfy all the requirements)
  - If more than one CE satisfies the job requirements, the CE with the best rank is chosen by the MM

Brokered Job Submission, Grid Data

Scenario 3: CE is not specified in the JDL

- WMS contacts Data Management service to find out which SE’s have copies of the requested input data sets
- MM makes best effort match between
  - Computing resources for which user is authorized
  - SE’s “nearby” which can provide the requested data sets via the requested transfer protocol
  - Any optional output SE specified in the job description
- MM strategy consists of submitting jobs close to data!
- The main two phases of the match making algorithm remain unchanged:
  - Requirements check
  - Rank computation
- The matchmaking is only performed for CEs satisfying the data-access requirements (i.e. which are close to data)

Proxy Renewal

- Why?
  - To avoid job failure because it outlived the validity of the initial proxy
- WMS support automatic proxy renewal mechanism as long as the user credentials are handled by a proxy server.
  - The Proxy is automatic renewed by WMS without user intervention for all the job life

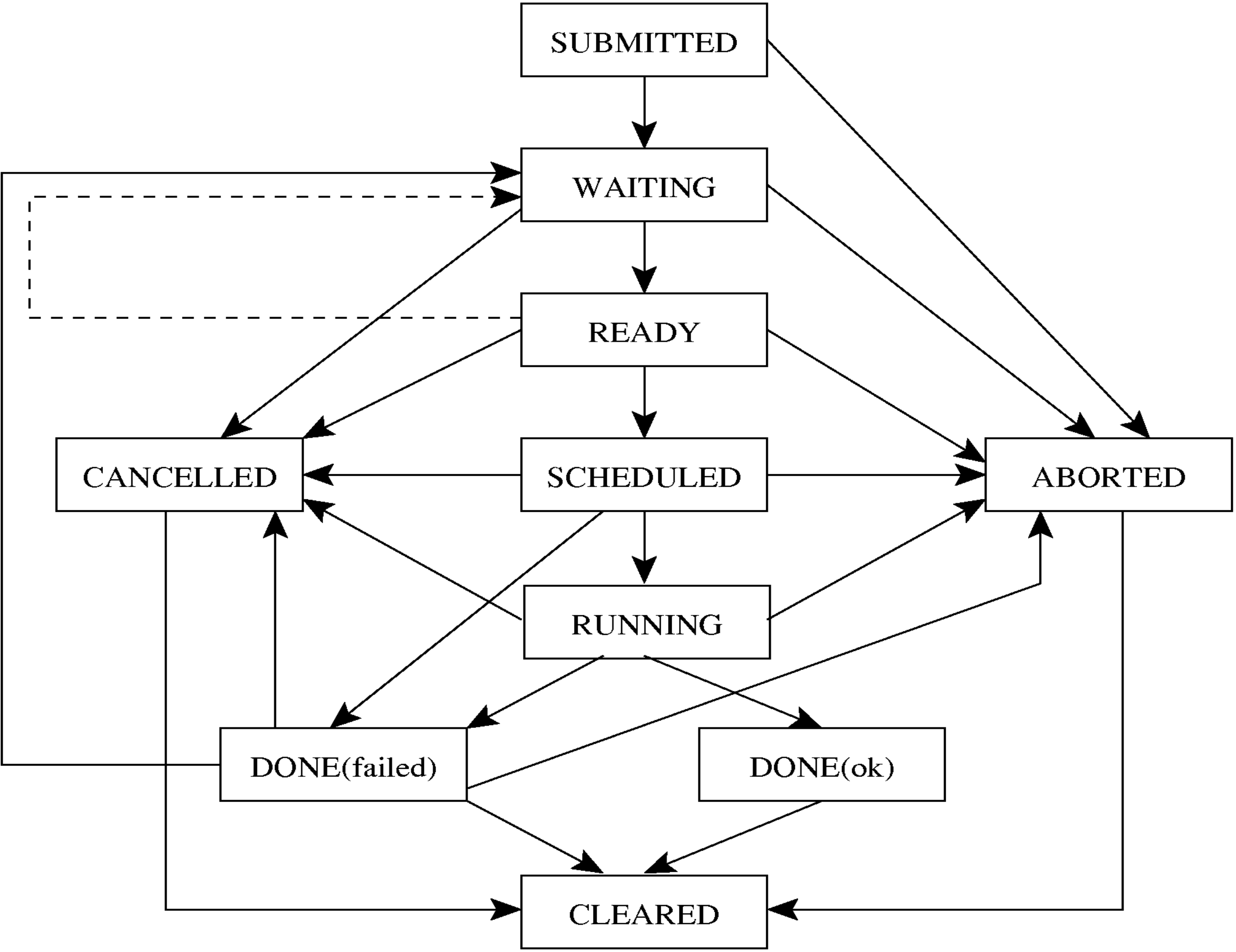
Logging and Bookkeeping

- Purpose
  - track Grid jobs during their life
  - capture passing job control between Grid components
  - provide user with high-level view on job state
  - short-term post-mortem analysis
- Main features
  - important points in job life gathered as L&B events
    - transfer of job between grid components
    - finding suitable computing element
    - starting/terminating execution
  - events delivered to L&B server reliably but in non-blocking way
  - job state computed by fault-tolerant state machine
  - user can query job state or register for receiving notifications

Logging and Bookkeeping data

- Jobs
  - primary entity of interest
  - assigned unique identifier
  - all data in L&B are related to jobs
- Events
  - record information on points of interest in job life
  - the only way to enter information into L&B
  - specific types (eg. Transfer, Match, Done)
  - carry additional, both generic and type-specific attributes (eg. timestamp, destination CE)
  - both “system” (job life) and user (arbitrary info)
- Job states
  - computed from events as they arrive
  - hard-coded state machine
  - event attributes mapped to job-state ones
  - fault-tolerance – missing or delayed events

Possible job states



Logging and Bookkeeping - usage

- Consumers
  - one-time queries on job state or raw events
    - all my running jobs
    - jobs that failed at CE X last week
  - notifications on specified job states changes
    - currently restricted on concrete jobid(s)
    - eg. tell me whenever one of these twenty jobs fails
- L&B Proxy – persistent storage of job state for WMS
  - gLite Workload Manager processing depends on job state
    - consistency checks
    - original job description retrieval on job resubmission
- Job statistics, statistics about successfulness of WMS/EGEE
- Future work - tracking other entities
  - Condor jobs
  - data transfer jobs
  - resource reservations

Job Provenance

- Motivation
  - preparing job submission requires a lot of work
  - the work is not completely reflected in job results
  - preserve information on Grid jobs
    - what were the executed jobs
    - job execution environment (installed software etc.)
    - track of execution (e.g. number of failures and resubmission)
  - allow data-mining in this information and assisted job re-running
    - "What were jobs of this VO, run on input data X, using (faulty) software Y?"

Job Provenance

- Gathered data
  - scalability issues
    - strict limits on reasonable JP record size
    - record volatile data only
  - job inputs
    - job description (JDL) as submitted to RB
    - miscellaneous input files (input sandbox)
    - do not copy input files from remote storage elements
  - job execution track
    - L&B data (when and where was the job planned and executed etc.)
    - “measurements” on CE (installed software, environment)
    - accounting data (DGAS)
  - user annotations (at run-time or afterwards)

Job Provenance – status and plans

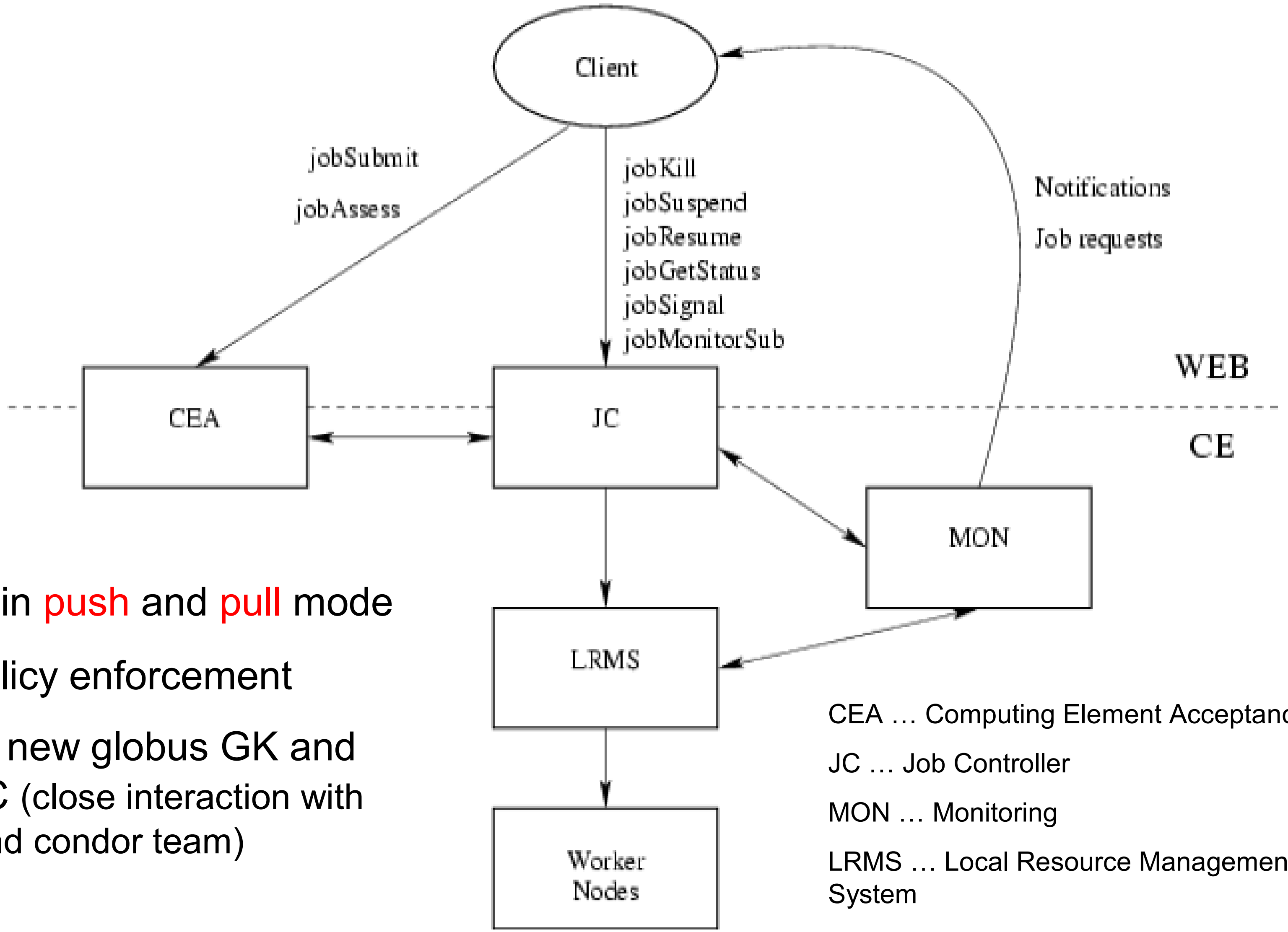
- Current status
  - implementation done, included in gLite 1.5 RC
  - supported information sources: L&B and input sandboxes
  - deployed at development testbed, receiving first real jobs
- Immediate plans
  - deployment in larger scale
  - user-side CLI and integration in gLite WMS GUI to support re-running jobs
  - more complex authorization
- Longer-term plans
  - integration with Grid accounting (DGAS)
  - support for non-gLite-WMS jobs (CREAM CE, Condor)
  - interface to gLite Storage Element

Computing Element

- Service representing a computing resource
- Refers to a Cluster of Computational Resources, also heterogeneous.
- Main functionality: job management
  - Run jobs
  - Cancel jobs
  - Suspend and resume jobs, send signals to them, get status or notification.
  - Provide info on “quality of service”
    - How many resources match the job requirements ?
    - What is the estimated time to have the job starting its execution ? (ETT)

- Used by the WM or by any other client (e.g. end-user)
- CE architecture accommodated to support both push and pull model
  - Push model: the job is pushed to the CE by the WM
  - Pull model: the CE asks the WM for jobs
- These two models are somewhat mirrored in the resource information flow
  - In order to 'pull' a job a resource must choose where to 'push' information about itself (CE Availability message)

gLite Computing Element



- Works in push and pull mode
- Site policy enforcement
- Exploit new globus GK and CondorC (close interaction with globus and condor team)

CEA … Computing Element Acceptance

JC … Job Controller

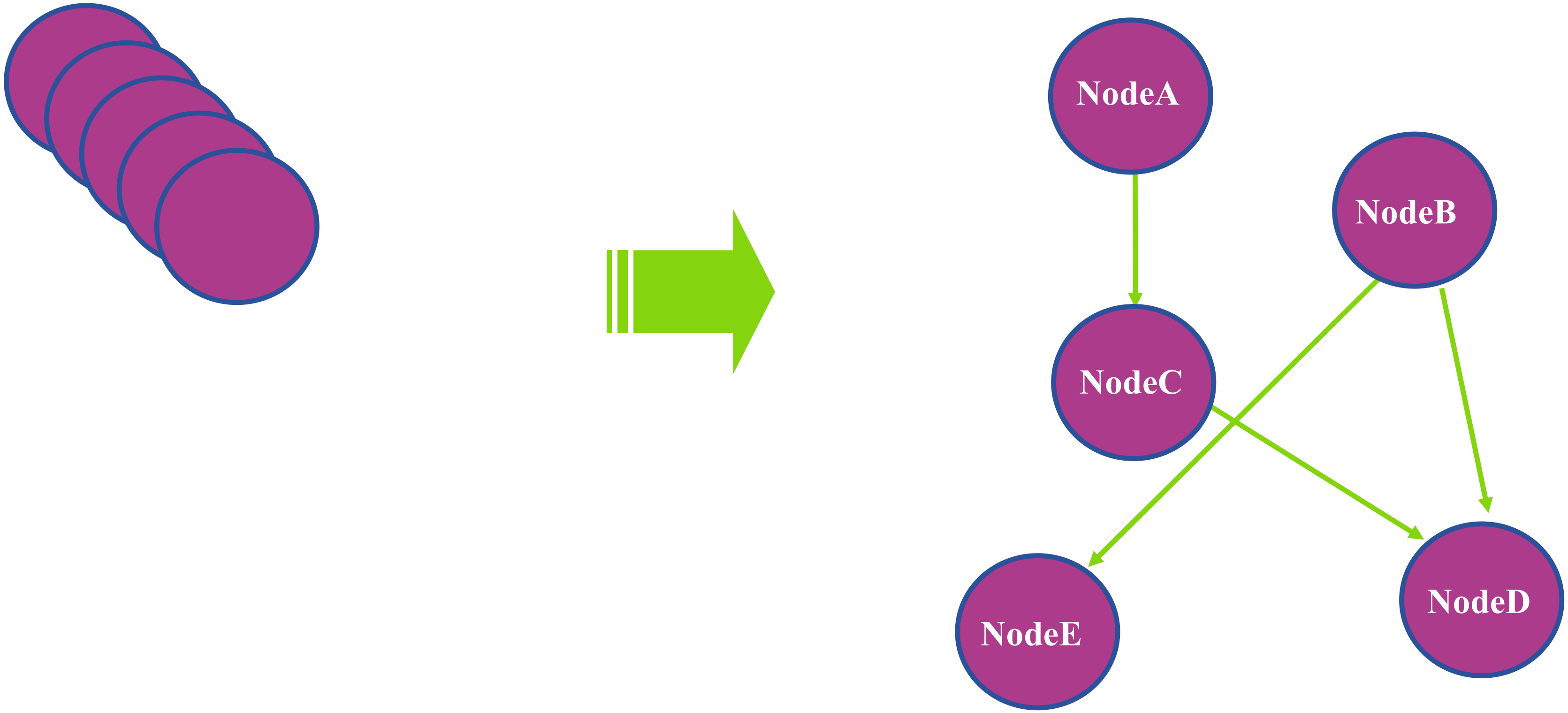
MON … Monitoring

LRMS … Local Resource Management System

Directed Acyclic Graph (DAG)

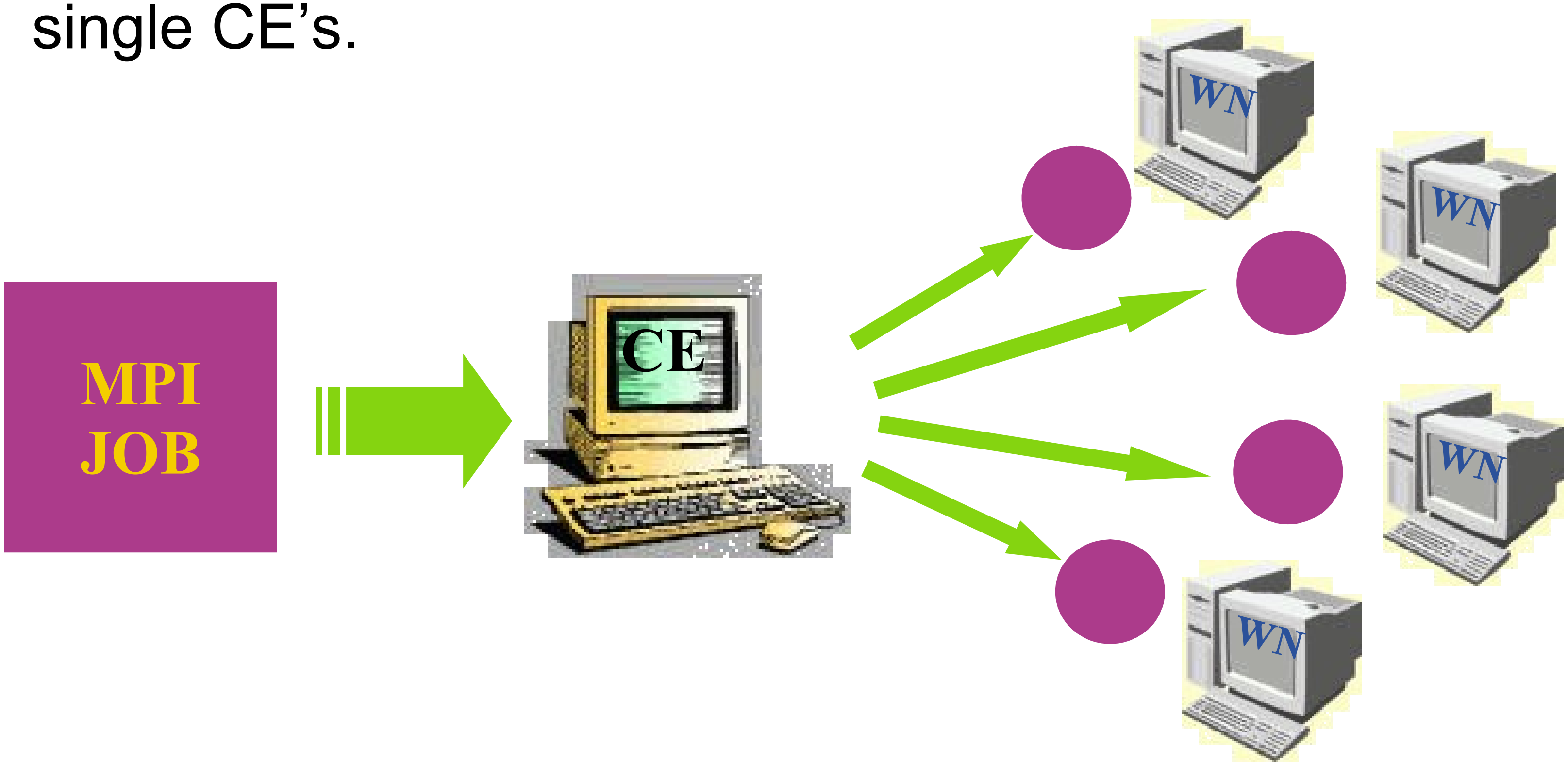
- A DAG represents a set of jobs:

  Nodes = *Jobs* *Edges* = *Dependencies*



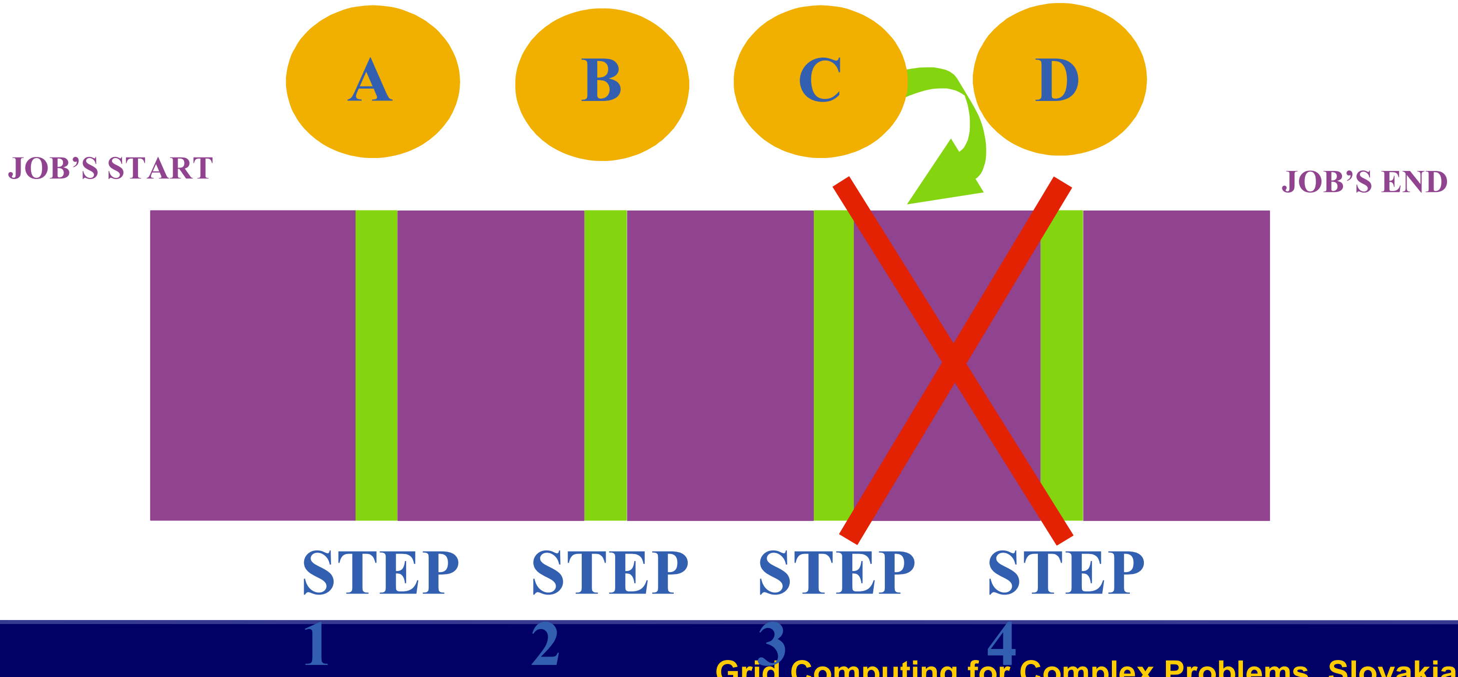
Message Passing Interface (MPI)

- The MPI job is run in parallel on several processors.
- Libraries supported for parallel jobs: MPICH.
- Currently, execution of parallel jobs is supported only on single CE's.



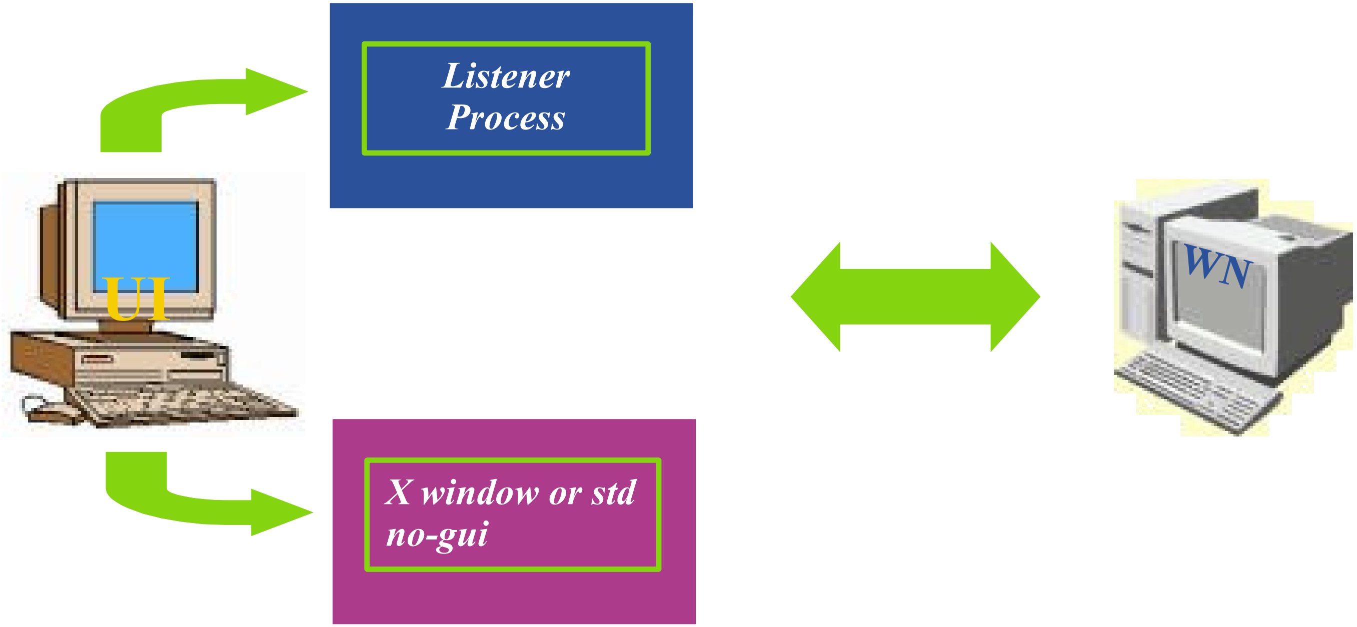
Logical Checkpointable Job

- It is a job that can be decomposed in several steps;
- In every step the job state can be saved in the LB and retrieved later in case of failures;
- The job can start running from a previously saved state instead from the beginning again.



Interactive Job

- It is a job whose standard streams are forwarded to the submitting client.
- The DISPLAY environment variable has to be set correctly, because an X window is open.



Job Preparation

- You need to provide
  - A complete (enough) job description
    - What program?
    - What data?
    - Any requirements on OS, installed software, ??
  - Possibly a program
    - You're submitting in *unknown territory!*
    - Program portably!
    - Don't rely on hard-coded paths or special locations
    - The program you send may not even be in $HOME!
  - Perhaps some input data
  - Perhaps instructions on what to do with the output

How to Write a Job Description

- Here is a minimal job description (call it mytest.jdl)

```
Executable = “/bin/echo”;
Arguments = “Goede Morgen”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
OutputSandbox = {“stderr.log”, “stdout.log”};
```

- We specified
  - The program to run and its arguments
  - Directed the standard error and output streams to files
  - Told it what to do with the output

Job Description Language (JDL)

- Based upon Condor’s *CLASSified ADvertisement language (ClassAd)*
- ClassAd is an extensible language
- Sequence of attributes (key,value pairs) separated by semi-colons.

```
Executable = “/bin/echo”;
Arguments = “Goede Morgen”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
OutputSandbox = {“stderr.log”, “stdout.log”};
```

Types of Attributes

- The supported attributes are grouped in two categories:
  - *Job*

    Define the job itself
  - Resources
    - Taken into account by the RB for carrying out the matchmaking algorithm
    - *Computing Resource (Attributes)*

      Used to build expressions of Requirements and/or Rank attributes by the user

      Have to be prefixed with “other.”
    - *Data and Storage resources (Attributes)*

      Input data to process, SE where to store output data, protocols spoken by application when accessing SEs

Resource Attributes

- **Requirements**
  - Job requirements on computing resources
  - Specified using attributes of resources published in the Information System
  - If not specified, default value defined in UI configuration file is considered
    - Default: other.GlueCEStateStatus == "Production" (the resource has to be in the Production grid)
- **Rank**
  - Expresses preference (how to rank resources that have already met the Requirements expression)
  - Specified using attributes of resources published in the Information Service
  - If not specified, default value defined in the UI configuration file is considered
    - Default: - other.GlueCEStateFreeCPUs (the highest number of free CPUs)

“Data” Attributes

- **InputData** (optional)
  - Refers to data used as input by the job: these data are published in the Replica Catalog and stored in the SEs)
  - PFNs and/or LFNs
- **DataAccessProtocol** (mandatory if InputData specified)
  - The protocol or the list of protocols which the application is able to speak with for accessing ***InputData*** on a given SE
- **OutputSE** (optional)
  - The hostname of the output SE
  - RB uses it to choose a CE that is compatible with the job and is close to SE
- **OutputData** (optional)
  - Output Data that will be registered at the end of the job

Example JDL File

```
Executable = “gridTest”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
InputSandbox = {“/home/joda/test/gridTest”};
OutputSandbox = {“stderr.log”, “stdout.log”};
InputData = “lfn:testbed0-00019”;
DataAccessProtocol = “gridftp”;
Requirements = other.Architecture==“INTEL” && \
          other.OpSys==“LINUX” && other.FreeCpus >=4;
Rank = “other.GlueHostBenchmarkSF00”;
```

VOCE - Generic description

- VOCE - Virtual Organization for Central Europe

VO is dynamic pool of resources & users from different domains grouped together for a particular purpose

- provides **complete grid infrastructure** under EGEE wings
- officially registered as currently the one and only “Regional VO” for Central European (CE) region
- based on **regional principle**
  - VOCE spans the whole CE Federation
  - core services operated by CESNET
  - resources are provided by several institutions across the CE (these resources are available to all users registered in VOCE)

VOCE - Generic description (2)

- VOCE - Virtual Organization for Central Europe
  - **fully production environment**
    - VOCE environment allows Grid newcomers to get quickly first experience with Grid computing
    - simultaneously allows users to smoothly move to production use of the Grid in the same environment
  - **self-contained infrastructure**
    - designed not to rely on external services
    - currently on LCG middleware but move to gLite under way

VOCE aims

- VOCE - Virtual Organization for Central Europe
  - **incubator for new applications** and application areas
    - assistance in adapting a software for use on the Grid
    - even for applications that do not have any computing experience
    - outsourcing the burden of running an grid infrastructure to VOCE
  - **generic VO**
    - not bound to any particular application
    - interested in broad scale of application areas
    - can also mediate a collaboration with other similar project we are aware of (being deeply involved in many Grid projects/activites in Europe)

VOCE aims (2)

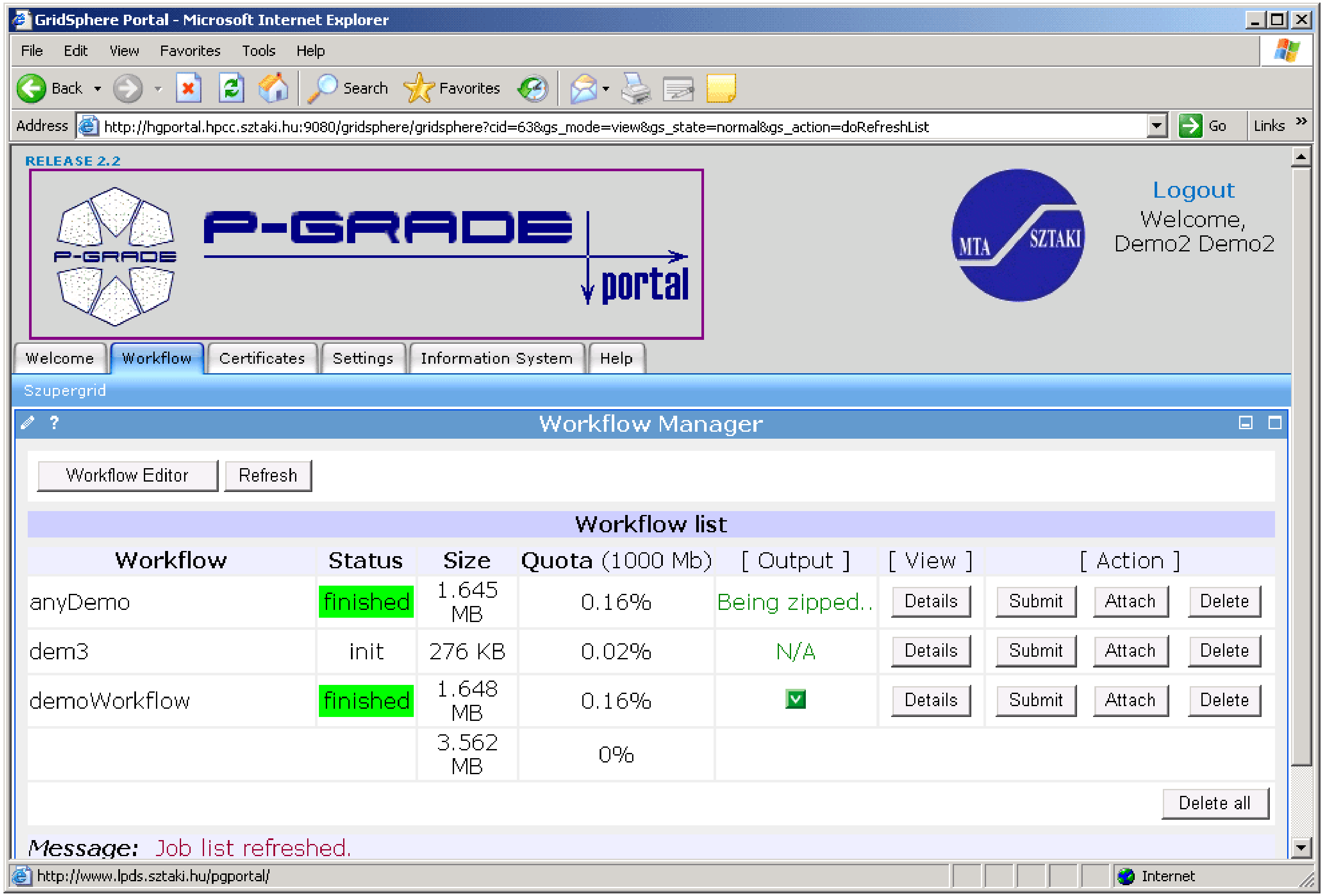
- VOCE - Virtual Organization for Central Europe
  - **allocated resources** (guaranteed)
  - basic level of security retained
    - no anonymous users
    - based on accredited CA's approved by the EuGridPMA body
  - **primary audience**
    - grid newcomers with no experience with Grid
    - small application groups that do not have resources and/or skills to build and operate their own infrastructure
  - easy registration
    - fully electronical using a web form
    - the applicant needs a certificate issued by a trusted CA

VOCE - Provided services

- VOCE - Virtual Organization for Central Europe
  - region specific solutions provided
    - generic flexible framework - **CHARON system** - that is available for application programs & jobs management
    - access to VOCE trough **P-GRADE portal**
    - **AFS installation** available for easy integration with local national Grid projects (currently used on Czech farms & UI)
    - **parallel execution support** (MPI)
  - GENIUS portal

VOCE - Provided services (2)

- P-GRADE – GUI to access VOCE



Address: http://hgportal.hpcc.sztaki.hu:9080/gridsphere/gridsphere?cid=63&gs_mode=view&gs_state=normal&gs_action=doRefreshList

Logout

Welcome, Demo2 Demo2

Workflow Manager

Workflow list

Workflow	Status	Size	Quota (1000 Mb)	[Output]	[View]	[Action]		
anyDemo	finished	1.645 MB	0.16%	Being zipped..	Details	Submit	Attach	Delete
dem3	init	276 KB	0.02%	N/A	Details	Submit	Attach	Delete
demoWorkflow	finished	1.648 MB	0.16%	✓	Details	Submit	Attach	Delete
		3.562 MB	0%					

Message: Job list refreshed.

http://www.lpds.sztaki.hu/pgportal/

Current VOCE status

- Infrastructure elements overview
  - elementary infrastructure (WMS, CE, WNs, SE) fully functional and configured to accept users from CE

UI	skurut4.cesnet.cz
MyProxy	skurut3.cesnet.cz
RB	skurut3.cesnet.cz
VOCE LDAP	meta-ldap.cesnet.cz
VOCE VOMS	odorn.ics.muni.cz

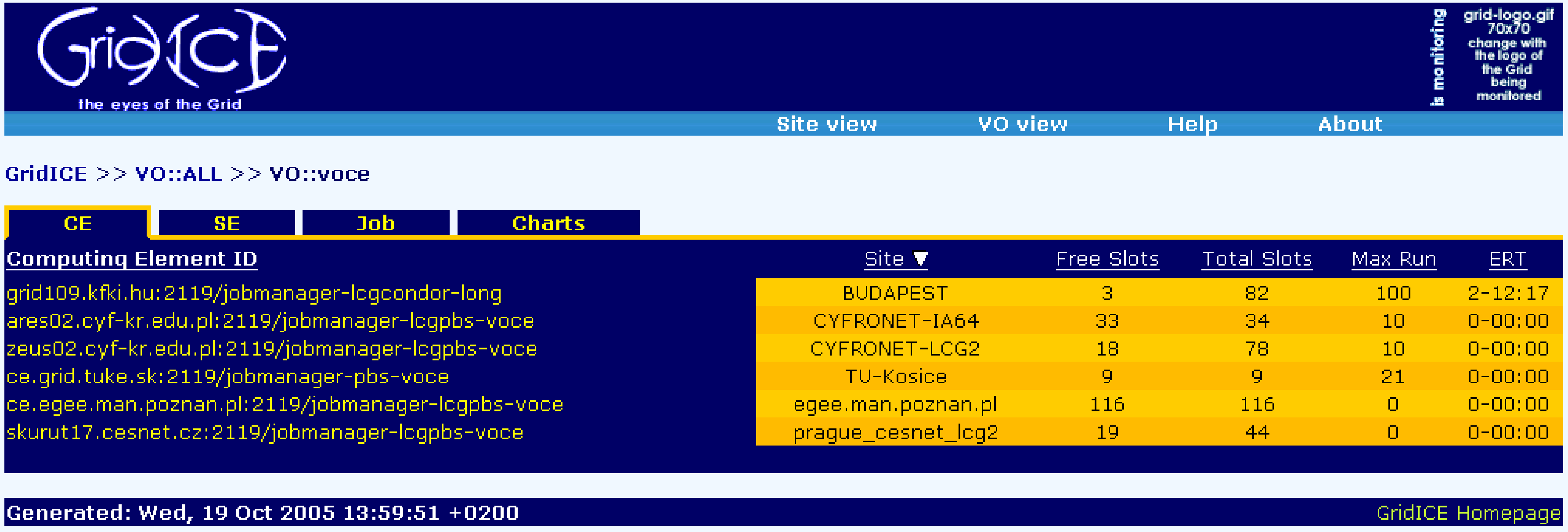
(temporary solution for tests)

GENIUS portal	https://skurut4.cesnet.cz/ (uses GILDA MyProxy server)

  - access to UI using GSISSH, UI account is created automatically after registration to VOCE

Current VOCE status (2)

- Summary of resources


GridICE
the eyes of the Grid
is monitoring
grid-logo.gif 70x70 change with the logo of the Grid being monitored
Site view
VO view
Help
About

GridICE >> VO::ALL >> VO::voce

CE | SE | Job | Charts

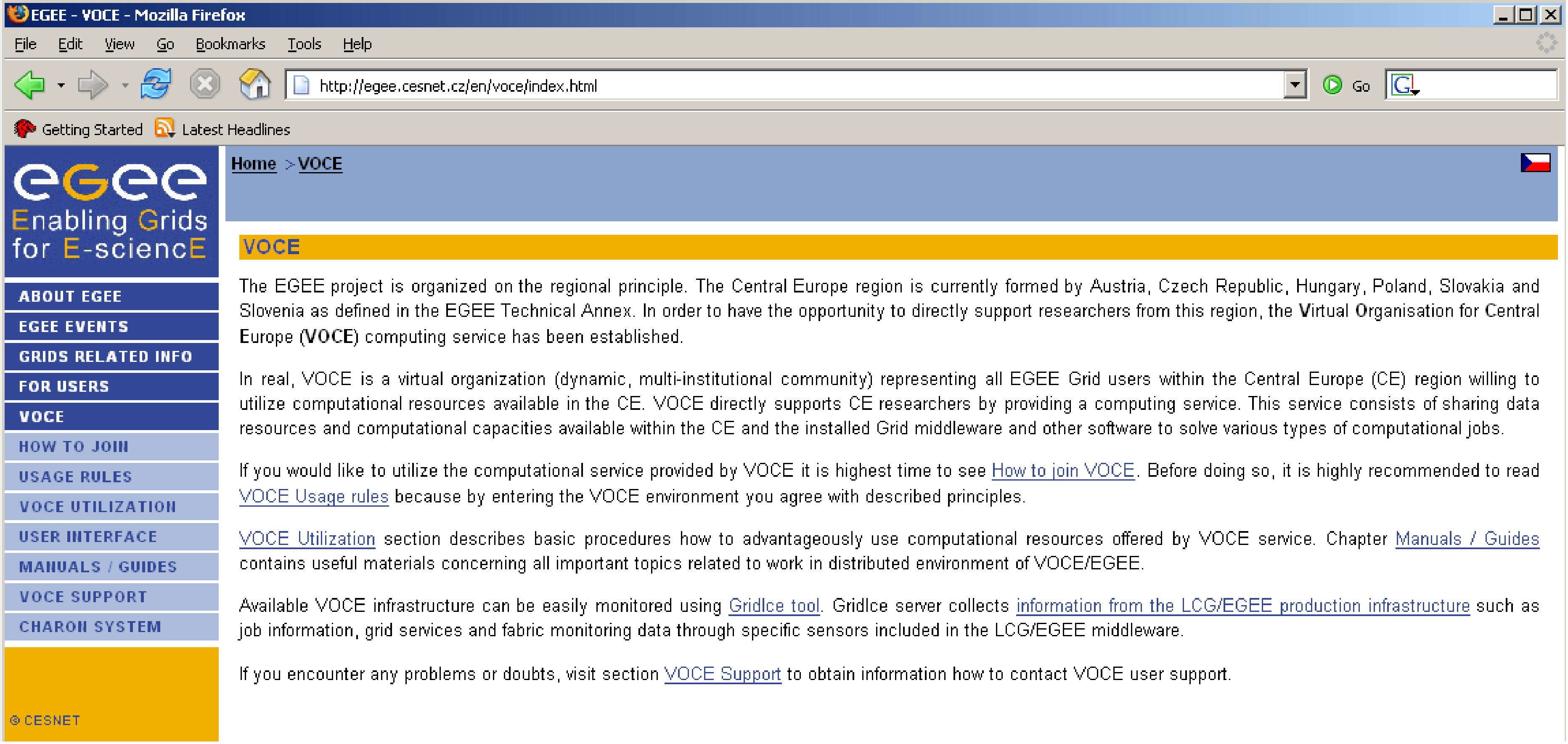
Computing Element ID	Site ▼	Free Slots	Total Slots	Max Run	ERT
grid109.kfki.hu:2119/jobmanager-lcgcondor-long	BUDAPEST	3	82	100	2-12:17
ares02.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-voce	CYFRONET-IA64	33	34	10	0-00:00
zeus02.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-voce	CYFRONET-LCG2	18	78	10	0-00:00
ce.grid.tuke.sk:2119/jobmanager-pbs-voce	TU-Kosice	9	9	21	0-00:00
ce.egee.man.poznan.pl:2119/jobmanager-lcgpbs-voce	egee.man.poznan.pl	116	116	0	0-00:00
skurut17.cesnet.cz:2119/jobmanager-lcgpbs-voce	prague_cesnet_lcg2	19	44	0	0-00:00

Generated: Wed, 19 Oct 2005 13:59:51 +0200 GridICE Homepage

- resources from CESNET (Czech Republic)
  PSNC, CYFRONET, ICM (Poland)
  II-SAS (Slovakia)
  KFKI (Hungary)
- almost 40 registered users from 10 institutes and 4 countries
- in total 539 CPUs, about 5.9 TB disk space

VOCE web

- Documentation
  - VOCE portal at http://egee.cesnet.cz/en/voce/



VOCE

The EGEE project is organized on the regional principle. The Central Europe region is currently formed by Austria, Czech Republic, Hungary, Poland, Slovakia and Slovenia as defined in the EGEE Technical Annex. In order to have the opportunity to directly support researchers from this region, the Virtual Organisation for Central Europe (**VOCE**) computing service has been established.

In real, VOCE is a virtual organization (dynamic, multi-institutional community) representing all EGEE Grid users within the Central Europe (CE) region willing to utilize computational resources available in the CE. VOCE directly supports CE researchers by providing a computing service. This service consists of sharing data resources and computational capacities available within the CE and the installed Grid middleware and other software to solve various types of computational jobs.

If you would like to utilize the computational service provided by VOCE it is highest time to see How to join VOCE. Before doing so, it is highly recommended to read VOCE Usage rules because by entering the VOCE environment you agree with described principles.

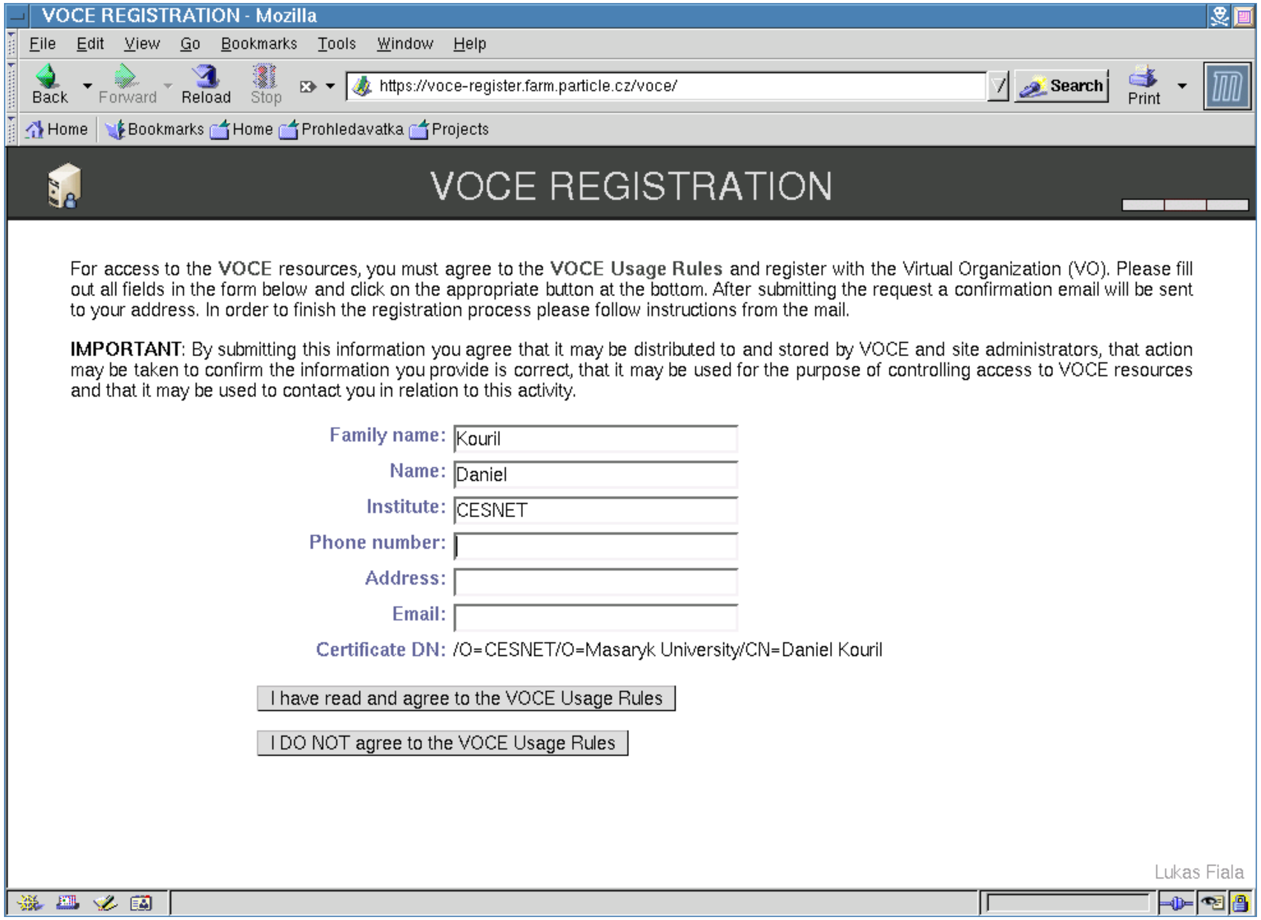
VOCE Utilization section describes basic procedures how to advantageously use computational resources offered by VOCE service. Chapter Manuals / Guides contains useful materials concerning all important topics related to work in distributed environment of VOCE/EGEE.

Available VOCE infrastructure can be easily monitored using GridIce tool. GridIce server collects information from the LCG/EGEE production infrastructure such as job information, grid services and fabric monitoring data through specific sensors included in the LCG/EGEE middleware.

If you encounter any problems or doubts, visit section VOCE Support to obtain information how to contact VOCE user support.

VOCE web (2)

- User registration
  - VOCE registration at http://voce-register.farm.particle.cz/


VOCE REGISTRATION - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
https://voce-register.farm.particle.cz/voce/
Search
Print
Home Bookmarks Home Prohledavatka Projects

VOCE REGISTRATION

For access to the VOCE resources, you must agree to the VOCE Usage Rules and register with the Virtual Organization (VO). Please fill out all fields in the form below and click on the appropriate button at the bottom. After submitting the request a confirmation email will be sent to your address. In order to finish the registration process please follow instructions from the mail.

IMPORTANT: By submitting this information you agree that it may be distributed to and stored by VOCE and site administrators, that action may be taken to confirm the information you provide is correct, that it may be used for the purpose of controlling access to VOCE resources and that it may be used to contact you in relation to this activity.

Family name:	Kouril
Name:	Daniel
Institute:	CESNET
Phone number:	
Address:	
Email:	
Certificate DN:	/O=CESNET/O=Masaryk University/CN=Daniel Kouril

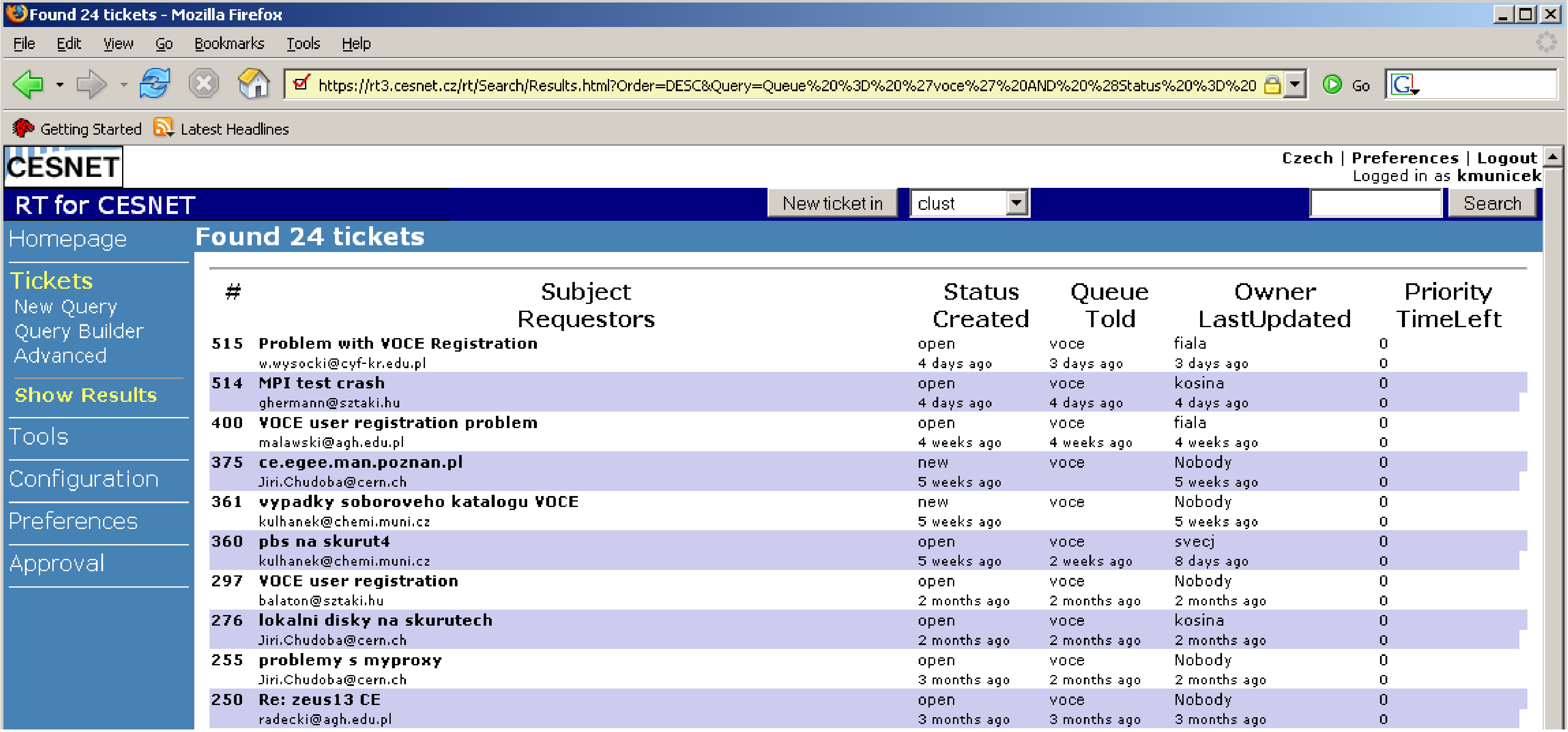
I have read and agree to the VOCE Usage Rules

I DO NOT agree to the VOCE Usage Rules

Lukas Fiala

VOCE RT system

- Request tracking
  - Send requests to voce@cesnet.cz


Found 24 tickets - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://rt3.cesnet.cz/rt/Search/Results.html?Order=DESC&Query=Queue%20%3D%20%27voce%27%20AND%20%28Status%20%3D%20
Go
Getting Started
Latest Headlines
CESNET
Czech | Preferences | Logout
Logged in as kmunicek
RT for CESNET
New ticket in
clust
Search
Homepage
Tickets
New Query
Query Builder
Advanced
Show Results
Tools
Configuration
Preferences
Approval

Found 24 tickets

#	Subject	Status	Queue	Owner	Priority
	Requestors	Created	Told	LastUpdated	TimeLeft
515	**Problem with VOCE Registration**	open	voce	fiala	0
	w.wysocki@cyf-kr.edu.pl	4 days ago	3 days ago	3 days ago	0
514	**MPI test crash**	open	voce	kosina	0
	ghermann@sztaki.hu	4 days ago	4 days ago	4 days ago	0
400	**VOCE user registration problem**	open	voce	fiala	0
	malawski@agh.edu.pl	4 weeks ago	4 weeks ago	4 weeks ago	0
375	**ce.egee.man.poznan.pl**	new	voce	Nobody	0
	Jiri.Chudoba@cern.ch	5 weeks ago		5 weeks ago	0
361	**vypadky soboroveho katalogu VOCE**	new	voce	Nobody	0
	kulhanek@chemi.muni.cz	5 weeks ago		5 weeks ago	0
360	**pbs na skurut4**	open	voce	svecj	0
	kulhanek@chemi.muni.cz	5 weeks ago	2 weeks ago	8 days ago	0
297	**VOCE user registration**	open	voce	Nobody	0
	balaton@sztaki.hu	2 months ago	2 months ago	2 months ago	0
276	**lokalni disky na skurutech**	open	voce	kosina	0
	Jiri.Chudoba@cern.ch	2 months ago	2 months ago	2 months ago	0
255	**problemy s myproxy**	open	voce	Nobody	0
	Jiri.Chudoba@cern.ch	3 months ago	2 months ago	2 months ago	0
250	**Re: zeus13 CE**	open	voce	Nobody	0
	radecki@agh.edu.pl	3 months ago	3 months ago	3 months ago	0