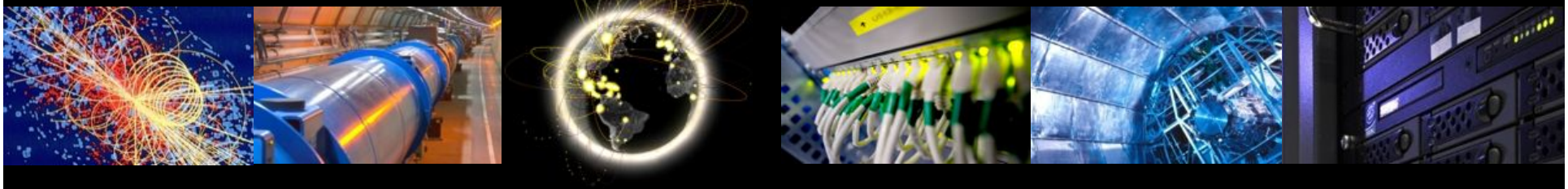
CERN-IT-GT

Grid Middleware gLite

Markus Schulz

August 2010
Openlab Summer Students



WLCG
Worldwide LHC Computing Grid

Overview

- Grid Computing?
- Constraints
- gLite
  - Security Model
  - Code Complexity
- Future

WLCG Worldwide LHC Computing Grid

Overview

- If you want to use gLite read the user guide:
- https://edms.cern.ch/document/722398/

- There is NO way around it ☺
- Unless you are in an experiment

WLCG Worldwide LHC Computing Grid

What is a Computing Grid?

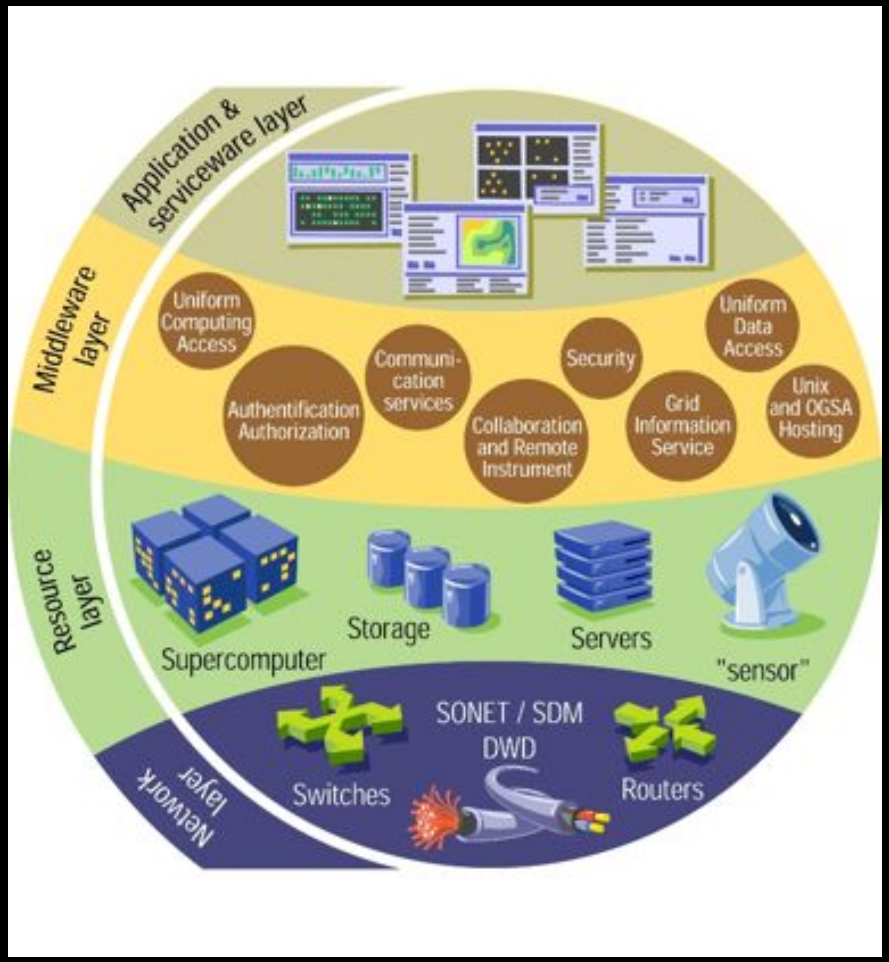
- There are many conflicting definitions
  - Has been used for several years for marketing…
    - Now they use Cloud
- Ian Foster and Karl Kesselman
  - "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. "
  - These are the people who started globus, the first grid middleware project
- From the user's perspective:
  - I want to be able to use computing resources as I need
  - I don't care who owns resources, or where they are
  - Have to be secure
  - My programs have to run there
- The owners of computing resources (CPU cycles, storage, bandwidth)
  - My resources can be used by any authorized person (not for free)
  - Authorization is not tied to my administrative organization
- **– NO centralized control of resources or users**

Constraints?

- The world is a fairly heterogeneous place
  - Computing services are extremely heterogeneous
- Examples:
  - Batch Systems (controlling the execution of your jobs)
    - LSF, PBS, TorQue, Condor, SUN-GridEngine, BQS, …..
    - Each comes with its own commands and status messages
  - Storage: Xroot, CASTOR, dCache, DPM, STORM,+++
  - Operating Systems:
    - Windows, Linux (5 popular flavors), Solaris, MacOS,….
    - All come in several versions
  - Site managers
    - Highly experienced professionals
    - Physicists forced to do it
    - Summer students doing it for 3 months…….

What is Grid Middleware?

- For today:
- The software that allows to build a system that full fills the above requirements



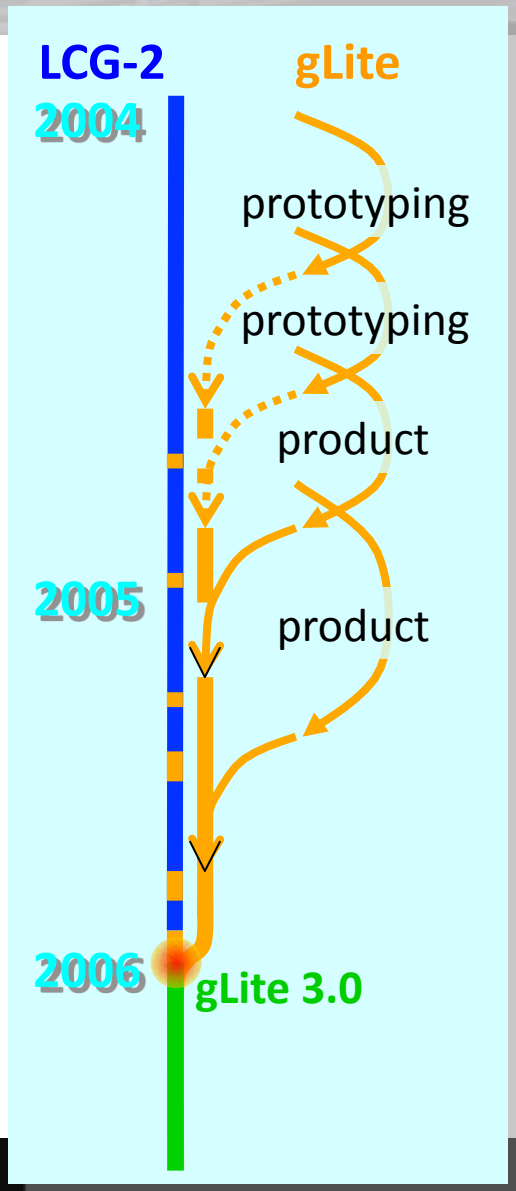
Software Approach

- Identify an AAA system that all can agree on
  - Authentication, Authorization, Auditing
  - That doesn't require local user registration
  - That delegates "details" to the users (Virtual Organizations)
- Define and implement abstraction layers for resources
  - Computing, Storage, etc.
- Define and implement a way to announce your resources
- Build high level services to optimize the usage
- Interface your applications to the system

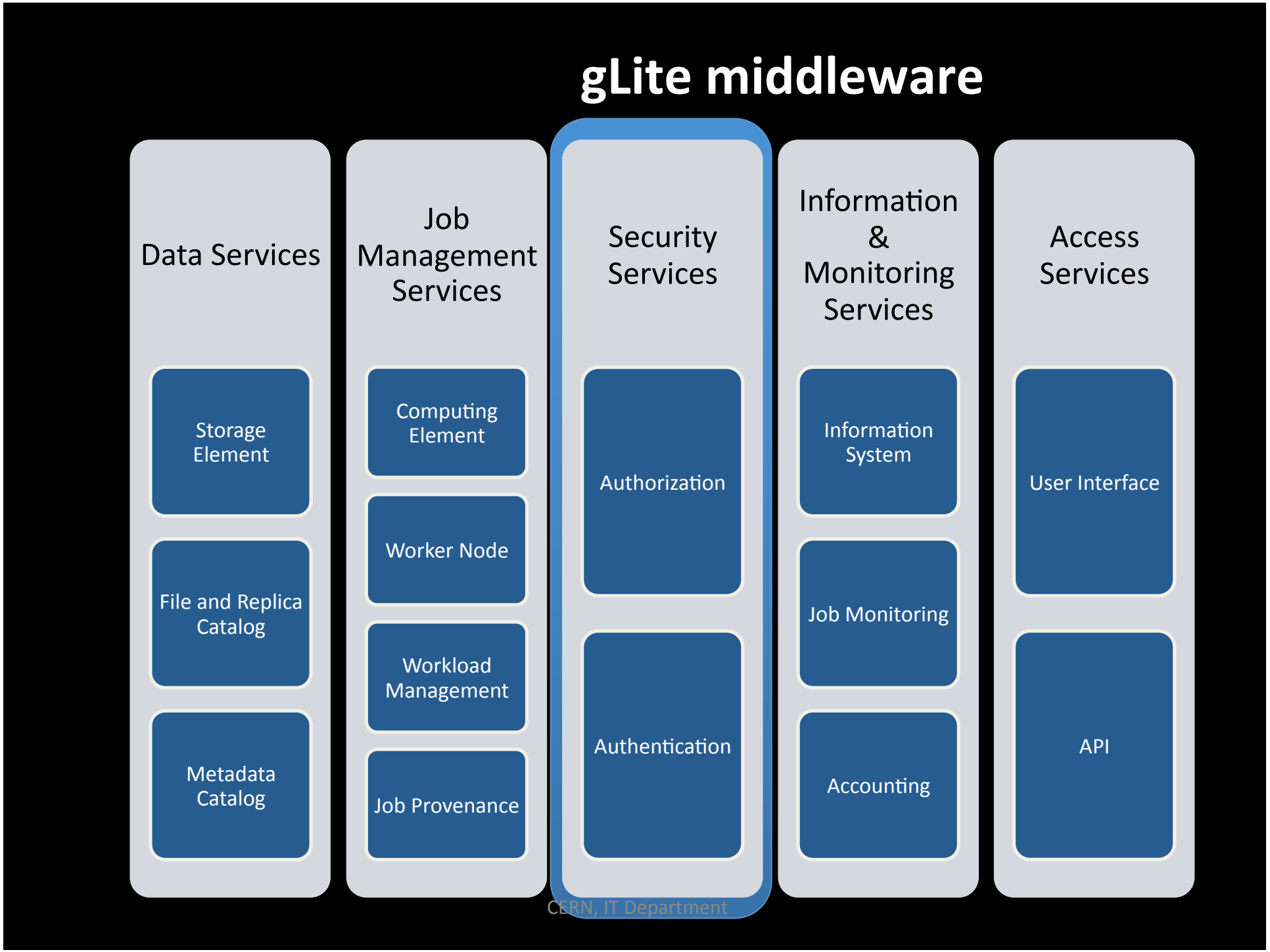
gLite as an example

gLite Middleware Distribution

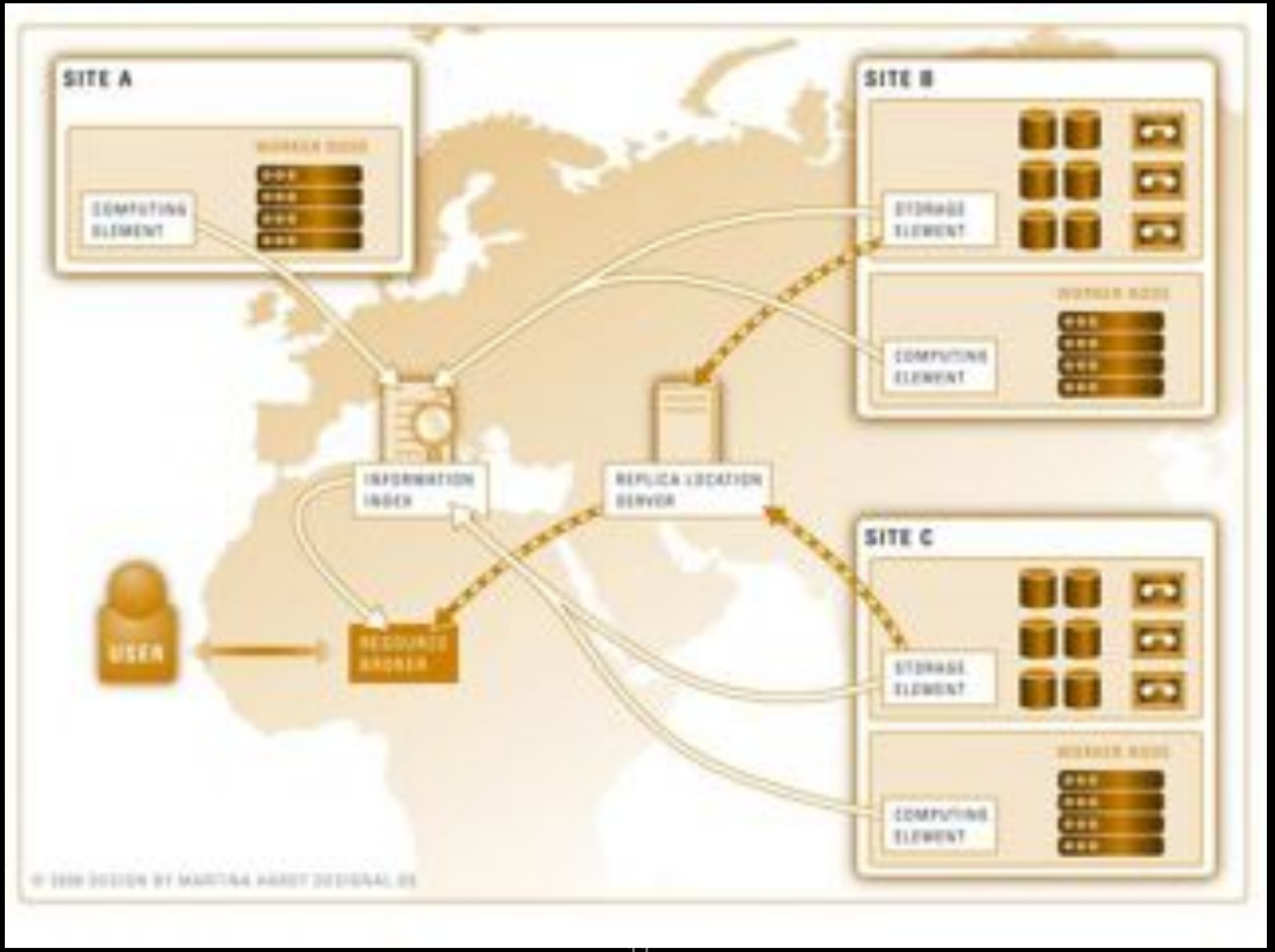
- Combines components from different providers
  - Condor and Globus 2 (via VDT)
  - LCG
  - EDG/EGEE
  - Others
- After prototyping phases in 2004 and 2005 convergence with LCG-2 distribution reached in May 2006
  - gLite 3.0
- Focus on providing a deployable MW distribution for EGEE production service



gLite
Lightweight Middleware for Grid Computing

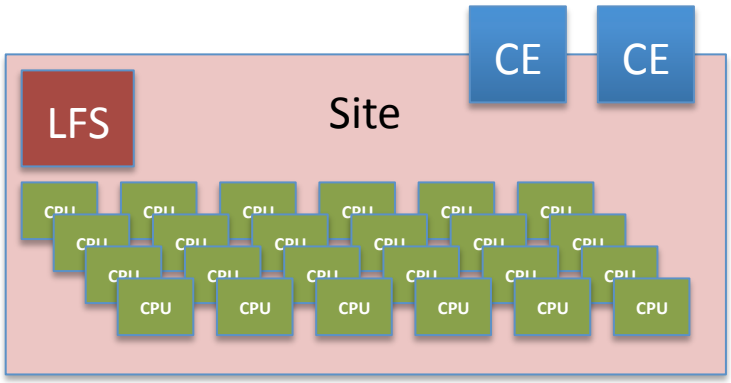

gLite middleware
Data Services
Storage Element
File and Replica Catalog
Metadata Catalog
Job Management Services
Computing Element
Worker Node
Workload Management
Job Provenance
Security Services
Authorization
Authentication
Information & Monitoring Services
Information System
Job Monitoring
Accounting
Access Services
User Interface
API
CERN, IT Department

The Big Picture



Computing Access

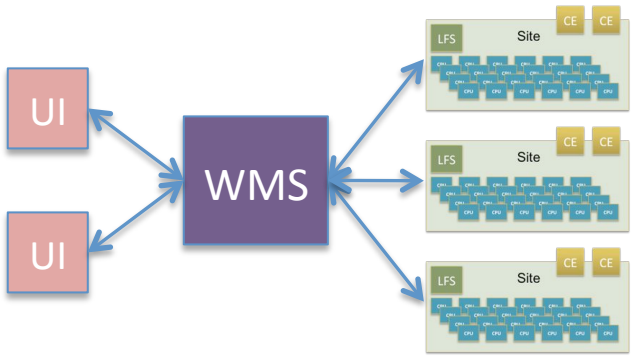
- Computing Elements (CE)
  - gateways to farms



- EGEE:
  - LCG-CE (450 instances)
    - Minor work on stabilization/scalability (50u/4KJ) , bug fixes
    - LEGACY
  - CREAM-CE (120 instances (up from 26))
    - Significant investment on production readiness and scalability
    - Handles direct submission (pilot job friendly)
    - SL4/SL5
    - BES standard compliant, parameter passing from grid <-> batch

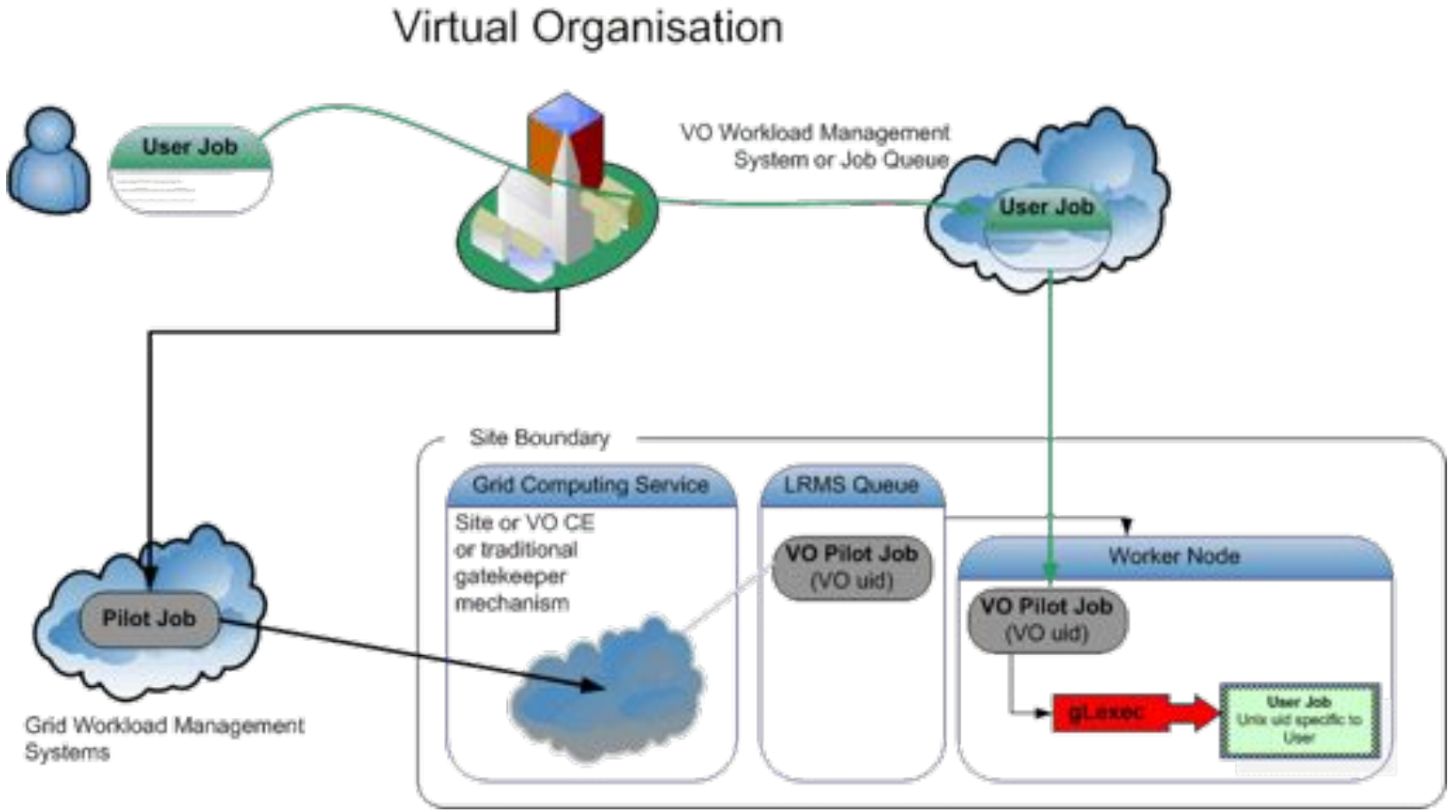
Workload Management

- EGEE WMS/LB
  - Matches resources and requests
    - Including data location
  - Handles failures (resubmission)
  - Manages complex workflows
  - Tracks job status
- EGEE WMS/LB (124 Instances)
  - Fully supports LCG-CE and CREAM-CE
    - Early versions had some WMS<->CREAM incompatibilities



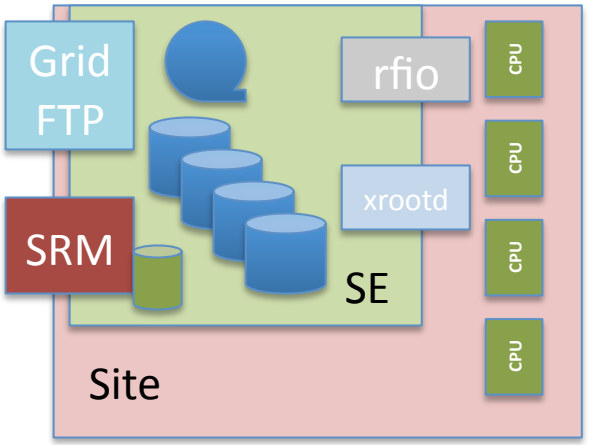
MultiUserPilotJobs

- Idea: Matching resources and jobs by Vos
- Pilot is a placeholder for the real job
- Identity is changed on demand on the WN



Data Management

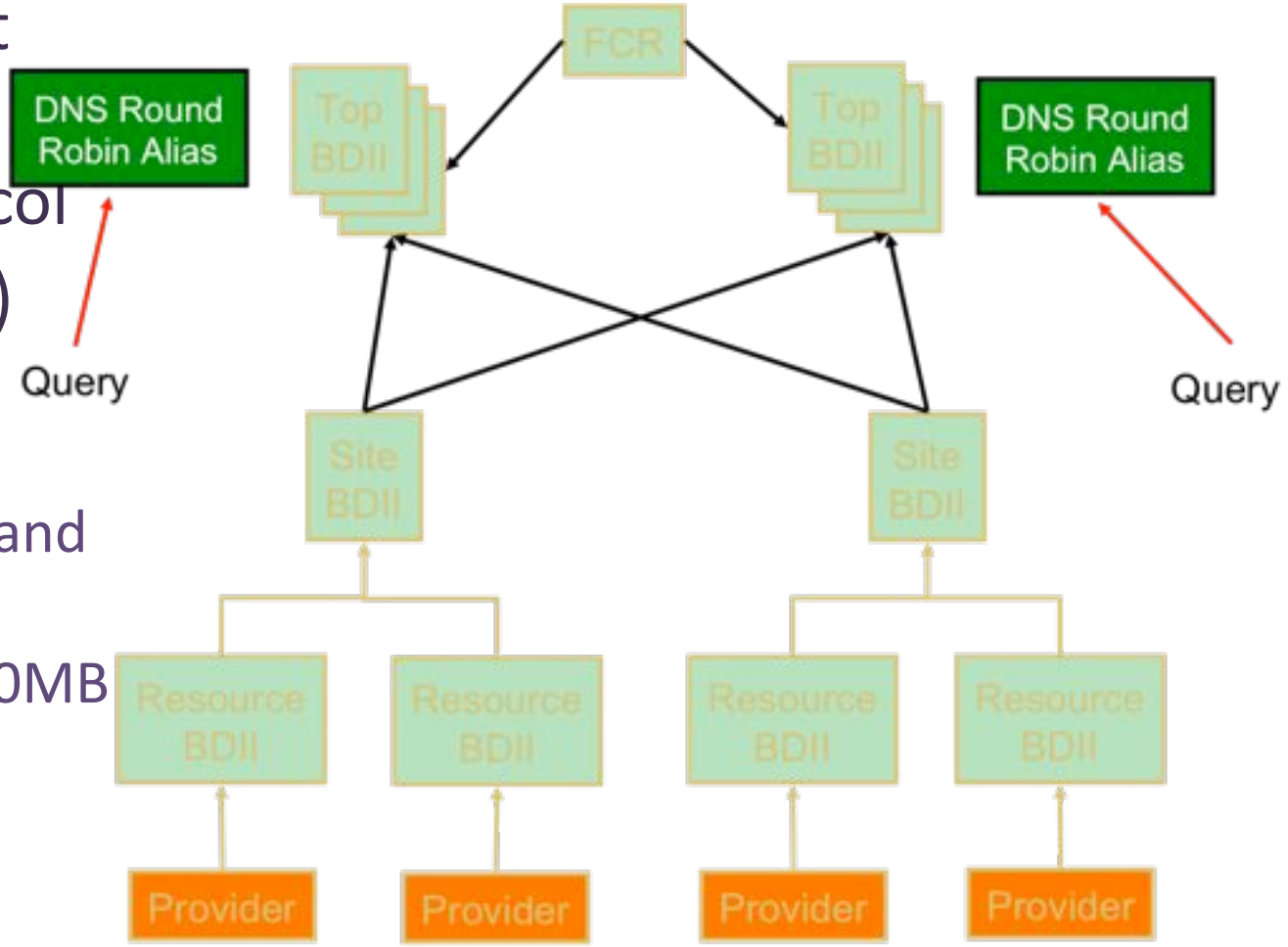
- Storage Elements (SEs)
  - External interfaces based on SRM 2.2 and gridFTP
  - Local interfaces: POSIX, dcap, secure rfio, rfio, xrootd
  - DPM (241)
  - dCache (82)
  - STORM (40)
  - BestMan (26)
  - CASTOR (19)
  - "ClassicSE" (27) → legacy since 2 years….



- Catalogue: LFC (local and global)
- File Transfer Service (FTS)
- Data management clients gfal/LCG-Utils

Information System

- BDII
- Light weight Database
- LDAP protocol
- GLUE 1.3 (2) Schema
  - Describes resources and their state
  - Approx 100MB
- Several hundred instances

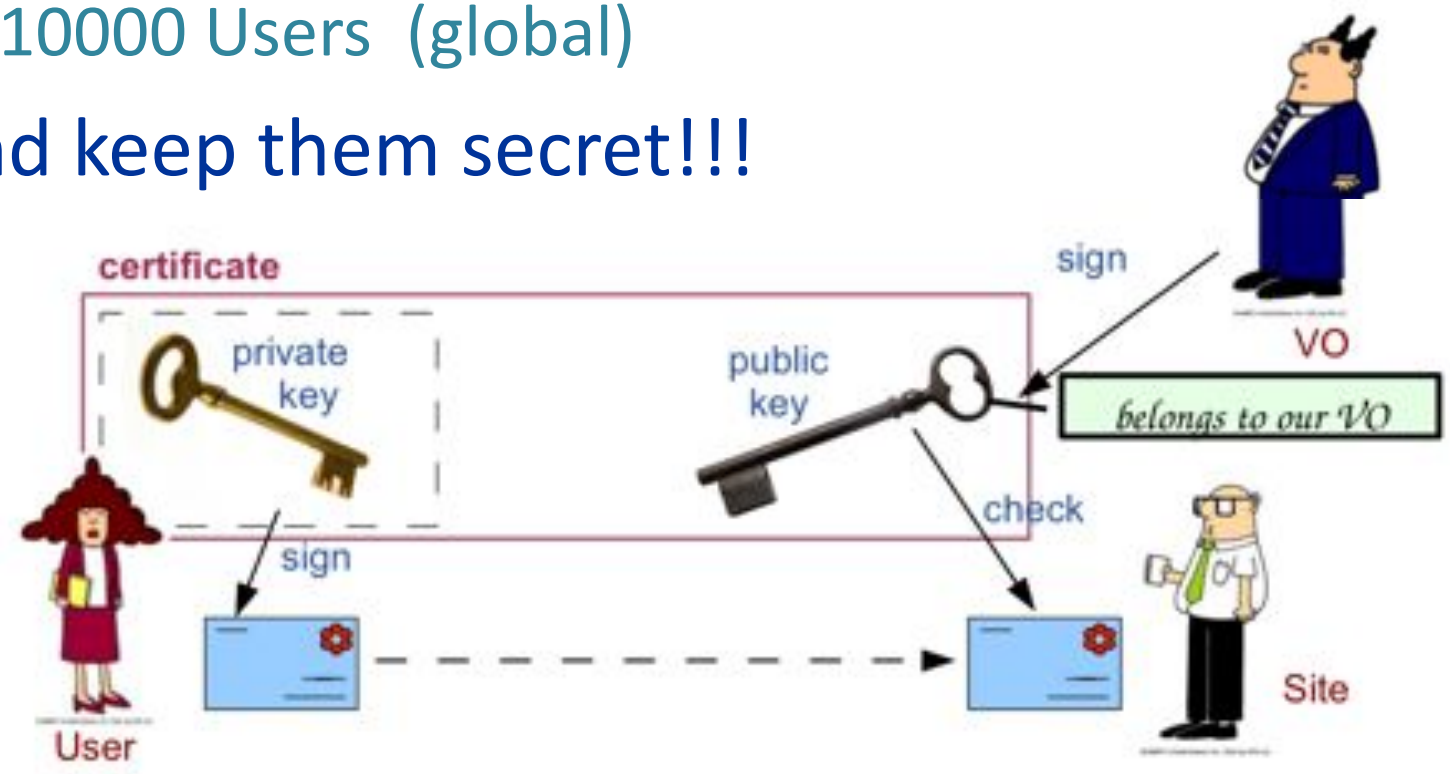


Authentication

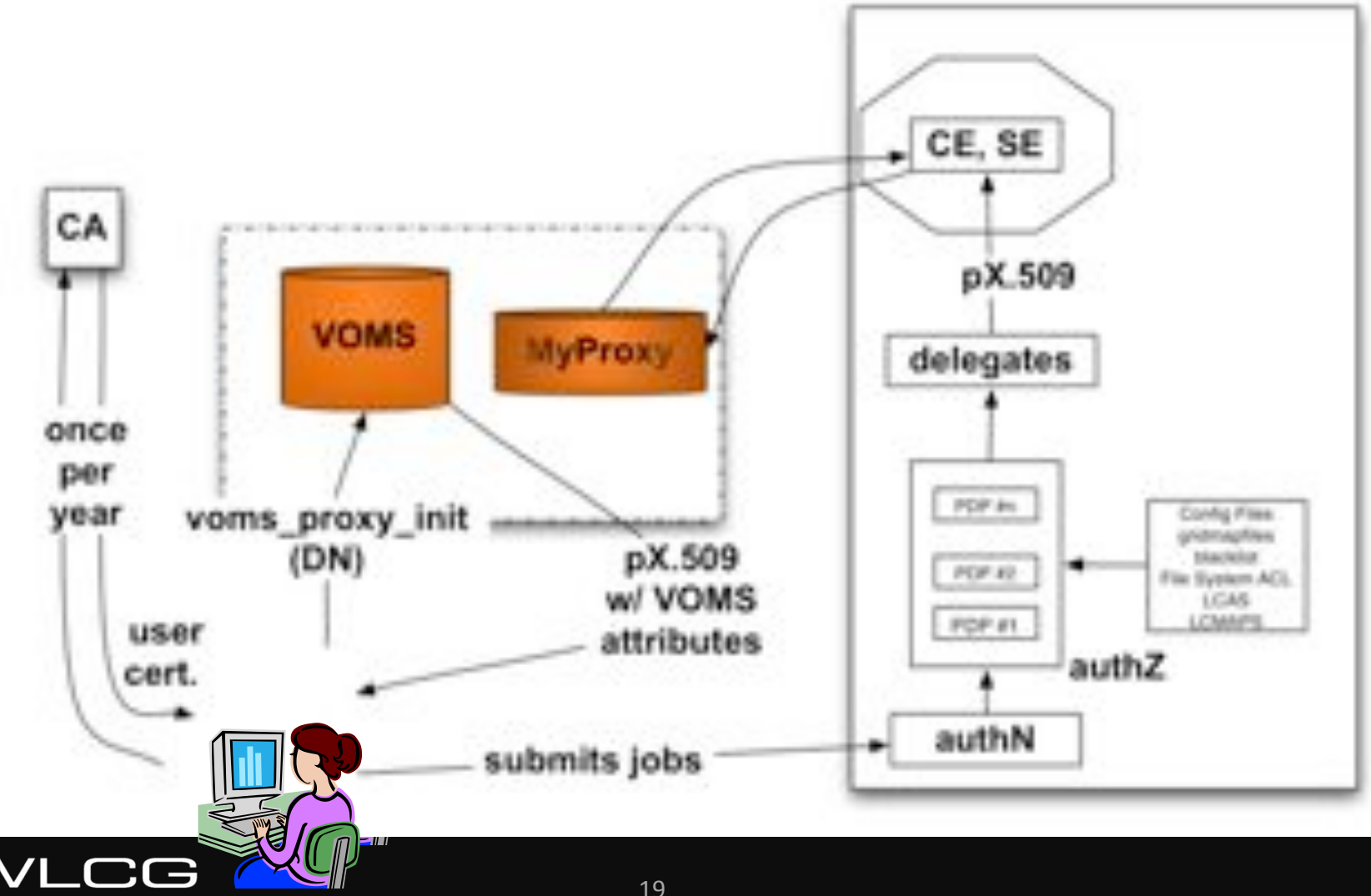
- Authentication is based on X.509 PKI infrastructure (Public Key)
  - Certificate Authorities (CA) issue (long lived) certificates identifying individuals (much like a passport)
    - Commonly used in web browsers to authenticate to sites
  - Trust between CAs and sites is established (offline)
  - In order to reduce vulnerability, on the Grid user identification is done by using (short lived) proxies of their certificates
- Short-Lived Credential Services (SLCS)
  - issue short lived certificates or proxies to its local users
    - e.g. from Kerberos or from Shibboleth credentials
- Proxies can
  - Be delegated to a service such that it can act on the user's behalf
  - Be stored in an external proxy store (MyProxy)
  - Be renewed (in case they are about to expire)
  - Include additional attributes -→ Authorization

Public Key Based Security

- How to exchange secret keys?
  - 340 Sites (global)
    - With hundreds of nodes each?
  - 200 User Communities (non local)
  - 10000 Users (global)
- And keep them secret!!!



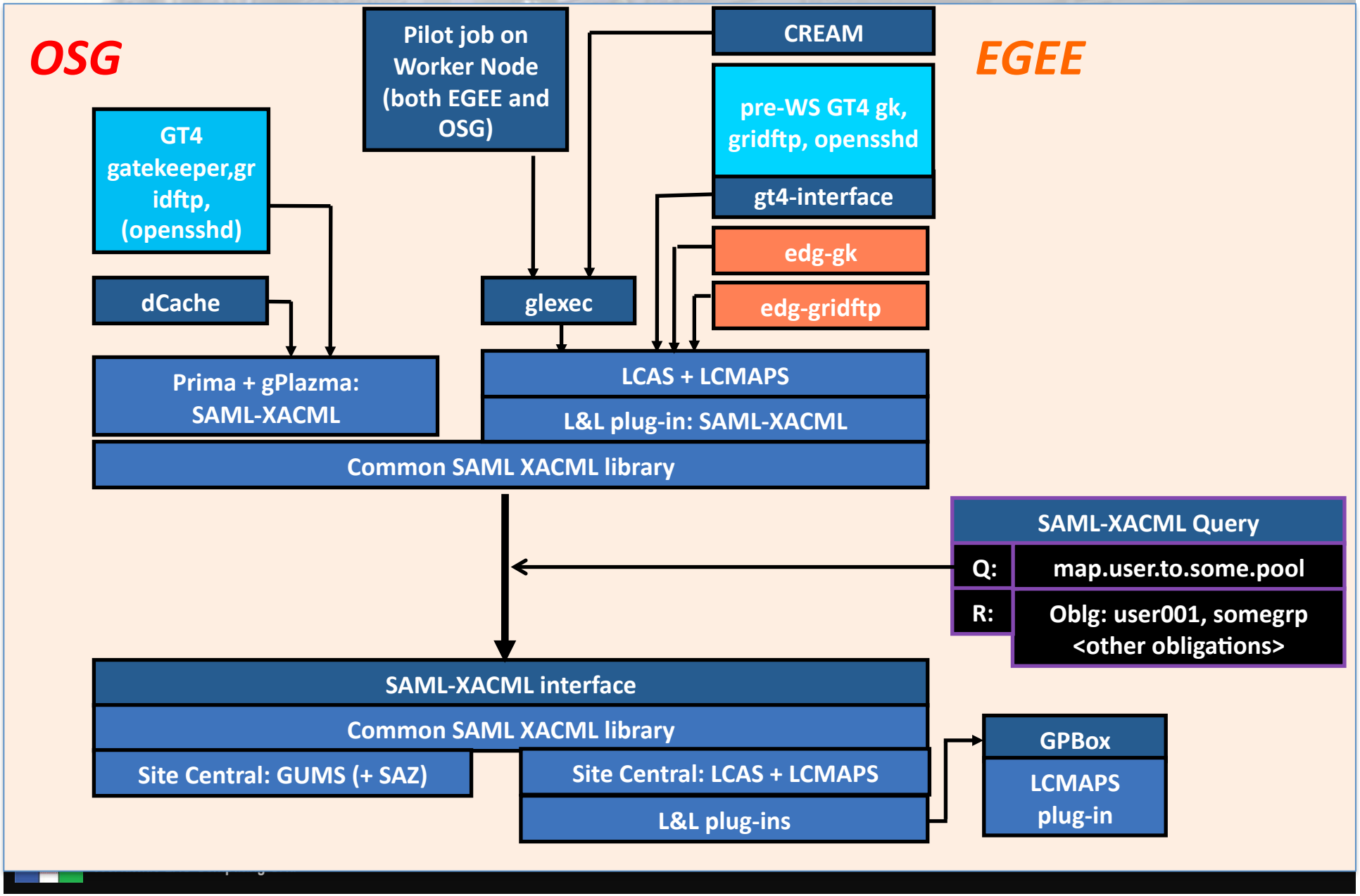
Security - overview



Authorization

- VOMS is now a de-facto standard
  - Attribute Certificates provide users with additional capabilities defined by the VO.
  - Basis for the authorization process
- Authorization: currently via mapping to a local user on the resource
  - glexec changes the local identity (based on suexec from Apache)
- Designing an authorization service with a common interface agreed with multiple partners
  - Uniform implementation of authorization in gLite services
  - Easier interoperability with other infrastructures
  - Prototype being prepared now

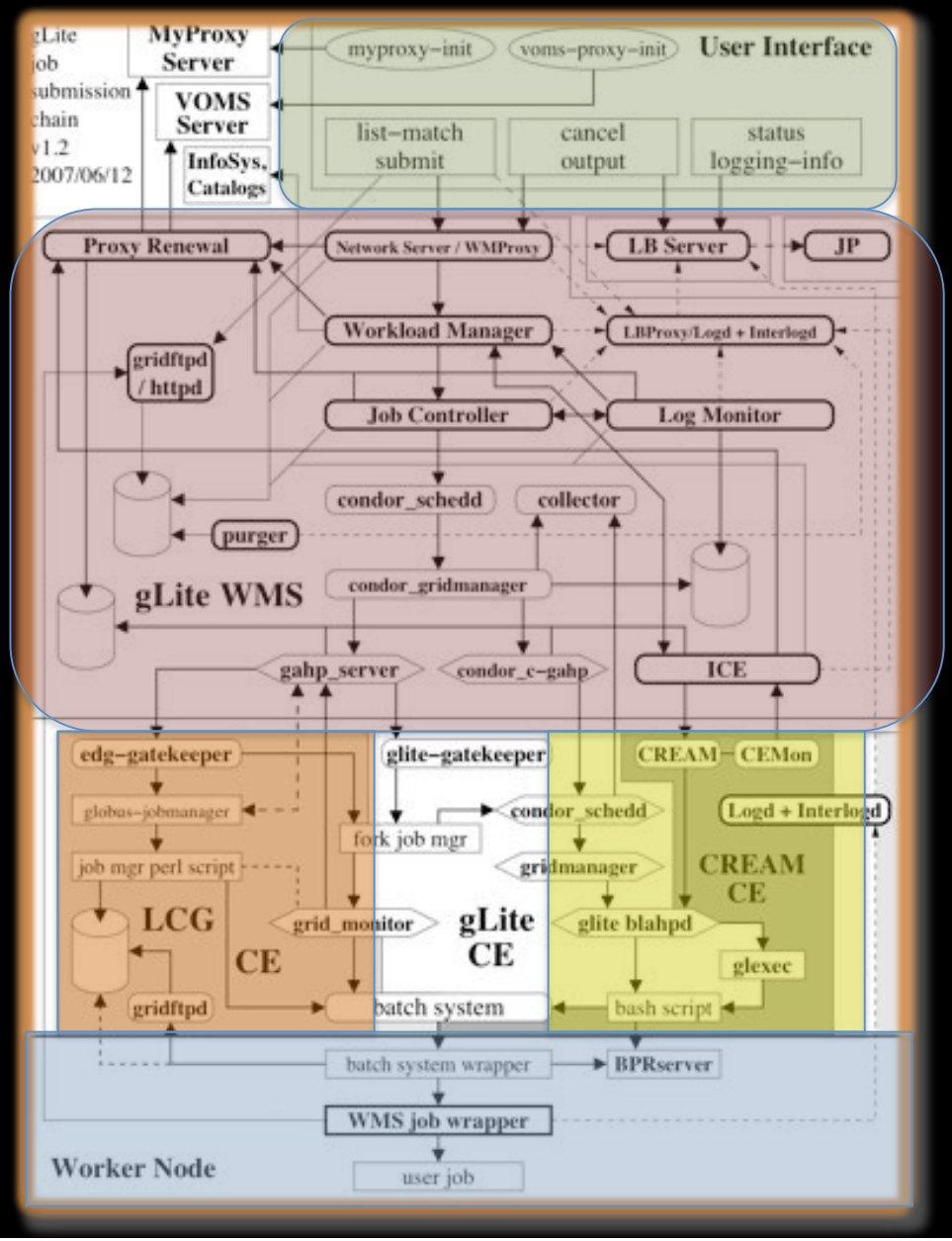
Common AuthZ interface



Impossible to discuss all components

- Illustrate complexity
- Some Data Management Details

Workload Management (compact)



Desktops

A few
~50 nodes

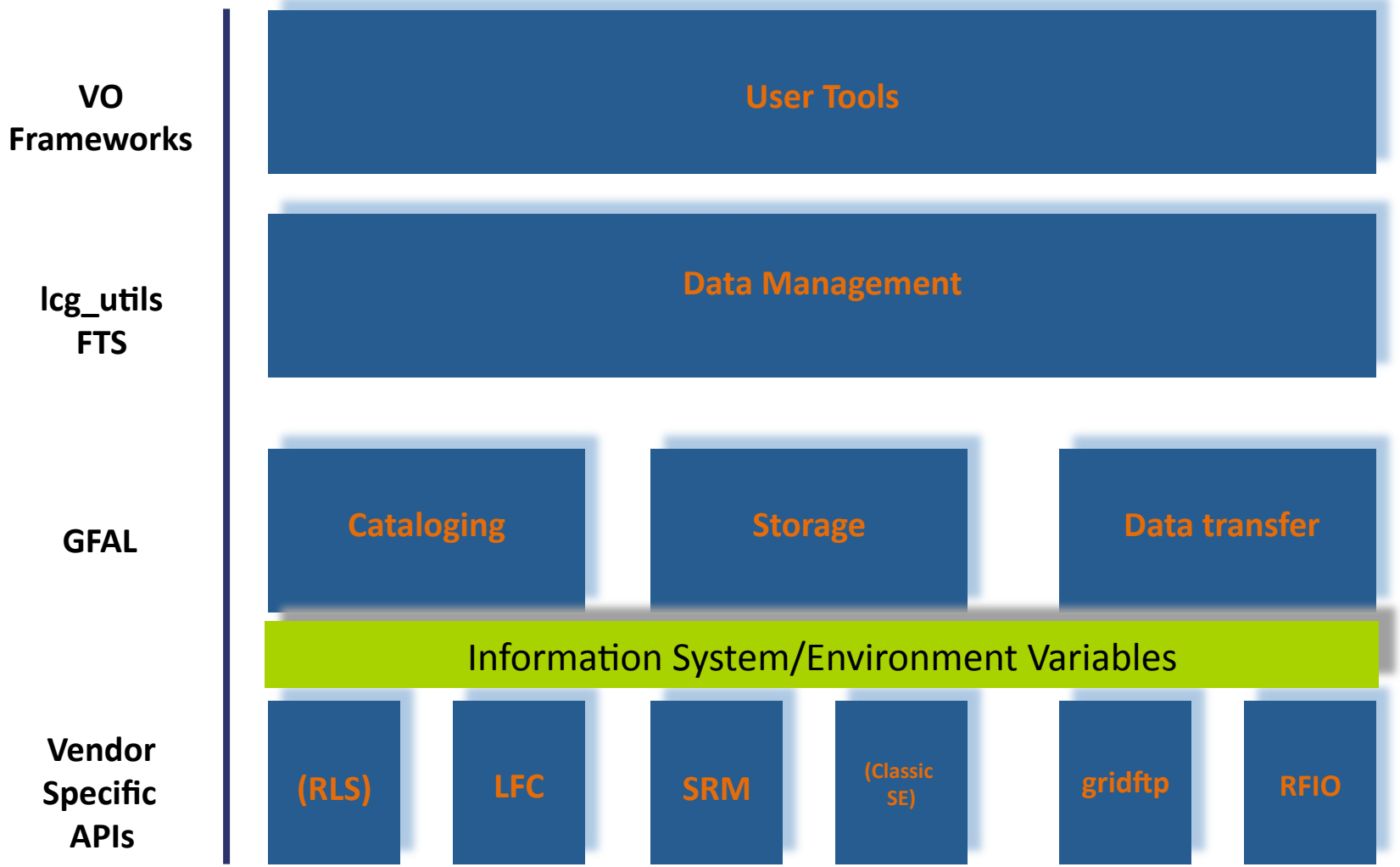
1-20 per site

1-24000
per site

Job Description Language

```
[
  Executable = “my_exe”;
  StdOutput = “out”;
  StdError = “err”;
  Arguments = “a b c”;
  InputSandbox = {“/home/giaco/my_exe”};
  OutputSandbox = {“out”, “err”};
  Requirements = Member(
    other.GlueHostApplicationSoftwareRunTimeEnvironment,
    "ALICE3.07.01“
  );
  Rank = -other.GlueCEStateEstimatedResponseTime;
  RetryCount = 3
]
```

Data Management



General Storage Element

- Storage Resource Manager (SRM)
  - hides the storage system implementation (disk or active tape)
  - handles authorization
  - translates SURLs (Storage URL) to TURLs (Transfer URLs)
  - disk-based: DPM, dCache,+; tape-based: Castor, dCache
  - Mostly asynchronous
- *File I/O*: *posix-like* access from local nodes or the grid

  ➔ GFAL (Grid File Access Layer)

Approach to SRM

- An abstraction layer for storage and data access is necessary
  - Guiding principle:
  - Non-interference with local policies
- Providing all necessary user functionality and control
  - Data Management
  - Data Access
  - Storage management
  - Control:
    - Pinning files
    - Retention Policy
    - Space management and reservation
  - Data Transfers
- Grid enabled and based on current technology
  - Interface technology (gSOAP)
  - Security Model (gsi security)
  - To integrate with the grid infrastructure

Motivation (for HEP)

- Distributed processing model
  - Data Management, Data Access, Storage Resource Management
  - User community is experiment centric
    - No longer institute centric
    - Requires radical change in Authentication/Authorisation technology
- But:
- Many existing and heavily used heterogeneous (local) storage systems
  - Different models and implementations for
    - local storage hierarchy
      - transparent/explicit
    - Synchronous/Asynchronous operations
    - Cluster file system based / disk server based
    - Plethora of Data Access Clients
    - Authorization and authentication
      - Often local, mostly UID/GID or AFS like ACLs, Kerberos, +++
    - Wide area transfers
      - FTP doors, proprietary
    - ……
  - Deeply integrated with local computing fabrics
  - Representing decade long, massive investment
  - Have to respect local policies and resource allocations

Methods

- Standard Document ~100 pages
  - Relative short (see NFS-4.1 ~260 pages)
- Space Management (11)
- Permission (3)
- Directory (6)
- Data Transfer (17)
- Discovery (2)
- Too many?
  - Not as many as a naïve count indicates
  - Most methods are asynchronous -> divide by two
    - Srm<method> and srmStatusOf<method>Request
- Very flexible behavior → complex verification/clients
- WLCG Addendum tried to simplify both dimensions

Implementation(s)

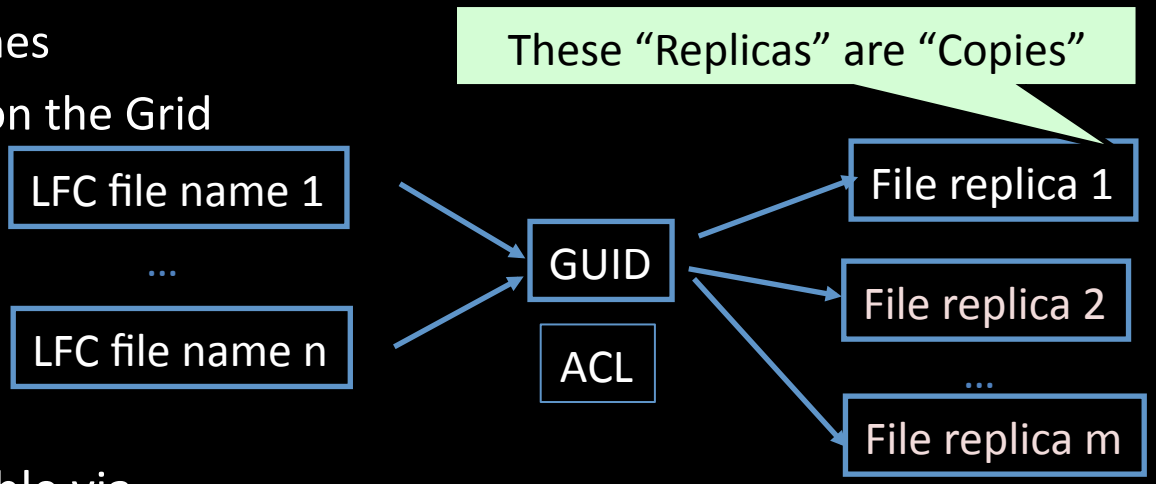
- WSDL to generate client stubs
- gSOAP
- GSI for A(A)

- External interface is the “easy” part
- Integration with the existing storage system is the challenge
- Different approaches concerning the accepted level of entanglement between SRM and storage
  - New storage systems have less problems
    - Designed for SRM

SRM basic and use cases tests

LCG "File" Catalog

- The LFC stores mappings between
  - Users' file names
  - File locations on the Grid



- The LFC is accessible via
  - CLI, C API, Python interface, Perl interface
    - Supports sessions and bulk operations
  - Data Location Interface (DLI)
    - Web Service used for match making:
      - given a GUID, returns physical file location
- ORACLE backend for high performance applications
  - Read-only replication support

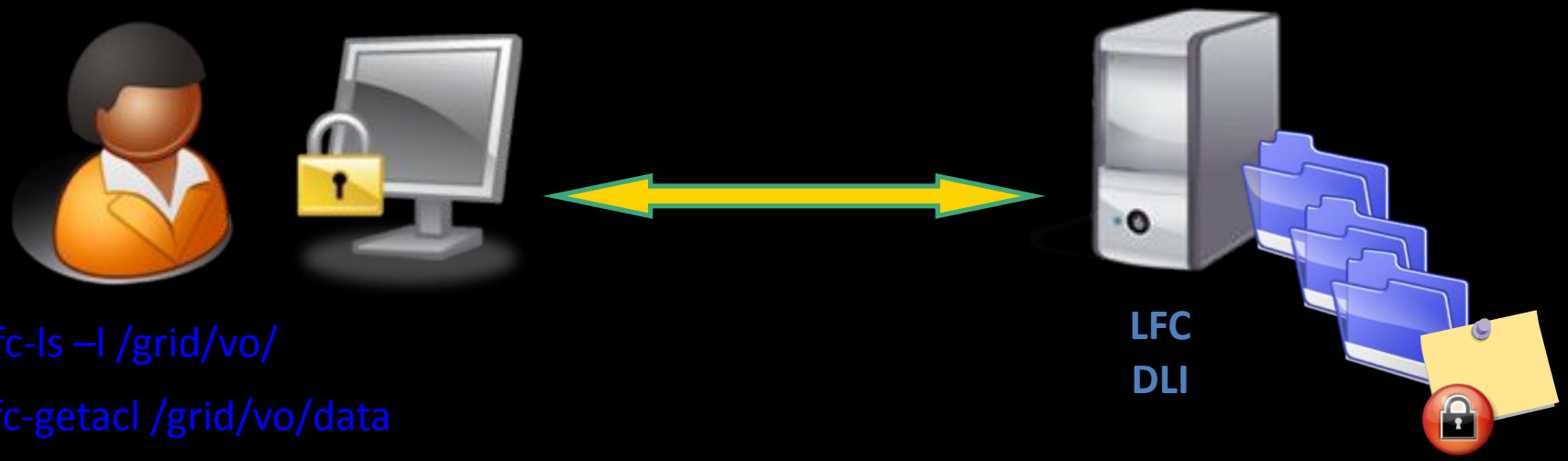
All files are "Write Once Read Many"

LFC features

Hierarchical Namespace
GSI security
Permissions and ownership
ACLs (based on VOMS)
Virtual ids
- Each user is mapped to (uid, gid)

VOMS support
- To each VOMS group/role corresponds a virtual gid

Bulk operations



lfc-ls –l /grid/vo/

lfc-getacl /grid/vo/data

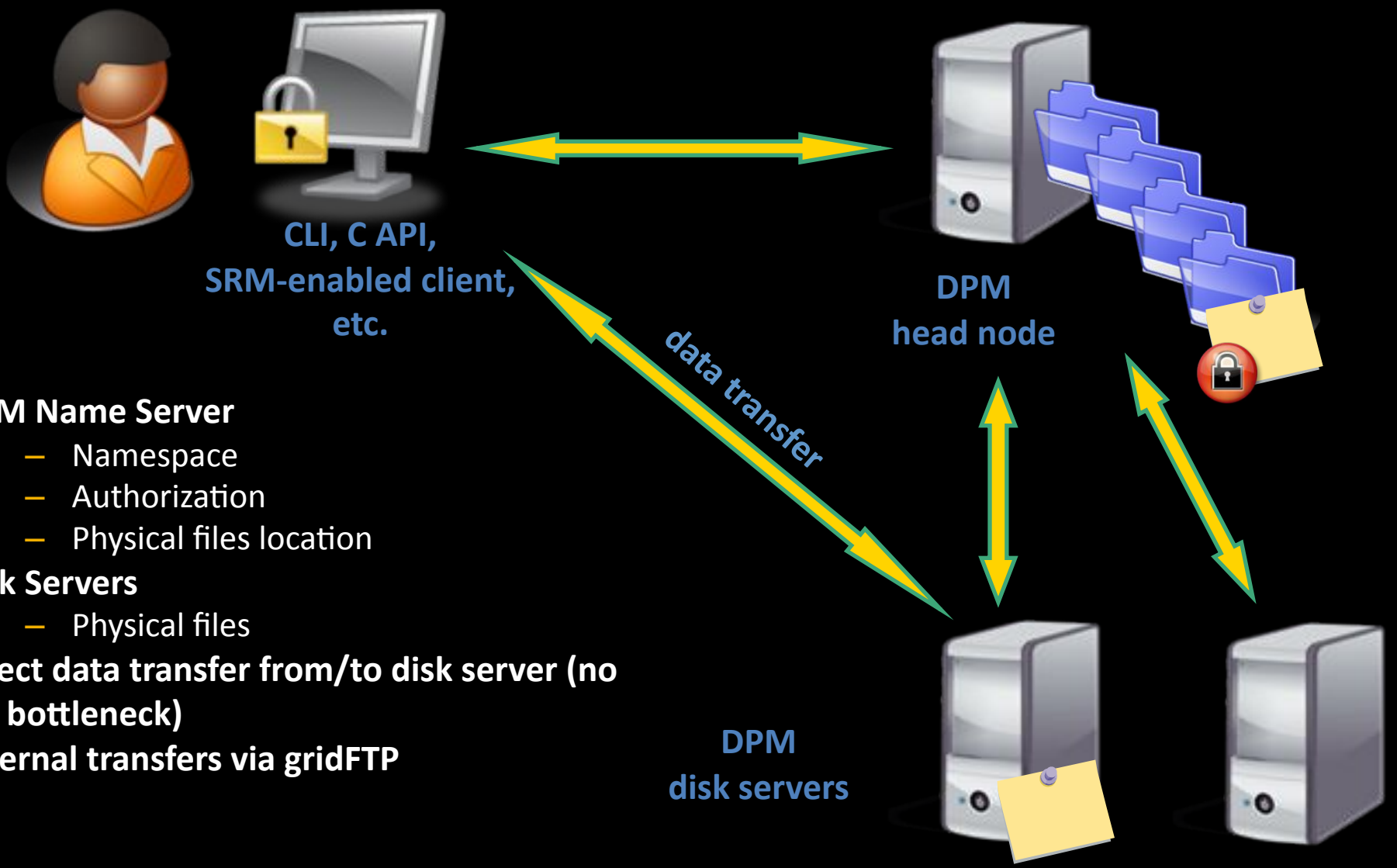
DPM

- Disk Pool Manager
  - Manages storage on disk servers
  - SRM support
    - 1.1
    - 2.1 (for backward compatibility)
    - 2.2 (released in DPM version 1.6.3)
  - GSI security
  - ACLs
  - VOMS support
  - Secondary groups support (see LFC)

DPM strengths

- Easy to use
  - Hierarchical namespace
    - $ dpns-ls /dpm/cern.ch/home/vo/data
  - Many protocols supported (including HTTPS)
- Easy to administrate
  - Easy to install and configure
  - Low maintenance effort
  - Easy to add/drain/remove disk servers
- Target: small to “medium” sites (1.6 PB)
  - Single disks --> several disk servers

DPM: user's point of view



DPM Name Server
- Namespace
- Authorization
- Physical files location

Disk Servers
- Physical files

Direct data transfer from/to disk server (no bottleneck)

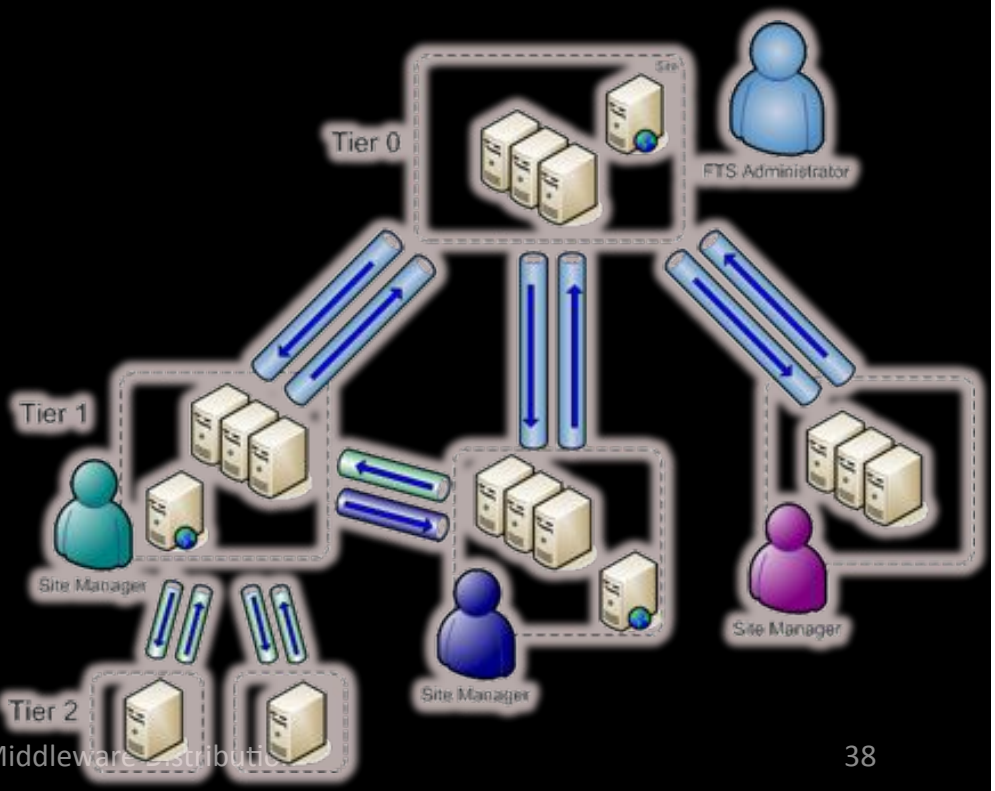
External transfers via gridFTP

GFAL & lcg_util

- Data management access libs.
  - Shield users from complexity
  - Interacts with information system, catalogue and SRM-SEs
- GFAL
  - Posix like C API for file access
  - SRMv2.2 support
  - User space tokens correspond to
    - A certain retention policy (custodial/replica)
    - A certain access latency (online/nearline)
- lcg_util (command line + C API)
  - Replication, catalogue interaction etc.

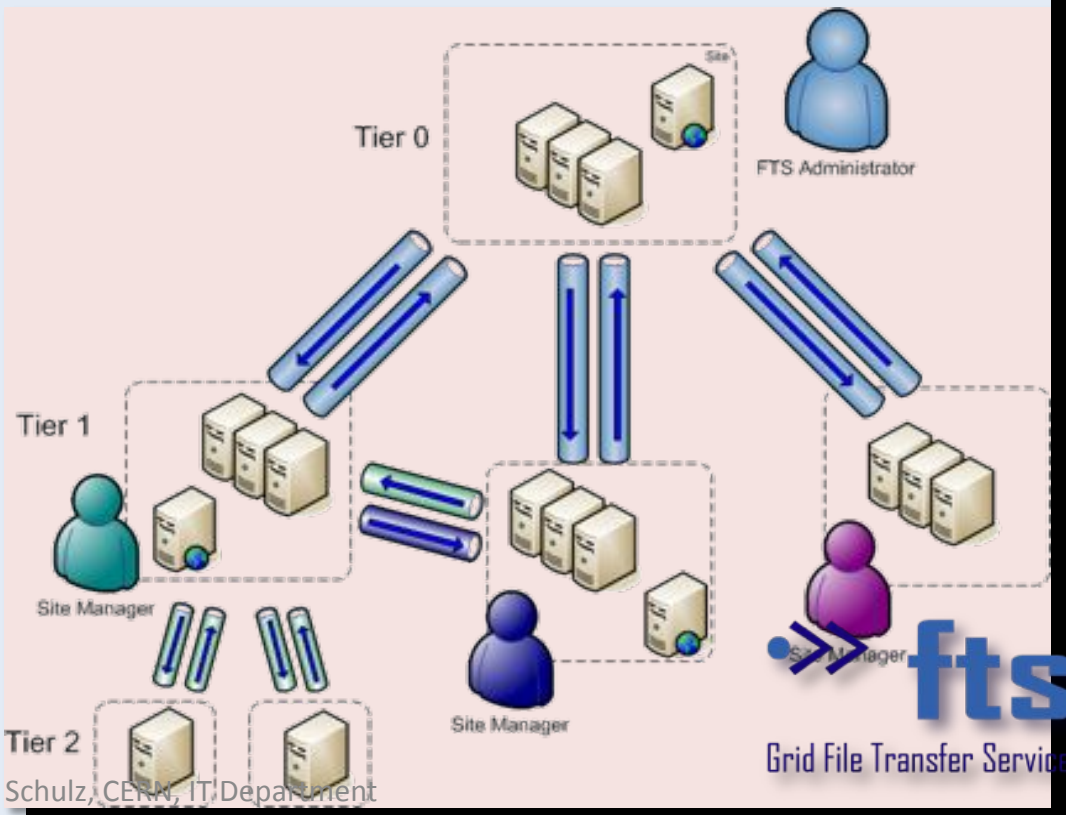
FTS overview

- gLite File Transfer Service is a reliable data movement fabric service (batch for file transfers)
  - FTS performs bulk file transfers between sites
  - Transfers are made between any SRM-compliant storage elements (both SRM 1.1 and 2.2 supported)
- It is a **multi-VO** service, used to balance usage of site resources according to the SLAs agreed between a site and the VOs it supports
- VOMS aware



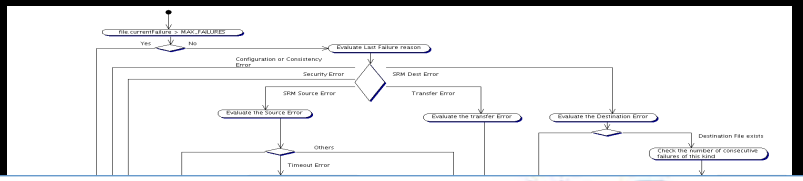
File Transfer Service

- FTS: Reliable, scalable and customizable file transfer
  - Multi-VO service, used to balance usage of site resources according to the SLAs agreed between a site and the VOs it supports
  - WS interface, support for different user and administrative roles (VOMS)
  - Manages transfers through channels
    - mono-directional network pipes between two sites
  - File transfers handled as jobs
    - Prioritization
    - Retries in case of failures
  - Automatic discovery of services
- Designed to scale up to the transfer needs of very data intensive applications
  - Demonstrated about 1 GB/s sustained
  - Over 9 petabytes transferred in the last 6 months (> 10 million files)



FTS: key points

- Reliability
  - It handles the retries in c
    storage / network failure
    - VO customizable retry lo
  - Service designed for
    high-availability deployn
- Security
  - All data is transferred se
    SRM / gridFTP
  - Service audits all user / a
- Service and performance
  - Service stability: it is des
    storage and network res
  - Service recovery: integra
    level degradation



FTS Report

fts

Disclaimer
This page contains a report generated from information stored in the FTS Database and is intended for reporting purposes only. Since the format will probably change in the future, it's therefore recommended not to use parsing robots on it.

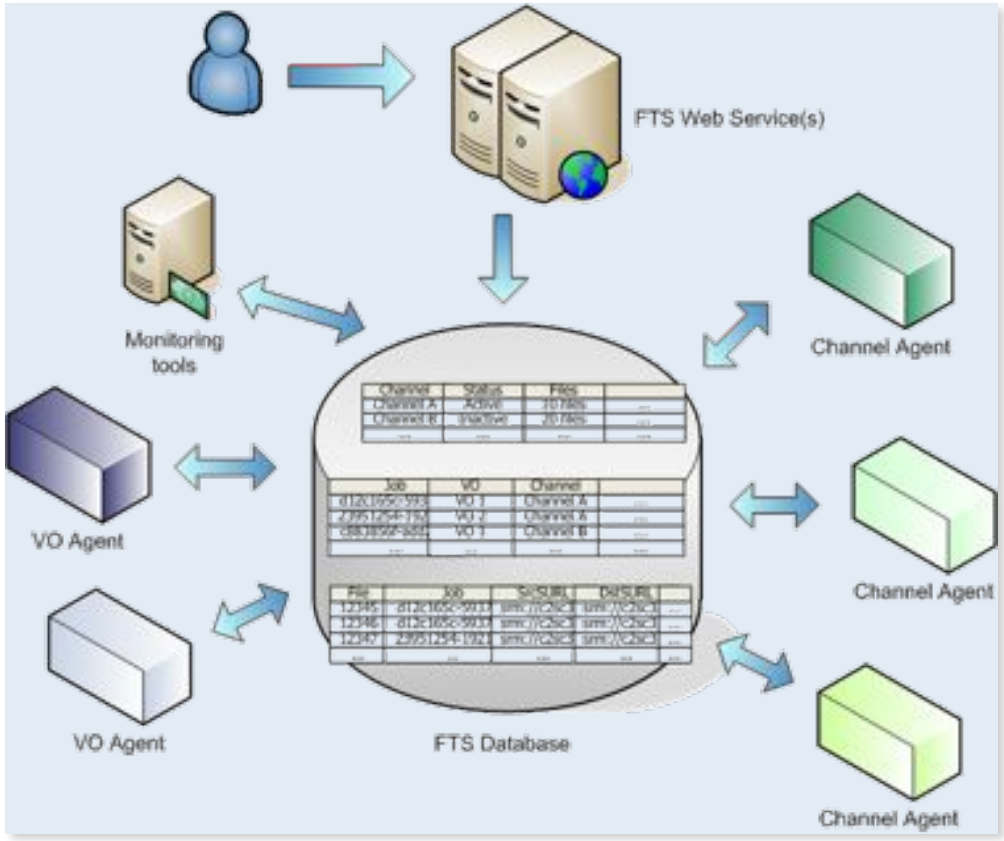
Statistics concerning all the transfers performed yesterday
Between 2006-10-12 08:00:00 +02:00 and 2006-10-13 08:00:00 +02:00

CERN-* Filter Show VO details

Channel Name	VO Name	Total	% Failures	# Succ.	# Fail.	1st Failure Reason	% 1st Failure Reason	2nd Failure Reason	% 2nd Failure Reason	Avg. Size (GB)	Avg. Duration (sec)	Avg. Tx Rate (MB/sec)	Eff. Tx Bytes (GB)	Tx Bytes (GB)
CERN-PIC	[All]	12262	73.97	3192	9070	Dest SRM	56.22	Other	37.53	0.53	263.83	1.62	1700.41	1700.41
	atlas	8937	99.02	7	8925	Dest SRM	57.13	Other	38.89	0	220	0	0	0
	cms	200	0	200	0					2.7	767.55	3.64	561.36	561.36
	dteam	974	0.51	969	5	Other	80	Source SRM	20	0.95	356.31	2.86	923.83	923.83
	lhcb	2145	6.53	2005	140	Source SRM	99.29	Other	0.71	0.11	365.85	0.81	215.32	215.32
	ops	3	0	3	0					0	202.67	0	0	0
CERN-FNAL	[All]	9699	59.26	3544	5155	Other	84.91	Source SRM	14.89	0.85	478.22	2.59	3028.91	3027.57
	alice	1155	82.6	201	954	Other	99.58	Dest SRM	8.31	1.86	1865.85	1.31	372.95	372.95
	atlas	4552	88.52	519	3994	Other	84.85	Source SRM	15.95	1.79	1428.84	1.49	926.26	926.57
	cms	227	3.08	220	7	Dest SRM	85.71	Source SRM	14.29	2.53	348.65	10.86	555.61	555.61
	dteam	1877	3.99	1834	43	Other	86.85	Source SRM	9.3	0.95	276.64	4.01	980.67	980.91
	lhcb	1725	9.1	1568	157	Source SRM	99.36	Other	0.64	0.12	346.83	1.96	191.52	191.52
	ops	3	0	3	0					0	27	0.81	0	0
CERN-GARA	[All]	6792	42.55	5051	3741	Dest SRM	83.77	Source SRM	12.22	1.34	108.82	15.4	6777.95	6784.92
	alice	3134	15.12	2660	474	Source SRM	57.17	Dest SRM	41.14	1.66	909.53	18.43	4426.44	4430.29
	atlas	2018	53.33	942	1076	Dest SRM	72.6	Source SRM	16.54	1.15	344.44	9.62	1085.07	1087.6
	dteam	1488	69.32	1349	2139	Dest SRM	98.74	Other	0.99	0.93	91.89	14.66	1260.74	1261.32
	lhcb	148	35.14	96	52	Dest SRM	92.31	Other	3.85	0.06	76.1	0.93	5.7	5.7
	ops	4	0	4	0					0	97.25	0.02	0	0
CERN-INFN	[All]	11492	42.31	6630	4862	Dest SRM	43.85	Other	37.7	1.13	396.77	3.21	7514.29	7514.84
CERN-CERN	[All]	1536	39.71	926	610	Source SRM	58.36	Dest SRM	15.9	0.37	287.71	0.39	67.89	69.08
CERN-ASCC	[All]	6851	23.54	5236	1613	Source SRM	50.84	Other	28.89	1.14	1098.8	1.88	5955.91	6080.58
CERN-GRIDKA	[All]	12755	21.36	10028	2727	Source SRM	64.36	Other	32.53	0.87	371.87	3.19	8762.02	8767.63
CERN-TRIUMF	[All]	2264	28.63	1781	483	Other	61.77	Source SRM	31.1	1.84	395.15	3.62	1847.25	1817.13
CERN-BNL	[All]	13875	19.42	11281	2714	Source SRM	69.97	Other	24.24	0.44	190.39	3.41	4951.59	4980.34
CERN-IN2P3	[All]	11697	13.76	10087	1610	Source SRM	46.57	Other	47.45	1.22	296.21	6.33	12329.63	12329.63
CERN-FNAL	[All]	917	4.58	875	42	Transfer	97.62	Other	2.38	0	379.88	0	0	0

Click on the Channel Name to show the VO details

FTS server architecture



Experiments interact via a web-service

VO agents do VO-specific operations (1 per VO)

Channel agents do channel specific operation (e.g. the transfers)

Monitoring and statistics can be collected via the DB

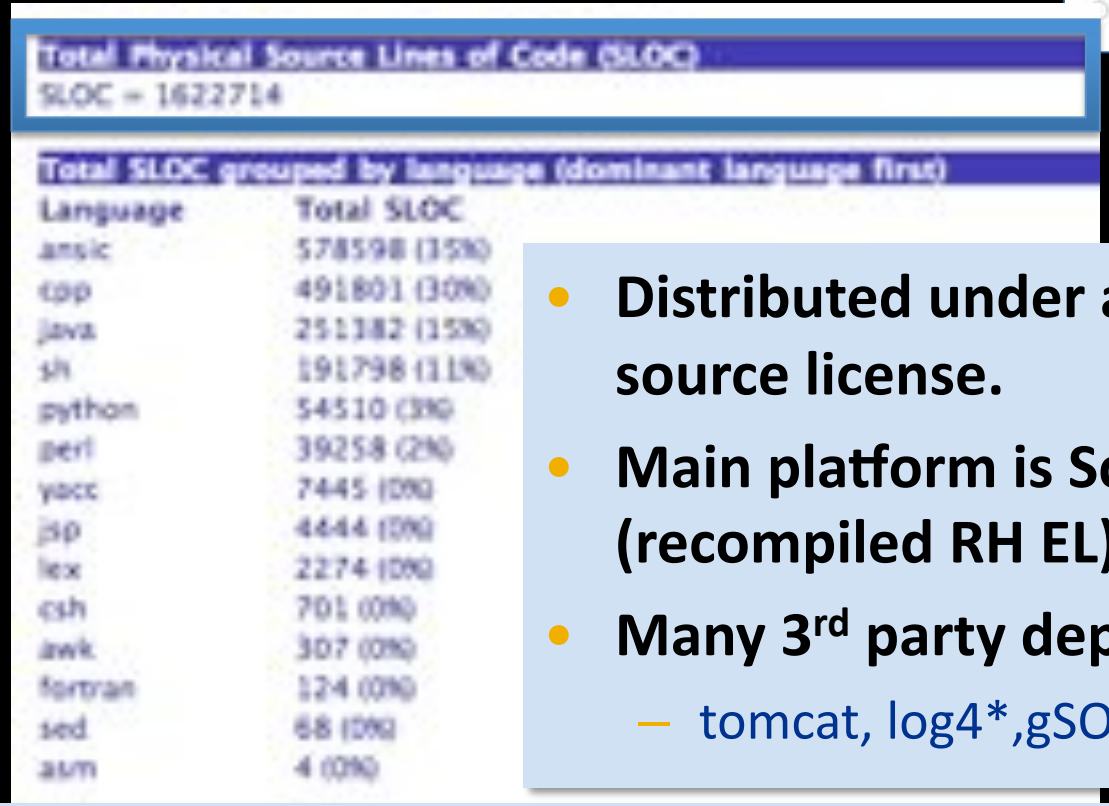
- All components are decoupled from each other
  - Each interacts only with the (Oracle) database

What Other Software is in gLite

- Encrypted Data Storage
  - DICOM SE, HYDRA (distributed key store)
- Several grid enabled storage systems
- Meta Data Catalogues
  - AMGA
- Logging and Bookkeeping
  - Doing exactly this
- Accounting
  - APEL, DGAS
- ARGUS
  - Global/local authorization and policy system

How does the code look like?

gLite code base



Total Physical Source Lines of Code (SLOC)
SLOC = 1622714

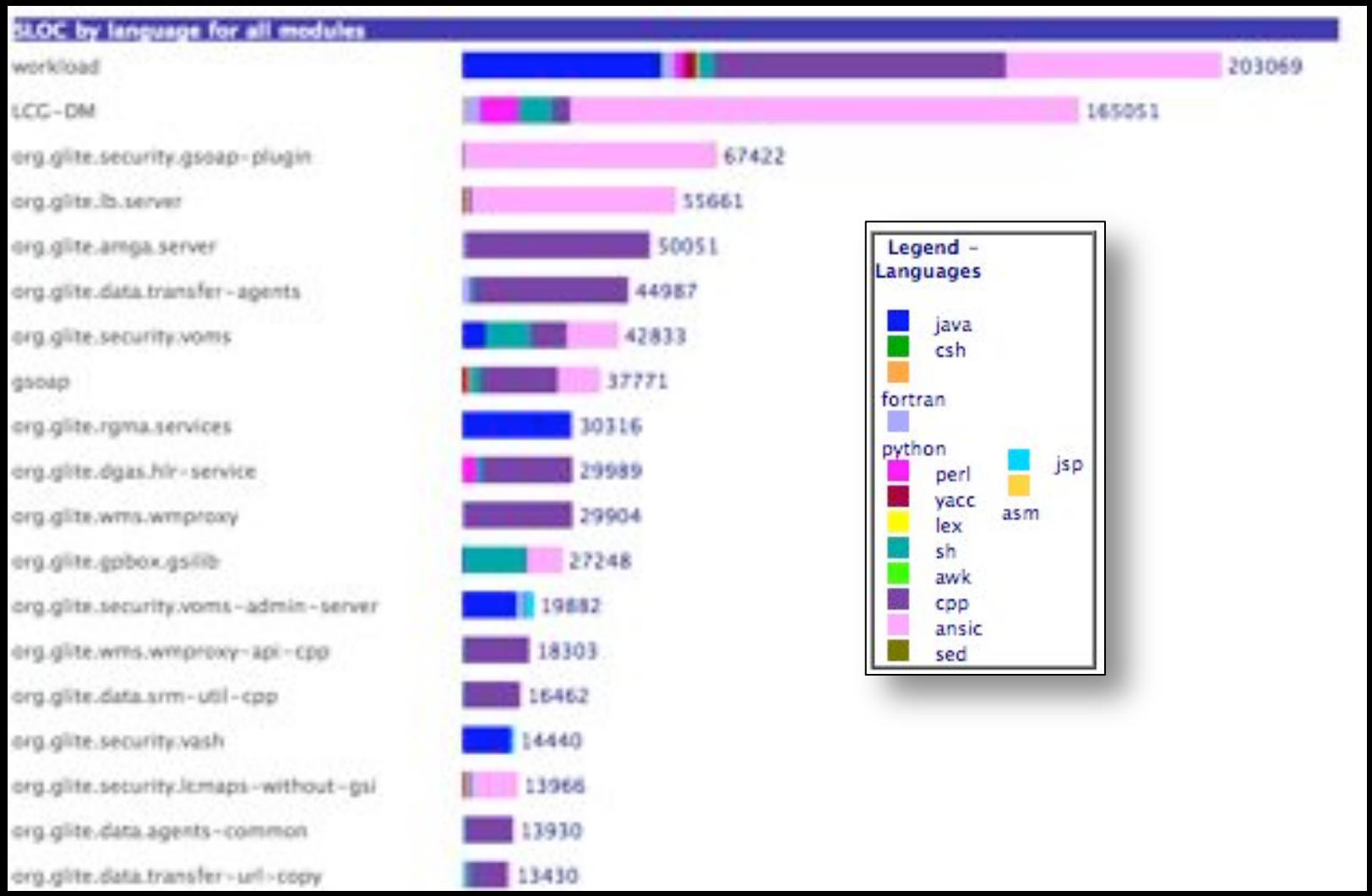
Total SLOC grouped by language (dominant language first)

Language	Total SLOC
ansic	578598 (35%)
cpp	491801 (30%)
java	251382 (15%)
sh	191798 (11%)
python	54510 (3%)
perl	39258 (2%)
yacc	7445 (0%)
jsp	4444 (0%)
lex	2274 (0%)
csh	701 (0%)
awk	307 (0%)
fortran	124 (0%)
sed	68 (0%)
asm	4 (0%)

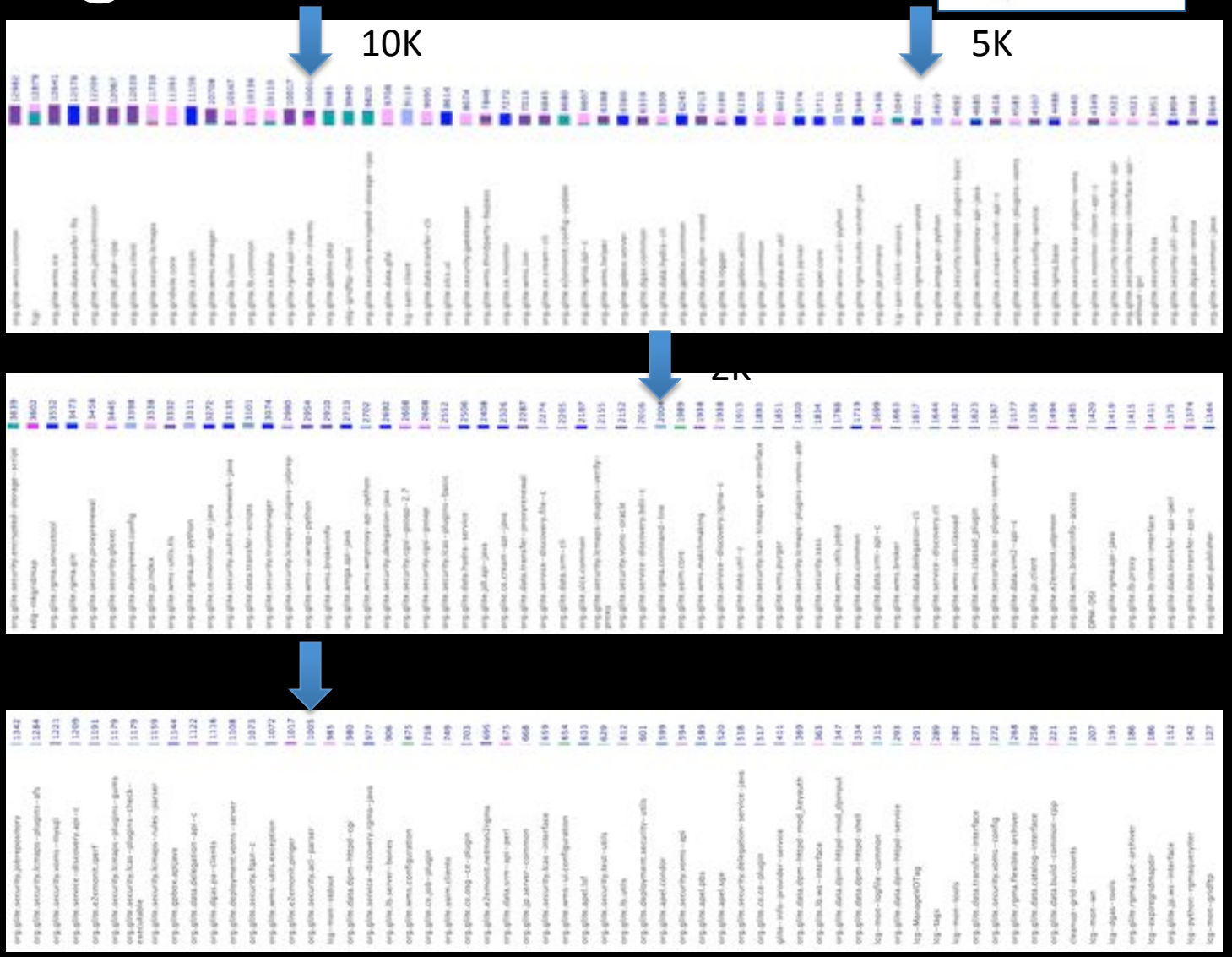
- **Distributed under an open source license.**
- **Main platform is Scientifc Linux (recompiled RH EL).**
- **Many 3rd party dependencies**
  - tomcat, log4*,gSOAP , ldap etc.

- **~ 20 FTEs, 80 people, 12 institutes (mostly academic)**
- **Geographically distributed, independent**
  - Coding conventions, Documentation, Naming Conventions
  - Testing and quality, dependency management

gLite code details



gLite code details

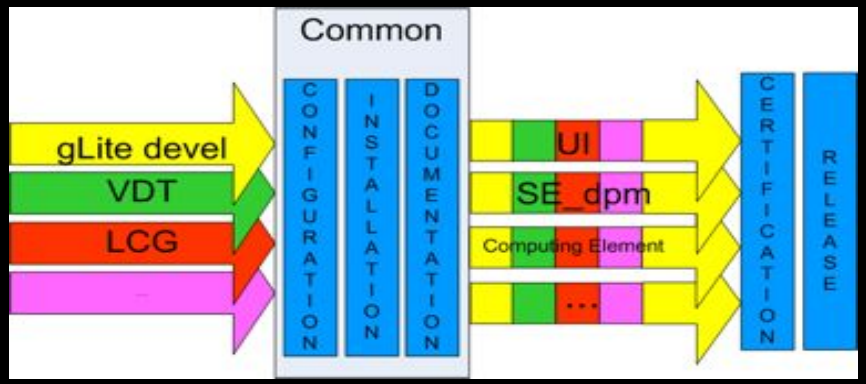


gLite code details

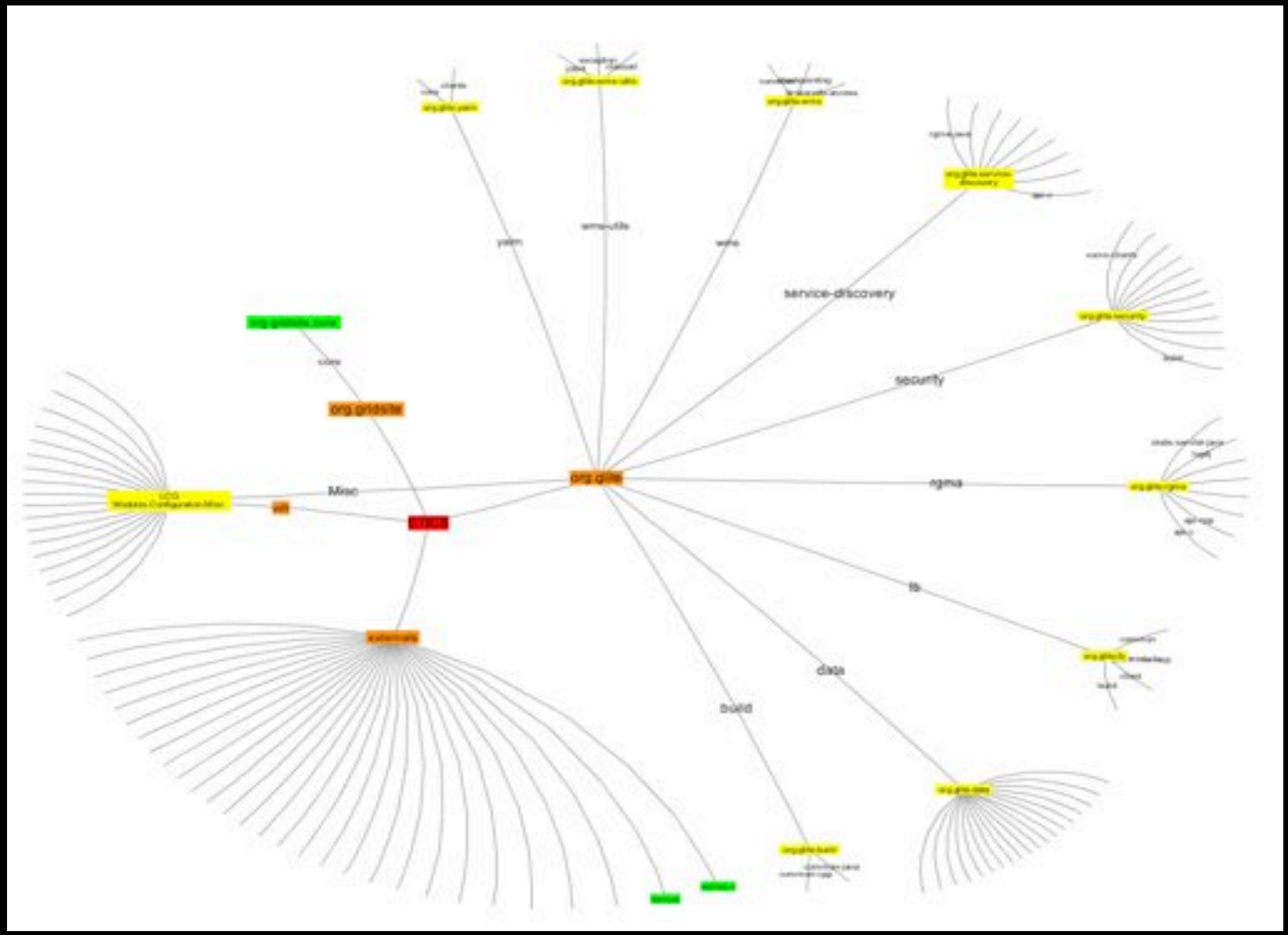
Complex external and internal cross dependencies

→Integration, configuration management was always a challenge

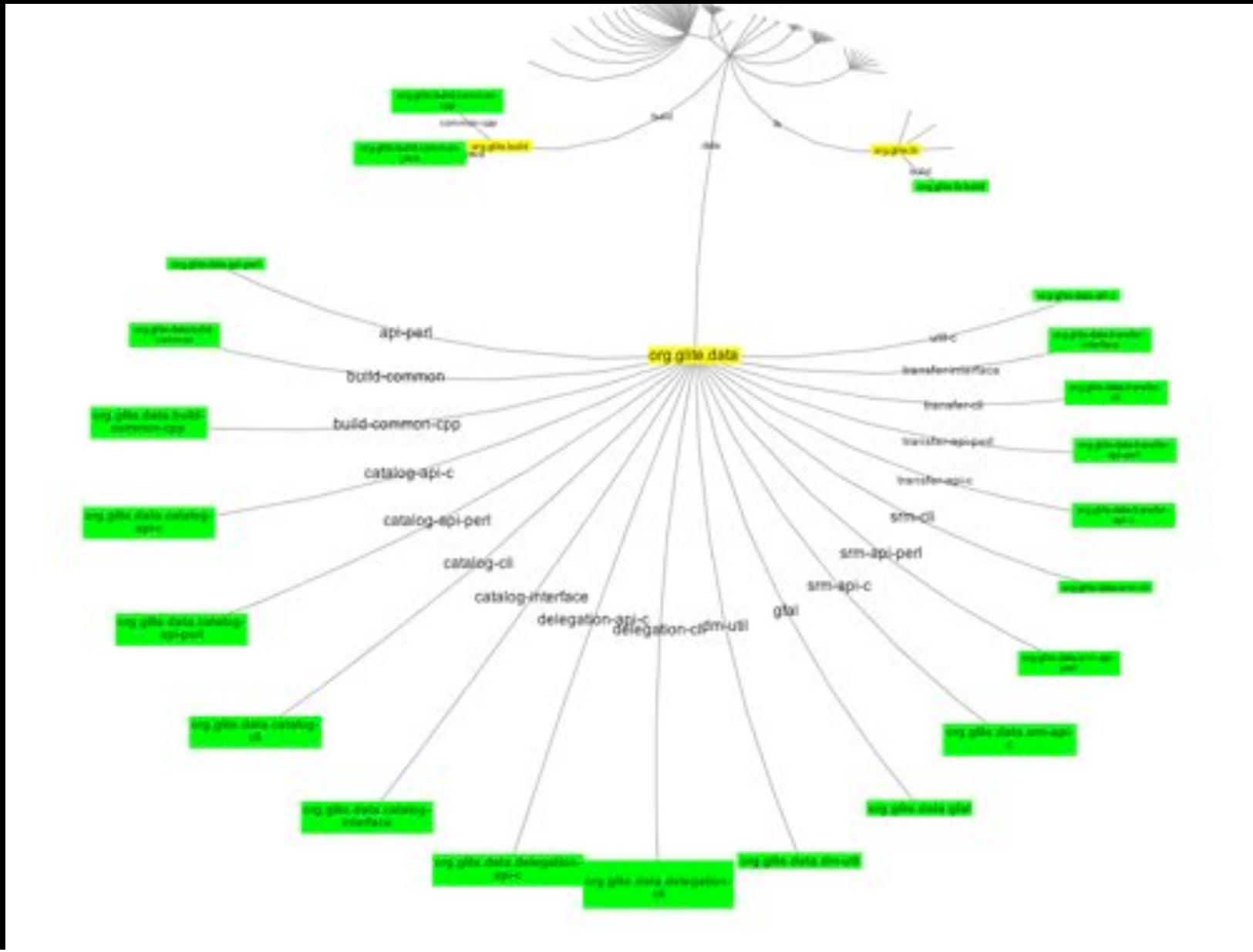
→The components are grouped together to ~30 services



Complex Dependencies

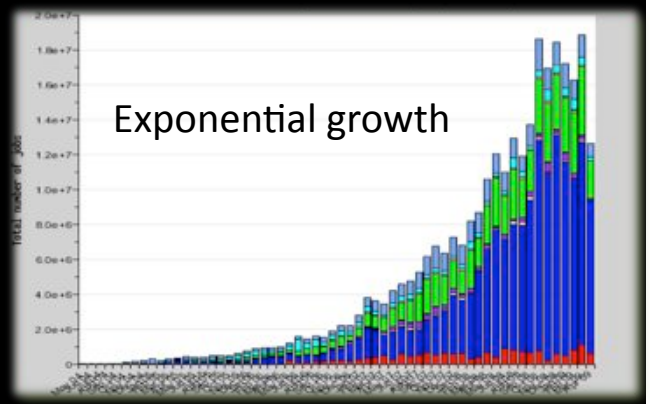


Example: Data Management



Stability of the software

- All components still see frequent changes
- Many developments started 2002
  - Why do we still need changes?
    - Scale of the system increased rapidly
    - Number of user and use cases increased
      - Deeper code coverage
      - New functional requirements
    - Less tolerance to failures
      - Implementation of fail over
    - Emerging standards
      - Project started when no standards where available
      - Incremental introduction



How do we manage the code?

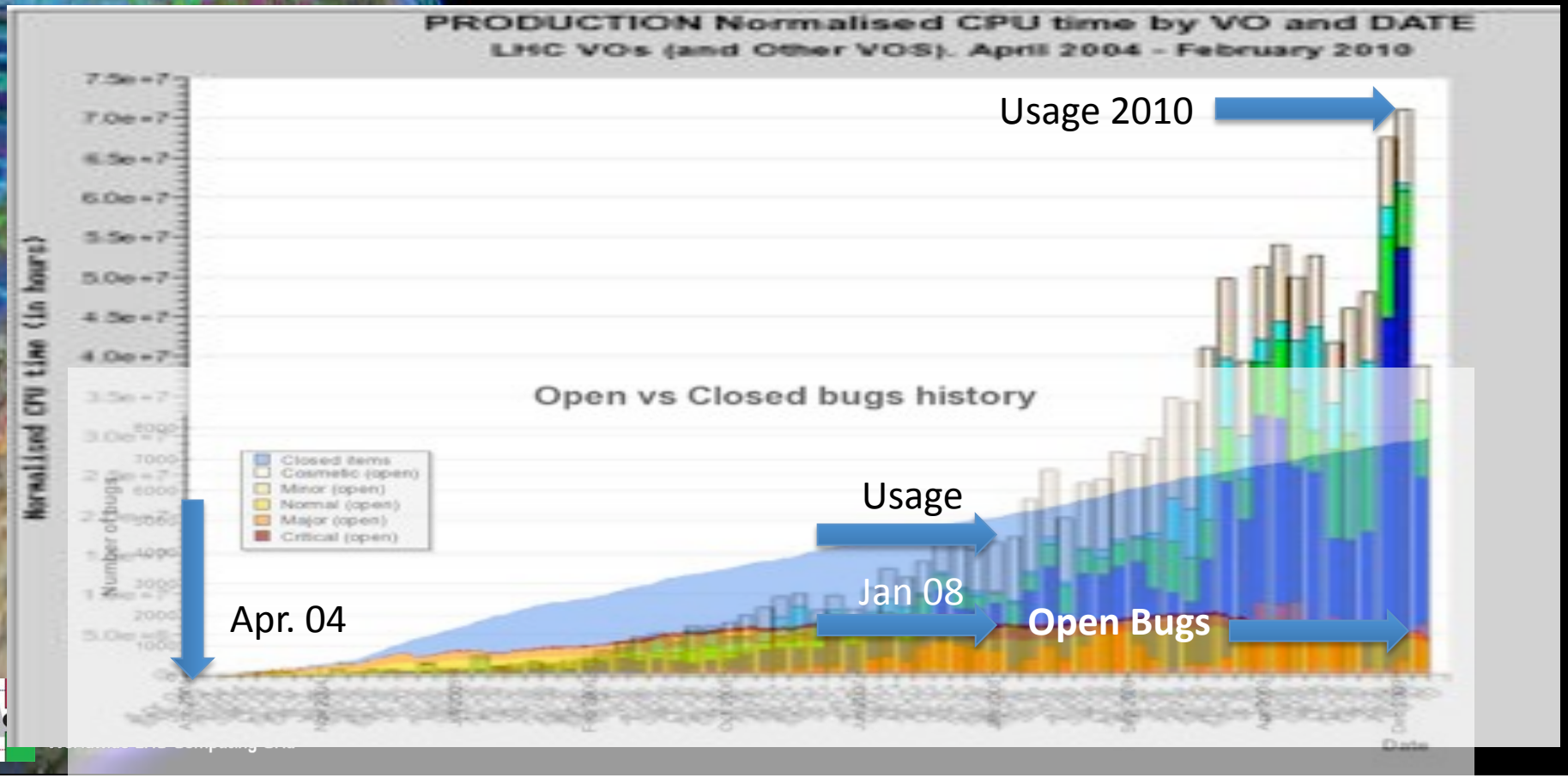
- Builds and test are mostly managed by ETICS
- Configuration management by YAIM
  - modular bash shell script
    - >37 000 lines, >30 modules
- Complex certification and release process

YAIM

WLCG Worldwide LHC Computing Grid

Error rates / Usage

- Bug rate almost flat
- Exponential increase in usage
- Example: gLite



How does the futlook like?

- Focus on standardization and interoperation
  - Driving the process
    - OGF etc.
- Focus on stability
- Simplifying the system
- Integrating virtualization
- Integrating Clouds
- Moving to EMI

WLCG Worldwide LHC Computing Grid

European Middleware Initiative (EMI)

Primary Objectives

Consolidate

Consolidate the existing middleware distribution simplifying services and components to make them more sustainable (including use of off-the-shelf and commercial components whenever possible)

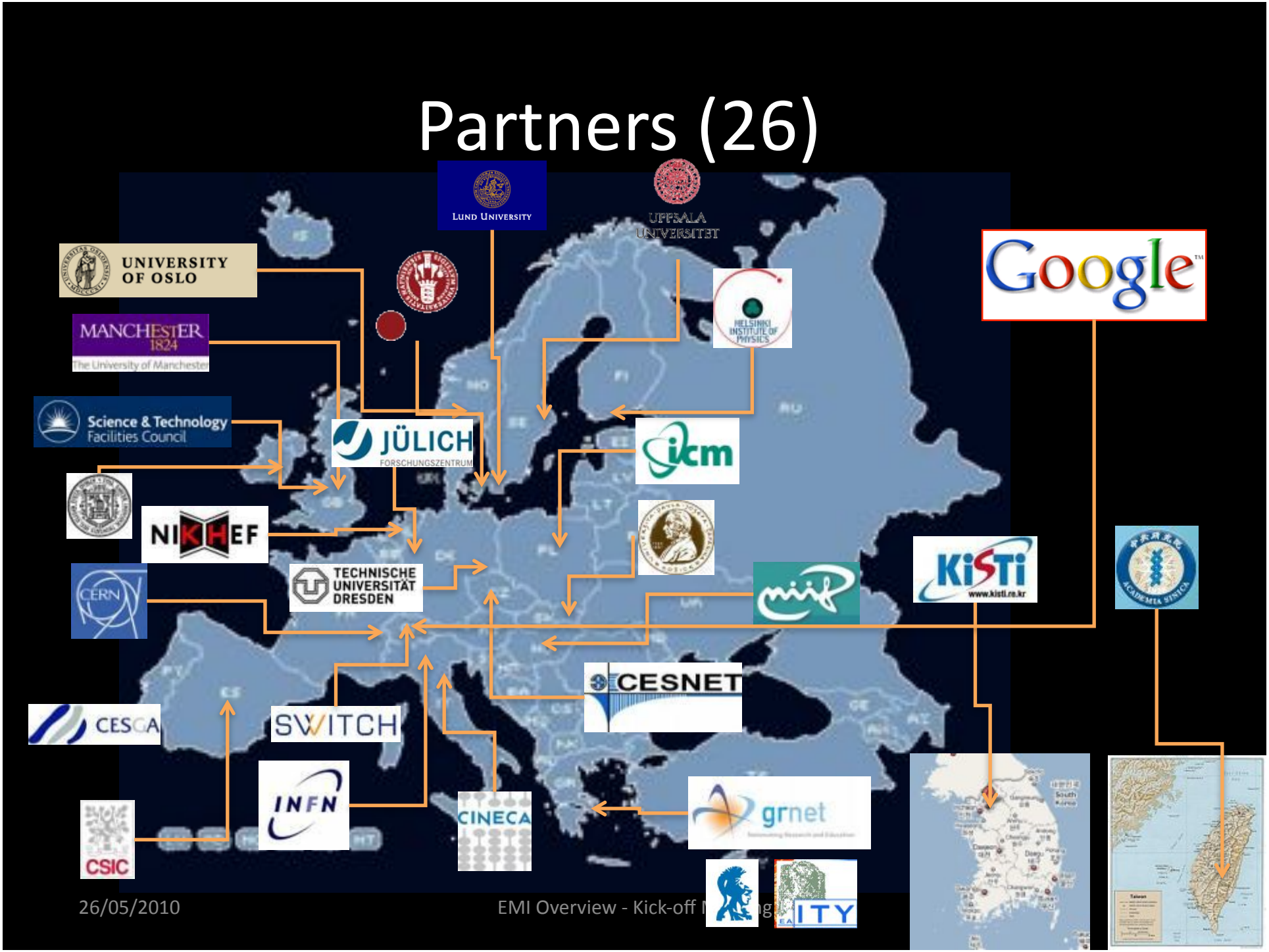
Evolve

Evolve the middleware services/functionality following the requirement of infrastructure and communities, mainly focusing on operational, standardization and interoperability aspects

Support

Reactively and proactively maintain the middleware distribution to keep it in line with the growing infrastructure usage

Partners (26)


Lund University
Uppsala Universitet
University of Oslo
Google
Manchester 1824
The University of Manchester
Helsinki Institute of Physics
Science & Technology Facilities Council
Jülich Forschungszentrum
icm
NIKHEF
KiSTi
www.kisti.re.kr
CERN
Technische Universität Dresden
CESNET
CESGA
SWITCH
INFN
CINECA
grnet
CSIC
ITY

Technical Areas

Compute Services	A-REX, UAS-Compute, WMS, CREAM, MPI, etc
Data Services	dCache, StoRM, UAS-Data, DPM, LFC, FTS, Hydra, AMGA, etc
Security Services	UNICORE Gateway, UVOS/VOMS/VOMS-Admin, ARGUS, SLCS, glExec, Gridsite, Proxyrenewal, etc
Infrastructure Services	Logging and Bookkeeping, Messaging, accounting, monitoring, virtualization/clouds support, information systems and providers