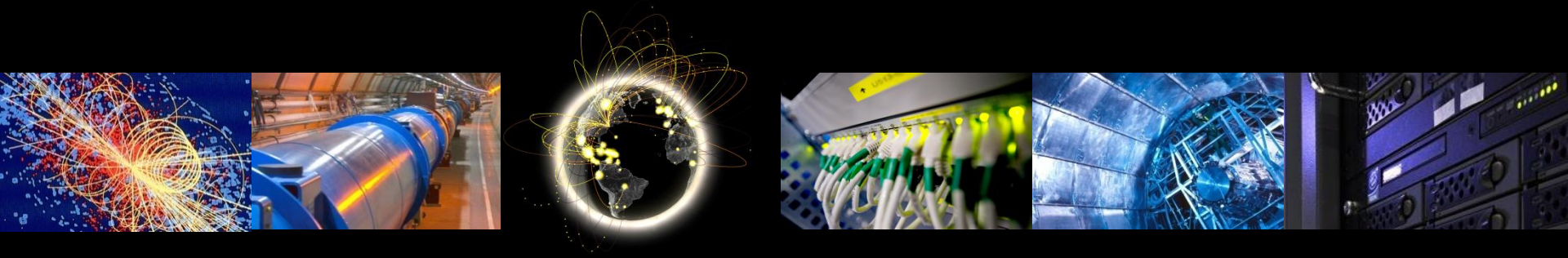
Grid Middleware

David Smith
IT Grid Technology Group, CERN

July 2012
Openlab Summer Student Lecture



WLCG
Worldwide LHC Computing Grid

Overview

- Grid Computing?
- Constraints
- EMI
  - Overview
  - Security Model
  - Data Management
  - Why is this difficult?
- Future

WLCG
Worldwide LHC Computing Grid

Overview

- If you want to use EMI read the user documentation:
- http://www.eu-emi.eu/documentation/

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

WLCG
Worldwide LHC Computing Grid

Impossible to discuss all components

- Illustrate complexity
- Some Security Details
- Bit of Data Management

WLCG
Worldwide LHC Computing Grid

What is a Computing Grid?

- There are many conflicting definitions
  - Has been used for several years for marketing…
    - Now Cloud is more common….
- 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 a short time…….

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

European Middleware Initiative (EMI)

EMI is partially funded by the European Commission under Grant Agreement RI-261611

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 (24)

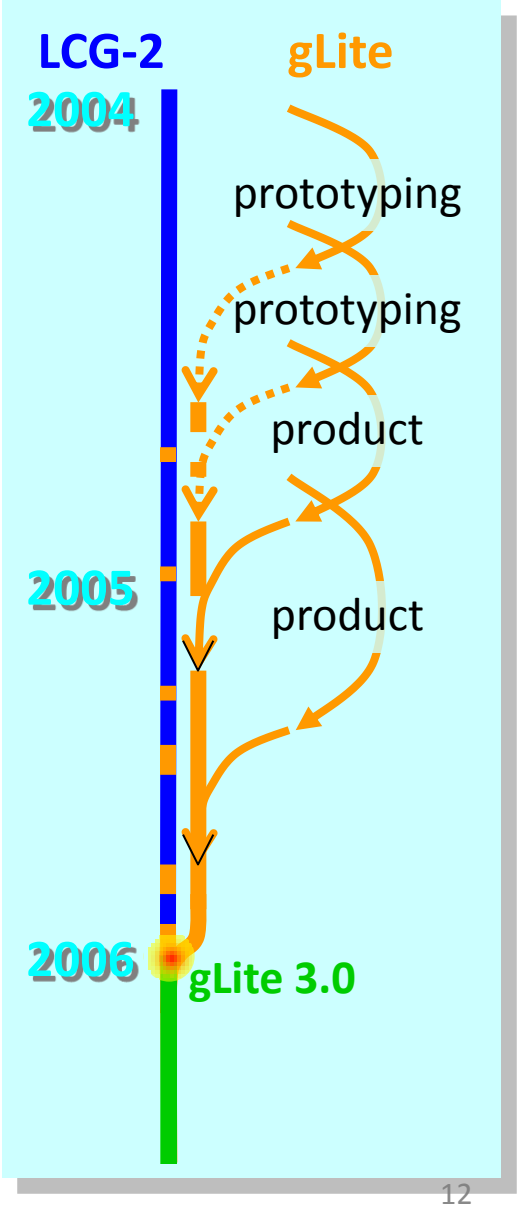

Lund University
Uppsala Universitet
University of Oslo
DESY
Helsinki Institute of Physics
Science & Technology Facilities Council
JÜLICH
Forschungszentrum
FOM
Technische Universität Dresden
CERN
KiSTi
www.kisti.re.kr
TWGrid
CESNET
CESGA
SWITCH
INFN
CINECA
grnet
CSIC

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**

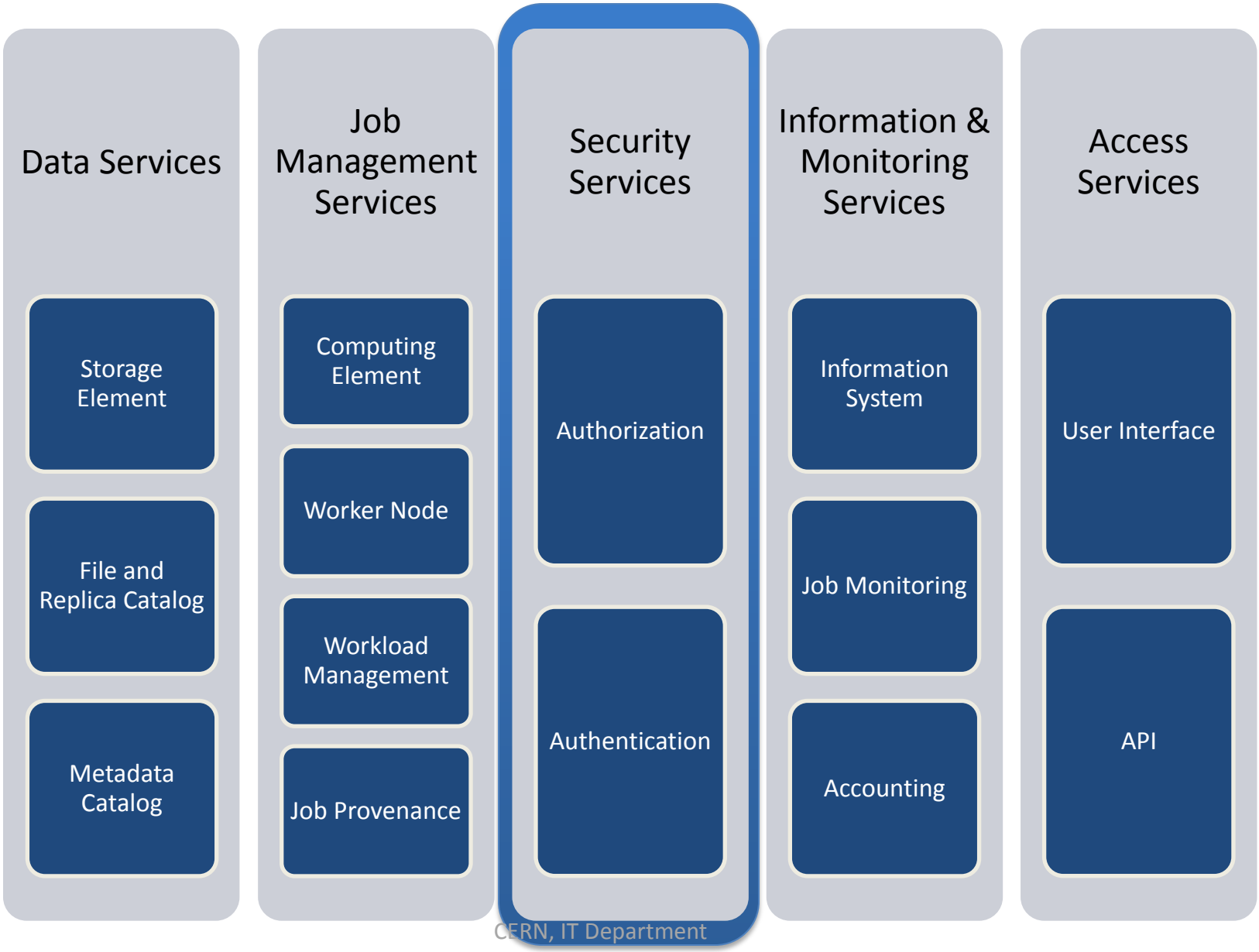
EMI Middleware Distribution

- Combines components from different providers
  - Condor, LCG-2 and Globus 5
  - ARC
  - gLite (from EDG, EGEE I, II & III projects)
  - dCache, StoRM, UNICORE, Nagios probes, …
- After prototyping phases in 2004 and 2005
  - convergence with LCG-2 distribution reached in May 2006
  - gLite 3.0, 3.1 & 3.2
  - EMI Project starts 2010
- Focus on providing a deployable MW distribution for EGI production service

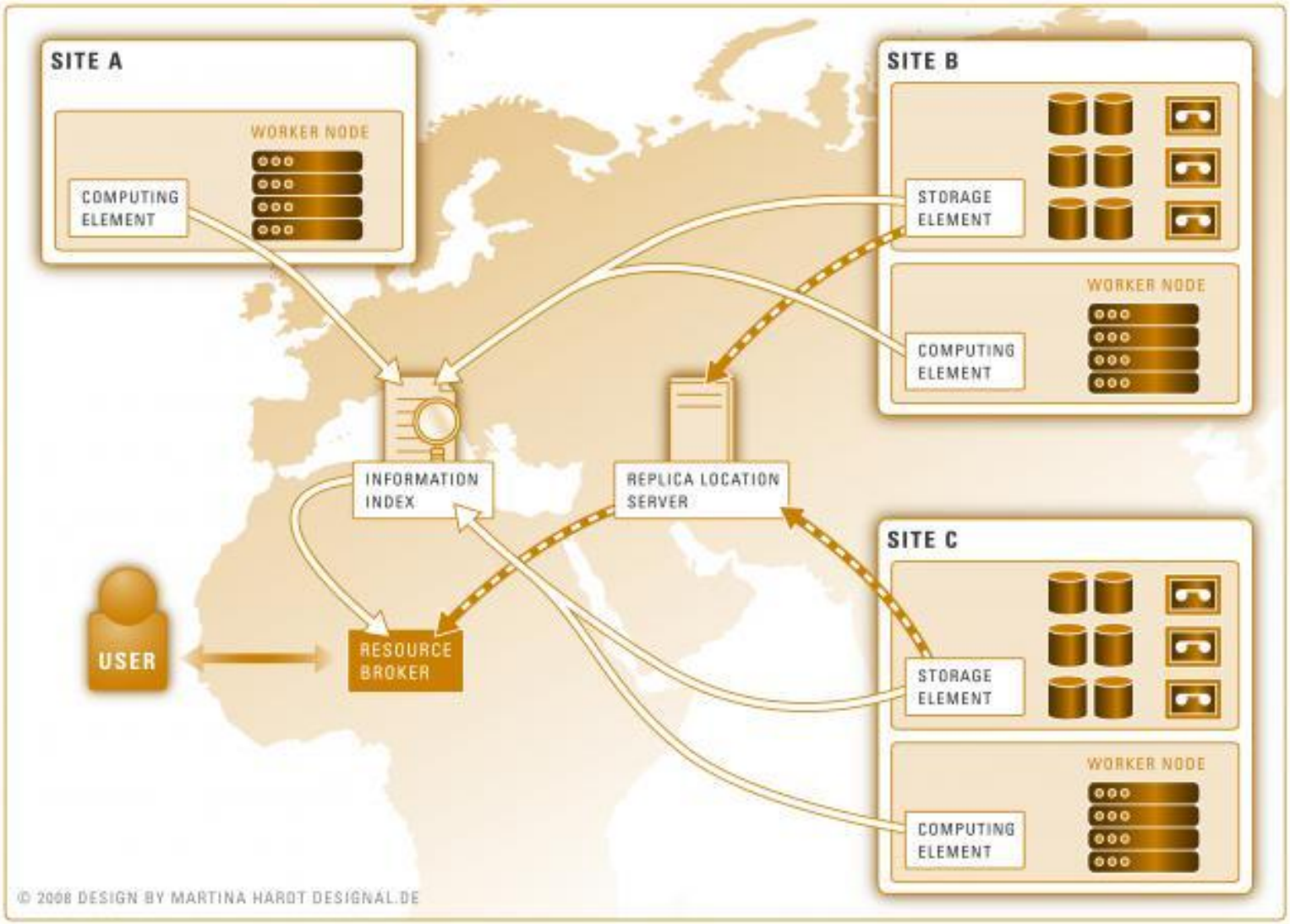


gLite
Lightweight Middleware for Grid Computing

EMI Middleware

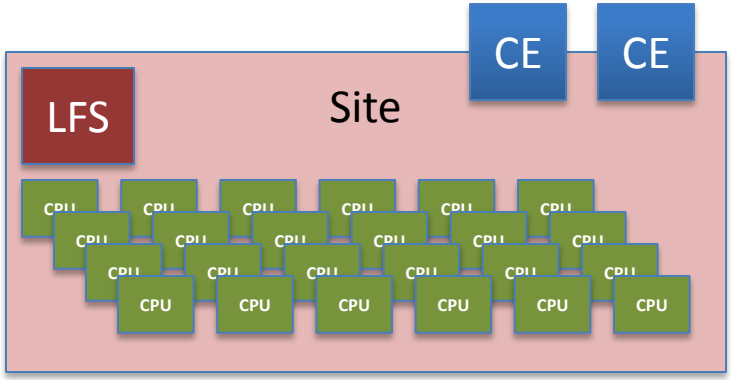


The Big Picture



Computing Access

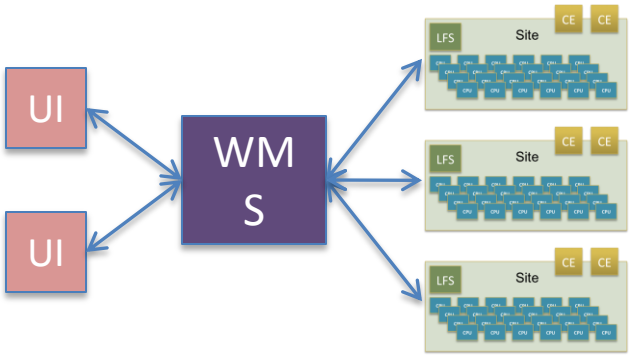
- Computing Elements (CE)
  - gateways to farms



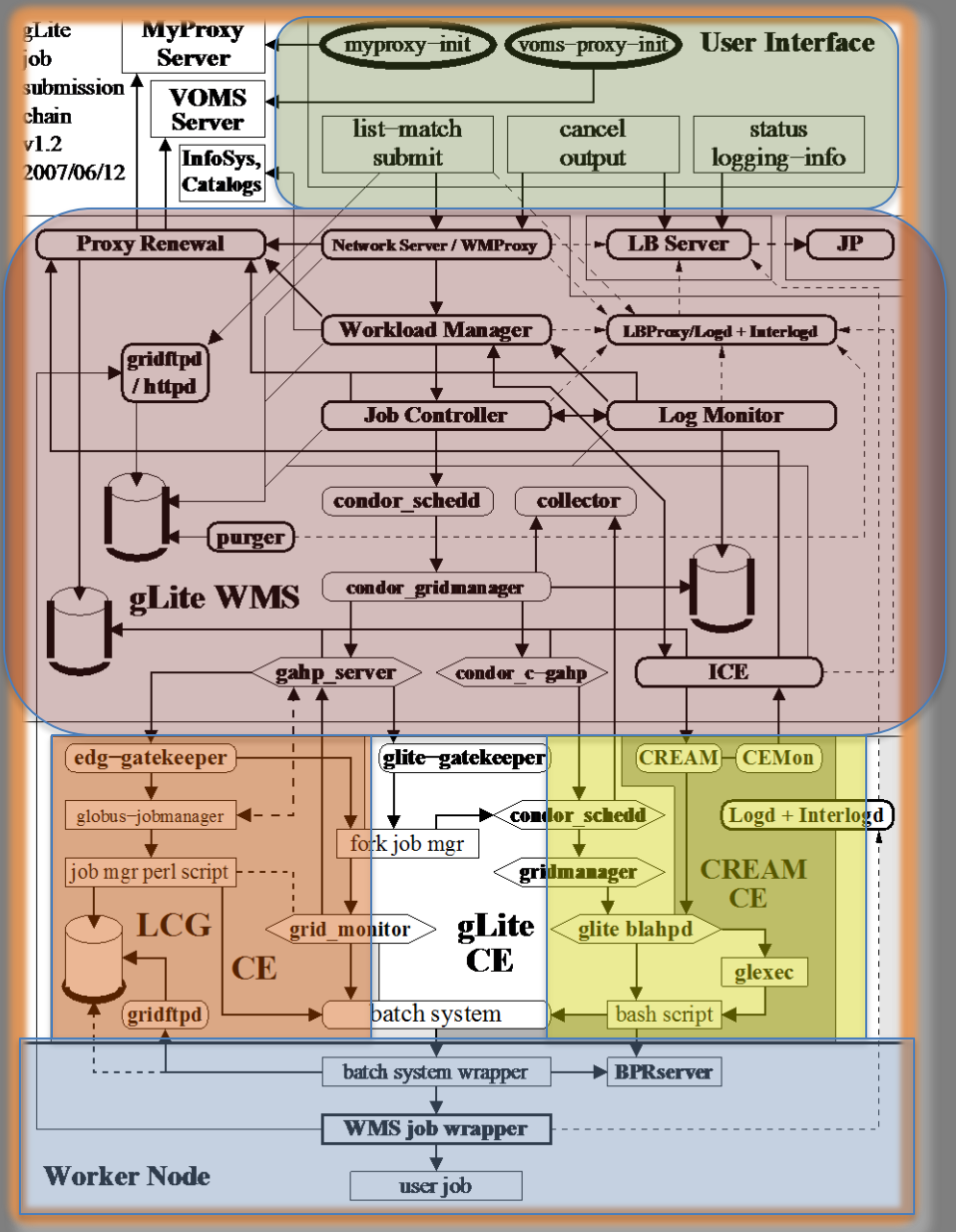
- On EGI from EMI:
  - LCG-CE (37instances)
    - LEGACY (this was the workhorse until 2 years ago)
  - ARC-CE (11 instances)
    - NorduGrid compute element
  - CREAM-CE (250 instances)
    - Significant investment on production readiness and scalability
    - Handles direct submission (pilot job friendly)
    - SL5/SL6
    - ICE interfaces available (in the future EMI-ES); supports parameter passing from grid <-> batch

Workload Management

- EMI WMS/LB
  - Matches resources and requests
    - Including data location
  - Handles failures (resubmission)
  - Manages complex workflows
  - Tracks job status
- EMI WMS/LB (149 Instances)
  - Fully supports LCG-CE, CREAM-CE and ARC-CE
    - Early versions had some WMS<->CREAM incompatibilities



Workload Management (compact)



Desktops

A few ~150 nodes

1-20 per site

1-24000 per site

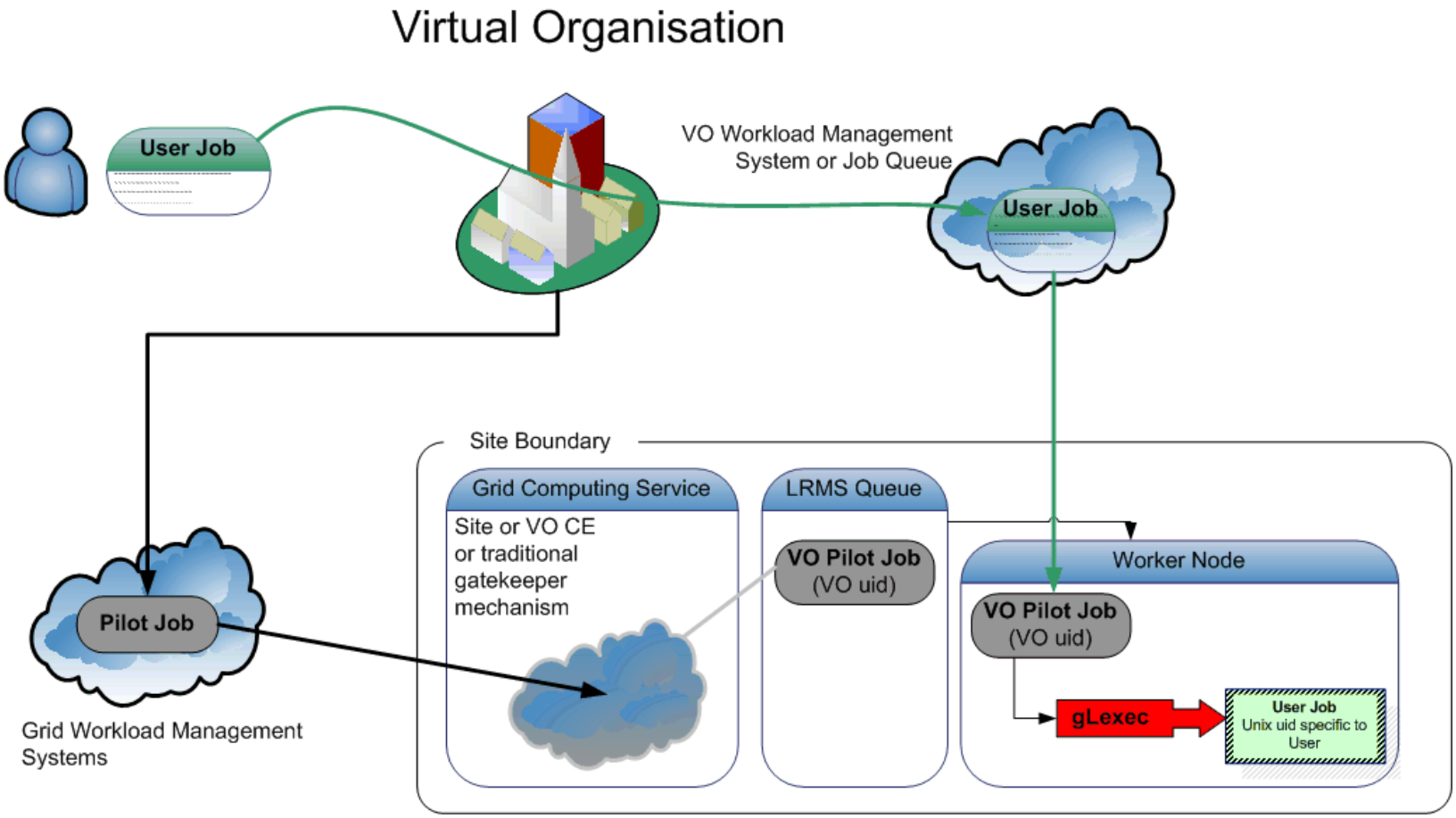
WLCG Worldwide LHC Computing Grid

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
]
```

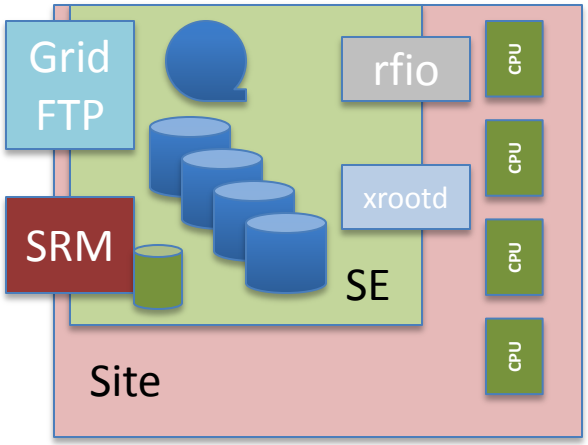
MultiUserPilotJobs

- 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

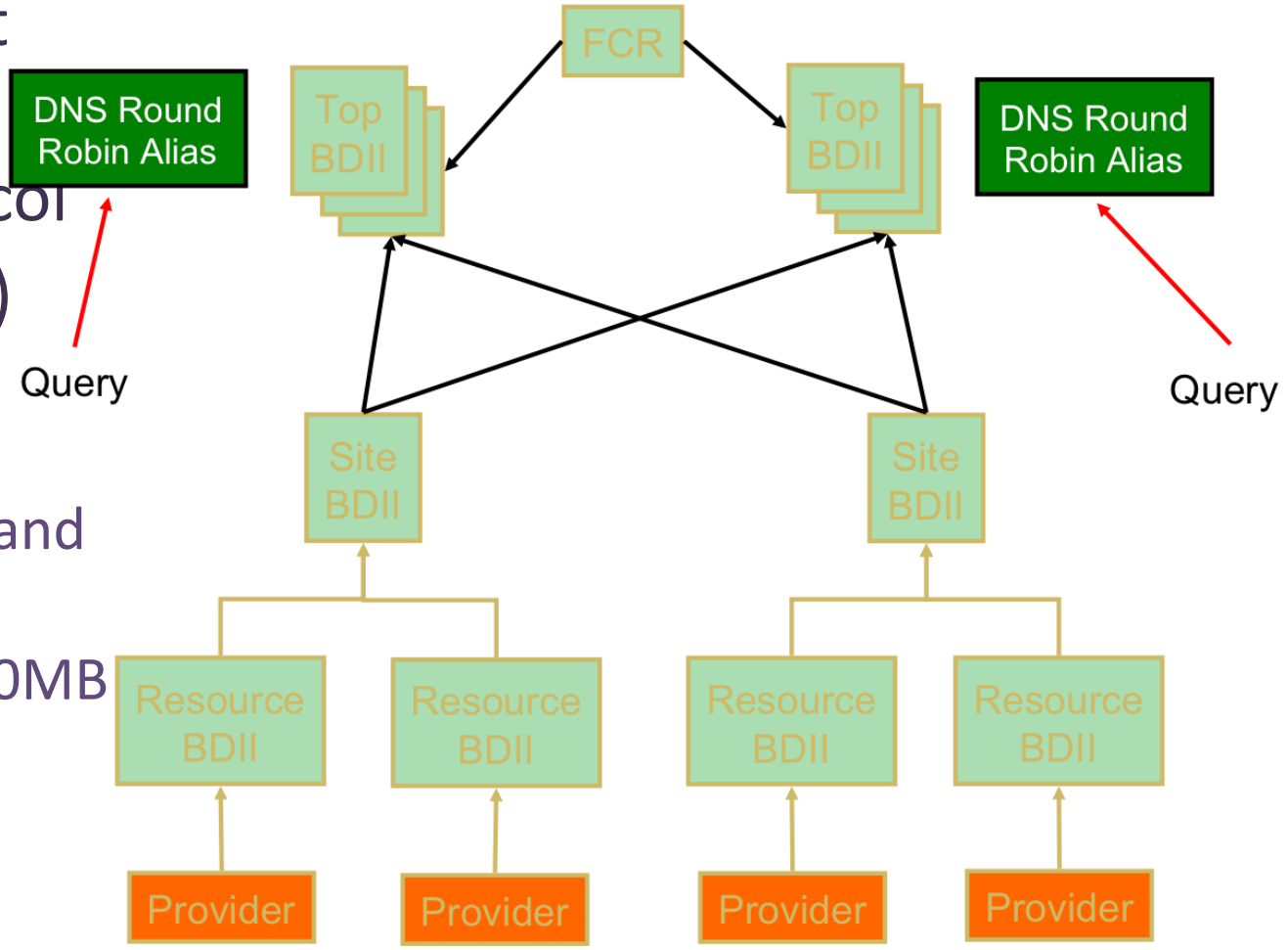
- Storage Elements (SEs)
  - External interfaces based on SRM 2.2 and gridFTP
  - Local interfaces: POSIX, dcap, secure rfio, rfio, xrootd
  - DPM (245)
  - dCache (80)
  - STORM (50)
  - BestMan (50)
  - CASTOR (20)
  - "ClassicSE" (50) → legacy since 3 years….



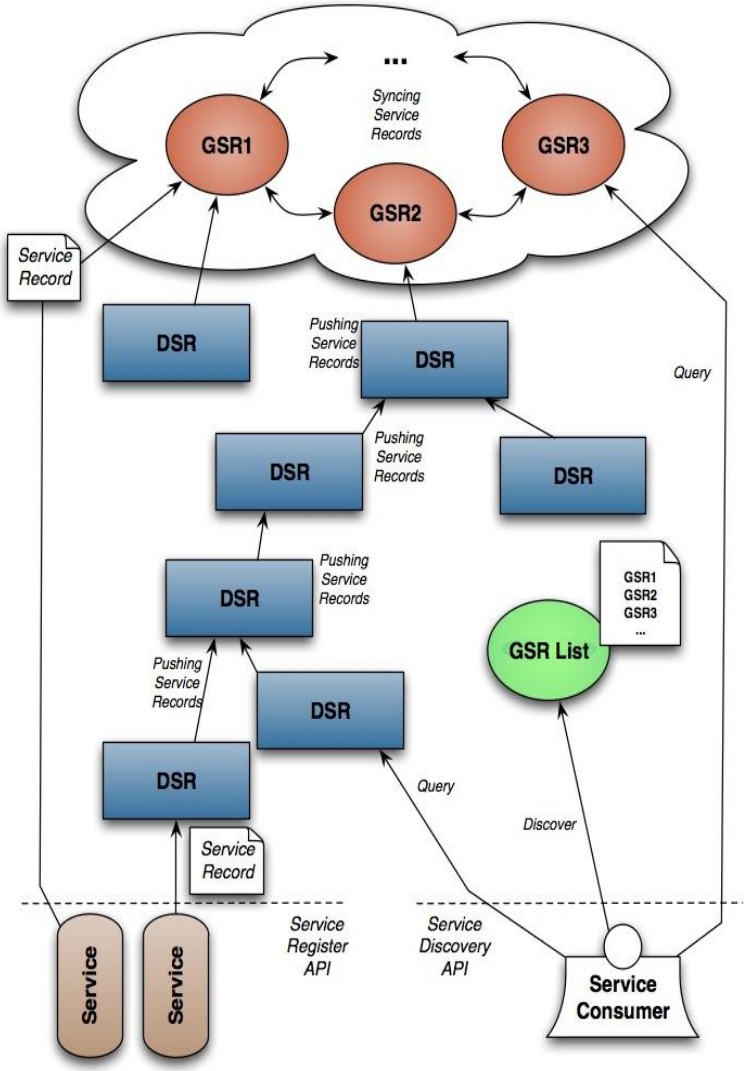
- Catalogue: LFC (local and global)
- File Transfer Service (FTS)
- Data management clients gfal/LCG-Utils

Information System

- BDII
- Light weight Database
- LDAP protocol
- GLUE 2 (1.3) Schema
  - Describes resources and their state
  - Approx 170MB
- Several hundred instances



New system: EMIR

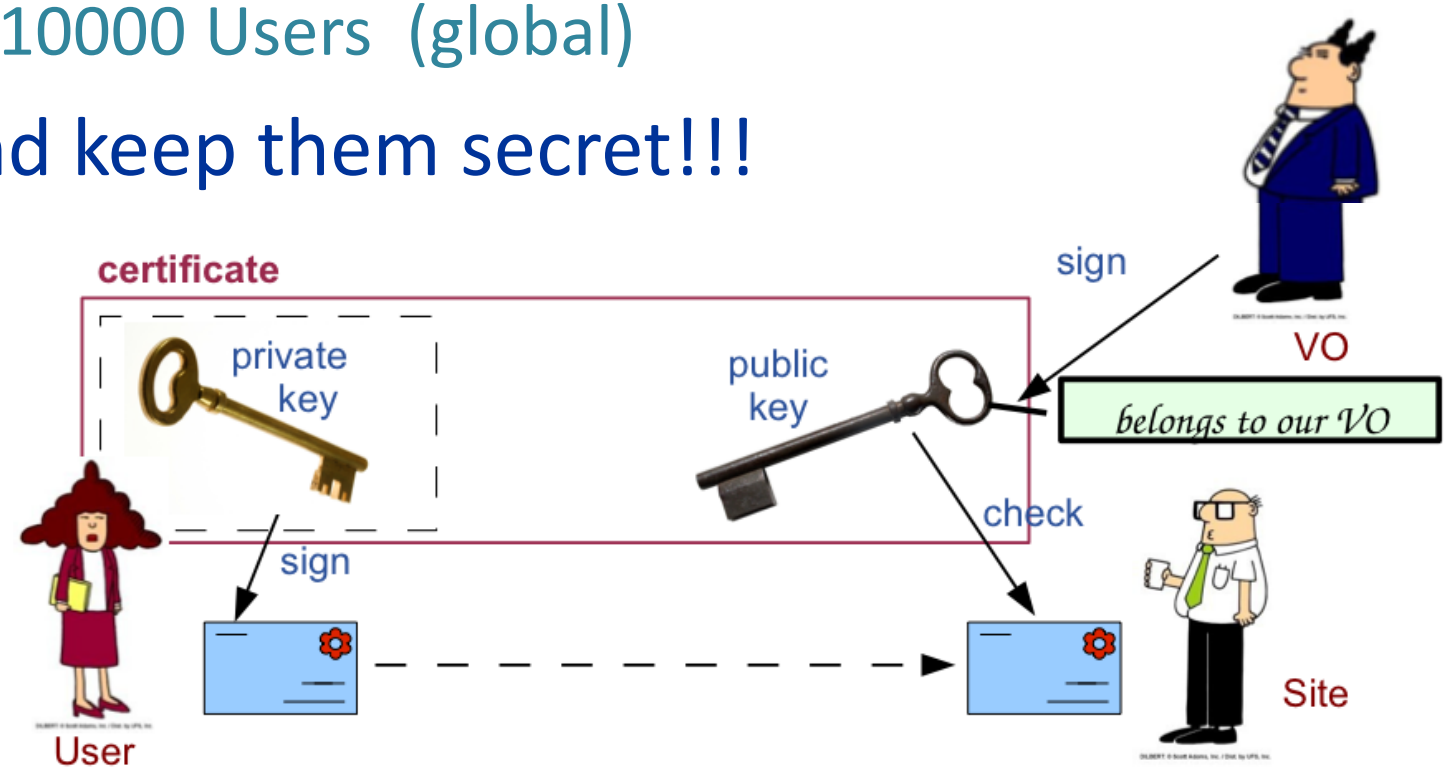


Authentication

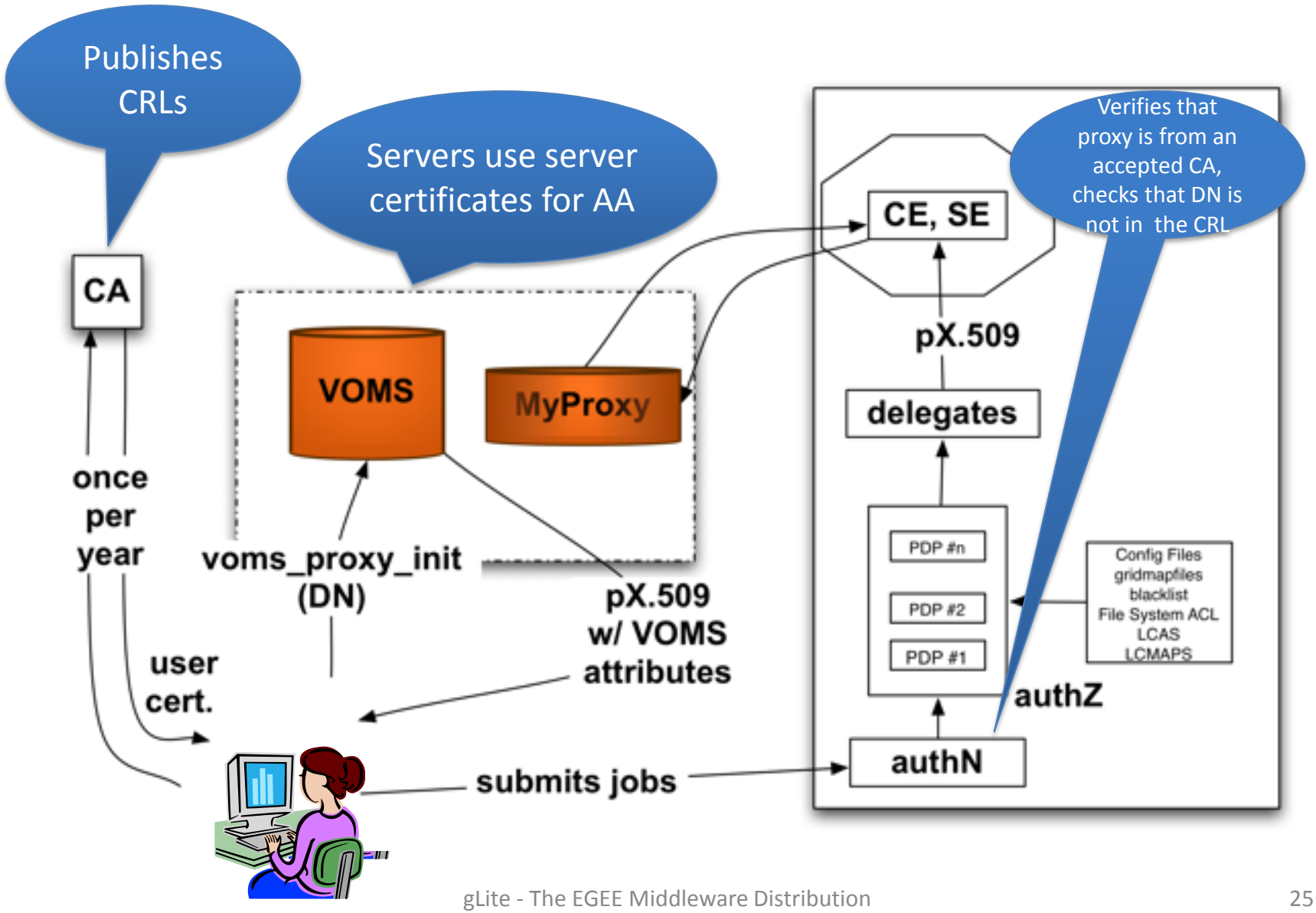
- Authentication is based on X.509 PKI infrastructure (Public Key)
  - Certificate Authorities (CA) issue (long lived) certificates identifying individuals (much like a passport)
    - Commonly used in web browsers to authenticate to sites
  - Trust between CAs and sites is established (offline)
  - 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
  - Identify a user to a service
  - Be delegated to a service such that it can act on the user's behalf
  - Be renewed by external store (MyProxy)
  - 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
- Job isolation: 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
  - Eventual uniform implementation of authorization in EMI services
  - Easier interoperability with other infrastructures
  - Prototype being prepared now

CAs and all that

- Certification Authorities
  - And Registration Authorities
  - have to be recognized by the International Grid Trust Federation (http://www.igtf.net/)
  - federation of the APGridPMA, The Americas Grid PMA and EUGridPMA
    - http://www.eugridpma.org/members/worldmap/
  - igtf maintains a list of accredited CAs
    - their public keys and URL for getting CRLs
  - CRL Certificate Revocation Lists
    - contain “bad” DNs from a CA to block users

What's in a Credential?

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 5 (0x5)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=CH, O=CERN, OU=cern.ch, CN=CERN CA
        Validity
            Not Before: Sep 11 11:37:57 2010 GMT
            Not After : Nov 30 12:00:00 2011 GMT
        Subject: O=Grid, O=CERN, OU=cern.ch, CN=John Doe
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA  Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:ab:8d:77:0f:56:d1:00:09:b1:c7:95:3e:ee:5d:
                    c0:af:8d:db:68:ed:5a:c0:17:ea:ef:b8:2f:e7:60:
                    2d:a3:55:e4:87:38:95:b3:4b:36:99:77:06:5d:b5:
                    4e:8a:ff:cd:da:e7:34:cd:7a:dd:2a:f2:39:5f:4a:
                    0a:7f:f4:44:b6:a3:ef:2c:09:ed:bd:65:56:70:e2:
                    a7:0b:c2:88:a3:6d:ba:b3:ce:42:3e:a2:2d:25:08:
                    92:b9:5b:b2:df:55:f4:c3:f5:10:af:62:7d:82:f4:
                    0c:63:0b:d6:bb:16:42:9b:46:9d:e2:fa:56:c4:f9:
                    56:c8:0b:2d:98:f6:c8:0c:db
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Base Url:
                http://home.cern.ch/globus/ca
            Netscape Cert Type:
                SSL Client, S/MIME, Object Signing
            Netscape Comment:
               For DataGrid use only
            Netscape Revocation Url:
                http://home.cern.ch/globus/ca/bc870044.r0
            Netscape CA Policy Url:
                http://home.cern.ch/globus/ca/CPS.pdf
    Signature Algorithm: md5WithRSAEncryption
        30:a9:d7:82:ad:65:15:bc:36:52:12:66:33:95:b8:77:6f:a6:
        52:87:51:03:15:6a:2b:78:7e:f2:13:a8:66:b4:7f:ea:f6:31:
        aa:2e:6f:90:31:9a:e0:02:ab:a8:93:0e:0a:9d:db:3a:89:ff:
        d3:e6:be:41:2e:c8:bf:73:a3:ee:48:35:90:1f:be:9a:3a:b5:
        45:9d:58:f2:45:52:ed:69:59:84:66:0a:8f:22:26:79:c4:ad:
```

Who has issued it

Start and end date

DN == Identity

Public Key

RSA

- $n = p \cdot q$ (p, q prime)
- $\varphi(n) = (p-1)(q-1)$
- $1<e<\varphi(n)$, $\gcd(e, \varphi(n)) = 1$
- $d = e^{-1} \pmod{\varphi(n)}$

- (n, e) is the public key
- (n, d) is the private key

e.g. for a message m

- $c = m^e \pmod{n}$ (operation with public key)
- $m = c^d \pmod{n}$ (operation with private key)

VOMS Proxy?

- VO membership, Group and Role
  - authorization based on FQAN
    - Fully Qualified Attribute Name
    - <group name>[/Role=<role name>]
    - example: /cms/SusySearch/Role=Simulation
  - VO manages VO group and role membership
  - User requests attributes
- Delegation
- Time limited
  - for long running tasks → MyProxy Service
    - renews proxies, known services can retrieve new proxies

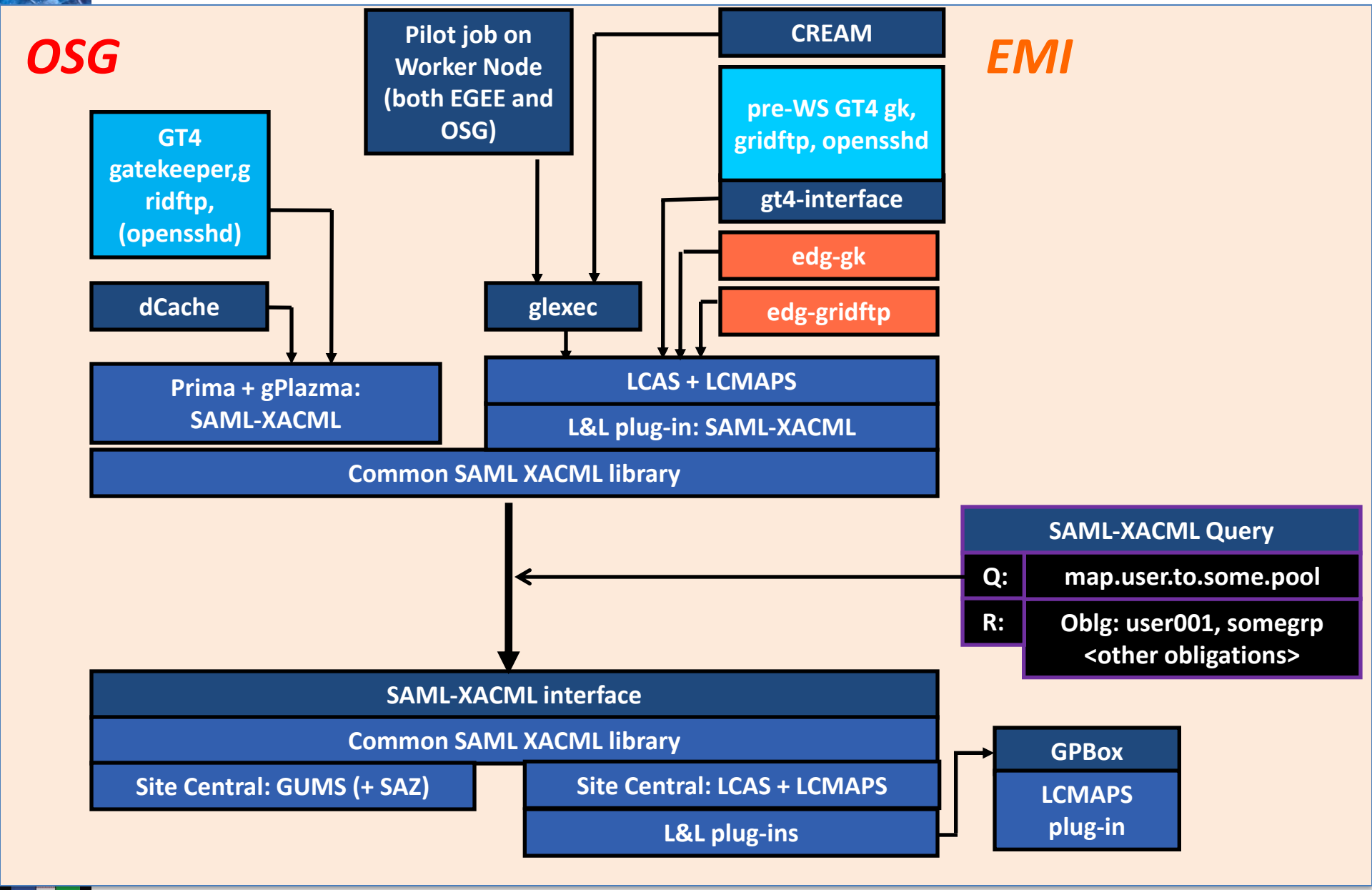
Problems with grid Security

- Public Key infrastructure
  - all CAs have to provide updated CRLs in time
- Computational overhead
  - each authentication costs about 7 round trips
  - Roughly 10ms computing
- gsi version of x509 makes session reuse difficult
  - Overhead for “small” operations
- User handling of private key (confidential)
- The “error phase space” is large
  - needs expertise to debug

Problems with grid Security 2

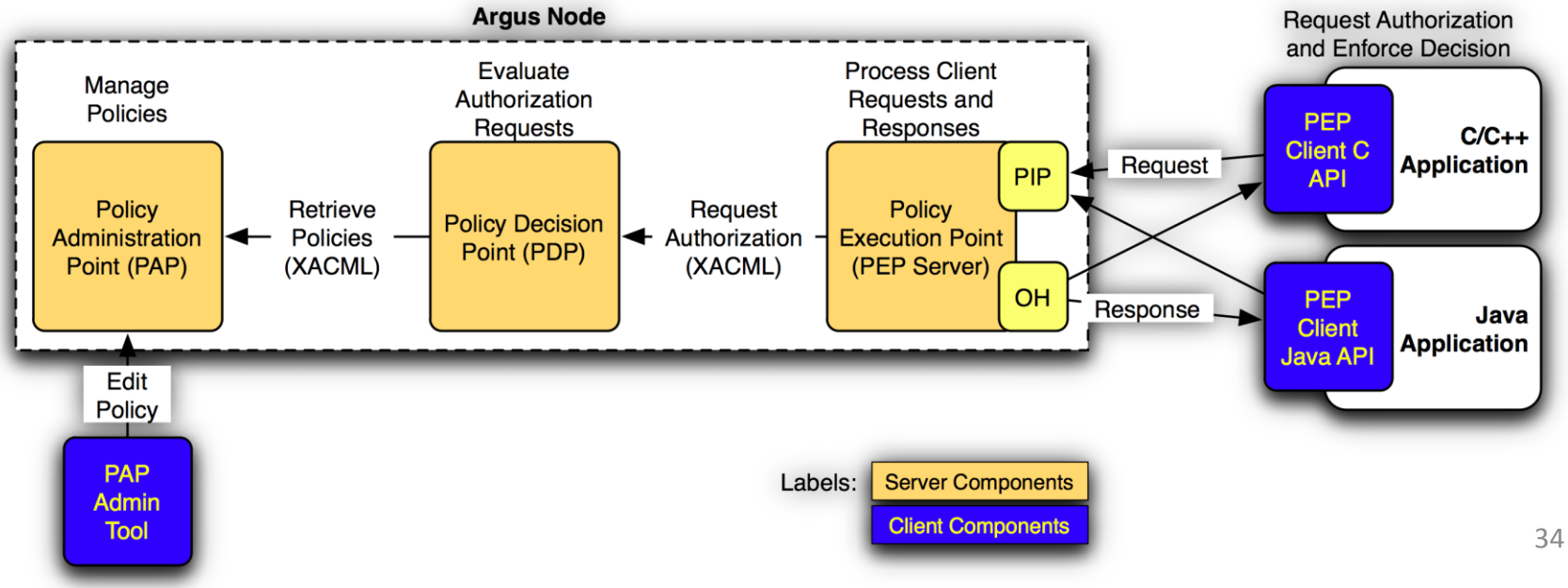
- In competition with existing systems
  - Shibboleth Identity Provider, Kerberos ....
  - Interfaces have been created KCA etc.
    - but not 1:1 mapping
- Enforcing site wide policies is difficult
  - each service handles banning independently
- VOMS
  - very flexible (maybe too flexible)
  - multiple roles
    - but no clear discrimination between data management and job prio
    - the semantic is not defined (what can I do with the role)
  - mapping to underlying fabric difficult
    - Often UNIX group and user IDs.....
  - Different infrastructures and services use different implementation
    - Interoperability !!!
    - Sysadmins are challenged

Common Authentication interface



ARGUS: Authorization

- System for consistent authorization
  - Based on global and site wide policies
  - PAP Policy Administration Point
  - PDP Policy Decision Point
  - PEP Policy Enforcement Point
  - EES Execution Environment Service mapping for users
  - System to combine global and local policies



Will it be easy?

- XACML is too complex → SPL
  - simplified policy language

```
<xacml:PolicySet xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
        combining-algorithm:first-applicable" PolicySetId="9784d9ce-16a9-41b9-9d26-b81a97f93616" Version="1">
   <xacml:Target>
      <xacml:Resources>
         <xacml:Resource>
            <xacml:ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
               <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">.*</xacml:AttributeValue>
               <xacml:ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false"/>
            </xacml:ResourceMatch>
         </xacml:Resource>
      </xacml:Resources>
   </xacml:Target>
   <xacml:PolicyIdReference>public_2d8346b8-5cd2-44ad-9ad1-0eff5d8a6ef1</xacml:PolicyIdReference>
</xacml:PolicySet>
<xacml:Policy xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os" PolicyId="public_2d8346b8-5cd2-44ad-9ad1-0eff5d8a6ef1"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable" Version="1">
   <xacml:Target>
      <xacml:Actions>
         <xacml:Action>
            <xacml:ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
               <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">.*</xacml:AttributeValue>
               <xacml:ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false"/>
            </xacml:ActionMatch>
         </xacml:Action>
      </xacml:Actions>
   </xacml:Target>
   <xacml:Rule Effect="Deny" RuleId="43c15124-6635-47ee-b13c-53f672d0de77">
...
```

Will it be easy?

- SPL

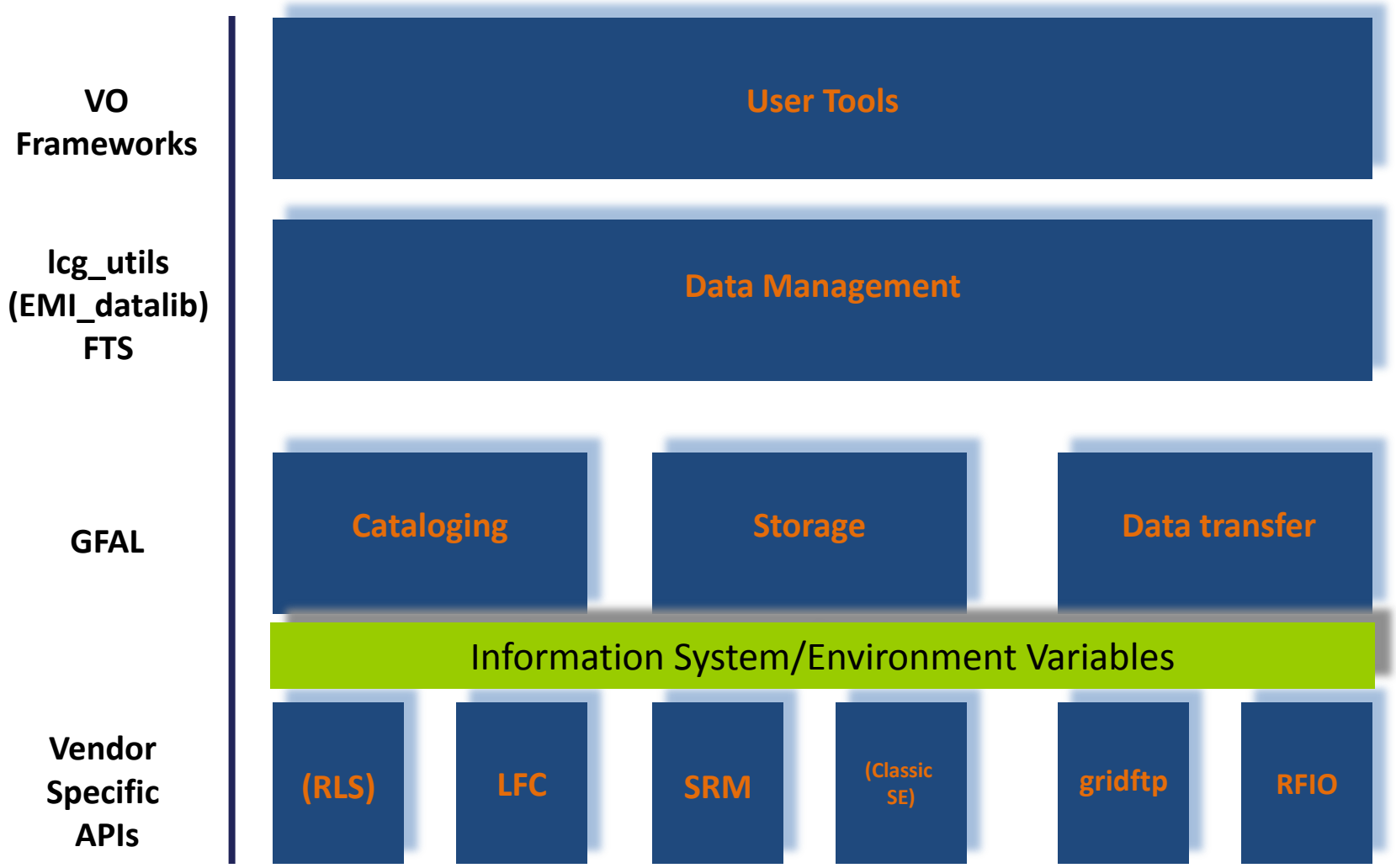
Banning

```
resource ".*" {
    action ".*" {
        rule deny {

subject="/C=CH/O=SWITCH/CN=Valery
Tschopp" }
    }
}
```

Enabling

```
resource "http://grid.switch.ch/wn" {
    action "http://glite.org/xacml/action/execute" {
        rule permit { fqan="/atlas" }
    }
}
```

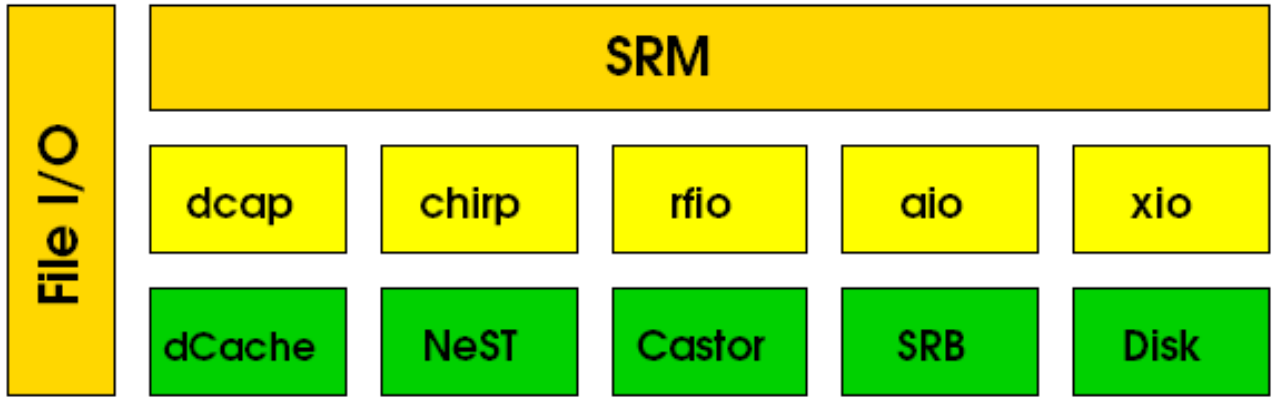
Data Management



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

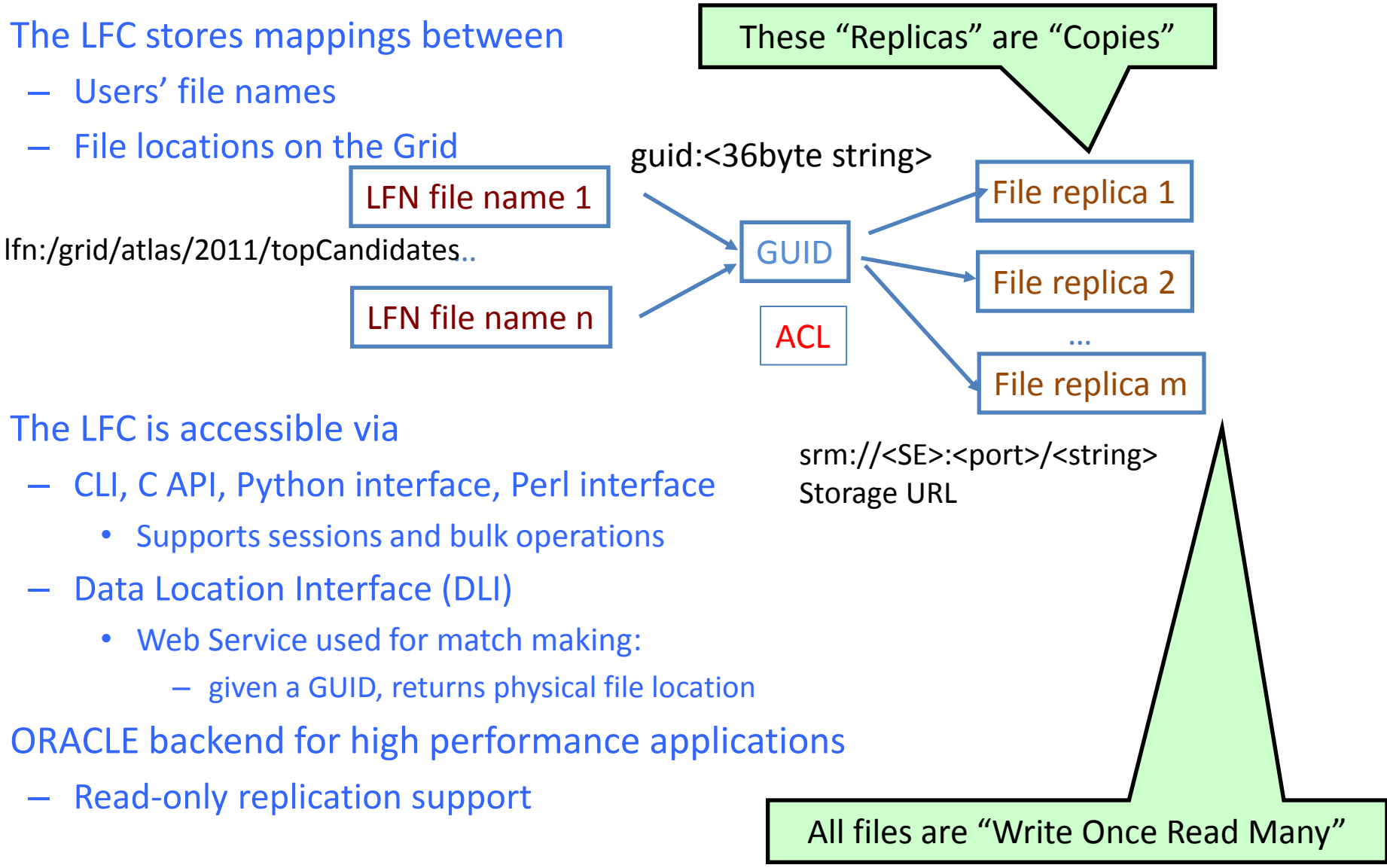
- Providing all necessary user functionality and control
  - Data Management
  - Data Access
  - Storage management
  - Control:
    - Pinning files
    - Retention Policy
    - Space management and reservation
  - Data Transfers
- Grid enabled and based on current technology
  - Interface technology (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

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



LFC features

Hierarchical Namespace

GSI security

- **Permissions and ownership**
- **ACLs (based on VOMS)**

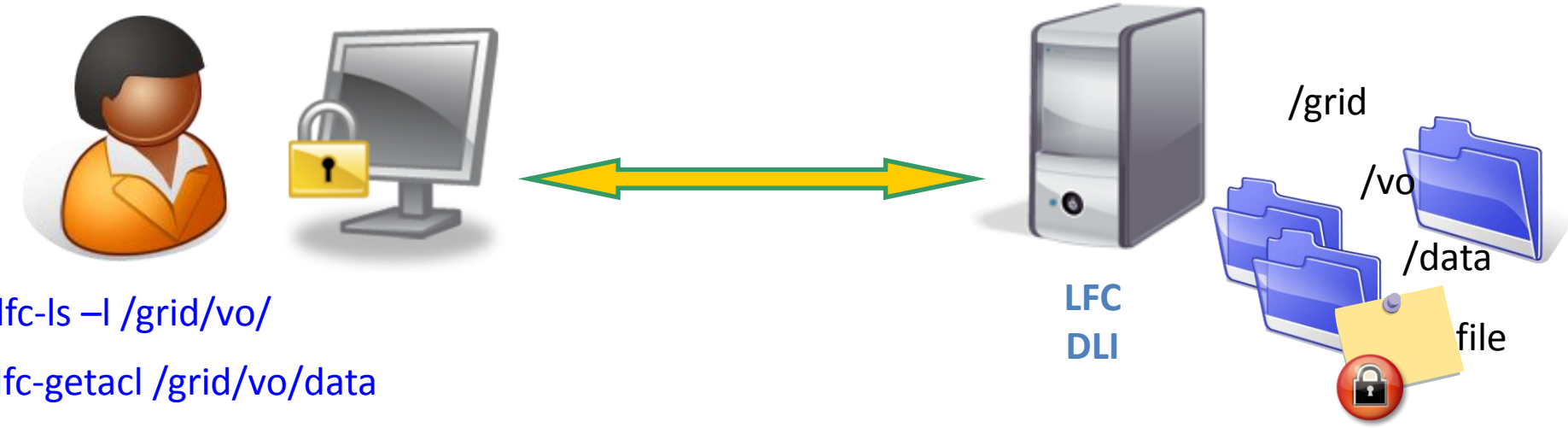
Virtual ids

- 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

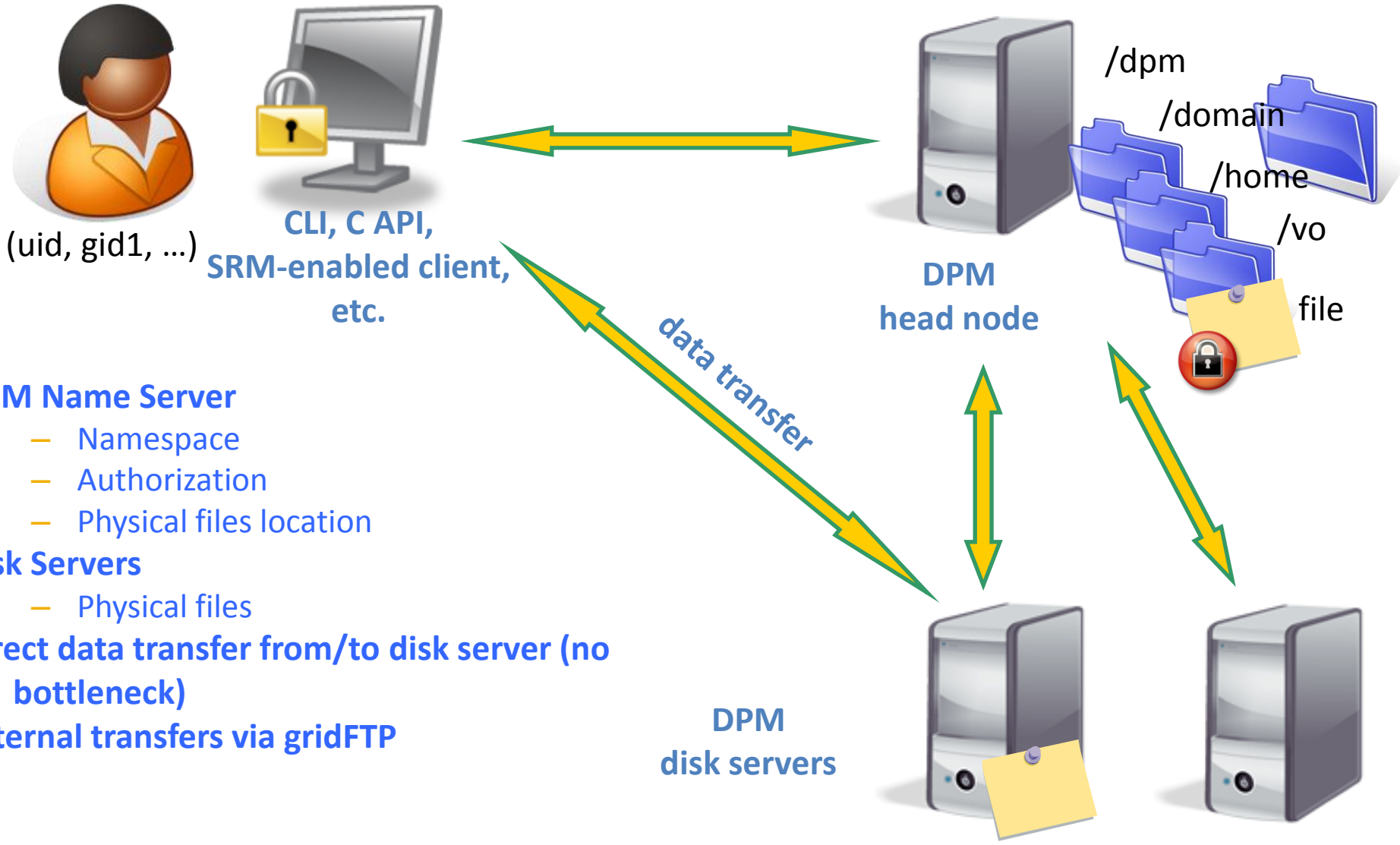
DPM

- Disk Pool Manager
  - Manages storage on disk servers
  - Decouples local namespace from SURL namespace
    - allows to add/remove disks in an invisible way
    - SURL → TURL translation
    - Transport URL: <protocol>://<string>
      - gsiftp://diskserver001.cern.ch/data/atlas/file400001
  - SRM support
    - 2.2 (released in DPM version 1.6.3)
  - GSI security
  - ACLs
  - VOMS support
  - Secondary groups support (see LFC)
  - Replication for hot files

DPM strengths

- Easy to use
  - Hierarchical namespace
    - $ dpns-ls /dpm/cern.ch/home/vo/data
  - Many protocols supported (including HTTPS, xroot, S3, NFS 4)
- Easy to administrate
  - Easy to install and configure
  - Low maintenance effort
  - Easy to add/drain/remove disk servers
- Target: small to “medium” sites (34PB total)
  - 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)

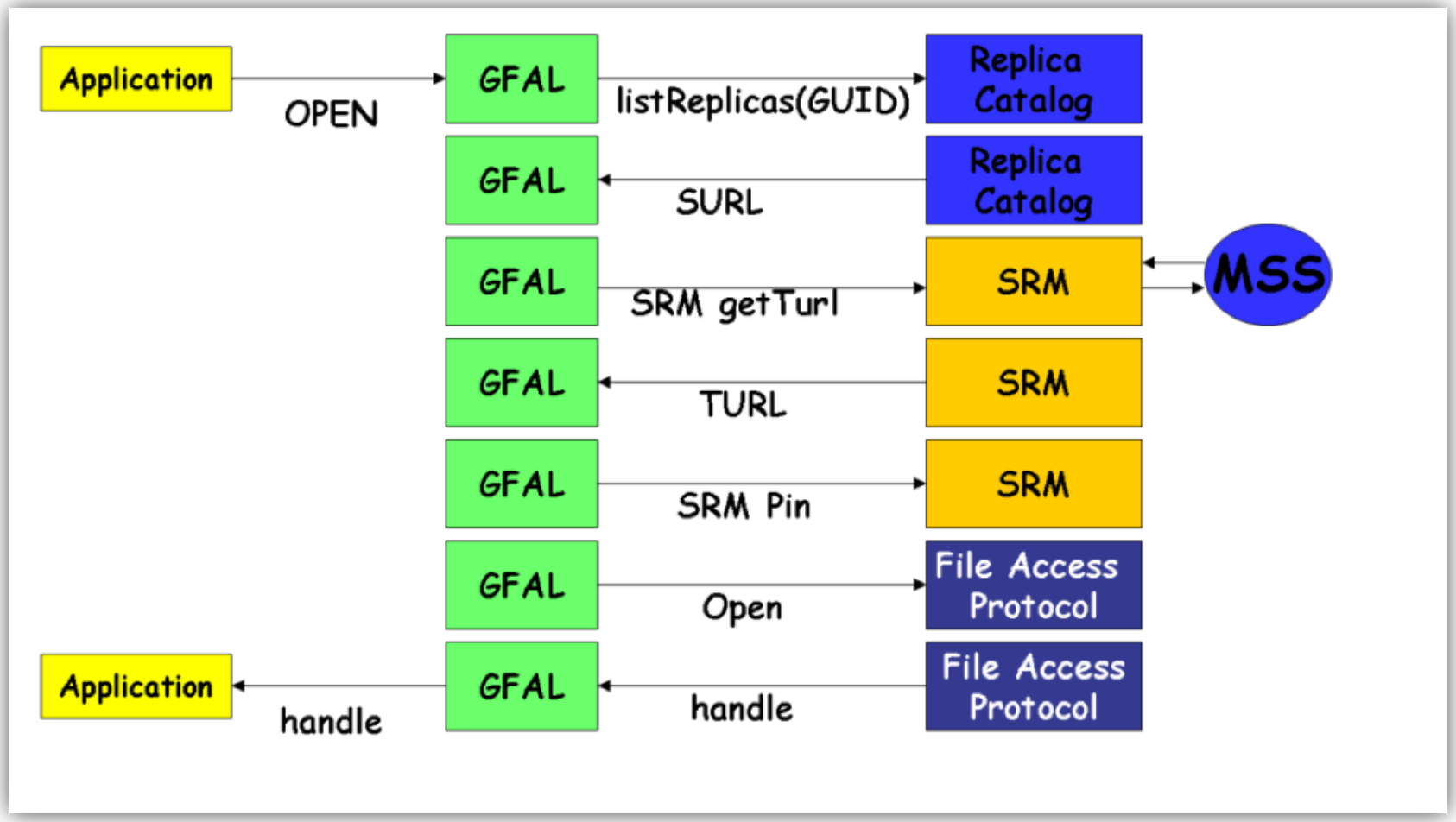
External transfers via gridFTP

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)
  - http://www.youtube.com/watch?v=dgyyFJvyK9g
- lcg_util (command line + C API)
  - Replication, catalogue interaction etc.

gfal: what really happens

- example: Open a file



gfal: some api calls

- example:
- similar to POSIX
  - but different

.....

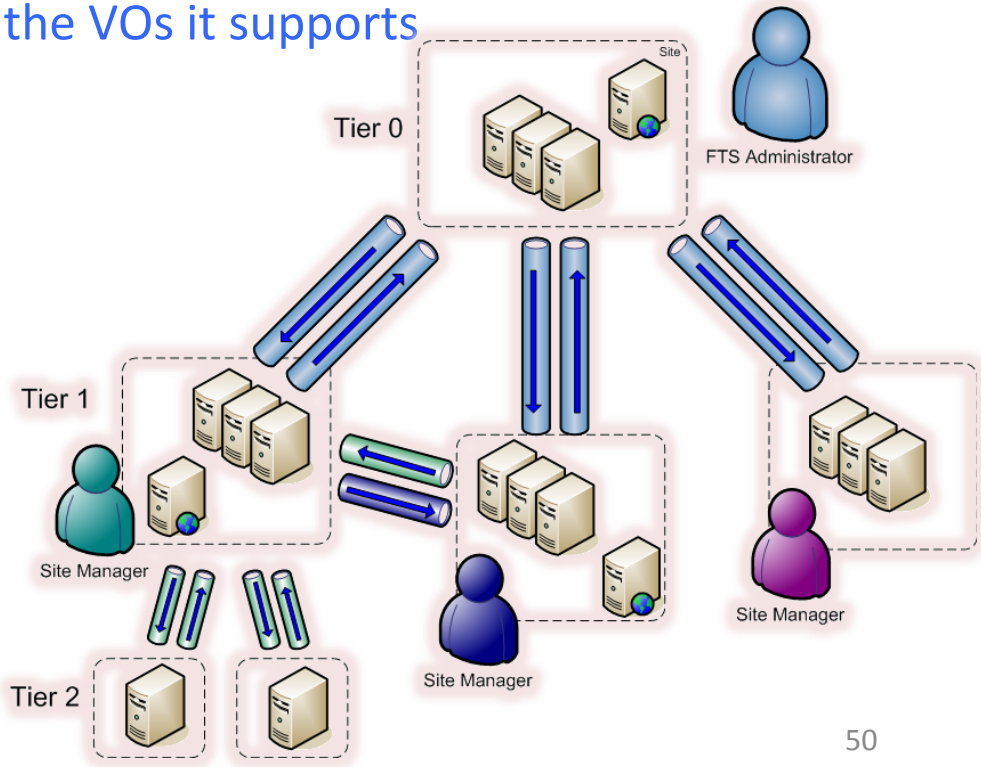
```
void call_gfal_read(char *filename, size_t block_size)
{
        int FD; //file descriptor
        int rc; //error code
        int array_size= block_size/sizeof(int);
        int* readValues= new int[ array_size ];

        if((FD = gfal_open ( filename, O_RDONLY,0 )) < 0) {
                perror ("error in gfal_open");
                exit(1);
        }
        cout << "File is successfully opened\n";

        if ((rc=gfal_read (FD, readValues, block_size)) != block_size ) {
                if (rc < 0) perror("error in gfal_read");
                else cerr << "gfal_read returns " << rc << endl;
        }
        cout << "File is successfully read\n";

        for(int i=0; i<array_size; i++)
                cout << "\treadValues[" << i << "] = " << readValues[i] << endl;

        if ((rc= gfal_close (FD)) < 0) {
                perror ("error in gfal_close");
                exit(1);
        }
        cout << "Close successful ..."<<endl;
}
```

Data Management Problems

- Access: POSIX like
  - programs need to be modified
  - solution: NFS-4.1, WebDav, fuse
- Catalogues and SEs are not synchronized
  - dark data.....
  - synchronization tool in development

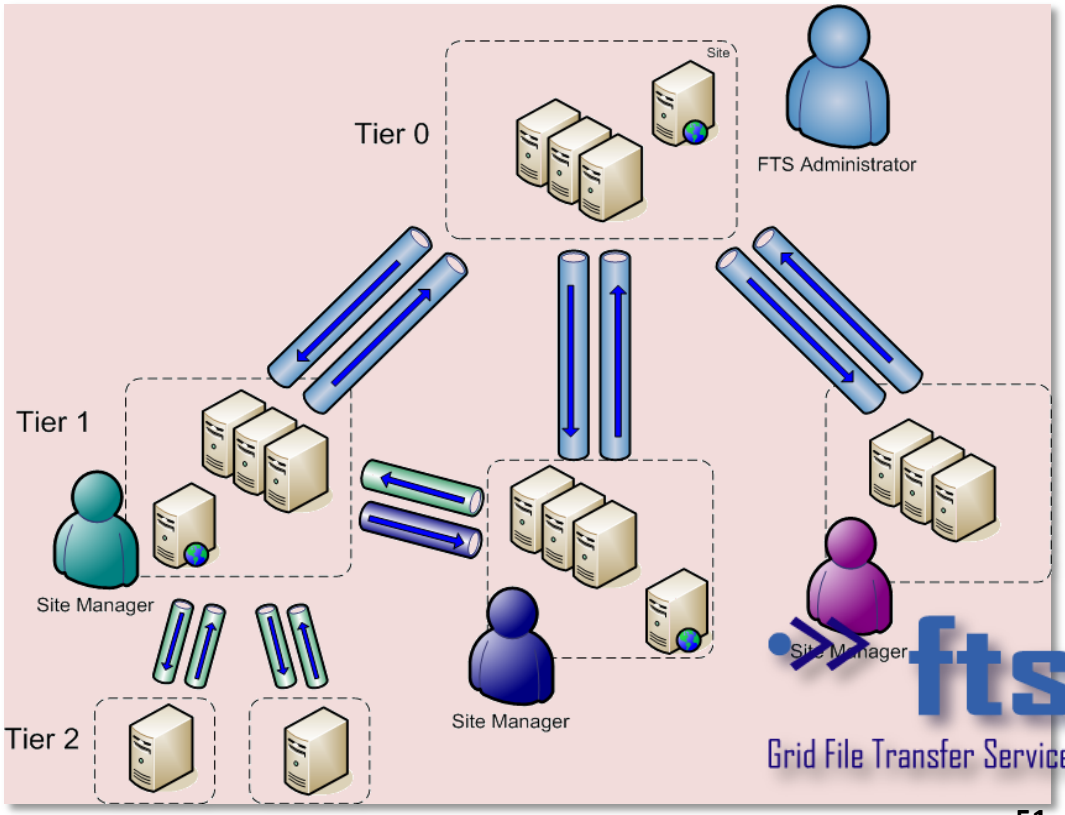
FTS overview

- 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



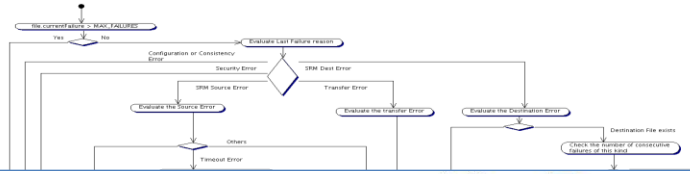
File Transfer Service

- 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 6 months (> 10 million files)



FTS: key points

- Reliability
  - It handles the retries in case storage / network failures
    - VO customizable retry logic
  - Service designed for high-availability deployment
- Security
  - All data is transferred secure SRM / gridFTP
  - Service audits all user / adm
- Service and performance
  - Service stability: it is designe storage and network resourc
  - Service recovery: integration level degradation



FTS Report

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.

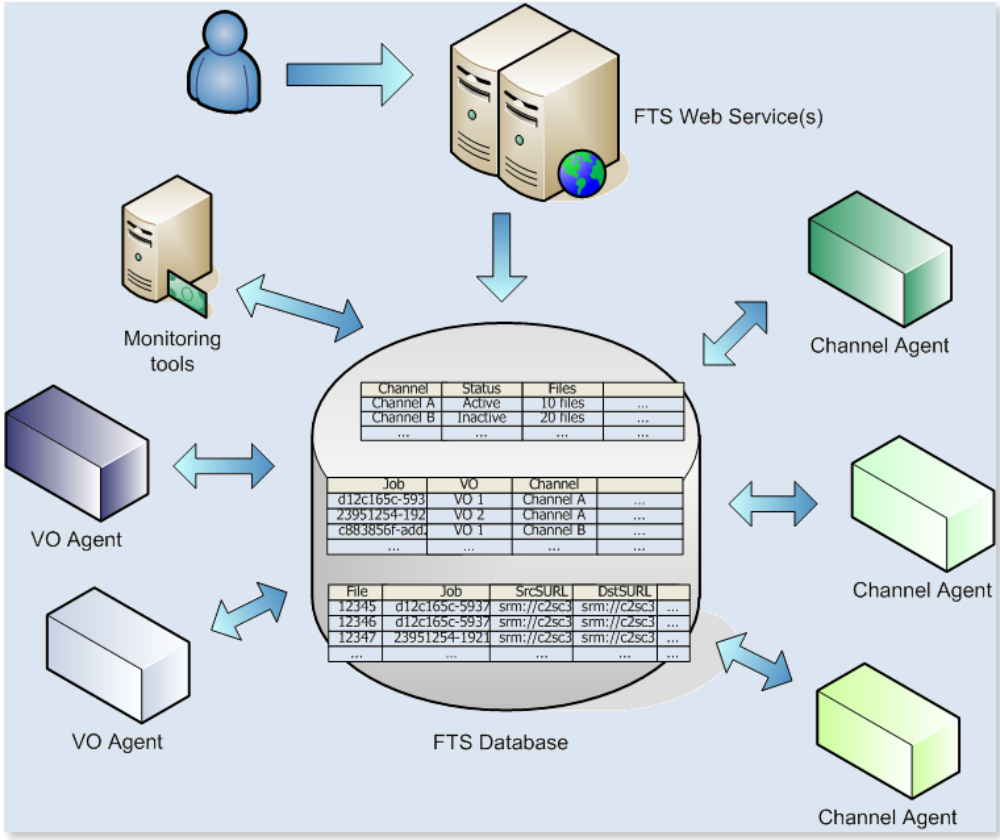
Statistics concerning all the transfers performed yesterday
Between 2006-10-12 08:00:00 +02:00 and 2006-10-13 08:00:00 +02:00

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	263.03	1.62	1700.41	1700.41
	atlas	8932	99.92	7	8925	Dest SRM	57.13	Other	38.08	0	220	0	0	0
	cms	208	0	208	0					2.7	767.55	3.64	561.26	561.26
	dteam	974	0.51	969	5	Other	80	Source SRM	20	0.95	356.31	2.88	923.83	923.83
	lhcb	2145	6.53	2005	140	Source SRM	99.29	Other	0.71	0.11	165.85	0.81	215.32	215.32
	ops	3	0	3	0					0	202.67	0	0	0
CERN-RAL	[All]	8699	59.26	3544	5155	Other	84.91	Source SRM	14.88	0.85	478.22	2.59	3026.81	3027.57
	alice	1155	82.6	201	954	Other	99.58	Dest SRM	0.31	1.86	1805.05	1.11	372.95	372.95
	atlas	4512	88.52	518	3994	Other	84.85	Source SRM	15.15	1.79	1428.94	1.49	926.26	926.57
	cms	227	3.08	220	7	Dest SRM	85.71	Source SRM	14.29	2.53	348.65	10.08	555.61	555.61
	dteam	1077	3.99	1034	43	Other	86.05	Source SRM	9.3	0.95	276.64	4.01	980.47	980.91
	lhcb	1725	9.1	1568	157	Source SRM	99.36	Other	0.64	0.12	146.03	1.16	191.52	191.52
	ops	3	0	3	0					0	27	0.01	0	0
CERN-SARA	[All]	8792	42.55	5051	3741	Dest SRM	83.77	Source SRM	12.22	1.34	108.02	15.4	6777.95	6784.92
	alice	3134	15.12	2660	474	Source SRM	57.17	Dest SRM	41.14	1.66	109.53	18.43	4426.44	4430.29
	atlas	2018	53.32	942	1076	Dest SRM	72.4	Source SRM	16.54	1.15	144.44	9.42	1085.07	1087.6
	dteam	3488	61.32	1349	2139	Dest SRM	98.74	Other	0.98	0.93	81.91	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	395.77	3.21	7514.29	7614.84
CERN-CERN	[All]	1536	39.71	926	610	Source SRM	58.36	Dest SRM	15.9	0.07	287.71	0.38	67.89	69.08
CERN-ASCC	[All]	6851	23.54	5238	1613	Source SRM	50.84	Other	28.89	1.14	1098.6	1.08	5955.81	6080.58
CERN-GRIDKA	[All]	12755	21.38	10028	2727	Source SRM	64.36	Other	32.53	0.87	371.97	3.19	8762.02	8767.53
CERN-TRIUMF	[All]	2244	20.63	1781	463	Other	61.77	Source SRM	31.1	1.04	395.15	3.63	1847.25	1917.13
CERN-BNL	[All]	13975	19.42	11261	2714	Source SRM	69.97	Other	24.24	0.44	190.38	3.41	4951.59	4960.34
CERN-IN2P3	[All]	11697	13.76	10087	1610	Source SRM	48.57	Other	47.45	1.22	296.21	5.33	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



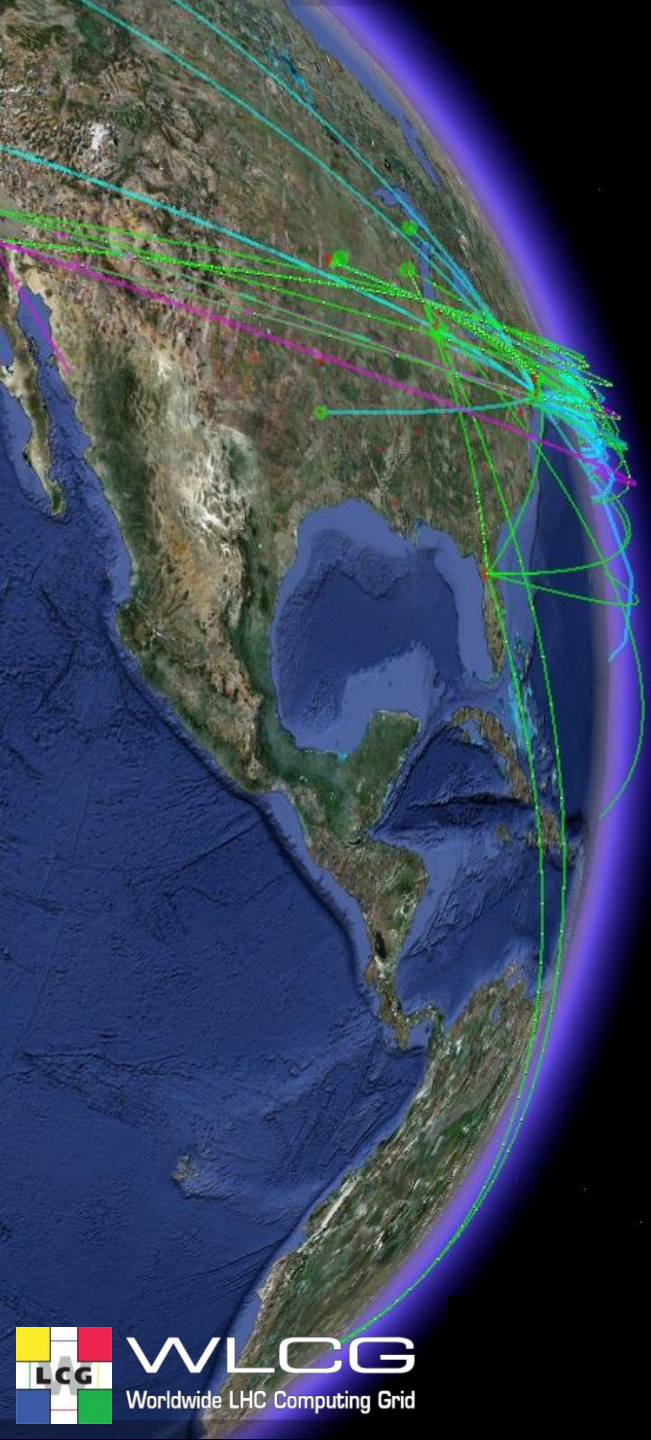
Experiments interact via a web-service

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

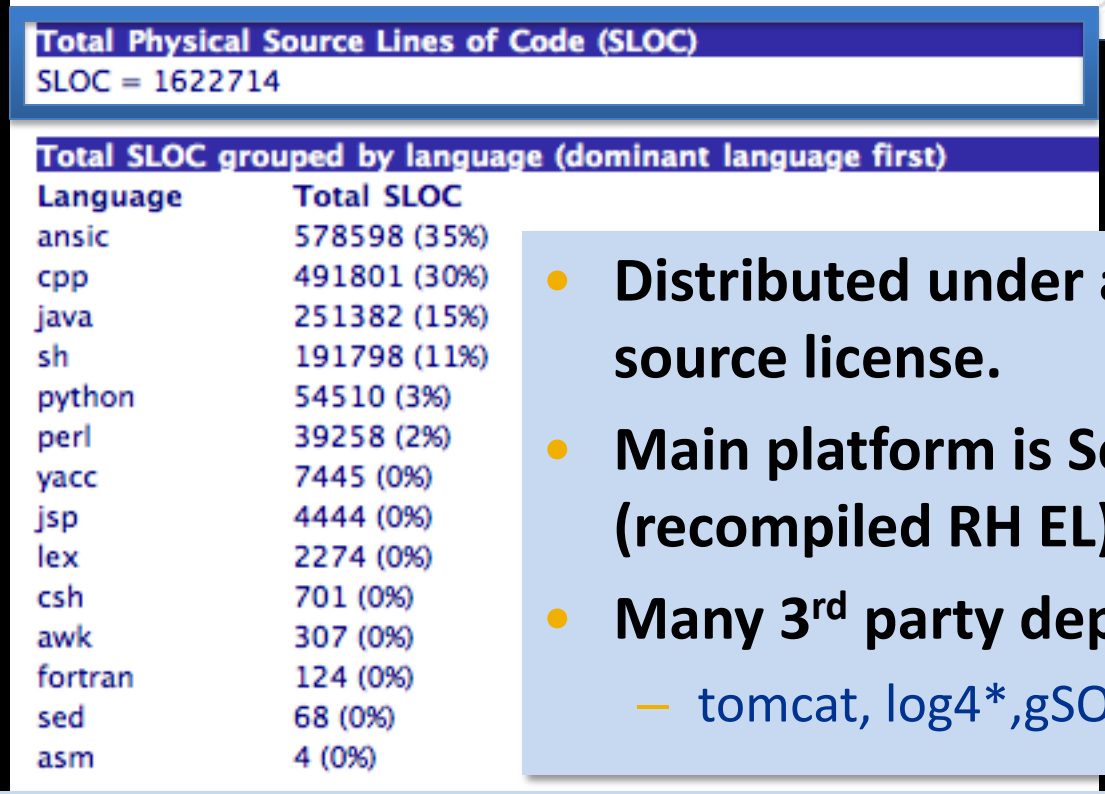
- All components are decoupled from each other
  - Each interacts only with the (Oracle) database
- FTS 3 in development



What does the code look like?

WLCG
Worldwide LHC Computing Grid

gLite code base



Total Physical Source Lines of Code (SLOC)

SLOC = 1622714

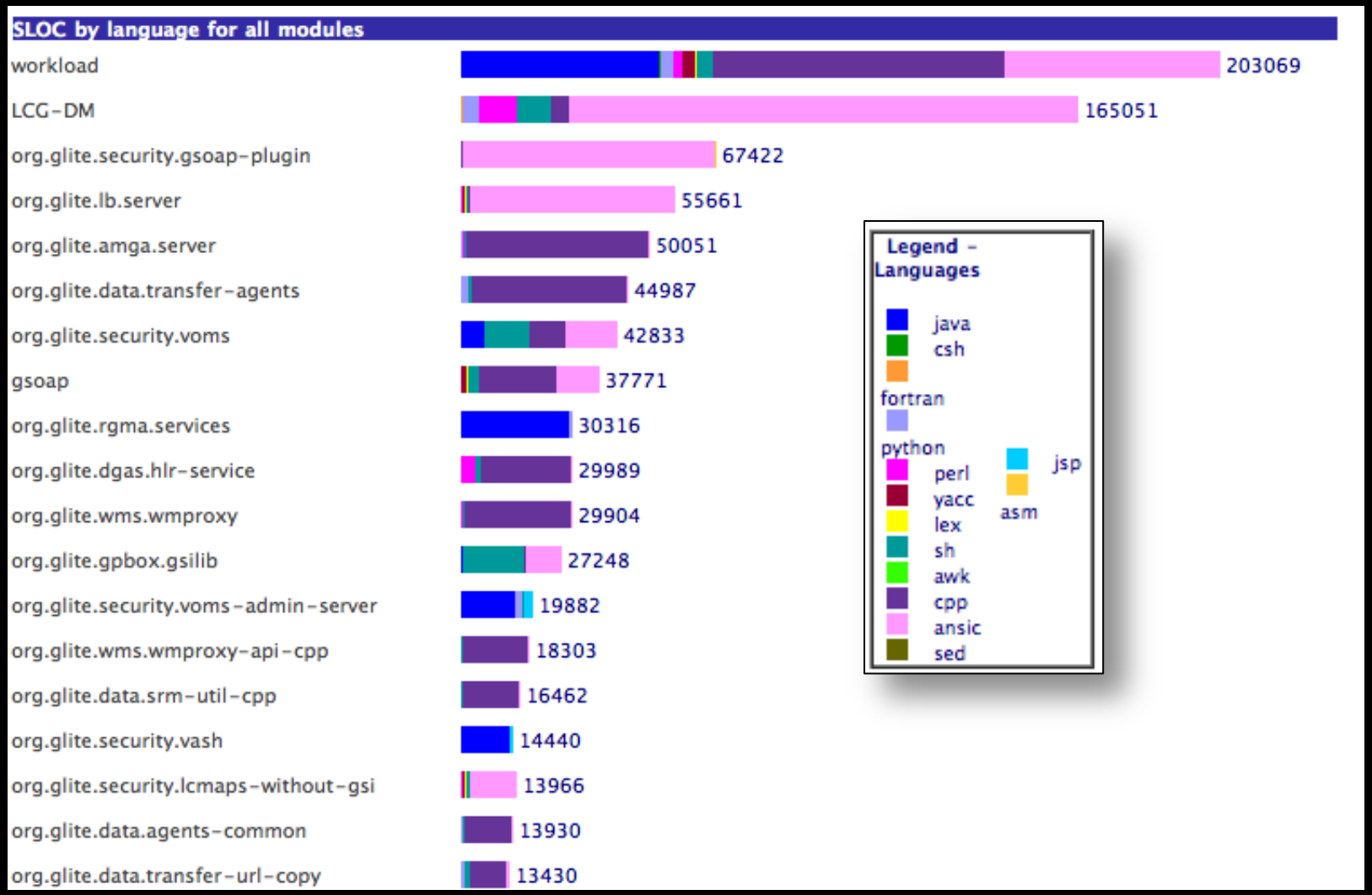
Total SLOC grouped by language (dominant language first)

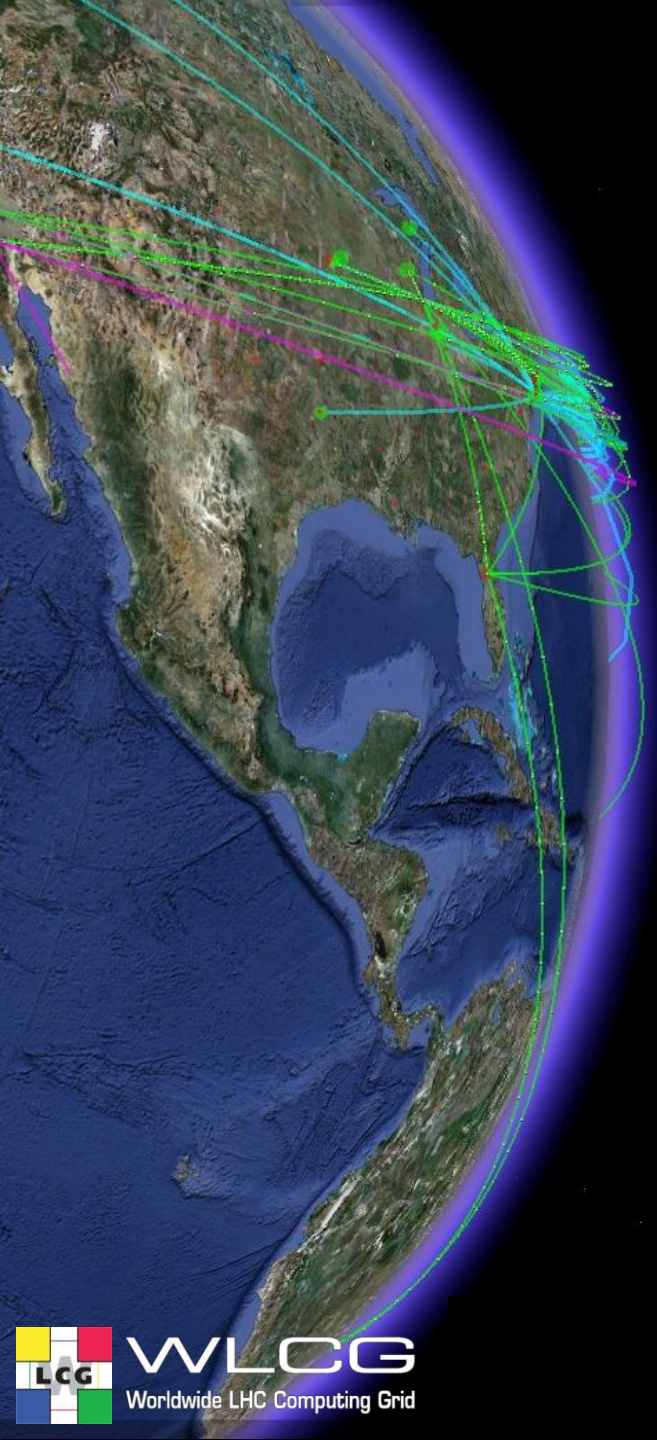
Language	Total SLOC
ansic	578598 (35%)
cpp	491801 (30%)
java	251382 (15%)
sh	191798 (11%)
python	54510 (3%)
perl	39258 (2%)
yacc	7445 (0%)
jsp	4444 (0%)
lex	2274 (0%)
csh	701 (0%)
awk	307 (0%)
fortran	124 (0%)
sed	68 (0%)
asm	4 (0%)

- **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





How do we manage the code?

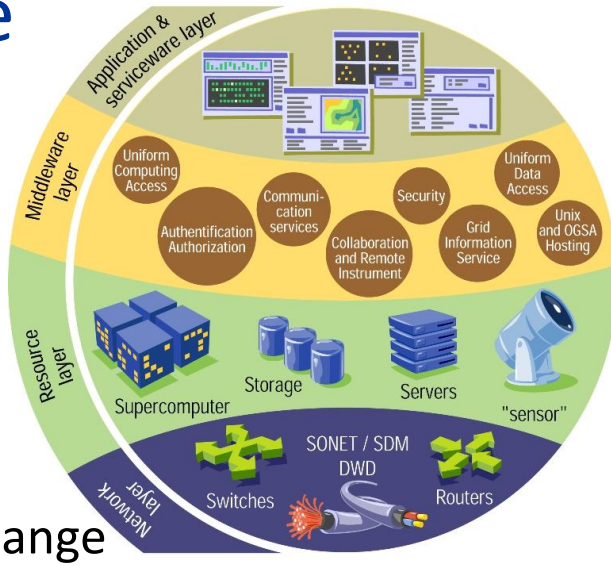
- Builds and test by the development team
- Configuration management by YAIM
  - modular bash shell script
    - >37 000 lines, >30 modules
- Complex certification and release process

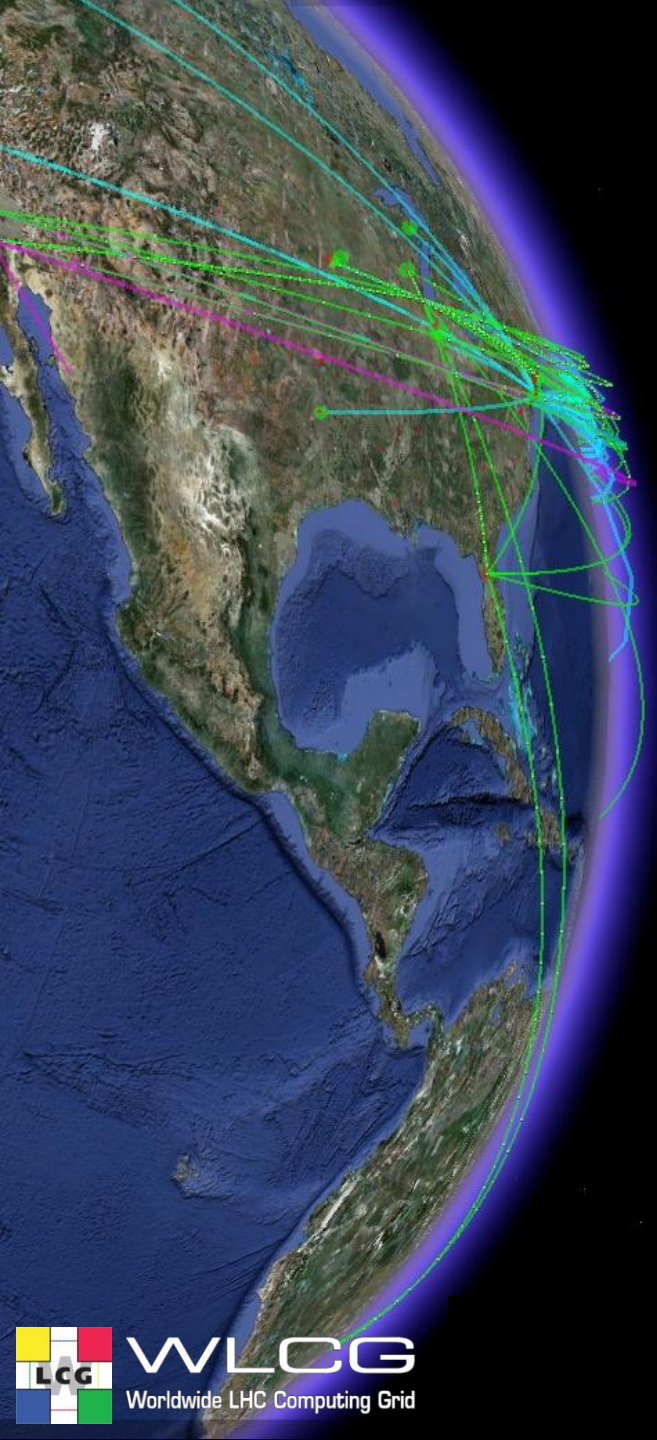
YAIM

WLCG
Worldwide LHC Computing Grid

Why is this complicated

- Heterogeneity is the main issue
- Middleware is in the “middle”
  - has to work with many front and back-ends
  - this results in many “adaptors”
    - which have to change when the “ends” change
  - testing is much more difficult
    - basically a combinatorial problem
    - everything has to work with everything....
- Too much functionality???

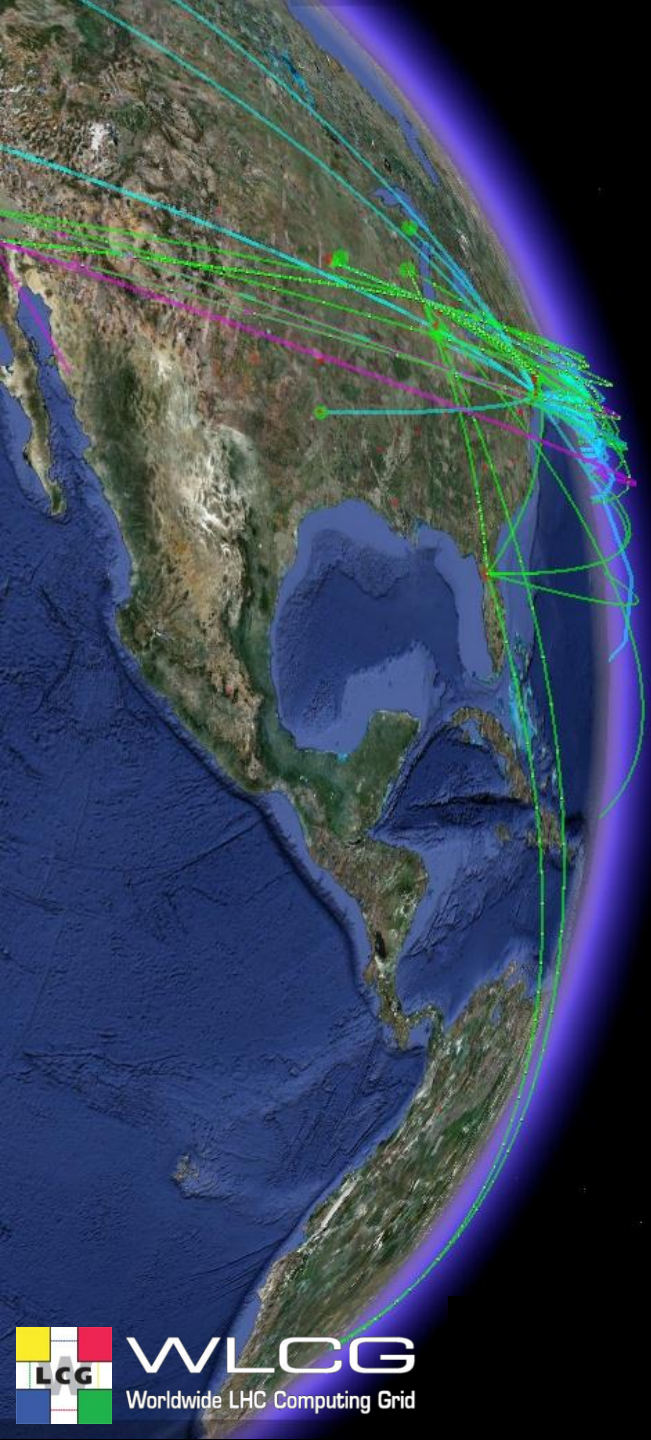




How does the future look like?

- Focus on standardization and interoperation
  - Driving the process
    - OGF etc.
- Focus on stability
- Simplifying the system
- Integrating virtualization
- Integrating Clouds
- End of the EMI project

WLCG
Worldwide LHC Computing Grid



Thanks

- Slides in this presentation have been collected from several sources
  - contributions from many people