Enabling Grids for E-sciencE

# gLite Data Management Components

*Presenter : Markus W. Schulz, CERN, IT-GD*
*On behalf of the Grid Data Management Team*
*CHEP07, 2-7 Sept 2007, Victoria BC Canada*

*jean-philippe.baud@cern.ch*

Lightweight Middleware for Grid Computing

www.eu-egee.org

Information Society and Media

# WLCG & EGEE

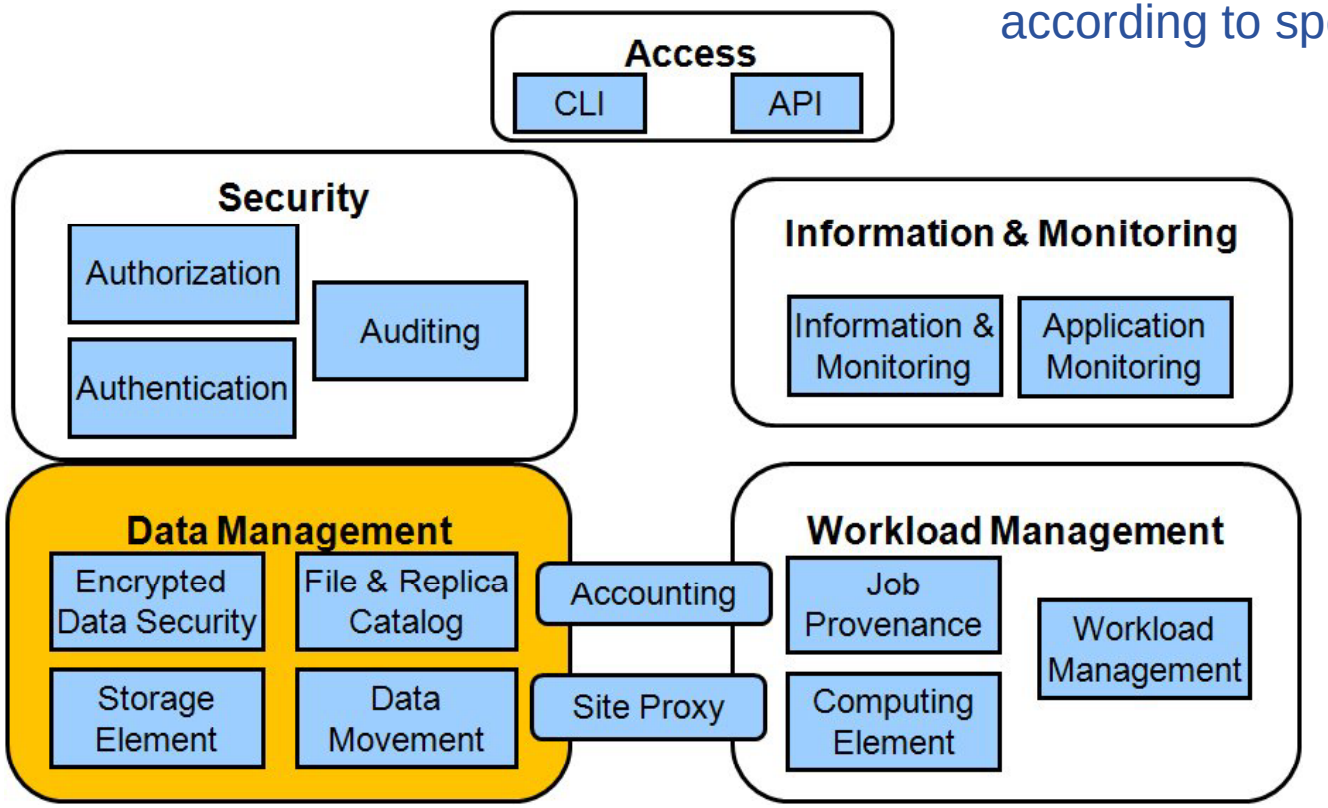
- **Large Hadron Collider (LHC)**
  - 15 PB/year to be stored at rates up to 1.5GB/sec (ALICE) and 100-150 MB/sec (ATLAS, CMS, LHCb)
  - Data sharing : ~500 Institutes, 5000 physicists, computer scientists and engineers

  *Need an infrastructure to support data analysis, sharing and storage*
- **Worldwide LHC Computing Grid (WLCG)**
  - Data processing based on a Tier-Model (Tier-0, Tier-1,Tier-2)
  - Use of Open Science Grid (OSG), EGEE (EGEE), NDGF,+
- ***Will focus on EGEE data management contribution***
- **Enabling Grids for E-SciencE (EGEE)**
  - Grid infrastructure for science (HEP, medicine, astronomy, ...)
  - 240+ sites, 45+ countries
  - Uses gLite as a lightweight, open source middleware distribution

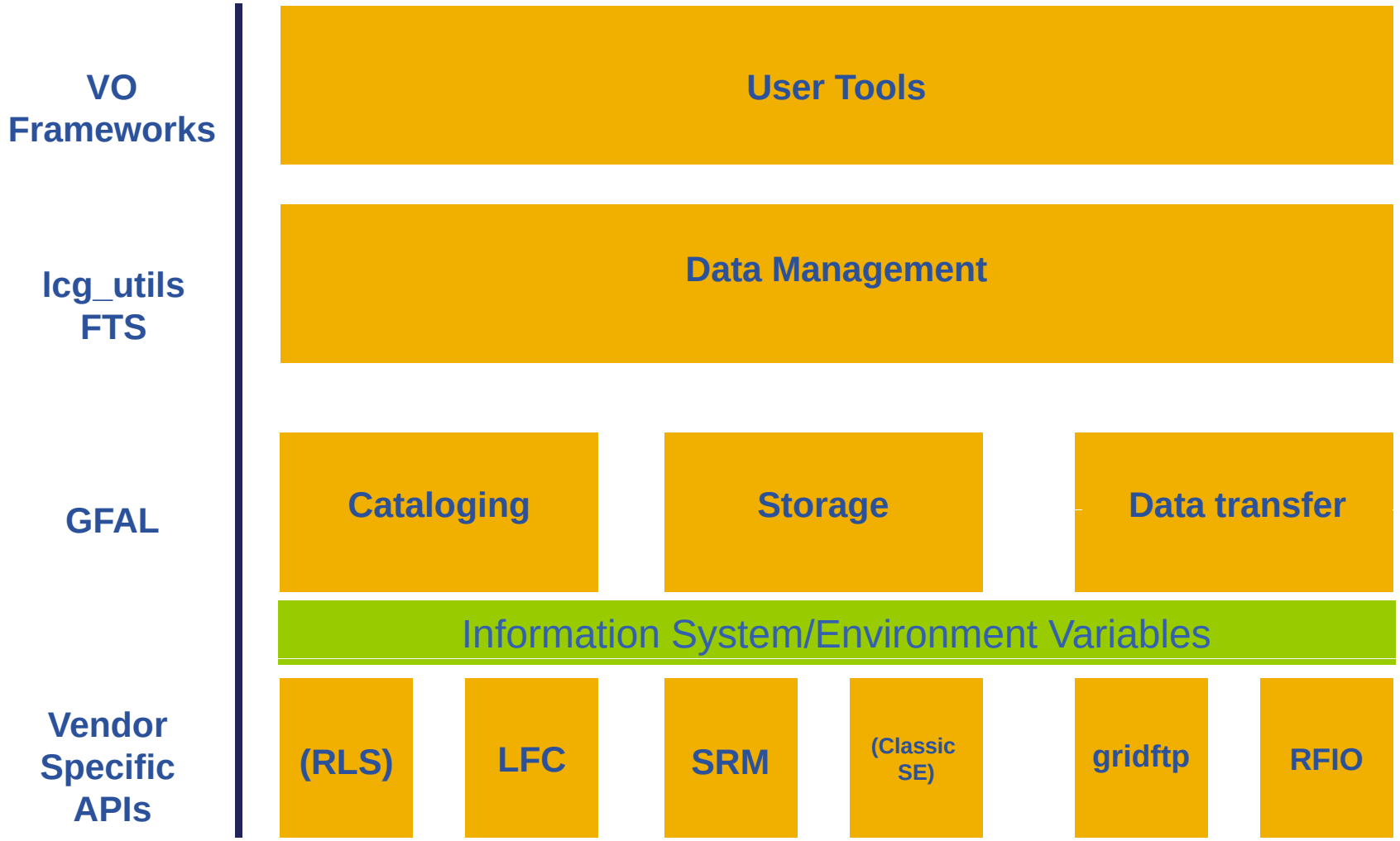
# gLite architecture

## Service Oriented Architecture

- interoperability between grids
- support of grid standards
- flexible exploitation of the grid services according to specific needs



# EGEE Data Management



# LFC

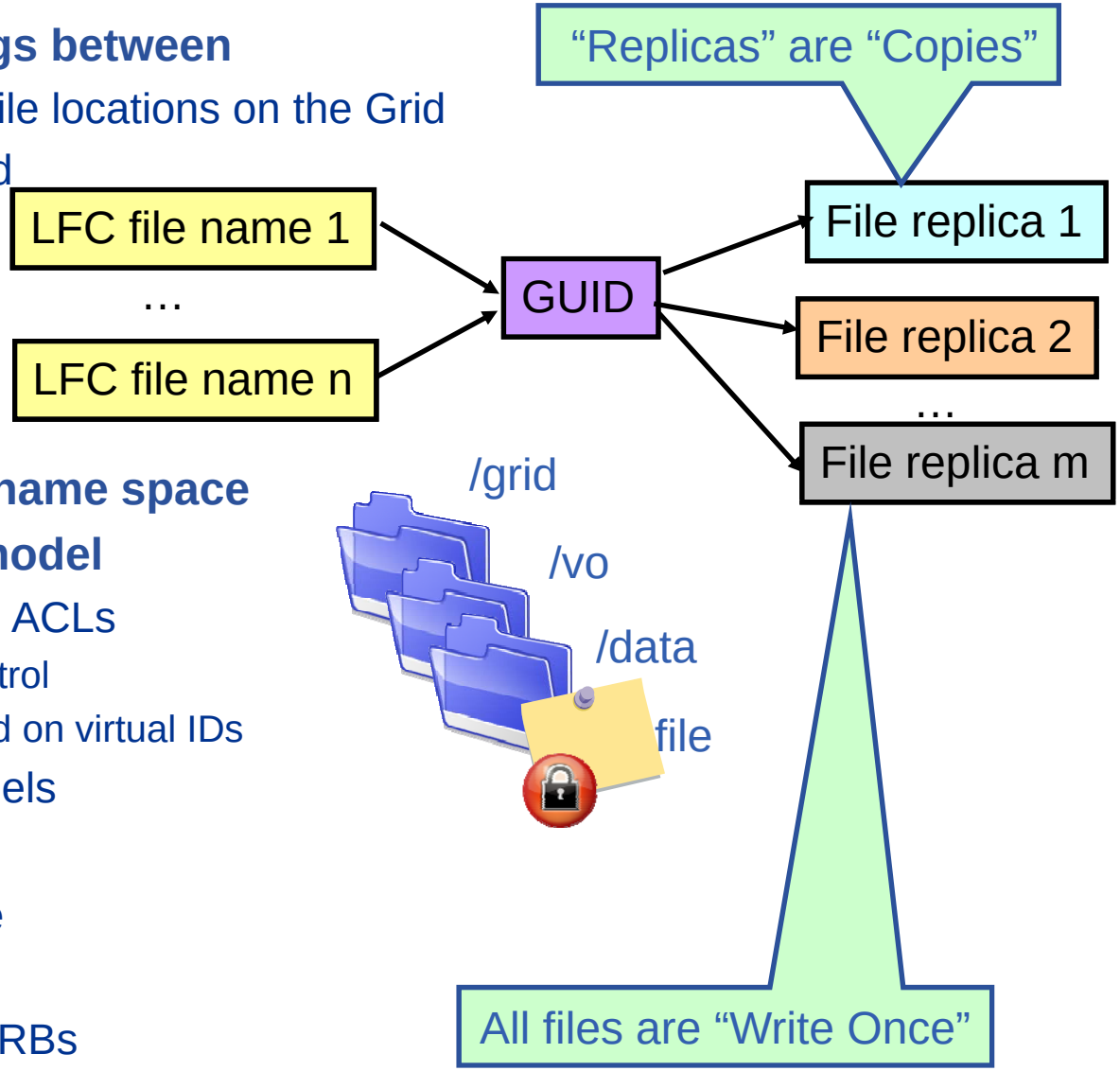
**LCG File Catalog**

**LHC Computing Grid File Catalog**

**Large Hadron Collider Computing Grid File Catalog**

# LCG "File" Catalog

- **The LFC stores mappings between**
  - Users' file names and file locations on the Grid
  - Stores Permissions and
  - Ownership
  - Simple metadata




- **Provides a hierarchical name space**
- **Supports GSI security model**
  - Including VOMS based ACLs
    - Very fine grained control
    - Implementation based on virtual IDs
  - Soon: encrypted channels
- **Simple DLI interface**
  - Data Location Interface
  - GUID <---> Location
  - Integration with WMS&RBs

# LFC

- **MySQL and ORACLE back-ends**
  - Ensures scalability and allows small scale deployment
  - Read only replication of catalogue (awaiting wider deployment)
- **Multi-threaded C server**
  - Supports multiple instances for load balancing
- **Thread-safe C clients**
  - Python & Perl bindings
  - Command line interface
- **Supports sessions to avoid authentication costs**
  - GSI is very expensive!
- **Bulk methods to reduce the number of round trips**
  - Under test by ATLAS --> 20 times faster
- **Widely used in EGEE:**
  - **largest LFC instance contains 8 millions entries**

*See poster session 2 "LFC status and recent developments" for more details*

# DPM

# Disk Pool Manager

# SRM

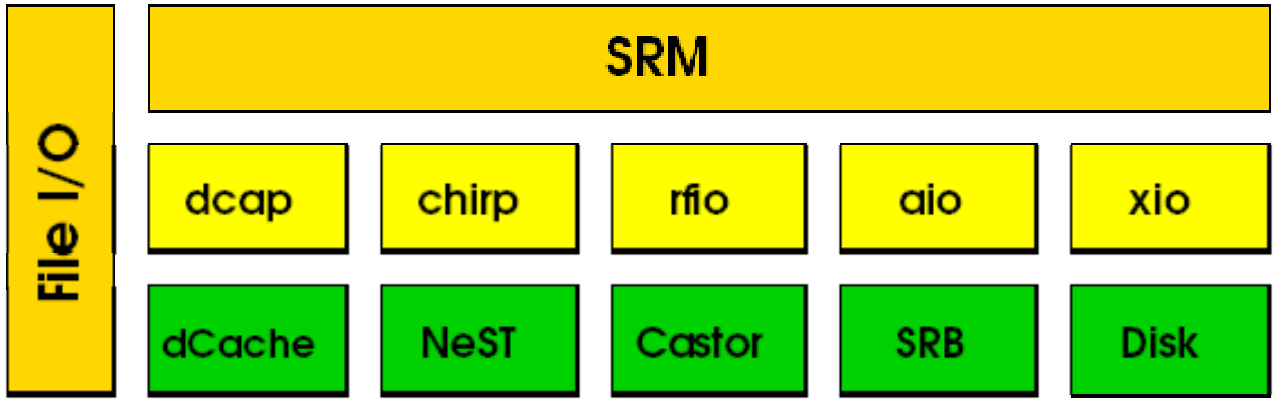
# Storage Resource Manager

# Storage Element

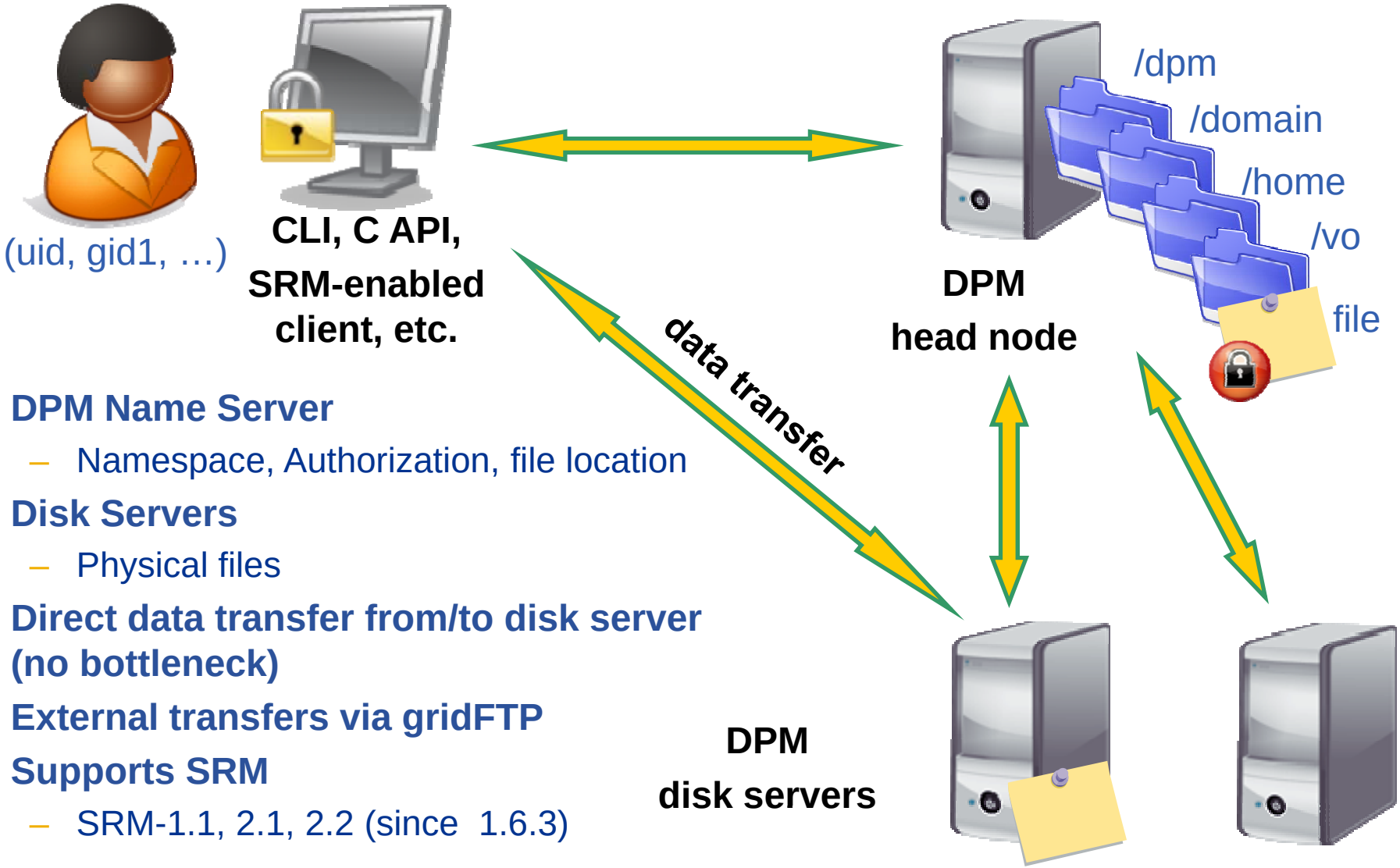
- **Storage Resource Manager (SRM)**
  - Standard that hides the storage system implementation (disk or active tape)
  - handles authorization
  - Web service based on gsoap
  - translates SURLs (Storage URL) to TURLs (Transfer URLs)
  - disk-based: DPM, dCache,Storm; tape-based: Castor, dCache
  - SRM-2.2
    - Space tokens (manage space by VO/USER), advanced authorization,
    - Better handling of directories, lifetime, +++++++

*See talk "SRM 2.2 definition and testing experience" for more details (today)*

- ***File I/O*: *posix-like* access from local nodes or the grid**
  - ➔ GFAL (Grid File Access Layer)



# DPM: user's point of view




- **DPM Name Server**
  - Namespace, Authorization, file location
- **Disk Servers**
  - Physical files
- **Direct data transfer from/to disk server (no bottleneck)**
- **External transfers via gridFTP**
- **Supports SRM**
  - SRM-1.1, 2.1, 2.2 (since 1.6.3)

# DPM

## Addresses the storage needs of Tier-2 and smaller sites

- Focus on easy setup and maintenance
- Multi-threaded C implementation
- Name server DB
  - Keeps track of the status of files and their physical locations
  - MySQL and ORACLE back ends
    - Simplifies integration in existing local DB infrastructure
    - Ensures scalability
  - Shares code with LFC --> fix once run twice!
- Thread-safe C client and command line interface
  - http/https DPM browser ( implemented, very soon to be released)
  - users and site managers interact with DPM at different levels
- GSI and VOMS based authorization and fine grained ACLs
  - Implemented via virtual IDs -> no excessive use of pool accounts
  - Pool access control on VO basis

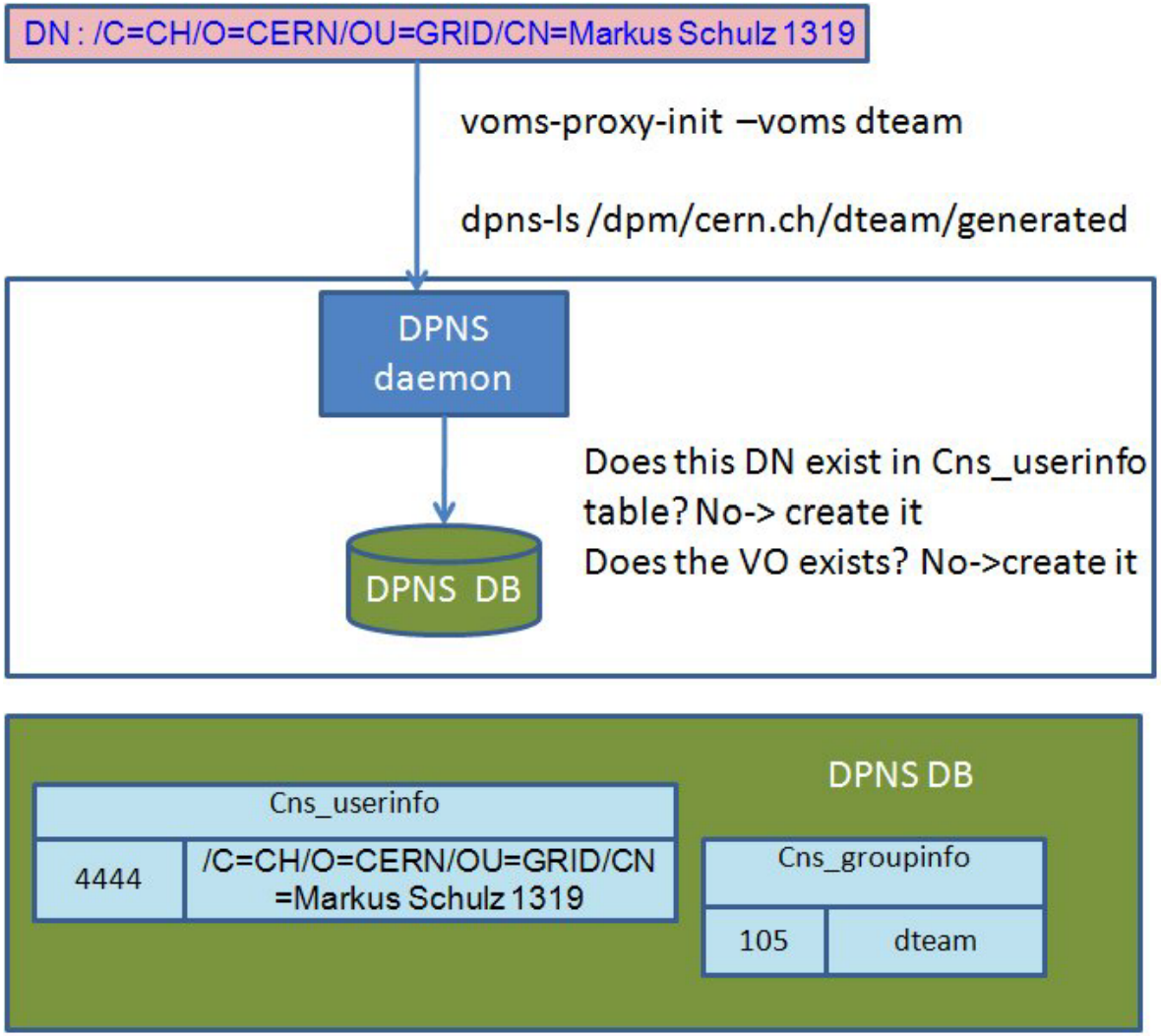
# DPM

## Different file access and transfer protocols

- Secure Remote File Input/Output (RFIO)
  - Secure file transfer and manipulation.
  - Implementation of thread-safe C client and a command line interface
  - Support of streaming mode.
- GSIFTP allows remote file transfer
  - New gridftp plugin is implemented to support gridFTP-2
- Xrootd: useable but still limited
  - no support of grid/voms certificates yet
- https/http: web access based on Apache.
  - Protocol http or https can be specified at transfer time

*See poster session 2 “DPM Status and Next Steps” for more details*

# Virtual ID concept

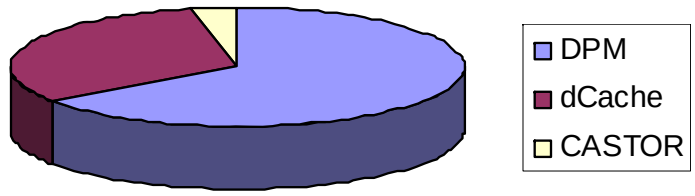



- No need to create pool accounts
- No need to change the /etc/passwd file
- Faster check of ACL than with pattern matching on DN/FQAN

# LFC & DPM deployment status

- **EGEE Catalog**
  - 110 LFCs in production
    - 37 central LFCs
    - 73 local LFCs

- **EGEE Storage Elements**
  - CASTOR
  - dCache
  - DPM
    - 96 DPMs in production
    - Supporting 135 VOs




**Storage Element instances published in EGEE's Top BDII**

- **LFC and DPM**
  - Stable and reliable production quality services
  - Well established services
  - Require low support effort from administrators and developers

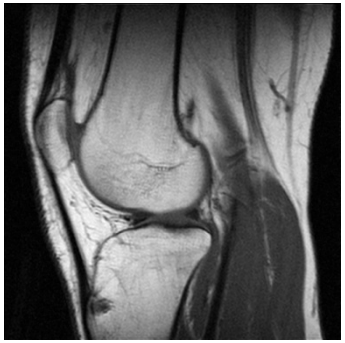
# Medical Data Management

**Problem : Medical institutes request data storage encryption**

- Use of the DICOM standard for medical image handling
- Image retrieval and storage from/in DICOM servers : security issues

**Solution : Extension of the data management tools (under way)**

- File encryption on the fly, local decryption
- Use of HYDRA for split key management
- Use of the LFC to register/retrieve system data
  - Replicas location, filesize, …
- Use of srmv2 to get the turls
- Use of I/O protocols, gridftp to load medical images
- Access control based on VOMS



*See poster session 2 “Medical Data Management” for more details*

# Lcg-util & GFAL

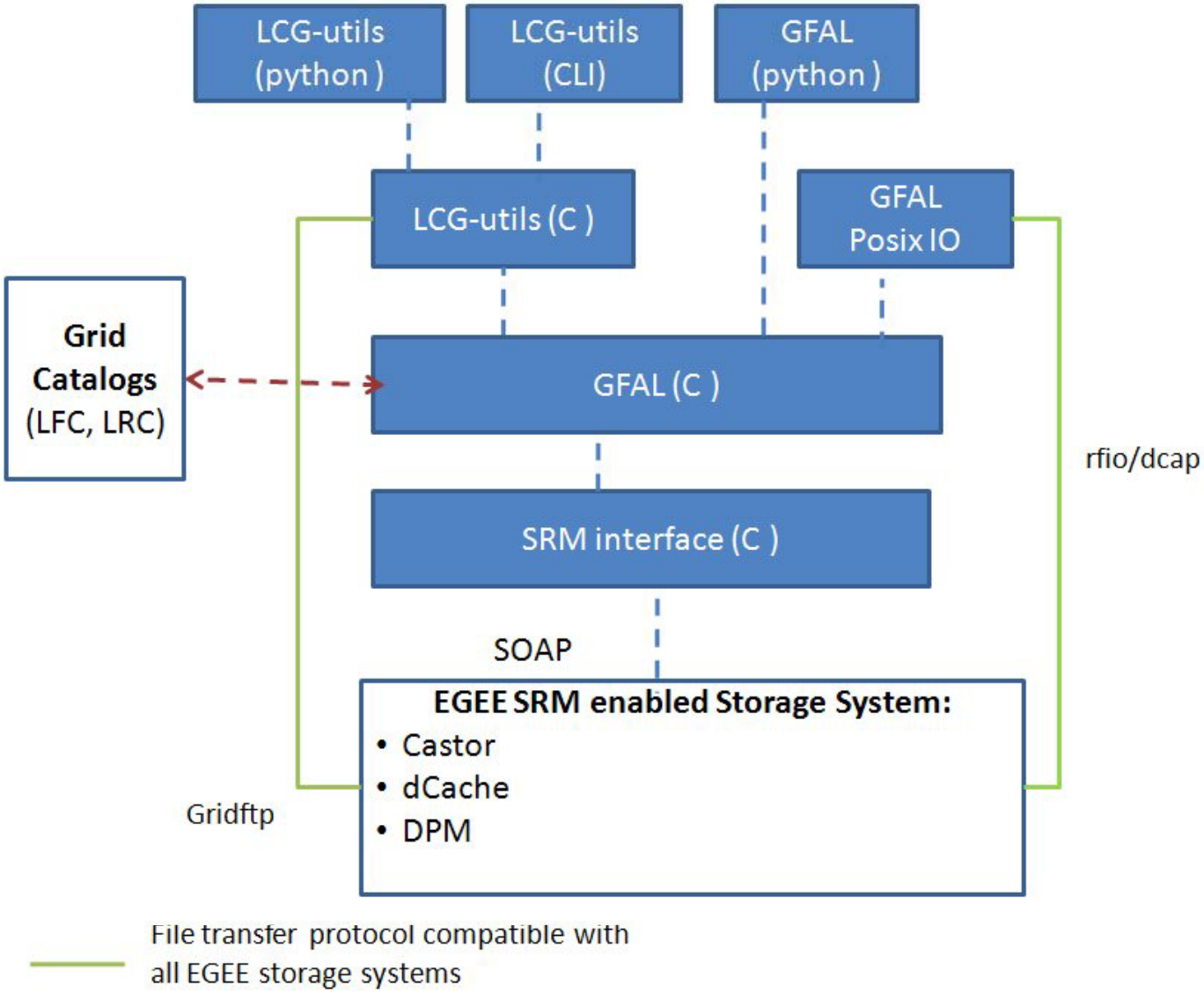
## Grid File Access Layer

# Lcg-util/gfal features

- **Purpose: Create the illusion of POSIX I/O**
  - Shield users from complexity
  - Interact with the information system, catalogue, SRMs
    - Can be used with/without information system/ catalogue
- **LCG-util :**
  - Command line and C-Api
  - Covers most common use cases
    - Replication, catalogue interaction etc,
  - high level tool box
- **Gfal:**
  - Posix like C API for file access
  - SRMv2.2 support
    - user space tokens for retention policy (custodial/replica) & access latency (online/nearline)

*See Poster session 2 “lcg-util and gfal” for more details*

# Lcg-util/gfal architecture



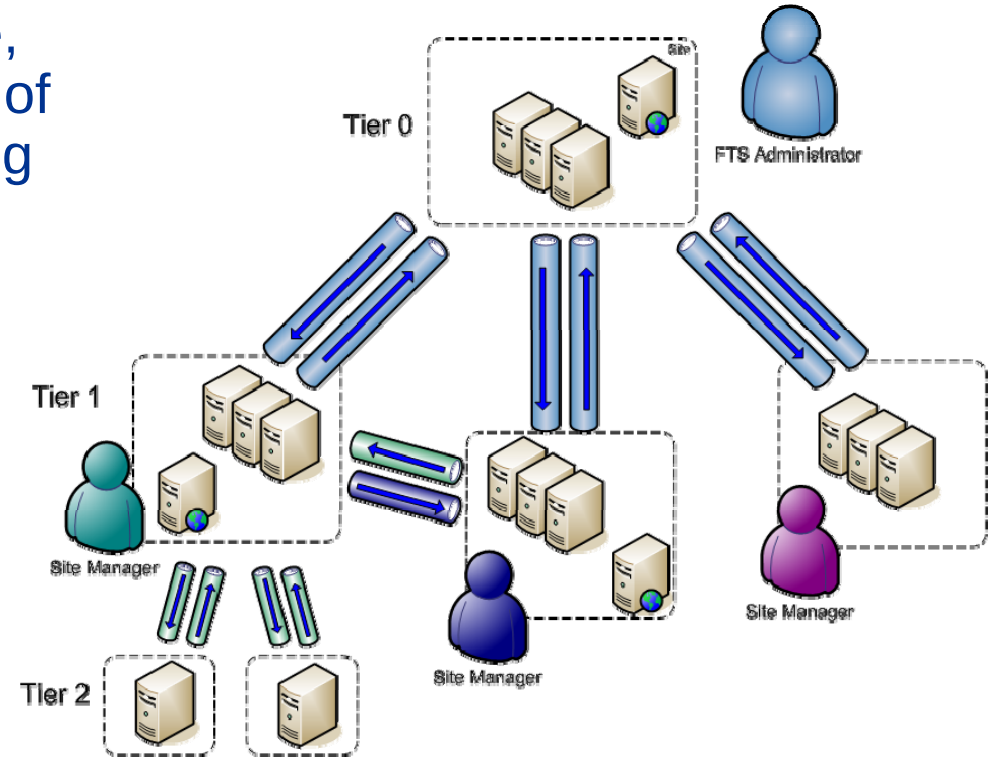
# FTS

## File Transfer Service

Grid File Transfer Service

# FTS overview

- **gLite File Transfer Service is a reliable data movement service (batch for file transfers)**
  - FTS performs bulk file transfers between multiple 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

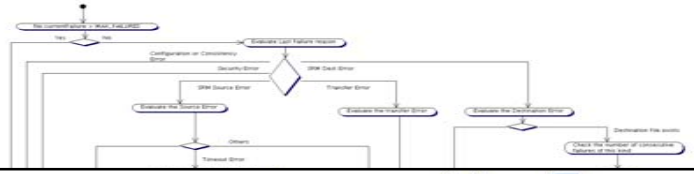



Grid File Transfer Service

- **Why is it needed ?**
  - For the user, the service it provides is the reliable point to point movement of Storage URLs (SURLs) and ensures you get your share of the sites' resources
  - For the site manager, it provides a reliable and manageable way of serving file movement requests from their VOs and an easy way to discover problems with the overall service delivered to the users
  - For the VO production manager, it provides ability to control requests coming from his users
    - Re-ordering, prioritization,…
  - The focus is on the "**service**" delivered to the user
    - It makes it easy to do these things well with minimal manpower

# FTS: key points

- **Reliability**
  - It handles the retries in case of storage / network failure
    - VO customizable retry
  - Service designed for high-availability deploym
- **Security**
  - All data is transferred se SRM / gridFTP
  - Service audits all user /
- **Service and performan**
  - Service stability: it is des storage and network res
  - Service recovery: integra 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

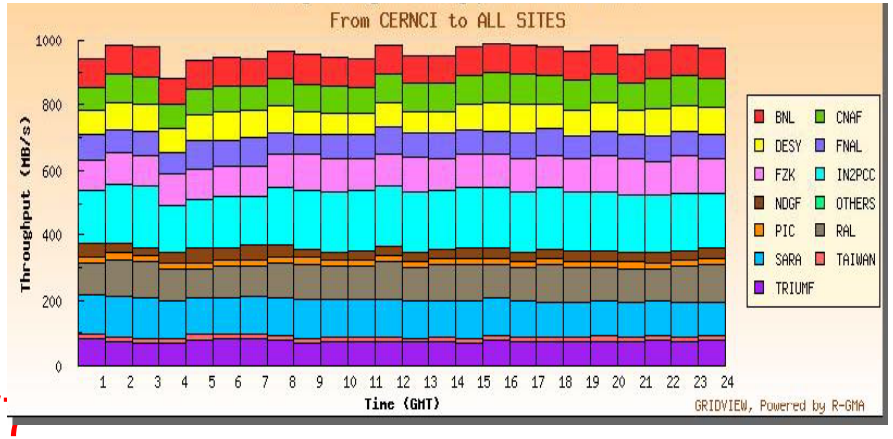
CERN-* Filter Show VO details

| 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

# Service scale

- **Designed to scale up to the transfer needs of very data intensive applications**
- **Currently deployed in production at CERN**
  - Running the production WLCG tier-0 data export
  - Target rate is ~1 Gbyte/sec 24/7
  - Over 9 petabytes transferred in last 6 months > 10 million files




- **Also deployed at ~10 tier-1 sites running a mesh of transfers across WLCG**
  - Inter-tier1 and tier-1 to tier-2 transfers
  - Each tier-1 has transferred around 0.2 – 0.5 petabytes of data

- **New features in FTS 2.0**
  - Better security model (certificate delegation)
  - Support for SRM v2
  - More administrative tools, more advanced monitoring features to make it easier to operate the overall service
  - Soon:
    - Better support for clouds and channel sets
    - Black/White-listing SEs
    - Better integration with VO workflow management
      - *Call backs, hooks in the state machine*
      -
- **Focus continues upon service monitoring and easing the service operations together with closer integration of FTS with experiment software frameworks**

*see talk "Building the WLCG file transfer service" for more details*

# Is this all?

**Definitely NO!!!**

- The AMGA meta data catalogue ( by Birger Koblitz)
  - Widely used by experiments
  - Is in the process to be integrated in the gLite distribution
- Many data management tools and services developed by Vos

- Lessons learned
  - A DM stack can only be developed with production feedback
  - The right balance between exposing details and hiding is hard to find
  - There will be more to do

# Summary

**Current status**

- Data Management framework is usable
- LFC, FTS, DPM and lcg-util/gfal are used in production on a large scale

**Outlook & Future**

- ACL synchronization between LFC and SEs
- Improvements to lcg-util/gfal
  - e.g. flexibility to work independently of the LFC
- Better tools to check consistency in DPM
- Extension of Xrootd to support grid/voms certificates
- Finish medical data management implementation
- DPM : quota on pools and accounting
- Operational improvements to the FTS

- Continue the dialog with the user communities to focus effort

# LFC bulk methods

```
//deleting replicas (if force=1) and LFNs if mapped to one of guid in the list
   of guids
```

- `int DLL_DECL lfc_delfilesbyguid(int nbguids, const char **guids, int force, int *nbstatuses, int **statuses)`

```
//deleting replicas (if force=1) and LFNs (specified by the **paths)
```

- `int DLL_DECL lfc_delfilesbyname(int nbfiles, const char **paths, int force, int *nbstatuses, int **statuses)`

```
//deleting replicas (if force=1) and LFNs if LFN starts with
   concat(path,pattern)
```

- `int DLL_DECL lfc_delfilesbypattern(const char *path, const char *pattern, int force, int *nbstatuses, struct Cns_filestatus **statuses)`

```
//deleting replicas stored on the given SE and associated with
one guid provided by the list of guids
```

- `int DLL_DECL lfc_delreplicas(int nbguids, const char **guids, char *se, int *nbstatuses, int **statuses)`

```
//list the content of the directory + replica info if matches the
pattern
```

- `struct lfc_direnrep DLL_DECL * lfc_readdirxp(Cns_DIR *dirp, char *pattern, char *se)`