EMI data management description & status

IT-GT-DMS

Michail Salichos
Martin Hellmich

Overview

- background
- technical details
- current status
- summary
- references

background

- The GT (Grid Technology) group is responsible for maintaining and evolving CERN's grid software components and grid monitoring infrastructure
- IT-GT-DMS section develops, maintains and supports the EMI data management services, such as DPM/LFC, FTS, gfal/lcg-util
- The File Transfer Service (**FTS**) is used to distribute the LHC data between participating sites for storage and analysis
  - FTS is the service responsible for distributing the majority of LHC data across the WLCG infrastructure
- **LCG Utils** is a suite of client tools for data movement written for the LHC Computing Grid
- Grid File Access Library (**GFAL**). GFAL provides a POSIX-like interface for I/O operations on Grid files, effectively hiding the interactions with the LFC, the SEs and SRM

FTS2

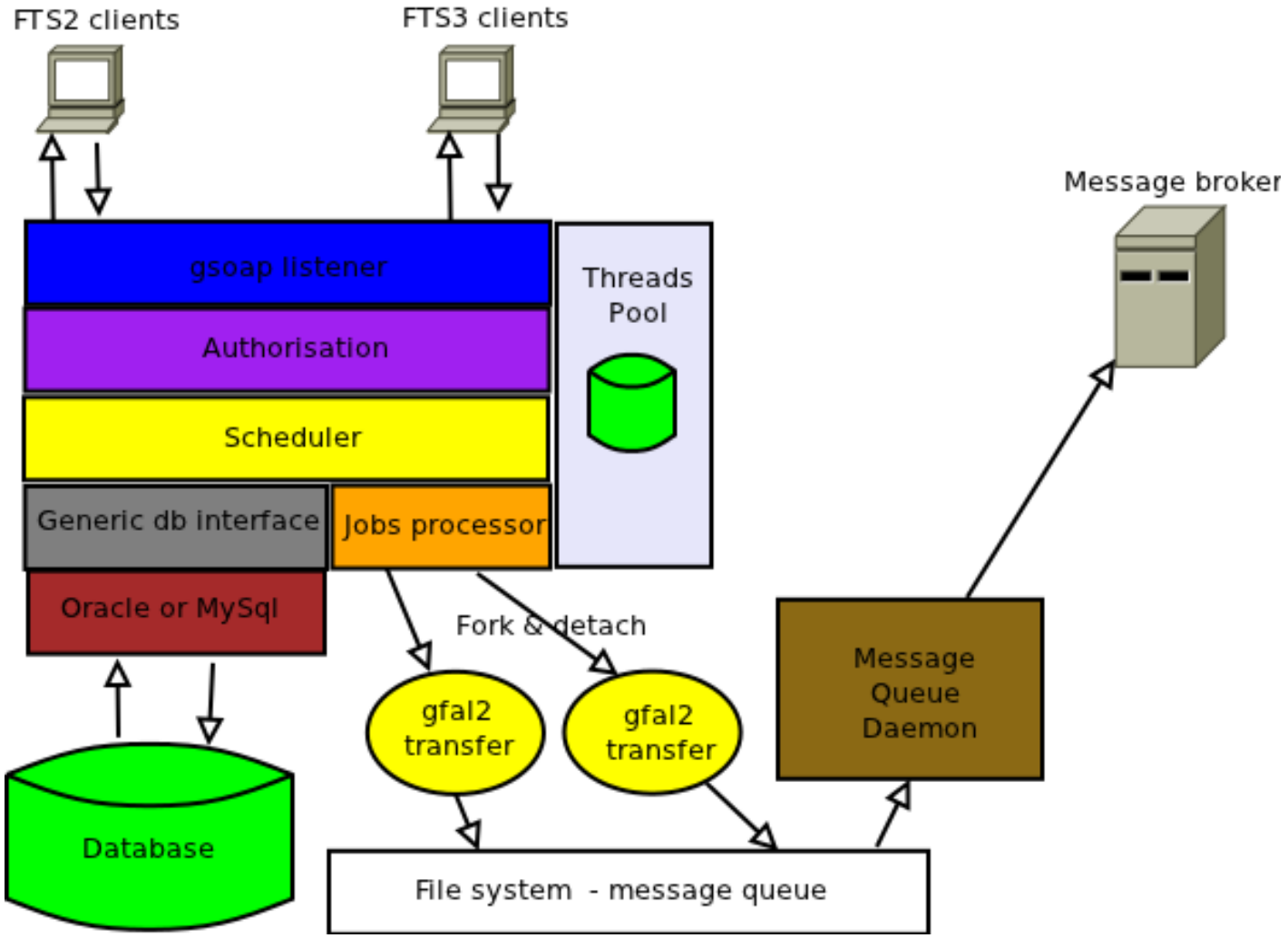
- Currently in production (FTS 2.2.8)
- T-party transfers
- Static channel model configuration between sites
- Administration and monitoring of transfers
- Oracle database back-end
- SRM & GSIFTP protocol support
- Example commands:
  - glite-transfer-submit, submits a transfer job
  - glite-transfer-status, displays the status of an ongoing transfer job
  - glite-transfer-list, lists all submitted transfer jobs owned by the user
  - glite-transfer-cancel, cancels a transfer job

https://svnweb.cern.ch/trac/glitefts/wiki/FTSUserGuide

FTS3

- Simplified configuration and administration of the service
  - Zero configuration supported
- Auto-tuning transfers (experimental)
- Functionally comparable to FTS2
- Channel-less / endpoint-centric
- Generic db interface – multiple back-ends support (oracle and MySql for now)
- Generic transfer protocol interface on top of gfal2
- Improved fair-share and transfer optimization
- GridFTP session reuse (FTS3 clients only)
- Web-based monitoring and configuration
- REST-full interface for transfer submission and status request ???

FTS3 architecture



lcg-util

- Built on top of GFAL
- copy files between CE (Compute Element), WN (Worker Node) and a SE (Storage Element)
- register entries in the file catalog (LFC)
- replicate files between Ses
- Example commands
  - lcg-cp, copies a grid file to a specific location
  - lcg-cr, copy and register a file
  - lcg-del, deletes a file

https://svnweb.cern.ch/trac/lcgutil/wiki/lcg_util

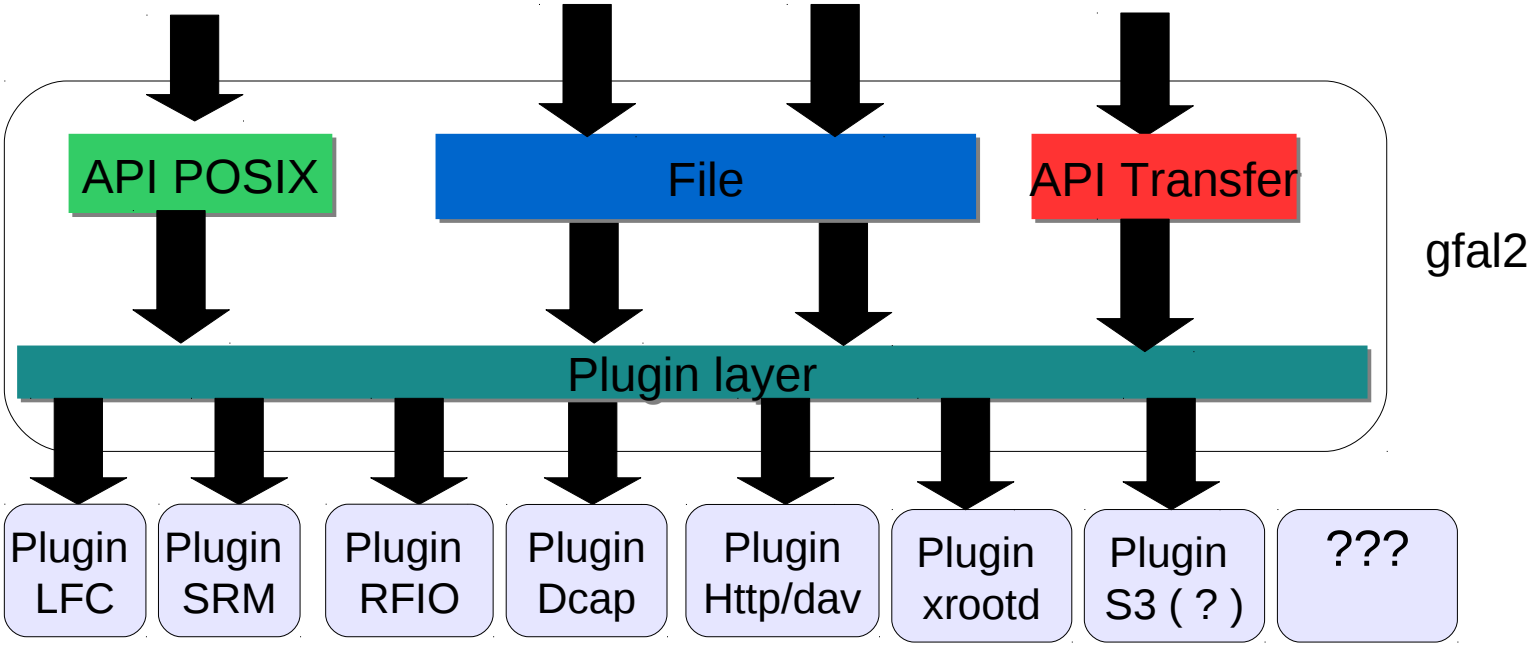
GFAL2

- One library for all Grid and Cloud data access and management
- One library for all data access and transfer protocols
  - Supported protocols
    - GridFTP, DCAP/GSIDCAP, RFIO/ RFIO secured
    - LFN, SRM, HTTP/ Webdav, (S3 ?)
    - XROOTD (developed by NorduGrid)
    - plug-ins mechanism
- Fully thread-safe

GFAL2 current status

- Already released in EMI2 and EPEL
  - **yum install gfal2-all gfal2-doc gfalFS**
- Will be packaged for Debian support
- continuous development, debugging and keep adding new features

GFAL2 architecture



Any Cloud with Http/Dav

Any xrootD point

Extensible to any storage system : Amazon S3, HDFS, etc...

GFAl2 can do

- **Meta-data operations**
  - stat, rm, mkdir, mv, rmdir, etc...
  - listing directory, xattr, etc
- **Remote I/O in any protocol**
  - open/read/write/lseek/close
  - GET/PUT
  - pread/pwrite
- **High level file transfer in any protocol**
  - session-reuse, spacetoken, parallel streams, ...

FTS / lcg-util /gfal roadmap

- Released in EMI (1,2,3) and EPEL (lcg-util / gfal for now, FTS3 in the near future)
- Continuously supported and improved
  - https://svnweb.cern.ch/trac/fts3/roadmap
  - https://svnweb.cern.ch/trac/lcgutil/roadmap

References

- FTS2
  - https://svnweb.cern.ch/trac/glitefts/wiki
- FTS3
  - https://svnweb.cern.ch/trac/fts3/wiki
- GFAL / lcg-util
  - https://svnweb.cern.ch/trac/lcgutil/wiki
- Source code
  - https://svnweb.cern.ch/trac/fts3
  - https://svnweb.cern.ch/trac/lcgutil

IT-GT-DMS

DPM/LFC

Martin Hellmich, CERN

Amsterdam, 27.11.2012

Overview

- SRM
- Storage elements
  - DPM
- Catalog
- Authentication
- Standards support
- Federation

SRM

	SRM function[2]
Transfer Management	
Upload / download a complete file	srmPrepareToPut/GetWithGetDone
Manage transfers.	srmAbort/Suspend/Resume Request
Balance over multiple transfer servers.	srmPrepareToGet[c]
Manage third-party copy	
Negotiating a transport protocol	srmGetTransferProtocols
Namespace Interaction	
Querying information about a file (stat)	srmLs
Upload data integrity information (chksums)	
Check integrity information	srmLs
Creating/Deleting data and directories	srmMkdir srmRmdir srmRm srmMv
Changing ownership, perms and ACLs	srmSet/Check/GetPermission
Storage Capacity Management	
Query used capacity (like df)	srmGetSpaceMetaData/Tokens
Create/remove reservations; assign characteristics	srmReserve/Update/Release Space
Targeting uploads to specific reservation	srmPrepareToPut
Moving files between reservations	srmChangeSpaceForFiles
Server Identification	
Test service availability and information	srmPing

+File Locality Management srmBringOnline

Storage Elements

- StoRM
  - frontend for cluster file systems
- DCache
  - supports tape

10 PByte

- DPM/LFC
  - disk pools
  - focus on easy operation

2 PByte

All support SRM

EMI INFSO-RI-261611

DPM

- Multiple protocols
  - HTTP, NFS4.1/pNFS (readonly), XROOT, GridFTP
  - HTTP frontend is Apache
- Disk pool backend
- Extensibility through plugins

DPM upcoming

- Multiple protocols
  - NFS4.1/pNFS read-write, faster GridFTP
- standardized backends
  - S3 pools (Amazon, OpenStack, Huawei)
  - HDFS pools
  - POSIX FSs (GPFS, ...)
- Plugins in Python

EMI INFSO-RI-261611

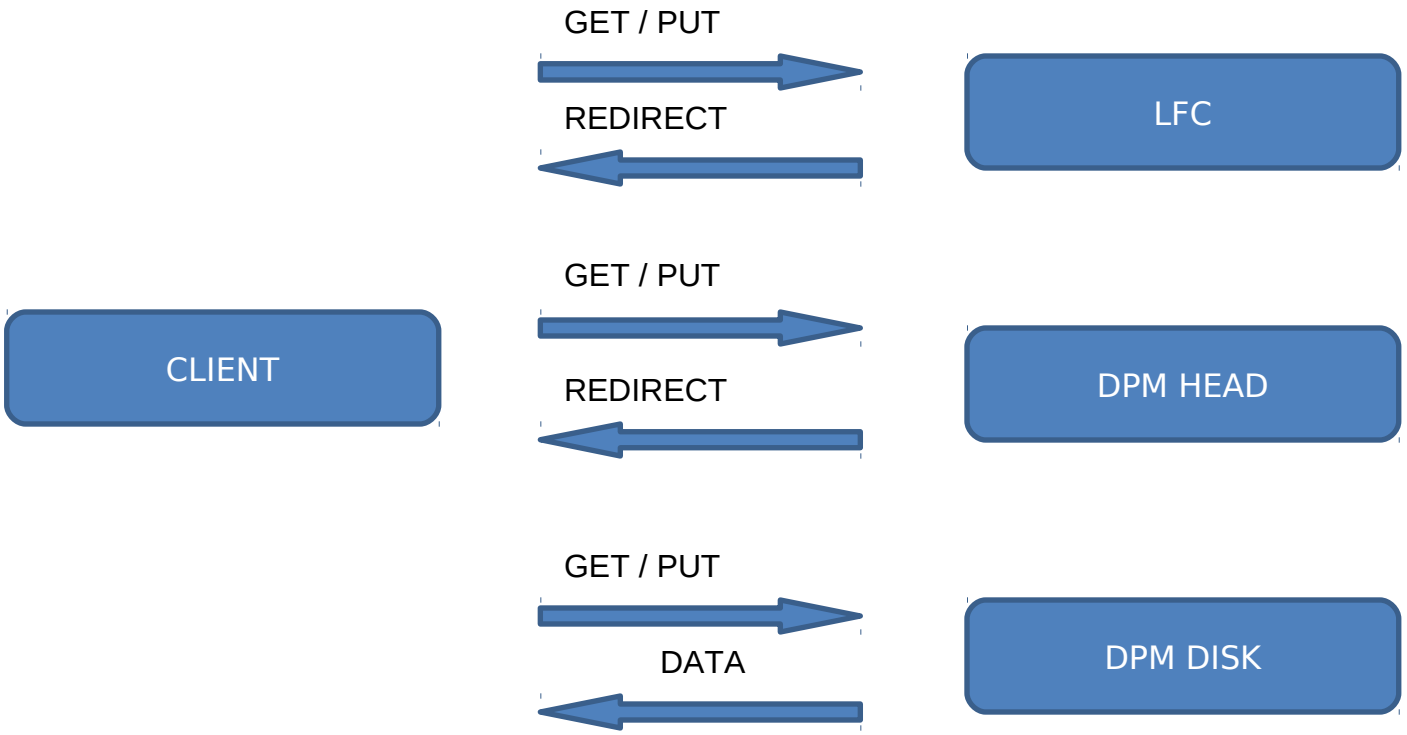
DPM Deployment

- Availability
  - EMI repos
  - Fedora & epel (incl. source rpms!)
- Deployment
  - Yaim (EMI)
  - puppet (github)
- Monitoring
  - Nagios

Catalog -- LFC

- CERN ATLAS LFC:
  230M entries // 293M replicas
- Oracle and MySql support

How it works



- This has been demonstrated using the HTTP interface to LFC & DPM
- Can even incorporate dCache

Authentication

Authentication: X.509
Authorization: voms

except also kerberos for NFS4.1

EMI INFSO-RI-261611

Standards support

SRM for data management

POSIX, HTTP for transfers/access

X.509 for authentication

Nagios for monitoring

Yaim for configuration

also puppet for DPM and LFC

http/webdav Federations

Single entry point to access many storage managers

Automatic failover

Result caching, no DB maintenance

Redirect based on geography

Very fast (~2k ops/sec w/ full cache)

EMI INFSO-RI-261611

Support --- DPM/LFC

- Support for DPM/LFC for LHC Experiments
- Best effort otherwise
- Agreements encouraged

Collaboration being set up – Join!

References

DPM: https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm

LFC: https://svnweb.cern.ch/trac/lcgdm/wiki/Lfc

HTTP Dynamic Federations:
https://svnweb.cern.ch/trac/lcgdm/wiki/Dynafeds

HDFS, S3:
https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Dev/Dmlite
(links are under „plug-ins“)

Use Case Analysis

Generic to all projects

EUROPEAN MIDDLEWARE INITIATIVE

storage and transfer management solutions for large data

- LFC: file-related metadata
- Scientific metadata is handled by the experiments
- Data placement logic / replication is also with the experiments
- Focus on standards-compliance makes our systems easy to integrate

- Remote data access!

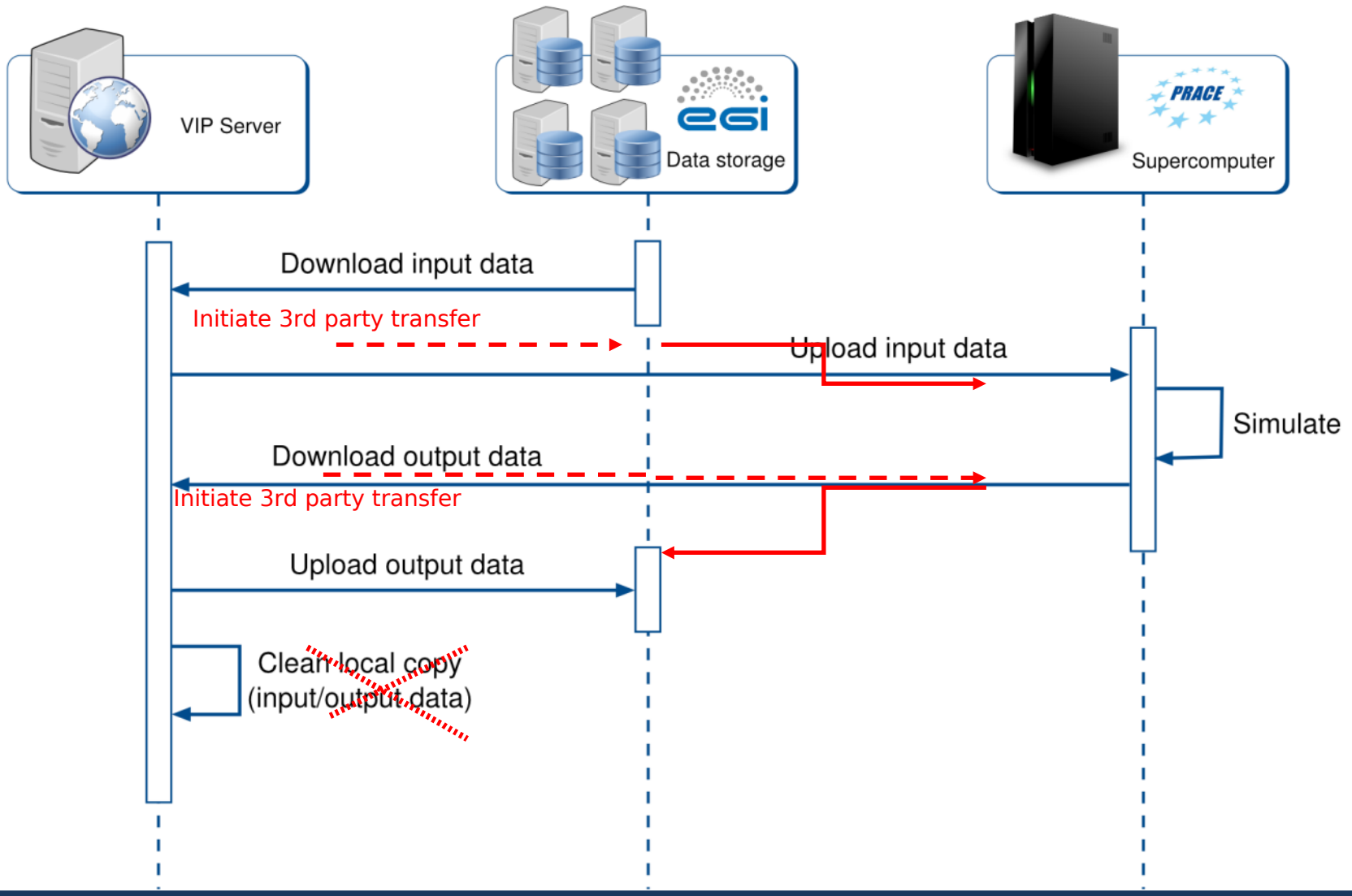
EMI INFSO-RI-261611

VPH

- DICOM backend was developed for DPM
- Hydra DB with distributed keys is still in EMI

- Integrate HDFS and S3 cloud pools into the grid
- Http federation for p-medicine data access

VPH Scenario



EPOS

- PID => GUIDs (40 byte strings)

- “dropbox”
  mount storage w/ gfalFS or NFS4
  send a link, browse with webDAV

VERCE

- We don't have streaming, but good experiences with remote IO
- HDFS support

Win + Linux Compatibility?

HTTP / WebDAV!

ScalaLife

- MapReduce: HDFS integration
- Space reservation: SRM
- Quick & Dirty analysis: Remote file IO
  physicists use ROOT