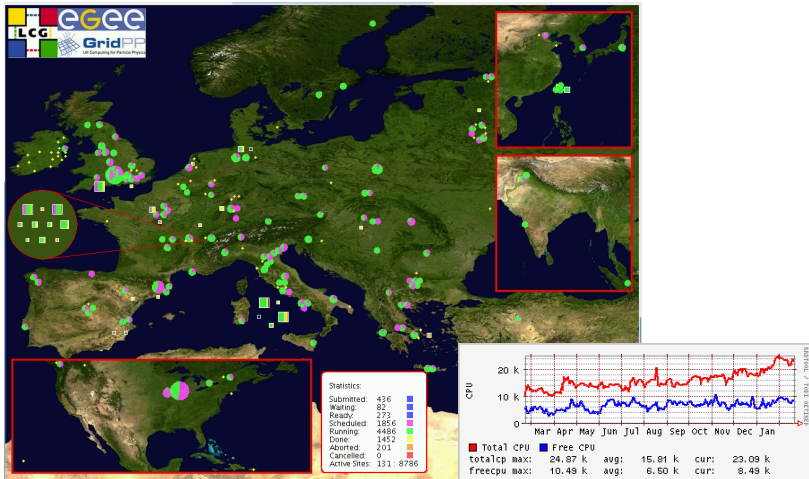


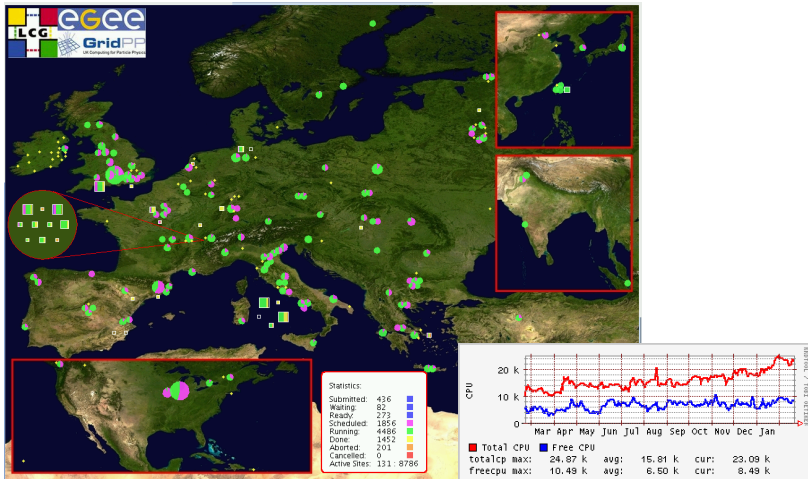
Medical and Secure Data on a Grid.

*John White (for the HIP Technology team
@ CERN)*

John dot White nospam at nospam cern dot ch

- **MSc 1994. Experimental Particle Physics**
 - Test-beam. ATLAS detector.
- **PhD 1998. Experimental Particle Physics**
 - OPAL detector LEP. $\tau \rightarrow h n \pi^0 \nu_\tau$.
 - One of first (50?) webpage authors...
 - First signs of a Higgs boson?
 - Detection of $H \rightarrow \gamma\gamma$ at ATLAS.
- **Back to CERN 1998-2001.**
 - Running OPAL vertex detector.
 - Involved with online computing.
 - Migration of code HP-UX to Linux.
- **2001 HIP and “The Grid”.**
 - Skin mole detection system.
 - Grid DB access.
 - Proxy delegation service.
 - Security cluster manager.
 - Overall MW deputy manager.





- 54 Countries, 267 Sites, 114k CPUs 20PB Storage.
- <http://gridportal.hep.ph.ic.ac.uk/rtm/>

- EGEE offers the largest production grid facility in the world open to many applications:
 - Archeology.
 - Astronomy & Astrophysics.
 - Civil Protection.
 - Computational Chemistry.
 - Computational Fluid Dynamics.
 - Computer Science/Tools.
 - Condensed Matter Physics.
 - Earth Sciences.
 - Finance (through the Industry Task Force).
 - Fusion.
 - Geophysics.
 - High-Energy Physics.
 - Life Sciences.
 - Multimedia.
 - Material Sciences.



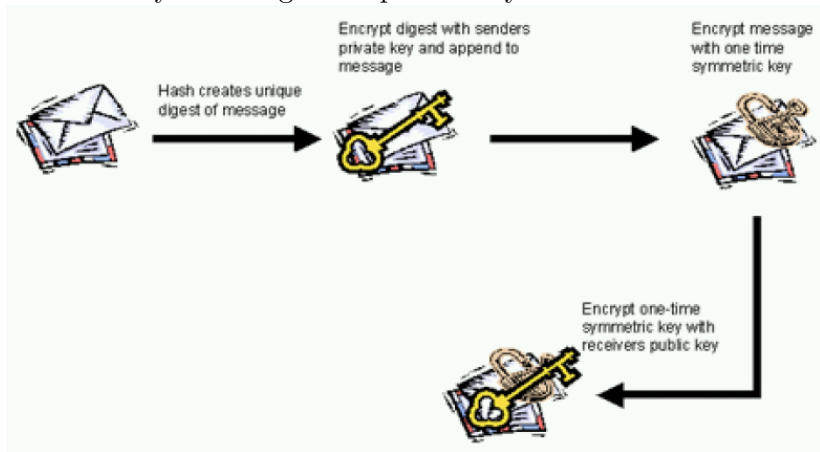
Lightweight Middleware for
Grid Computing

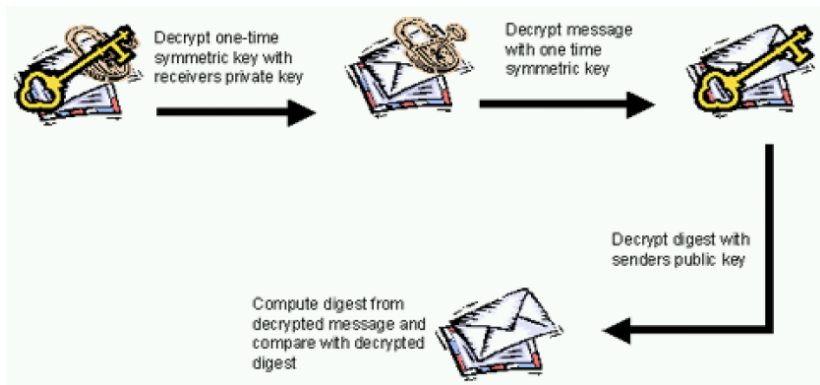
- The gLite middleware essentially defines a Grid infrastructure for high throughput computing.
- gLite “security”:
 - A collection of components.
 - Protects the infrastructure of the middleware.
 - Based on a common theme.
- gLite middleware security based on:
 - **PKI security.** Authentication, Authorization, credential issuance and renewal.
 - **X.509 certificate-based** Credential presentation and exchange.
 - **Virtual Organizations.**

Public Key Infrastructure (PKI).

- Open public encryption standards.
- Users possess Public/Private key pairs.
 - Public key provides identity.
 - Private key securely authenticates.
- Combination of Symmetric/Asymmetric encryption.
- PKI security scheme provides:
 -
 -
 -
 -

Public keys exchanged via private keys or X.509 certificate.





Public Key Infrastructure (PKI).

- Open public encryption standards.
- Users possess Public/Private key pairs.
 - Public key provides identity.
 - Private key securely authenticates.
- Combination of Symmetric/Asymmetric encryption.
- PKI security scheme provides:
 - Digital signature **Authenticates** the message.
 - Message encrypted to ensure **Confidentiality**.
 - Digital signature ensures the **Integrity** of the message.
 - Digital signature uniqueness provides **Non-Repudiation**.

- In the X.509 PKI scheme: X.509 Certificate binds the public key to a Distinguished Name (DN) or other.
- User generates key pair, sends public key as request to higher authority.
 - Certificate obtained from a **Certificate Authority (CA)**.
 - ▶ eg. Verisign, AOL Time Warner, Equifax etc. (CC)
 - ▶ CA should be recognized ¹.
 - ▶ Needs to follow procedure to verify identity.
 - The combination of CA-issued certificate and private key uniquely identifies the user.
 - ▶ Distinguished name (DN) of certificate identifies owner.
 - ▶ Cryptographic routines check the certificate chain.
 - ▶ Private key held confidentially by the owner.

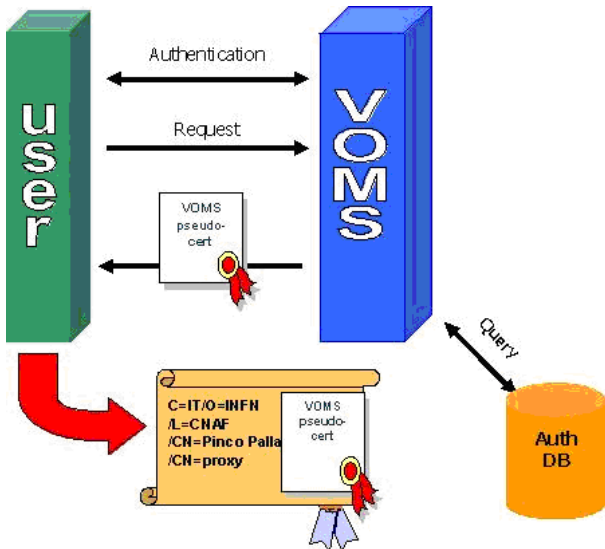
¹<http://www.privacydigest.com/2008/08/11/e+passports+signed+sealed+delivered+not+you+may+t>

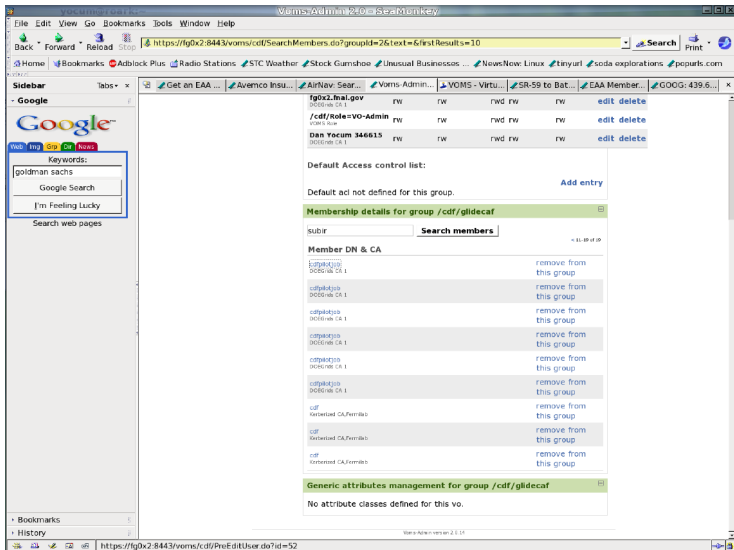
- In the X.509 PKI scheme: X.509 Certificate binds the public key to a Distinguished Name (DN) or other.
- User generates key pair, sends public key as request to higher authority.
 - Certificate obtained from a **Certificate Authority (CA)**.
 - ▶ eg. Verisign, AOL Time Warner, Equifax etc. (CC)
 - ▶ CA should be recognized ¹.
 - ▶ Needs to follow procedure to verify identity.
 - The combination of CA-issued certificate and private key uniquely identifies the user.
 - ▶ Distinguished name (DN) of certificate identifies owner.
 - ▶ Cryptographic routines check the certificate chain.
 - ▶ Private key held confidentially by the owner.
- **This identity valid within the space of the CA.**

¹<http://www.privacydigest.com/2008/08/11/e+passports+signed+sealed+delivered+not+you+may+t>

Virtual Organization: A collection of people and resources.

- Credentials and membership managed by the Virtual Organization Management System (VOMS).
- Consisting of:
 - VOMS server.
 - Administrative interface.
 - CLI clients and Java and C APIs.
- From the VO Admin point of view:
 - VOMS-Admin interface to add/delete members/groups/roles.
- From the VO member point of view:
 - Assigned to groups and assumes roles within groups.





The screenshot shows a web browser window titled "VomsAdmin2fo@SeamMonkey" with the URL `https://lg0x2-8443/voms/cdf/SearchMembers.do?groupId=2&text=&firstResults=10`. The browser's address bar and search bar are visible. The main content area displays a table of VOMS users and their roles.

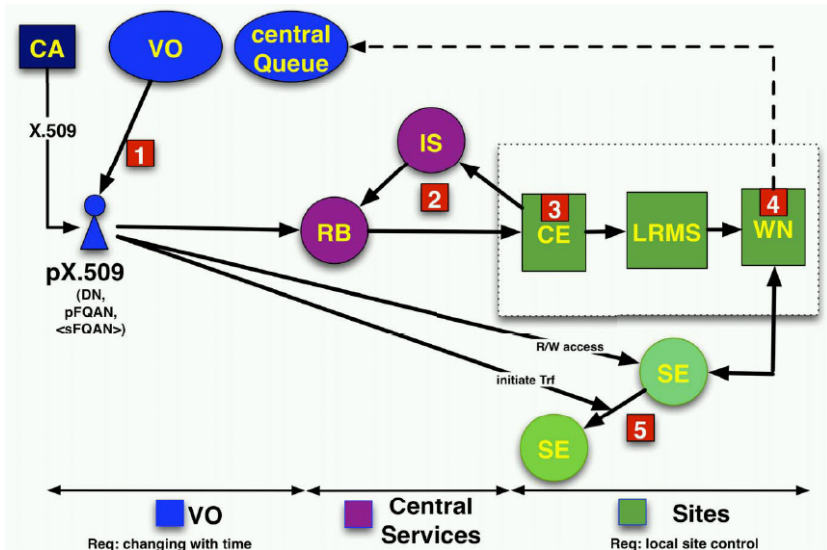
Username	Role	Permissions	Actions
fg0x2.fhal.gov	rw	rw rwd rw rw	edit delete
/cdf/Role=VO-Admin	rw	rw rwd rw rw	edit delete
Dan Yocum 346615	rw	rw rwd rw rw	edit delete

Below the table, there are sections for "Default Access control list" (with an "Add entry" link), "Default acl not defined for this group", and "Membership details for group /cdf/glidecaf". The membership details section includes a search box and a list of members with "remove from this group" links:

- [subir](#)
- [cdffal@jps](#) [remove from this group](#)
- [cdffal@jps](#) [remove from this group](#)
- [cdffal@jps](#) [remove from this group](#)
- [cdffal@jps](#) [remove from this group](#)
- [cdffal@jps](#) [remove from this group](#)
- [cdffal@jps](#) [remove from this group](#)
- [cdf](#) [remove from this group](#)
- [cdf](#) [remove from this group](#)
- [cdf](#) [remove from this group](#)

At the bottom, there is a section for "Generic attributes management for group /cdf/glidecaf" which states "No attribute classes defined for this vo." The browser's status bar at the bottom shows the URL `https://lg0x2-8443/voms/cdf/PreEditUser.do?id=52`.

- Therefore every user in a VO is characterized by attributes (group, role).
 - The combined values form unique attributes (FQANs).
 - Can represent an FQAN as a sequence of group names.
 - Each may be qualified with one or several roles [and capabilities] in that group.
- In general, an FQAN has the following form:
- /VO[/group[/subgroup(s)]][/Role = role]
 - eg. the FQANs for the a member may be:
 - /computingcompany.com/Administration/Role=Director
 - /computingcompany.com/Research/Role=Lead
 - /computingcompany.com/Finance/Role=User
- Certificate distinguished name (DN)+ VOMS attributes identifies a unique presence within the VO.



- Identity management within a VO.
 - Services need to understand these credentials.
 - Grid-enabled services do understand these.
- Credentials with VOMS attributes used for:
 - Submitting jobs (**RB**).
 - Write/read/modify data (**SE**).
 - Transferring data (**SE** to **SE**).

Real use case with commercial parallels.

Biomedical Research.

- EU and national regulations on data.^{2 3}
- Patient data used in research must be protected.
 - Anonymized and encrypted.
- Must be useable in a Grid environment.
- User's CA/VO-issued credentials gain access to:
 - Shared and external datasets or databases.
 - HPC/IT-centers, Hospitals, Financial institutions.
 - Commercial Clouds, Financial institutions.
 - Shared and connected instruments
 - ▶ Scanners, Sensors, Detectors.
- **Similar concerns in business data handling?**

²<http://www.guardian.co.uk/politics/2009/mar/03/medical-records-data>

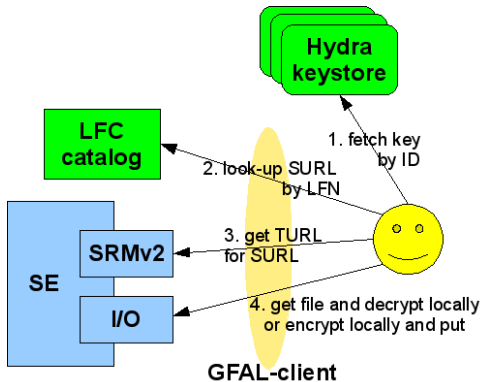
³<http://www.guardian.co.uk/technology/2006/jul/06/epublic.guardianweeklytechnologysection>



Hydra is a distributed key storage solution.

- Encryption key generated on demand.
- Split and distributed to multiple keystores (databases)
- The splitting scheme is “non-trivial”...
Shamir Secret Sharing Scheme.
- Need N out of M key parts to reconstruct key.
 - Mathematically proven secure: $< N$ parts not enough to reconstruct key.
 - Fault Tolerant: unavailability of M-N keystores not a problem.
 - eg. 3 of 5 scheme:
Two databases may be unavailable.
Need to crack three to reveal key.

General Standalone scheme.



- User may manage key IDs. (Or use metadata service)
- SURLs held in catalogue such as LFC.
- TURLs obtained through GFAL.
- Files transferred from TURL.
- (De)Encryption locally.
- Provides a general Encrypted Data Storage (**EDS**).

```

jwhite@pchip12:~$
[pchip12] /home/jwhite > glite-eds-key-register 25_02_2009_key2
A key has been generated and registered for ID '25_02_2009_key2'
[pchip12] /home/jwhite > wd5sum /tmp/test_file_to_encrypt
146207ba0c5b373c7e6f3e8b05adcle /tmp/test_file_to_encrypt
[pchip12] /home/jwhite > glite-eds-encrypt 25_02_2009_key2 /tmp/test_file_to_encrypt /tmp/encrypted_file

File '/tmp/test_file_to_encrypt' has been successfully encrypted
with key '25_02_2009_key2'
and written to '/tmp/encrypted_file'
[pchip12] /home/jwhite > glite-eds-decrypt 25_02_2009_key2 /tmp/encrypted_file /tmp/decrypted_file

File '/tmp/encrypted_file' has been successfully decrypted
with key '25_02_2009_key2'
and written to '/tmp/decrypted_file'
[pchip12] /home/jwhite > wd5sum /tmp/decrypted_file
146207ba0c5b373c7e6f3e8b05adcle /tmp/decrypted_file
[pchip12] /home/jwhite > []

```

```

root@pchip12:~$
-----
| 3 |         4 | 128 |
| 9 |         5 | 64  |
-----
2 rows in set (0.00 sec)

mysql> select * from t_entry;
-----
| entry_id | entry_name | owner_id | group_id | user_perms | group_perms | other_perms | schema_id | creation_time |
-----
| 3 | test01 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-20 13:23:58 |
| 4 | test02 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-20 13:24:33 |
| 5 | test03 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-20 13:42:56 |
| 6 | test04 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-20 13:56:01 |
| 7 | test05 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-20 14:58:34 |
| 8 | test11 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-20 17:16:49 |
| 9 | test12 | 5 | 2 | 239 | 0 | 0 | 1 | 2009-02-20 17:17:10 |
| 10 | testkey | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 17:13:42 |
| 11 | test_11_02_2009_01 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 17:14:04 |
| 12 | test_11_02_2009_02 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 17:21:10 |
| 13 | test_11_02_2009_03 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 17:21:29 |
| 14 | test_11_02_2009_04 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 17:54:20 |
| 15 | test_11_02_2009_05 | 5 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 18:00:13 |
| 16 | test_11_02_2009_06 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 18:04:55 |
| 17 | test_11_02_2009_07 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-23 18:05:36 |
| 18 | test_25_02_2009_v1 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-25 16:50:01 |
| 19 | test_25_02_2009_v2 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-25 16:50:12 |
| 20 | test_25_02_2009_v3 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-25 16:50:24 |
| 21 | test_25_02_2009_v4 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-25 16:52:47 |
| 22 | test_25_02_2009_v5 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-25 16:53:06 |
| 23 | 25_02_2009_key1 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-25 16:54:07 |
| 24 | 25_02_2009_key2 | 1 | 2 | 239 | 0 | 0 | 1 | 2009-02-25 16:54:27 |
-----
22 rows in set (0.00 sec)

mysql> []

```


- `/home/jwhite > glite-eds-key-register 25_02_2009_key2`

A key has been generated and registered for ID '25_02_2009_key2'

- `/home/jwhite > md5sum /tmp/test_file_to_encrypt`

14b207bab50b373c7e5f3e9b05adc1e /tmp/test_file_to_encrypt

- `/home/jwhite > glite-eds-encrypt 25_02_2009_key2 /tmp/test_file_to_encrypt /tmp/encrypted_file`

File '/tmp/test_file_to_encrypt' has been successfully encrypted with key '25_02_2009_key2' and written to '/tmp/encrypted_file'.

- `/home/jwhite > glite-eds-decrypt 25_02_2009_key2 /tmp/encrypted_file /tmp/decrypted_file`

File '/tmp/encrypted_file' has been successfully decrypted with key '25_02_2009_key2' and written to '/tmp/decrypted_file'.

- `/home/jwhite > md5sum /tmp/decrypted_file`

14b207bab50b373c7e5f3e9b05adc1e /tmp/decrypted_file

- `/home/jwhite > glite-eds-key-register 25_02_2009_key2`

A key has been generated and registered for ID '25_02_2009_key2'

- `/home/jwhite > md5sum /tmp/test_file_to_encrypt`

14b207bab50b373c7e5f3e9b05adcle /tmp/test_file_to_encrypt

- `/home/jwhite > glite-eds-encrypt 25_02_2009_key2 /tmp/test_file_to_encrypt /tmp/encrypted_file`

File '/tmp/test_file_to_encrypt' has been successfully encrypted with key '25_02_2009_key2' and written to '/tmp/encrypted_file'.

- `/home/jwhite > glite-eds-decrypt 25_02_2009_key2 /tmp/encrypted_file /tmp/decrypted_file`

File '/tmp/encrypted_file' has been successfully decrypted with key '25_02_2009_key2' and written to '/tmp/decrypted_file'.

- `/home/jwhite > md5sum /tmp/decrypted_file`

14b207bab50b373c7e5f3e9b05adcle /tmp/decrypted_file

- **OK. So we can encrypt/decrypt...**

- `/home/jwhite > glite-eds-key-register 25_02_2009_key2`

A key has been generated and registered for ID '25_02_2009_key2'

- `/home/jwhite > md5sum /tmp/test_file_to_encrypt`

14b207bab50b373c7e5f3e9b05adcle /tmp/test_file_to_encrypt

- `/home/jwhite > glite-eds-encrypt 25_02_2009_key2 /tmp/test_file_to_encrypt /tmp/encrypted_file`

File '/tmp/test_file_to_encrypt' has been successfully encrypted with key '25_02_2009_key2' and written to '/tmp/encrypted_file'.

- `/home/jwhite > glite-eds-decrypt 25_02_2009_key2 /tmp/encrypted_file /tmp/decrypted_file`

File '/tmp/encrypted_file' has been successfully decrypted with key '25_02_2009_key2' and written to '/tmp/decrypted_file'.

- `/home/jwhite > md5sum /tmp/decrypted_file`

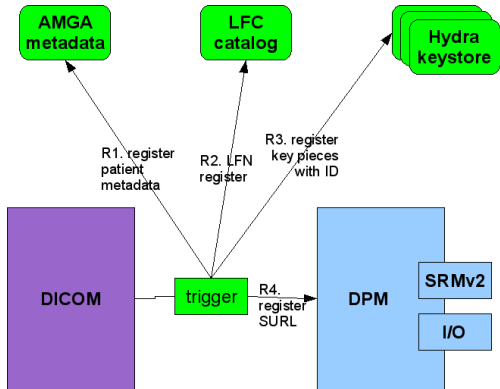
14b207bab50b373c7e5f3e9b05adcle /tmp/decrypted_file

- **OK. So we can encrypt/decrypt...**

- **Where is this used?**

- Hydra integrated to produce **Medical Data Manager** (MDM).
- Medical data: Managed by DICOM storage.
 - **D**igital **I**maging and **C**ommunications in **M**edicine (**DICOM**): standard.
 - DICOM designed for internal hospital usage.
 - Should not be exposed to general Grid environment.
- EGEE solution: extension of Data Management tools.
 - DICOM/DPM interface.
 - Encryption/decryption of data on the fly.
 - Meta-data management.
 - SRMv2, GFAL, gridFTP for SURLs, TURLs and transfers.
 - Encryption key management/protection. (**Hydra**)

File Registration

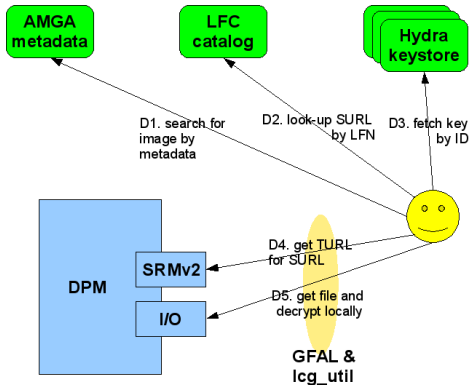


- DPM-DICOM trigger:
 - Registered to AMGA, LFC.
 - Uses LFC (or DICOM) GUID for ID.
 - Registers keys under ID to Hydra.
 - Stores encrypted file to DPM.
- GUID important (retrieval).

dpm-dicom-trigger <DICOM file name>

MDM-register <DICOM file name>

File Retrieval.



- LFC gives GUID for Hydra ID.
- Security at:
 - AMGA, LFC, Hydra, SE.
- DPM I/O access via: gridftp, rfi(s), http(s).
- The encrypted file can be retrieved from any GFAL.

```
lcg-cp -bD srmv2 <SURL><ID> <encrypted file name>
glite-eds-decrypt <ID> <encrypted file name> <file name>
or
```

```
glite-eds-get -i <ID> rfi:///dpm/example.org/home/biomed/mdm/<ID> <file name>
```

1. Install the Hydra services:

- Install the packages from gLite repository:
 - `yum update tomcat5`
 - `yum update jpackage`
 - `yum install mysql-server`
 - `yum install glite-BDII`
 - `yum install bdii glite-info-templates`
 - `yum install glite-HYDRA.mysql`
- Please see:
<https://twiki.cern.ch/twiki/bin/view/EGEE/DMEDS>
- Installed on each Hydra key store (machine).

2. Install the gLite UI for clients:

- `yum install glite-UI`

Example configuration file (eg. host1.example.org).

```
HYDRA_INSTANCES="1"
HYDRA_DBNAME_1=hydra_db_table_name_on_host1
HYDRA_DBUSER_1=host1_db_manager_account
HYDRA_DBPASSWORD_1=<secret>
HYDRA_CREATE_1=/your_vo
HYDRA_ADMIN_1=<admin-dn on host 1>
```

```
HYDRA_PEERS="2 3"
HYDRA_CREATE_2=/your_vo
HYDRA_ID_2=1
HYDRA_HOST_2=host2.example.com
HYDRA_CREATE_3=/your_vo
HYDRA_ID_3=1
HYDRA_HOST_3=host3.example.net
```

Repeated on peers

host2.example.com **and** host3.example.net.

Configuration.

- Previous configuration file input to gLite YAIM.
 - `yaim -c -s <path_to_config_file> -n HYDRA`
 - Produces the `services.xml` file.
 - Sets up the Tomcat context resources
- Needs to be repeated on the other peers.
- Start the Tomcat service on each peer.

- Organization members identified by certificates (or other).
 - Own CA or (commercial?) other.
- Organization groups and roles defined within VOMS.
 - VOMS Server provides the group/role attributes to users .
 - VOMS groups/roles administered through the VOMS-Admin interface.
- Users access the encrypted data storage with credentials.
 - VOMS attributes define the access to data.
- Data held in Encrypted Data Storage.
 - Keys split and held at separated locations.
 - Access to data and keys determined by credentials and VOMS attributes.



Lightweight Middleware for
Grid Computing

What does HIP Technology do?.

- Java Trustmanager.
 - Proxy certificate routines for Java.
 - Chain-checking, namespace checking etc.
- Java Security Utilities.
 - Extraction of VOMS attributes etc.
- Secure Data Storage.
 - Encrypted data, key splitting.
- Pseudonymity Service.
 - Pseudo-anonymous access to Grid resources.
- Proxy Delegation Service.
 - Secure method to move proxies.
- New Authorization Service.
 - XACML/SAML-based Authorization.



Lightweight Middleware for
Grid Computing