Medical Data Management

Ákos Frohner on behalf of the Grid DM Team

CERN – MDM Demo, 2007-12-18

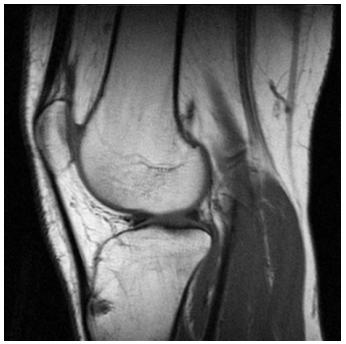
Biomed Requirements

Problem : Medical institutes request data storage encryption

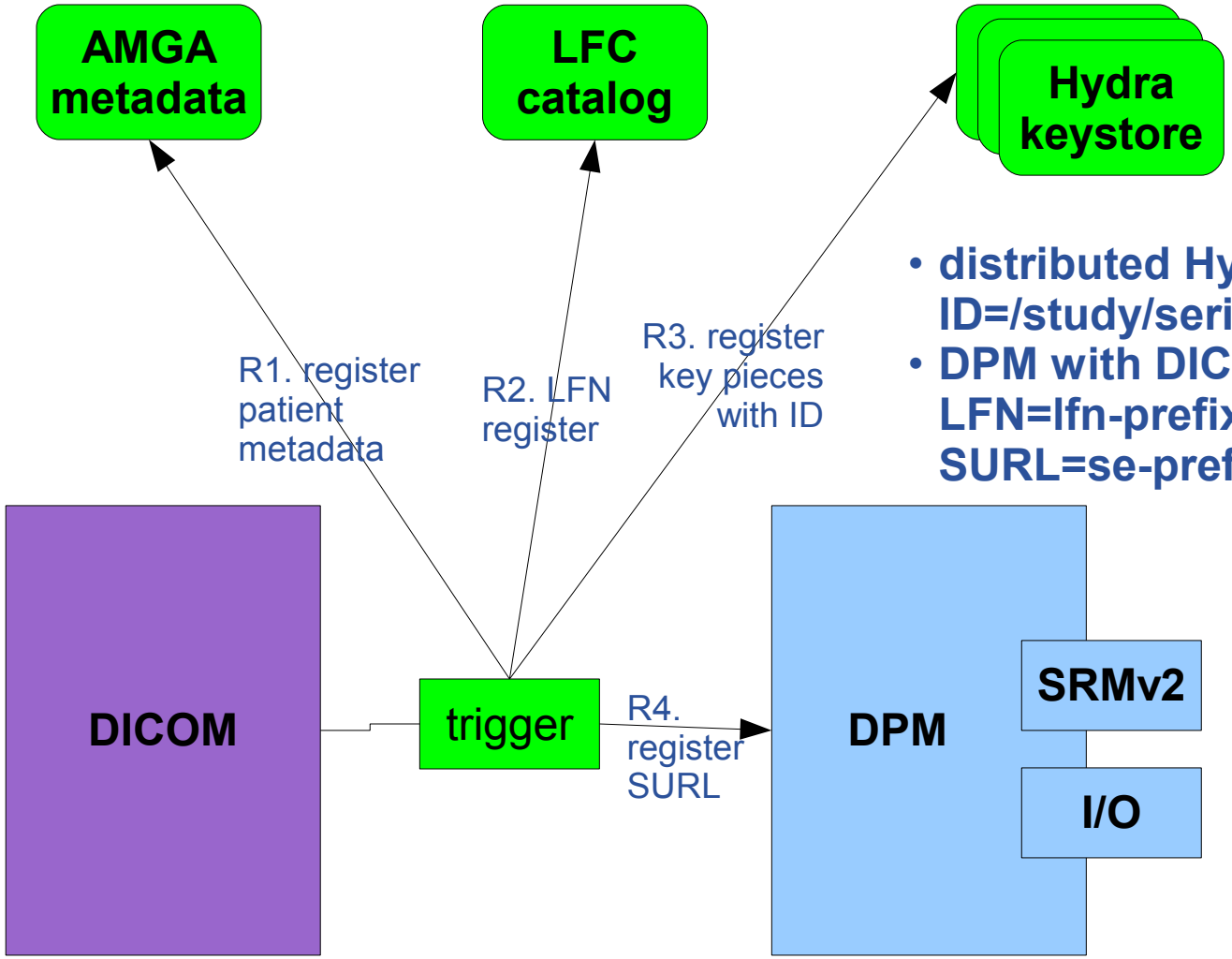
- Use of the DICOM standard for medical image handling
- Image retrieval and storage from/in DICOM servers : security issues

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



Registration



- distributed Hydra keystore
  ID=/study/series/SOP ID
- DPM with DICOM back-end
  LFN=lfn-prefix/study/series/SOP ID
  SURL=se-prefix/study/series/SOP ID
  - back-end is part of a standard DPM service
  - encrypted channel to LFC

Registration example

dpm-dicom-trigger MDM_test-0.78/test/dicom/jm0301-00032.dcm

- uploads a file to DICOM
- fetches the file from DICOM (file size may change!)
- calculates the encrypted size and ID, for example: ID=/1.2.826.0.1.3680043.2.1143..2006020212450229 8.29/1.2.826.0.1.3680043.2.1143..2006020212450229 8.29/1.2.826.0.1.3680043.2.1143..2006020212450446 8.51
- calls 'dpm-register' to register the file into DPM with the SFN /dpm/<domain>/home/biomed/mdm/<ID> and with the PFN dicom:///<ID> and marked as near-line copy
- calls 'glite-eds-register' to create an en/decryption key and adds the DPM host to be able to read it (i.e. glite-eds-setacl)
- (registers the file in LFC with the LDN /grid/biomed/mdm/<ID>)
- (registers the file metadata in AMGA)

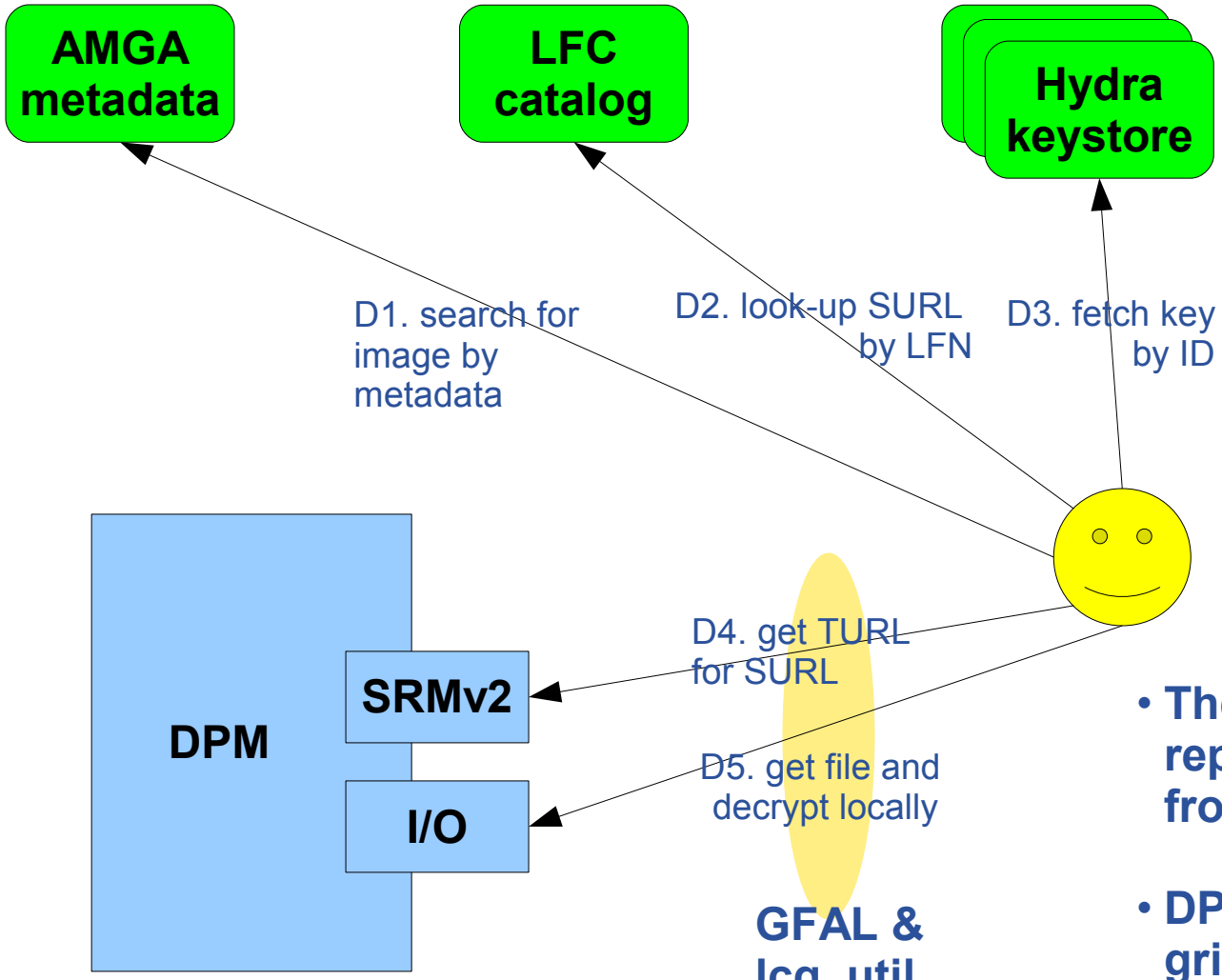
Key splitting

Keys are split for security and reliability reasons using the Shamir's Secrect Sharing Scheme (org.glite.security.ssss)

- **standalone library and CLI**
- **modified Hydra service and Hydra client library/CLI**
- **the client contacts all services for key registration, retrieval and to change permissions**
  - there is no synchronization or transaction coordinator service

```
$ glite-ssss-split-passwd -q 5 3 secret
137c9547aba101ef 6ee7adbbaacac1ef 1256bcc160eda592 fdabc259cdfbacc9
    3113be83f203d794
$ glite-ssss-join-passwd -q 137c9547aba101ef NULL 1256bcc160eda592 NULL
    3113be83f203d794
secret
```

Retrieval



- **The encrypted file can be replicated and retrieved from any other SE.**
- **DPM I/O access via: gridftp, rfio(s), http(s)**

Retrieval Example

lcg-cp -bD srmv2 srm://dpm.example.org:8446/srm/managerv2?SFN=/dpm/example.org/home/biomed/mdm/<ID> file:picture.enc

glite-eds-decrypt <ID> picture.enc picture

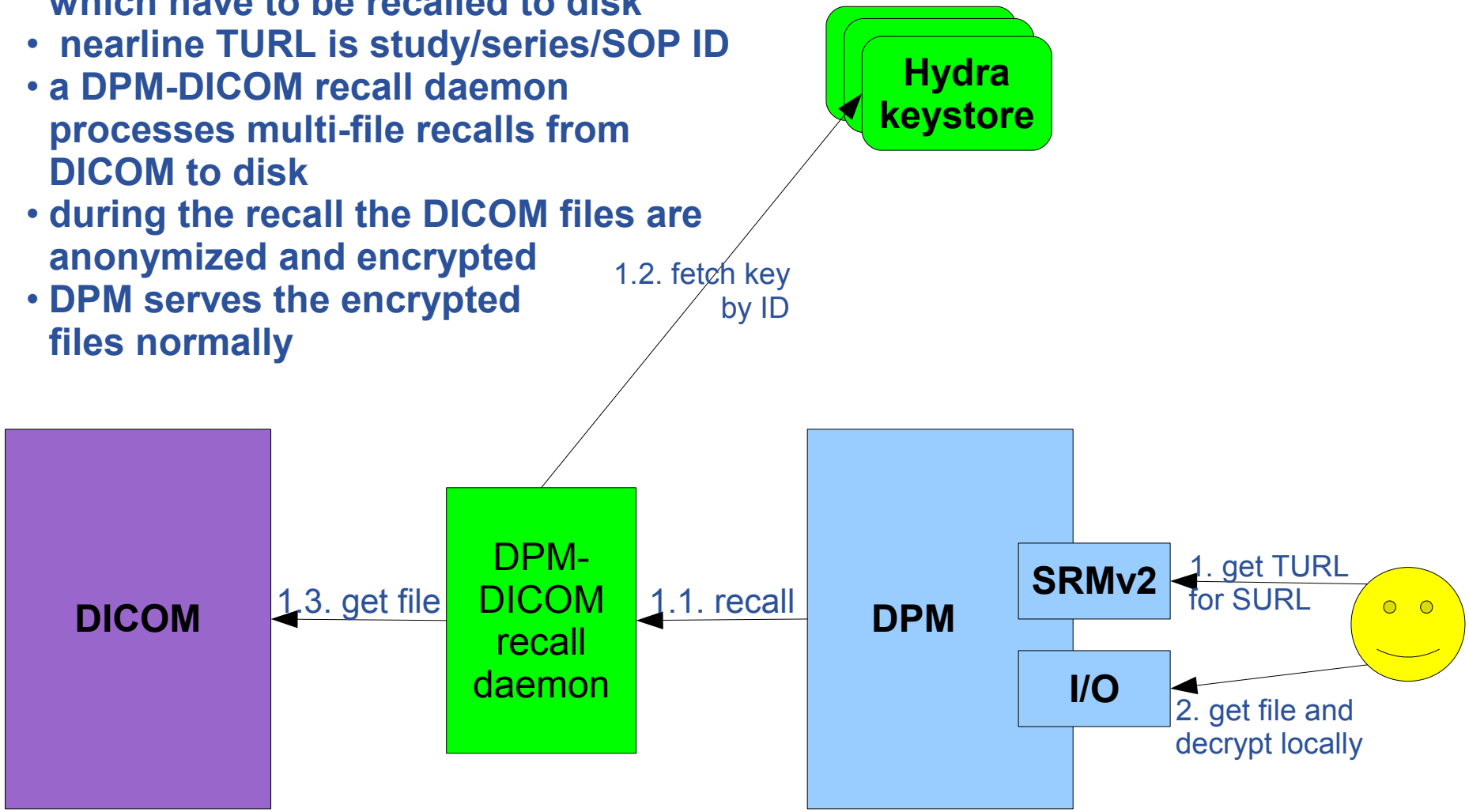
glite-eds-get -i <ID> rfio:////dpm/example.org/home/biomed/mdm/<ID> picture

- file is opened via gfal_open()
- decryption key is fetched for <ID>
- loop on gfal_read(), glite_eds_decrypt_block(), write()

'glite-eds-get' is a simple utility over the EDS library.

DPM-DICOM interaction

- DICOM files are marked as 'nearline', which have to be recalled to disk
- nearline TURL is study/series/SOP ID
- a DPM-DICOM recall daemon processes multi-file recalls from DICOM to disk
- during the recall the DICOM files are anonymized and encrypted
- DPM serves the encrypted files normally



DPM-DICOM Details

DPM-DICOM recall daemon

- multiple multi-file requests
- DPM starts the retrieval by a direct call, however jobs are also stored in DB
- recall daemon manages the DB and DPM interactions and calls a plug-in for each PFN (see dicom:///<ID>)
- simple plug-in interface, within a shared library in libdpm_dicom.so:
  int dpm_dicomcopyfile (char *dicom_fn, char *turl, int *errcode, char *errbuf)
- plug-in retrieves the encryption key from Hydra, retrieves the file from DICOM and stores the file via rfio

Status and Plans

- DPM-DICOM recall daemon is coded
  to be released in DPM v1.7.0, January 2008
- DICOM image anonymization and encryption ready
- Hydra key splitting is ready
  to be released in December 2007
- Hydra/GFAL CLI is coded, being tested
  to be released in December 2007
- LFC encryption is coded, needs integration and tests
- Still needs to test multi-file, concurrent requests and various error conditions