Encrypted Data Storage in EGEE

Ákos Frohner, Péter Kunszt, Ricardo Brito Da Rocha (CERN),
Patrick Guio (University of Bergen), Zoltán Farkas (Sztaki)
Johan Montagnat (CNRS, I3S laboratory), Daniel Jouvenot (LAL laboratory), Christophe Pera (CNRS, CREATIS laboratory)

Motivation

Medical community as the principal user

- large amount of images are produced
- privacy concerns vs. processing needs
- ease of use (image production and application)

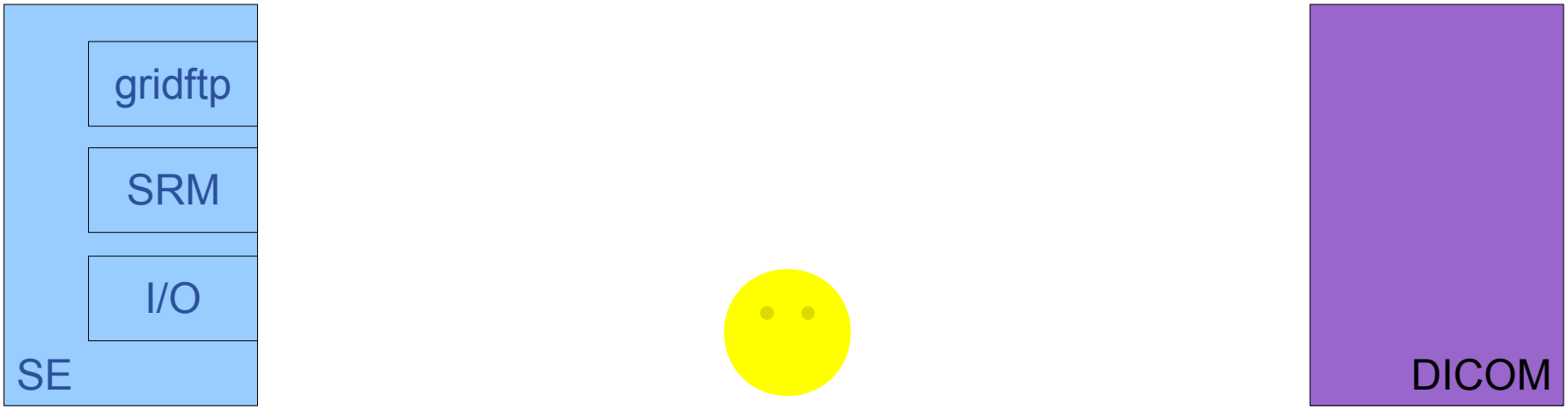
Strong security requirements

- anonymity (patient data is separate)
- fine grained access control (only selected individuals)
- privacy (even storage administrator cannot read)

MDM = Medical Data Management

Building Blocks

- Hospitals: DICOM = Digital Image and COmmunication in Medicine
- Grid: SE = SRM + gridftp + I/O
- and a client (application processing an image)
- [data transfer services among storage facilities]



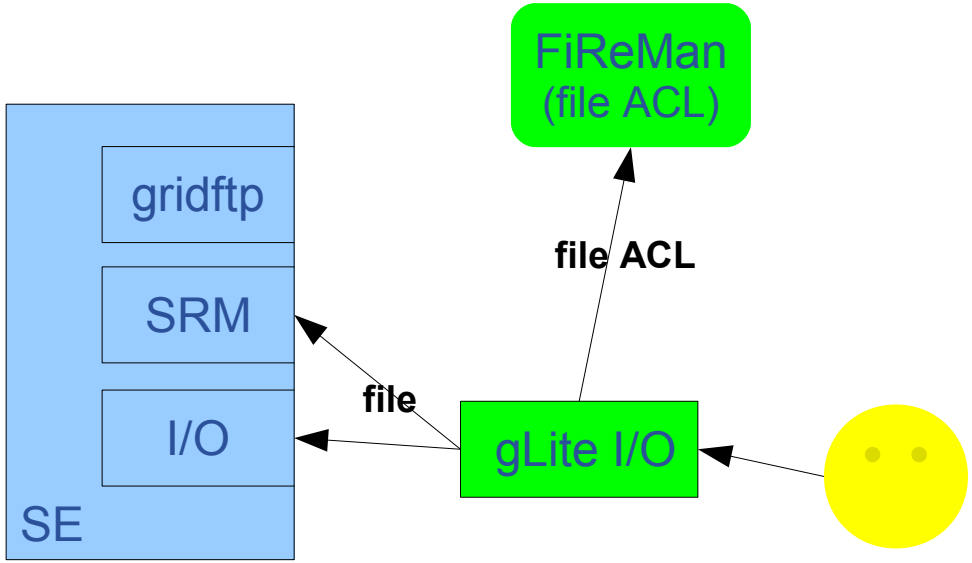
Goal: data access at any location

Access Control

- complex access patterns with many individuals enlisted
- no ACL support in currently used Storage Elements

"wrap" the SE into a service which enforces ACLs

- gLite I/O: authorization enforcement
- File and Replica Manager (FiReMan): ACL store

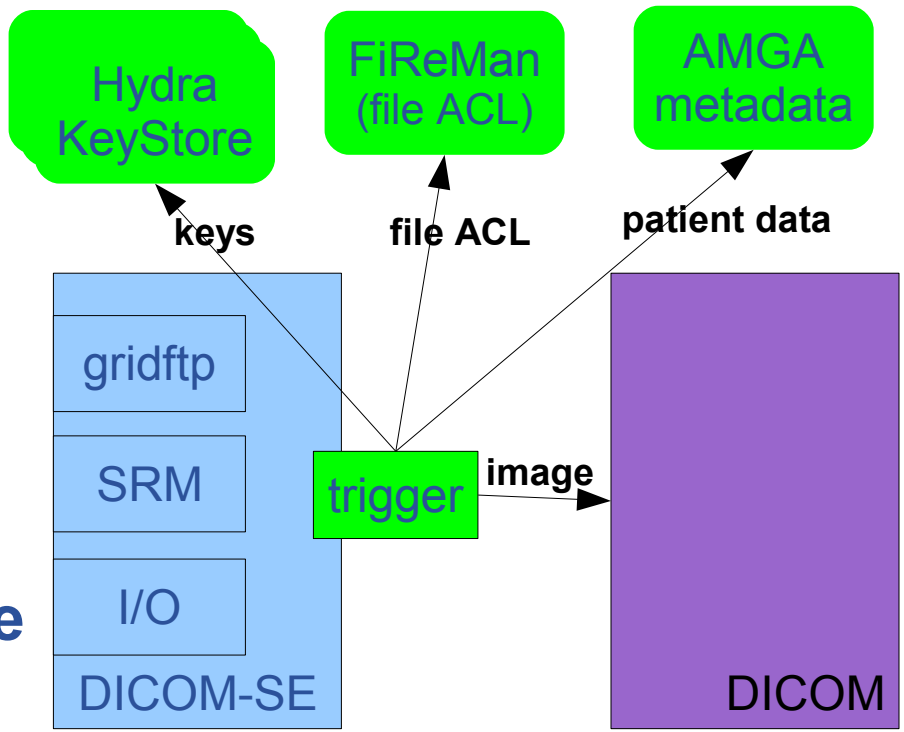


Encryption...

"wrapping" DICOM to satisfy the security requirements:

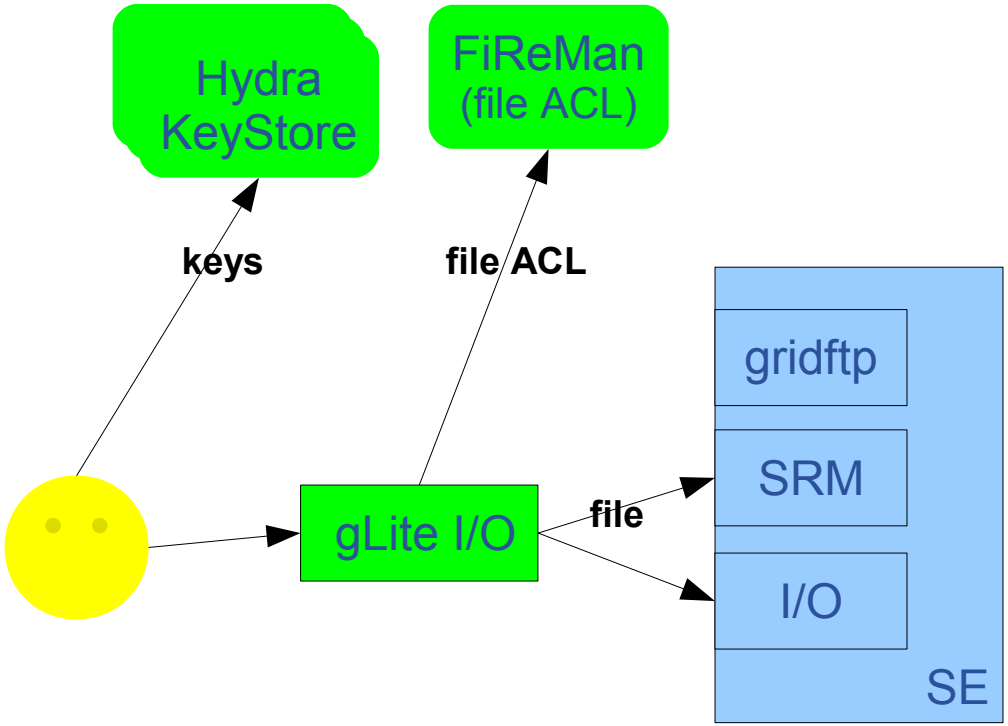
- anonymity: patient data is separated and stored in AMGA
- access control: ACL information on individual files in FiReMan
- privacy: per-file keys are distributed among the Hydra key servers with fine grained access control

Image is retrieved from DICOM and processed to be "exported" to the grid.



... and decryption

- key is retrieved from the Hydra key servers
- data is decrypted block-by-block in memory only (OpenSSL cyphers)
- encryption also works for output data



Conclusion and Future Work

- components are part of the gLite software stack
- tested with applications – see the MDM demo

- SRMv2 includes access control functions: “wrapping” of SE could be eliminated
- ease application integration (e.g. Parrot or FUSE)
- integrating key distribution algorithms (m-of-n key split)