CHIMERA – A NEW, FAST, EXTENSIBLE NAME SERVICE

Tigran Mkrtchyan, DESY, Hamburg

What about?

In this presentation I will show how we reference files stored in HSM.

I will show how we do it now, how we want to do in the future and why.

We want to provide a generic solution which can fit into your needs as well.

Why?

Storage systems need to handle filenames and actual data locations.

In case of regular file systems names are located “near” the data.

In case of complex storage systems we need a central service for filenames of data, distributed over a large number of storage locations (disks, tapes).

How?

- Unique file ID independent from name
- Path ⇔ ID maping
- Mechanism for clients to store metadata
- Directory tags, inherited by subdirectories
- Callbacks on FS events (at least on ***rm***)

How?

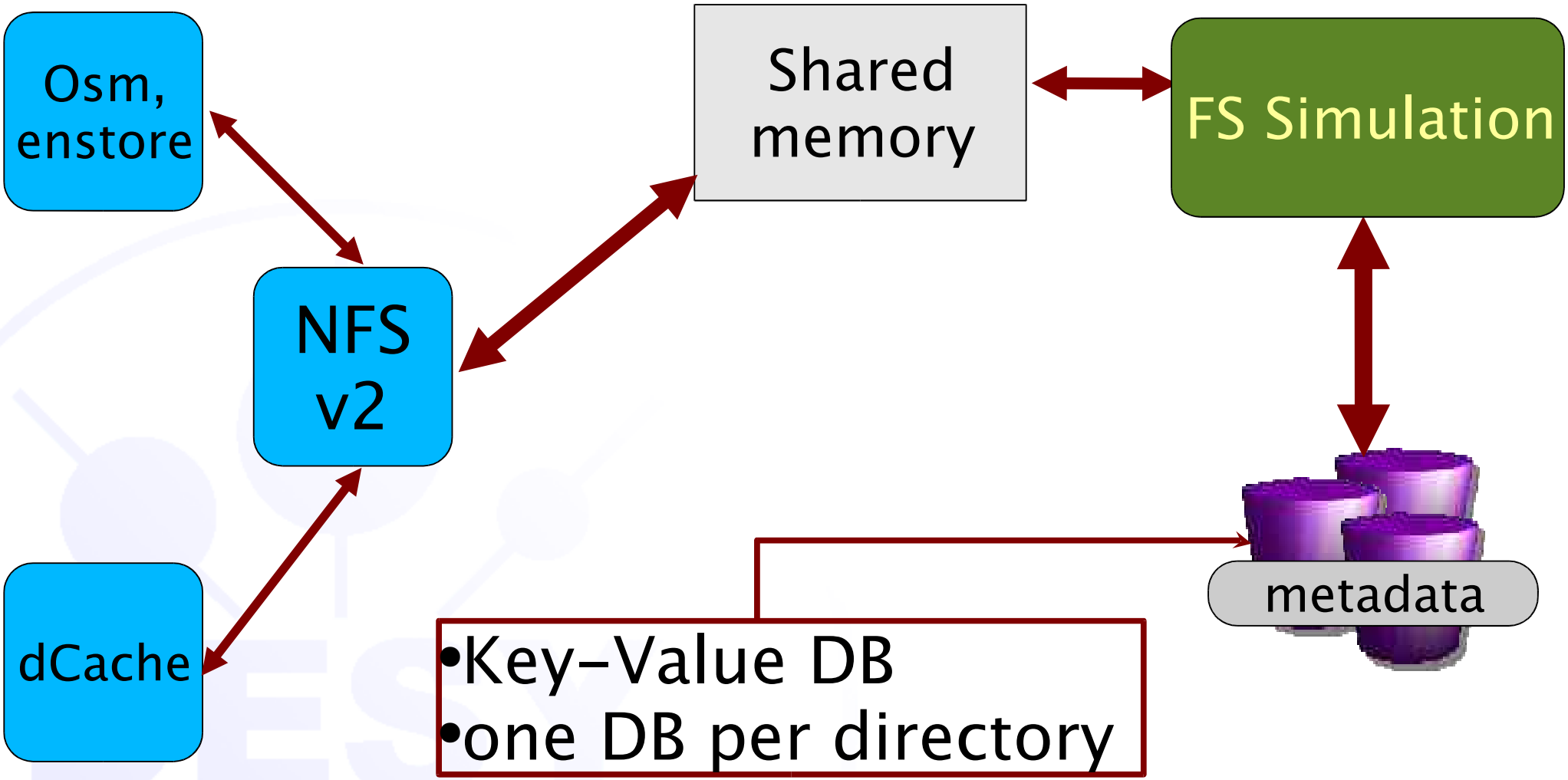
In 1997 we have introduced the PNFS - NFSv2 based file system on top of database.

PNFS supports all NFSv2 namespace operations.

The actual data IO is done by HSM native store/retrieve utilities.

PNFS is able to store user defined metadata associated with files and directories.

Current approach



Current access

- ~1000 of clients
- More than 3 mln. entries (500 TB)
- 20 Hz data open/create
- 1kHz NFSops access rate

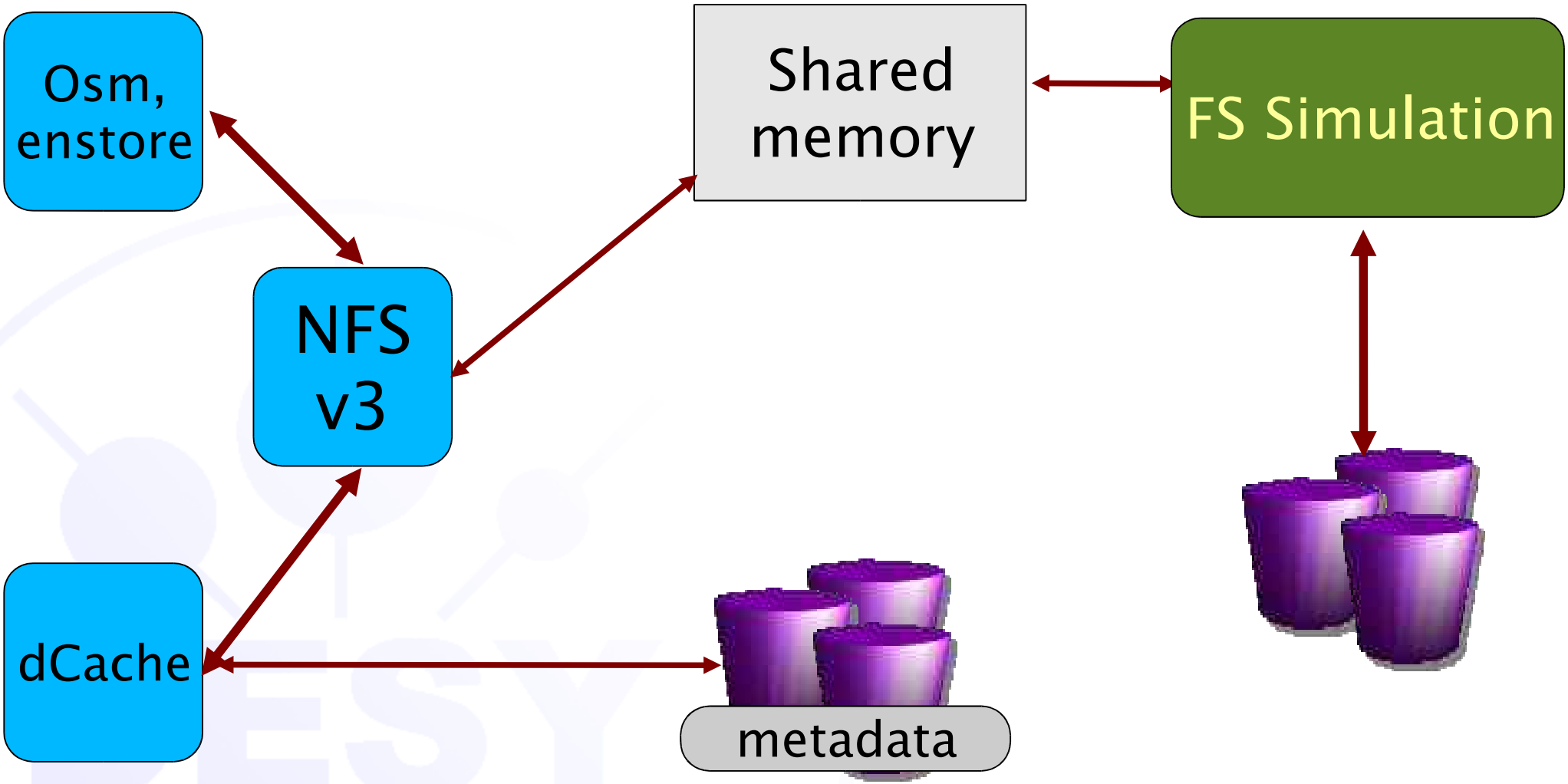
Clients

- Enstore – stores HSM related information
  tape name, file position and so on.
- OSM – stores HSM related information
  link to bitfileid database
- dCache – stores file locations, checksums,
  persistency flags.
- User-level applications
  (ls, find, mkdir, rm, mv)

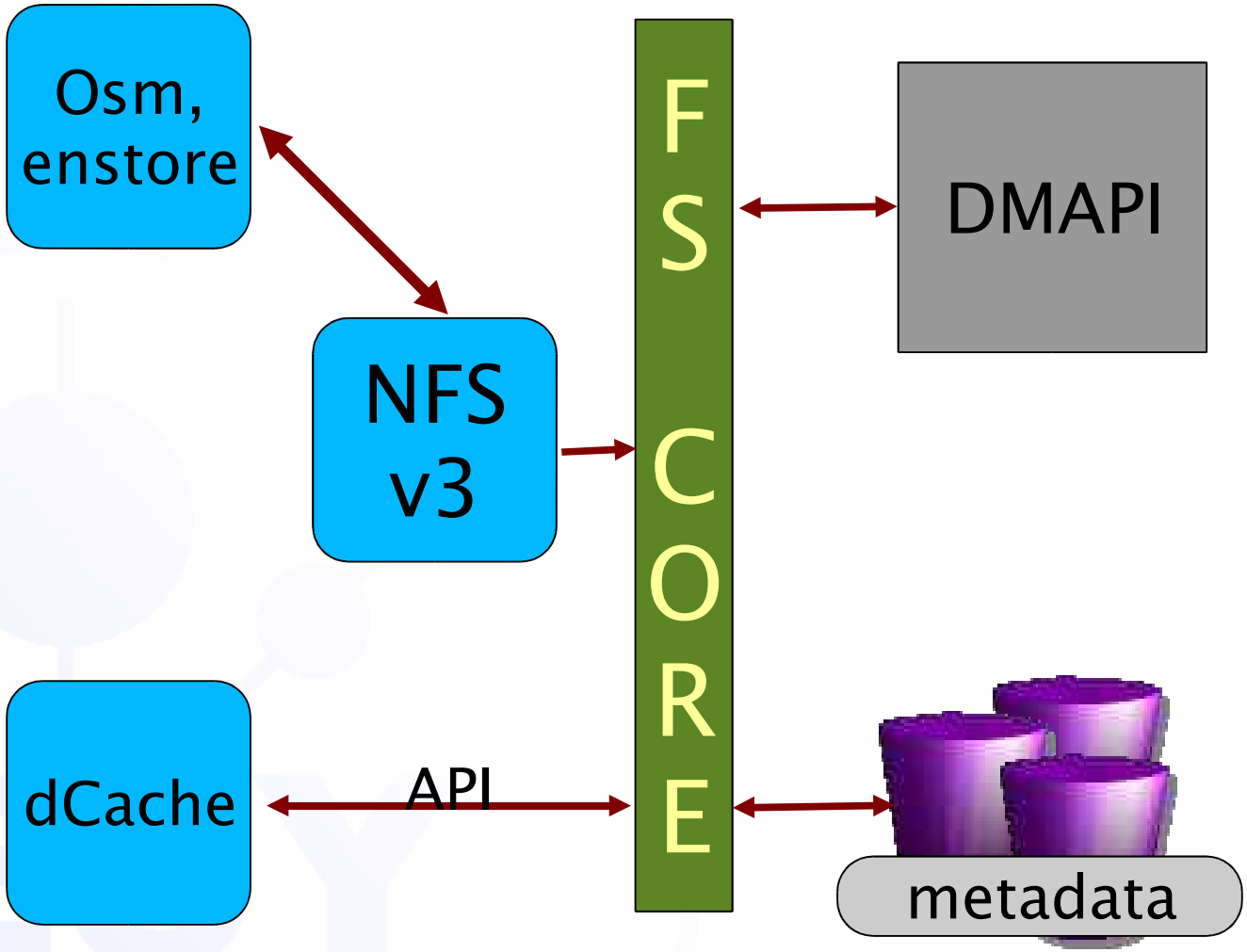
Limitation of PNFS

- Max. file size 2 Gb
- Metadata access only through NFS
  - no direct path for attached storage systems
  - all metadata types use same channel and store
    - heavy access to metadata by storage system has performance impacts on regular NFS operations.
- Metadata stored as BLOB
  - no metadata query functionality
- No ACLs
- NFSv2 security (no security)

Improved approach



The Goal



Benefits of RDBMS

- Query Language
- Backup
- Consistency check
- Triggers
- Stored procedures

JDBC allows to be database implementation independent

Benefits of DMAPI

- Well known
- Vendor supported
- Existing implementations for Sgi, Linux, Solaris
- Existing backup tools
- Can be shared by any know mechanism
- Posix ACL's

Disadvantages

- RDBMS
  dificult to put file system heirarchy into tables;
  performance with growing number of entries and clients not investigated;

- DMAPI
  metadata and filenames in the same location;
  no directory tags;

What about GRID?

- The original idea to merge namespace provider with Replica catalog was discarded due to lack of need

ACLs

- NT ACL's
- POSIX ACLs (many drafts, no actual standart)
  - Posix 1003.6 draft 13
  - Posix 1003.1e draft 15

- GRID-map file
  - More or less UNIX-like - readers/writers

Conclusions

- PNFS wokrs well, but need some modifications
- NFSv3 fron-end in test phase
- RDBM-based file system simulation under performance evaluations
- DMAPI-functionality under preparation
- Merging with GRID Replica catalog not necessary
- There is no outofbox solution for ACLs

Chimera?

In Greek mythology, a fire-breathing animal with a lion's head and foreparts, a goat's middle, a dragon's rear, and a tail in the form of a snake; hence any apparent hybrid of two or more creatures.