CHIMERA – A NEW, FAST, EXTENDIBLE NAME SERVICE

Tigran Mkrtchyan
for the dCache team

The dCache Team

dCache.ORG

Responsibility, dCache

Patrick Fuhrmann Rob Kennedy

Core Team (Desy and Fermi)

Jon Bakken
Mathias de Riese
Micheal Ernst
Alex Kulyavtsev
Birgit Lewendel
Dmitri Litvintsev

Tigran Mrktchyan
Martin Radicke
Neha Sharma
Vladimir Podstavkov

Responsibility, SRM

Timur Perelmutov

External Development

Nicolo Fioretti, BARI
Abhishek Singh Rana, SDSC

Support and Help

Maarten Lithmaath, CERN
Owen Synge, RAL

Motivations

dCache.ORG

Storage systems need to handle filenames and actual data locations.

In case of regular file systems namespace are the data store merged together.

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).

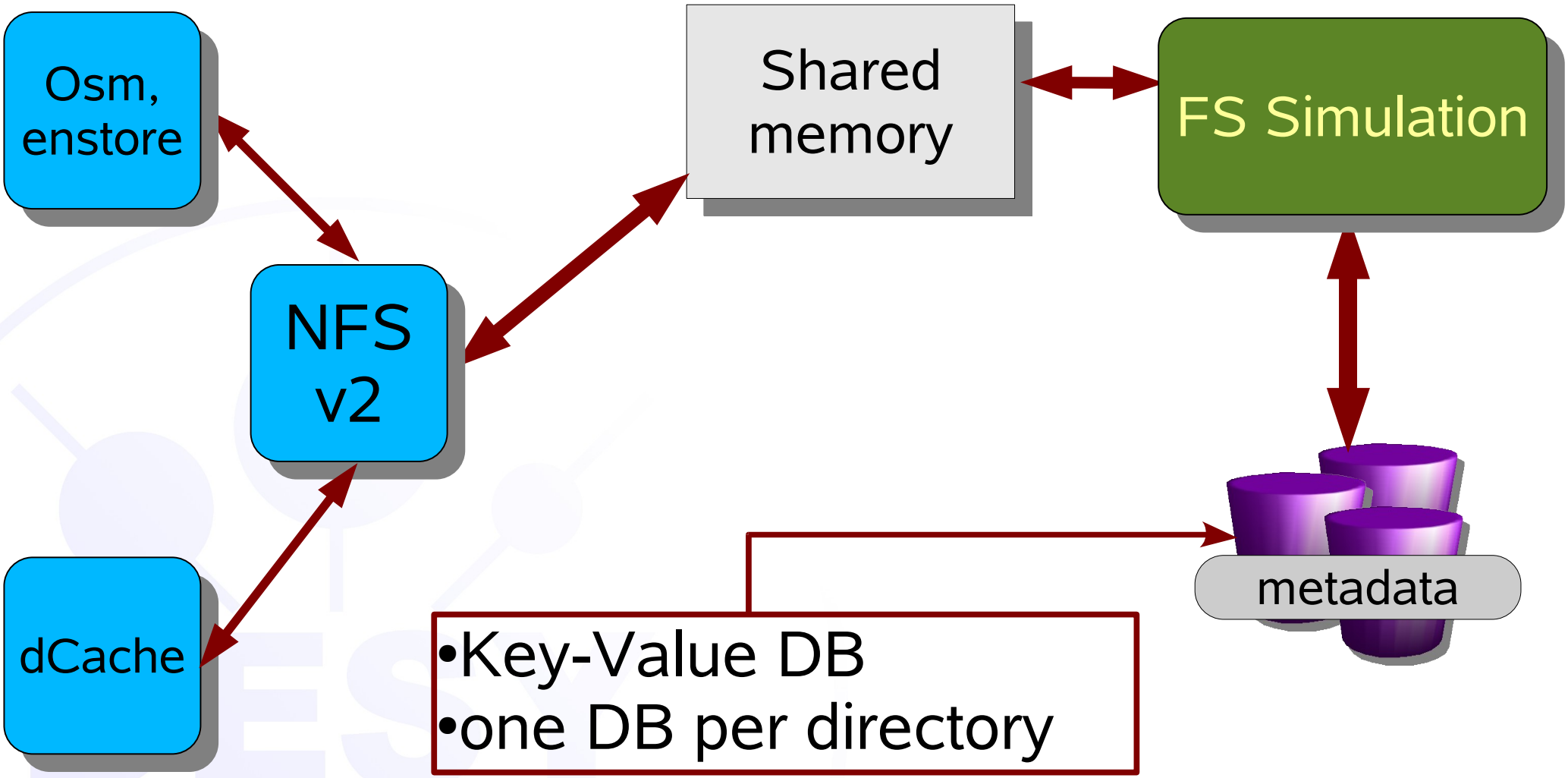
Requirements

dCache.ORG

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

Current approach

dCache.ORG



Current access

dCache.ORG

- ~1000 of clients
- More than $3*10^6$. entries (500 TB)
- 20 Hz data open/create
- 1kHz NFSops access rate

Clients

dCache.ORG

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

dCache needs

dCache.ORG

- Namespace Operations
- Storage Information
- File Locations in the pools
- Utility operations (path2id)

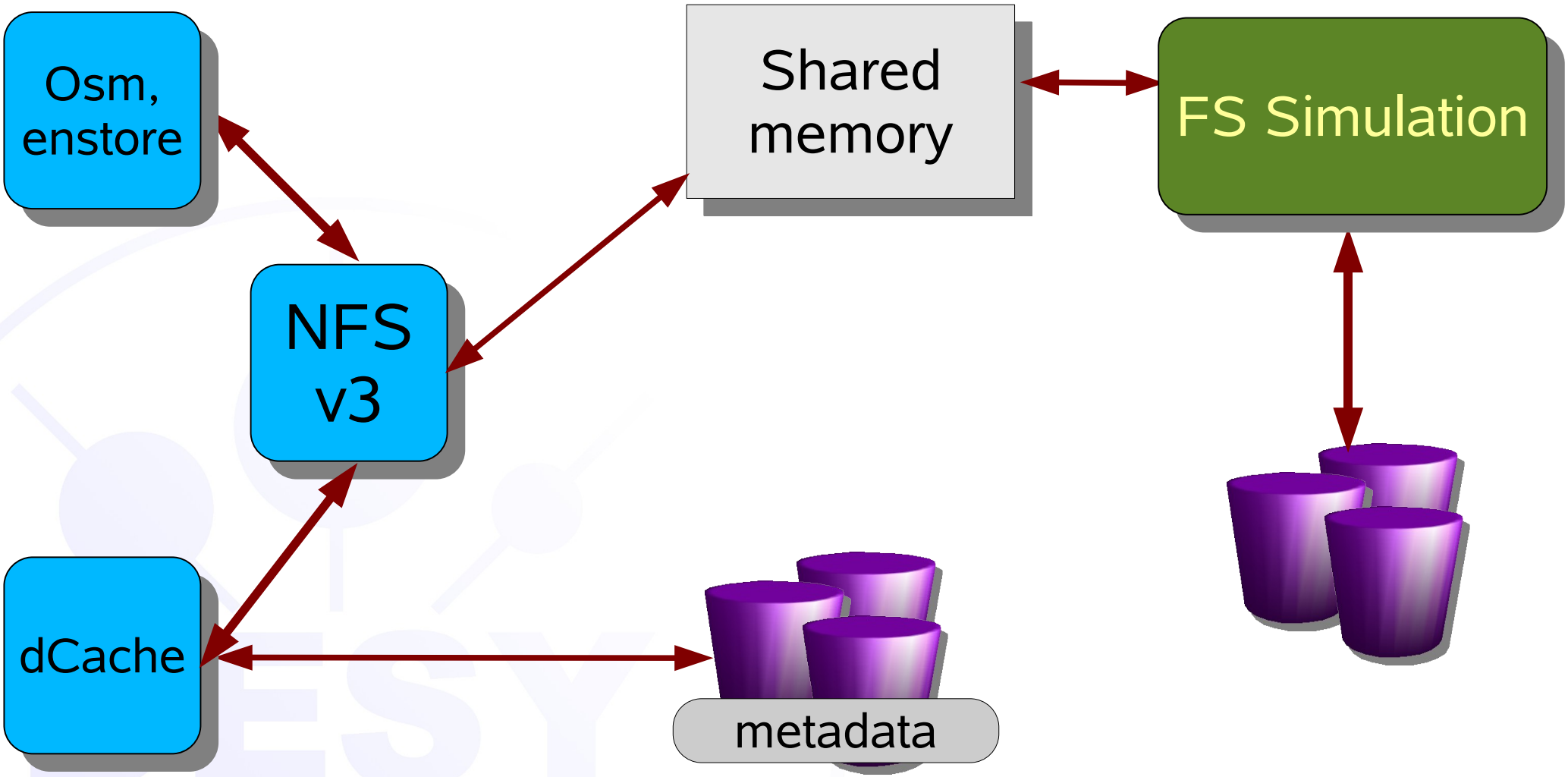
Limitation of PNFS

dCache.ORG

- Max. file size 2 Gb (NFSv2)
- 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)

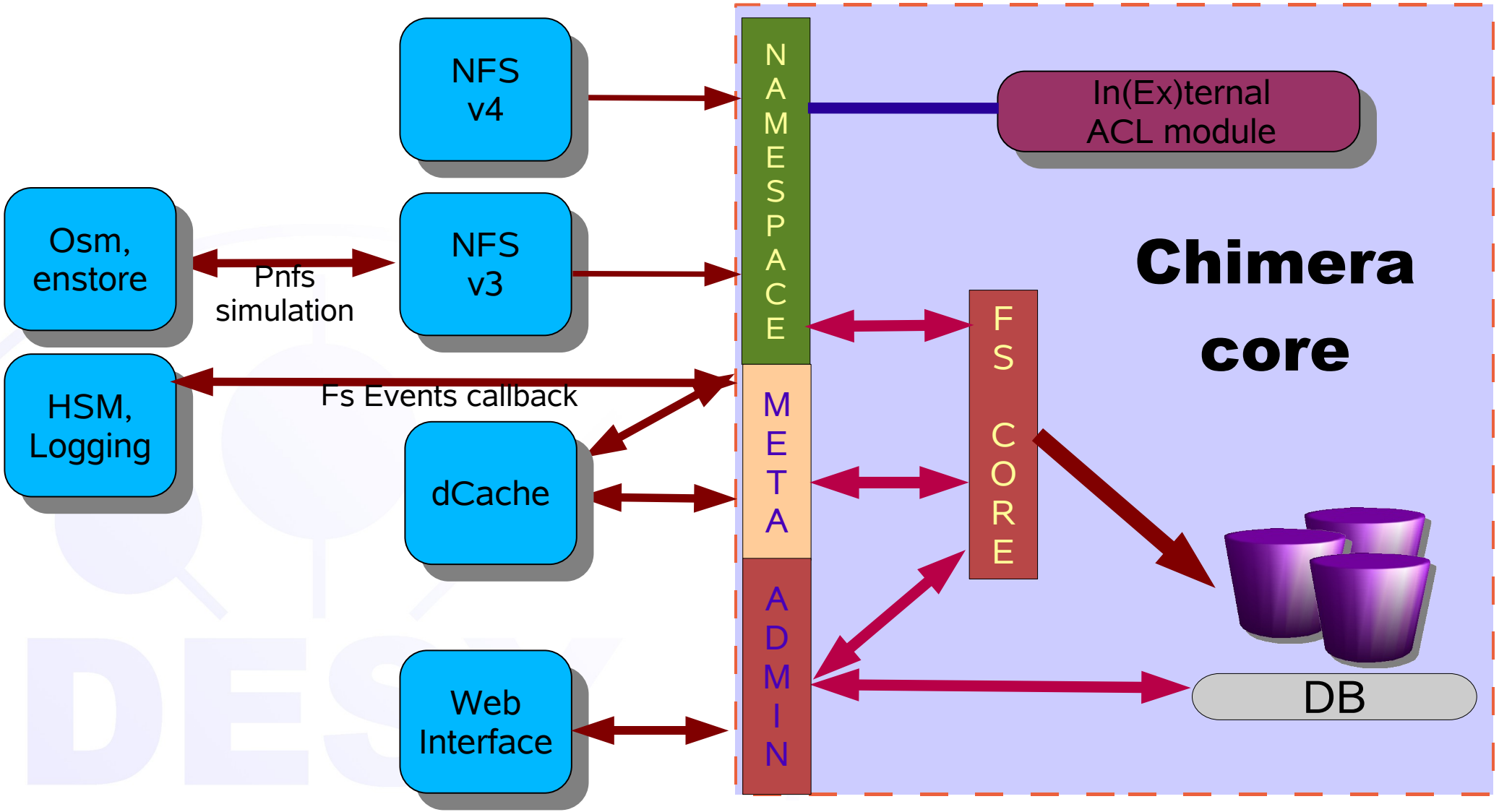
Improved approach

dCache.ORG



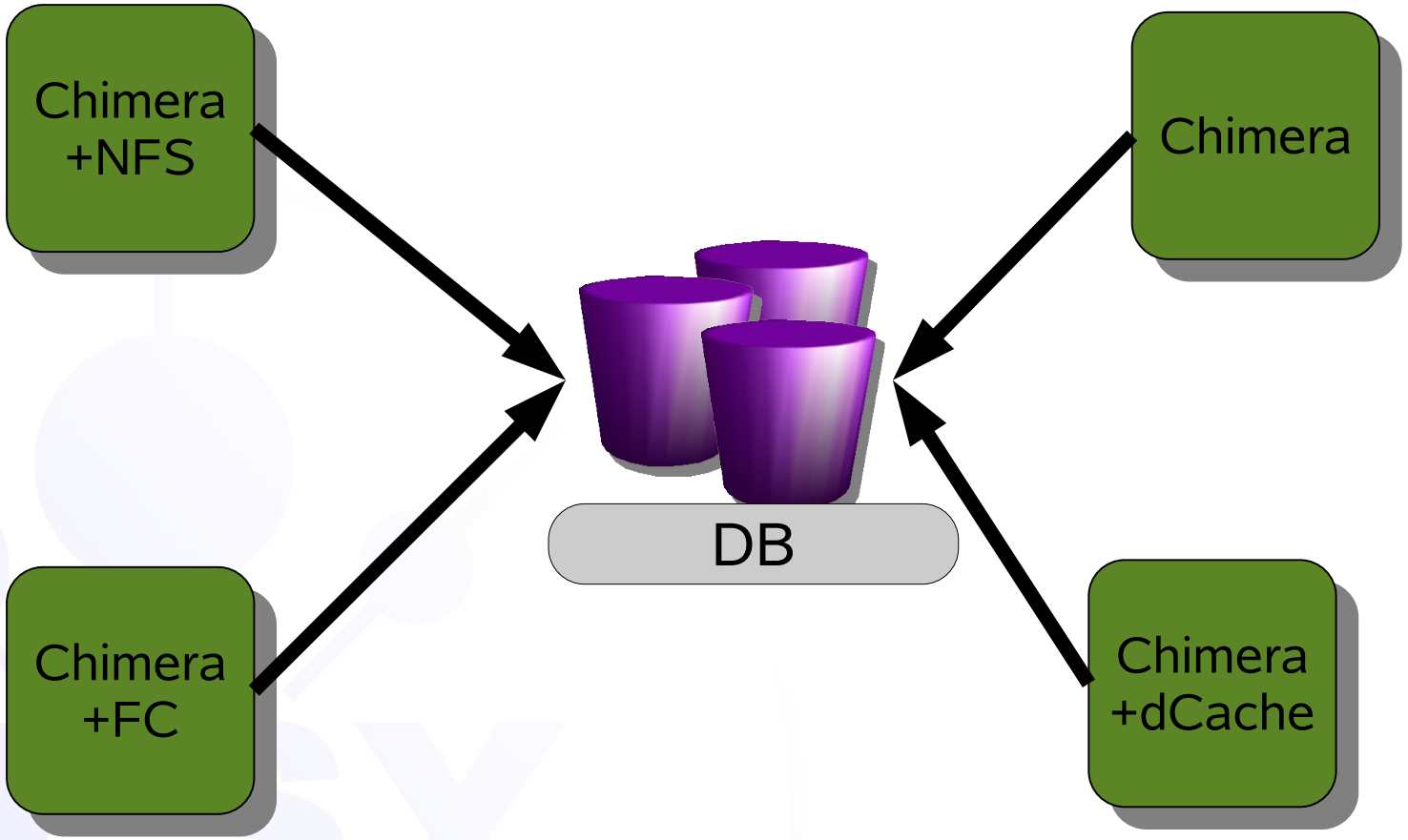
Project Goal

dCache.ORG



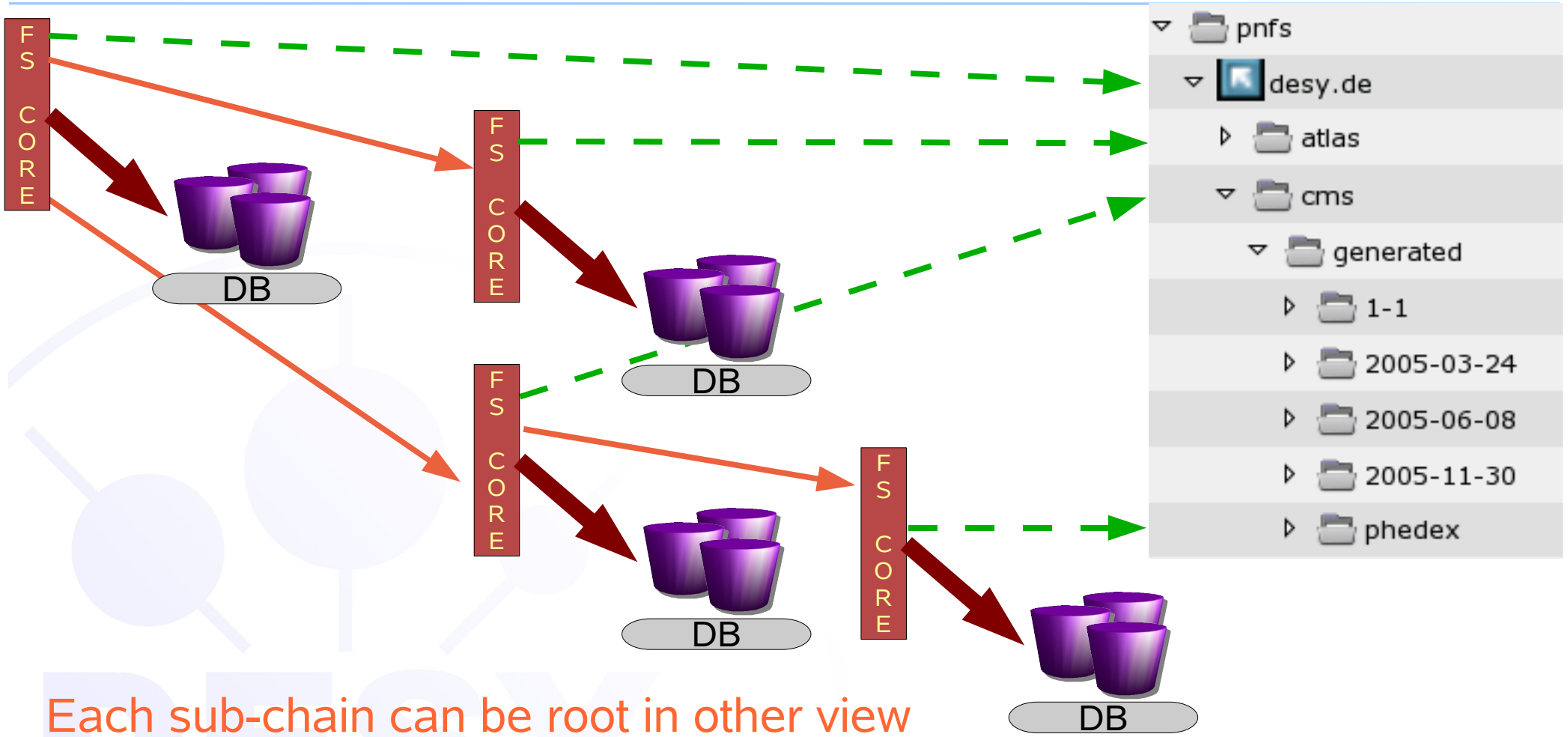
Project Goal (continue)

dCache.ORG



Filesystem chaining

dCache.ORG



Each sub-chain can be root in other view

The Benefits

dCache.ORG

- Filesystem view and metadata separated
- No NFS operations involved in API
  - stat over NFS end up with 3 ops (parent GATATTR LOOKUP, GETATTR)
- Pluggable authentication
  - unix, ACL, VOMS
- Extendable frontends
  - File browsers, replica catalogue, NFSv4

Why not a DMAPI FS?

dCache.ORG

- No out of box solutions
- Platform/Vendor dependent
- Coding still needed
- No experience

Benefits of RDBMS

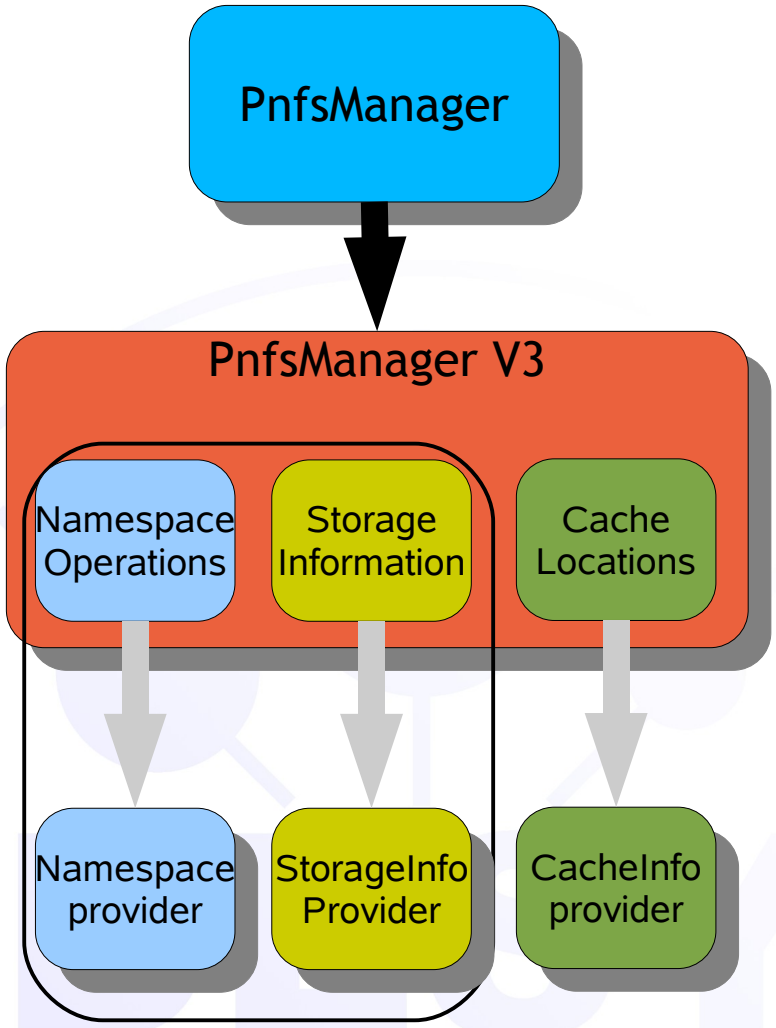
dCache.ORG

- Well known
- Query Language
  - Simple queries to get space usage, file numbers
- Backup
  - Some databases allows point in time recovery
- Consistency check
  - primary/foreign key
- Stored procedures
  - fs check

JDBC allows to be database implementation independent

dCache adaptation

dCache.ORG



- Chimera provides:
  - Namespace
  - Storage Information
- dCache provides:
  - file locations (aka “Companion”)

Current Status

dCache.ORG

- 200 file crates per second
- Tested with ORACLE, PostgreSQL, MySQL
- More than 1.500.000 files
- Full working prototype (NFSv3, dCache)
- Compatible with existing clients (dccp, osm)

ToDo

dCache.ORG

- Full scalable tests
- Migration mechanism for existing installations
- Moving from prototype to production (July 2006)
- HSM generic metadata structure
  - (need some help from other HSM experts)

Chimera?

dCache.ORG

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.