
CERN

EOS developments and AFS retirement plan

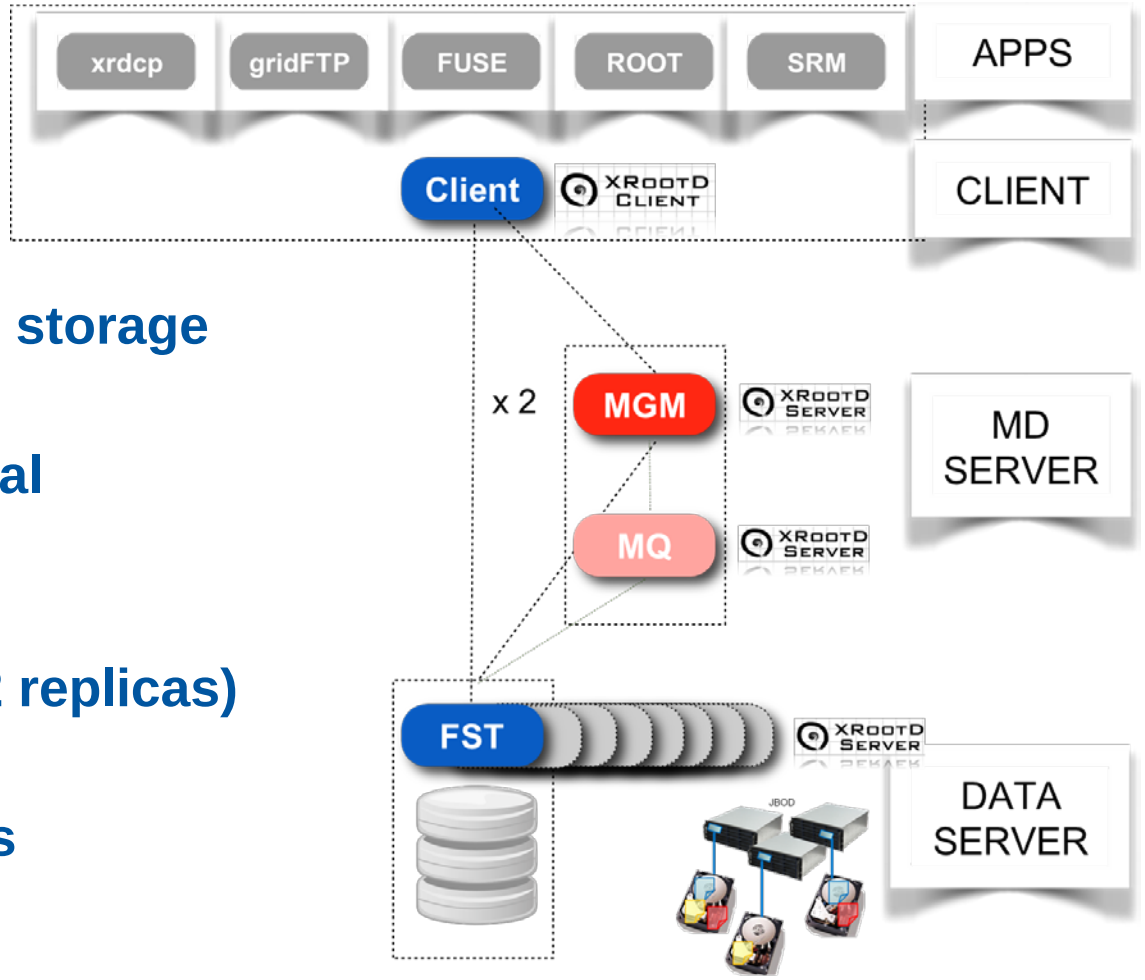
Elvin Sindrilaru - on behalf of the
EOS team and IT Storage Group

ALICE Offline Week – 02.11.2016

Outline

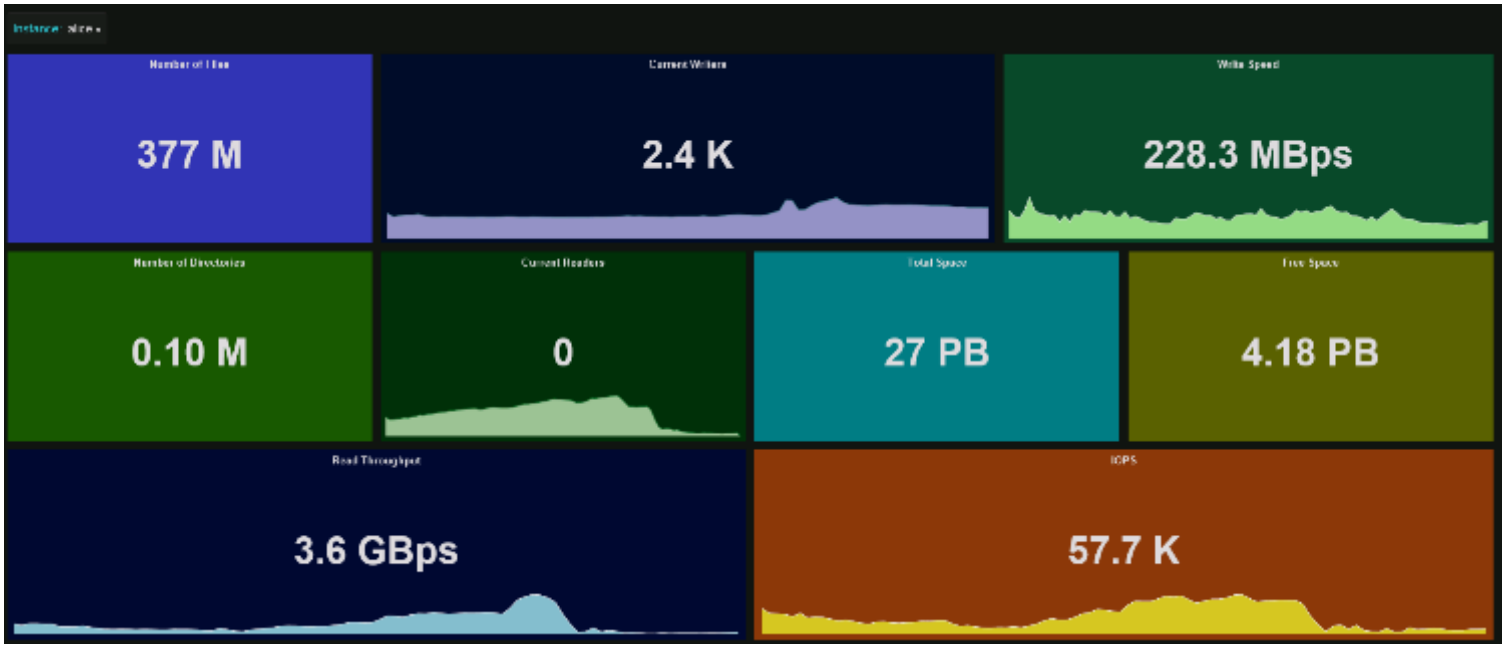
- EOS architecture
- Releases process and branches
- EOS FUSE status and improvements
- Kinetic Ethernet drives as diskserver backend
- Future namespace architecture
- AFS replacement
  - Motivation and plan
  - Impact and opportunities

EOS architecture



- Disk only physics file storage
- In memory hierarchical namespace
- File layouts (default 2 replicas)
- Physics data & others
- Low latency access

EOS ALICE instance



- No. files / no. directories ratio: **3500 : 1**
- Annual growth rate: **files ~ 61%**, **directories ~ 1%**
- Disk read / write:
  - **6.9 GB/s** avg. read
  - **330 MB/s** avg. write
- Namespace bootup time: **~ 60 min**
- Namespace size in memory: **~ 390 GB**

EOS releases and branches

- **Production** version
  - Branch: **beryl_aquamarine**
  - Release number: >= 0.3.210
  - Requires **XRootD 3.3.6**

- **Development** version (master)
  - Branch: **citrine**
  - Release number: >= 4.1.9
  - Requires **XRootD** 4.4.0

- **Feature branches** get merged into master e.g. kinetic, geo-scheduling, namespace devel. etc.

EOS FUSE status

- **Goal:** Help AFS retire gracefully

- Improved meta-data caching using the **Kernel buffer cache**
- Faster directory listing using **bulk meta-data queries**

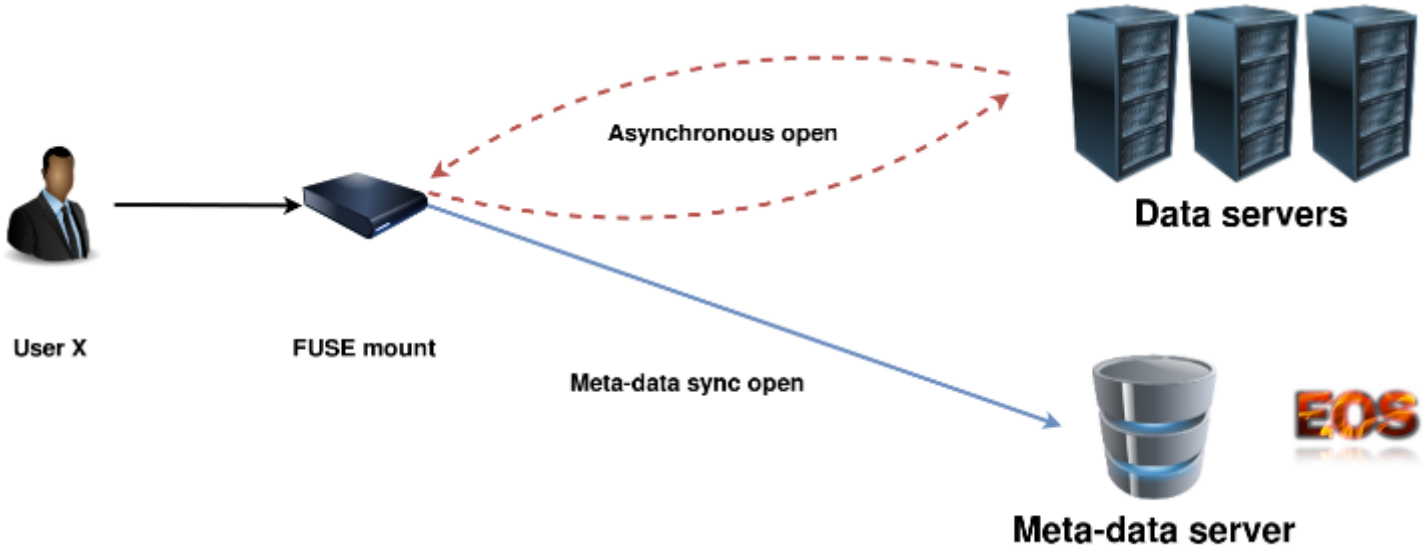
- **Multi-user mount** supporting user private **Kerberos** and **X509** authenticated connections
  - Already deployed on lxplus and lxbatch
  - Supports **user** and **session** bindings
  - Use **autofs** for better user experience

EOS FUSE multi-user mount

EOS FUSE latency optimisations

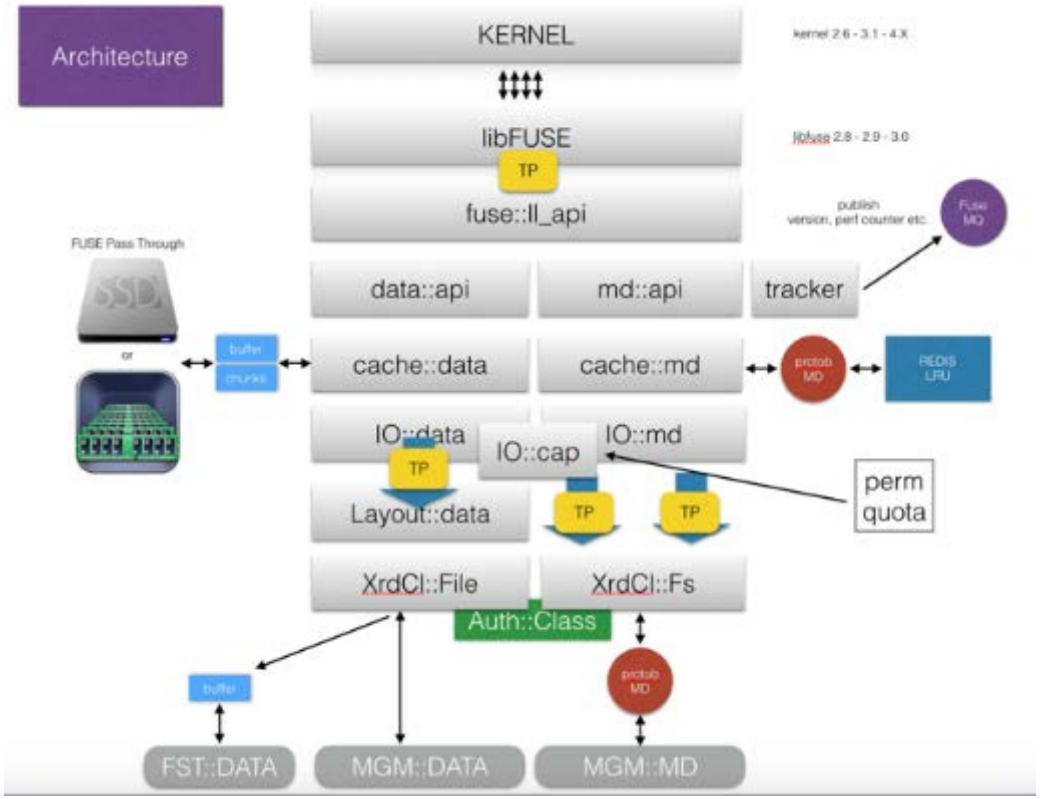
- **Write-back cache** with request aggregation

- **Lazy-open** implementation RO/RW
  - Separate meta-data and data paths
  - Data-server open happens on the first I/O operation
  - Hide latency using asynchronous open on data-server



EOS FUSE future

- **Separate data and meta-data paths** inside the FUSE module
- Use a **plugin-like model** for caching, local storage, authorization etc.
- Capitalize on the lessons learned concerning **latency optimizations**



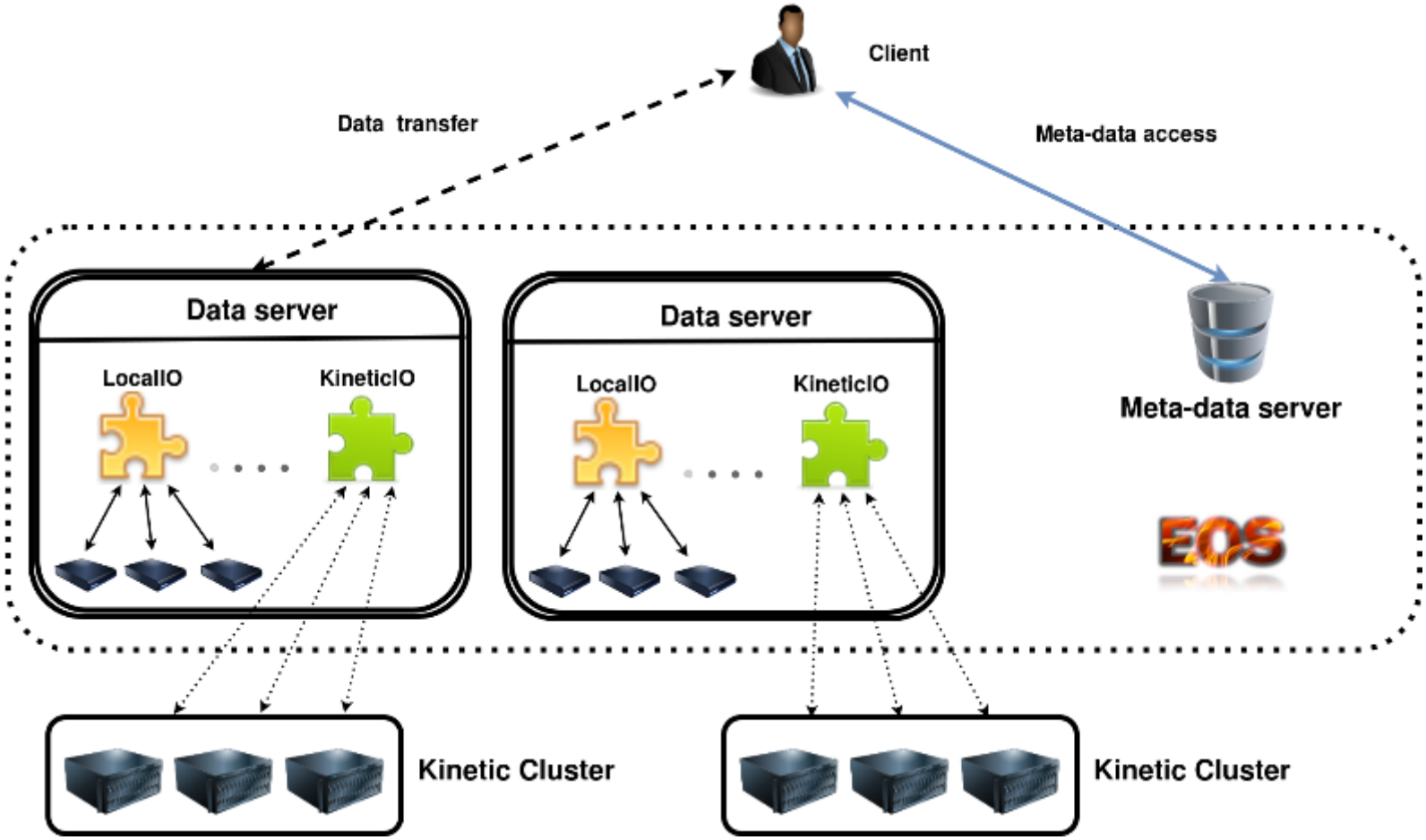
EOS Kinetic integration

- **Kinetic Open Storage Project**
  - HDDs with Ethernet interface
  - Key-value instead of block interface
  - Multi-vendor support: Seagate, Dell, Toshiba, RedHat, Cisco etc.
- **Benefits**
  - Reduced total cost of ownership (**TCO**)
  - **Robustness & scalability** – built-in replication, compression and CRC
  - Simple **abstract interface** – future proof against storage technology changes. Supported operations: put, get, delete, getnext etc.
- EOS integration done by **Paul Hermann Lensing, Seagate**

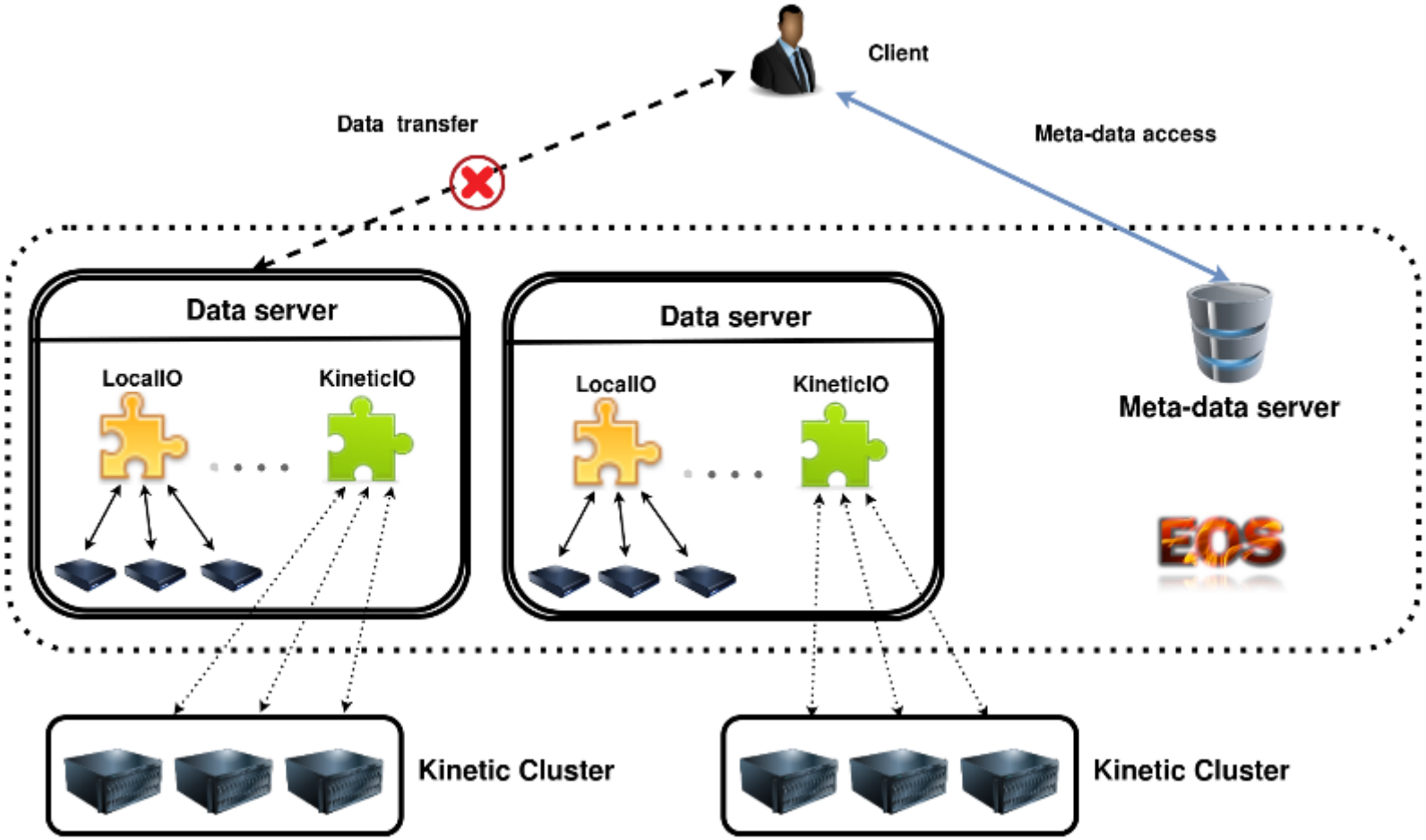
How EOS uses Kinetic?

- **Local cluster**
  - Attached to each individual **data-server**
  - Add Kinetic as a new **IO Plugin**
  - EOS is completely **agnostic** of the underlying IO access type

EOS with Kinetic local clusters



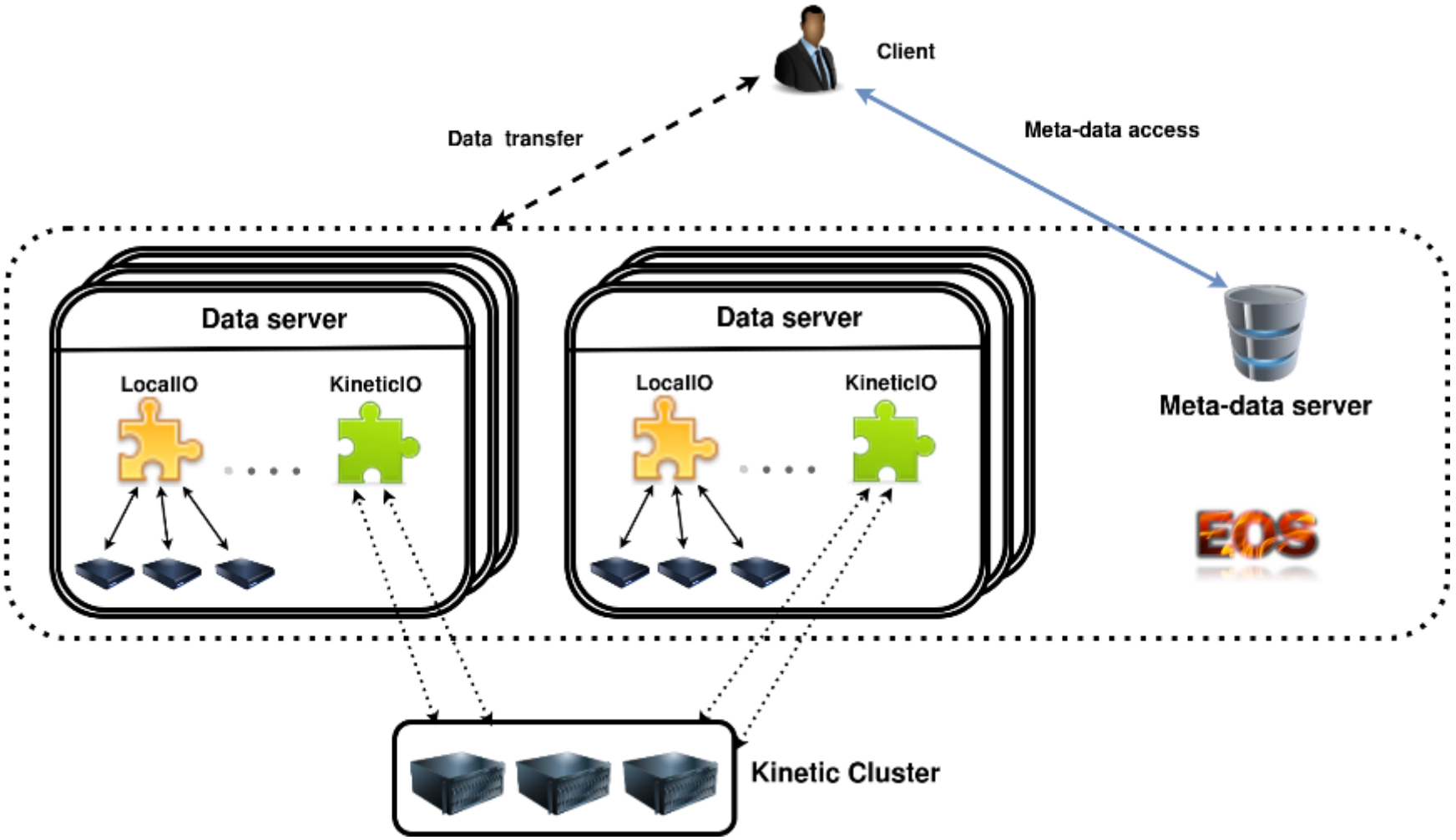
EOS with Kinetic local clusters



EOS Kinetic multi-path

- **One Kinetic cluster** shared by many data-servers
- Requires load-balancing and concurrency resolution → **Kinetic aware-scheduling**
- Fewer data-server can supply **higher storage capacity**
  - Data-server → Kinetic gateway
  - Fully utilize the combined **data-server network capacity**

EOS Kinetic multi-path

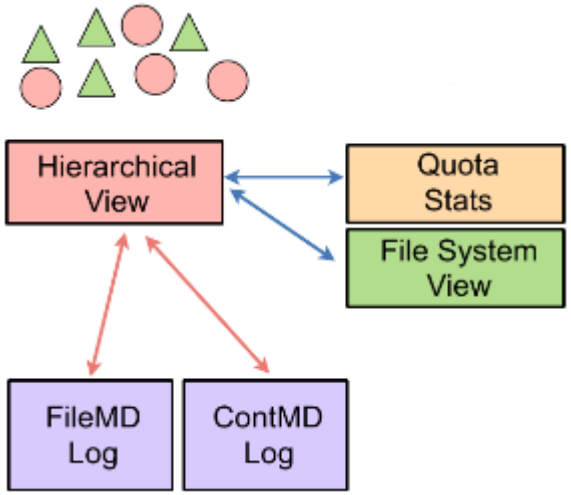


What is the EOS namespace?

- C++ library used by the EOS MGM node single-threaded
- Provides API for dealing with hierarchical collections of files

- **Filesystem elements**
  - Containers & files

- **Views**
  - Aggregate info about filesystem elem.
  - E.g QuotaView, FileSystemView etc.



- **Persistence objects**
  - Objects responsible for reading and storing filesystem elements
  - Implemented as binary change-logs

Namespace architecture pros/cons

- **Pros:**
  - Using hashes all in memory → **extremely fast**
  - Every change is logged → **low risk of data loss**
  - Views rebuilt at each boot → **high consistency**

- **Cons:**
  - For big instances it requires **a lot** of RAM
  - Booting the namespace from the change-log takes long

EOS Namespace Interface

- Prepare the setting for different namespace implementations
- Abstract a **Namespace Interface** to avoid modifying other parts of the code

- **EOS citrine 4.***

  - **Plugin manager** – able not only to dynamically load but also stack plugins if necessary

  - **libEosNsInMemory.so** – the original in-memory namespace implementation

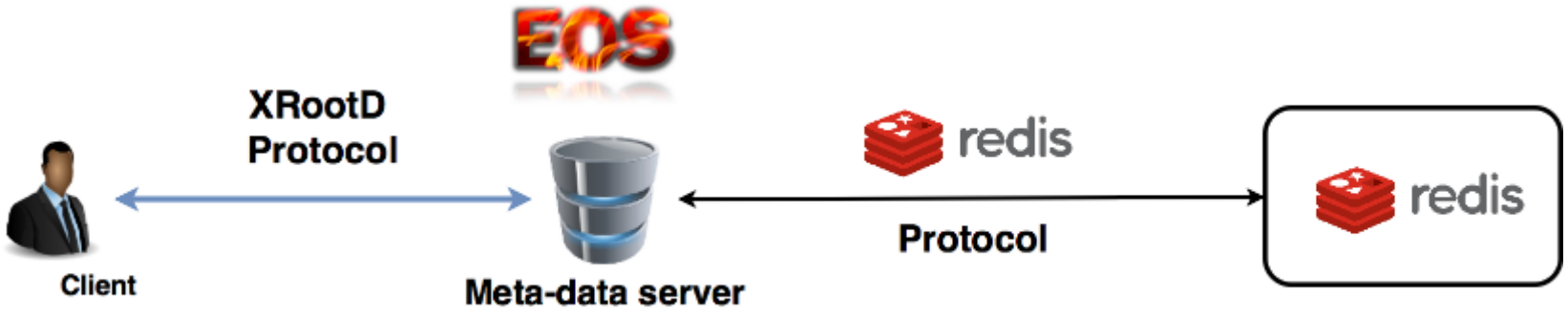
  - **libEosNsOnRados.so** – possible implementation on top of libRados

  - **libEosNsOnFilesystem.so** – possible implementation on top of a Linux filesystem

Why Redis?

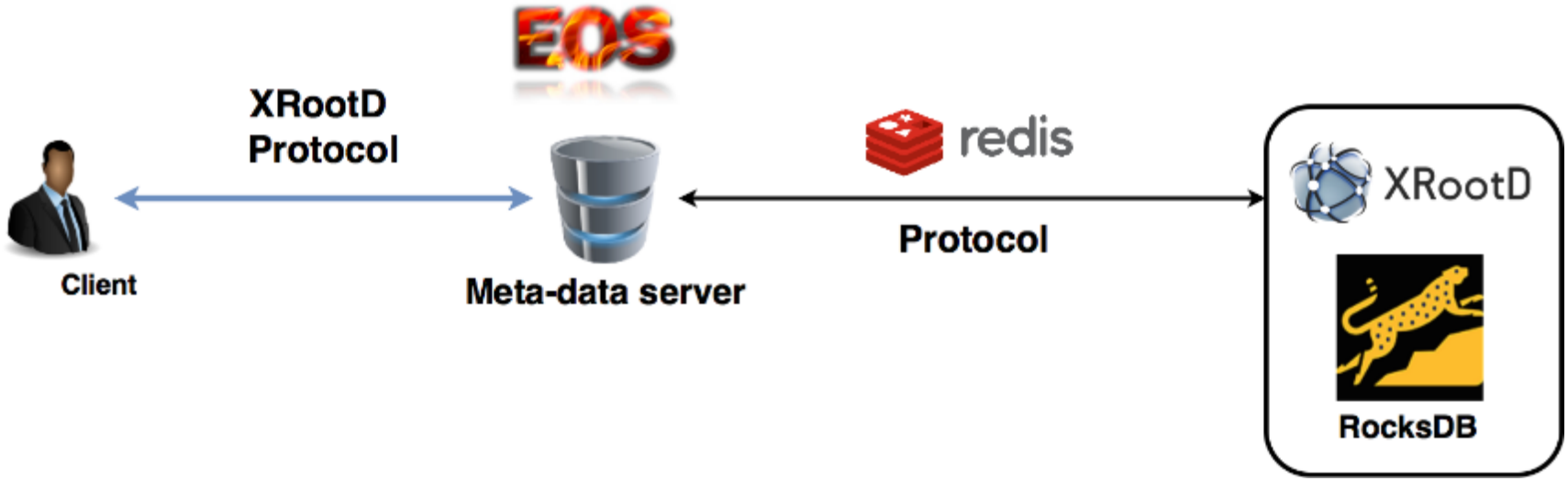
- **Redis** – in-memory **data structure store**
- Separate data from the application logic and user interface
- Supports various data structures: strings, hashes, lists, sets, sorted sets etc.

- Namespace implementation: **libEosOnRedis.so**
- **Light-weight EOS MGM** node that can easily be restarted or updated



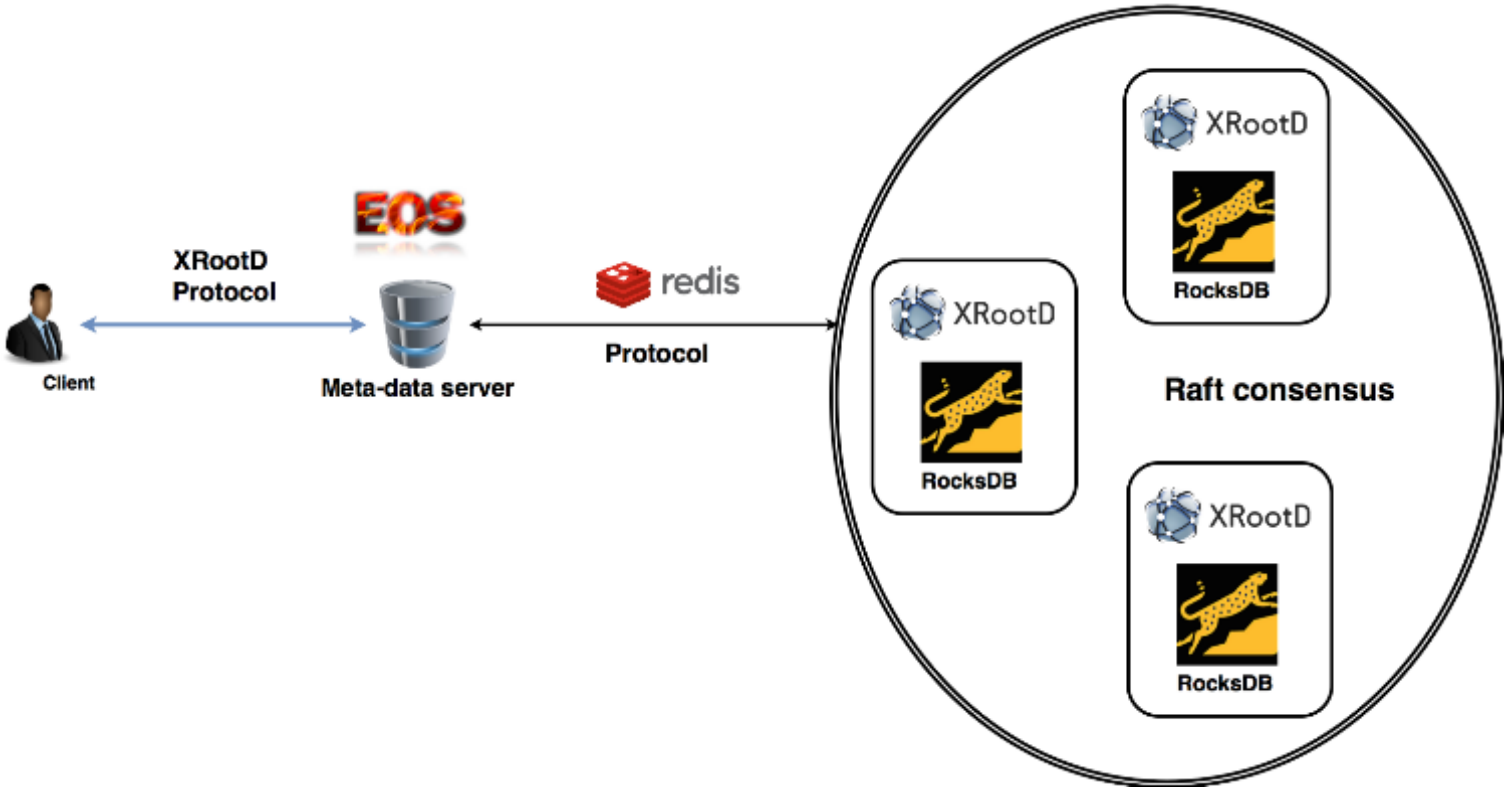
XRootD and Redis

- Replace Redis backend with XRootD
- Implemented as an XRootD **protocol plugin** – to be contributed upstream
- XRootD can use **RocksDB** as persistent key-value store



Namespace HA

- Ensure high-availability using the **Raft consensus algorithm**



AFS retirement plan

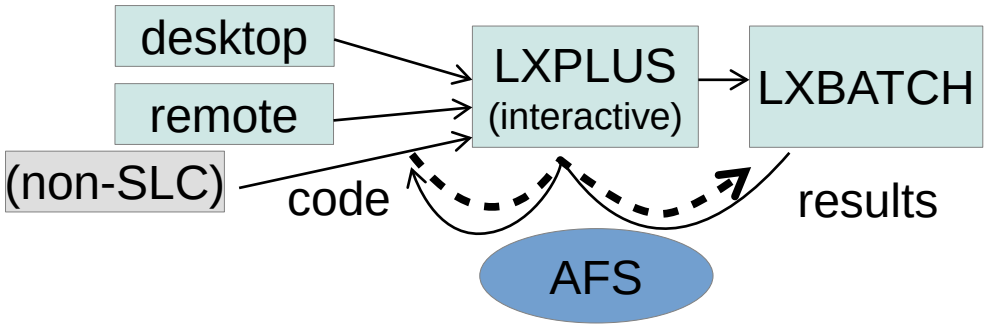
AFS status

- In use since **1990**
  - 35k users (5k active/day), 450 TB data, **3.5B files/dirs**, 3.5B accesses/day
  - Last year growth: **+80TB, +500M files**
  - Infrastructure: 50 (old=small) fileservers, 5DB servers, 1.2 FTE / 3 people

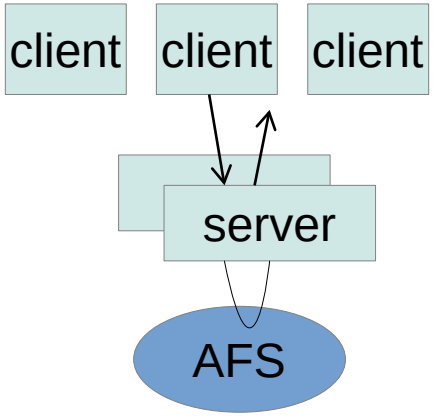
- Split into
  - **Personal $HOME** (2..10GB volumes)
    - Automatically created for every (UNIX) account
  - **Personal workspace** (10..100GB)
    - Self-service
  - **Shared project space** (1GB..10TB – vol. capped at 100GB)
    - Delegated admin powers
  - **Group shell environments**
    - “HEPIX” scripts (but apparently only remaining user …)

AFS usage at CERN (2)

- AFS is basis for local “Compute” workflow (non-grid)



- **Services:** Twiki, SVN, LXPLUS etc.



HEPIX 2016, J.Iven

Why phase out?

- OpenAFS project is in **(slow) decline**
  - Various “soft” indicators: releases, traffic, people, conferences,..
  - Pent-up changes: IPv6, DES (backward compat ... ®?)
  - Funding worries → ecosystem (2 companies, little else)
  - Ongoing client upkeep (including signed binaries on Win+Mac)

- **Technical** - widening gap
  - **Single point of failure** (per-volume) architecture vs ever-bigger machines
  - RX protocol vs “long fat pipes” - volmove, replication, backup..
  - Odd limitations (32k files in directory)

- But … project is **still “functional”** - new releases, slow changes

Where to go?

- **AFS** is **very good:**
  - Many small files – decentralized = scalable namespace
  - Rapid create/delete on single client = writeback cache
  - POSIXy enough for many applications (locks etc.)
  - Cache and read-only replicas can cope with (moderately) high loads
  - Secure (enough) for access from untrusted clients and remote
  - Multiplatform and free

- No single ready-made drop-in replacement …

=>> Need to go over use cases **one-by-one**

Where to go?

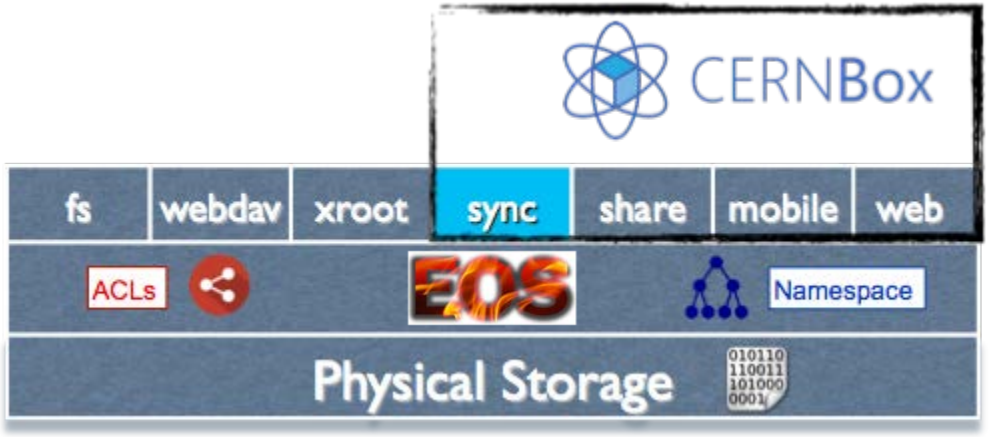
- **CERN Migration targets**
  - **CERNBOX** – human-generated content
  - **EOS-FUSE** – filesystem access
  - **EOS** – live data
  - **CVMFS** – (massive) software distribution
  - **CASTOR** – archive + dead data
  - **Delete (?)** – machine-generated junk & obsolete
  - **Special cases:** cluster-level filesystems (NFS, CEPHFS, HDFS)

- **Review:** Some use cases {c|sh}ould change: (after 26 years...)
  - Interactive analysis: **SWAN**
  - Temp files : use local disk or memory
  - Browsers, Mail: stay local
  - “defined” OS+compiler: VM / containers

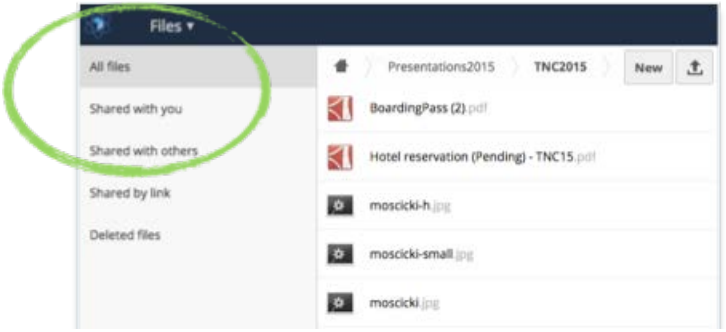
Why EOS?

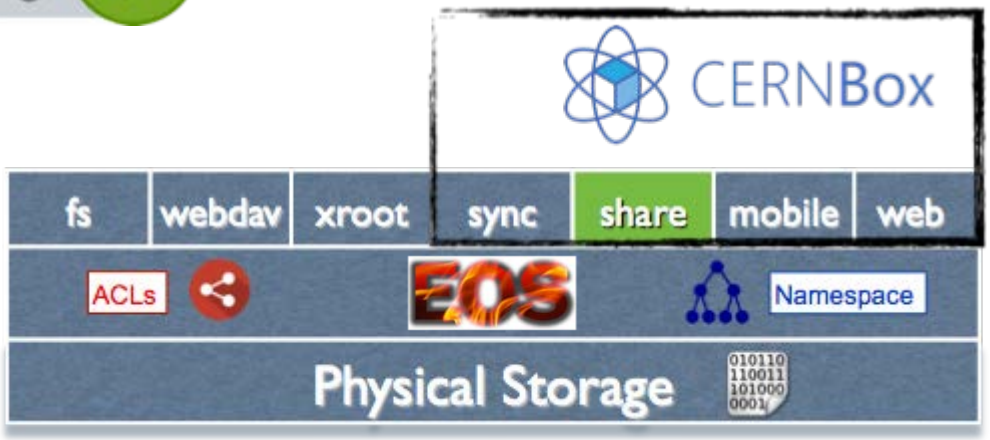
- Strategic:
  - EOS already holds most physics data at CERN
  - Building block for several new services
    - CERNBOX – very popular
    - SWAN – huge interest
    - Disk subsystem of future tape archive (CTA)
  - **EOS-FUSE** (single-user) is widely used in experiments
    - Despite not really being encouraged ...
- Full control over implementation
  - Flexibility
  - Non-standard – can extend at will

Access method: Sync



Access method: Share

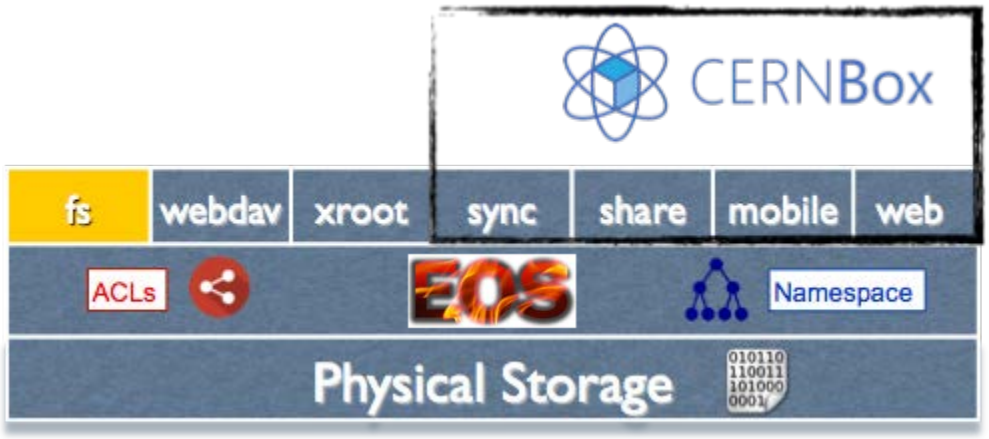




Access method: Web & Mobile

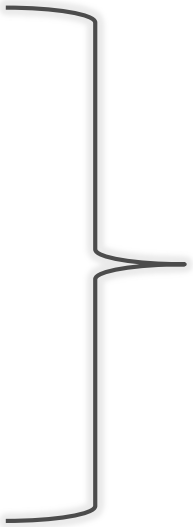
Access method: FUSE


```
[lmascett@lxplus2015 ~]#
[lmascett@lxplus2015 ~]# df -H -t fuse
Filesystem      Size  Used Avail Use% Mounted on
eosuser         506T   70T  437T  14% /eos/user
eosatlas         36P   17P   20P  45% /eos/atlas
eosalice         20P   11P  8.5P  57% /eos/alice
eoscms           28P   14P   15P  49% /eos/cms
eoslhcb          13P  7.6P  4.6P  63% /eos/lhcb
eospublic        16P  5.8P   11P  36% /eos/public
[lmascett@lxplus2015 ~]#
[lmascett@lxplus2015 ~]# ls -lc /eos/user/l/lmascett/
total 6644
drwx------. 1 lmascett c3       5 Dec 10 15:58 CERN
drwx------. 1 lmascett c3       0 Jan 26 18:18 debug
drwx------. 1 lmascett c3       0 Dec 11 09:43 download
drwx------. 1 lmascett c3       0 Oct 31 18:24 pdf
drwx------. 1 lmascett c3       1 Dec 11 09:44 personal
drwx------. 1 lmascett c3       8 Dec 10 12:11 pictures
```



Looks promising but …

- $HOME directories
- Multi-role LXPLUS:
  - External SSH access gateway
  - LSF submission machine
  - “default” SLC6/CC7 validated environment
  - Analysis compile, debug, run
  - 'acrontab' recipient, mail reading, browsing..
    → disentangle from “AFS”
- BATCH: LSF → CONDOR migration
  - Opportunity for better efficiency
  - (CONDOR will have AFS access)



Future Computing@CERN

- Account: split “UNIX” account from “AFS” account
  - Home directory is optional
- WEBAFS → **WEBEOS**: same setup. Try it out!
- AFS-the-free-backup: make people aware - we have tapes!

Phase out ~ timeline

1998

- “look at alternatives” …

2014

- “rather bad state”

2015

- **NOISE & Discovery**
  - Initial communication
  - Establish experiment communication
  - Use case discovery

Phase out ~ timeline

2016

- **EASY**
  - Web -> EOSFUSE
  - Projects -> EOSFUSE
  - Software -> CVMFS + Obvious cleanup

2017

- **HARDER**
  - Dead experiments – DPHEP?
  - More elaborate use-cases
  - Home directories (?)

2018-19

- **HARD**
  - As in “die-hard” …
  - No “LHC-stop-threat” in 2019

End of LS2

- **~ The end ~**

Summary

- **EOS FUSE**
  - Strategic development to satisfy as many use-cases as possible
- **EOS Namespace**
  - Meet scalability and growth demands
  - Prototype on top of Redis/XRootD and HA using Raft
- **AFS phase out** slowly starting
  - But not in “panic” mode
  - Attractive new tools & services – use them
  - Rethink use-cases, no 1-to-1 mapping

https://twiki.cern.ch/twiki/bin/viewauth/IT/AfsPhaseout

www.cern.ch