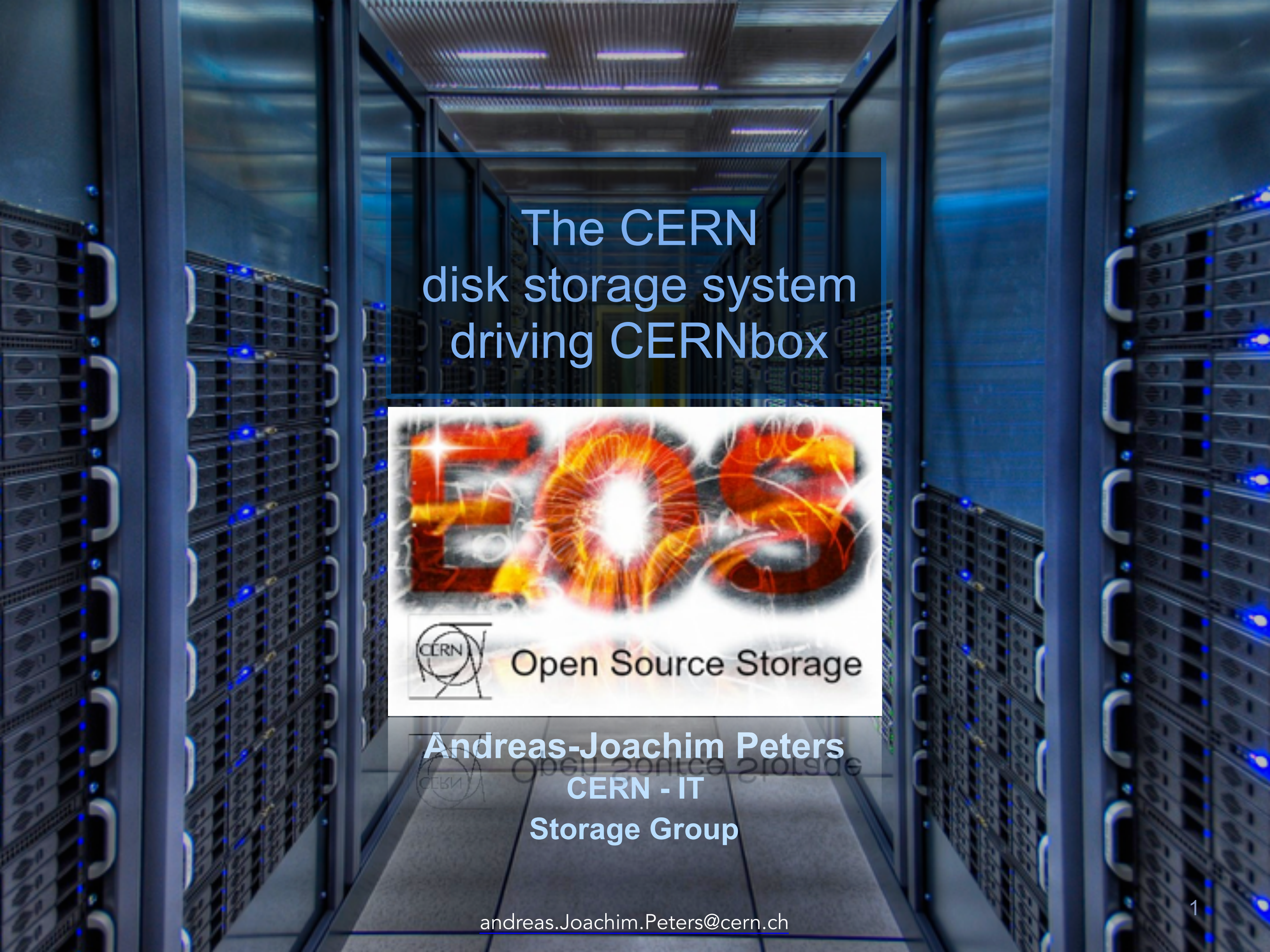


The CERN disk storage system driving CERNbox

Open Source Storage

Andreas-Joachim Peters

CERN - IT

Storage Group

andreas.Joachim.Peters@cern.ch

Contents

- What is EOS? What is XRootD?
- Features & Releases
- File Synchronisation Extensions
- Current Developments and Challenges

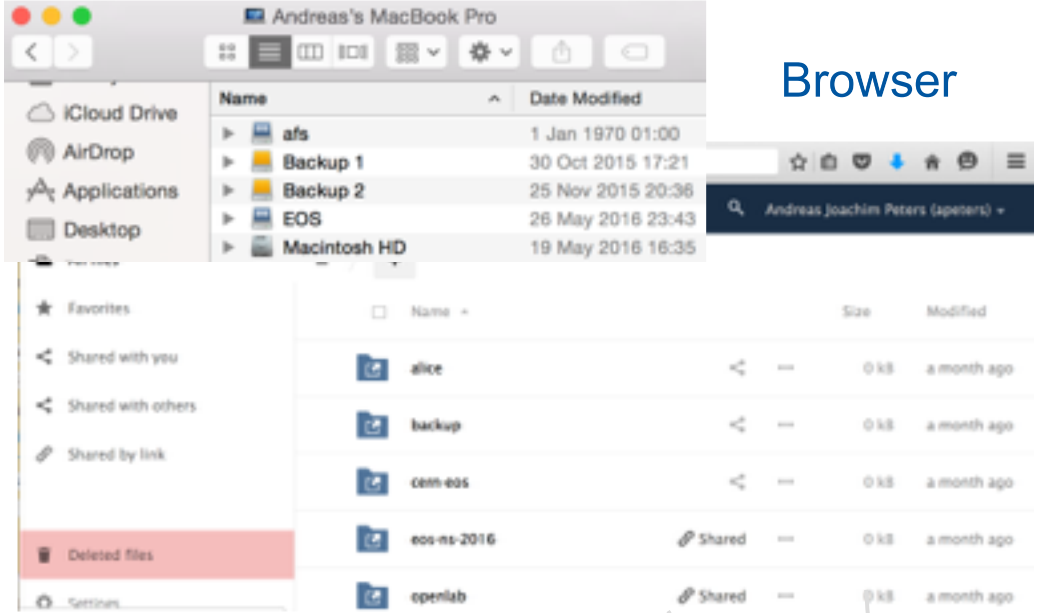
What is EOS ?

- Free storage software [GPL v3 License]
  developed since 2010 at CERN
  - optimised for **large installation**
    today 160 PB – 44k hard disks
  - **multi-protocol** access
    - remote disk(mounted) , HTTP(S), WebDAV …
  - **secure access** – strong authentication
    - kerberos & certificate authentication
  - **multi-user management**
    - fine-grained access control, quota system

How is it used?

Virtual Drive / Desktop

Browser



Andreas's MacBook Pro

- iCloud Drive
- AirDrop
- Applications
- Desktop

Name	Date Modified
afs	1 Jan 1970 01:00
Backup 1	30 Oct 2015 17:21
Backup 2	25 Nov 2015 20:36
EOS	26 May 2016 23:43
Macintosh HD	19 May 2016 16:35

Andreas Joachim Peters (apeters)

- Favorites
- Shared with you
- Shared with others
- Shared by link
- Deleted files
- Settings

Name		Size	Modified
alice		0 kB	a month ago
backup		0 kB	a month ago
cern-eos		0 kB	a month ago
eos-ns-2016	Shared	0 kB	a month ago
openlab	Shared	0 kB	a month ago

Batch Jobs / Cloud Resources

Applications

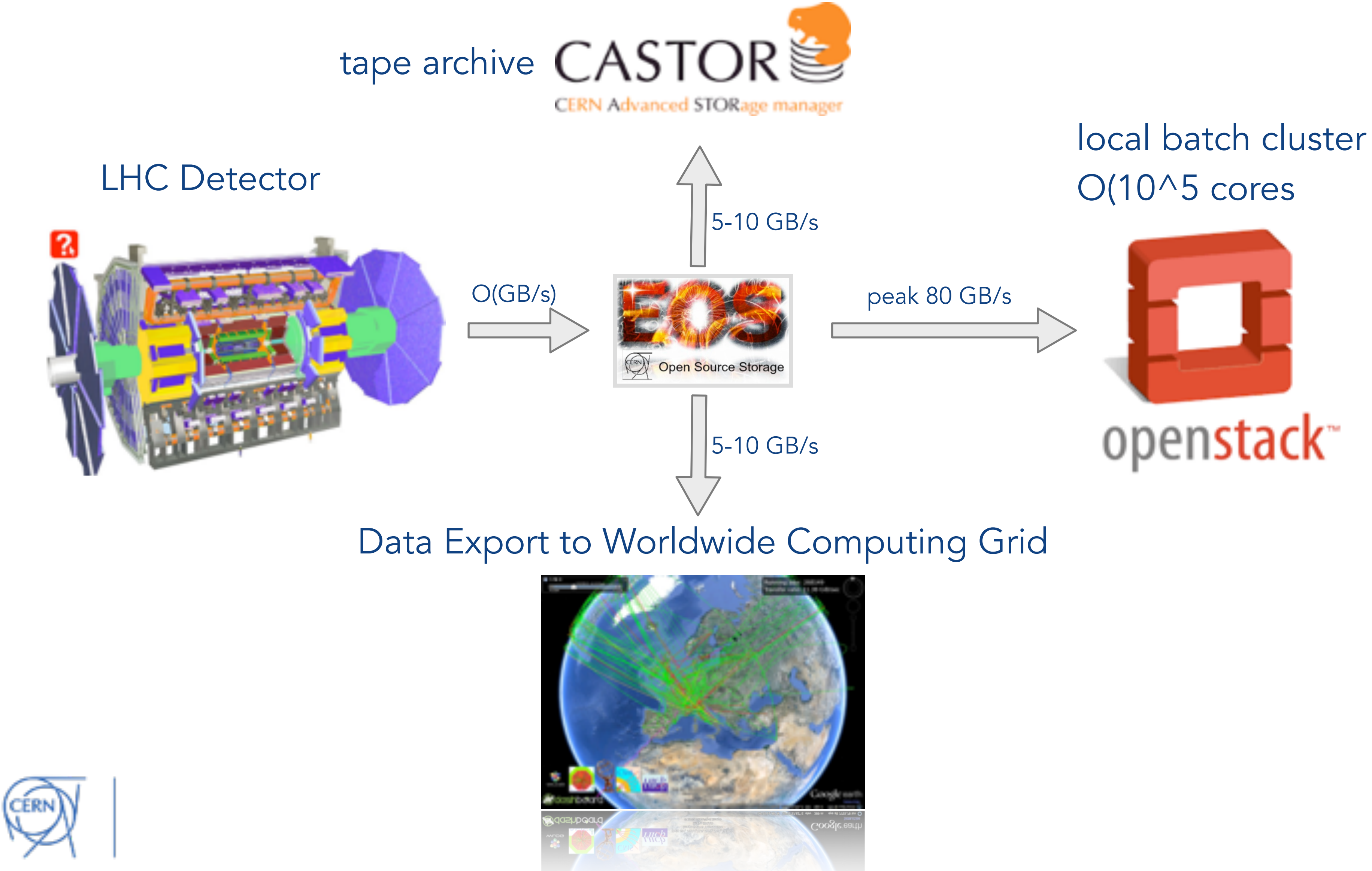
EOS Protocol

HTTP Protocol

XRootdD Protocol

```
daemon    1627  0.5  0.7 1491028 521244 ?     Sl   J                    tc/xrd.cf.mgm -m -l /var/log/eos/xrdlog.mgm -b -Rdaemon
daemon    1903  0.0  0.0 225612 18068 ?       S    J                    tc/xrd.cf.mgm -m -l /var/log/eos/xrdlog.mgm -b -Rdaemon
daemon    9989  0.2  0.3 687680 203712 ?      Sl   J                    tc/xrd.cf.fst -l /var/log/eos/xrdlog.fst -b -Rdaemon
daemon   10028  0.1  0.0 278960 38564 ?       S    J                    tc/xrd.cf.fst -l /var/log/eos/xrdlog.fst -b -Rdaemon
daemon   30402  0.3  0.1 235840 97732 ?       Sl                        :/xrd.cf.mq -l /var/log/eos/xrdlog.mq -b -Rdaemon
```

How is it used?



tape archive

LHC Detector

O(GB/s)

5-10 GB/s

peak 80 GB/s

local batch cluster

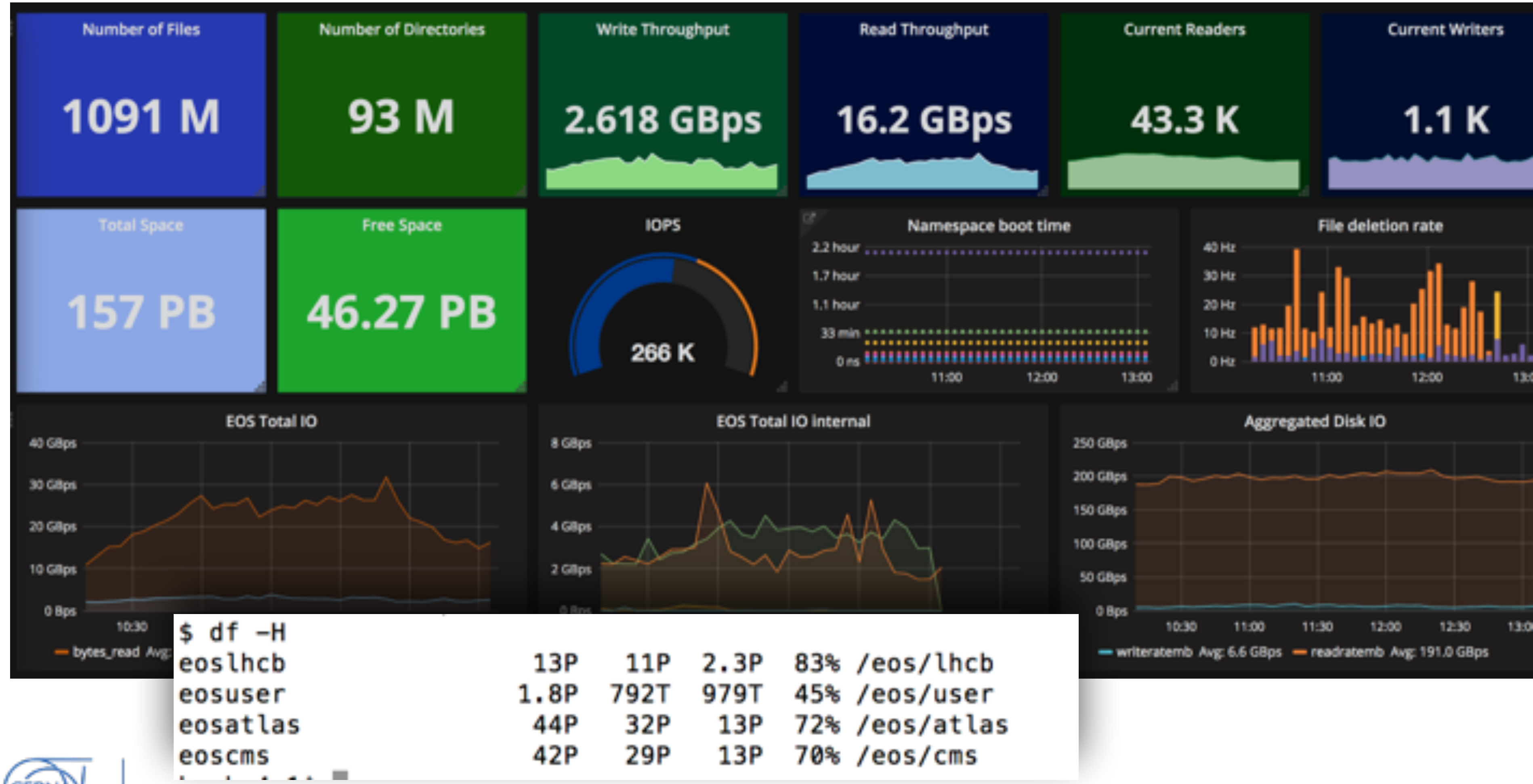
O(10^5 cores

5-10 GB/s

Data Export to Worldwide Computing Grid

EOS Service at CERN

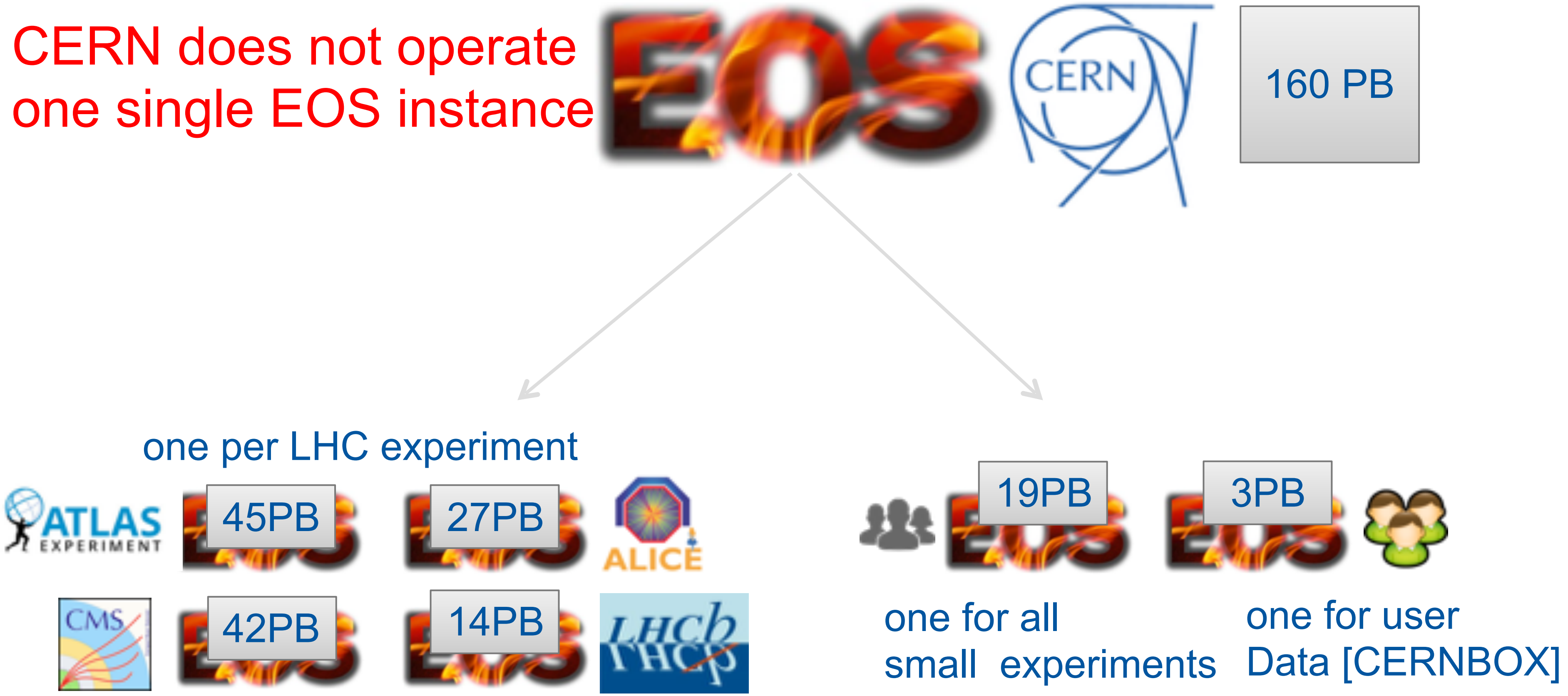
~1 - 40.000 concurrent clients per instance



```
$ df -H
eoslhcb                     13P    11P   2.3P   83% /eos/lhcb
eosuser                    1.8P   792T   979T   45% /eos/user
eosatlas                    44P    32P    13P   72% /eos/atlas
eoscms                      42P    29P    13P   70% /eos/cms
```

EOS Service at CERN

CERN does not operate one single EOS instance



160 PB

one per LHC experiment

45PB

27PB

42PB

14PB

19PB

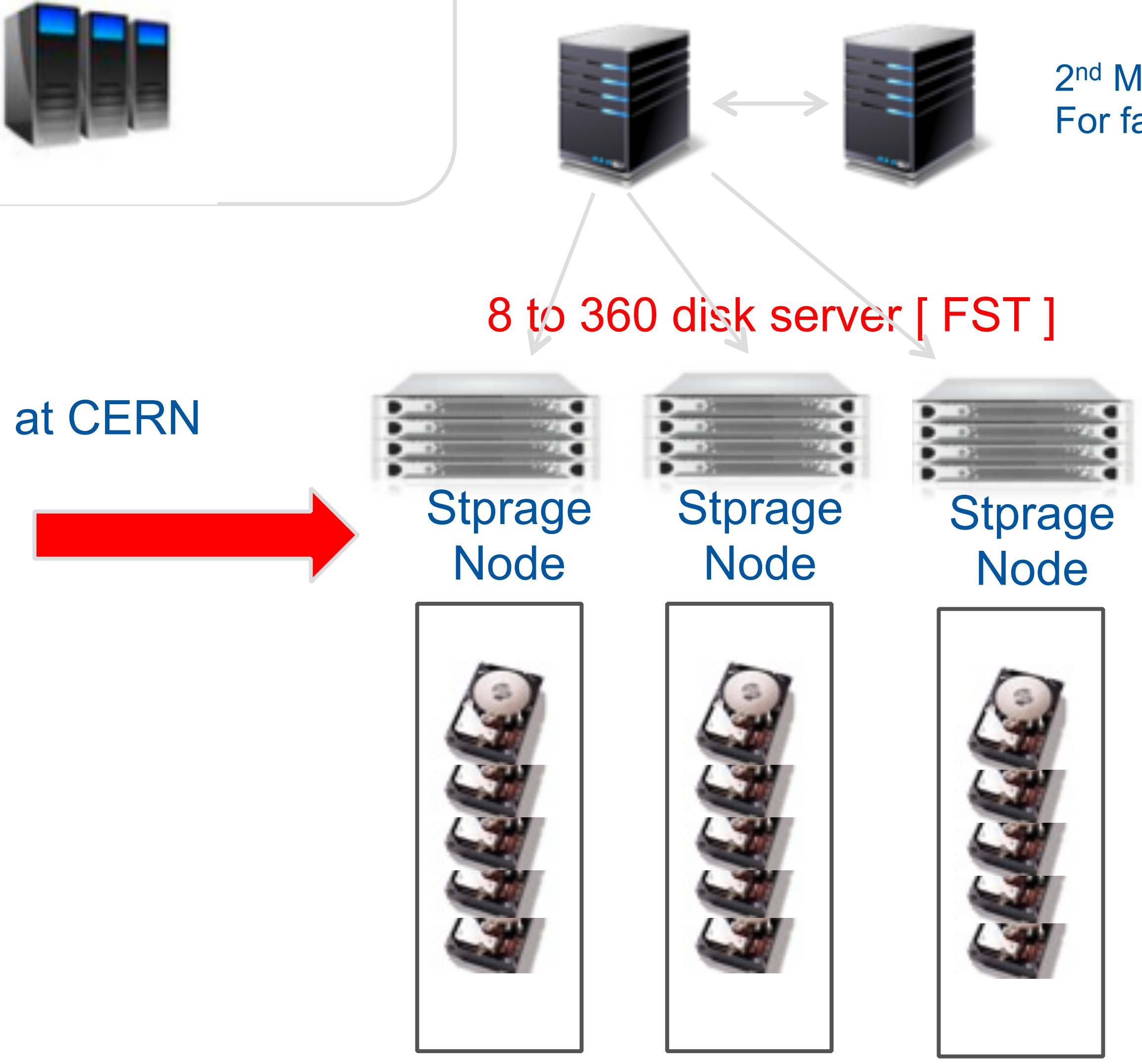
3PB

one for all small experiments

one for user Data [CERNBOX]

Six separate EOS production instances (+ others)

EOS Instance



2 meta data server [MGM/MQ]

2nd MGM/MQ passive
For failover

8 to 360 disk server [FST]

One EOS instance at CERN

Stprage Node

Stprage Node

Stprage Node

24-96 disks

XRootD - core framework

like Apache **httpd** is a framework to implement web services, XRootD is the framework for EOS

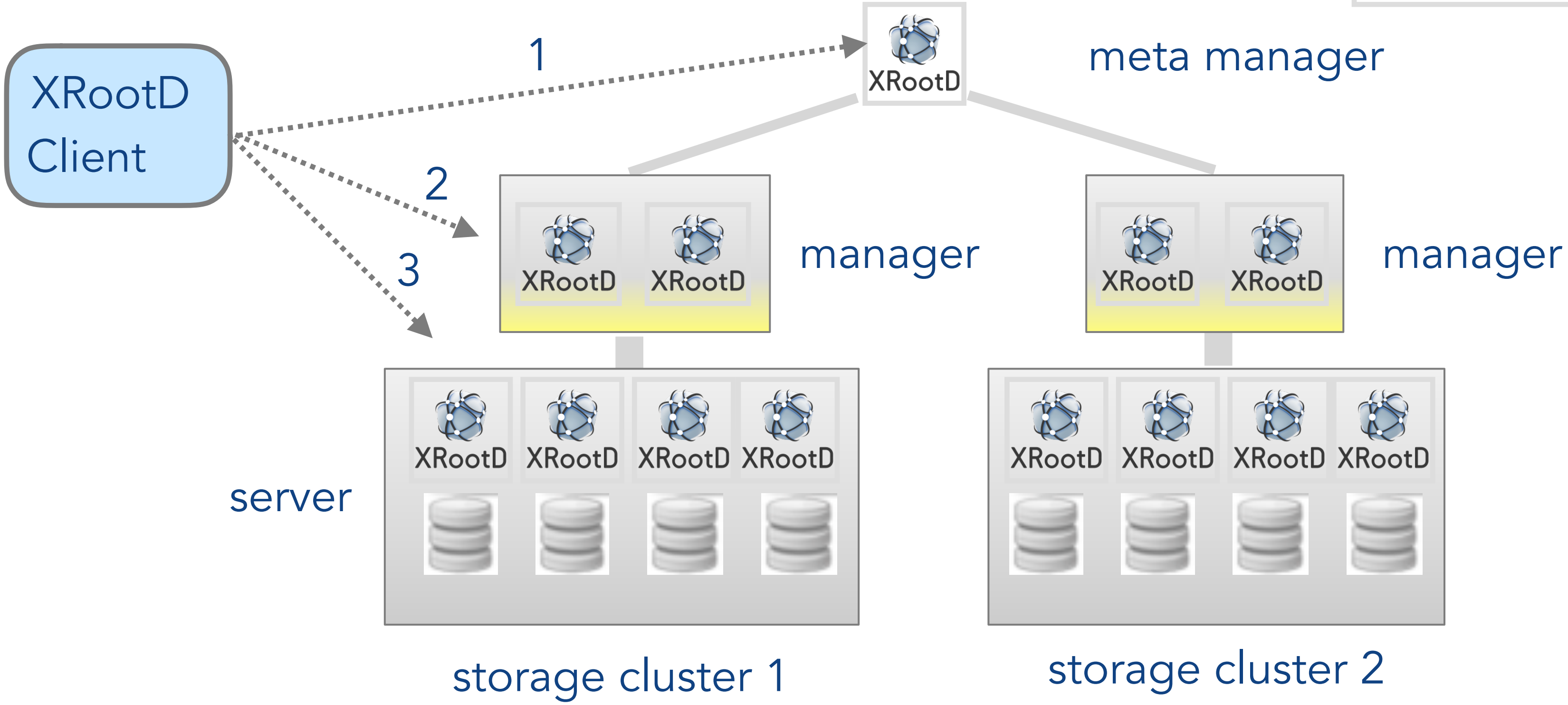
XRootD

- XRootD is a **multithreaded** C++ **client/ server framework** providing a remote access protocol
  - authentication, meta-data, data interfaces as plugins
- XRootD protocol designed for **efficient remote file access** (unlike HTTP) in LAN/WAN
  - synchronous/asynchronous IO interfaces
  - latency optimisations like vector reads
  - checksums
  - storage clustering with hierarchical redirection
  - third party copy

65kHz sync requests
1.5M Hz async requests
default TP 2k threads
ok 40k clients

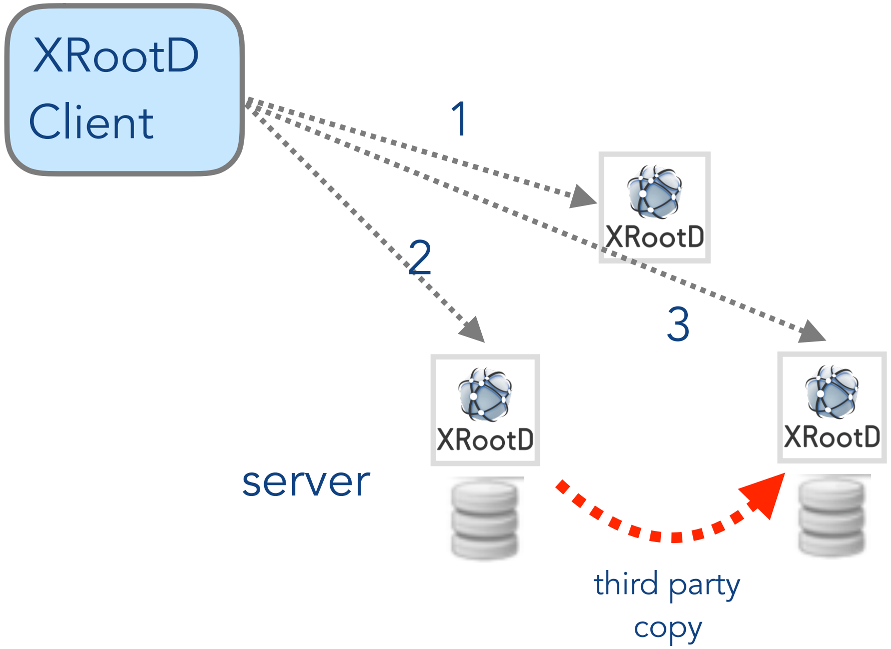
http://xrootd.org

redirection model



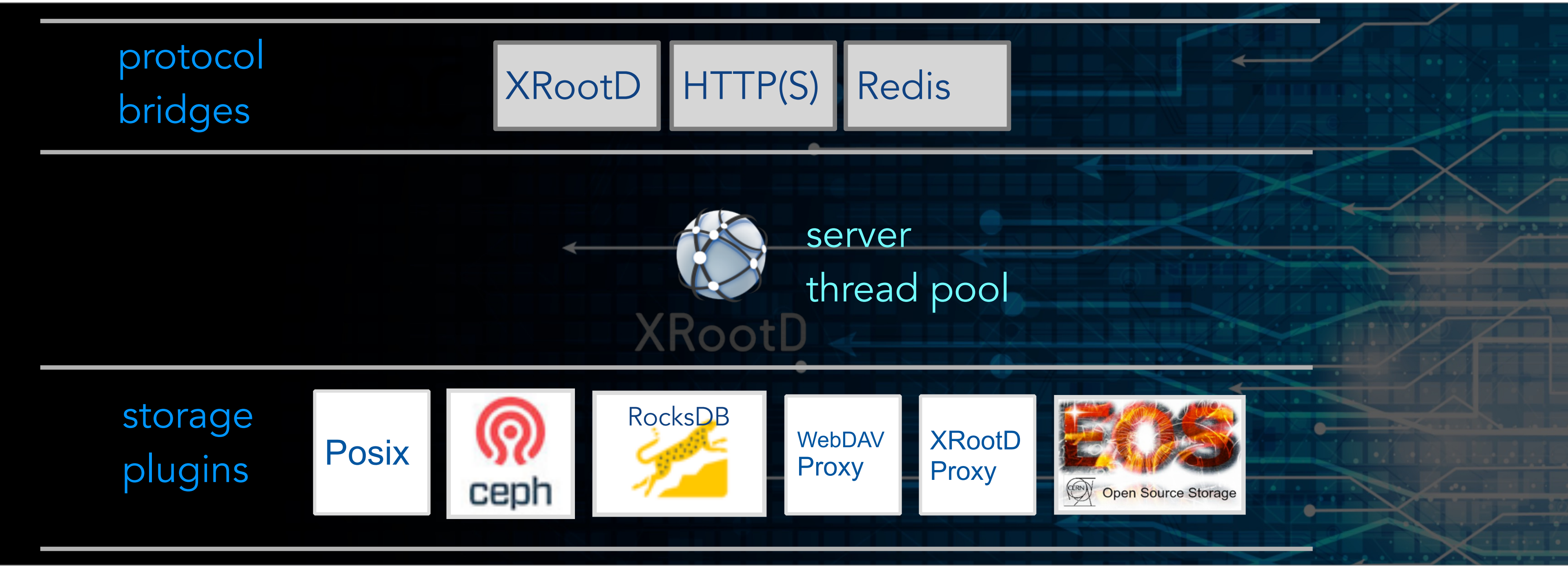
- Storage resources are arranged in a tree structure with top-level subscription
- Clients can start discovering resources at any level and get redirect between tree levels to locate a resource or to fall back in case of error conditions

3rd party copy



- data flows between servers and not though a client
- client monitors progress and can interrupt third party copy at any time

protocols & plugins



protocol bridges

XRootD

HTTP(S)

Redis

server thread pool

XRootD

storage plugins

Posix

ceph

RocksDB

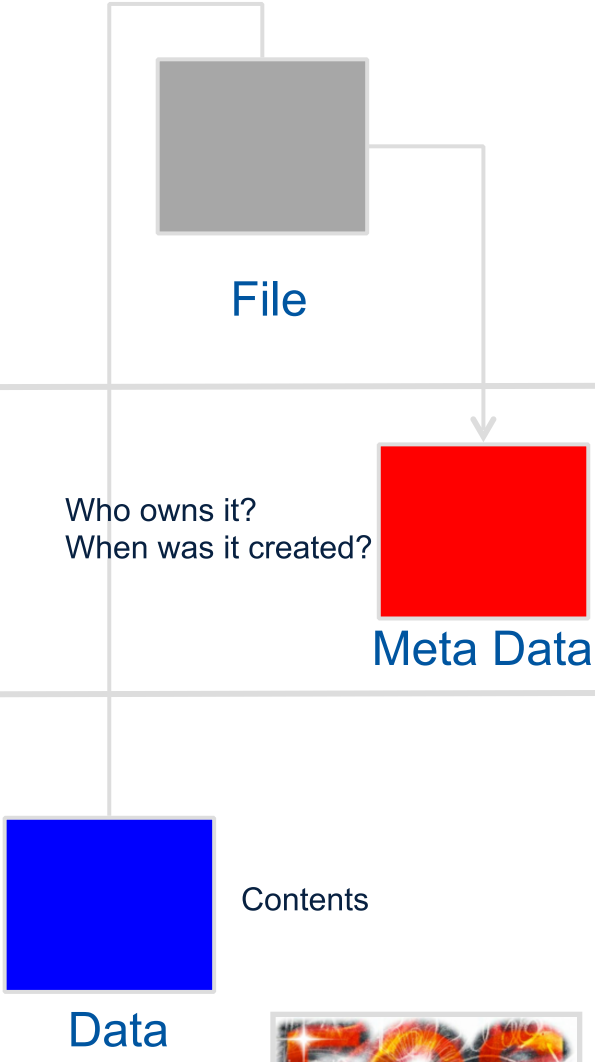
WebDAV Proxy

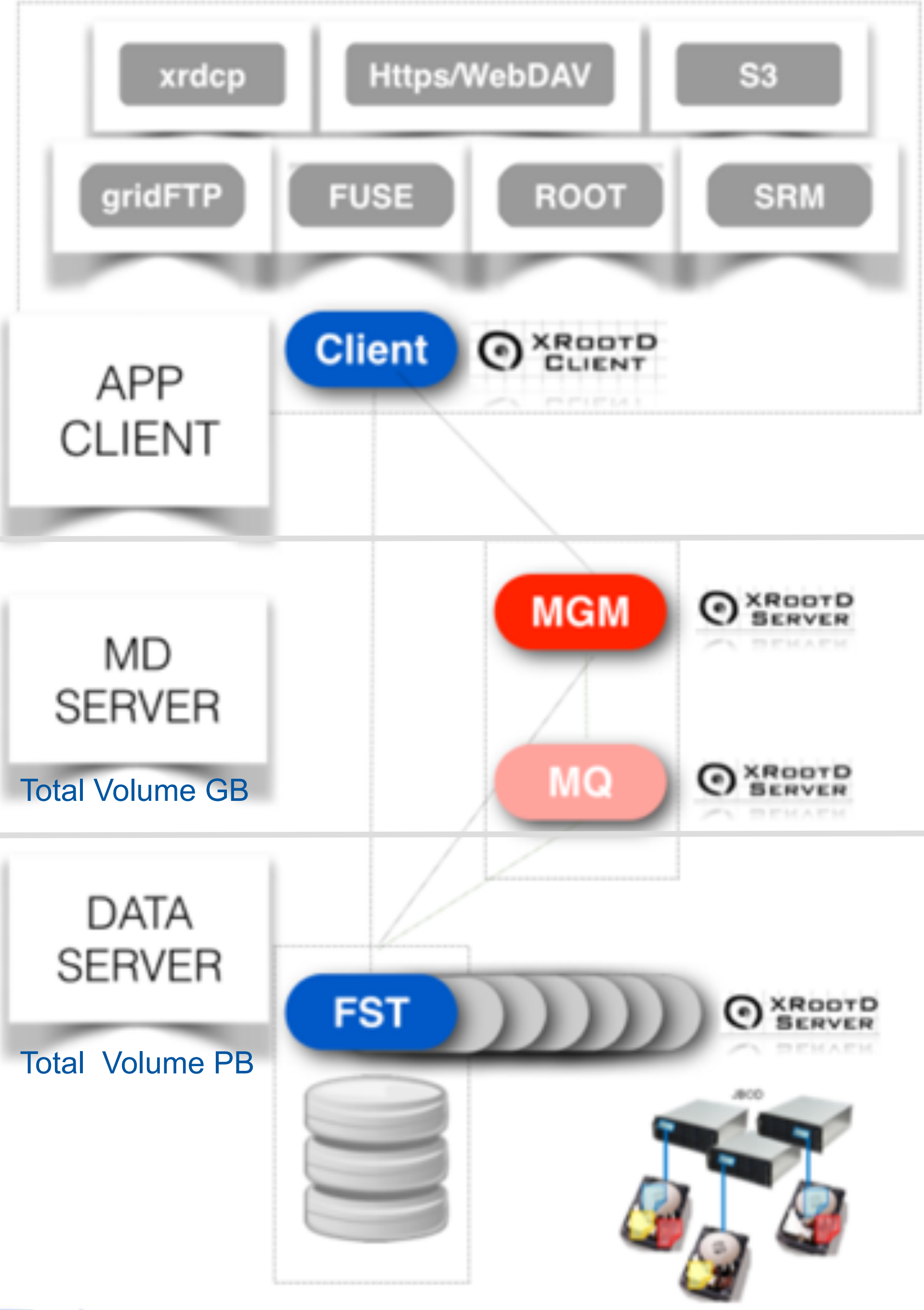
XRootD Proxy

EOS

Open Source Storage

Architecture

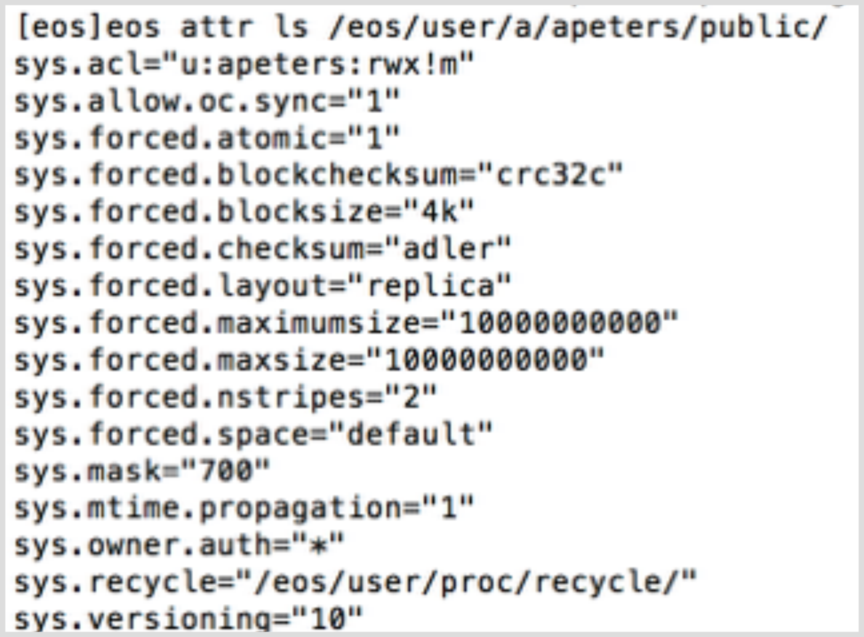




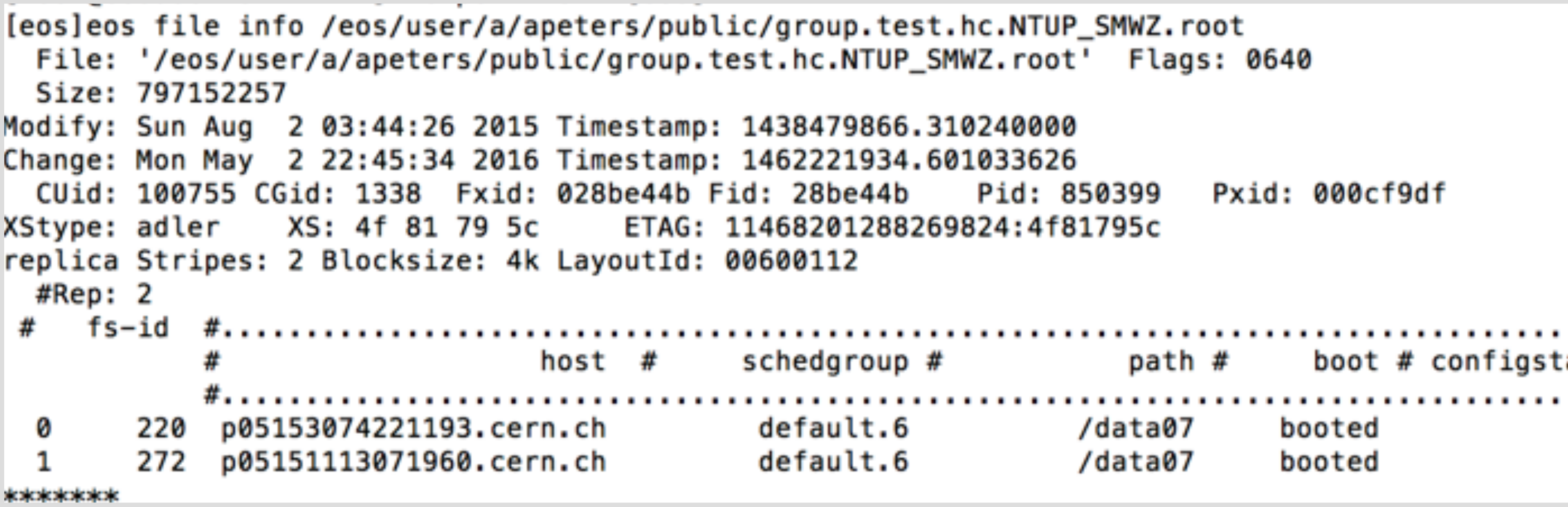
A glimpse of EOS features …

- **low latency** due to in-memory namespace (ms)
- ‘cheap’ – uses JBOD *just a bunch of disks*
  no RAID controller but software implementation to **replicate** or **erasure encode** files for redundancy
- rich **access control** lists *who can read files …*
- user, group & project **quota** system *each user has 2TB …*
- easy to operate and deploy
- EOS **server** runs on **Linux** platform
- EOS **client** runs on **Linux**, **OSX** platform
- via **CIFS** bridge/**WebDav** accessible from **Windows**

An example EOS File



```
[eos]eos attr ls /eos/user/a/apeters/public/
sys.acl="u:apeters:rwx!m"
sys.allow.oc.sync="1"
sys.forced.atomic="1"
sys.forced.blockchecksum="crc32c"
sys.forced.blocksize="4k"
sys.forced.checksum="adler"
sys.forced.layout="replica"
sys.forced.maximumsize="10000000000"
sys.forced.maxsize="10000000000"
sys.forced.nstripes="2"
sys.forced.space="default"
sys.mask="700"
sys.mtime.propagation="1"
sys.owner.auth="*"
sys.recycle="/eos/user/proc/recycle/"
sys.versioning="10"
```



```
[eos]eos file info /eos/user/a/apeters/public/group.test.hc.NTUP_SMWZ.root
  File: '/eos/user/a/apeters/public/group.test.hc.NTUP_SMWZ.root'  Flags: 0640
  Size: 797152257
Modify: Sun Aug  2 03:44:26 2015 Timestamp: 1438479866.310240000
Change: Mon May  2 22:45:34 2016 Timestamp: 1462221934.601033626
  CUid: 100755 CGid: 1338  Fxid: 028be44b Fid: 28be44b    Pid: 850399   Pxid: 000cf9df
XStype: adler    XS: 4f 81 79 5c     ETAG: 11468201288269824:4f81795c
replica Stripes: 2 Blocksize: 4k LayoutId: 00600112
  #Rep: 2
 #    fs-id  #.........................................................................
             #                    host  #      schedgroup #            path #      boot # configst
             #.........................................................................
  0      220  p05153074221193.cern.ch         default.6           /data07      booted
  1      272  p05151113071960.cern.ch         default.6           /data07      booted
*******
```

EOS Releases named after gemstones

Beryl Aquamarine

V 0.3.X

Citrine

V 4.X

XRootD V3 Server

IPV4

namespace in-memory
data on attached disks

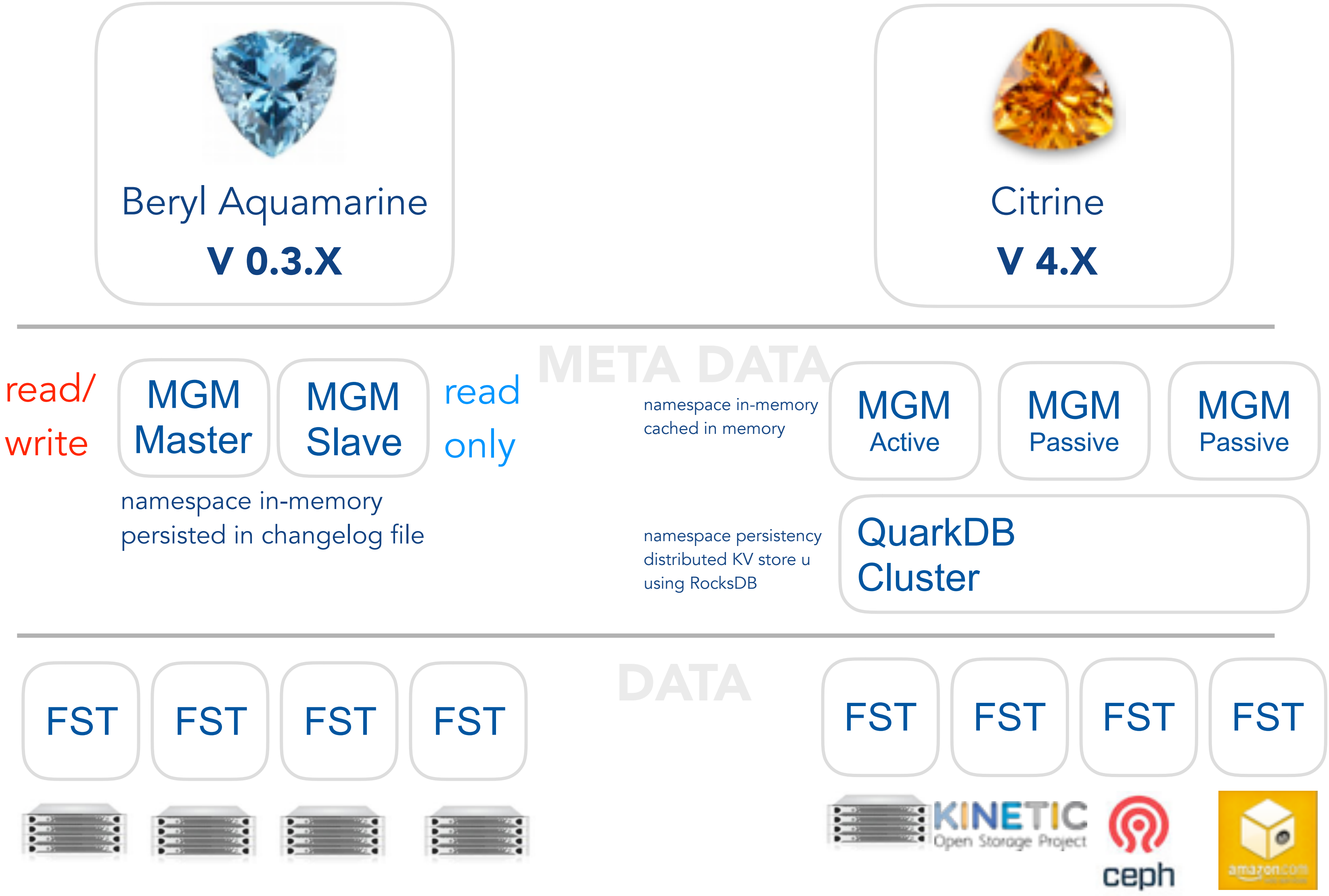
XRootD V4 Server

IPV6

plugins for meta
data & data persistency

Software Repository	https://github.com/cern-eos/eos/ https://gitlab.cern.ch/dss/eos
Web	https://eos.cern.ch
Documentation	https://eos.readthedocs.io

EOS Architectural Evolution



Beryl Aquamarine

V 0.3.X

Citrine

V 4.X

META DATA

read/
write

MGM Master

MGM Slave

read
only

namespace in-memory
persisted in changelog file

namespace in-memory
cached in memory

MGM Active

MGM Passive

MGM Passive

namespace persistency
distributed KV store u
using RocksDB

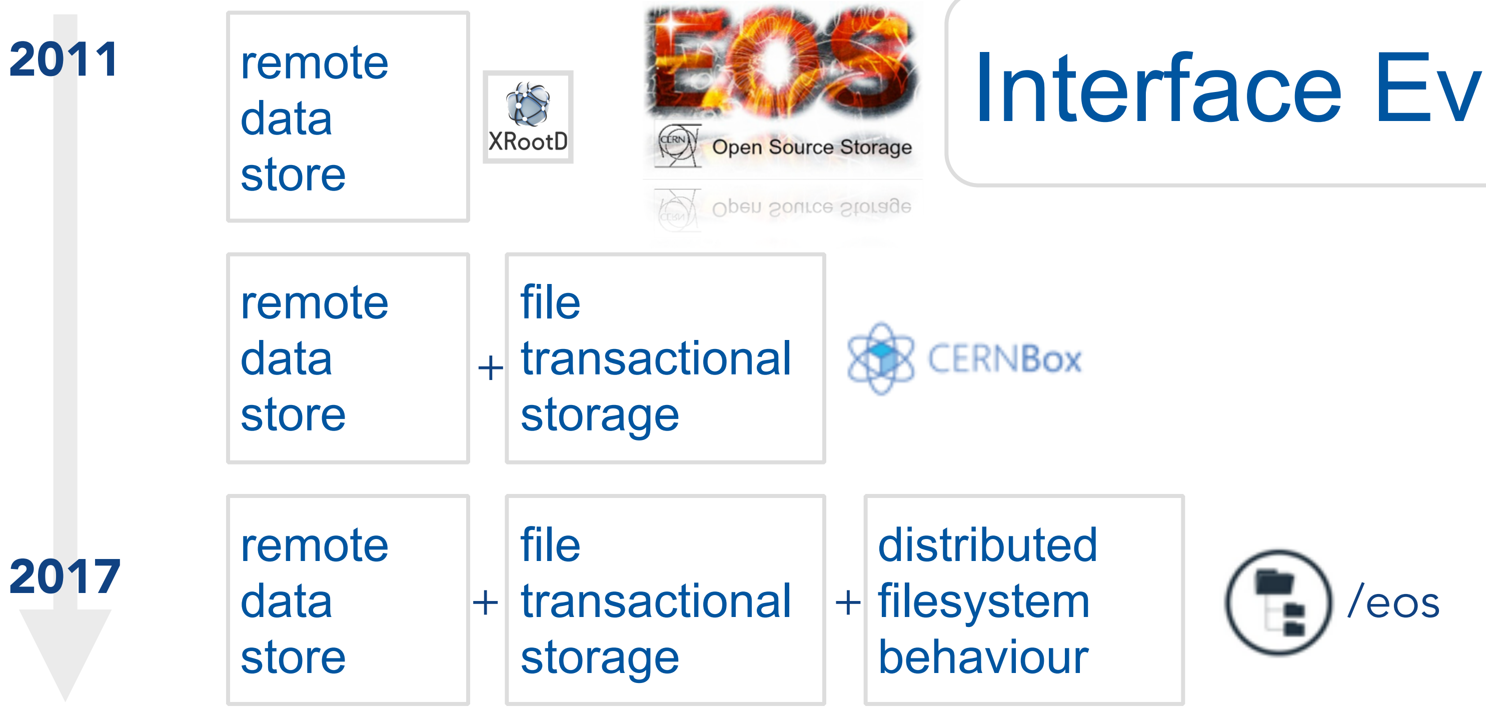
QuarkDB Cluster

DATA

FST FST FST FST

FST FST FST FST

Interface Evolution

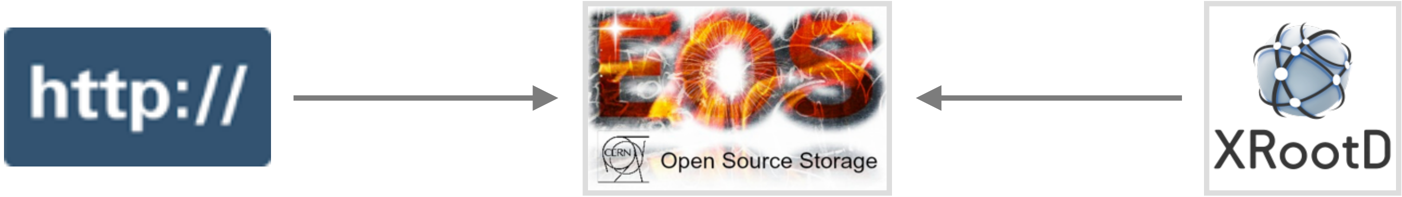


Evolution

EOS has started 6 years ago as a remotely accessible data storage system with *posix-similar* **interface.** The interfaces has been extended to provide **file transaction functionality**. The most recent architectural change is to provide mounted filesystem semantics.

The challenge:

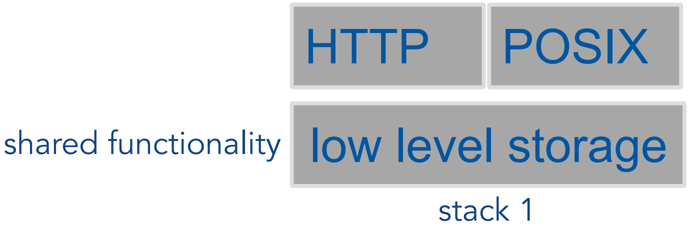
How to marry two different worlds in the same storage system and make them visible to each other?

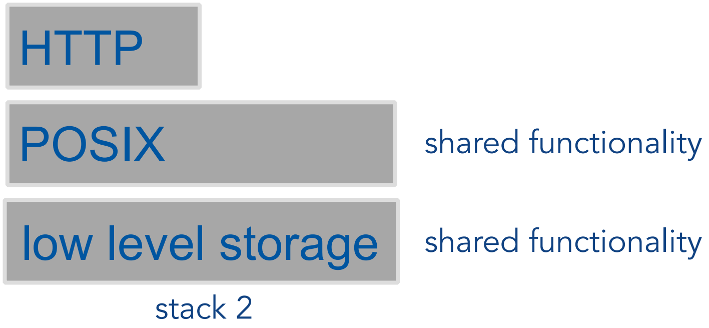


File Transaction IF
GET/PUT
no update
checksums

POSIX
mutable files

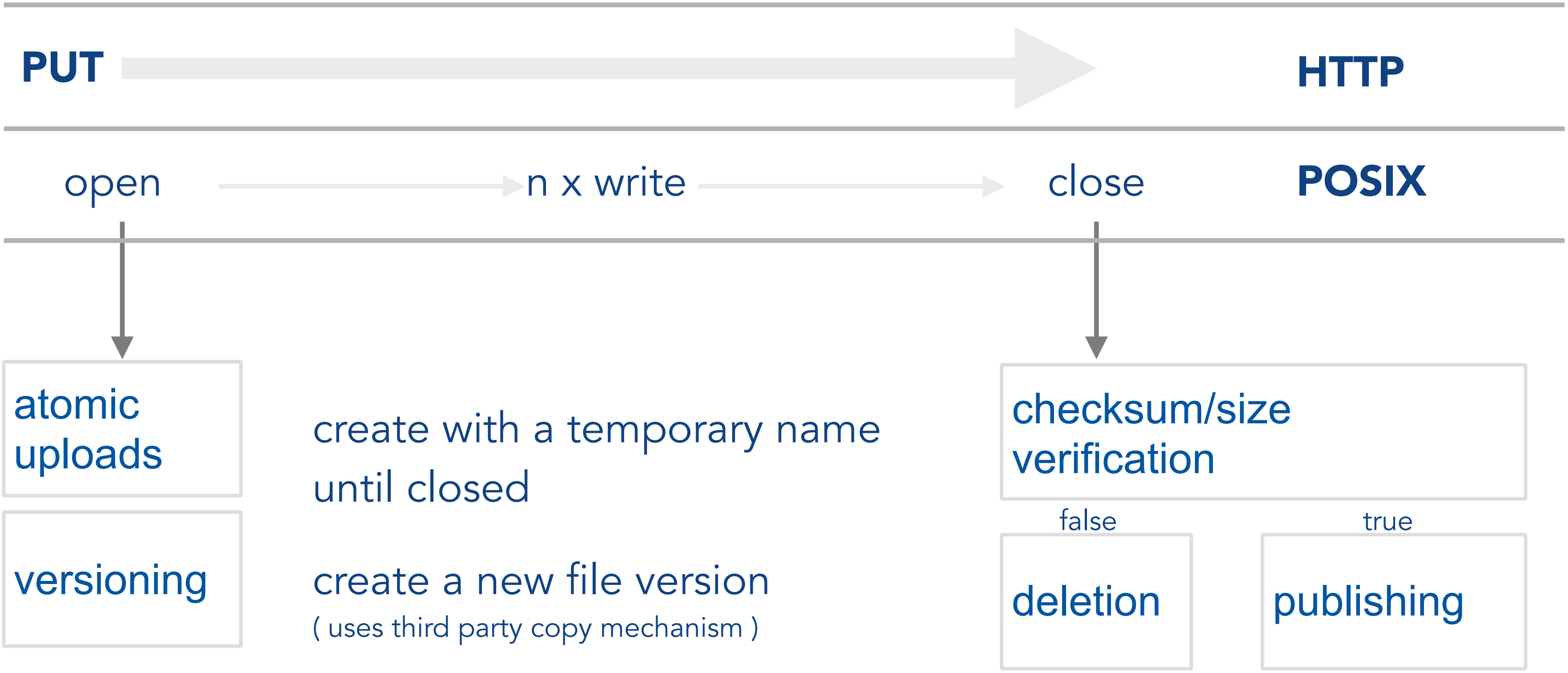
Where to implement extensions …


HTTP
POSIX
shared functionality
low level storage
stack 1


HTTP
POSIX
shared functionality
low level storage
shared functionality
stack 2

Sync & Share Extensions

File Transaction Model



PUT

HTTP

open

n x write

close

POSIX

atomic uploads

create with a temporary name until closed

versioning

create a new file version

(uses third party copy mechanism)

checksum/size verification

false

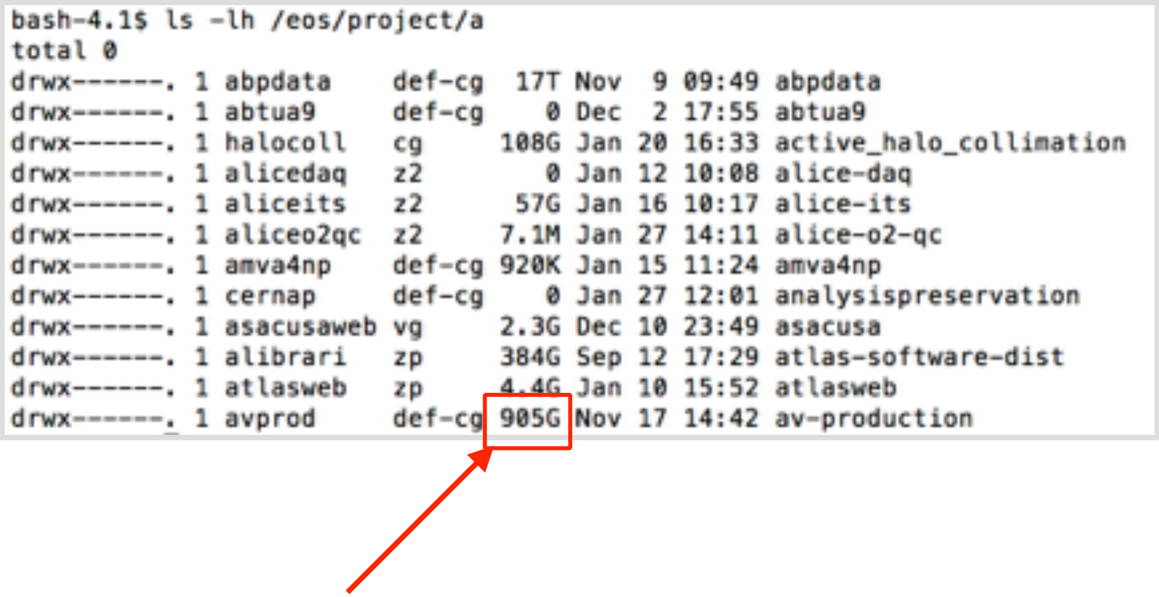
deletion

true

publishing

Sync & Share Extensions

Tree Accounting



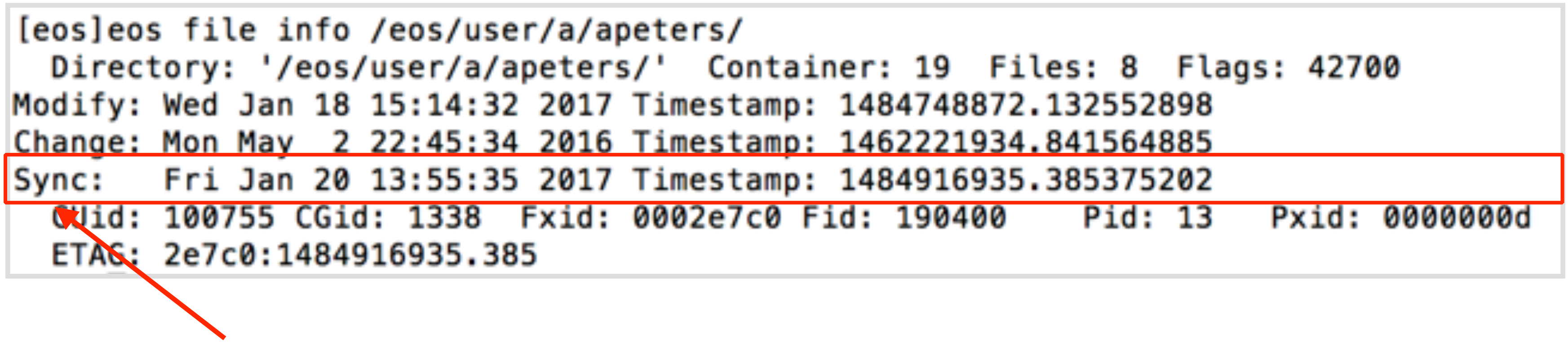
```
bash-4.1$ ls -lh /eos/project/a
total 0
drwx------. 1 abpdata    def-cg  17T Nov  9 09:49 abpdata
drwx------. 1 abtua9     def-cg    0 Dec  2 17:55 abtua9
drwx------. 1 halocoll   cg     108G Jan 20 16:33 active_halo_collimation
drwx------. 1 alicedaq   z2        0 Jan 12 10:08 alice-daq
drwx------. 1 aliceits   z2      57G Jan 16 10:17 alice-its
drwx------. 1 aliceo2qc  z2     7.1M Jan 27 14:11 alice-o2-qc
drwx------. 1 amva4np    def-cg 920K Jan 15 11:24 amva4np
drwx------. 1 cernap     def-cg    0 Jan 27 12:01 analysispreservation
drwx------. 1 asacusaweb vg     2.3G Dec 10 23:49 asacusa
drwx------. 1 alibrari   zp     384G Sep 12 17:29 atlas-software-dist
drwx------. 1 atlasweb   zp     4.4G Jan 10 15:52 atlasweb
drwx------. 1 avprod     def-cg 905G Nov 17 14:42 av-production
```

directory size is showing the size sum of all files in the subtree

Subtree accounting is an expensive operation. One file added requires a meta data update of all parent directories in the directory tree. However the operation can be lazy executed. There is a more fine-grained functionality provid by quota accounting.

Sync & Share Extensions

Synchronisation Time



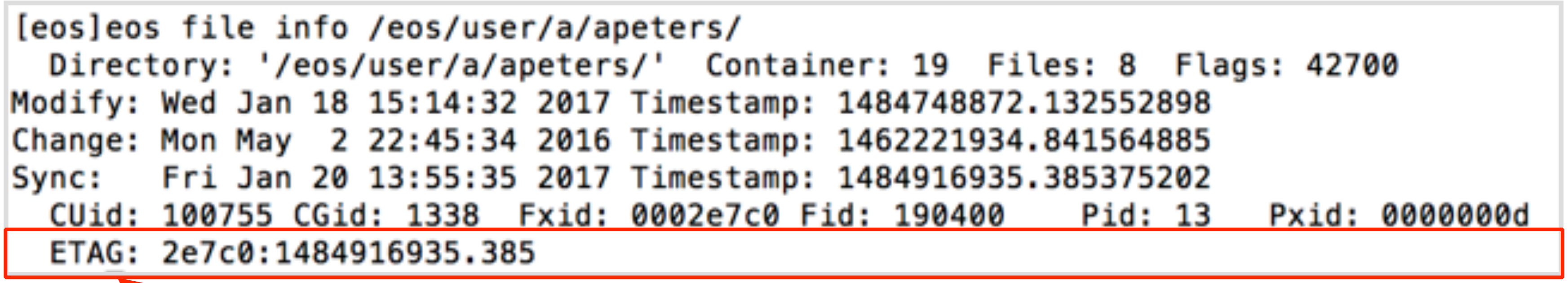
```
[eos]eos file info /eos/user/a/apeters/
  Directory: '/eos/user/a/apeters/'  Container: 19  Files: 8  Flags: 42700
Modify: Wed Jan 18 15:14:32 2017 Timestamp: 1484748872.132552898
Change: Mon May  2 22:45:34 2016 Timestamp: 1462221934.841564885
Sync:   Fri Jan 20 13:55:35 2017 Timestamp: 1484916935.385375202
  CUid: 100755 CGid: 1338  Fxid: 0002e7c0 Fid: 190400    Pid: 13   Pxid: 0000000d
  ETAG: 2e7c0:1484916935.385
```

time of the lastest meta data modification time in this subtree

Synchronisation time propagation is an expensive operation.
One file added requires a meta data update of all parent directories in the directory tree. However the operation can be lazy executed.

Sync & Share Extensions

ETAGs



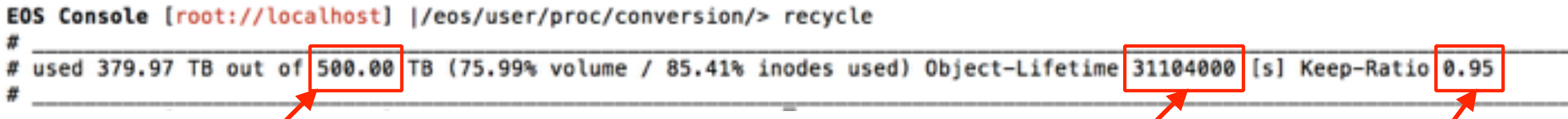
```
[eos]eos file info /eos/user/a/apeters/
  Directory: '/eos/user/a/apeters/'  Container: 19  Files: 8  Flags: 42700
Modify: Wed Jan 18 15:14:32 2017 Timestamp: 1484748872.132552898
Change: Mon May  2 22:45:34 2016 Timestamp: 1462221934.841564885
Sync:   Fri Jan 20 13:55:35 2017 Timestamp: 1484916935.385375202
  CUid: 100755 CGid: 1338  Fxid: 0002e7c0 Fid: 190400    Pid: 13   Pxid: 0000000d
  ETAG: 2e7c0:1484916935.385
```

ETAG for directories are built from id and synchronisation time. For files they are built from id and checksum

Sync & Share Extensions

Recycle Bin

- move deleted files into a recycle bin with time- and/or volume based retention
- allows recovery of accidental deletions
- allows recovery of old versions
- is an option configurable per directory via *xattr*



```
EOS Console [root://localhost] |/eos/user/proc/conversion/> recycle
# ____________________________________________________________________________________
# used 379.97 TB out of 500.00 TB (75.99% volume / 85.41% inodes used) Object-Lifetime 31104000 [s] Keep-Ratio 0.95
# ____________________________________________________________________________________
```

max. size of recycle bin

min. lifetime of files in recycle bin

low watermark

Sync & Share Extensions

- **ACLs** and **virtual roles/ids**
  - not limited by NFS or POSIX acls
    - additionally *write-once, no-deletion, quota-admin, chown, chmod, immutable, sticky ownership, mode-overlay*
    - defined for users, groups and virtual groups (egroups)
  - services like the CERNBox fronted can run with *sudo* privilege acting with a role defined by the connected user

Sync & Share Extensions

- **Workflow Engine**
  - trigger a workflow on filesystem events like
    `open write close delete prepare`
    (think of inotify)
  - under development to put a tape backend behind EOS

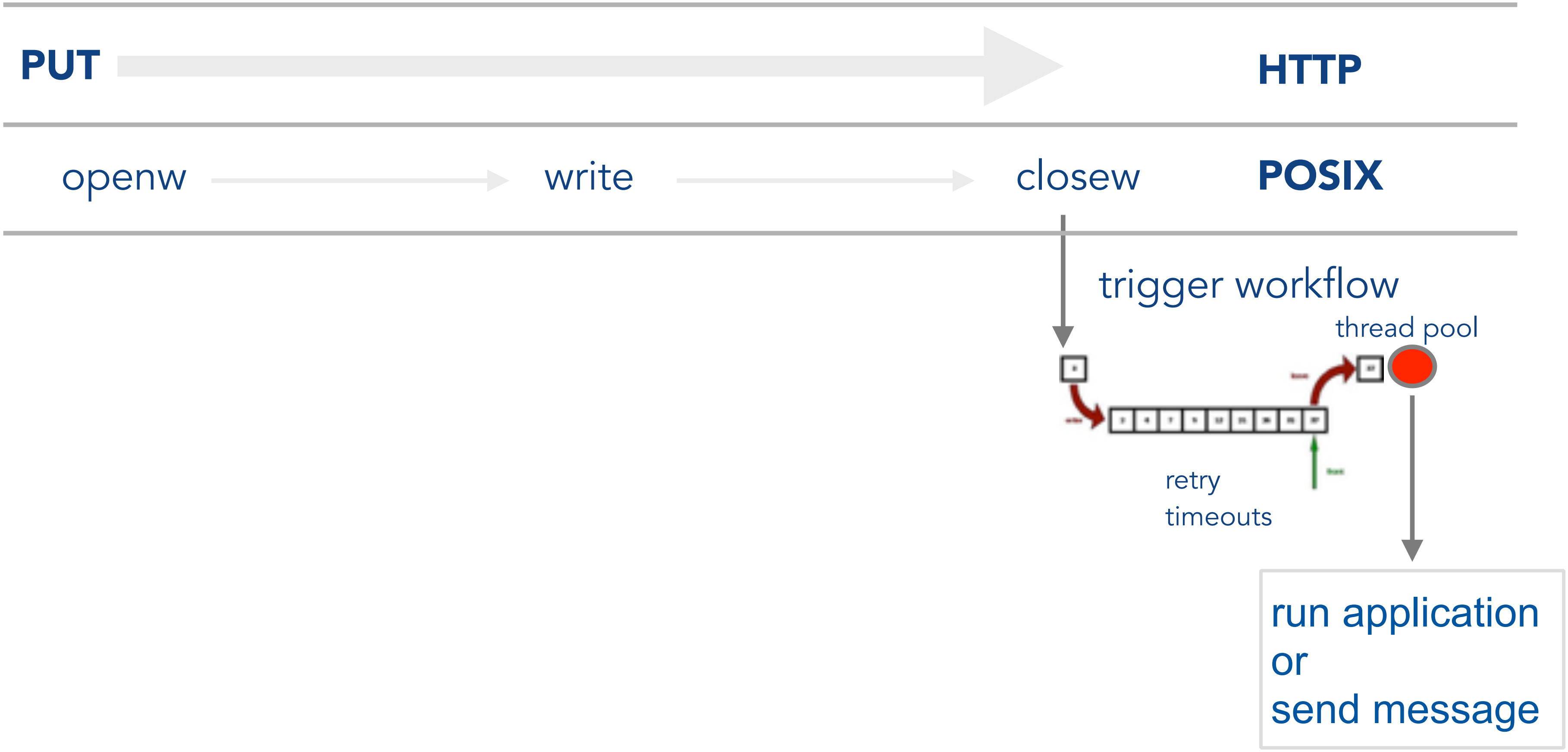
- A sync & share example
  - create a preview images for every new image file

    defined by a single extended attribute:

    ```
    sys.workflow.closew.default=
    “bash:shell:create-preview
    <eos::wfe::path>
    <eos::wfe::path>.thumbnail”
    ```

Workflows

File Transaction Model



PUT

HTTP

openw

write

closew

POSIX

trigger workflow

thread pool

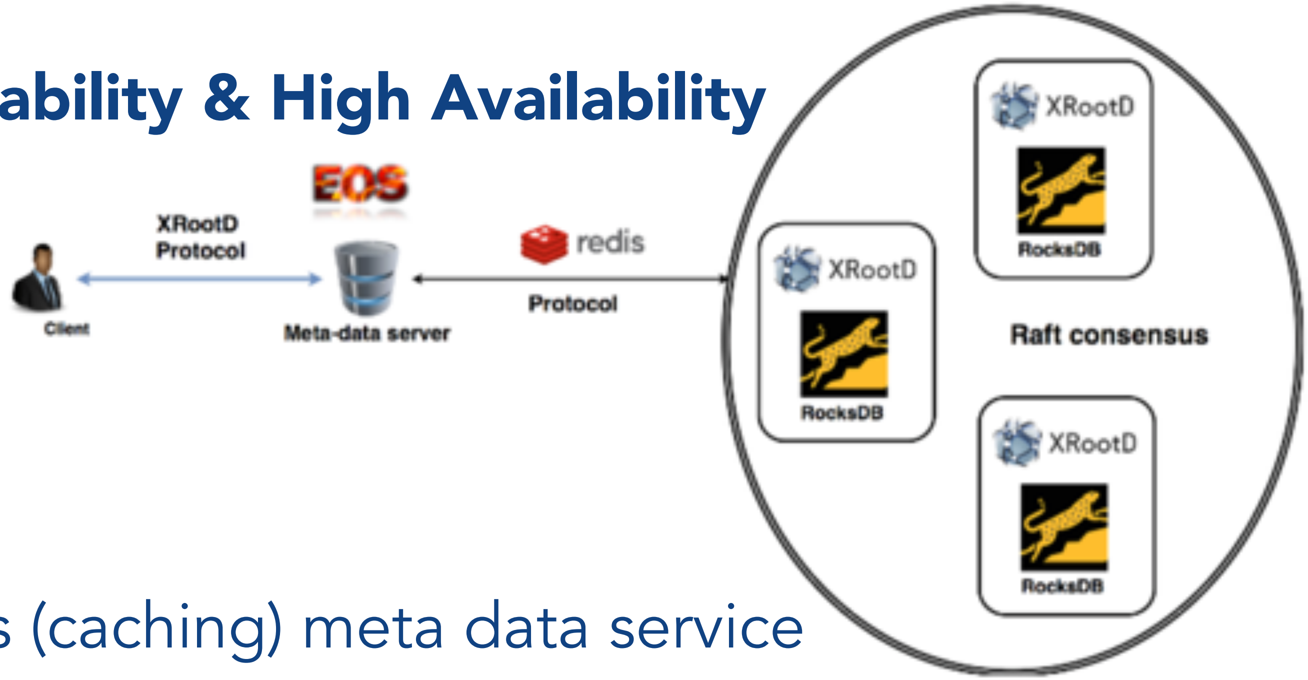
retry

timeouts

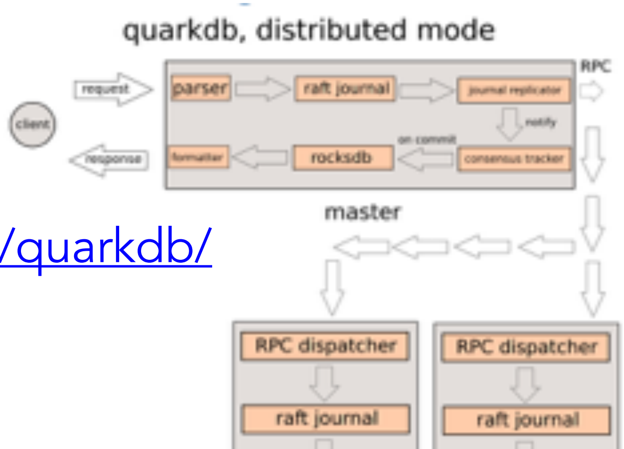
run application
or
send message

Current Developments

Namespace Scalability & High Availability

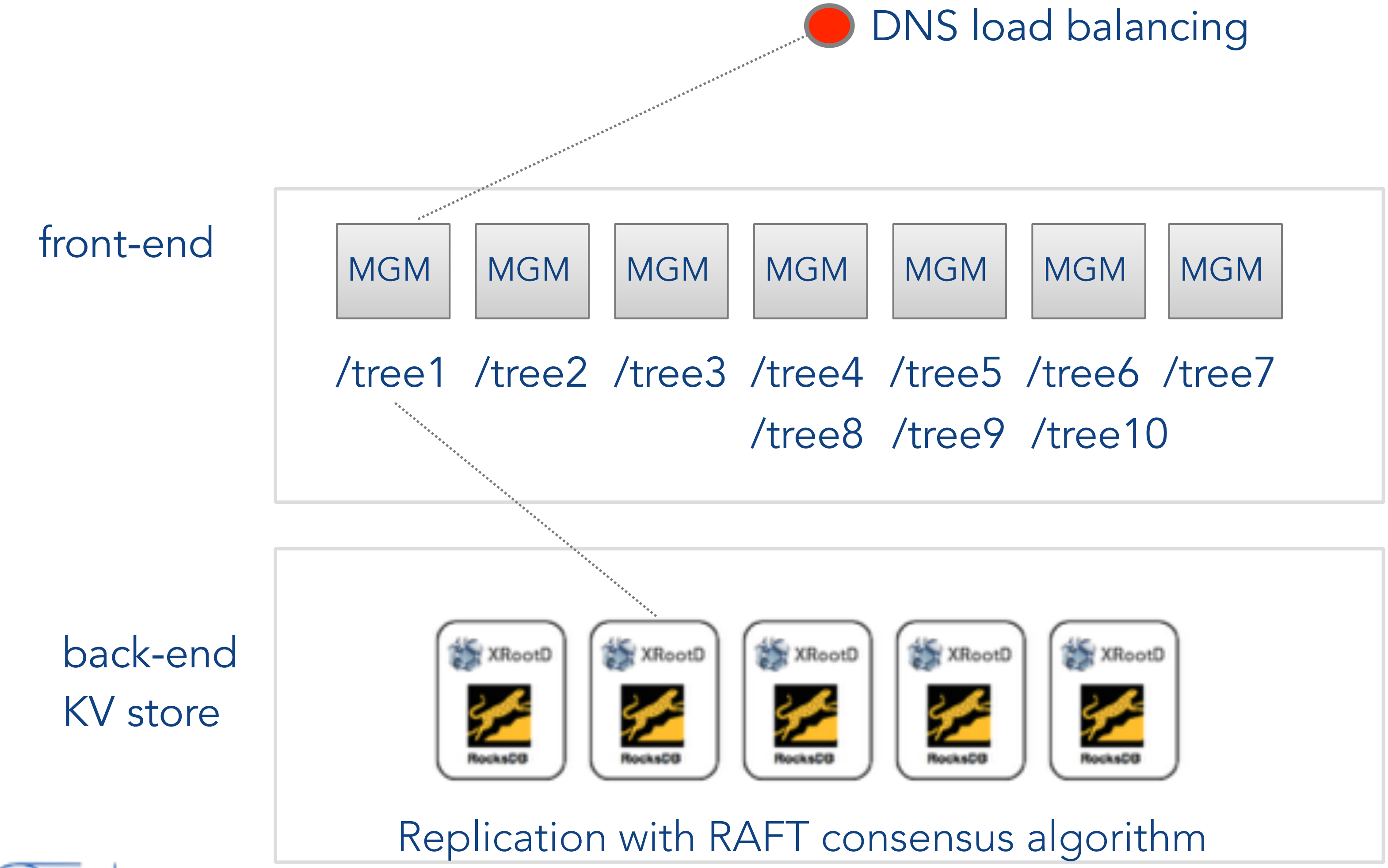


Create a stateless (caching) meta data service running in front of a distributed key-value store (QuarkDB)



QuarkDB
https://github.com/gbitzes/quarkdb/

Namespace Scalability & High Availability



DNS load balancing

front-end

MGM MGM MGM MGM MGM MGM MGM

/tree1 /tree2 /tree3 /tree4 /tree5 /tree6 /tree7

/tree8 /tree9 /tree10

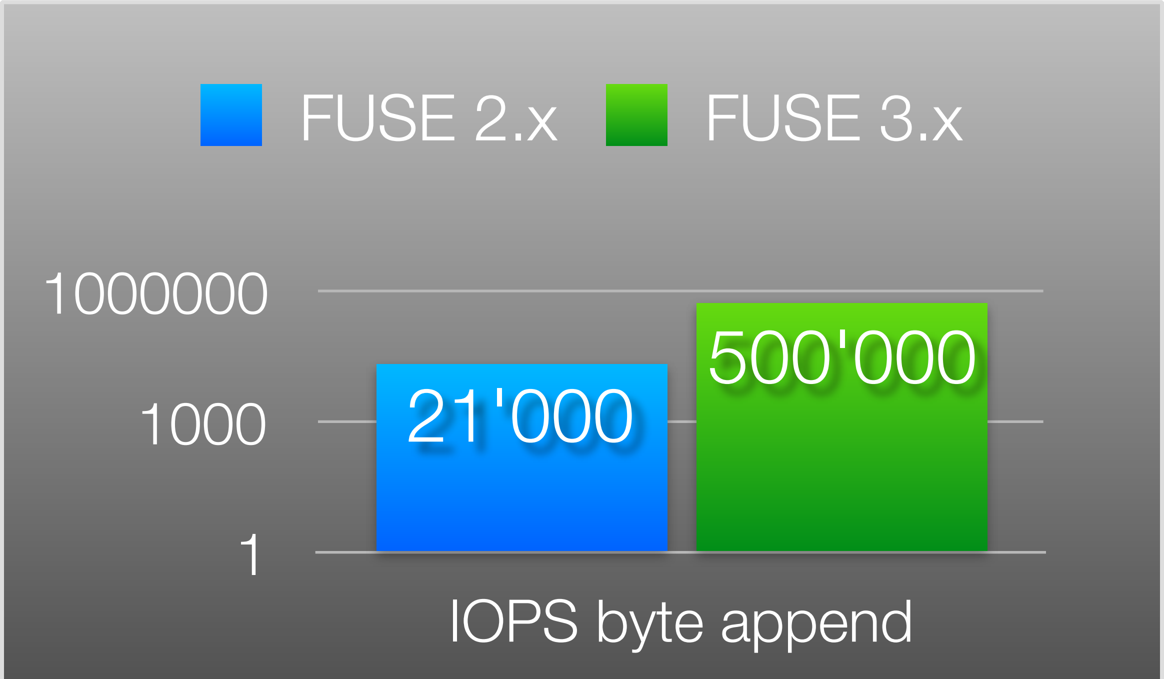
back-end
KV store

Replication with RAFT consensus algorithm

Current Developments

EOS as a FileSystem - FUSE

- a high-performant filesystem interface is a **key-feature to gain access** to a universe of standard applications/services
  e.g. CIFS/NFS4, WebDAV, S3 …
- a user space implementation with FUSE is not as performant as a kernel driver, however significantly easier - no mainstream use case requires a kernel implementation

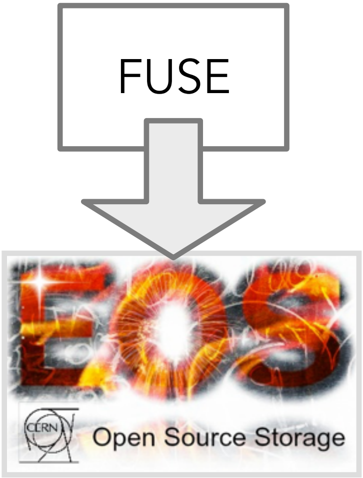


Performance boost for FUSE v3 with write-back cache

Current Developments

EOS as a FileSystem - FUSE$^{\text{3rd generation}}$

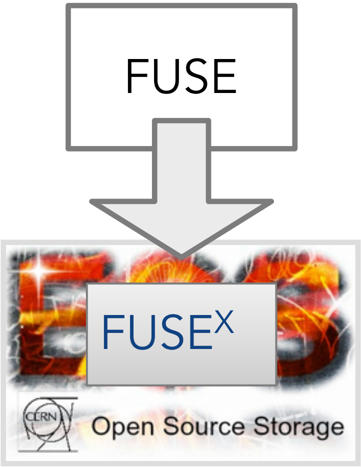
current implementation



client

server

new implementation



client

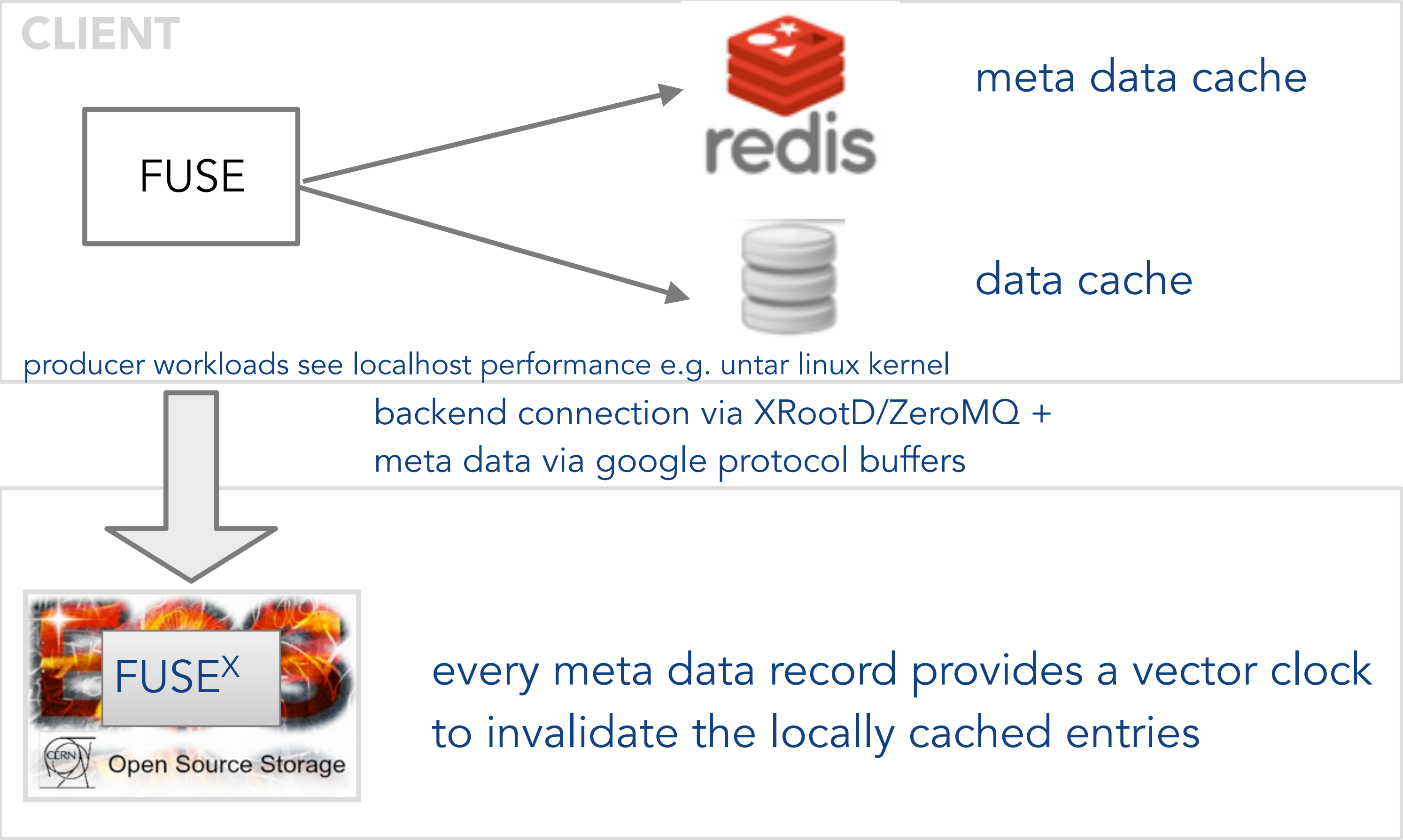
server

FUSE filesystem implemented as **pure client side** application without dedicated server side support.

Dedicated server-side support providing a fully asynchronous server->client communication, leases, locks, file inlining, local meta-data and data caching

Current Developments

EOS as a FileSystem - FUSE$^{\text{3rd generation}}$



producer workloads see localhost performance e.g. untar linux kernel

backend connection via XRootD/ZeroMQ +
meta data via google protocol buffers

every meta data record provides a vector clock
to invalidate the locally cached entries

Challenges

EOS as a FileSystem - FUSE$^{\text{3rd generation}}$ /scalable namespace

a filesystem can never hang

a filesystem can never be unavailable

a filesystem can never be inconsistent

a filesystem has to be as fast as possible

Service Stability CERNBox EOS typical uptime 2 month

High Availability CERNBox EOS restart 0.5-2 hours

Client Stability ~many months

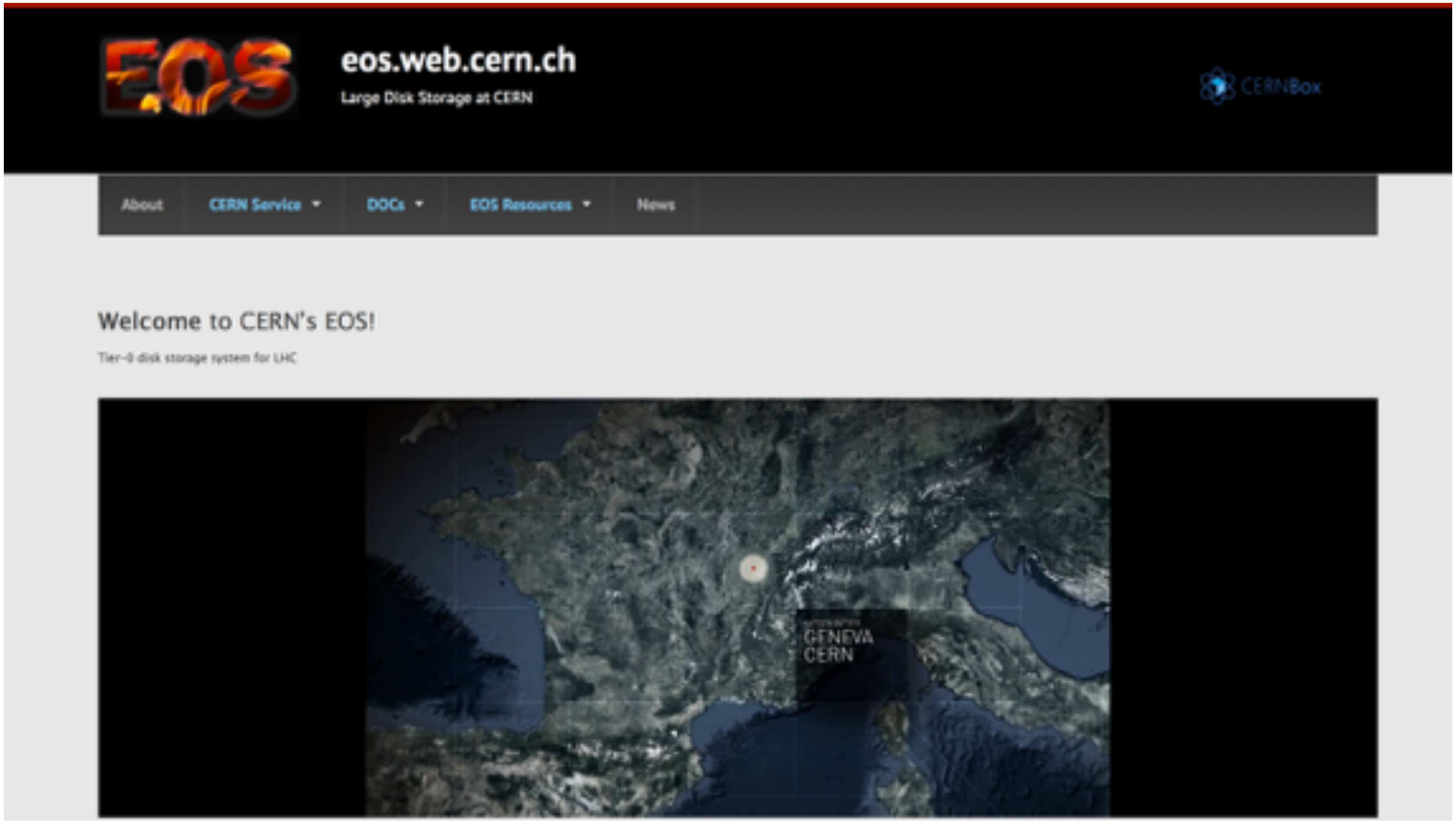
What you need to run EOS …

- you can run everything in a single machine
- Aquamarine meta data server requires 1 GB of memory per 1M files – should have enough memory - not required anymore in Citrine
- we currently provide software packages for Redhat 6 & CentOS 7 LINUX distributions, OSX client

MGM – meta data
FST – data storage
MQ - messaging
FUSE as file system layer
NGINX as HTTPS Server
SAMBA as Windows Server

Information about EOS …

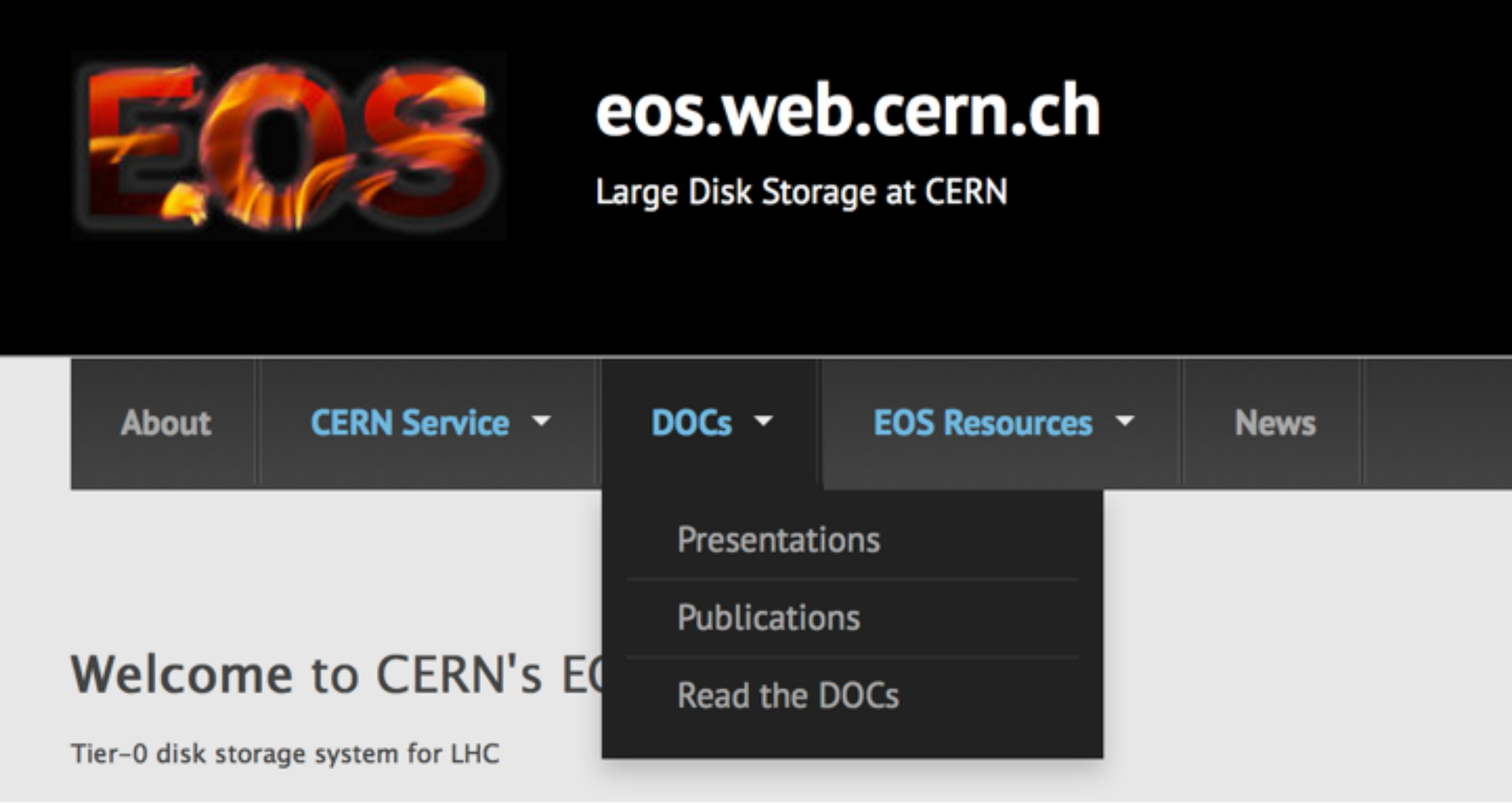
Entry point to EOS: http://eos.web.cern.ch
Email Contact: **eos-project.cern.ch**



eos.web.cern.ch
Large Disk Storage at CERN

Welcome to CERN's EOS!

Tier-0 disk storage system for LHC

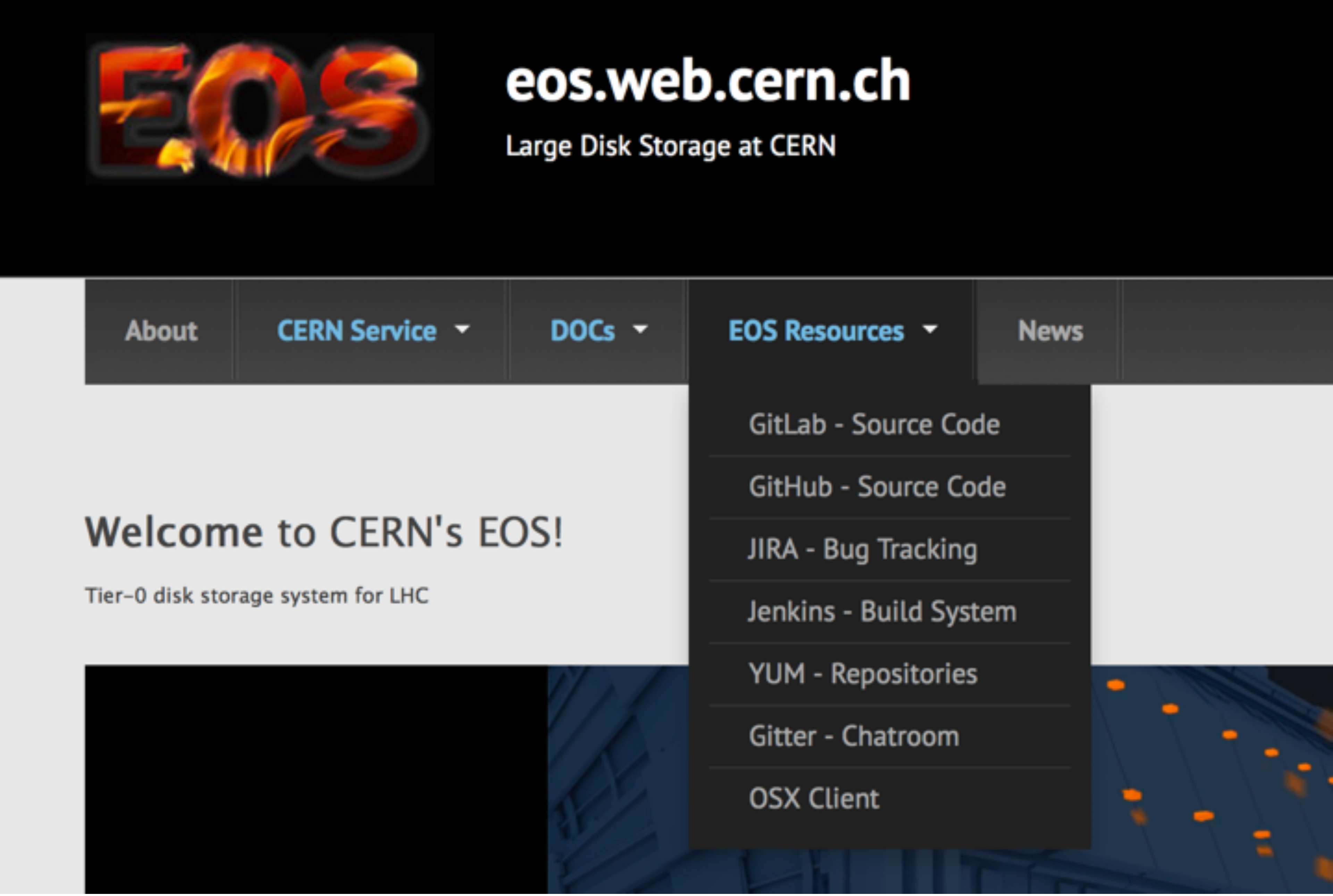


eos.web.cern.ch

Large Disk Storage at CERN

Welcome to CERN's EC

Tier-0 disk storage system for LHC



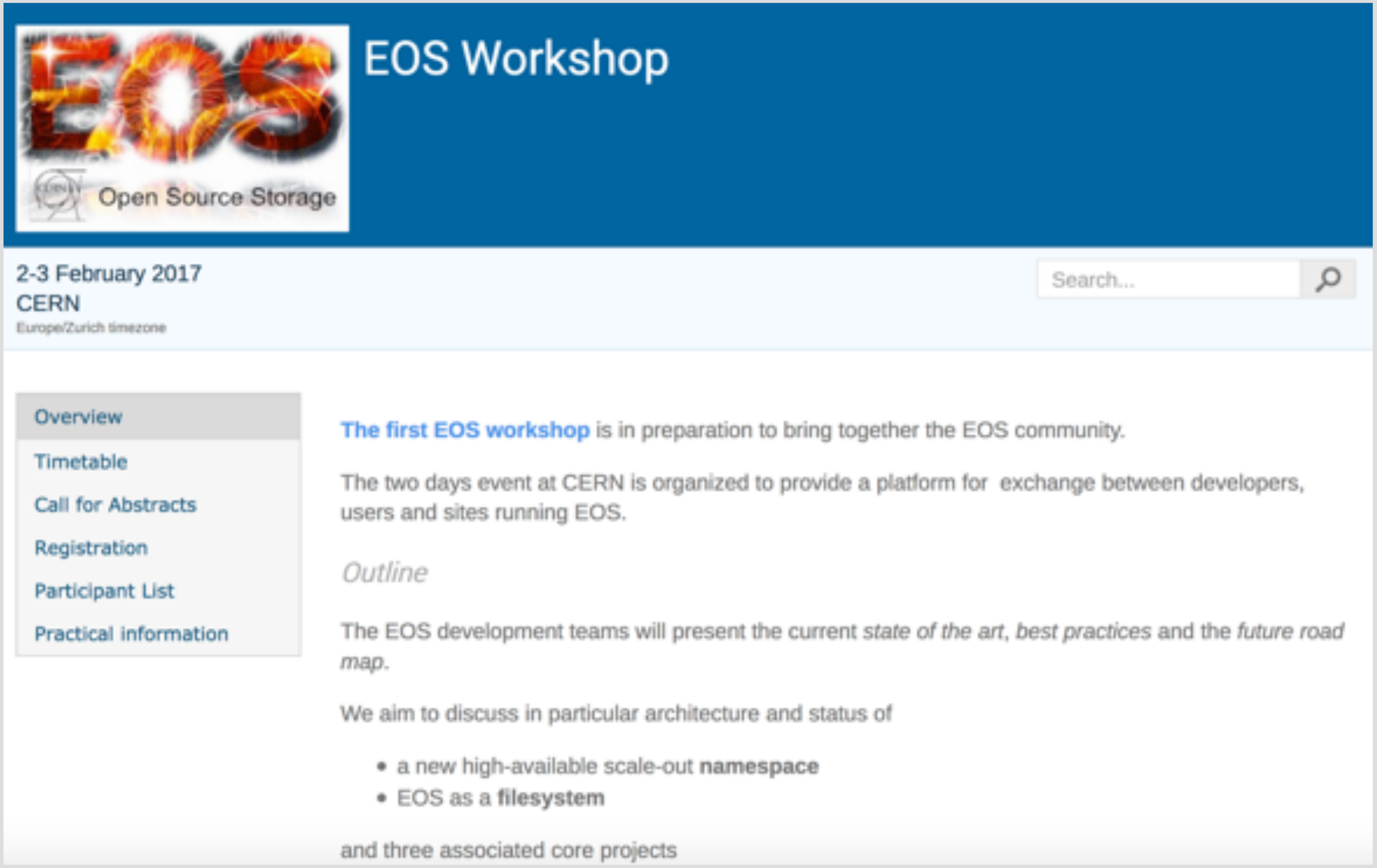
eos.web.cern.ch

Large Disk Storage at CERN

Welcome to CERN's EOS!

Tier-0 disk storage system for LHC

Join the EOS community !



EOS Workshop

2-3 February 2017
CERN
Europe/Zurich timezone

The first EOS workshop is in preparation to bring together the EOS community.

The two days event at CERN is organized to provide a platform for exchange between developers, users and sites running EOS.

Outline

The EOS development teams will present the current *state of the art*, *best practices* and the *future road map*.

We aim to discuss in particular architecture and status of

- a new high-available scale-out **namespace**
- EOS as a **filesystem**

and three associated core projects

https://indico.cern.ch/event/591485/overview

Summary & Outlook

- EOS is a multi-purpose storage system used as physics and user data storage at CERN
- storage platform usable for deployments from 1 to 1000 storage server
- Software under active development by CERN storage development group
  usage is absolutely free – open source project - contributions more than welcome
- Provides additionally a rich feature portfolio not discussed here
  e.g. geographically distributed storage, policies, storage lifecycle tools and many more
- base storage platform for high-level services like **CERNBOX** & **SWAN**

… and many more aspects which could not be mentioned here!

CERN

www.cern.ch

Thank You!