EOS Open Storage

eosxd & **eos**cfsd FUSE filesystems

CernVM Workshop 2024

16–18 Sept 2024
CERN
Europe/Zurich timezone

Elvin Alin Sindrilaru
& Andreas-Joachim Peters
for the EOS Project - CERN IT - Storage Group

Contents

- **EOS** at CERN
  - Architecture
  - Production usage of FUSE mounts
- FUSE clients in EOS
  - **eos**xd
    - Evolution
    - Outlook
  - **eos**cfsd
    - Design

What is **EOS**?

About EOS

EOS provides a service for storing large amounts of physics data and user files, with a focus on interactive and batch analysis.

Flexible

EOS is a storage solution for central data recording, analysis and processing++

Adaptable and Scalable

EOS supports thousands of clients with random remote I/O patterns with multi protocol support
WebDAV, CIFS, FUSE, XRootd, GRPC.

Over 930 PB at CERN

Designed for high capacity and low latency.

Security

EOS offers a variety of authentication methods:KRB5, X509, OIDC, shared secret, and JWT and proprietary token authorisation.

Sync & Share

EOS provides Sync&Share functionality for the CERNBox front-end services.

Tape Storage

EOS includes tape storage in combination with the CTA Cern Tape Archive software.

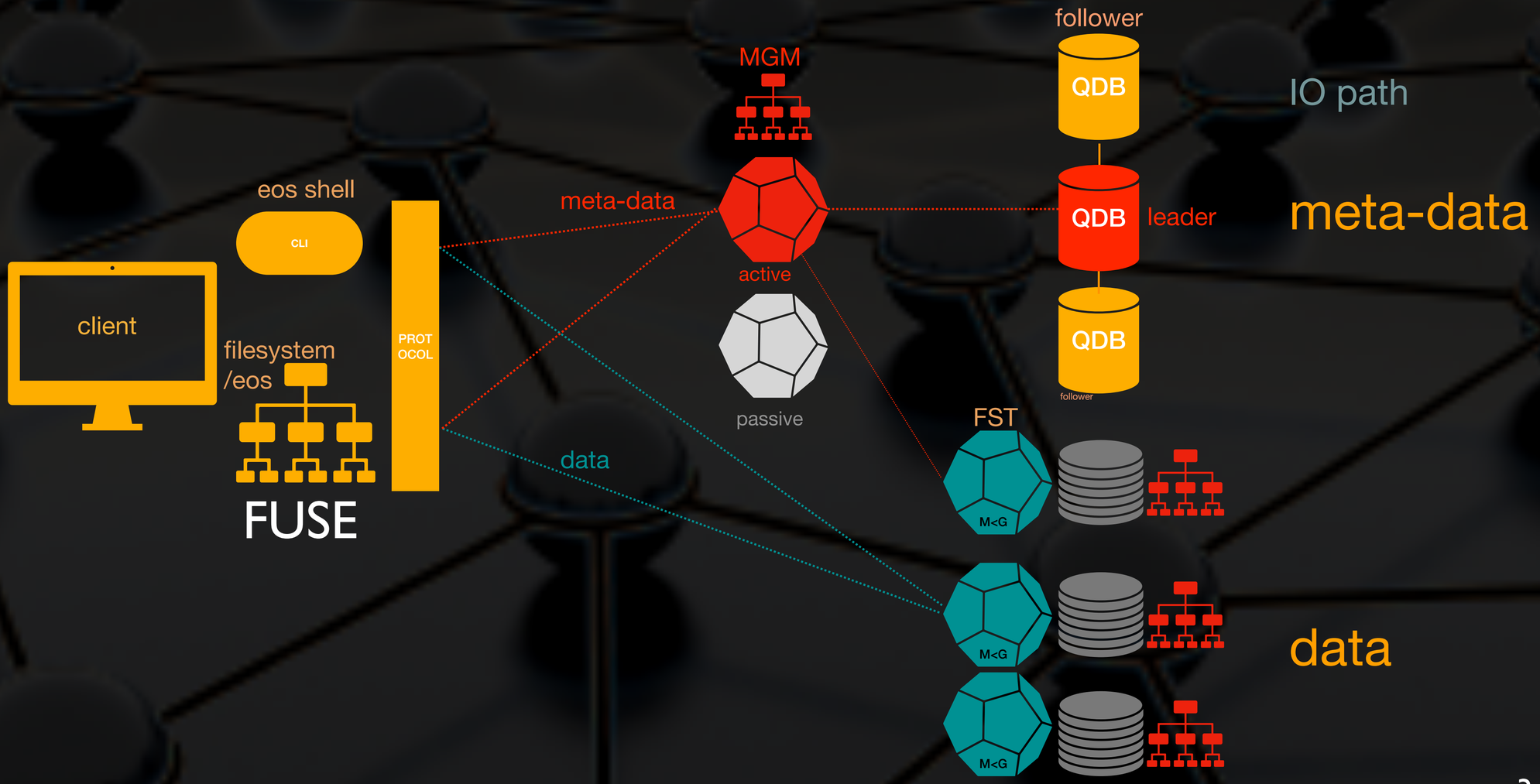
Main differences to mainstream filesystems:

- no client trust
  krb5, gsi, oidc
- clients are fully audited
- clients are regulated (rate limiting)
- subtree space accounting
- sync time aggregation
- file checksums
- user, group project quota
- rich ACL model (e-groups)
- distributed byte-range & flock
- not 100% POSIX
- supporting thousands of clients
- remote stack trace, logging & eviction

EOS Architecture

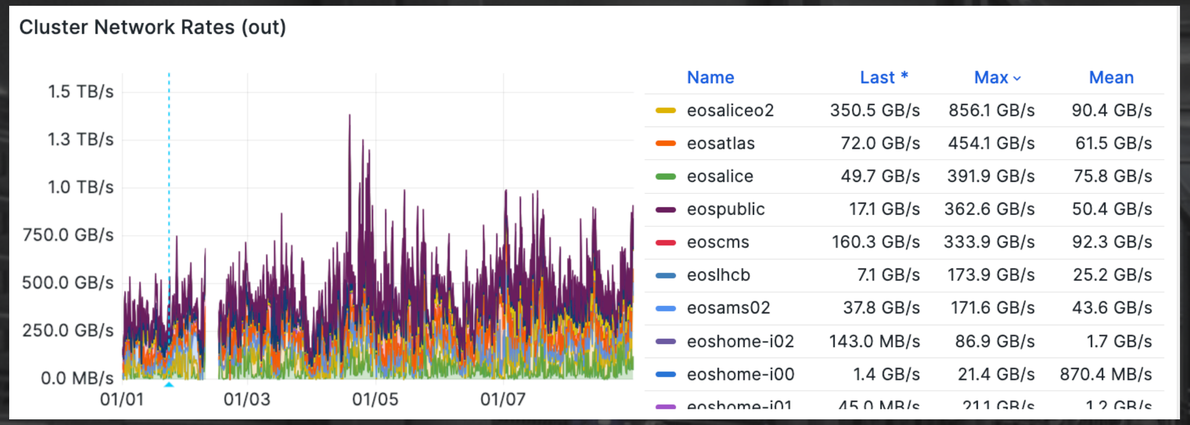
framework
XRootD

components
CLIENTs
MGM
MQ
FST
QuarkDB



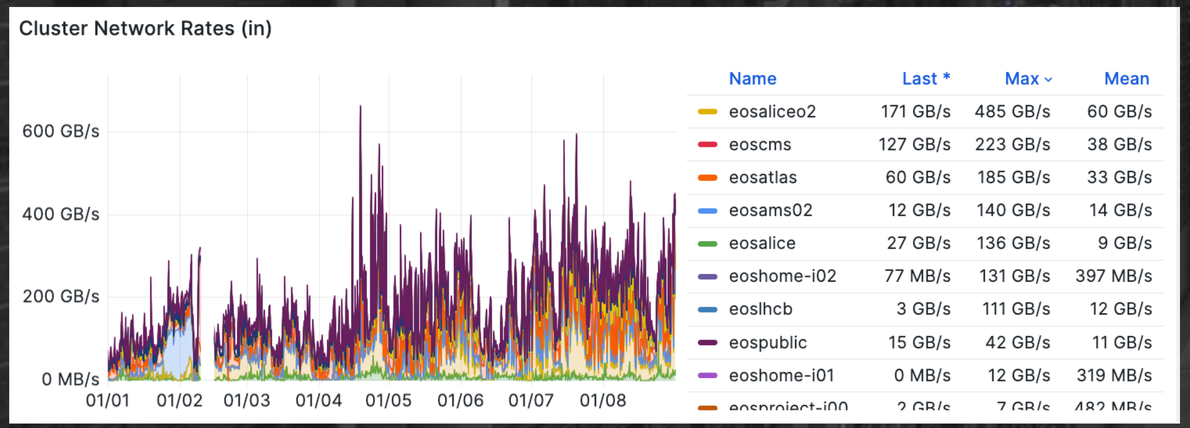
MGM meta-data server FST storage server QuarkDB meta-data persistency

/eos - **EOS** via FUSE at CERN



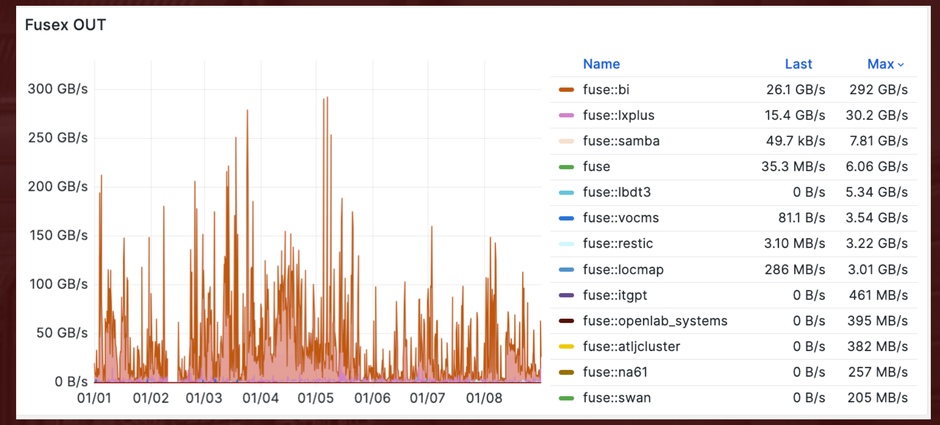
Name	Last *	Max	Mean
eosaliceo2	350.5 GB/s	856.1 GB/s	90.4 GB/s
eosatlas	72.0 GB/s	454.1 GB/s	61.5 GB/s
eosalice	49.7 GB/s	391.9 GB/s	75.8 GB/s
eospublic	17.1 GB/s	362.6 GB/s	50.4 GB/s
eoscms	160.3 GB/s	333.9 GB/s	92.3 GB/s
eoslhcb	7.1 GB/s	173.9 GB/s	25.2 GB/s
eosams02	37.8 GB/s	171.6 GB/s	43.6 GB/s
eoshome-i02	143.0 MB/s	86.9 GB/s	1.7 GB/s
eoshome-i00	1.4 GB/s	21.4 GB/s	870.4 MB/s
eoshome-i01	45.0 MB/s	211 GB/s	1.2 GB/s

read all protocols > 1 TB/s



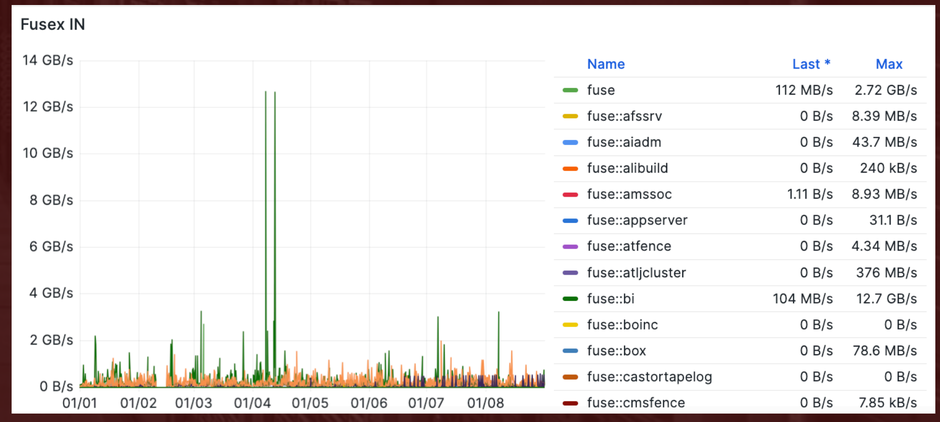
Name	Last *	Max	Mean
eosaliceo2	171 GB/s	485 GB/s	60 GB/s
eoscms	127 GB/s	223 GB/s	38 GB/s
eosatlas	60 GB/s	185 GB/s	33 GB/s
eosams02	12 GB/s	140 GB/s	14 GB/s
eosalice	27 GB/s	136 GB/s	9 GB/s
eoshome-i02	77 MB/s	131 GB/s	397 MB/s
eoslhcb	3 GB/s	111 GB/s	12 GB/s
eospublic	15 GB/s	42 GB/s	11 GB/s
eoshome-i01	0 MB/s	12 GB/s	319 MB/s
eosproject-i00	2 GB/s	7 GB/s	482 MB/s

write all protocols > 300 GB/s



Name	Last	Max
fuse::bi	26.1 GB/s	292 GB/s
fuse::lxplus	15.4 GB/s	30.2 GB/s
fuse::samba	49.7 kB/s	7.81 GB/s
fuse	35.3 MB/s	6.06 GB/s
fuse::lbdt3	0 B/s	5.34 GB/s
fuse::vocms	81.1 B/s	3.54 GB/s
fuse::restic	3.10 MB/s	3.22 GB/s
fuse::locmap	286 MB/s	3.01 GB/s
fuse::itgpt	0 B/s	461 MB/s
fuse::openlab_systems	0 B/s	395 MB/s
fuse::atljcluster	0 B/s	382 MB/s
fuse::na61	0 B/s	257 MB/s
fuse::swan	0 B/s	205 MB/s

read **FUSE** traffic peak 292 GB/s

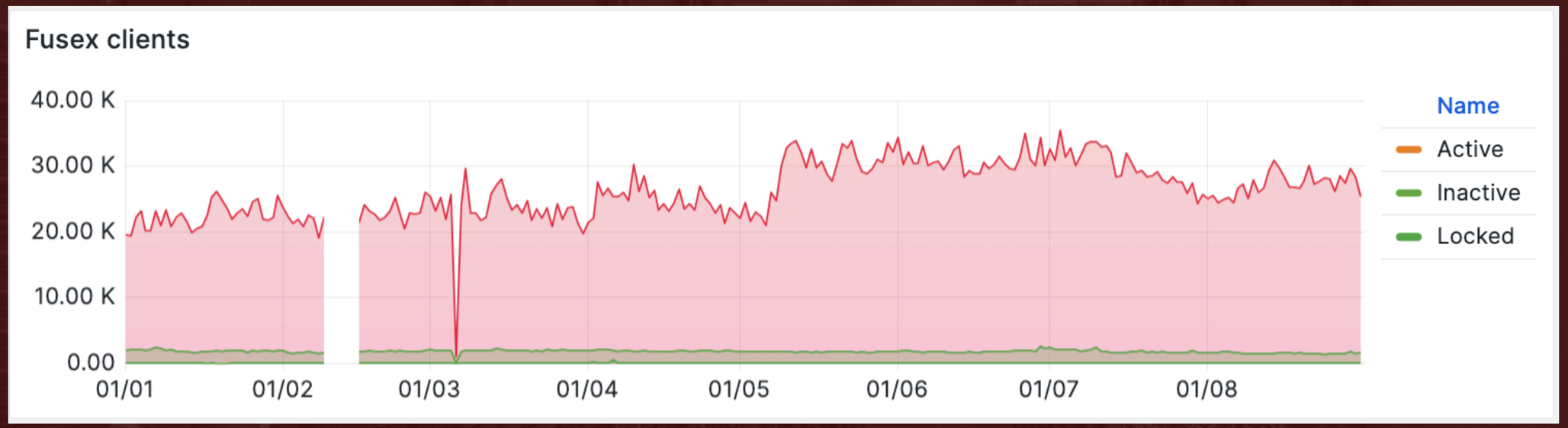


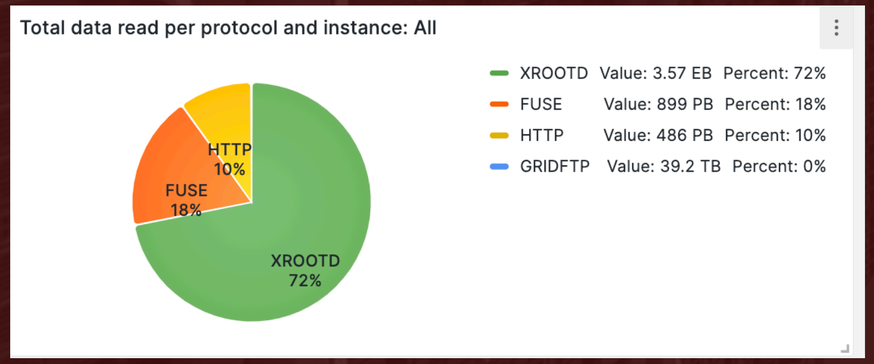
Name	Last *	Max
fuse	112 MB/s	2.72 GB/s
fuse::afssrv	0 B/s	8.39 MB/s
fuse::aiadm	0 B/s	43.7 MB/s
fuse::alibuild	0 B/s	240 kB/s
fuse::amssoc	1.11 B/s	8.93 MB/s
fuse::appserver	0 B/s	31.1 B/s
fuse::atfence	0 B/s	4.34 MB/s
fuse::atljcluster	0 B/s	376 MB/s
fuse::bi	104 MB/s	12.7 GB/s
fuse::boinc	0 B/s	0 B/s
fuse::box	0 B/s	78.6 MB/s
fuse::castortapelog	0 B/s	0 B/s
fuse::cmsfence	0 B/s	7.85 kB/s

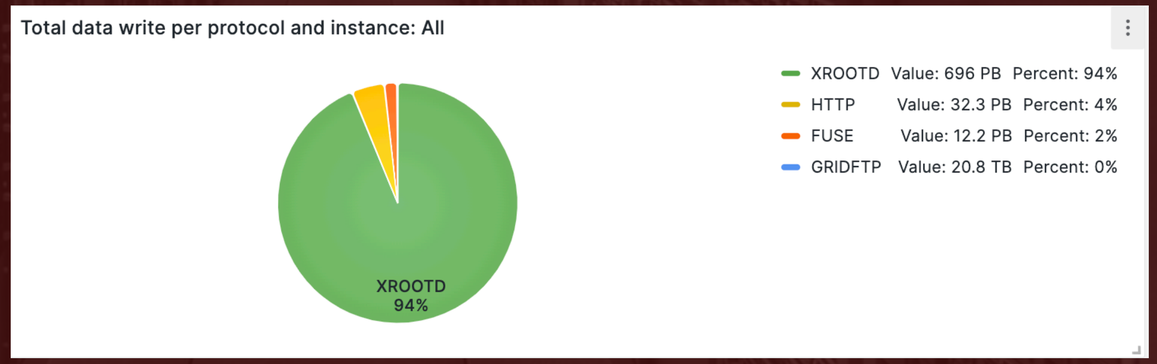
write **FUSE** traffic peak 6 GB/s

/eos - **EOS** via FUSE at CERN

Over 30k mount clients - 900 PB read 32 PB written via FUSE this year so far







/eos Clients

- until 7-2024 mainly **CentOS7** clients **libfuse2**
- since 7-2024 mainly **ALMA9** clients **libfuse2**

- never seen crashes due to FUSE kernel module or ***libfuse*** itself 👏👏👏
- **libfuse3** clients only used in SAMBA gateways for now

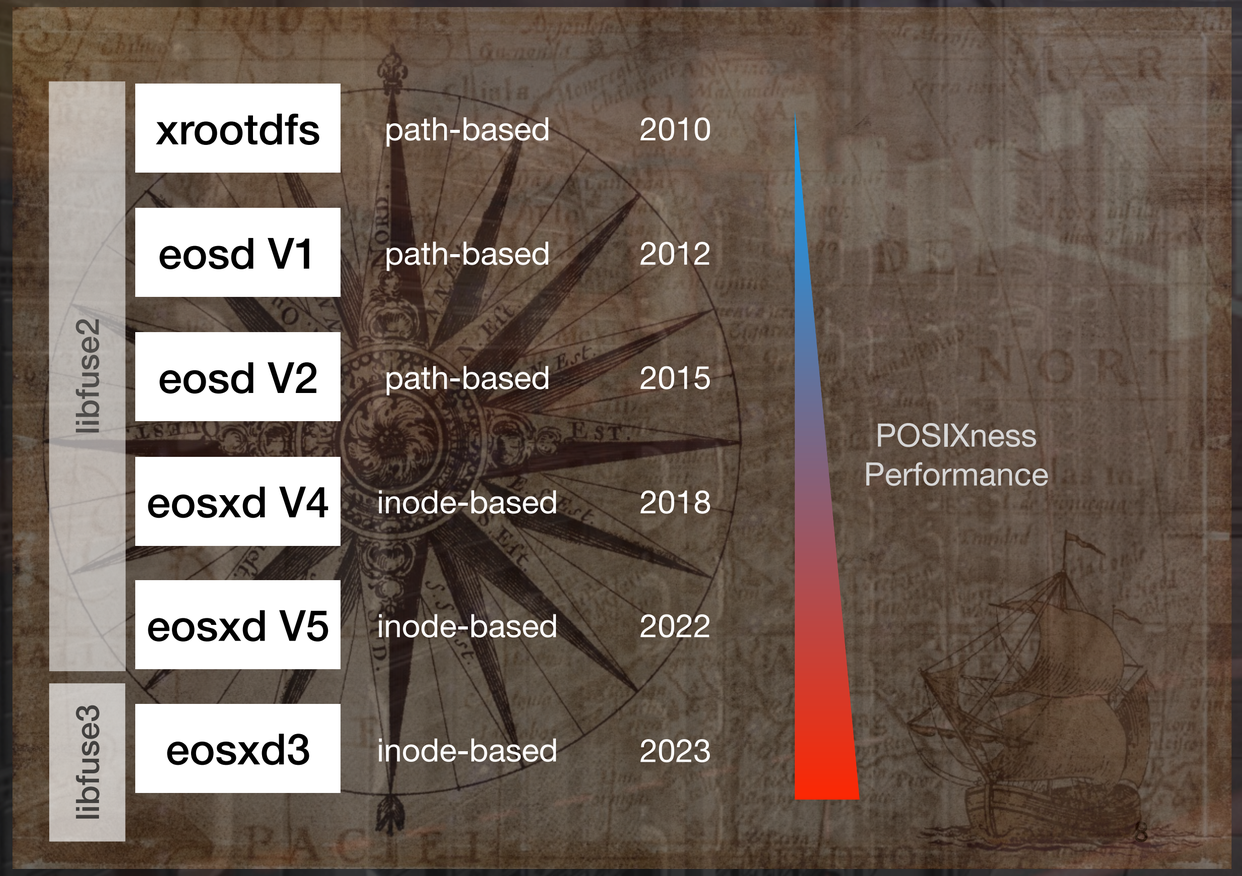
- today we have **few crashes per month** in population of 30k client
  - in the past often due to unsafe concurrent access on meta-data objects - never from ***libfuse***

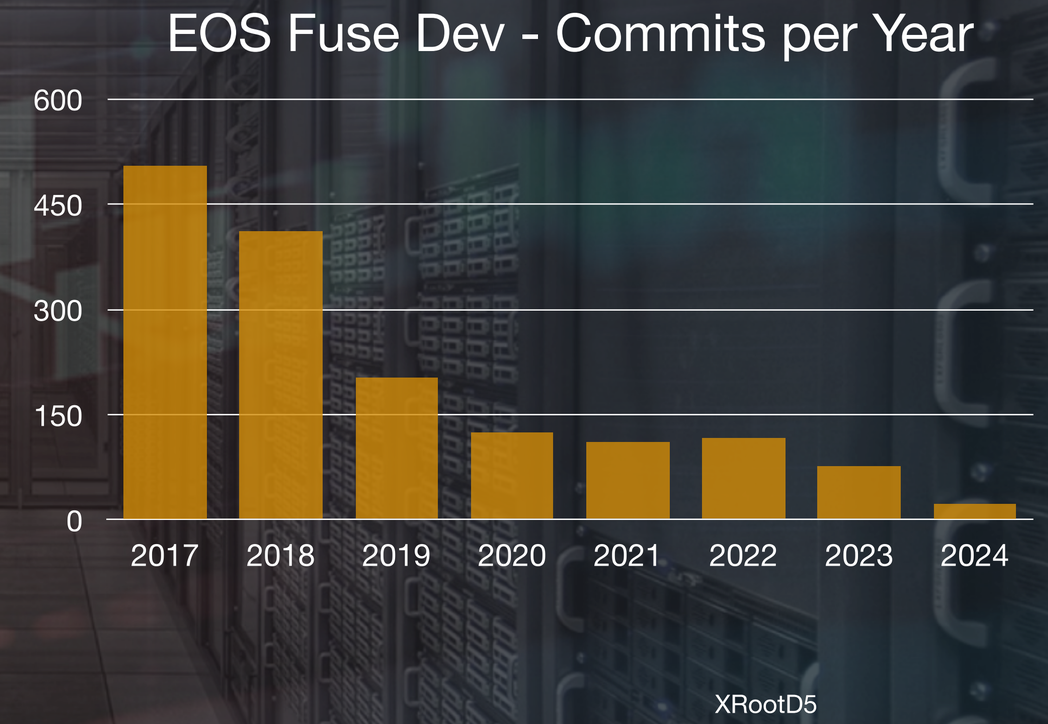
- in the past many issues due to the notification callback mechanisms of FUSE (inval, lookup, forget)
  - typical dead-lock scenario due to locked objects resulting in D state processes

7

EOS Fuse Filesystem

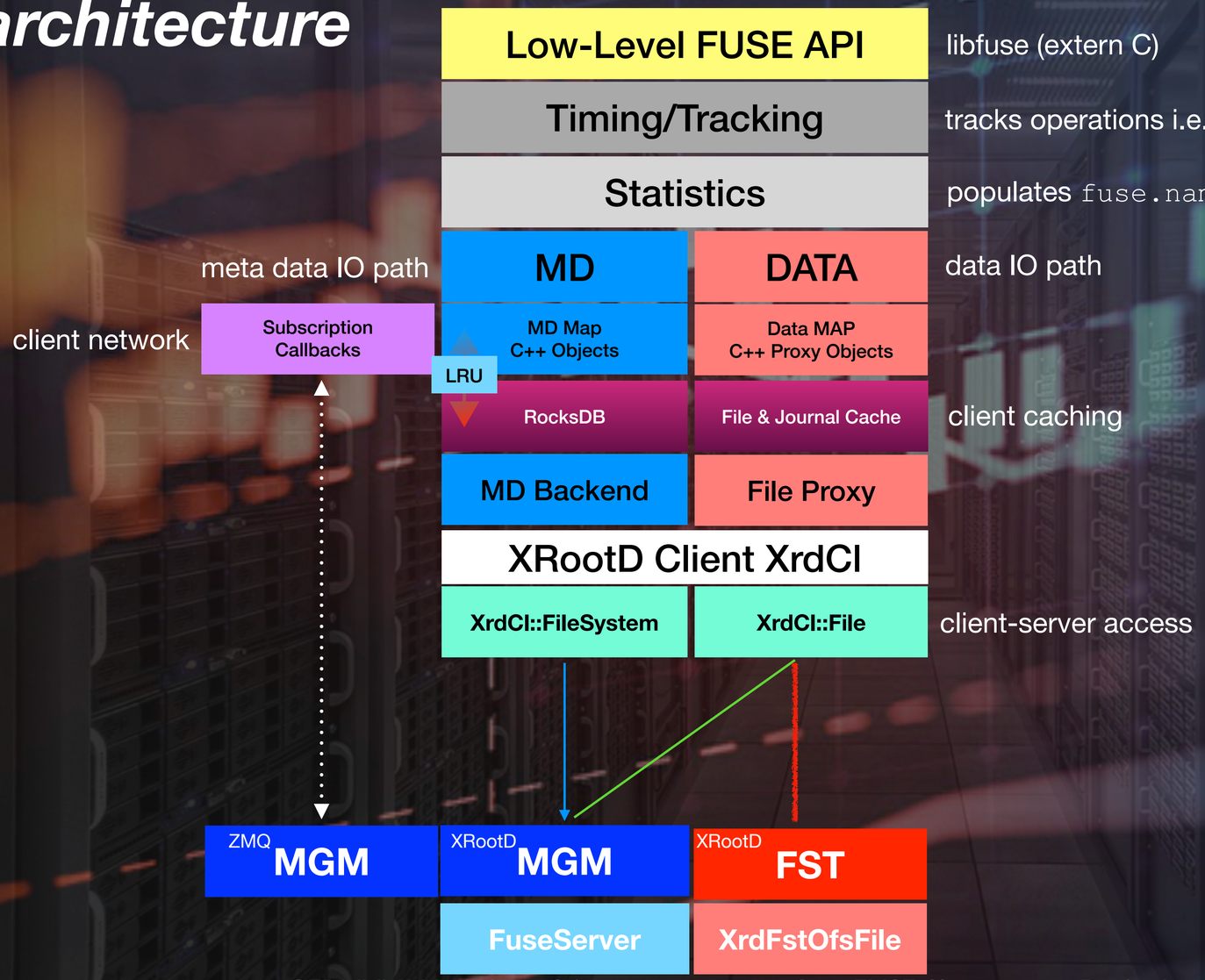
eosxd (EOS FUSE daemon) ***genealogy***





How EOS FUSE works …

eosxd *architecture*



Fuse Authentication / Authorization

- **EOS FUSE** supports various authentication mechanisms
  - **Kerberos** 5 (FILE, KEYRING)
  - GSI **Proxy** Certificates
  - Shared **Secrets**
  - **UNIX** (uid,gid) based authentication
  - **OIDC** (OAuth2 token)
  - **JWT** Token (SciTokens)
  - **EOS** Token

- The available AUTHs methods are pre-set by configuration
  - run-time AUTH configuration is **read from process environment** of the calling process or its parent process e.g. **KRB5CCNAME**
  - Heuristic to resolve dead-lock occurring when reading environment of a forking process

10

Fuse Authentication / Authorization

- **EOS FUSE** uses available credentials in configured order to create authenticated connections to the namespace server
  - authenticated connections are bound to process IDs

- Permissions are always delegated from server-side
  - either via **authenticated call** to namespace or **delegated capability** valid by default for 5 minutes until revoked
    - **every file open** is authorised locally and remote by the namespace
    - **every rm, mkdir, rmdir** is authorised locally and remote by the namespace
    - only **stat & ls** are authorised locally using a delegated capability

11

eosxd

Namespace FuseServer - central service for FUSE clients

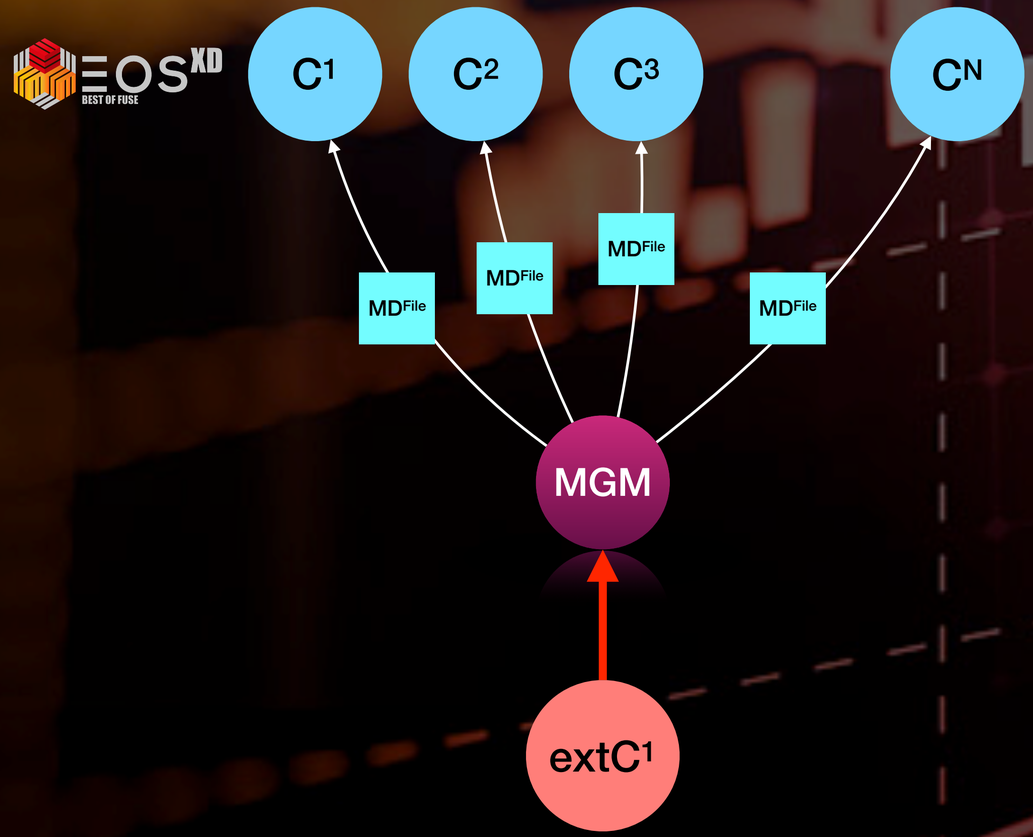
- The Fuse Server server-side CLI is used to …
  - show connected clients
    - and most important state information (stuck IO etc.)
  - evict unwanted clients
  - show notification subscriptions and permission delegation (capabilities)
  - drop state information
    - capabilities (directory subscriptions)
    - locks (delegated locks given to a client)
  - configure heartbeat and broadcast configuration
    - how often each client has to send a heartbeat
    - audience suppression to avoid too high callback rates

eosxd

Broadcasts & Callbacks

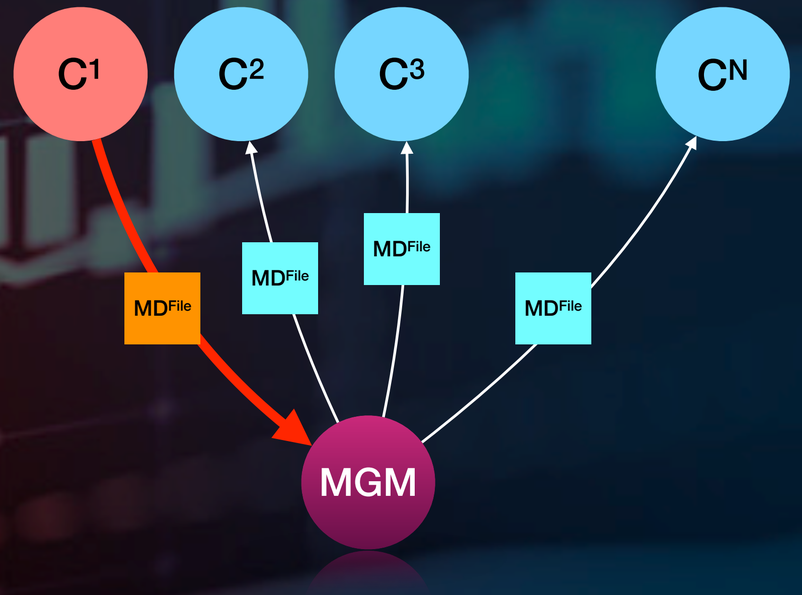
external broadcasts

an external Client (not FUSE) triggers a broadcast of file meta-data



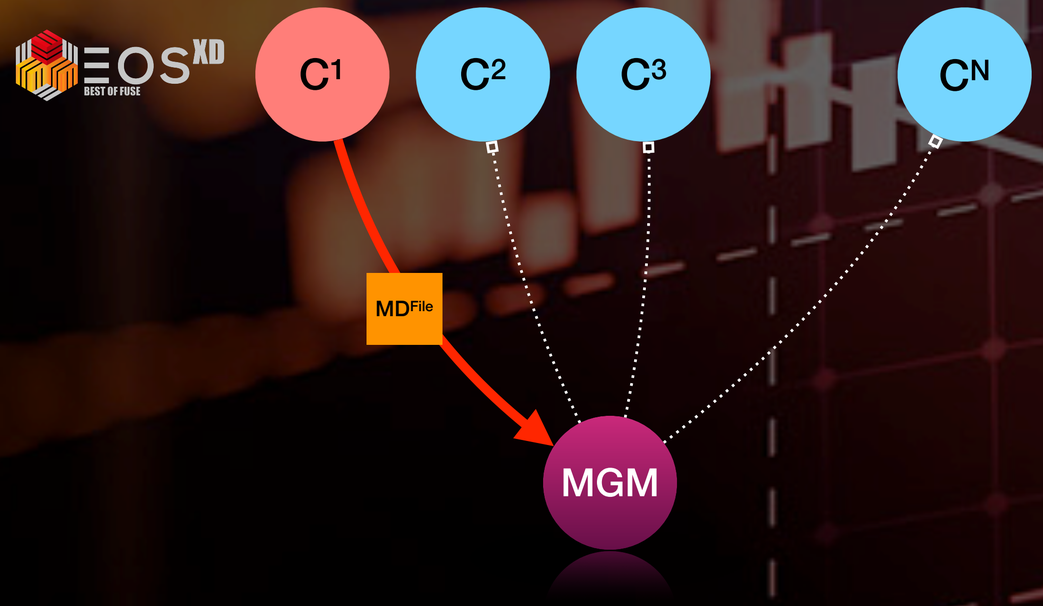
internal broadcasts

an internal client triggers a broadcast of file meta-data



eosxd

Broadcast Suppression



- an internal client would trigger a broad cast of file meta-data
- if the receiving audience is too large and the broadcast gets suppressed
- the not updating clients keep with an unsynchronised state until their subscriptions expire and a directory refresh is done
  - up to 300s by default

eosxd

Namespace FuseServer

- admin CLI for managing client mounts

eos fusex ls	list client nodes
eos fusex ls -l	list client nodes and some statistics per client
eos fusex evict	force unmount, trigger a stack trace, ask for logfile, truncate logfiles
eos fusex conf	configure heartbeat interval configure audience suppression
eos fusex caps	show directory subscriptions
eos fusex dropcap(s)	drop one or all subscriptions from a client
eos fusex droplocks	drop all locks for given inode and process

eos fusex ls -l

```
client : eosxd                    lxplus700.cern.ch 5.1.14   online
......   ino           : 40868
......   ino-to-del    : 0
......   ino-backlog   : 0
......   ino-ever      : 511893
......   ino-ever-del  : 18426
......   threads       : 60
......   total-ram     : 29.988 GB
......   free-ram      : 0.280 GB
......   vsize         : 2.016 GB
......   rsize         : 0.193 GB
......   wr-buf-mb     : 0 MB
......   ra-buf-mb     :24 MB
......   load1         : 19.33
......   leasetime     : 300 s
......   open-files    : 22
......   logfile-size  : 18576
......   rbytes        : 20397308261
......   wbytes        : 24307570949
......   n-op          : 4656820
......   rd60          : 0.00 MB/s
......   wr60          : 0.33 MB/s
......   iops60        : 303.83
......   xoff          : 830
......   ra-xoff       : 0
......   ra-nobuf      : 9489
......   wr-nobuf      : 0
......   idle          : 24
......   blockedms     : 0.00
```

eosxd

Client side Disk Caching

File & Journal Cache

```
"cache" : {
    "type" : "disk",
    "size-mb" : 512,
    "size-ino" : 65536,
    “file-cache-max-kb”,256,
    “file-journal-max-kb”, 131072
    "journal-mb" : 2048,
    "journal-ino" : 65536,
    "clean-threshold" : 85.0,
    "location" : "/var/cache/eos/fusex/cache/",
    "journal" : "/var/cache/eos/fusex/journal/",
    "read-ahead-strategy" : "static",
    "read-ahead-bytes-nominal" : 262144,
    "read-ahead-bytes-max" : 2097152,
    "read-ahead-blocks-max" : 16,
    "max-read-ahead-buffer" : 134217728,
    "max-write-buffer" : 134217728
  }
```

- There are **two data caches** which are persisted while a a mount is still up
  - **journal cache**
    - journalling writes
      (and also reads if a file has `attr:sys.file.cache=1`)
      with 128MB max. size per file
      - is used to recover writes
  - **file (start) cache**
    - caches the first 256kb of each file
      - triggered when a file is opened

eosxd

Virtual/Functional Attributes

- client CLI to interact with the mount

Information/Getter

Virtual Ext. Attribute	Meaning
eos.btime	file birth time
eos.ttime	latest modification time in subtree
eos.tsize	total size of directory subtree here
eos.dsize	**total size of files in this directory**
eos.dcount	number of subdirectories here
eos.fcount	number of files here
eos.name	EOS instance name
eos.hostport	EOS hostname:portname
eos.mgmurl	EOS instance URL root://…
eos.url.xroot	EOS XRootD TURL of the given file
eos.stats	get statistics output file
eos.quota	get quota output for the calling user

Authentication IF/Getter

Virtual Ext. Attribute	Meaning
eos.identity	get identity of the calling client
eos.identityparent	get identity of the calling client for the parent directory
eos.reconnect	force to drop current connection and create a new one
eos.reconnectparent	as above but for the parent

Modifications / Setter

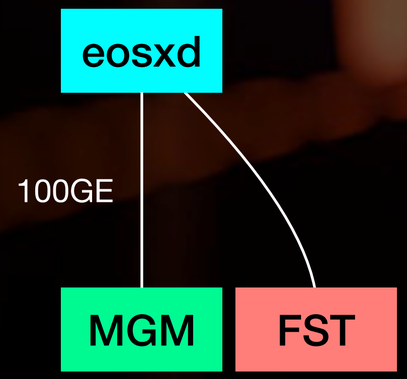
Virtual Ext. Attribute	ARG	Meaning
system.eos.debug	<level>	Set debug level
system.eos.dropcap	-	Drop subscription here
system.eos.dropcaps	-	Drop all subscriptions
system.eos.resetstat	-	Reset Stat Counter
system.eos.log	private|public	Change Logfile Perm.
system.eos.fuzz	all/config	Toggle Fuzzing mode

eosxd vs eosxd3
changes

- `readdirplus`
  - the function returning a directory listing attaches the stat information for each listed entry
    - avoids additional `getattr` calls for each listed child!
- `forgetmulti`
  - when the kernel wants to forget inodes, instead of recalling one by one a bulk request reduces the number of `forget` calls substantially
- `clone_fd` option
  - each FUSE thread creates a separate device file descriptor for each processing thread, which might improve performance.
- `write_back_cache` option
  - small writes are not hitting directly eosxd, but they are written into the buffer cache and then flushed as larger operations
  - increases single byte write performance from 16kHz to 0.5MHz
  - but does not allow remote invalidation
- `FUSE_SET_ATTR_CTIME`
  - allows to set the change time with the `utimes` function

eosxd3
Performance

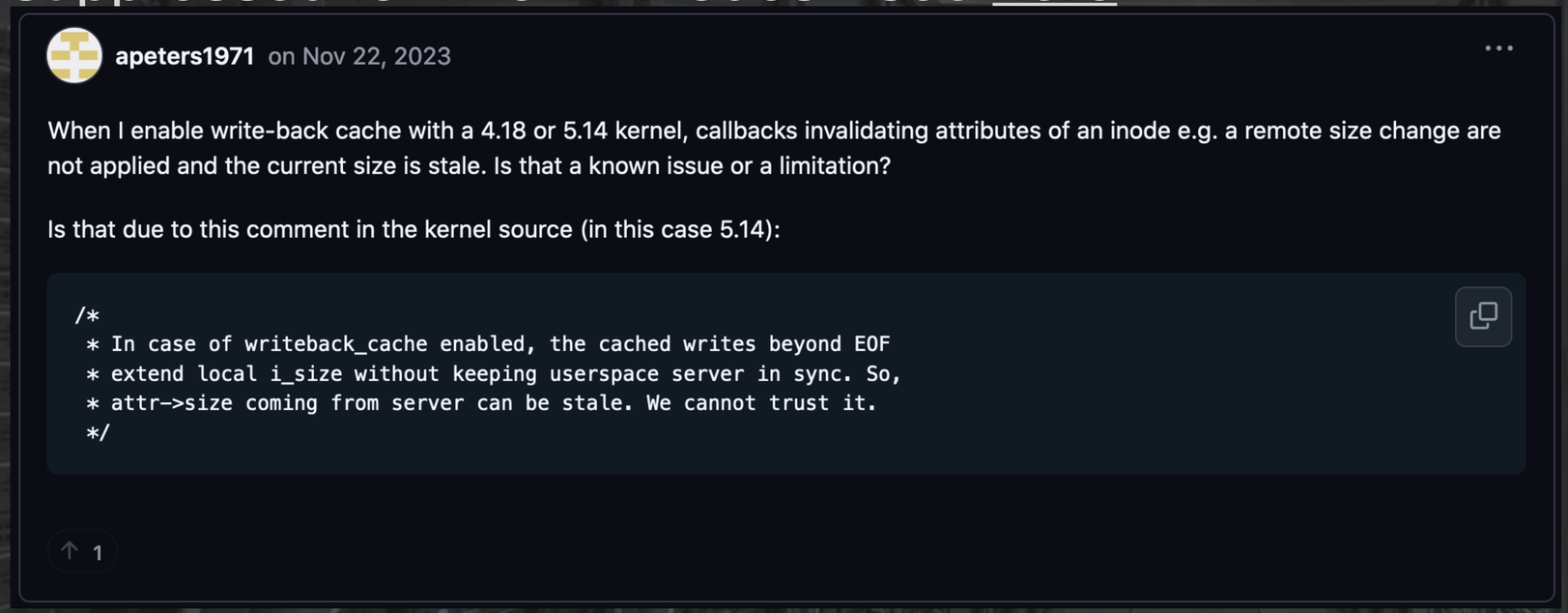
- 100 GE client/server idle instance test



Single Client	eosxd	eosxd3
Seq.Creation		**700** Hz
Par. Creation		**1000** Hz
w IOPS	**16 kHz**	**500 kHz**
Seq. Read		**1.6** GB/s
Seq. Write		**900** MB/s

EOS Fuse Problems

- FUSE3: write-back cache cannot be used because callbacks are suppressed for known inodes - see here



apeters1971 on Nov 22, 2023

When I enable write-back cache with a 4.18 or 5.14 kernel, callbacks invalidating attributes of an inode e.g. a remote size change are not applied and the current size is stale. Is that a known issue or a limitation?

Is that due to this comment in the kernel source (in this case 5.14):

```
/*
 * In case of writeback_cache enabled, the cached writes beyond EOF
 * extend local i_size without keeping userspace server in sync. So,
 * attr->size coming from server can be stale. We cannot trust it.
 */
```

↑ 1

- double-mounting on ALMA9 when using autofs
  - when moving from CENTOS7 to ALMA9 we had to add protection to avoid mounting a filesystem twice

20

EOS Fuse Outlook

- most relevant performance limits in **eos**xd are addressable in MGM + FST implementation - FUSE bottlenecks play only minor role in our environment

- **eos**xd would benefit from XATTR caching in kernel - which does not exist

- An architecture where **meta-data** caches can be directly shared between user-space and kernel would be desirable and simplify the user-space implementation - had a look at extFUSE

21

eoscfsd

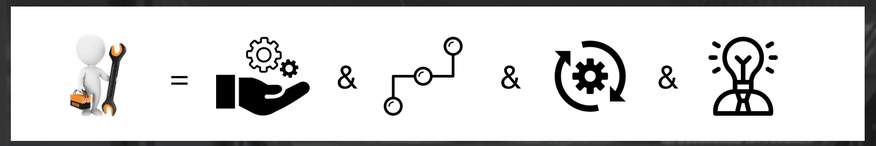
eoscfsd

CLIENT FILE SYSTEM

22

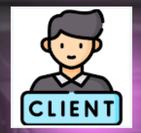
What is CFS?

CLIENT FILE SYSTEM



- **CFS** is a **FS abstraction layer written as a FUSE passthrough filesystem**
  - provides homogeneous POSIX API connecting to storage systems
    - as much POSIX as the layer it abstracts + tiny FUSE POSIX violations
  - allows to support arbitrary authentication methods for any back-end
  - allows shared extensions to standard POSIX API
  - provides an exit strategy from any back-end
- a tiny R&D project - which is now part of EOS releases and usable as **eos**cfsd

Architecture



core comes from example implementation of libfuse-3

low-level FUSE API

FUSE Passthrough Filesystem

convenience functionality

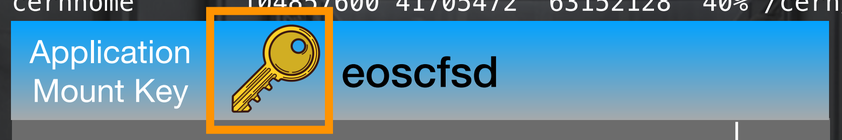
Mountable Filesystem

anything which can be mounted

CFS Prototype

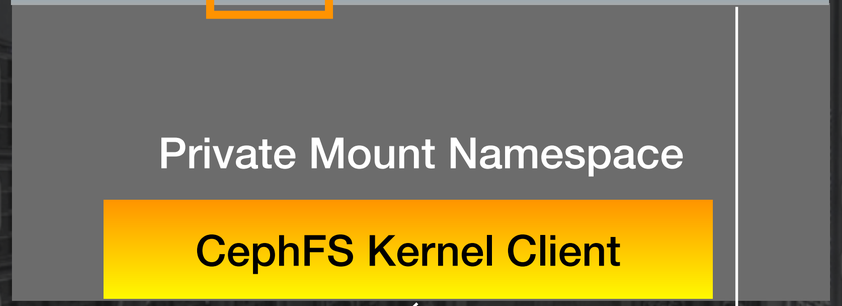
eoscfsd provides a high-performance pass-through implementation for POSIX filesystems. It adds kerberos authentication, remote configuration and mount key obfuscation. **eos**cfsd fetches the mount instructions for a named mount from a configurable HTTPS server.

```
[root~]# df /cern/home/
Filesystem     1K-blocks     Used Available Use% Mounted on
cernhome       104857600 41705472  63152128  40% /cern/home
```



no caching in kernel FUSE layer
read/write redirect to file descriptor

Local Mount Key **/etc/eoscfsd.key**



back-end invisible/inaccessible for
root user

https GET (1)

These keys don't remove the need to trust *root* on a host!

mount (2)

FS1 .. N

Encrypted Mountpass

ceph OneDrive Drive

25

Prototype Status

- RPMs available - package **eos-cfsd**
  - **kerberos** authentication
  - virtual **/.proc/** interface
    - kerberos ID to name **translation**
    - ***quota*** bool per user
    - ***enabled*** bool per user
    - **recycle** bin (enable/disable during mount)
    - **bulk deletion** wrapper via shell alias
      (remark: a shell wrapper opens up the possibility to use the recycle bin as free storage resource .. maybe we don't want to allow that)
  - virtual xattr 'cfs.id' = 'who am i'
  - statistics file in JSON format with operations/s etc.
  - autofs support

CFS Documentation

26

Performance

- unpacking the Linux kernel on CentOS9 office desktop with CephFS disk-based backend /cern/home/
  - cephfs native backend: untars ~ **1100** files/s
  - eos**cfs**d: untars ~**1000** files/s [90% of back-end]
  - afs: untars ~**250** files/s
- IO bottleneck introduced by FUSE is almost invisible
  (reading a file inside the CephFS kernel cache via eos**cfs**d)

```
[apeters@engine apeters]$ dd if=1GB of=/dev/null bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 0.311914 s, 3.4 GB/s
```

27

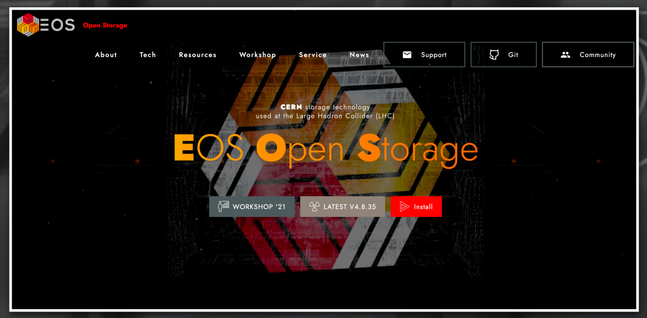
Performance Outlook

- In kernel version **< 6.9.** FUSE **passthrough** layer receives all read/write requests and redirects them to the back-end filesystem file descriptor
- In kernel version 6.9 FUSE **passthrough** layer can be bypassed for all read/write operations completely - which means you experience the native IOPS of the backend (IOPS limit we see in FUSE 16-20kHz)
- **eos**cfsd is a promising platform to create a back-end agnostic filesystem abstraction with customisable functionality add-ons
- If you are interested - don't hesitate to reach-out, test and/or contribute!

28

Useful Links

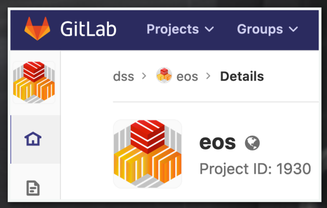
Web Page https://eos.cern.ch



29

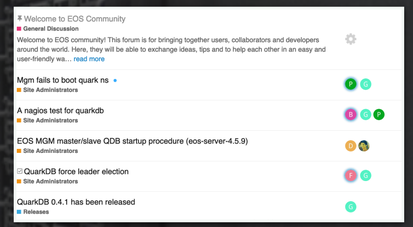
GITLAB Repository https://gitlab.cern.ch/dss/eos

GITHUB Mirror https://github.com/cern-eos/eos



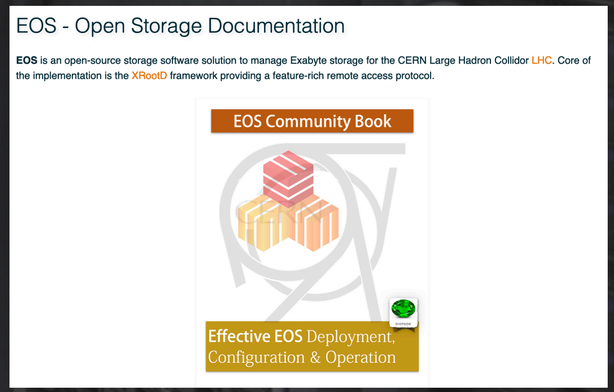
Community Forum https://eos-community.web.cern.ch/

email: eos-community@cern.ch



Documentation http://eos-docs.web.cern.ch/eos-docs/

Support email: eos-support@cern.ch



29


Thank you for your attention!
Questions?
SOCAR
VESTIAR
CHANGE