EOS architectural evolution and strategic development directions

Georgios Bitzes, Fabio Luchetti (CERN), Andrea Manzi (CERN) Mihai Patrascoiu,
Andreas Joachim Peters (CERN), Michal Kamil Simon (CERN), Elvin Alin Sindrilaru (CERN)

CHEP 2019

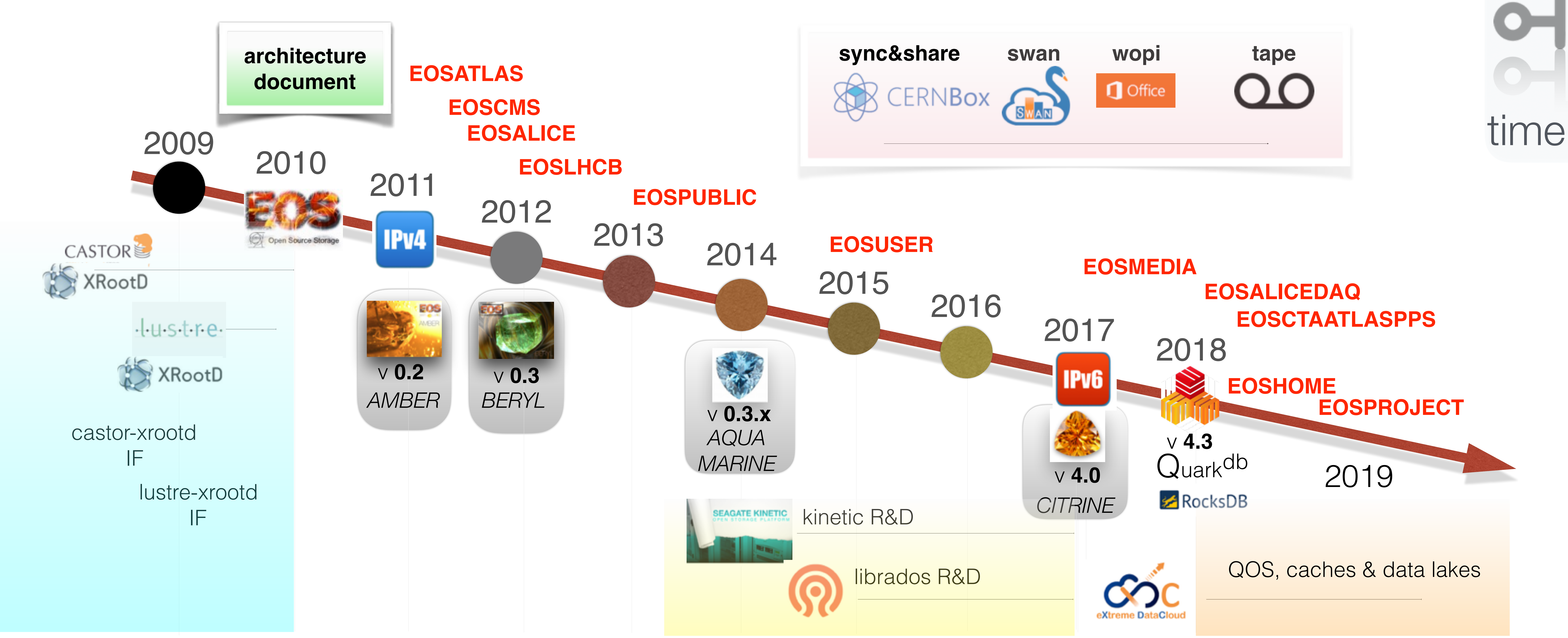
24th International Conference on Computing in High Energy & Nuclear Physics

Andreas-Joachim Peters
CERN IT Storage Group

Overview

- Introduction
- Architecture Evolution
- New Features
- Directions
- Summary & Outlook

Project History



architecture document

2009

castor-xrootd IF

lustre-xrootd IF

2010

EOSATLAS

EOSCMS

EOSALICE

2011

v 0.2 AMBER

2012

v 0.3 BERYL

EOSLHCB

EOSPUBLIC

2013

2014

v 0.3.x AQUA MARINE

EOSUSER

2015

kinetic R&D

librados R&D

2016

EOSMEDIA

2017

v 4.0 CITRINE

EOSALICEDAQ

EOSCTAATLASPPS

2018

v 4.3

EOSHOME

EOSPROJECT

2019

sync&share

swan

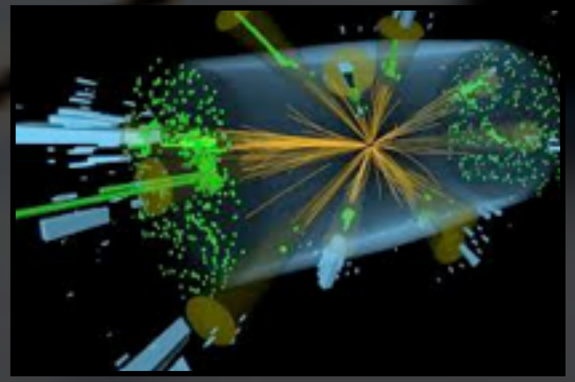
wopi

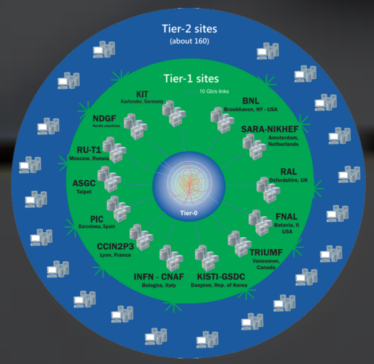
tape

QOS, caches & data lakes

timeline

What is EOS used for ...





- disk storage
    - raw data
    - analysis data
    - cernbox home & project spaces
    - cloudstore AARNet, Joint Research Centre JRC
    - Tier 2 & universities
    - online systems
- tape storage cache
    - Cern Tape Archive

Development Work Areas

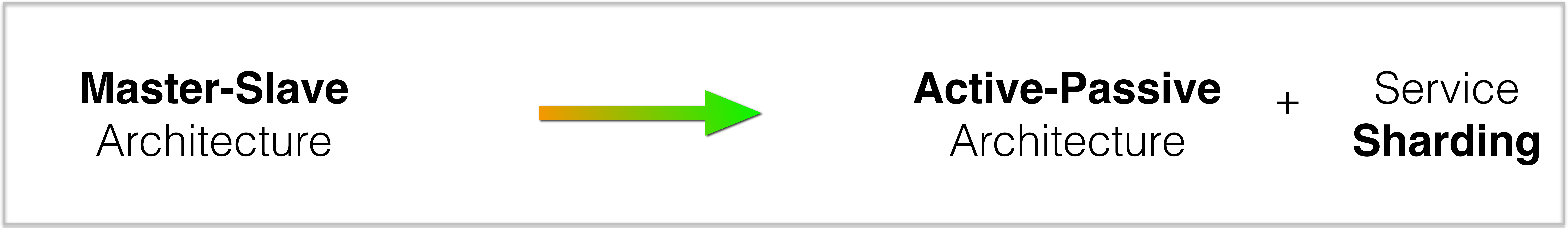
- **namespace architecture** (MGM)
- **storage consistency** (FST)
- **filesystem access** (eosxd/ACLs)
- **tape integration** (CTA)
- **protocols/API** (ProtBuf,XrdHttp,GRPC)
- **tokens & authorisation**

• tokens & authorisation

Architectural Evolution

EOS 2017

EOS 2019



Master-Slave Architecture

Active-Passive Architecture + Service **Sharding**

stateful
meta-data
service

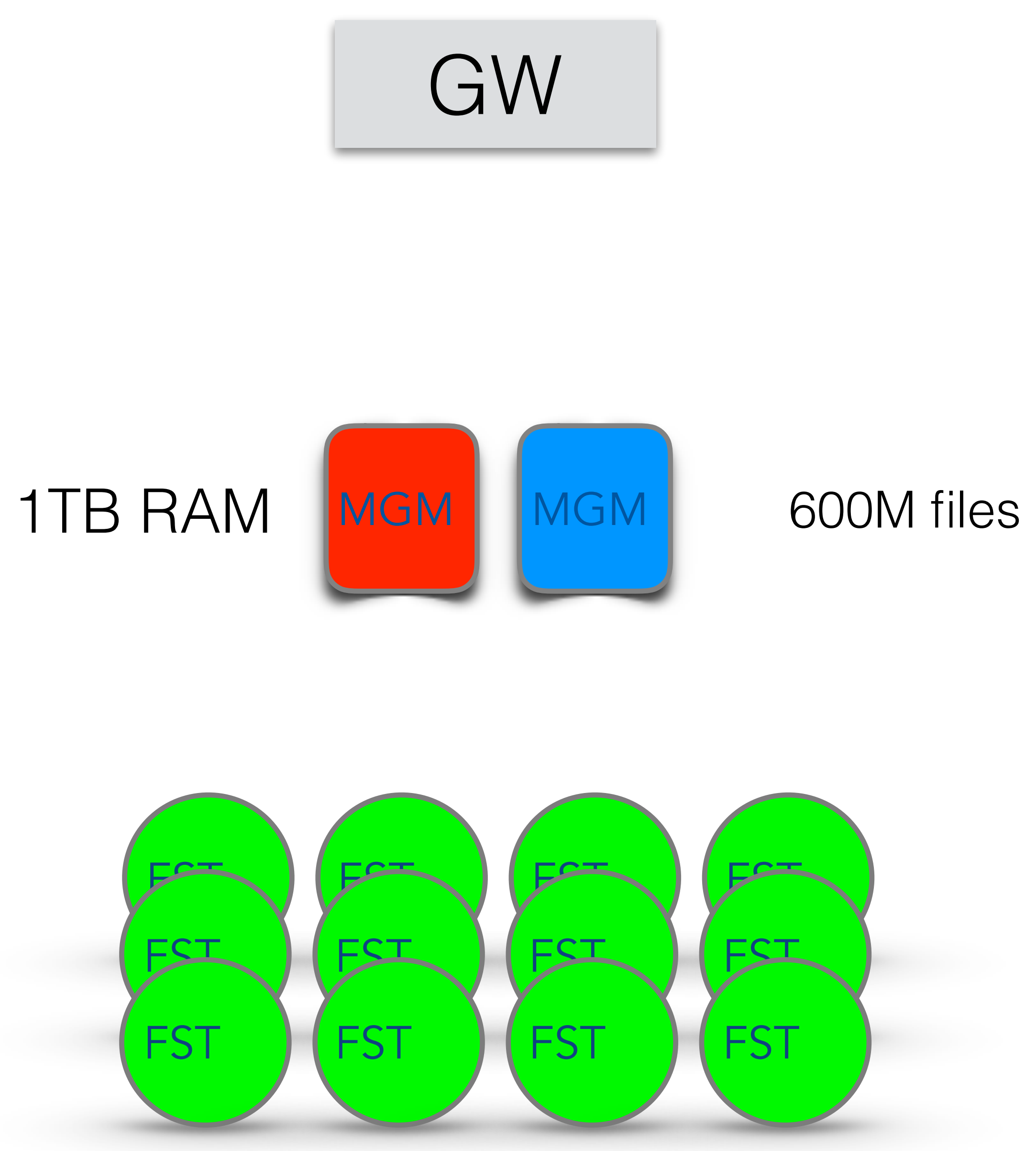
almost stateless
meta-data
service

scale-out
meta-data
performance

Architectural Evolution

CERNBOX 2017

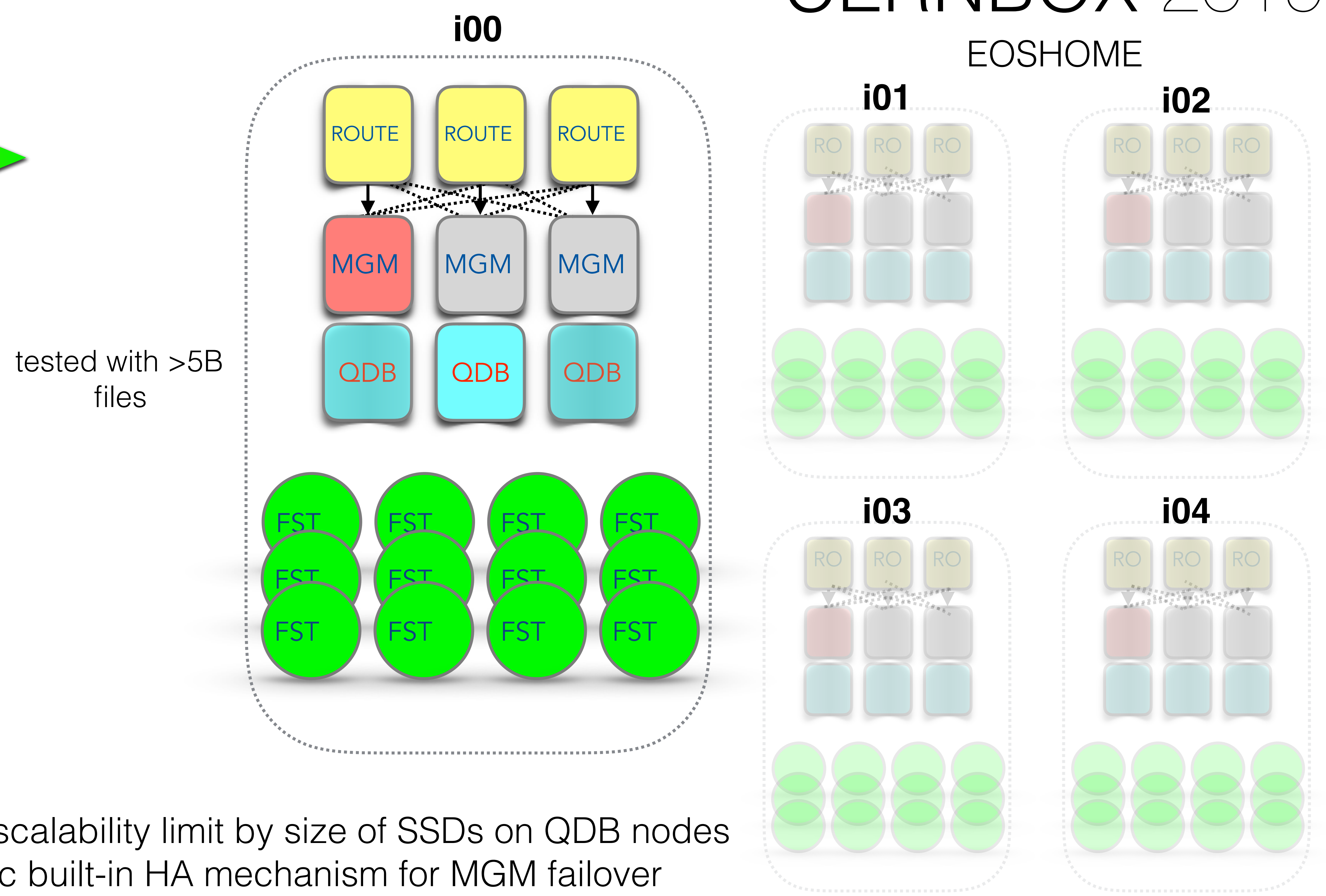
EOSUSER



at namespace scalability limit
availability constrained by infrequent long boot time of 2h

CERNBOX 2019

EOSHOME

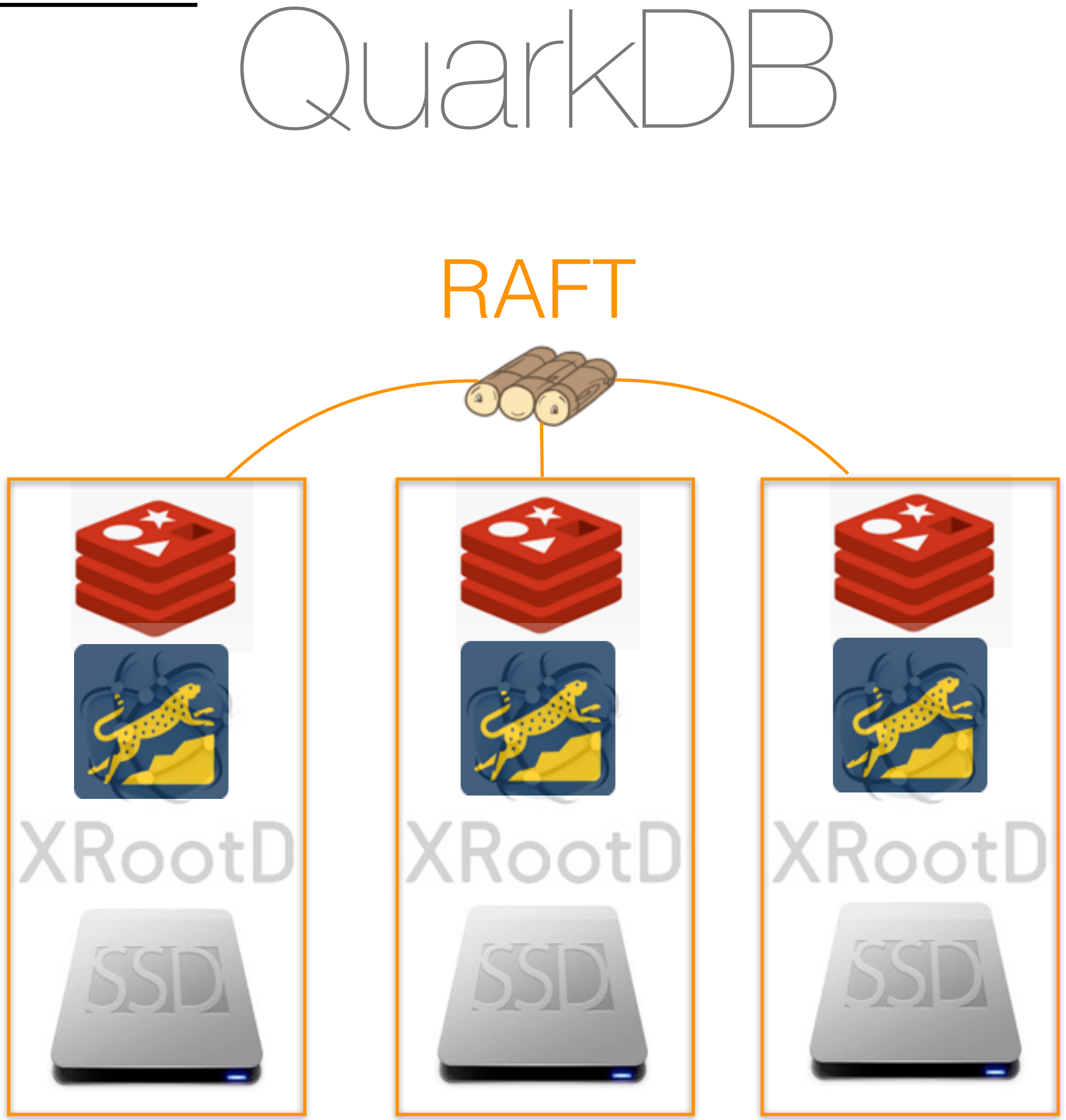


namespace scalability limit by size of SSDs on QDB nodes
automatic built-in HA mechanism for MGM failover

QuarkDB

https://github.com/gbitzes/QuarkDB

- **Introduction of QuarkDB as persistent KV store for namespace meta-data**
  - based on **REDIS** protocol, **RocksDB** & **RAFT** consensus algorithm
  - high-**available**, high-**performant**, **scalable**, low-**latency**
  - **extremely positive** production **experience**



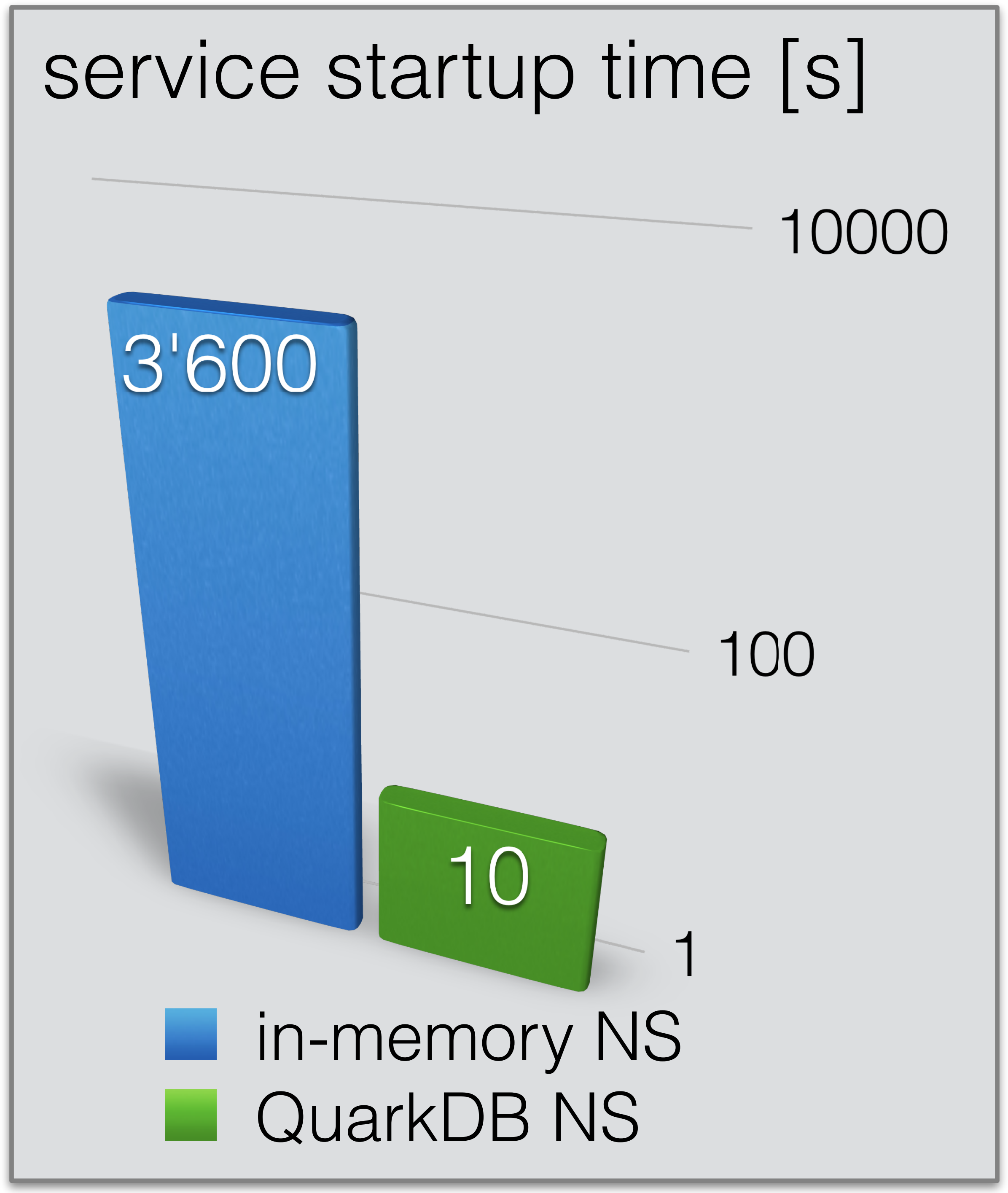
QDB api
- kv
- sets
- hashes
- pub-sub
- lease

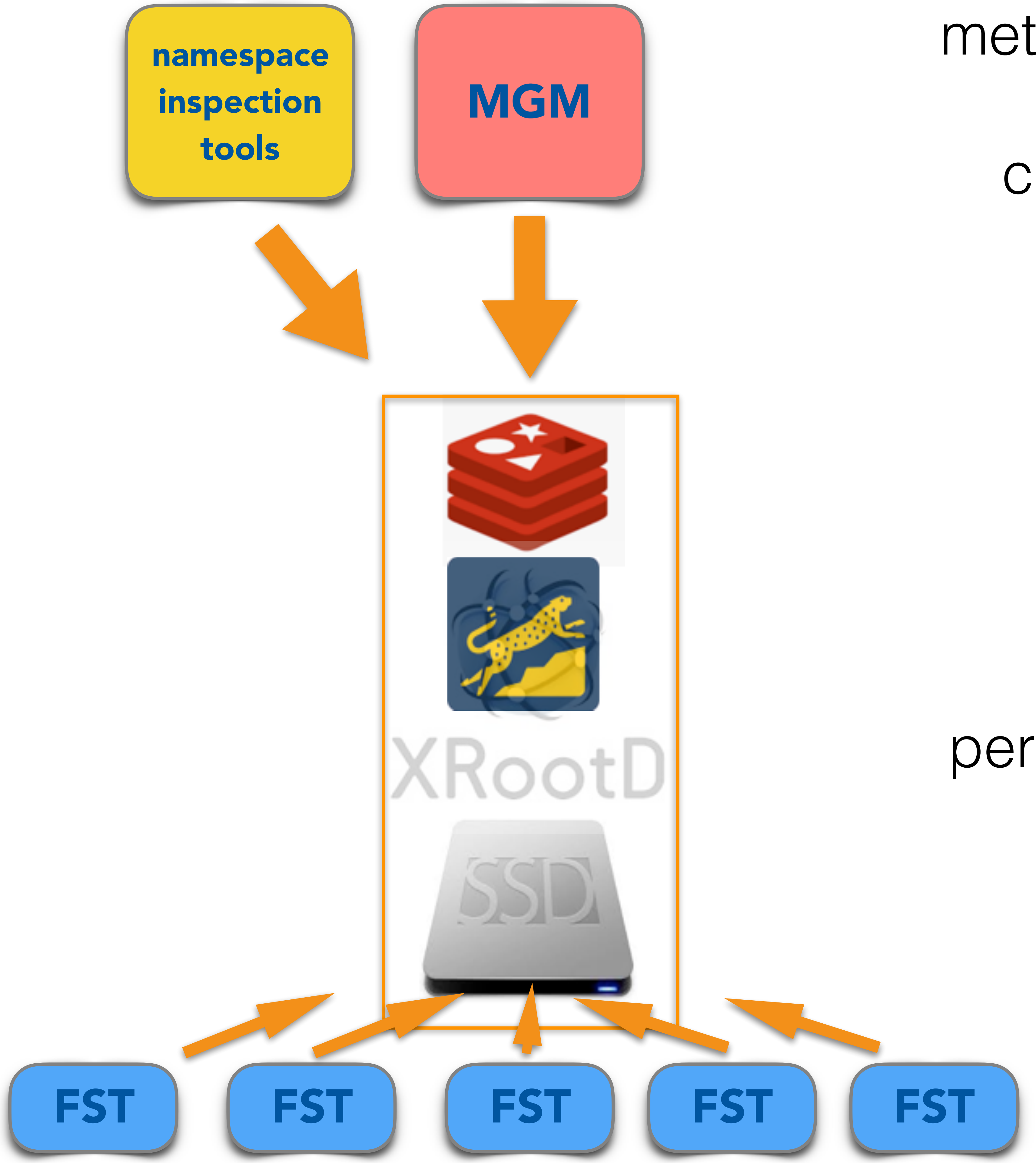
C++ client library
https://gitlab.cern.ch/eos/qclient

QDB performance example: retrieve KV@200kHz

QuarkDB Namespace

- service **startup time** was major **source** for service **downtime** for in-memory namespace





meta-data service **MGM** stateless with configurable cache

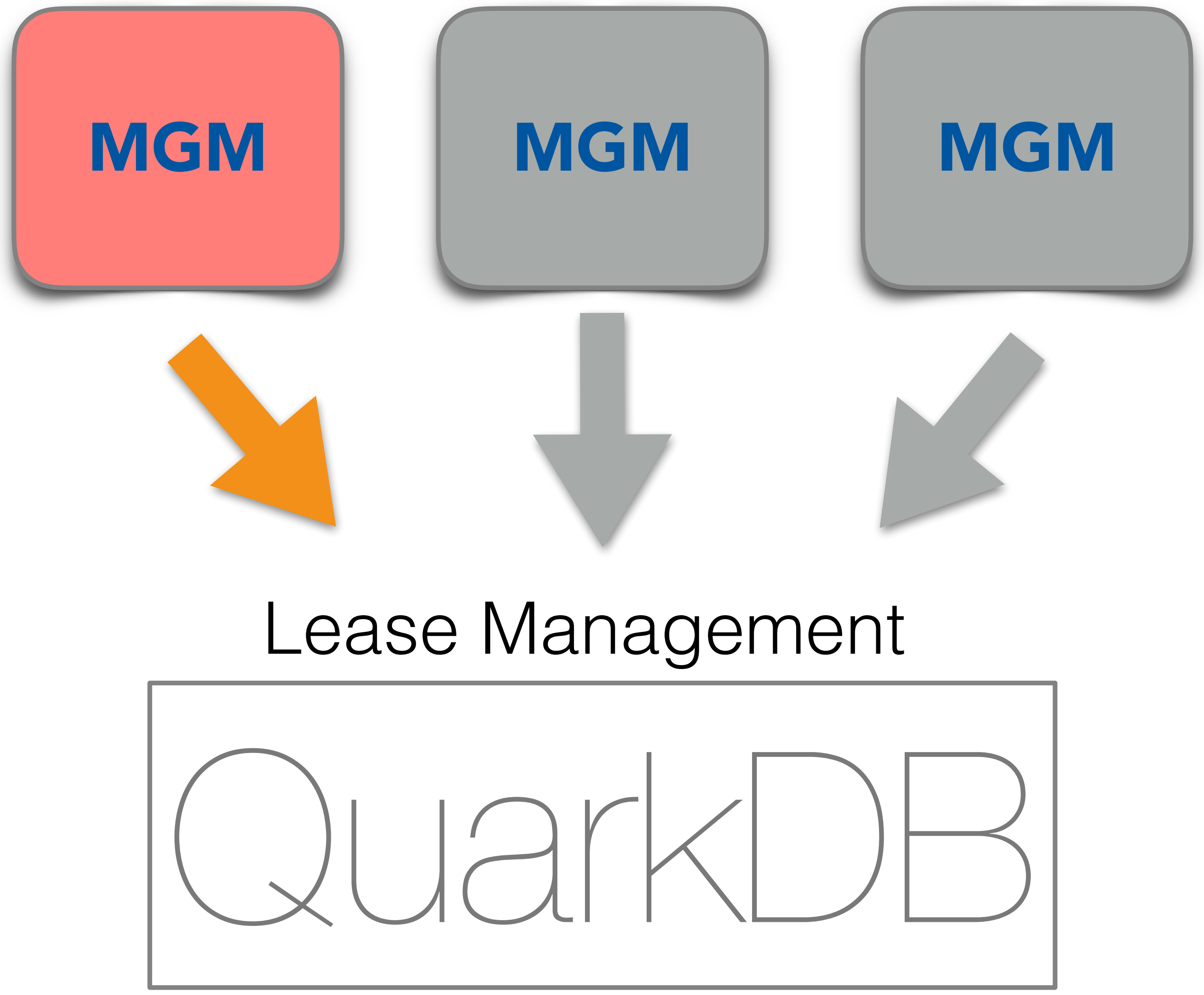
meta-data persistency with **QuarkDB**

storage server **FSTs**

QuarkDB for HA

QDB provides support for leases
to automatically fail-over meta-data server

- leases **renewed** every 5 seconds
- default **validity** is 10 seconds
- if a lease is required **configuration** is automatically **reloaded** and namespace becomes active
- service **fail-over within few seconds**



meta-data service **MGM**
stateless with
configurable cache

meta-data
persistency with **QuarkDB**

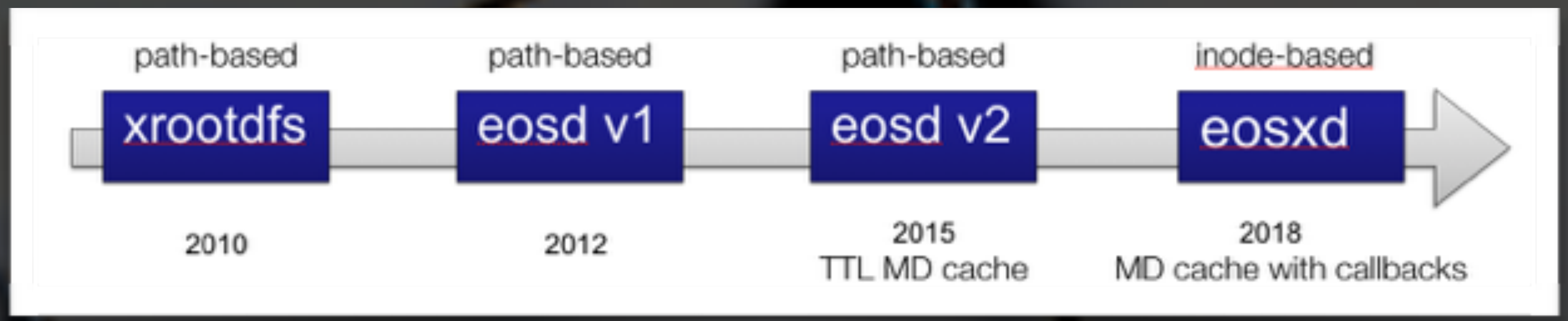
File System Consistency

EOS v4.6

- re-engineering of **FSCK** functionality
  - over the past 9 years accumulated replication inconsistencies in EOS instances
  - with transition to QuarkDB filesystem consistency check & repair broken
- FSCK components
  - **backward consistency check:** compare filesystem contents to namespace - *size, checksum, layout*
    - data scanner with inconsistency flagging & checksumming for each filesystem - by default all data scanned within one week
  - **forward consistency check**: compare namespace to filesystem contents
    - MGM scanner identifying missing replicas on filesystems
  - **repair engine** error collection & automatic repair actions

File System Access

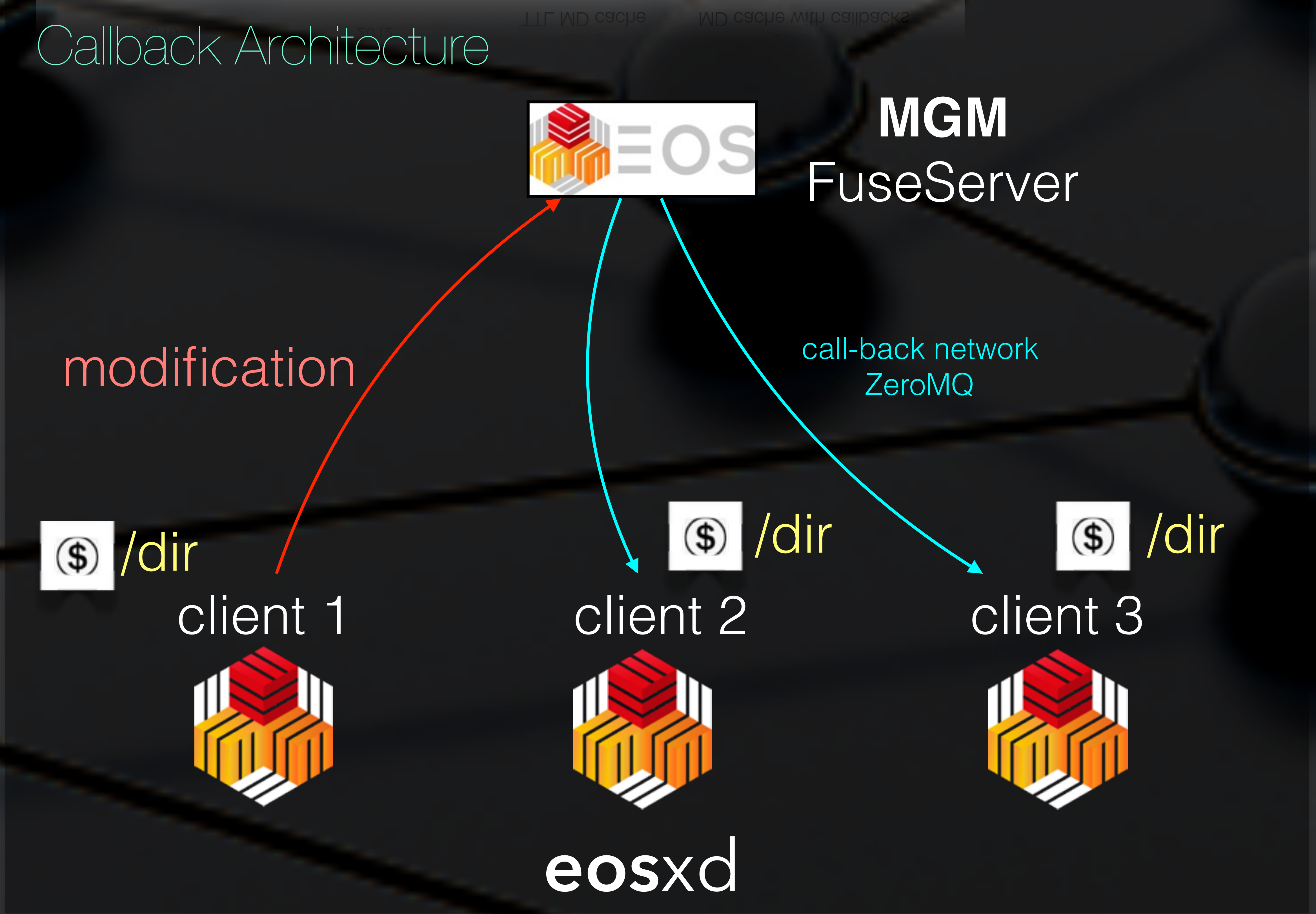
eosxd



avg. > 20k mount clients @ CERN for CERNBOX

eosxd

- better **POSIX**ness
- file **locks**, byte-range locks
- hard **links** within directories
- rich **ACL** client support
- local **caching & journaling**
- **bulk deletion**/protection
- strong **security**
- OIDC & Token support
- user,group & project **quota**
- based on **libfuse2**

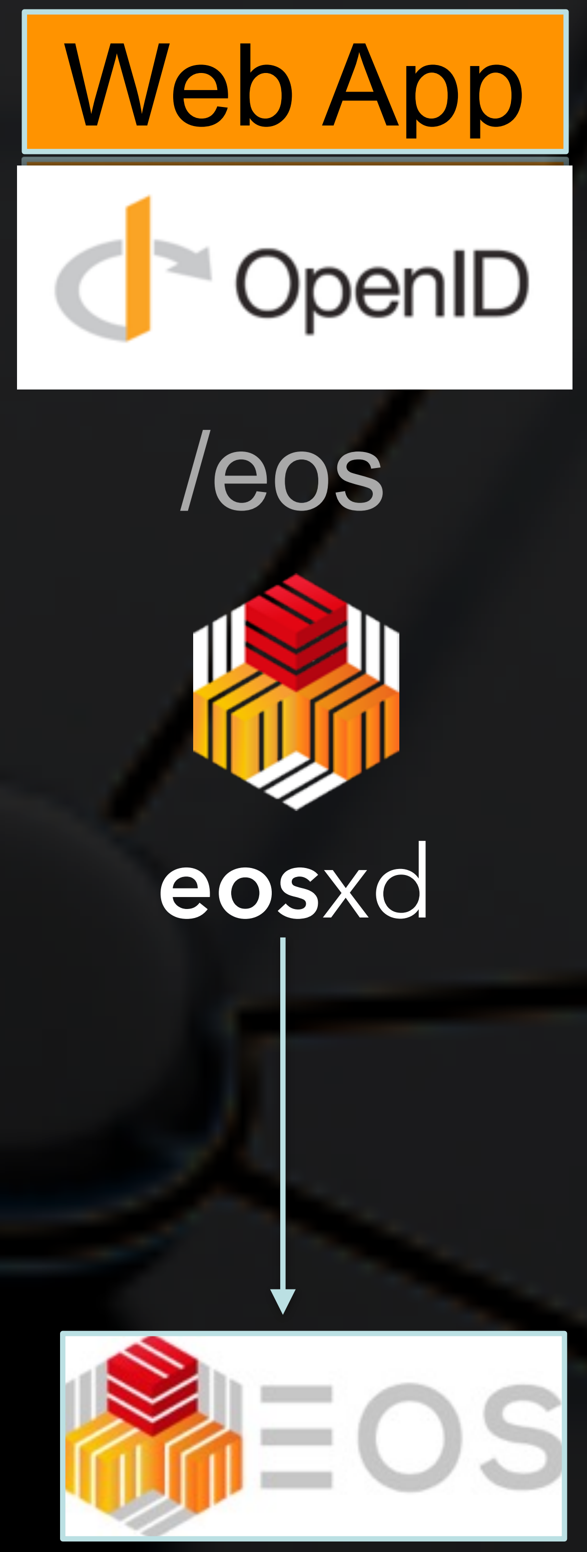


Latest developments

OIDC support as kerberos/x509 replacement

Snapshot support with COW functionality for consistent backups

Squashfs integration for software distribution



2019++

Tape Integration

EOS + Tape = EOSCTA

integrated support for tape into EOS file on tape=offline replica

- loose service coupling between EOS and CTA via protocol buffer interface & notification events - everything is synchronous
- no SRM, using XRootD protocol only - integrated with FTS

high disk capacity

EOSATLAS

TPC

File Transfer Service at CERN

Open source software to transfer data reliably and at large scale between storage systems

low disk capacity

EOSATLASCTA

short file lifetime

Operation Model

Cern Tape Archive

Protocol Support

GRPC support with token and x509 support

- mapping applications identity using GRPC token=>(uid,gid) or DN=>(uid,gid) mapping

Namespace interface

- metadata injection - used for Castor=>CTA meta-data migration
- mkdir|rmdir|touch|rm|unlink|ls|find|rename|symlink|setxattr|chown|chmod|acl|token|create-version|list-version|purge-version with streaming support for large responses

HTTP(S) support with token and x509 support

- using XrdHttp and external handler

HTTP TPC / XRootD with delegation support

- using default proxy server in front of EOS instances on gateway machines

S3 support with MINIO gateway

- via plug-in for MINIO developed by AARNet - *currently not deployed at CERN*

EOS Tokens

Bearer Token Support preparing coming WLCG authz changes

Proprietary format

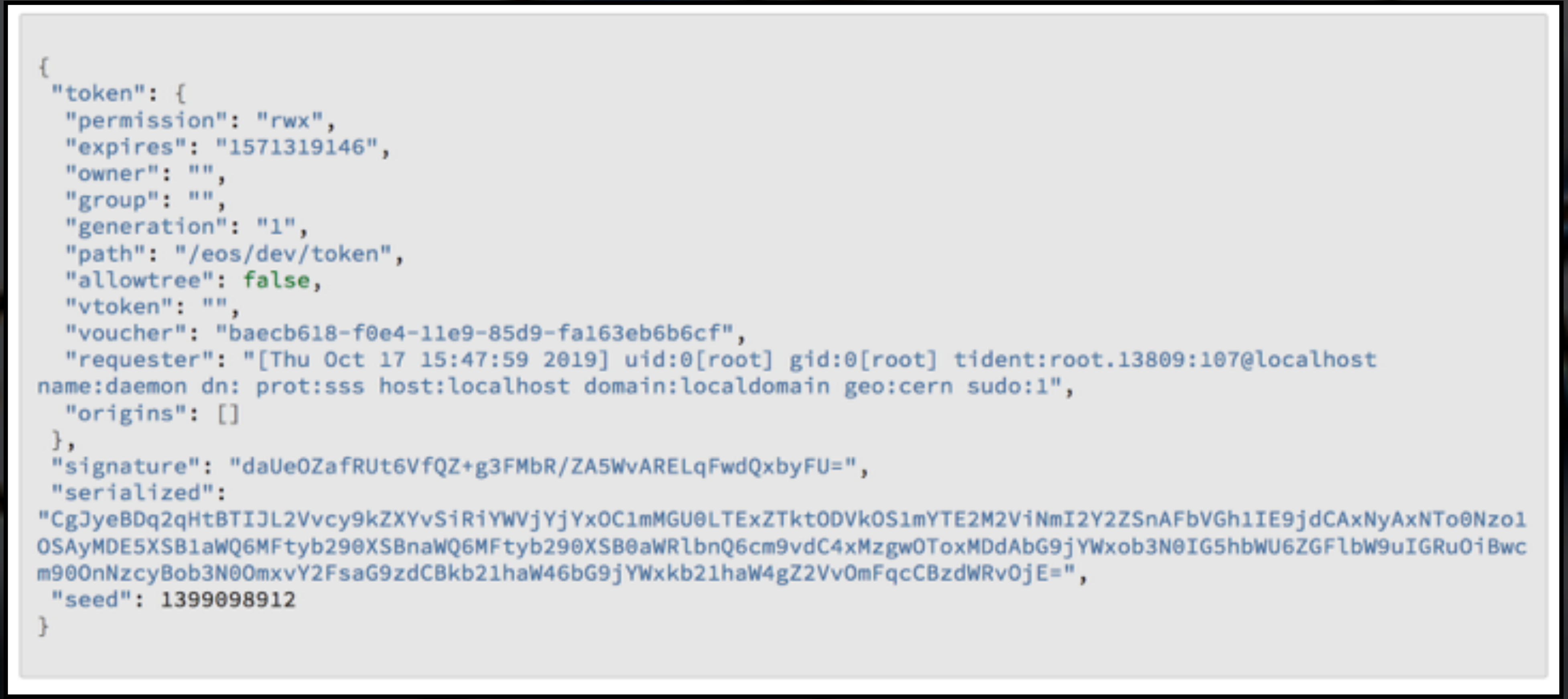
- Serialized **PROTOBUF** structure + **ZLIB** Compression + **Base64URL** encoding

Token carries

- a namespace scope file, directory or tree
- an ACL entry replacing locally stored ACLs - no need to invent new syntax like UPLOAD, DOWNLOAD…
- an optional role e.g. the owner when creating a file
- an optional set of origin restrictions - which clients can use this token and how do they have to be authenticated
  - we can enforce additional strong authentication if a bearer wants to use a token
- a generation value allows immediate token revocation of a given generation
- an expiration time

EOS Tokens

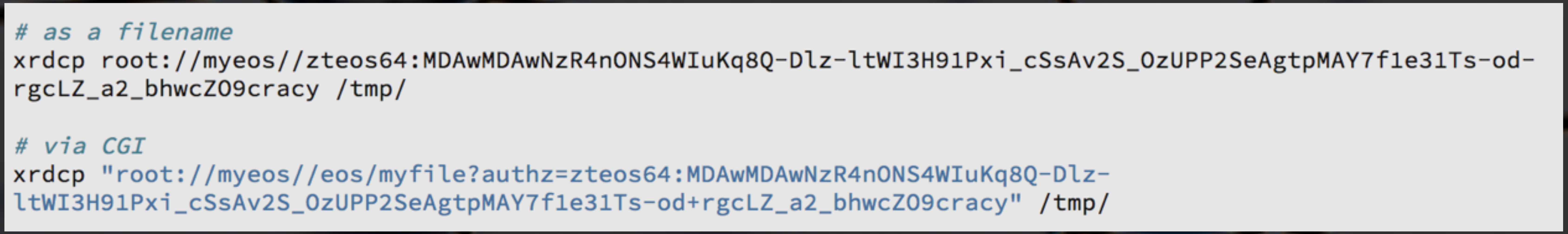
JSON representation



```
{
 "token": {
  "permission": "rwx",
  "expires": "1571319146",
  "owner": "",
  "group": "",
  "generation": "1",
  "path": "/eos/dev/token",
  "allowtree": false,
  "vtoken": "",
  "voucher": "baecb618-f0e4-11e9-85d9-fa163eb6b6cf",
  "requester": "[Thu Oct 17 15:47:59 2019] uid:0[root] gid:0[root] tident:root.13809:107@localhost
name:daemon dn: prot:sss host:localhost domain:localdomain geo:cern sudo:1",
  "origins": []
 },
 "signature": "daUeOZafRUt6VfQZ+g3FMbR/ZA5WvARELqFwdQxbyFU=",
 "serialized":
"CgJyeBDq2qHtBTIJL2Vvcy9kZXYvSiRiYWVjYjYxOC1mMGU0LTExZTktODVkOS1mYTE2M2ViNmI2Y2ZSnAFbVGh1IE9jdCAxNyAxNTo0Nzo1
OSAyMDE5XSB1aWQ6MFtyb290XSBnaWQ6MFtyb290XSB0aWRlbnQ6cm9vdC4xMzgwOToxMDdAbG9jYWxob3N0IG5hbWU6ZGFlbW9uIGRuOiBwc
m90OnNzcyBob3N0OmxvY2FsaG9zdCBkb21haW46bG9jYWxkb21haW4gZ2VvOmFqcCBzdWRvOjE=",
 "seed": 1399098912
}
```

Usage

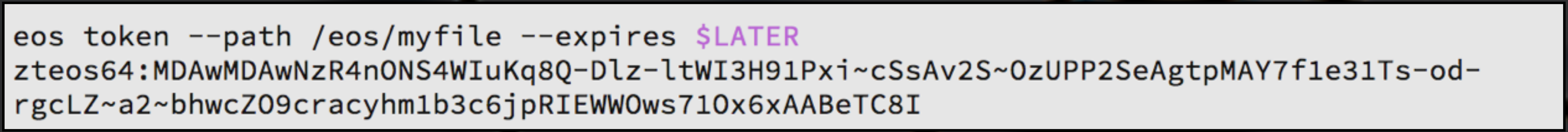
token **as filename** or **CGI** authz=<token> usable with **XRootD, HTTP, GRPC, eos**xd **(fuse)**



```
# as a filename
xrdcp root://myeos//zteos64:MDAwMDAwNzR4nONS4WIuKq8Q-Dlz-ltWI3H91Pxi_cSsAv2S_OzUPP2SeAgtpMAY7f1e31Ts-od-
rgcLZ_a2_bhwcZ09cracy /tmp/

# via CGI
xrdcp "root://myeos//eos/myfile?authz=zteos64:MDAwMDAwNzR4nONS4WIuKq8Q-Dlz-
ltWI3H91Pxi_cSsAv2S_OzUPP2SeAgtpMAY7f1e31Ts-od+rgcLZ_a2_bhwcZ09cracy" /tmp/
```

Creation



```
eos token --path /eos/myfile --expires $LATER
zteos64:MDAwMDAwNzR4nONS4WIuKq8Q-Dlz-ltWI3H91Pxi~cSsAv2S~OzUPP2SeAgtpMAY7f1e31Ts-od-
rgcLZ~a2~bhwcZ09cracyhm1b3c6jpRIEWWOws71Ox6xAABeTC8I
```

EOS Tokens

How can they be used?

- usable by applications for restricted ***on-behalf* access**
  via any supported access method - even fuse mounts

- can be used by CERNBOX services to provide shares and delegate permissions

- as internal format for **external tokens** WLCG/ALICE tokens

- as **single file token** like signed S3 URLs are used

http://eos-docs.web.cern.ch/eos-docs/using/tokens.html

General Directions

consolidation of new architecture, **improvement** of reliability & consistency and **optimisation** of internal storage services to profit from QuarkDB

look at **MD Scale-out** without service sharding subtree assignment to MGMs

support **HTTP** eco-system: establish **GRPC** as MD API,
DAV as Data API for front-end CERNBOX possibly also GRPC+flatbuffers as DATA API

establish/support **tokens** for applications and GRID access

focus on **erasure coding**

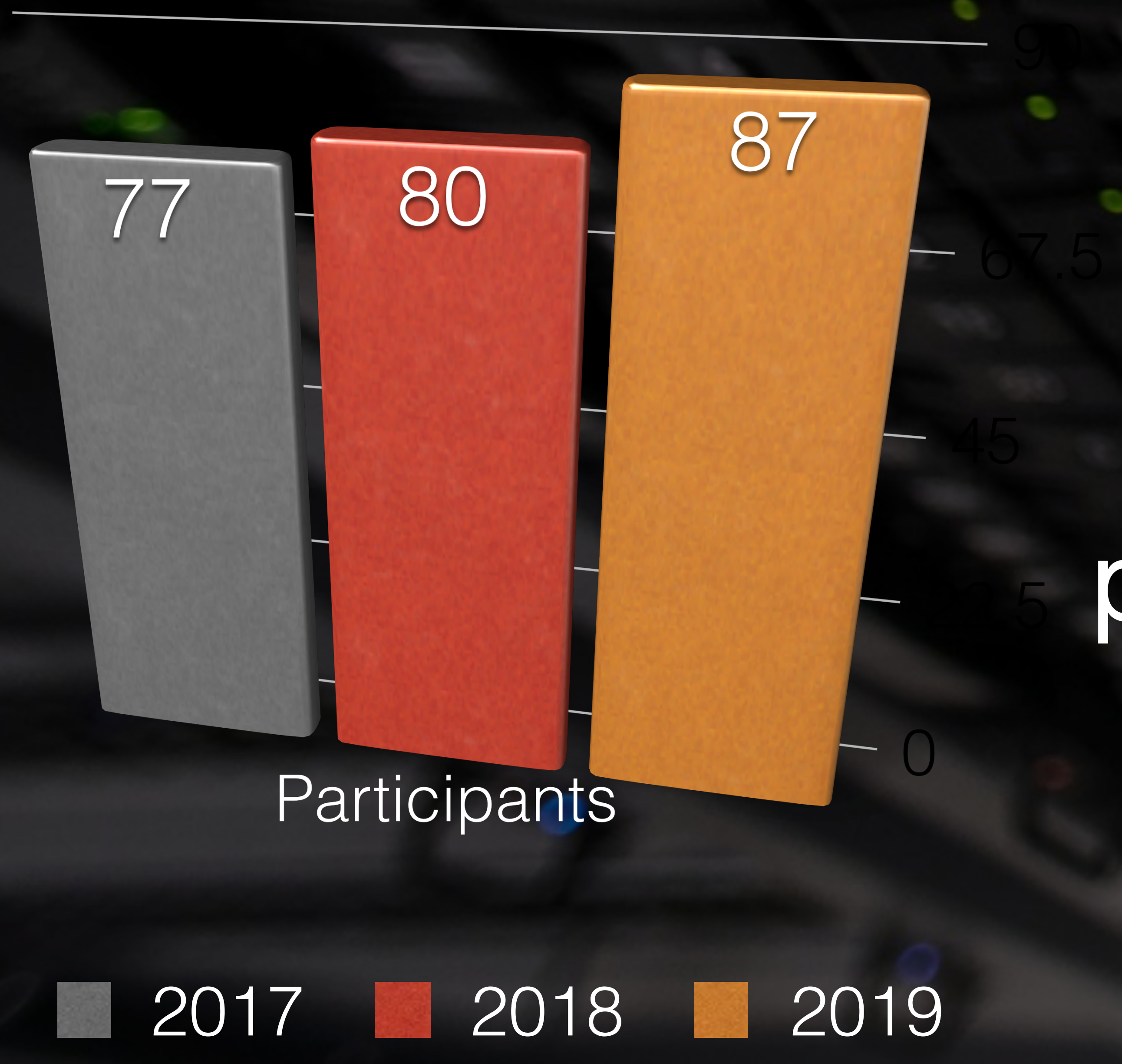
- pre-defined conversion policies for files from/to EC layouts
- light-weight object storage for sequential access & archiving use-cases - client-driven

Invitation

4th edition

EOS workshop 2020

3rd to 5th of february @ CERN



platform for exchange between developers, users, sites and people interested in storage technology

disk - tape - cloud - sync & share - devops

EOS CTA

https://eos.cern.ch

contributions in this conference … Disk$^{\text{Tape}}$ Storage

Code health in EOS: Improving test infrastructure and overall service quality

EOS architectural evolution and strategic development directions

Erasure Coding for production in the EOS Open Storage system

Evolution of the filesystem interface of the EOS Open Storage system

Seeking an alternative to tape-based custodial storage

Using the RichACL Standard for Access Control in EOS

CERN Tape Archive: production status, migration from CASTOR and new features

CERN Disk Storage Services: report from last data taking, evolution and future outlook towards Exabyte-scale storage

Migration of user and project spaces with EOS\CERNBox: experience on scaling and large-scale operations

Converging to Kubernetes for on-premise and hybrid clouds for CERNBox, SWAN, and EOS