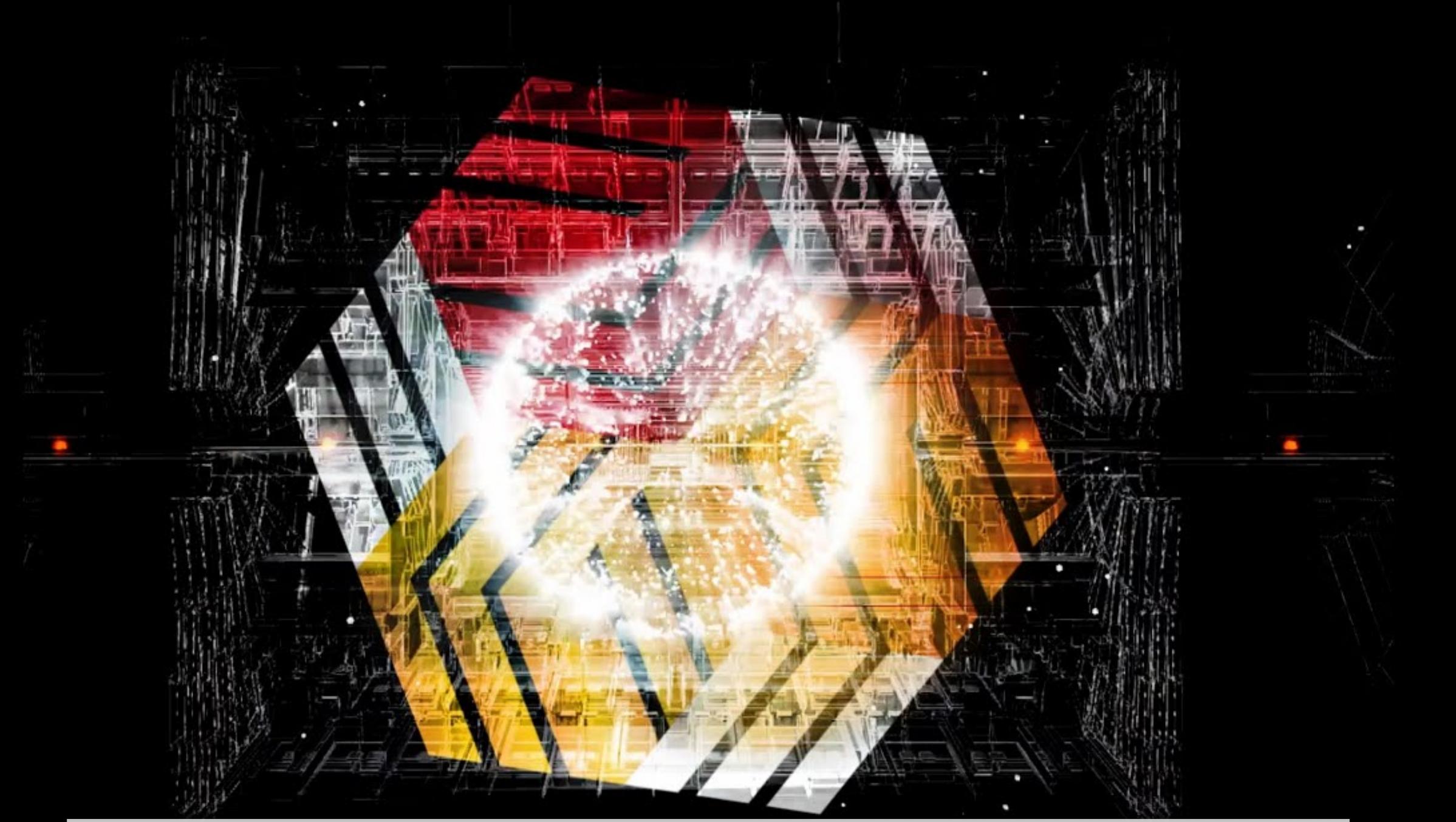


EOS & CTA

EOS Architecture, Deployment, Planning, Core Concepts,
EOS Showcasing, CTA Overview

WELVAVO CTA (dulSowwos S02)



Andreas-Joachim Peters & Elvin Alan Sindrilaru

for the EOS Project

Michael Davis

for the CTA Project



Overview

1 Introduction & Dive into EOS

2 Showcasing EOS Instance Configuration

3 Introduction to CTA

Introduction & Dive into EOS

Andreas-Joachim Peters

What is EOS ?

Open-Source Storage platform designed and developed in CERN IT

**Disk-based distributed filesystem Elastic,
Adaptable and Scalable**

Software solution for data recording, user analysis and data processing

Supports thousands of parallel clients

Multiprotocol support (native xrootd, FUSE, HTTP, WebDAV, CIFS)

**Offers a variety of authentication methods
(KRB5, X509, SharedSecret, tokens, unix)**

eos.web.cern.ch

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 700 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.

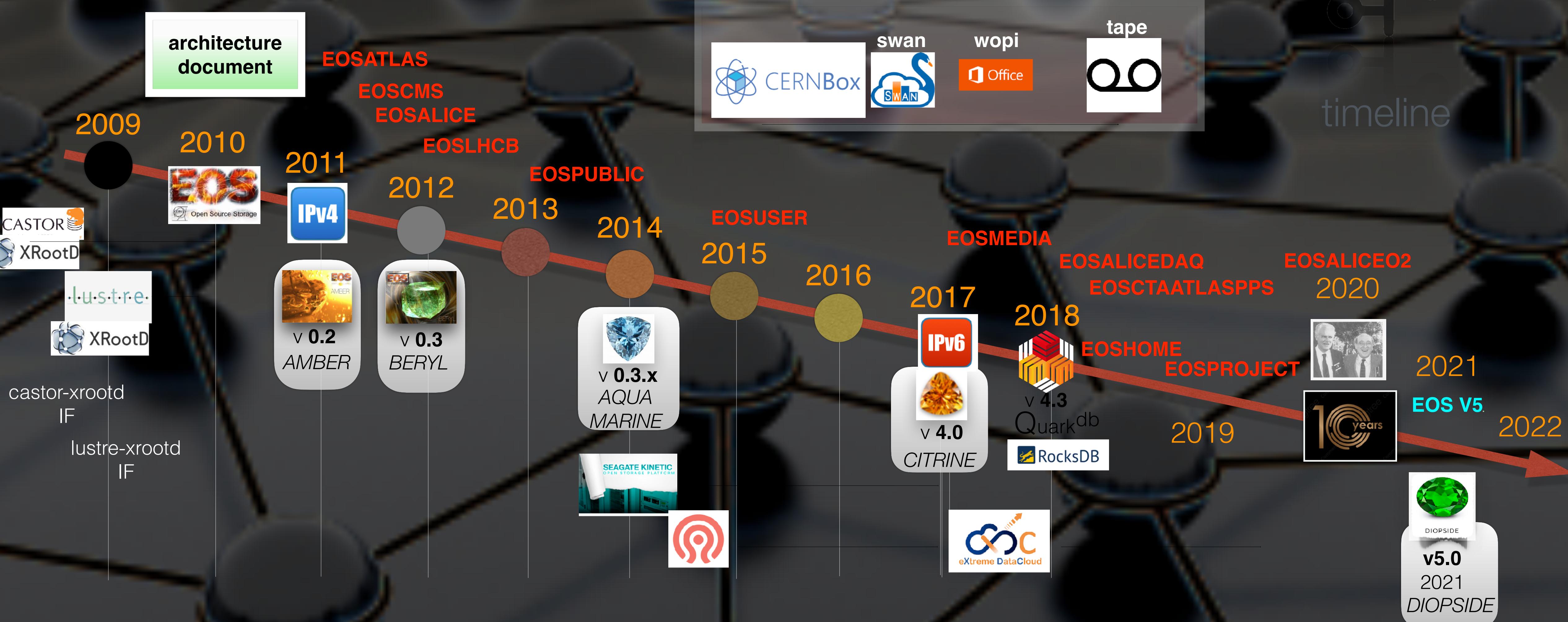


Tape Storage

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

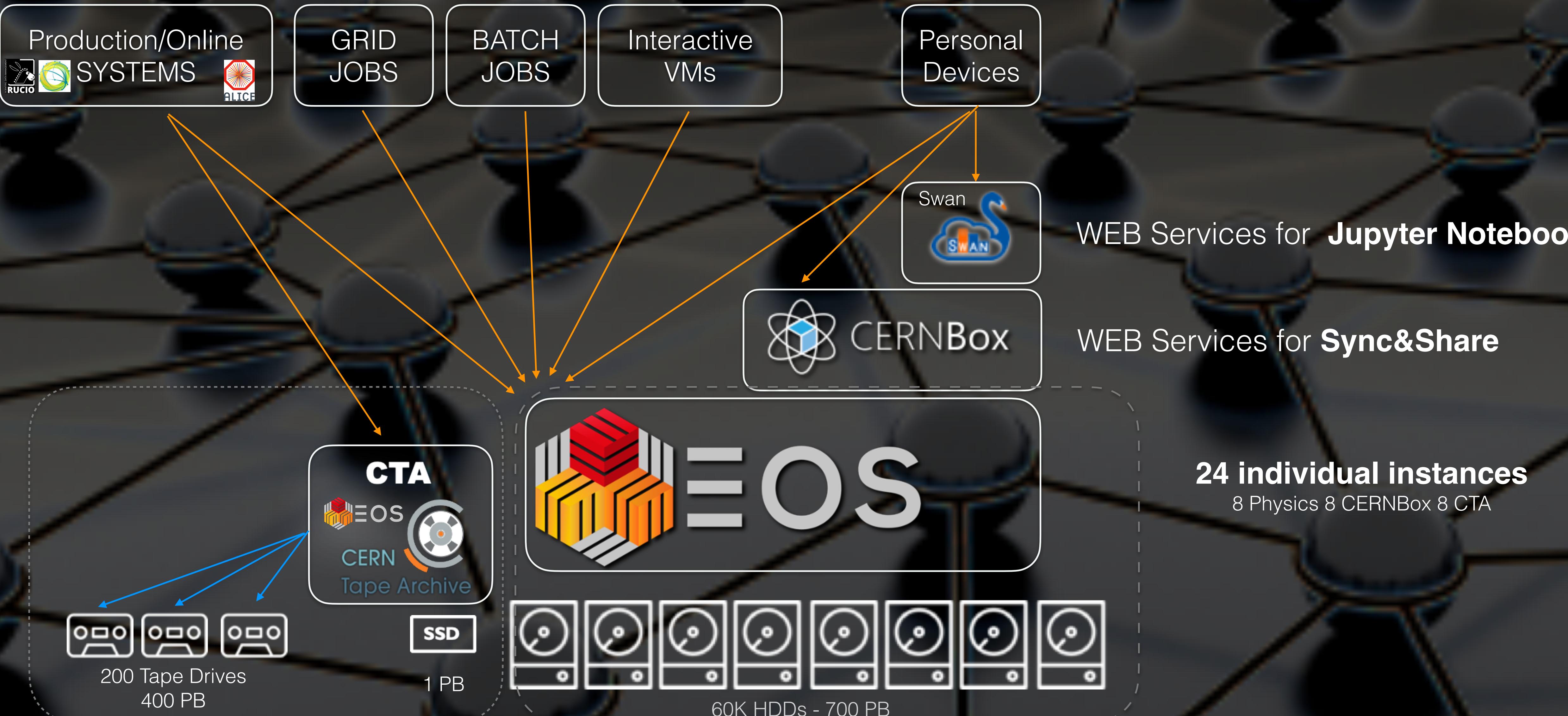


Project Timeline





How is it used at CERN?

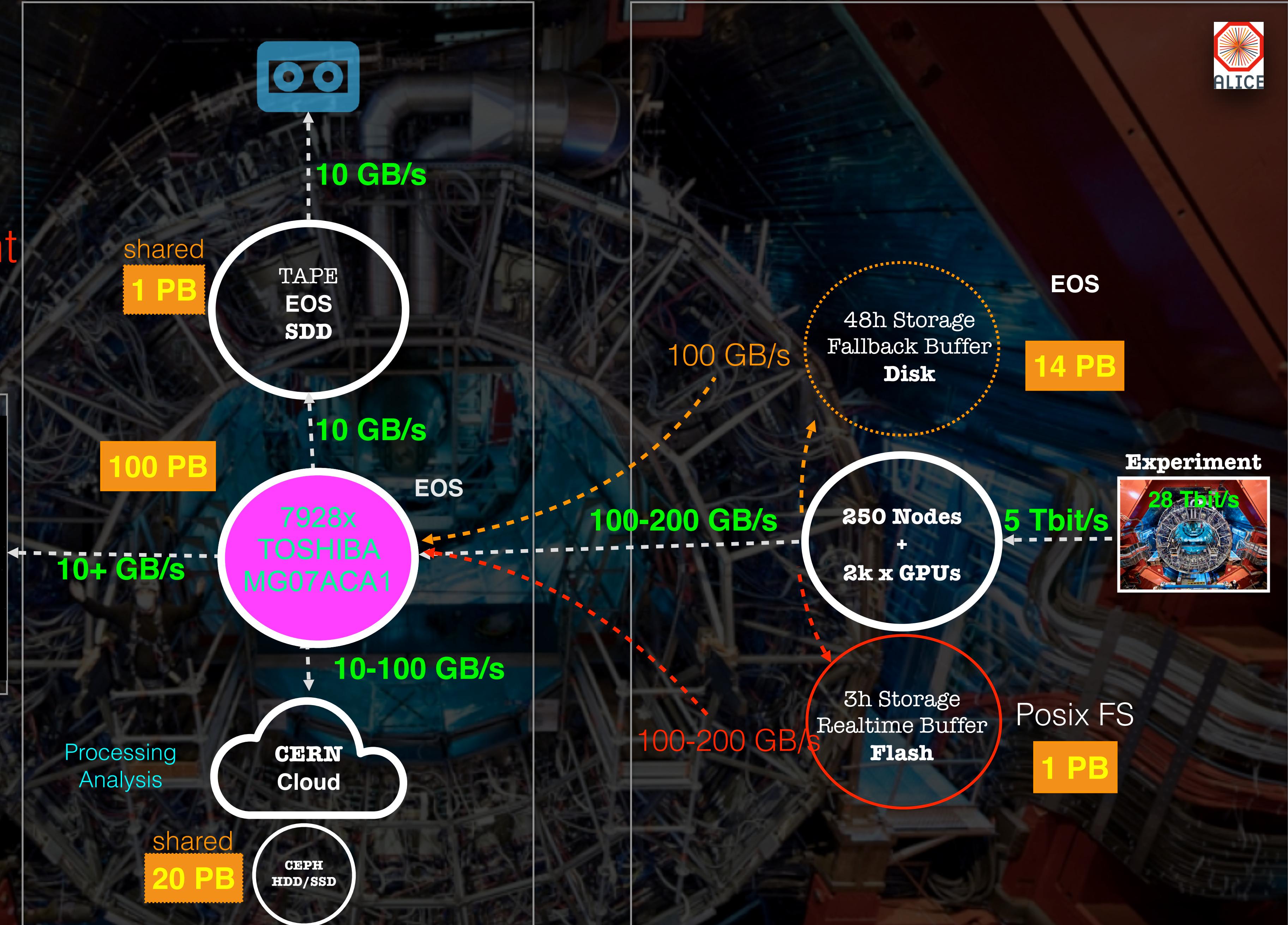
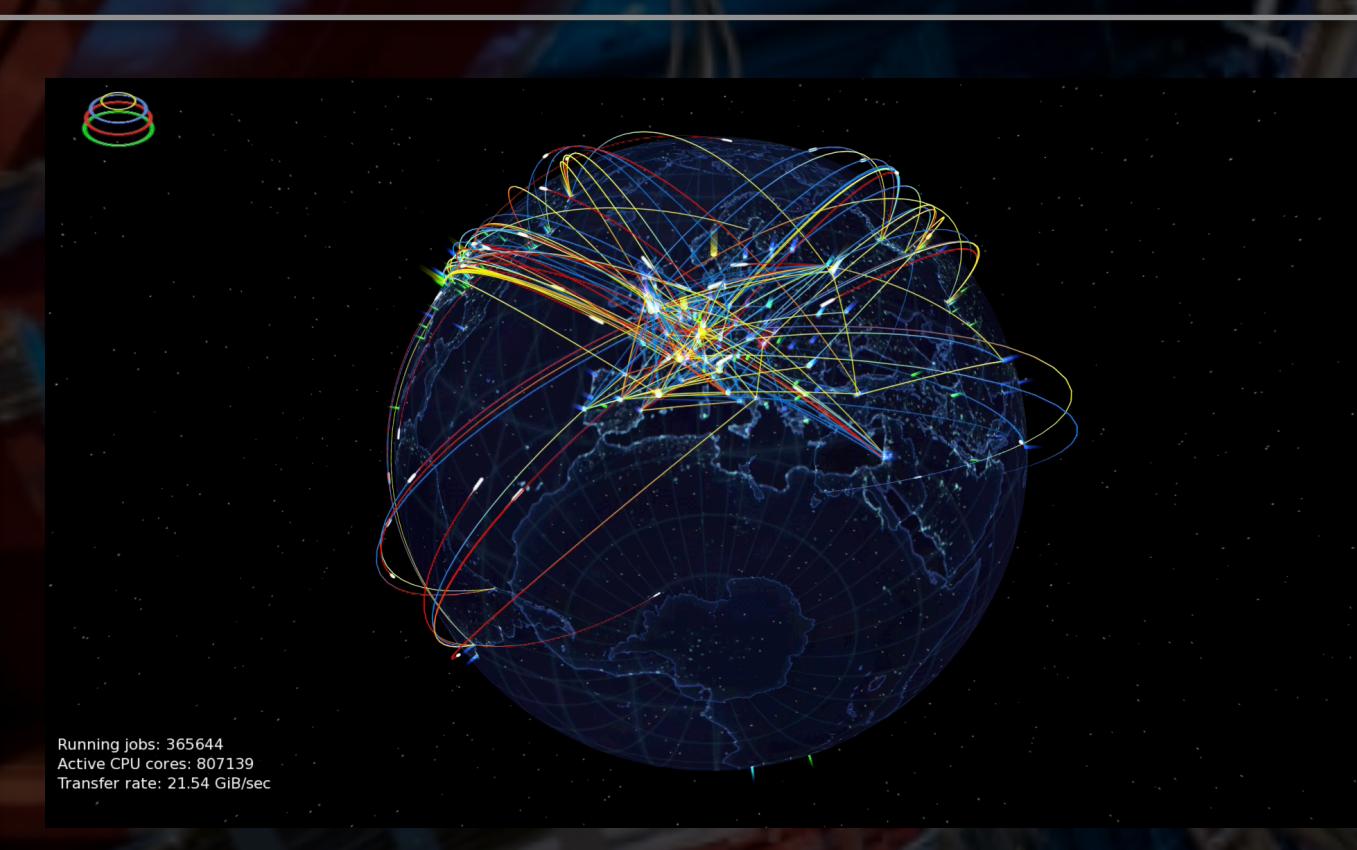


CERN Computer Center

CERN Experimental Site

Dataflow & Storage ALICE LHC Experiment

Worldwide LHC
Computing GRID



CERN Deployment

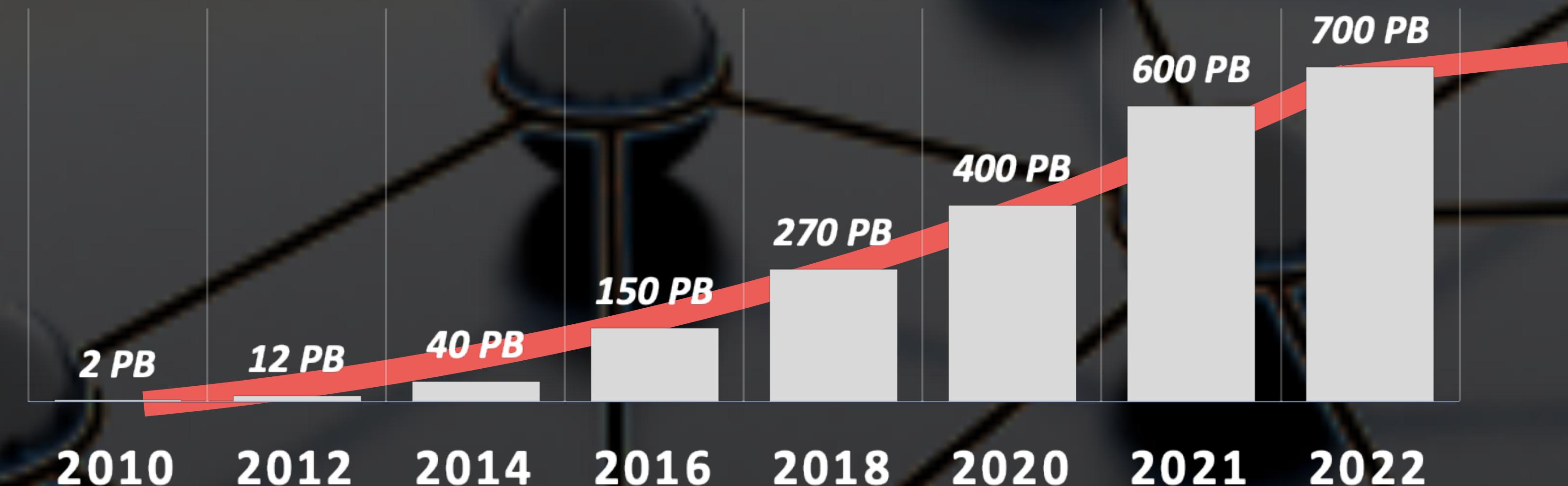
780 PB

60k HDD

1300 server

>7B files

- no significant growth during Run-3
- the number of disk server is shrinking - HDDs get bigger
- smallest EOS instance EOSHOME has no storage at all, largest EOS instance ALICEO2 114 PB usable space



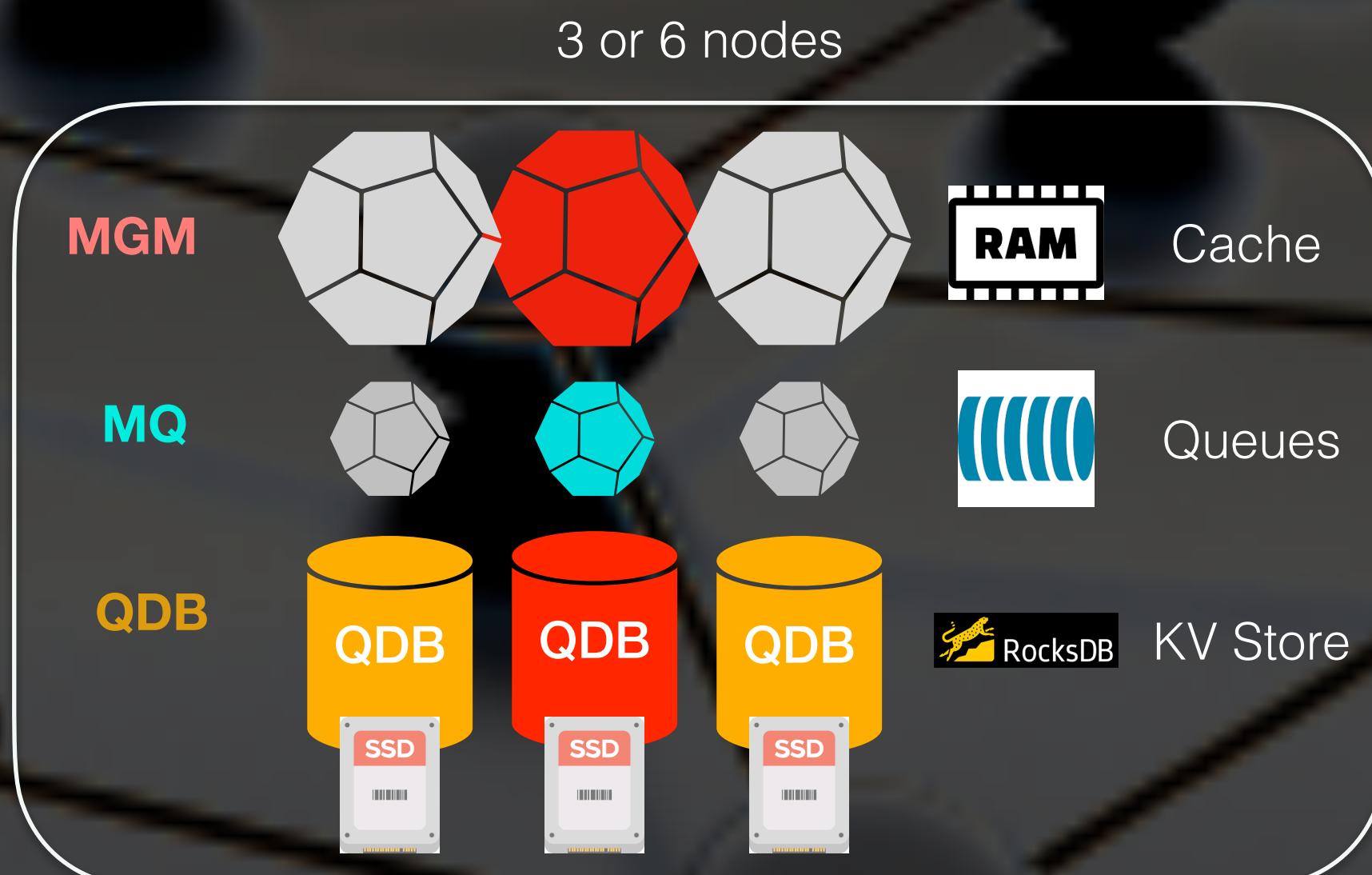


Service Architecture

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

High-available and low latency namespace

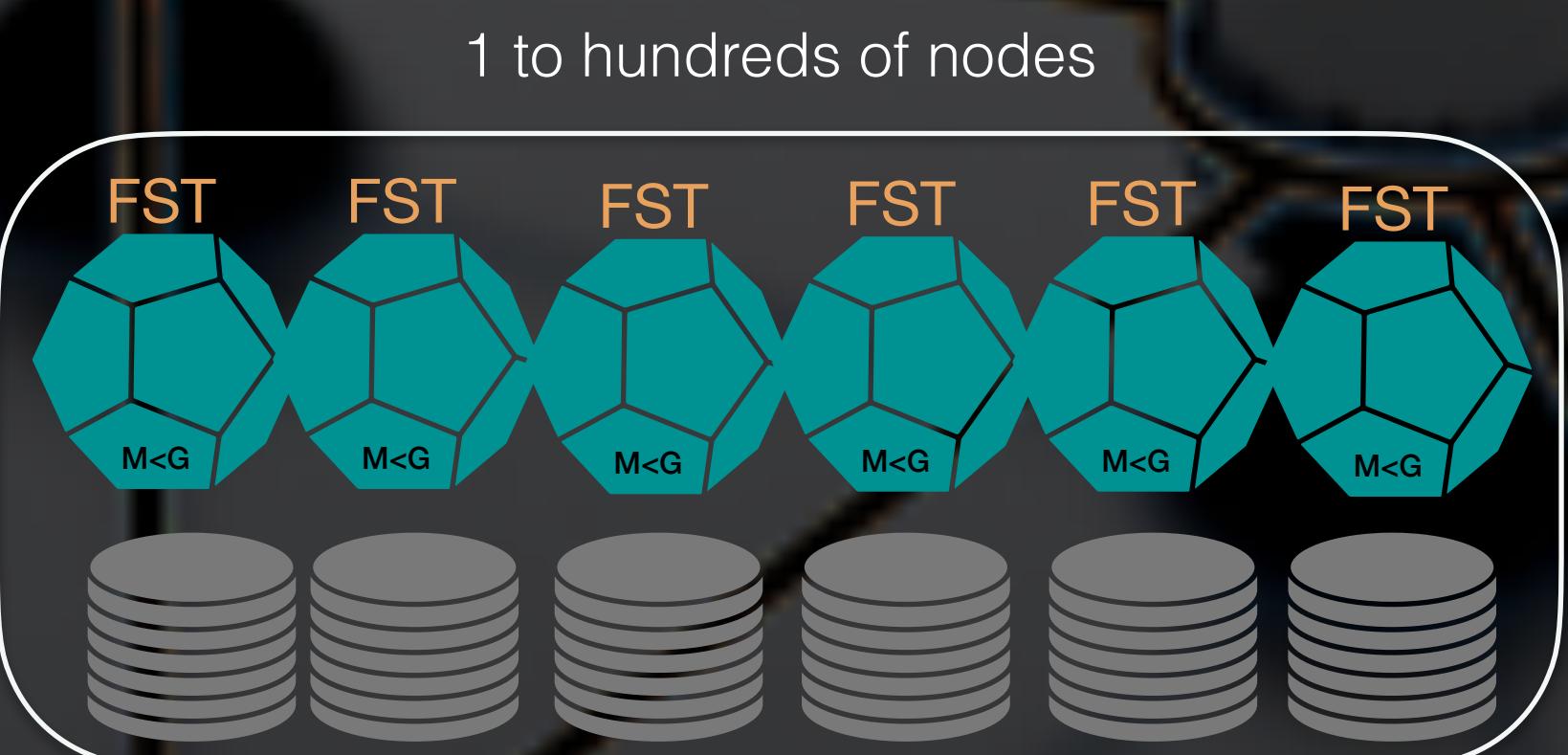
- namespace persisted on a high available key-value store
- working entries cached in-memory



Highly available and reliable file storage,

based on (cheap) JBODs:

- File replication across independent nodes and disks
- Erasure coding to optimize costs and data durability





Full Architecture

framework

XRootD

components

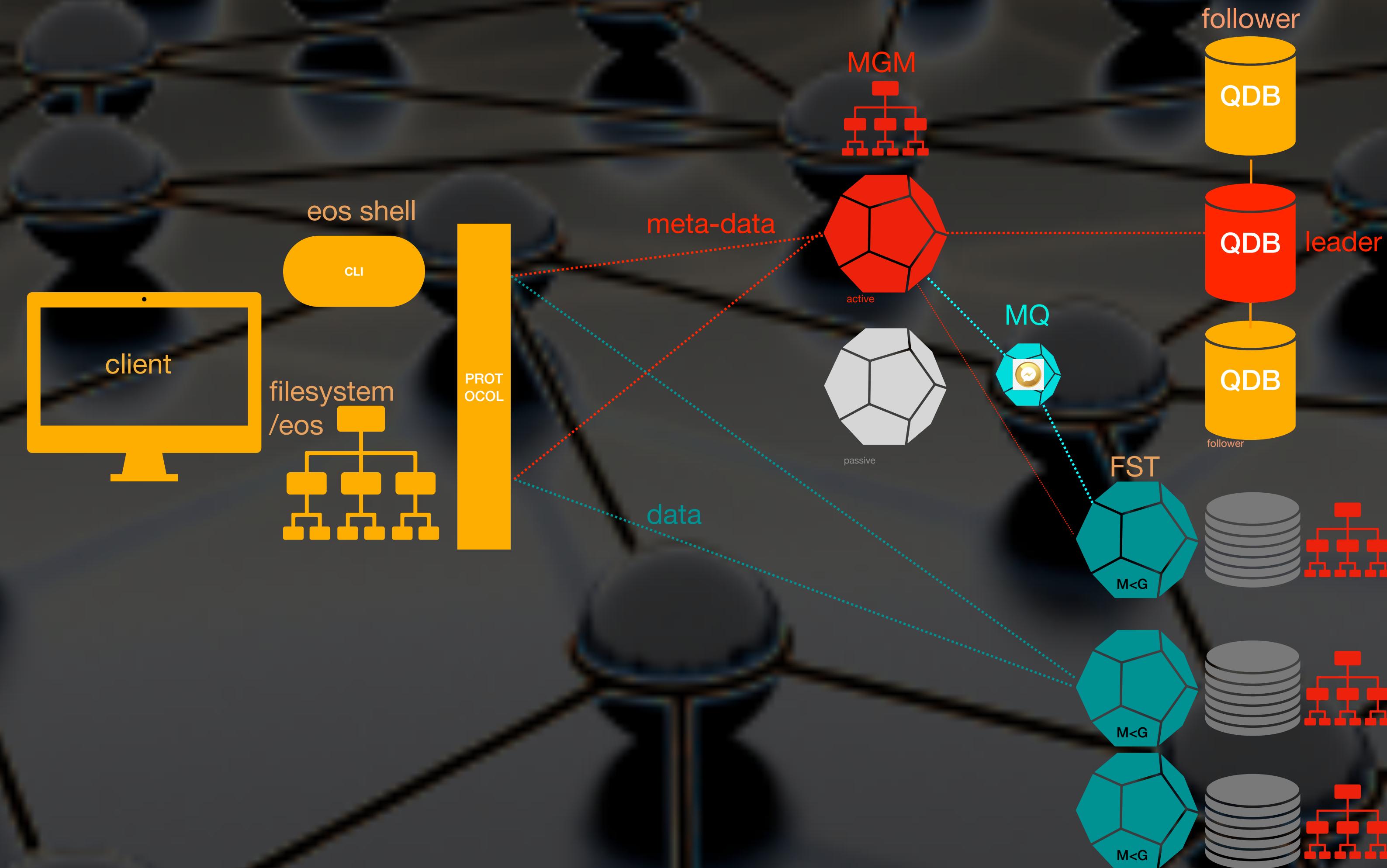
CLIENTs

MGM

MQ

FST

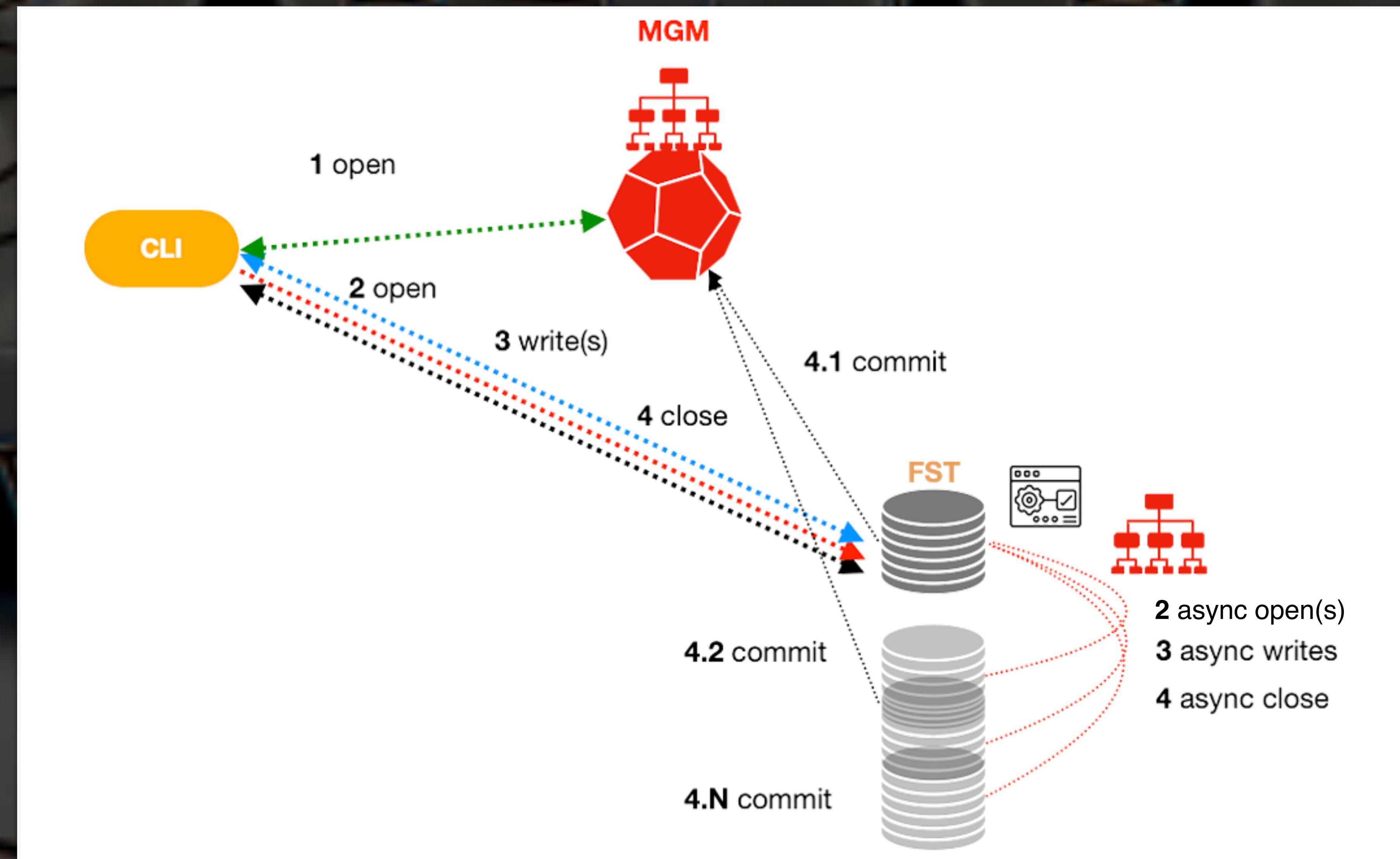
QuarkDB



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



EOS File Transactions





How users see EOS?

```
#  
# EOS SHELL  
#
```

```
$ eos ls -la /eos/  
$ eos cp /eos/myfile /tmp/
```

```
#  
# XROOTD PROTOCOL  
#
```

```
$ xrdcp root://eosinstance//eos/myfile /tmp/
```

```
#  
# HTTP PROTOCOL  
#
```

```
$ curl https://eosinstance/eos/myfile
```

```
#  
# Mounted Filesystem  
#
```

```
$ ls -la /eos/  
$ cp /eos/myfile /tmp/
```

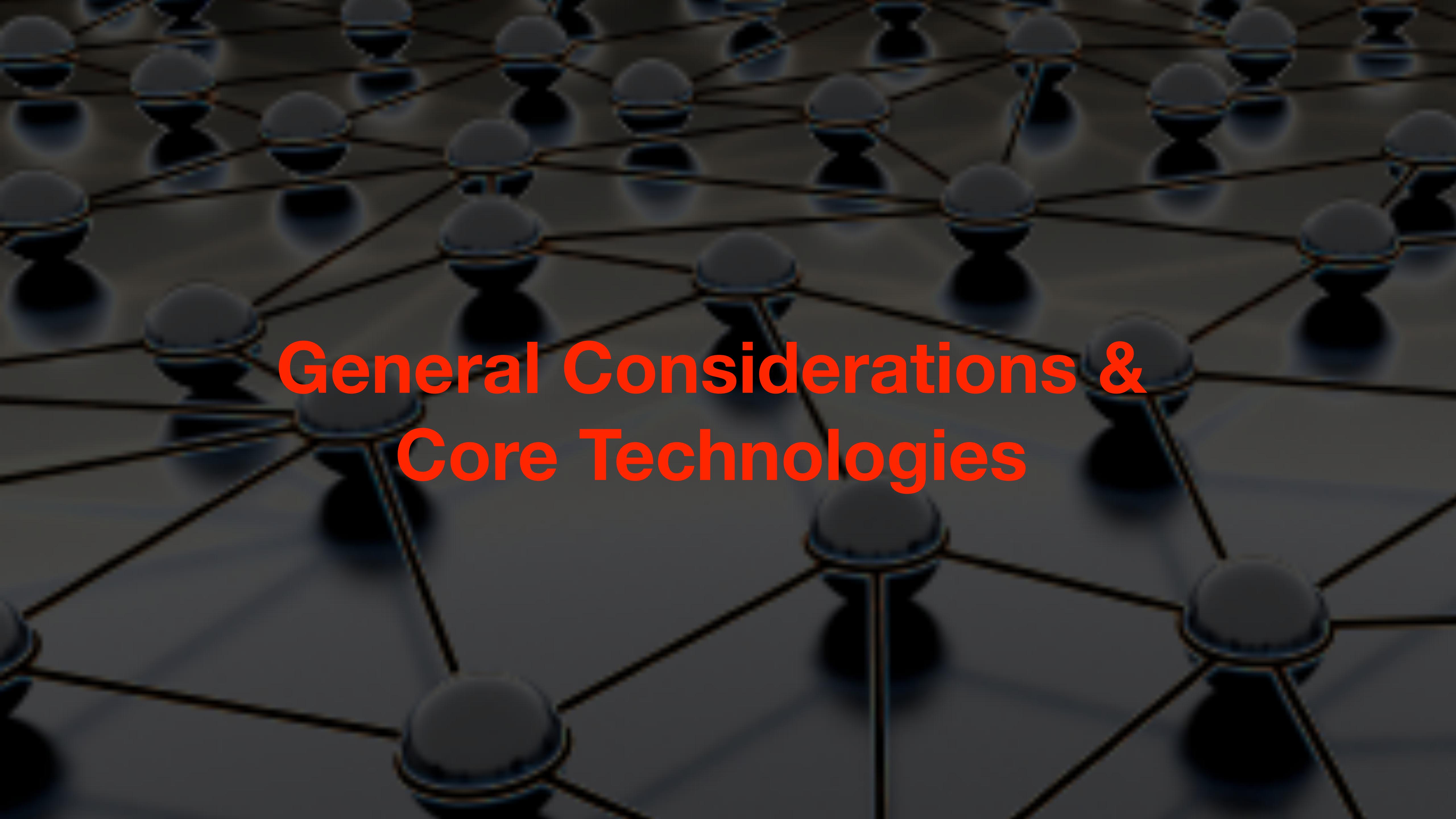
via web services ...

The screenshot shows the CERNBox UI interface. At the top, there's a navigation bar with icons for home, search, and user account. Below it, the path 'CERNBox > eos > user > a > apeters' is displayed. The main area is titled 'CERNBox UI' and shows a list of files and folders under 'All files'. The list includes:

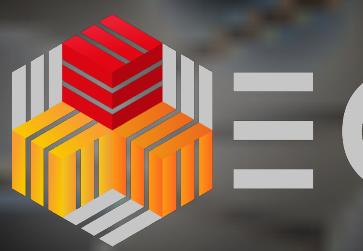
	Name	Shares	Size	Modified	Actions
<input type="checkbox"/>	acl	(edit)	3 kB	11 months ago	⋮
<input type="checkbox"/>	ajp-test		11.9 MB	2 years ago	⋮
<input type="checkbox"/>	alice		2 kB	1 year ago	⋮
<input type="checkbox"/>	anaconda		405.4 MB	8 months ago	⋮

or as very large filesystem ...

atlas	81P	69P	13P	86%	/eos/atlas
experiment	78P	65P	13P	84%	/eos/experiment
cms	72P	46P	27P	64%	/eos/cms
web	4.1P	2.9P	1.3P	70%	/eos/web
M&P	4.1P	2.9P	1.3P	70%	\eos\m&p



General Considerations & Core Technologies



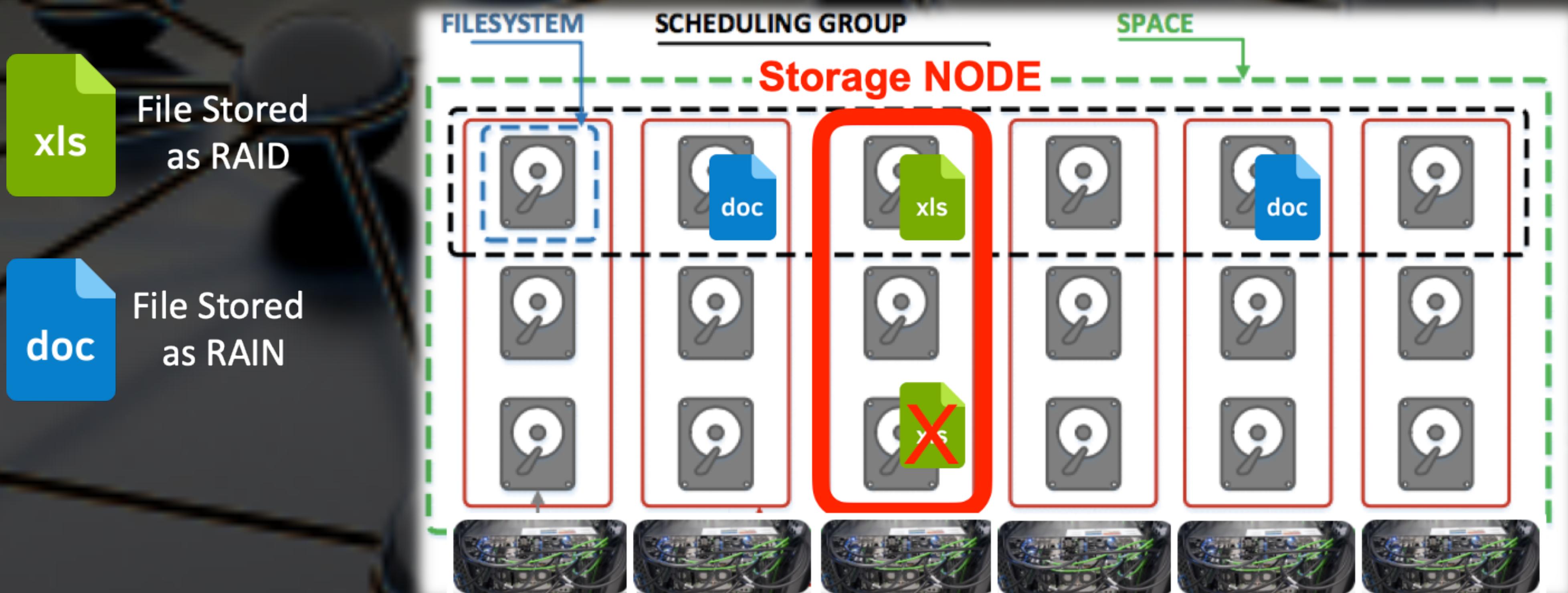
Storage Hardware Evolution

- Profiting from economy of scale
 - minimise price per TB
- Latest generation of storage servers
 - 8 trays (24x disks each) per system unit
 - 40Gbit eth
 - ~2300 TB (12TB drives)
 - 4 trays (24x disks each) per system unit
 - 100Gbit eth
 - ~1536 TB (16TB drives)
 - ~1728 TB (18TB drives)
- High Density JBOD
 - 2 trays (60x disks) per system unit
 - ~840 TB (14TB drives)



File Replication

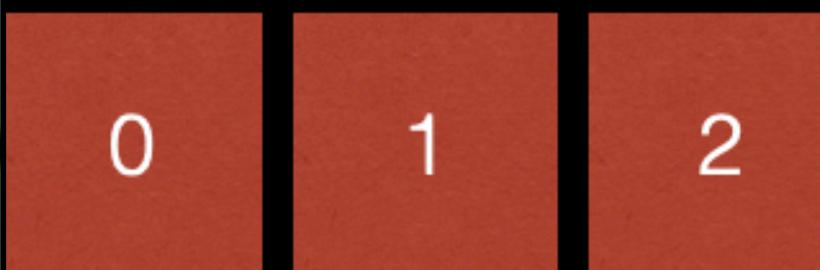
- File availability and Redundancy
 - RAID vs. RAIN
 - Files are replicated n-times on different disk on different machines
 - Protect against disk failure and storage node failure





Erasure Coding

Write



IO gateway



1



2



0.0



0.1

EOS EC IO Path

Read XrdCI



IO gateway



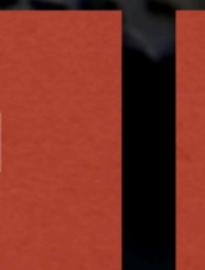
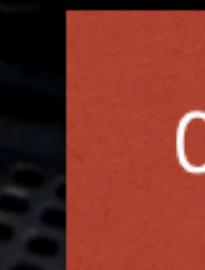
1



2

Read eoscp

direct IO



raw mirrored EC (0,2)



File Layouts

- layouts describe IO path and redundancy for a file - EOS supports
 - **plain** (1 copy), **replica** (n copies)
 - erasure coding: **raid6** (2 parity), **archive** (3 parity), **qrain** (4 parity)

Layout: **replica** Stripes: 1 Blocksize: 4k LayoutId: 00100012 Redundancy: **d1::t0**
#Rep: 1

no.	fs-id	host	schedgroup	path	boot	configstatus	drain	active	geotag
0	58	st-120hd-100gb009.cern.ch	default.57	/data57	booted	rw	nodrain	online	0513::EC

Layout: **raid6** Stripes: 6 Blocksize: 1M LayoutId: 20640542 Redundancy: **d3::t0**
#Rep: 6

no.	fs-id	host	schedgroup	path	boot	configstatus	drain	active	geotag
0	116	st-120hd-100gb010.cern.ch	default.56	/data56	booted	rw	nodrain	online	0513::EC
1	295	st-120hd-100gb013.cern.ch	default.56	/data56	booted	rw	nodrain	online	0513::EC
2	415	st-120hd-100gb015.cern.ch	default.56	/data56	booted	rw	nodrain	online	0513::EC
3	175	st-120hd-100gb011.cern.ch	default.56	/data56	booted	rw	nodrain	online	0513::EC
4	355	st-120hd-100gb014.cern.ch	default.56	/data56	booted	rw	nodrain	online	0513::EC
5	475	st-120hd-100gb016.cern.ch	default.56	/data56	booted	rw	nodrain	online	0513::EC

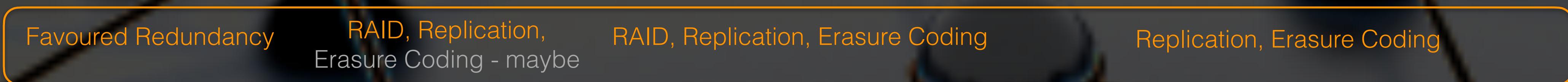


EOS Deployments

Production
Setups



many



few





The ideal Storage Software ...

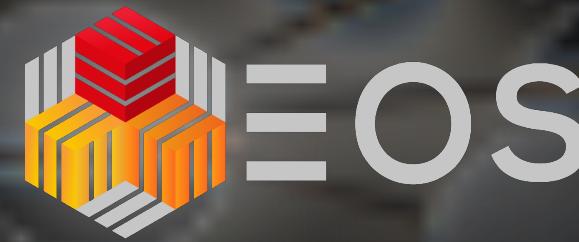
Install
Run
Use

yum install eos
eos start
Everything works!



EOS is simple, if you know what to do ... otherwise it can be really complex!

Does not need to be for you ! A green circular icon containing a white thumbs-up symbol.



Plan the Deployment

EOS consists of **four types** of daemons running with the *right* configuration



QDB: Meta-Data Persistency

MGM: Meta-Data Access

MQ: Messaging

FST: Data Persistency & Access

You **have to plan** the layout of your hardware and the deployment and configuration of EOS Services!



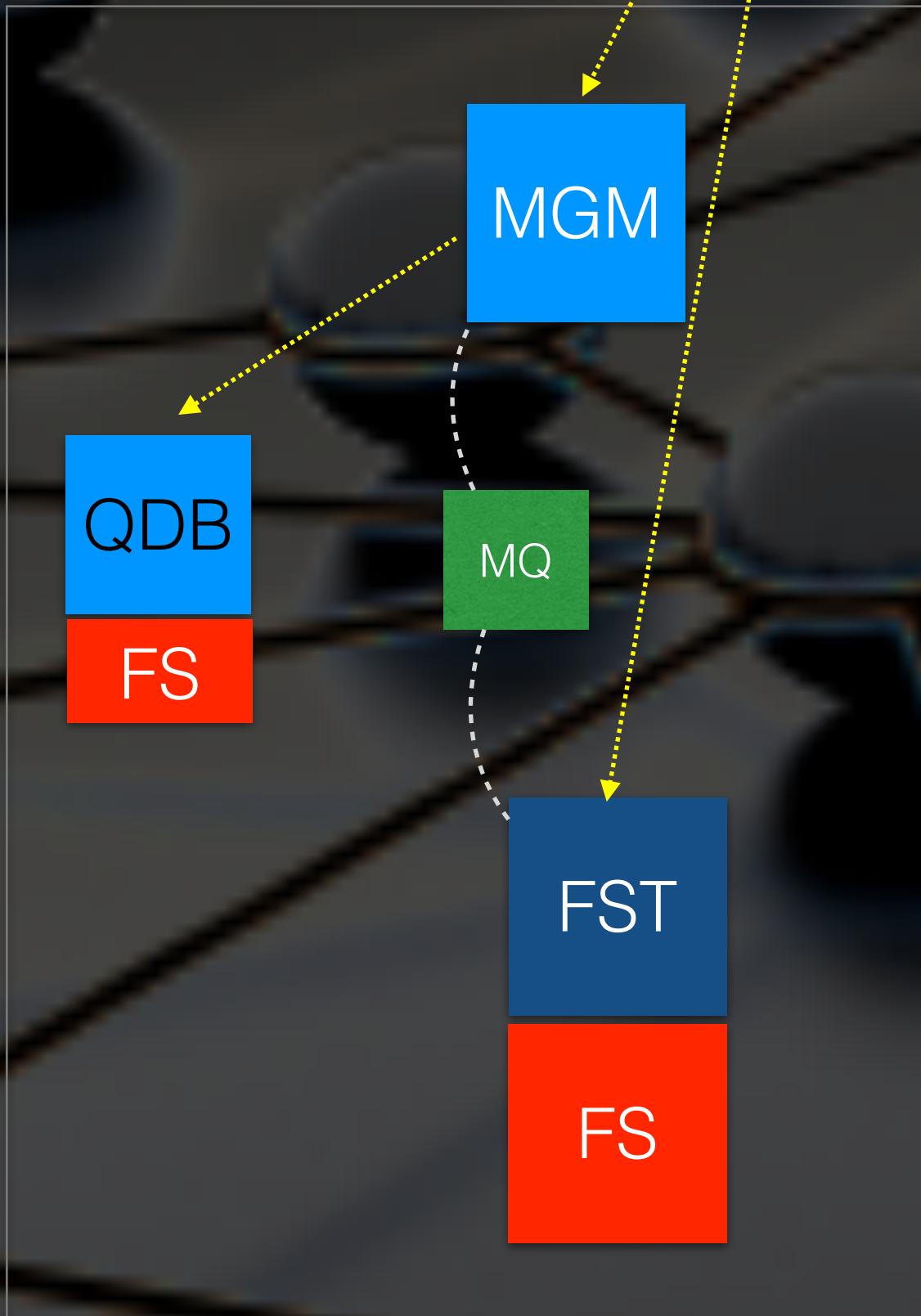
Deployment Scenarios Hardware Requirements



EOS

Client

node



All daemons in one
physical box:
QDB,MGM,MQ,FST

Deployment

Single Physical Box

Hardware Requirements:

QDB: **SSD/NVMe**
0.1-0.2 GB/Million Entries

MGM:
4 core - min. 8 GB

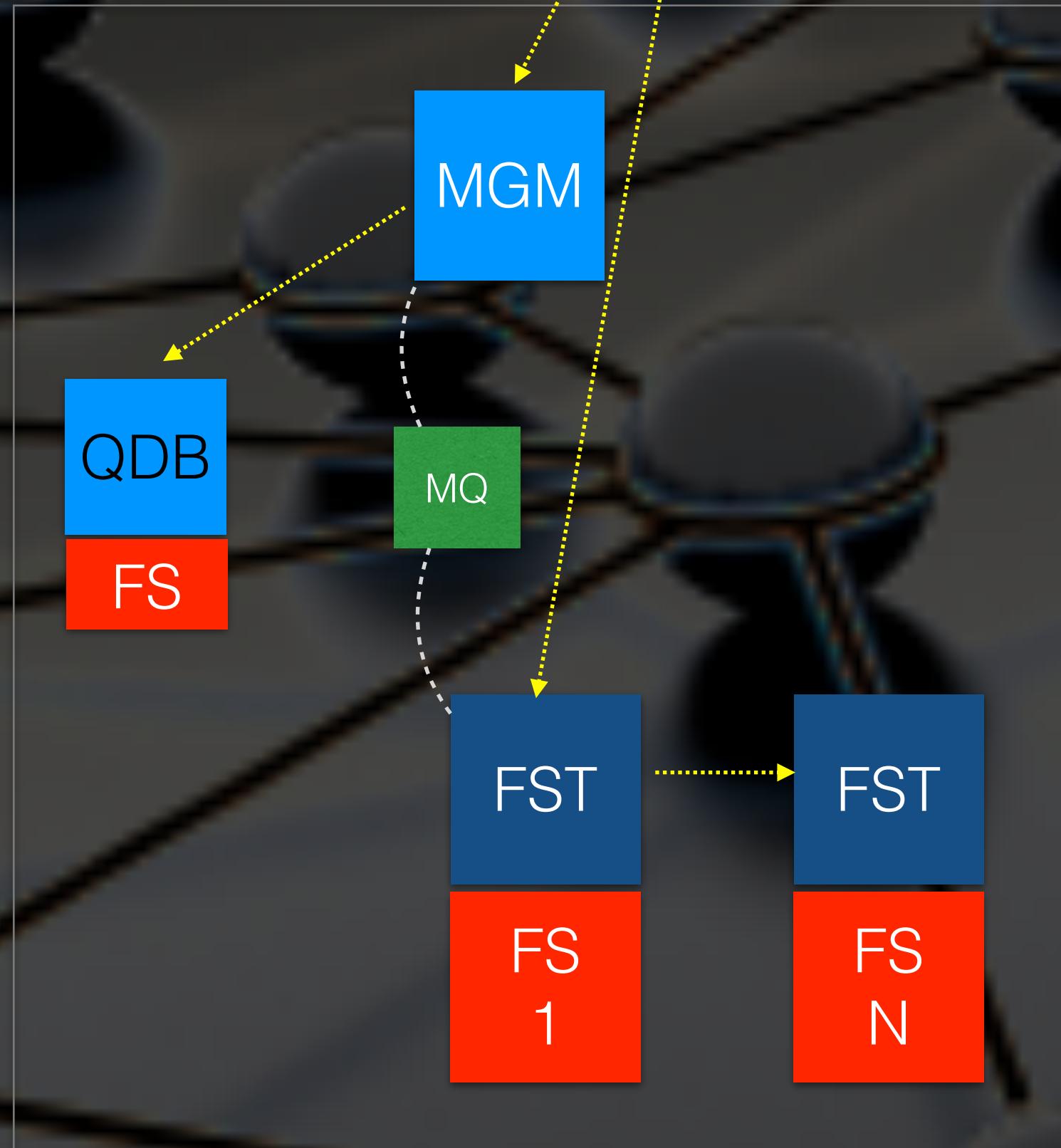
FST:
4 core - min. 8 GB
1 GB RAM / HDD
HDD FS: XFS+XAttr



Deployment Prototypes

Single Physical Box - Multiple FSTs

node



All daemons in one
physical box:
QDB,MGM,MQ,FST1-N

Hardware Requirements:

QDB: SSD/NVMe
0.1-0.2 GB/Million Entries

MGM:
4 core - min. 8 GB

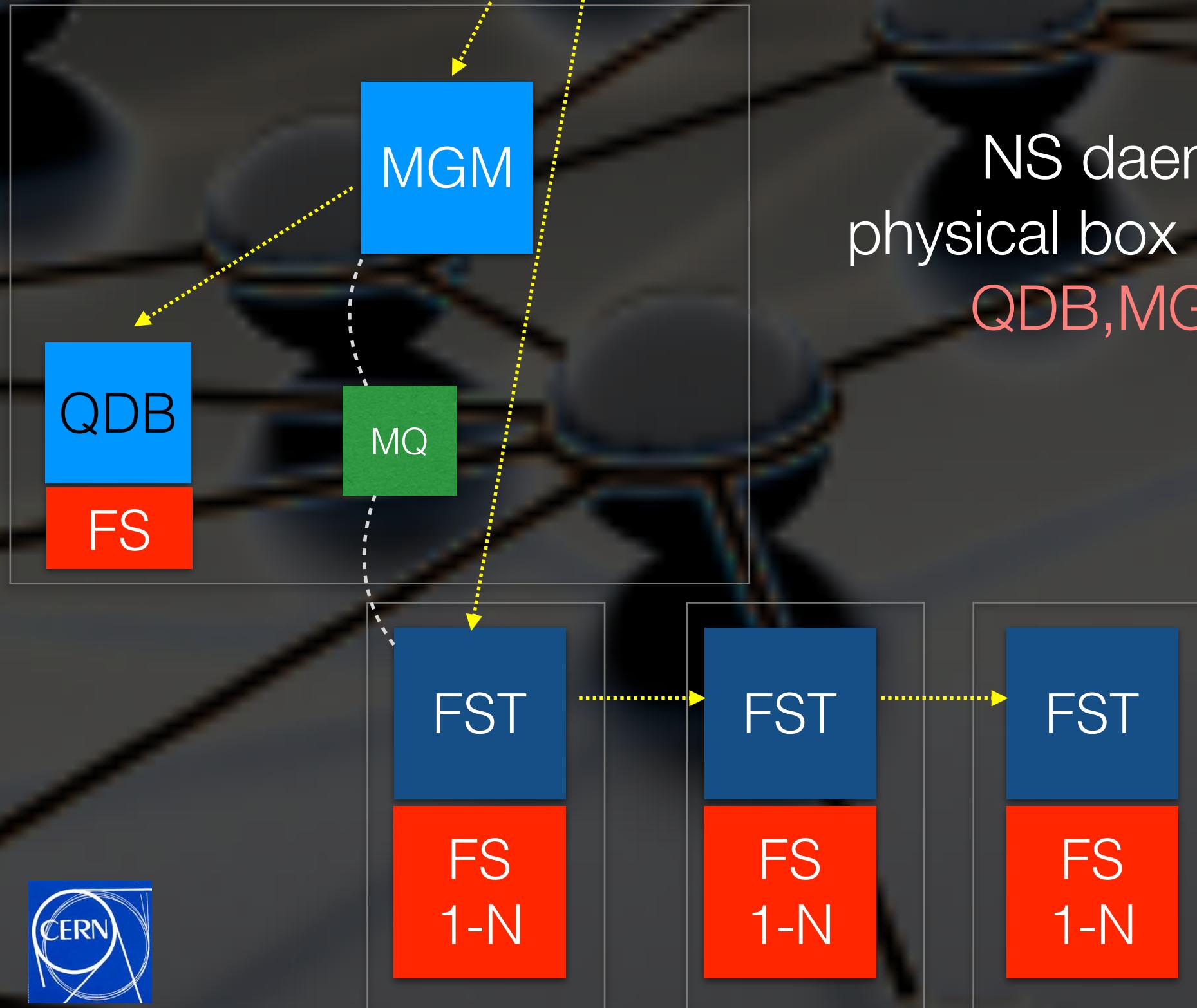
FST:
4 core - min. 8 GB
1 GB RAM / HDD
HDD FS: XFS+XAttr



Deployment Prototypes

Namespace + K Storage Nodes with N Filesystems

node



NS daemons in one
physical box + N FST Nodes:
QDB,MGM,MQ,FST

Hardware Requirements:

QDB: **SSD/NVMe**
0.1-0.2 GB/Million Entries

MGM:
4 core - min. 8 GB

FST:
4 core - min. 8 GB
1 GB RAM / HDD
HDD FS: XFS+XAttr

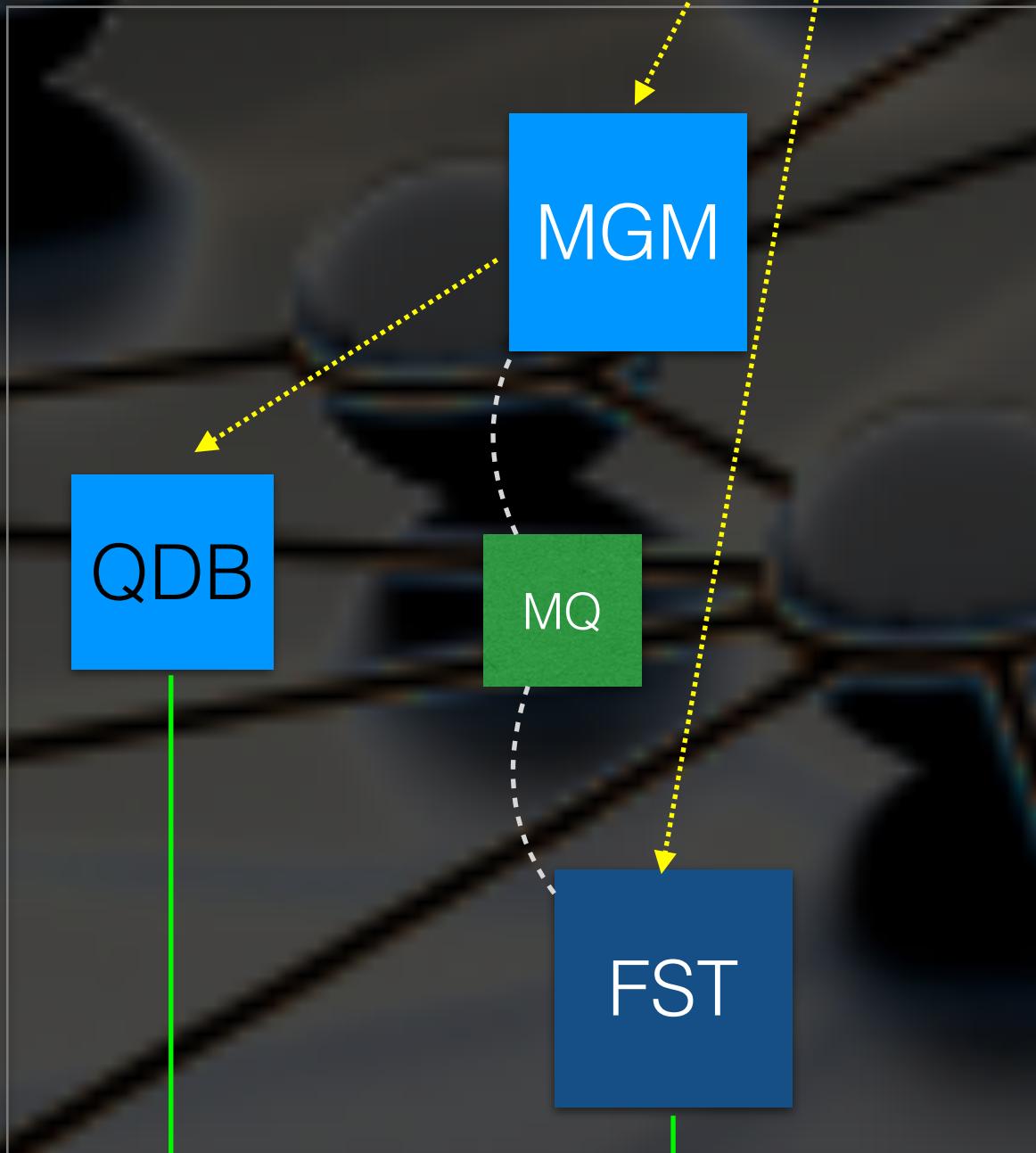




Deployment Prototypes

Single Virtual Box

node



All daemons in one
virtual box:
QDB,MGM,MQ,FST

Hardware Requirements:

QDB: **HIGH IOPS Virtual Disk**
0.1-0.2 GB/Million Entries

MGM:
4 core - min. 8 GB

FST:
4 core - min. 8 GB
1 GB RAM / HDD

HDD FS: remote FS with XAttr Support

High IOPS

High BW

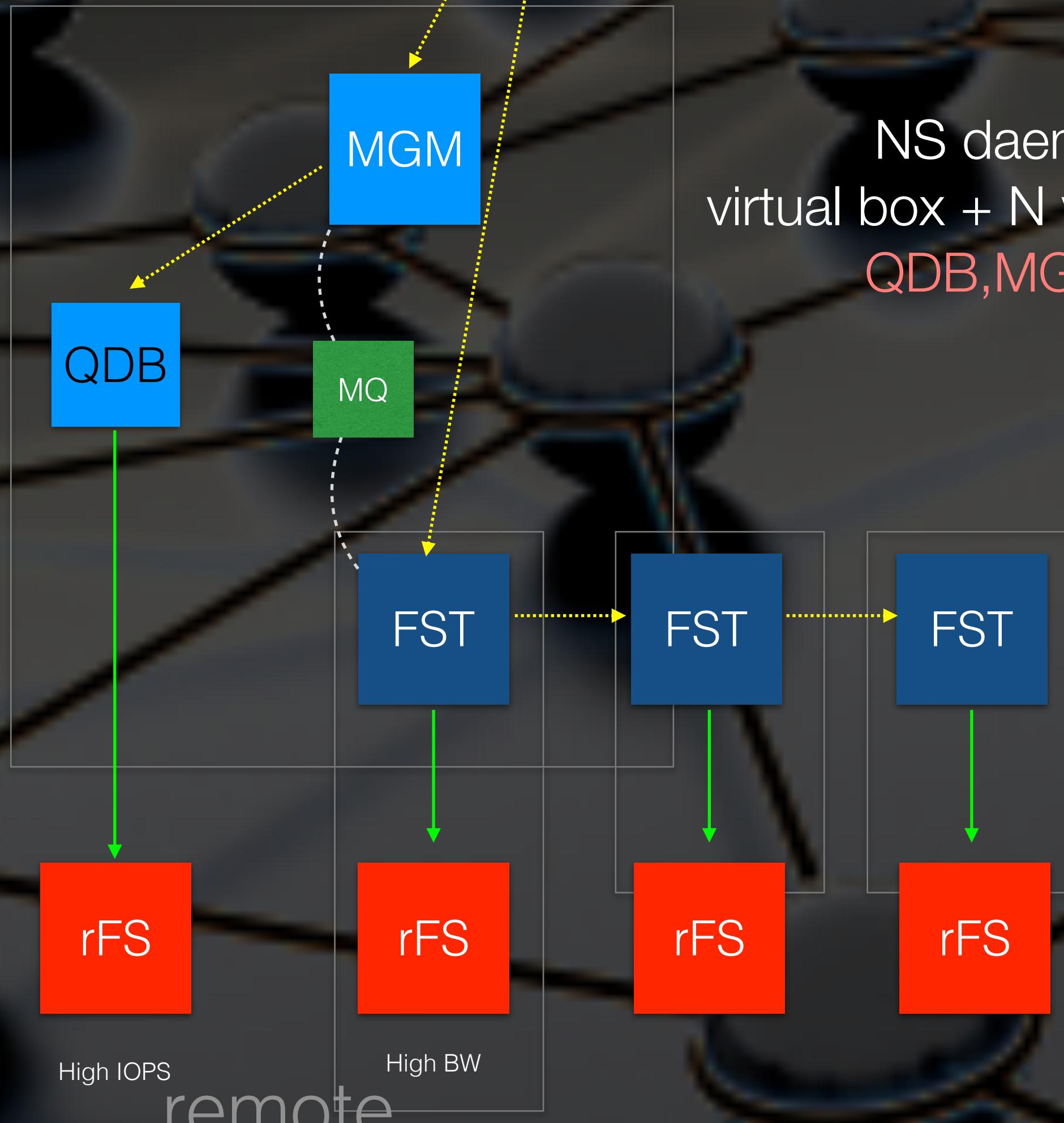
remote



Deployment Prototypes

Virtual Namespace + K Storage Nodes

node



Hardware Requirements:

QDB: **HIGH IOPS Virtual Disk**
0.1-0.2 GB/Million Entries

MGM:
4 core - min. 8 GB

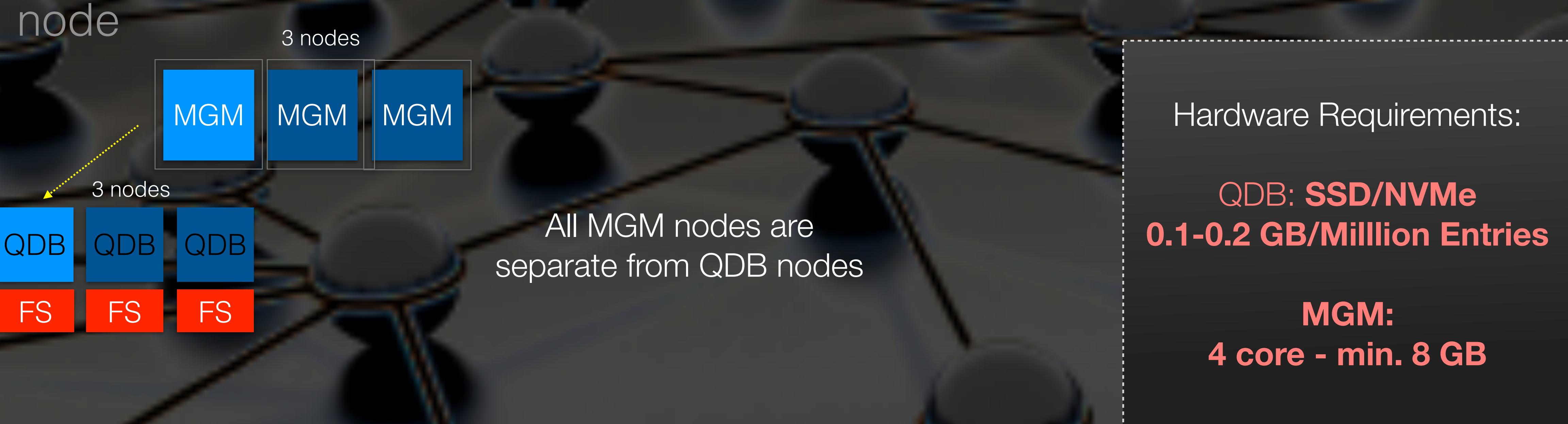
FST:
4 core - min. 8 GB
1 GB RAM / HDD

HDD FS: remote FS with XAttr Support



Deployment Prototypes

HA Namespace Setup - 6 Nodes



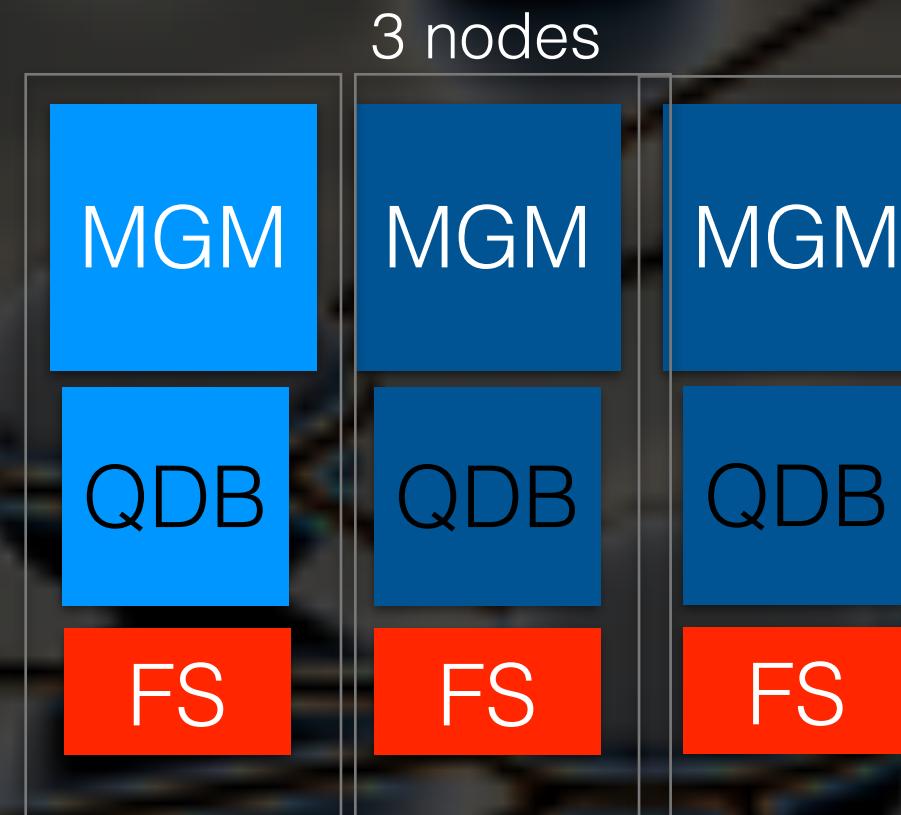
All MGM nodes are
separate from QDB nodes



Deployment Prototypes

HA Namespace Setup - 3 Nodes

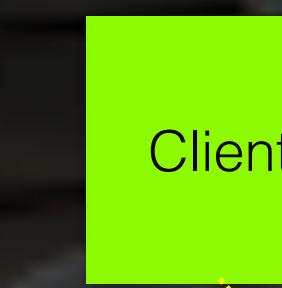
node



Hardware Requirements:

QDB: **SSD/NVMe**
0.1-0..2 GB/Million Entries

MGM+QDB:
4 core - min. 16 GB

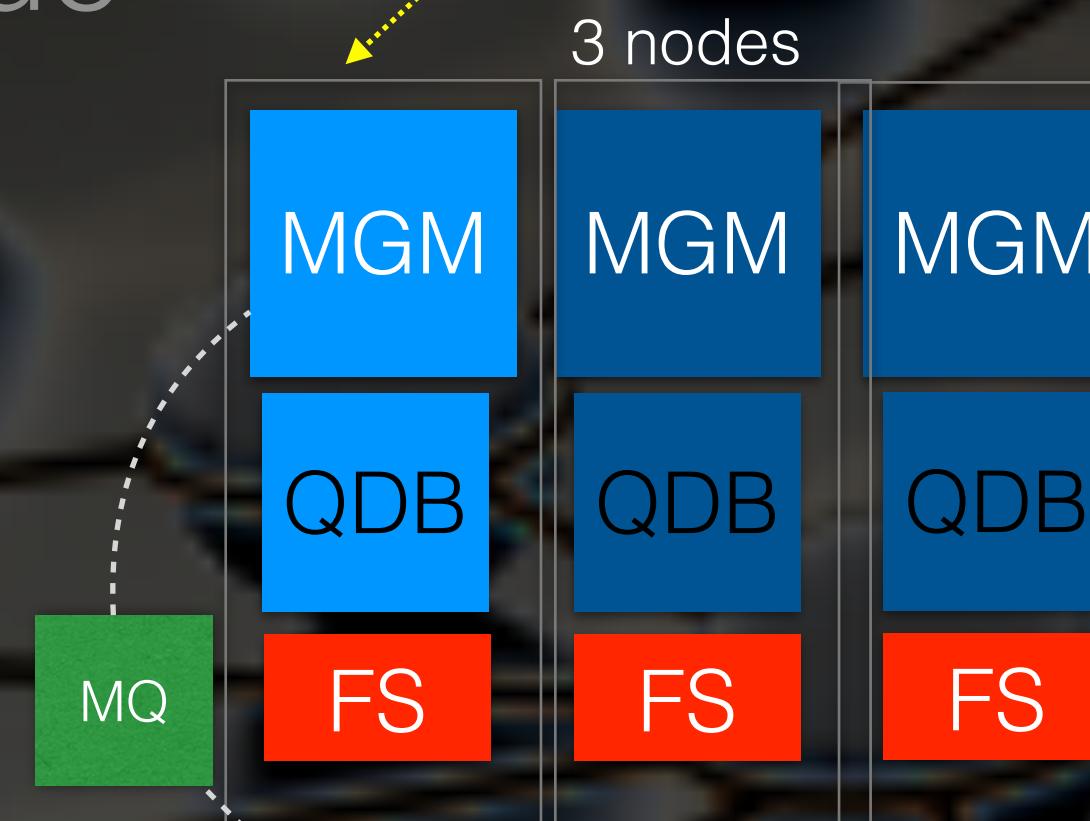


CERN Standard Deployment

no virtualisation/containerization

HA Namespace Setup - 3 Nodes

node



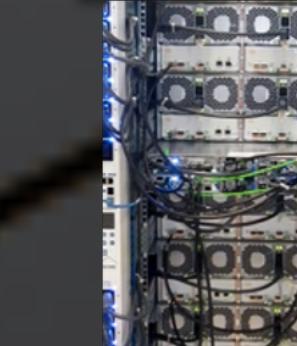
3 nodes

Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz (32 core),
386 GB RAM (2933 MHz DDR4), 2x 1.8 TB /var partition
(INTEL SSDSC2KB01)

3-3.5 GB/s streaming write per node with EC
6 GB/s streaming read per node with direct EC
5-6.5 GB/s streaming write per node with single copy files



96 HDDs
per FST



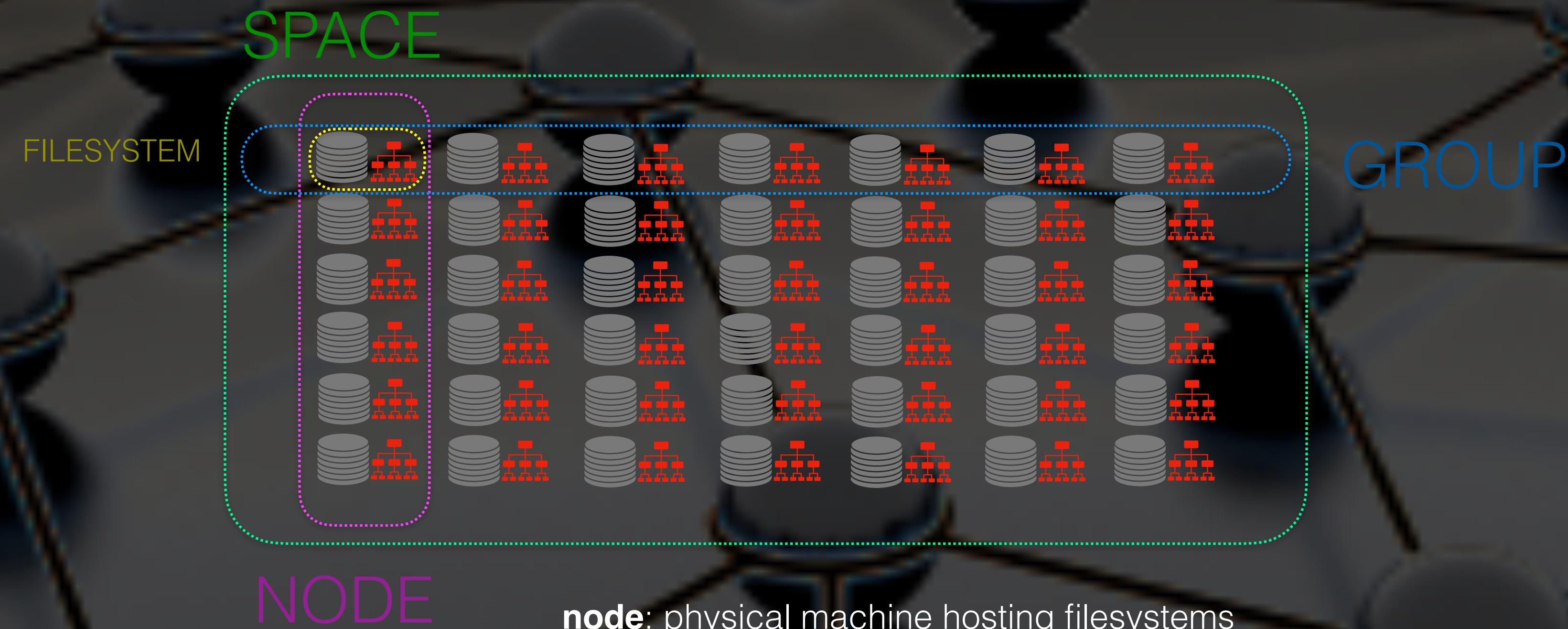
2x AMD EPYC 7302 16-Core Processor, 128 GB RAM (3200 MHz, DDR4),
96x TOSHIBA MG07ACA1 (14TB)

Storage Layout



FileSystem View: your Storage Layout

Concept of: **Space - Group - Node - Filesystem**



node: physical machine hosting filesystems

space: aggregation of groups = aggregation of filesystems

group: vertical aggregation of filesystems used for scheduling

filesystem: individual mounted device

```
[root@eosaliceo2-ns-01 (mgm:master mq:slave) ~]$ eos space ls
```

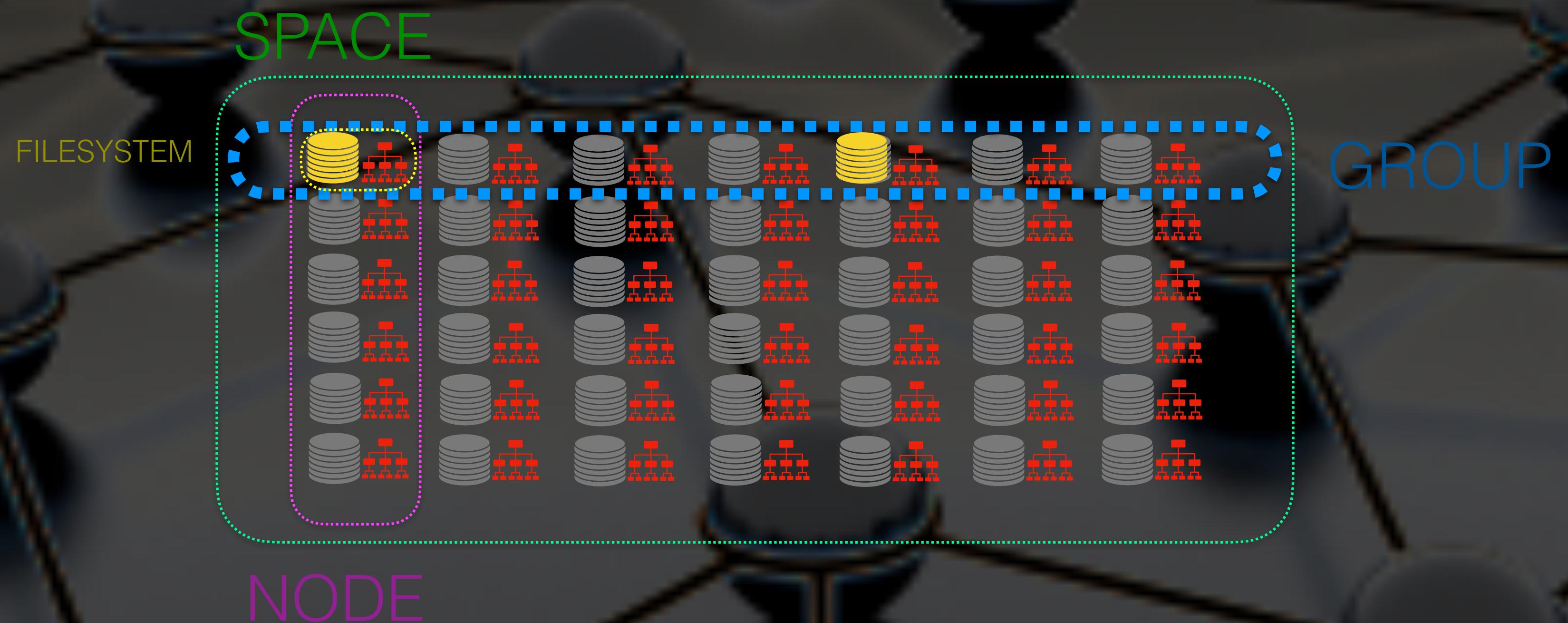
type	name	groupsize	groupmod	N(fs)	N(fs-rw)	sum(usedbytes)	sum(capacity)	capacity(rw)	nom.capacity	quota	balancing	threshold	converter	ntx	active	wfe	ntx	active	intergroup
spaceview	default	20	48	360	360	2.16 TB	4.32 PB	4.32 PB	0 B	off	off	20	on	400	0	off	1	0	off
spaceview	erasure	24	252	6168	6059	31.47 PB	85.14 PB	83.31 PB	0 B	off	off	6	on	400	0	off	1	0	off
spaceview	test	24	4	96	96	22.29 TB	1.15 PB	1.15 PB	0 B	off	off	20	on	2	0	off	1	0	off



File Placement in EOS

algorithm placing
2 replica

select group 0
select 2 disk



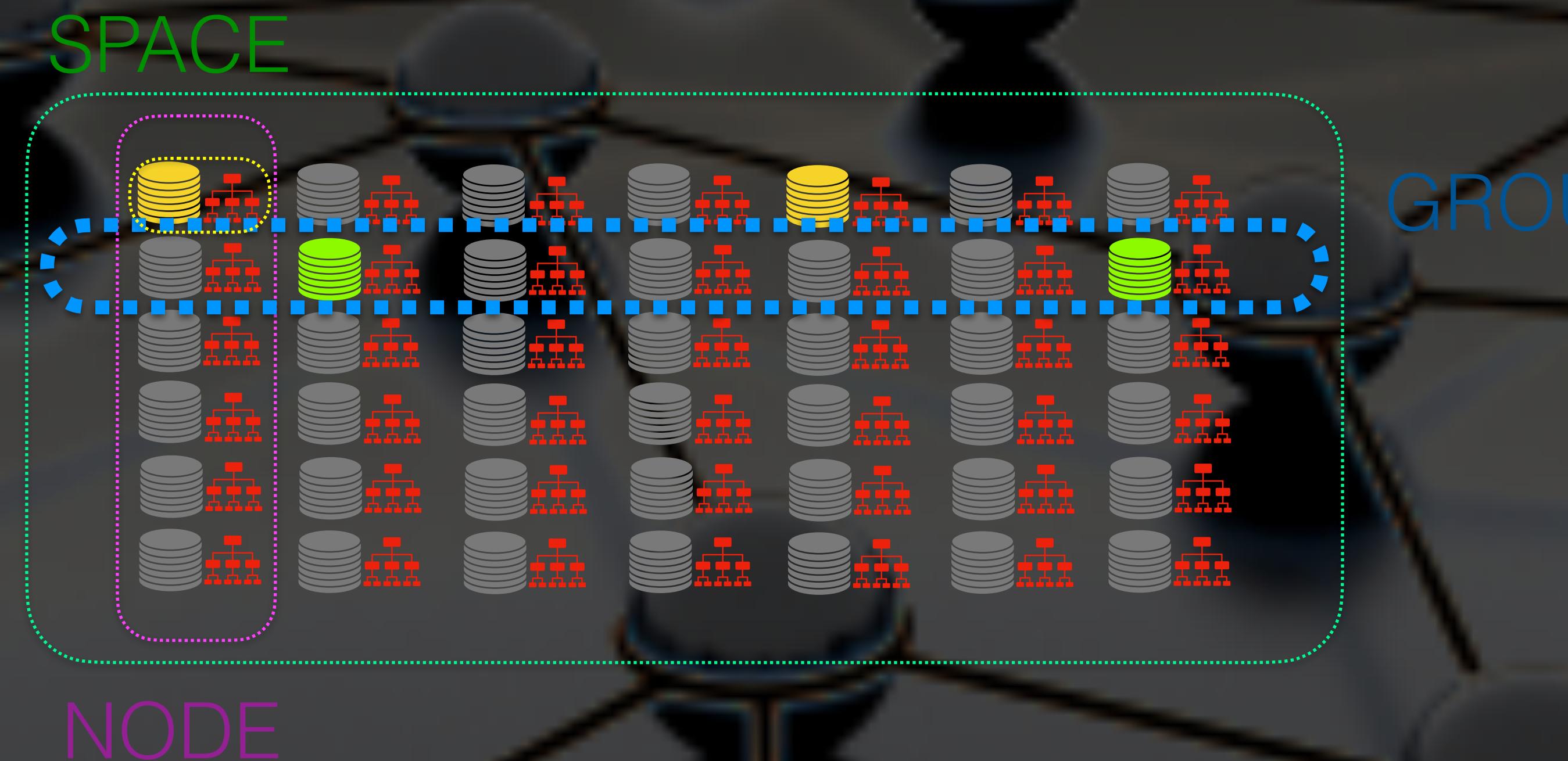
Placement Algorithm cycles through all the groups and picks N random available locations.
N=1 for single replica N=2 dual replica or e.g. N=6 EC(4+2)



File Placement in EOS

algorithm placing
2 replica

select group 1
select 2 disk



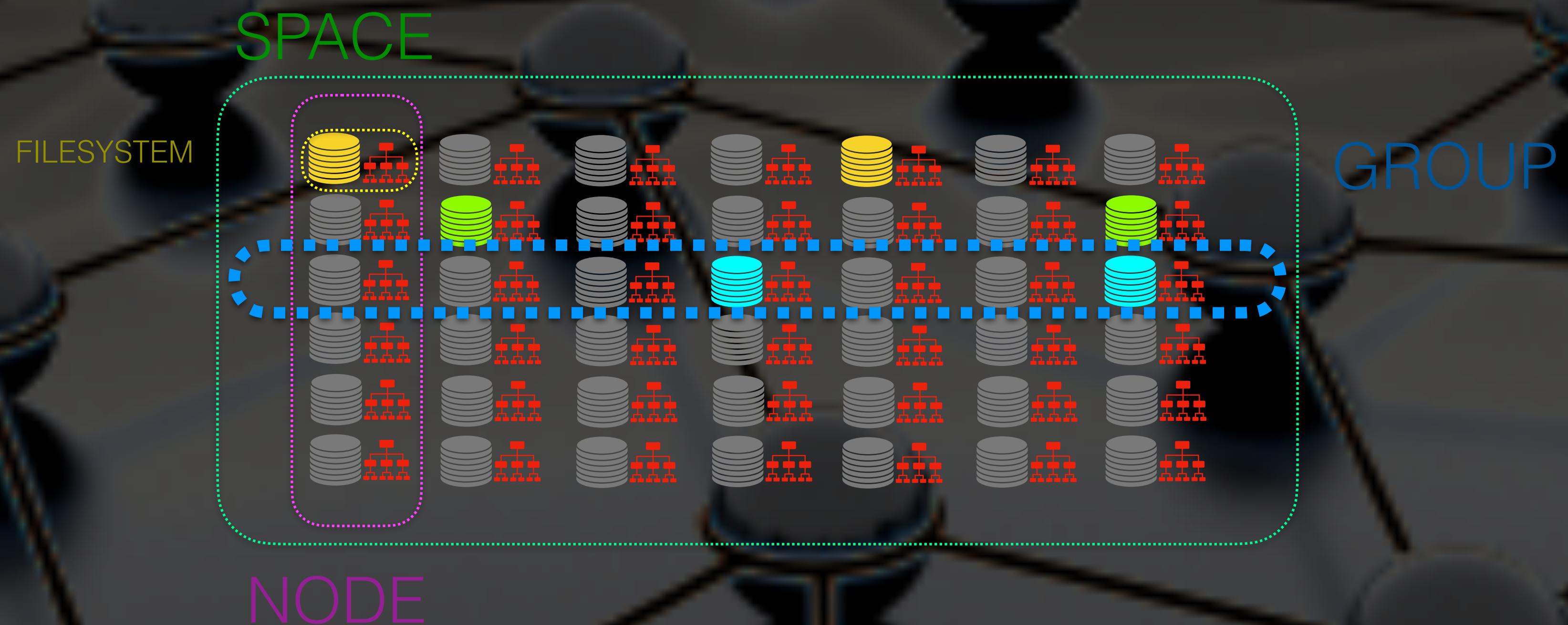
Placement Algorithm cycles through all the groups and picks N random available locations.
N=1 for single replica N=2 dual replica N=6 EC(4+2)



File Placement in EOS

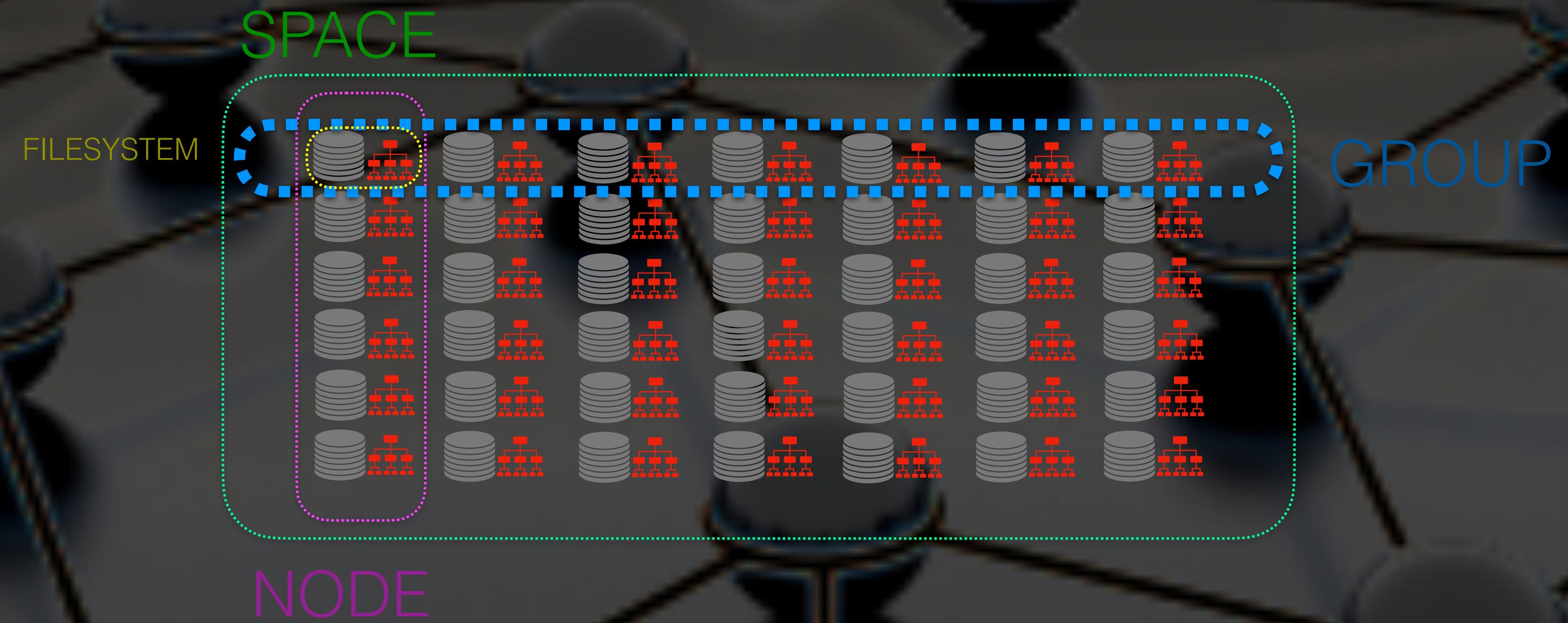
algorithm placing
2 replica

select group 2
select 2 disk



Placement Algorithm cycles through all the groups and picks N random available locations.
N=1 for single replica N=2 dual replica N=6 EC(4+2)

File Placement in EOS



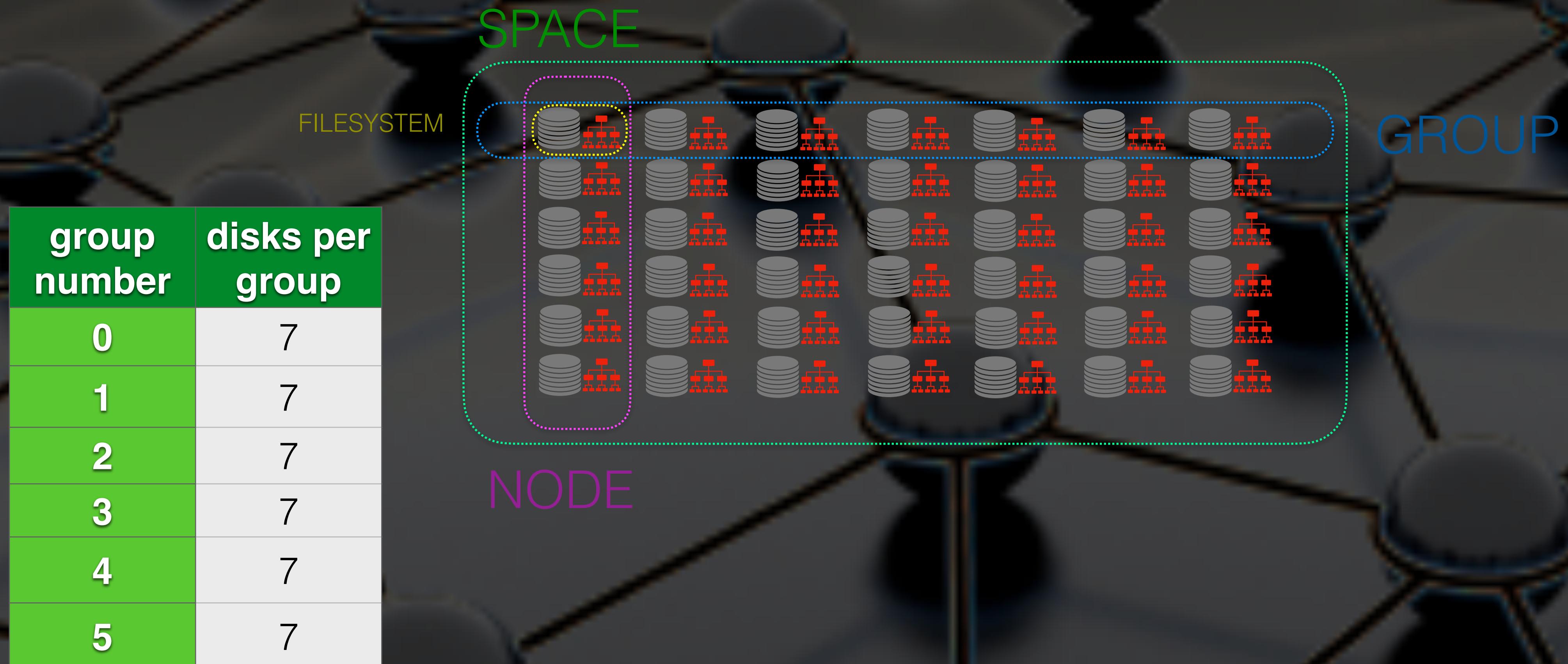
What you have to take into consideration:

1. **number of disks** in all scheduling groups **is equal** or very similar
2. the **storage space** per scheduling group should be **similar**, but it less important
3. in small setups: groups should have **at least 3 disks** and preferably an **uneven** number of disks



Scheduling Group Planning

The good case: all nodes have the same amount of HDDs of equal size!

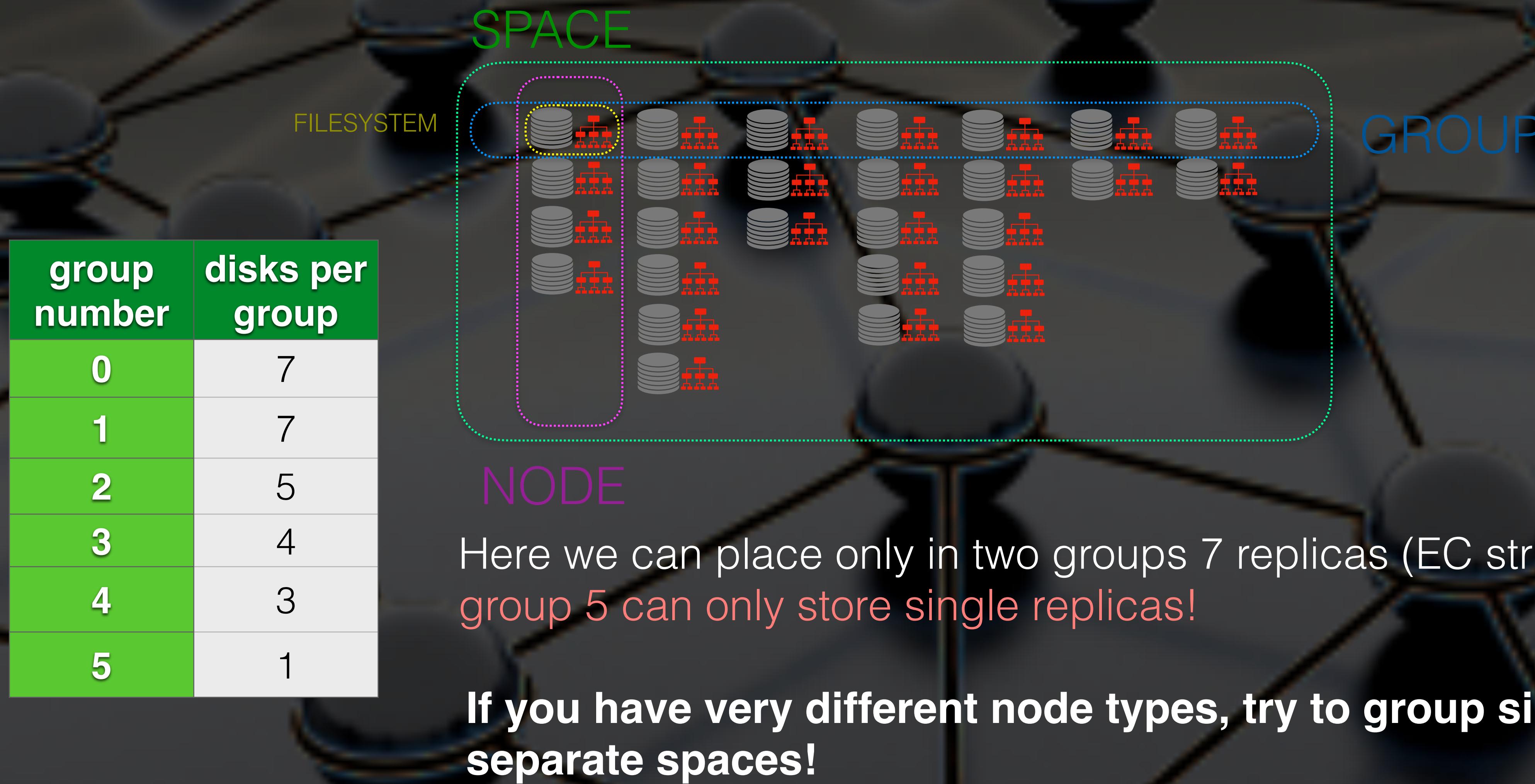


The performance/capacity ratio is the same in each group e.g. all groups are equally performant and can host 7 replicas/EC stripes.



Scheduling Group Planning

A bad case: nodes have varying number of HDDs





Multiple Spaces



In EOS you can define very flexible policies, which space to use in which circumstances!

Resource Optimisations

“Not enough memory - old CPUs - not enough disk/network”



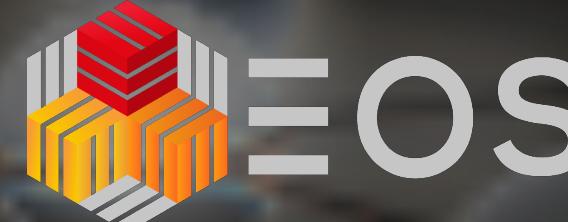
Reducing the MGM memory

- 1 **Reduce** the XRootD Threadpool in the Mgm XRootD configuration file (/etc/xrd.cf.mgm)

```
xrd.sched mint 64 maxt 4096 idle 300  
xrd.sched mint 16 maxt 256 idle 300
```

- 2 **Reduce** the file and directory container MD cache limits

```
 eos ns cache set -f 1000000  
 eos ns cache set -d 1000000
```



Reducing Disk Space Requirements

1 **Reduce** the EOS daemon log level from INFO to NOTICE

```
eos debug notice /eos/*/mgm  
eos debug notice /eos/*/fst
```

2 **Disable** report log files in /var/eos/report:

```
 eos io disable -r
```

3 **Reduce** the size of the QDB Raft journals:

```
redis-cli -p 7777 config-set raft.trimming 1000000:200000
```

see QuarkDB Documentation



Reducing CPU Requirements

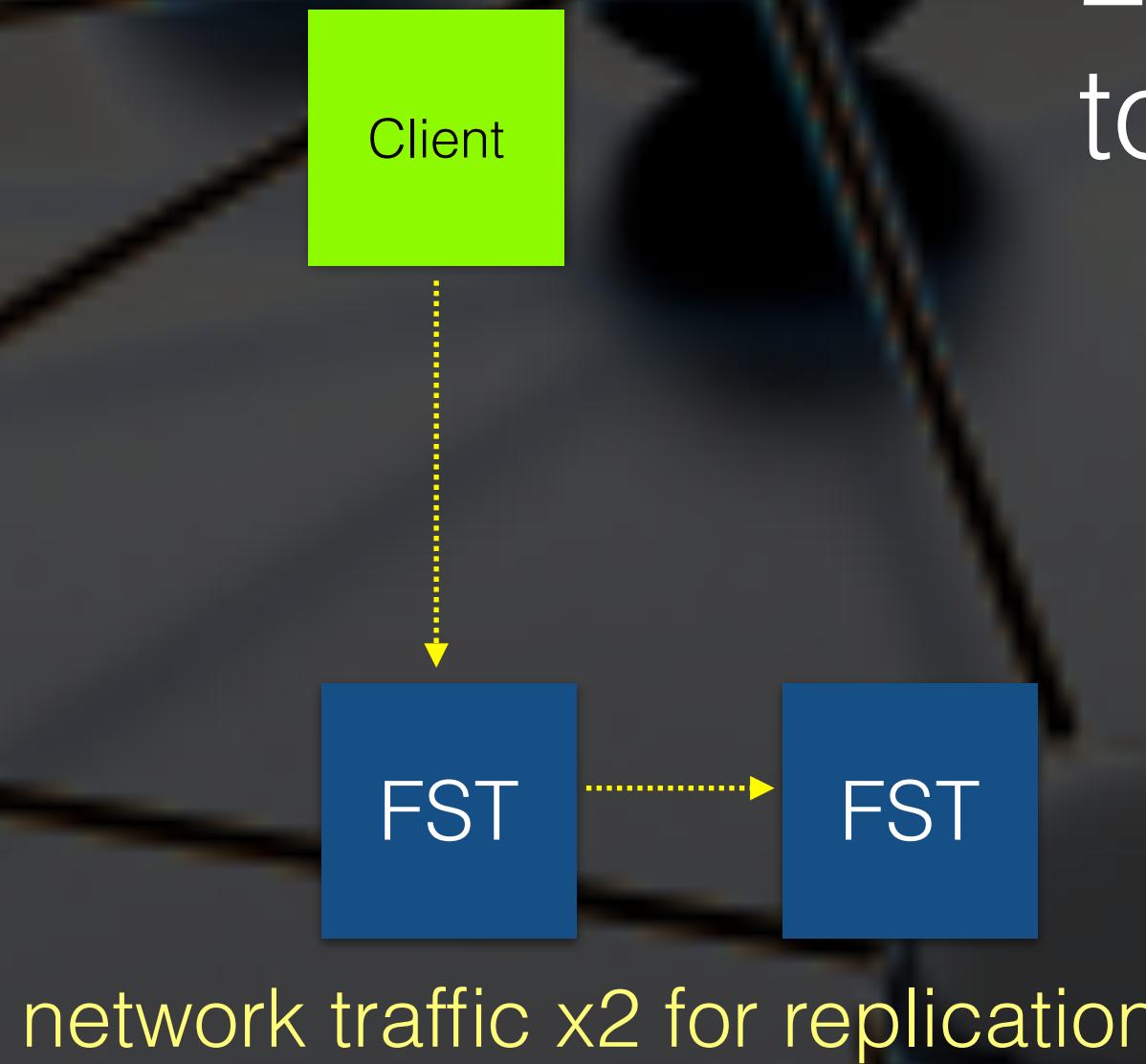
- 1 **avoid** erasure coding on old CPUs!
- 2 use **HW RAID** instead of software RAID or replication

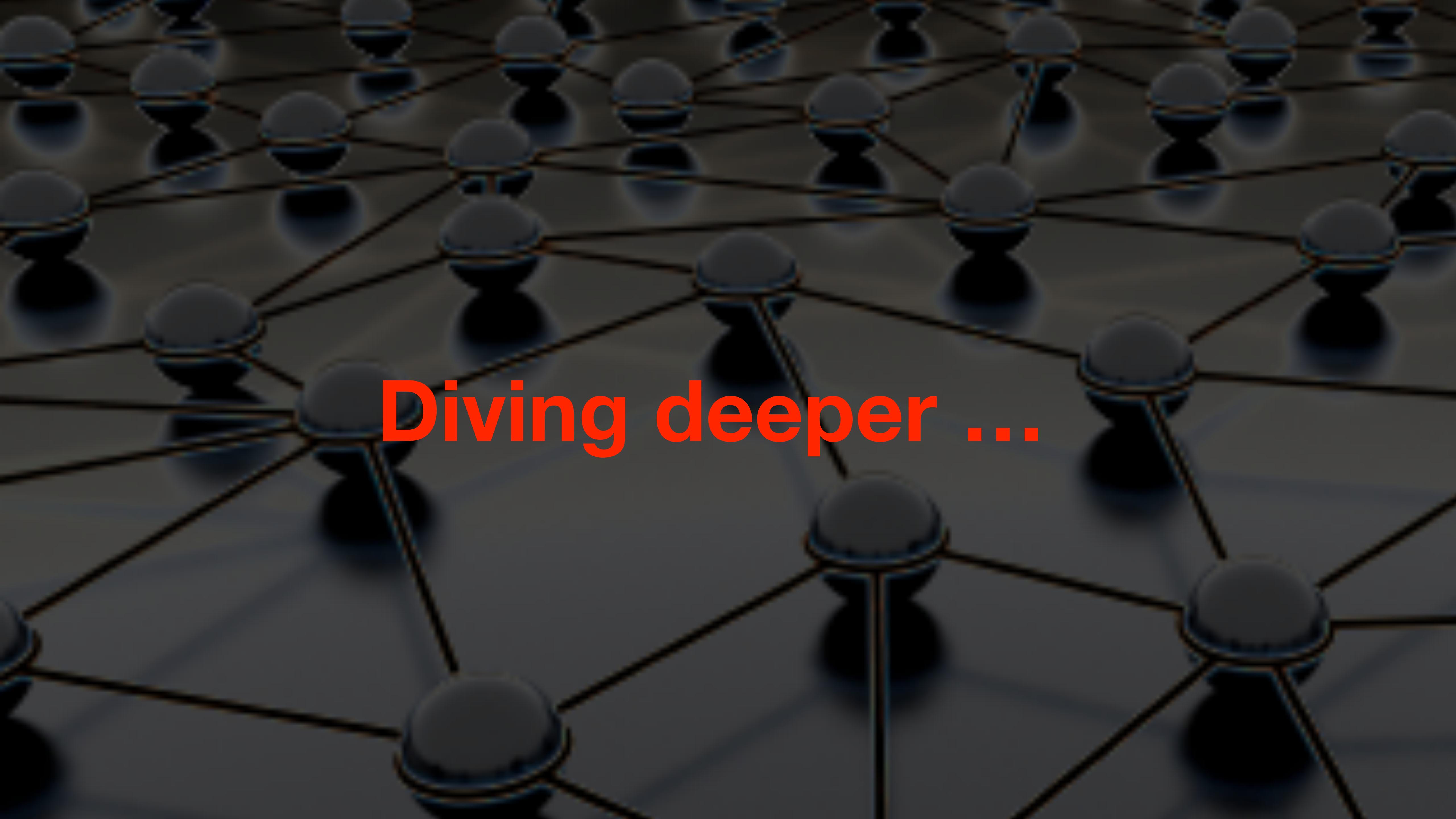


Reducing Network Requirements

- 1 **avoid** erasure coding if you have problems with network bandwidth
 - erasure coding creates traffic amplification x2 for read and write

- 2 use **HW RAID** instead of software RAID or replication to reduce network traffic





Diving deeper ...



XRootD Framework

EOS is 99% written in C++ (17)

- XRootD: 230k lines of C++
- EOS: 290k lines of C++

All EOS Services (daemon) are implemented as plug-ins into the **xrootd** process

A screenshot of a web browser window displaying the XRootD homepage. The address bar shows the URL https://xrootd.slac.stanford.edu. The page features a red header with the XRootD logo, which is a blue and white molecular-like icon, and the word "XRootD". Below the header is a navigation menu with links for "home", "download", "docs", "development", "collaboration", "contact", and "archive". The main content area contains a welcome message and a detailed description of the XROOTD project's goals and architecture. A footer at the bottom provides additional information about the software's features and performance.

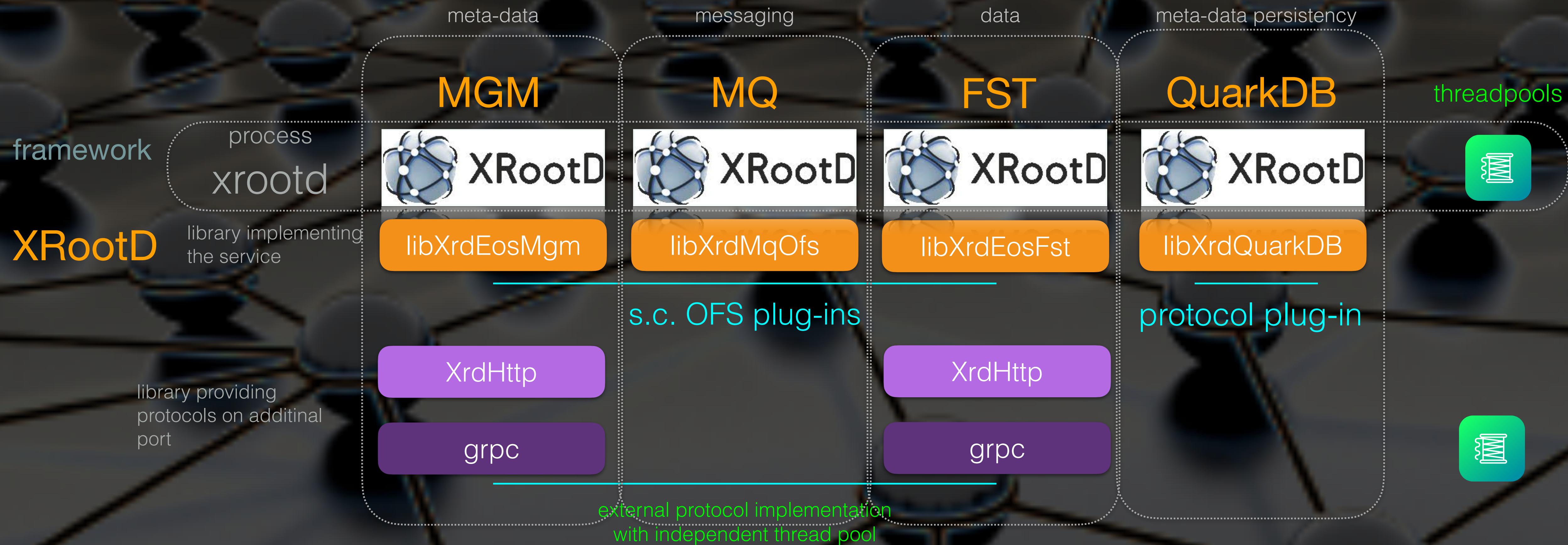
Welcome to the XRootD webpage

The XROOTD project aims at giving high performance, scalable fault tolerant access to data repositories of many kinds. The typical usage is to give access to file-based ones. It is based on a scalable architecture, a communication protocol, and a set of plugins and tools based on those. The freedom to configure it and to make it scale (for size and performance) allows the deployment of data access clusters of virtually any size, which can include sophisticated features, like authentication/authorization, integrations with other systems, WAN data distribution, etc.

XRootD software framework is a fully generic suite for fast, low latency and scalable data access, which can serve natively any kind of data, organized as a hierarchical filesystem-like namespace, based on the concept of directory. As a general rule, particular emphasis has been put in the quality of the core software parts.

XRootD Framework

xrootd.org



There are **additional plug-ins** used in MGM, MQ and FST services for **authentication, authorization** (e.g. alice token), additional native protocols plugins like http ...



Supported Protocols

client apps internal protocols

CLI uses root://

FUSE mount root://+zmq://

root://
http(s)://

XrdXrootd
users + internal
communication
XrdHttp
users

native protocol

grpc://
ZMQ://

grpc
users [reva]
ZMQ
eosxd [fuse]

native plug-in protocol



S3
sftp

add-on protocols

external protocols
via third-party gateways



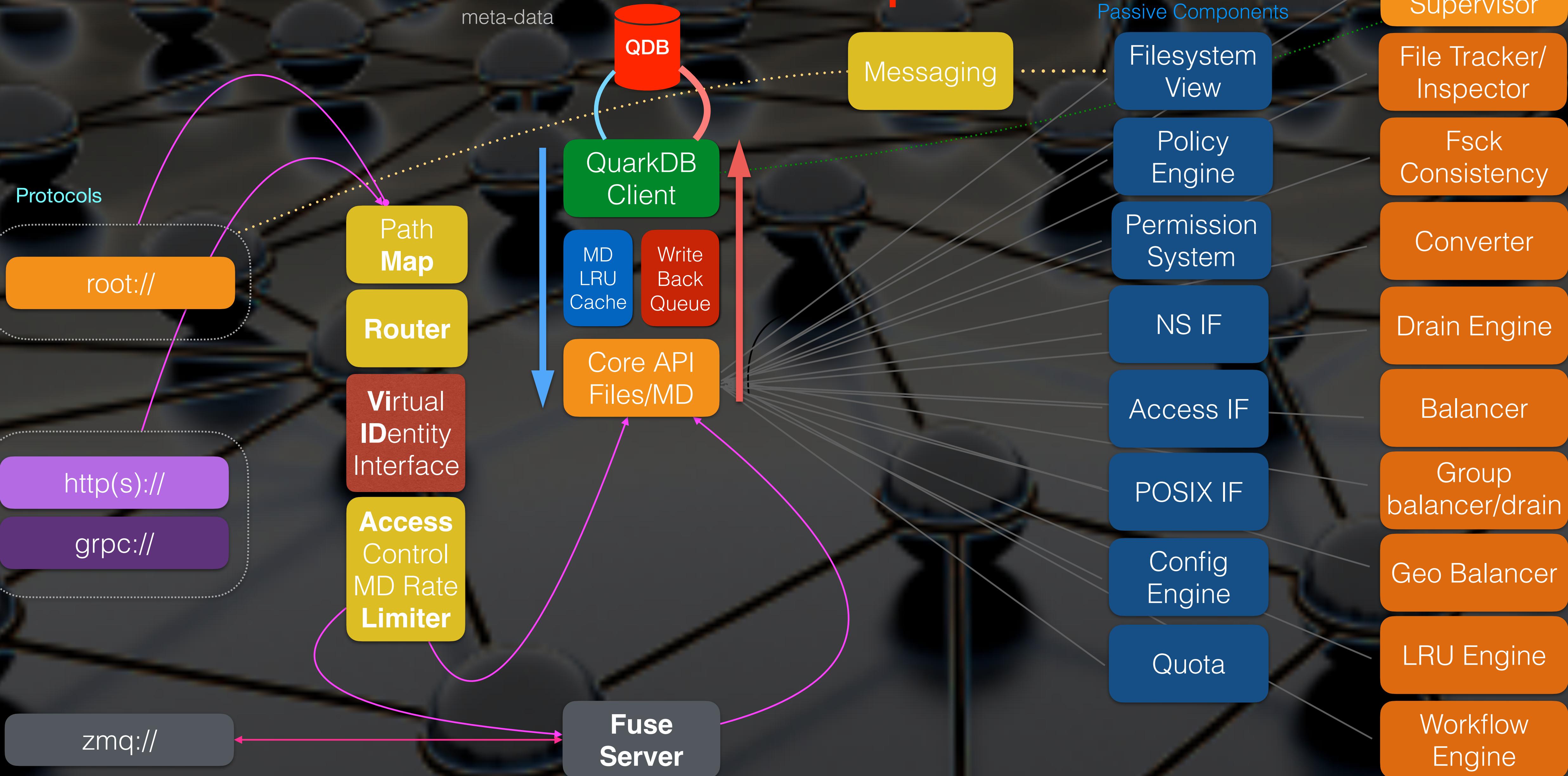
MGM - the heart of EOS

The **EOS MGM** is the central and most **complex** component of EOS

- externally looks like a simple process
- internally runs many different active and passive components, thread pools and protocols
 - many features are disabled by default
 - these are required only in more complex setups or for hardware exchange

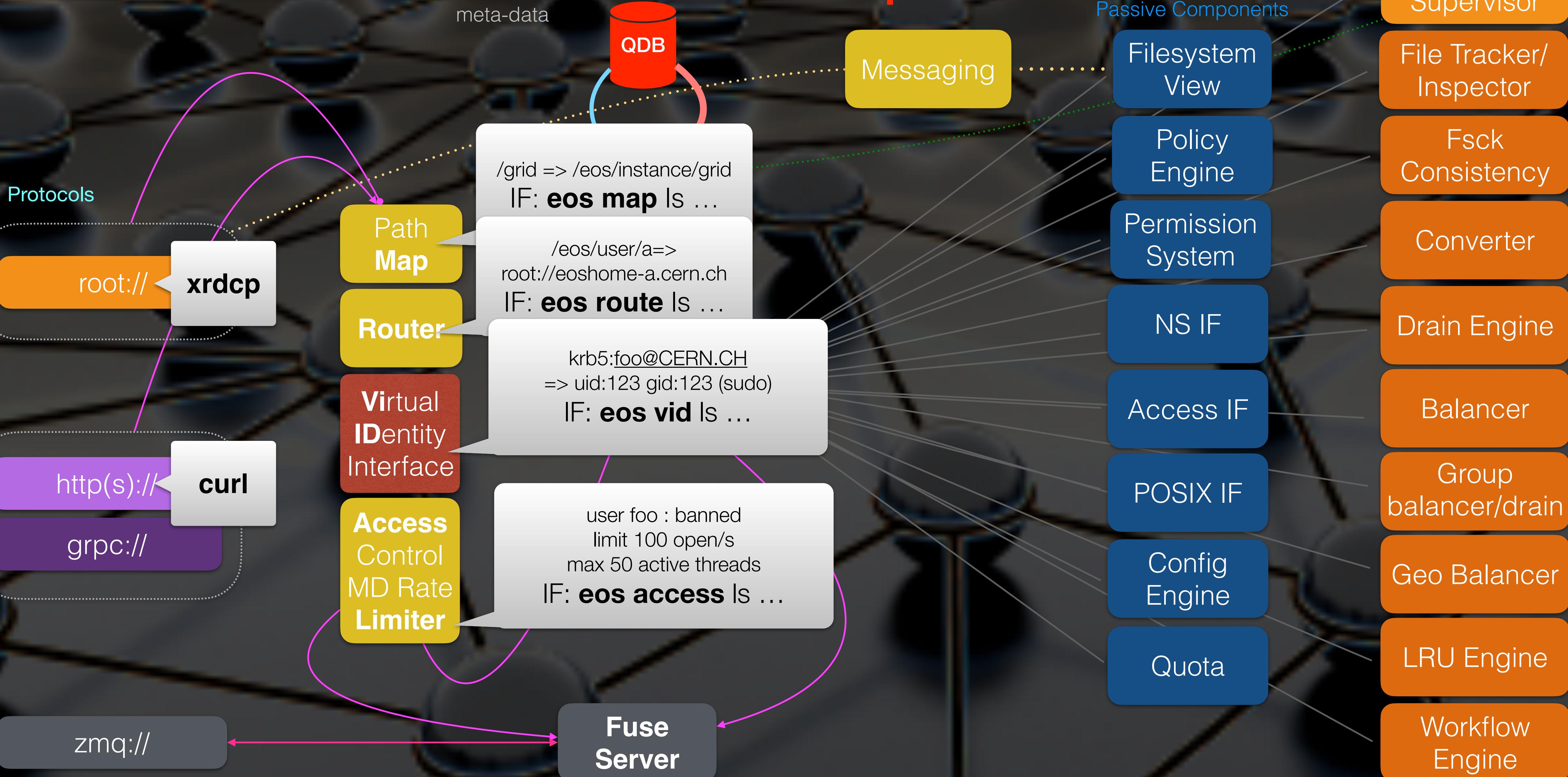


MGM Components



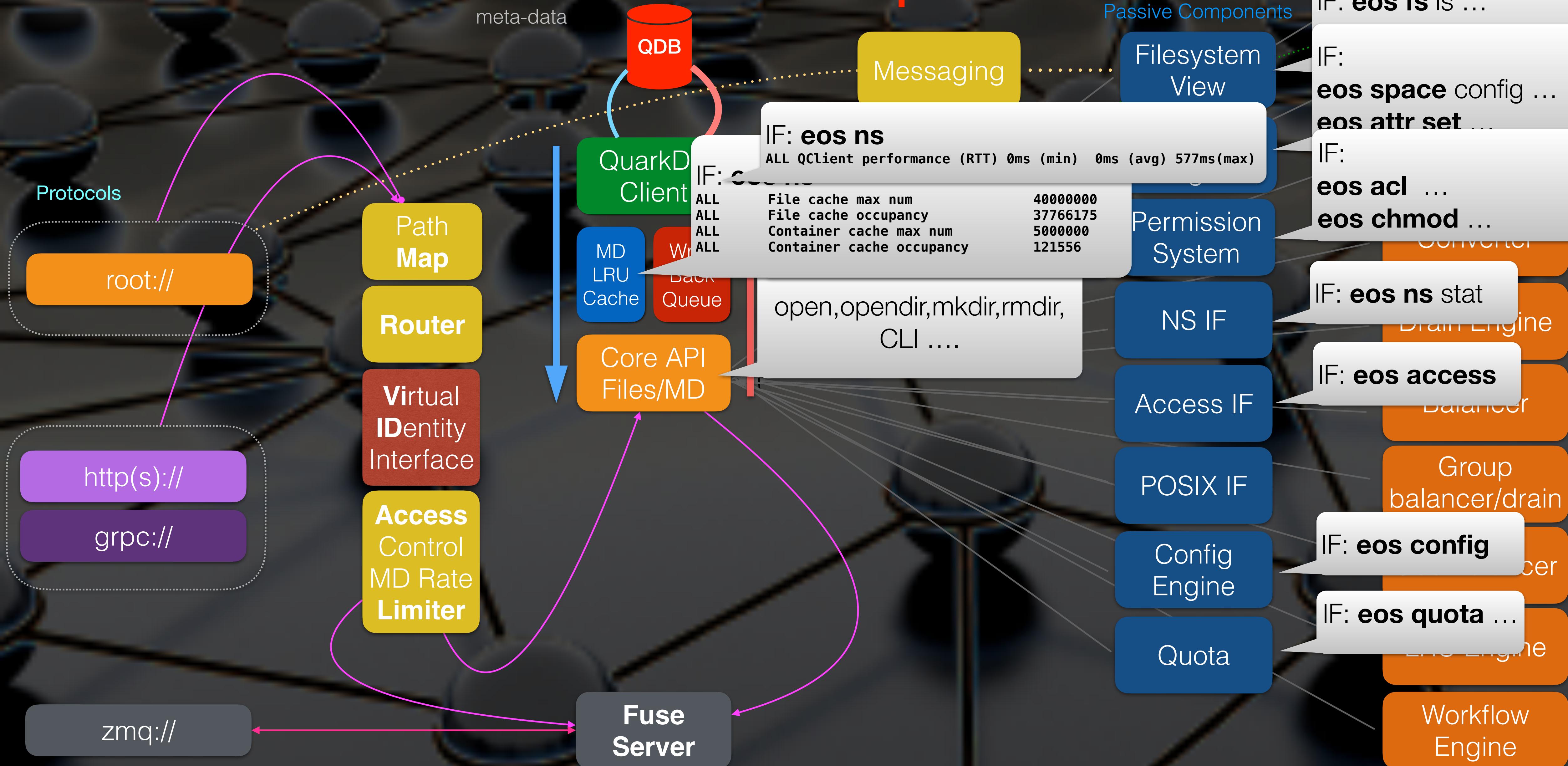


MGM Components



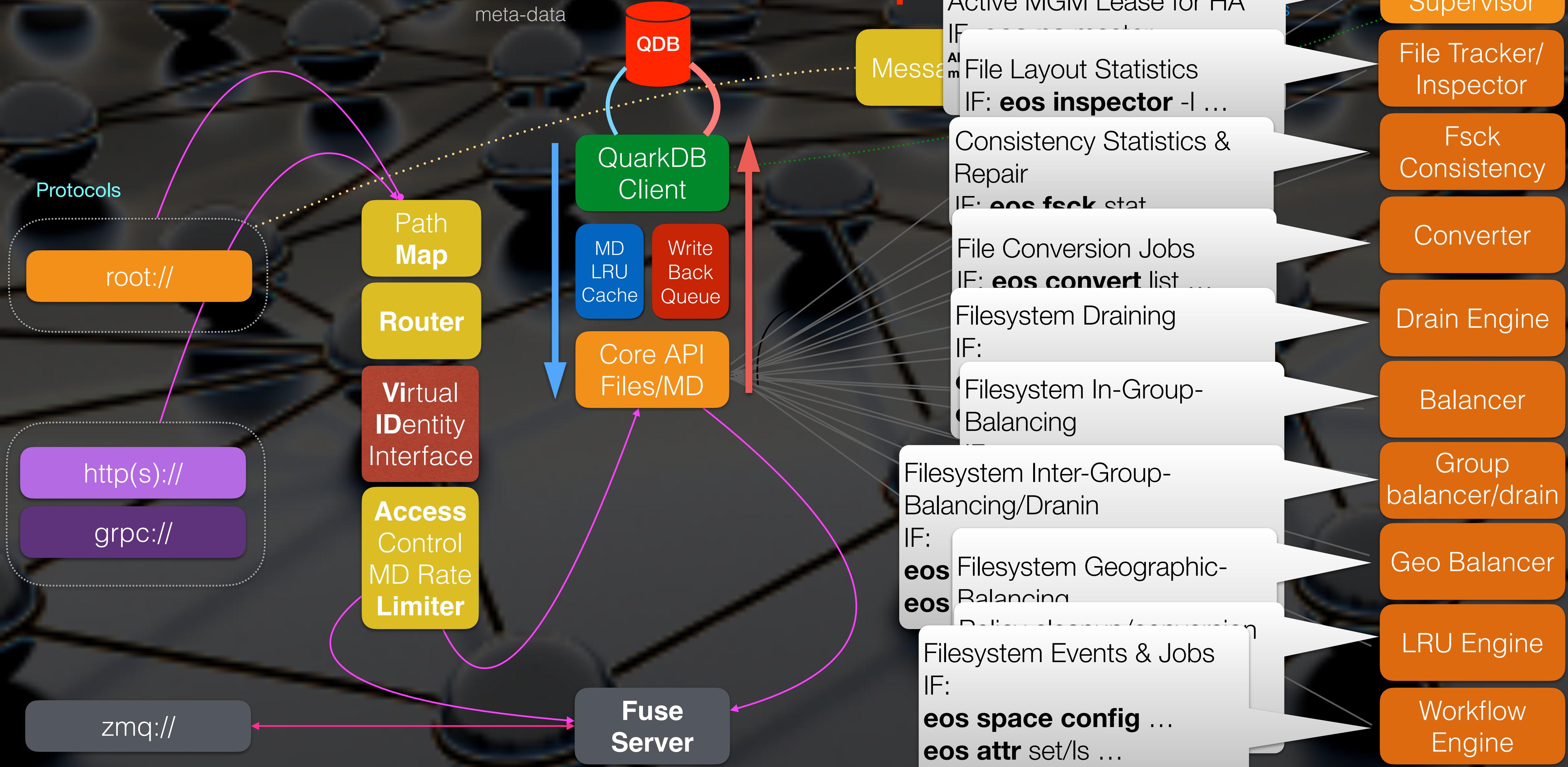


MGM Components





MGM Components





EOS MGM Minimum Feature Set

Active Components

Master
Supervisor

File Tracker/
Inspector

Fsck
Consistency

Converter

Drain Engine

Balancer

Group
balancer/drain

Geo Balancer

LRU Engine

Workflow
Engine

black components are
unused in a simple setup

Protocols

root://

http(s)://

grpc://

zmq://

meta-data



QuarkDB
Client

MD
LRU
Cache

Write
Back
Queue

Core API
Files/MD

Virtual
IDentity
Interface

Access
Control
MD Rate
Limiter

Path
Map

Router

Fuse Server

Messaging

Passive Components

Filesystem
View

Policy
Engine

Permission
System

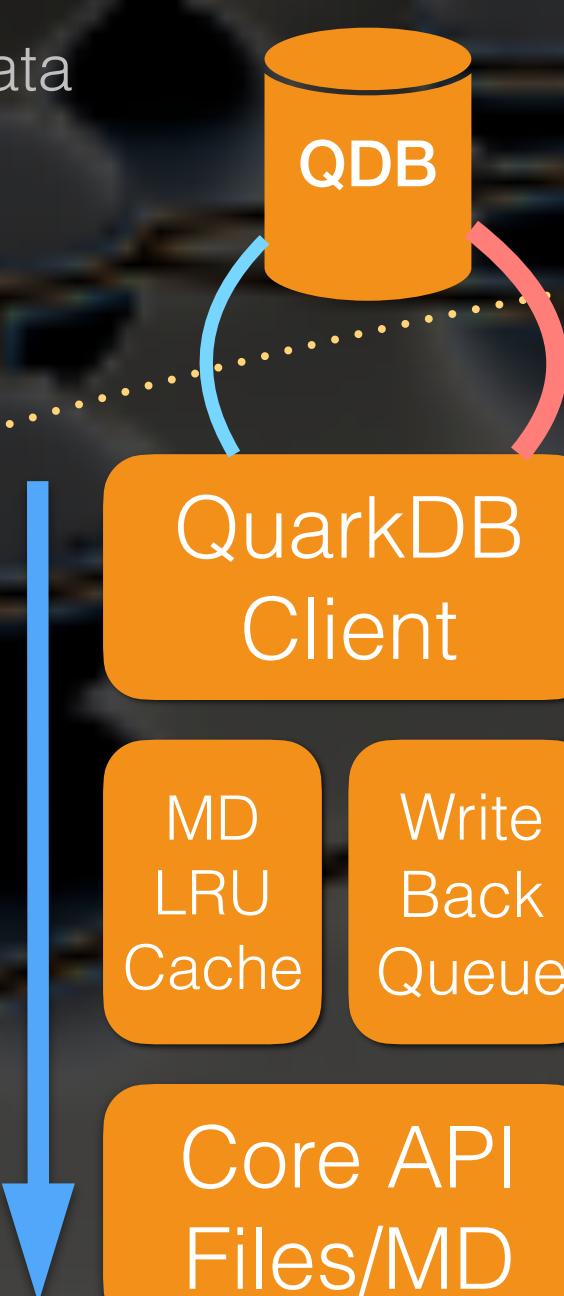
NS IF

Access IF

POSIX IF

Config
Engine

Quota

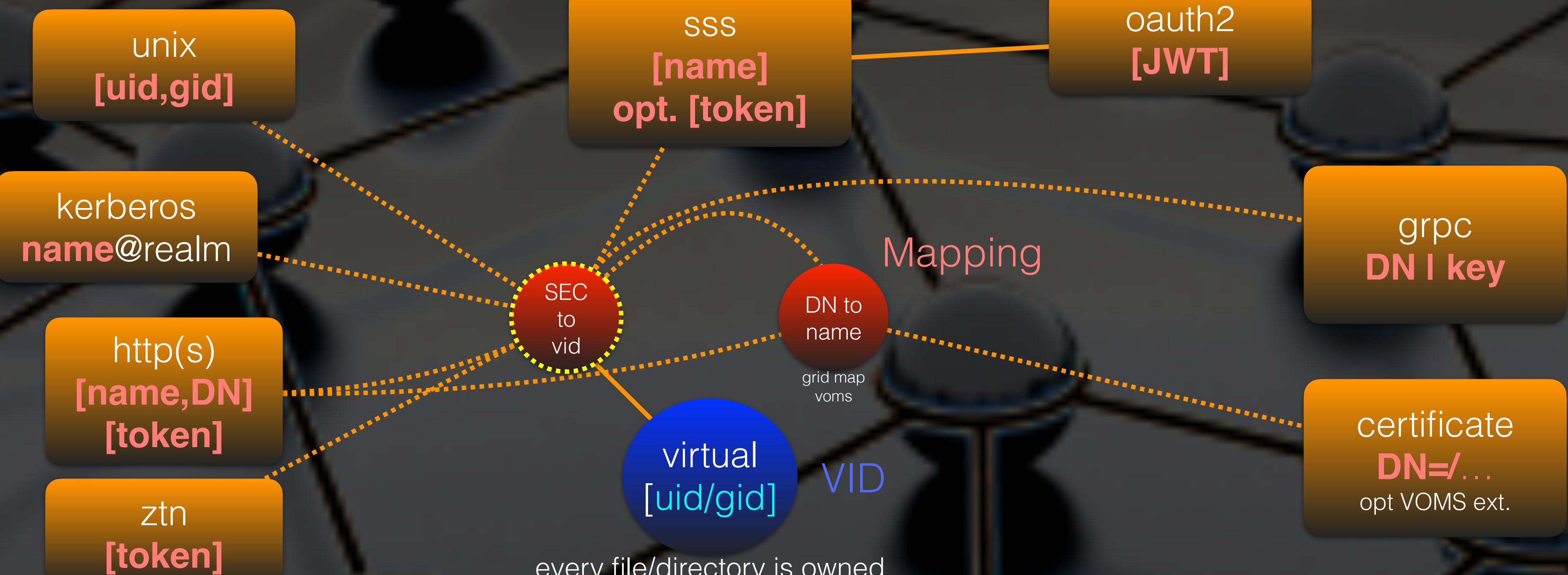


Virtual Identity Concept

Authentication

every [client sec,request] is mapped internally to a virtual identity pair [uid,gid]
based on the used authentication method and an optional requested role [ruid,rgid]

Authentication
method



```
EOS Console [root://localhost] | /eos/ajp/> ls -ln oauthfile
-rw-r--r-- 1 100755 1338 VID          1824 Sep 18 2019 oauthfile
EOS Console [root://localhost] | /eos/ajp/> ls -l oauthfile
-rw-r--r-- 1 apeters vl convenience 1824 Sep 18 2019 oauthfile
```

EOS Support & Resources

EOS has solid support from CERN

- crucial software for physics data storage
- crucial for user data storage CERNBOX
- crucial for archival storage CTA

EOS has an increasing community and is deployed in tens of storage installations world-wide

- RAL & CNAF exploring EOS and CTA
- many WLCG Tier-2 sites
- other sciences like JRC

Business support provided by [COMTRADE 360](#)

- Comtrade Fast FileSystem (CFFS)
- Windows Native client
- Ad-hoc features

EOS services at CERN run with **best effort support** only

EOS Resources

Web Page: <https://eos.web.cern.ch>

GIT Repository: <https://gitlab.cern.ch/dss/eos>

Community

Forum: <https://eos-community.web.cern.ch/>

email: eos-community@cern.ch

Documentation:

<http://eos-docs.web.cern.ch/eos-docs/>

<https://github.com/cern-eos/eos/wiki>

Support email: eos-support@cern.ch

EOS 5

Codename: Diopside



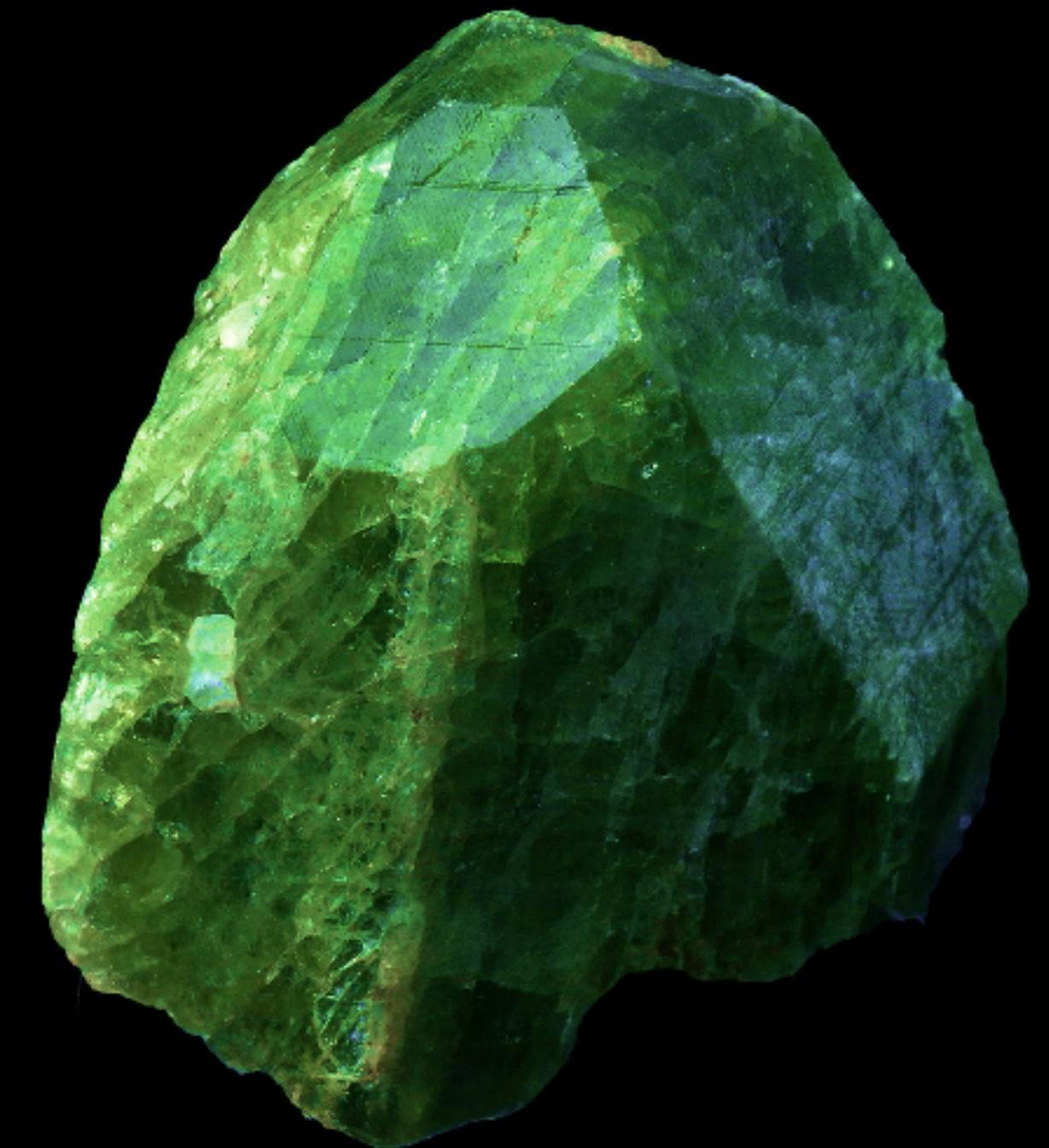
EOS



Showcasing EOS Instance Configuration

Elvin Alin Sindrilaru

Here is where **the workshop**
will continue tomorrow ...



Introduction to CTA

Michael Davis

Thank you for your attention!

Questions / Comments ?

<https://indico.cern.ch/event/1227241/>



The screenshot shows the header of the Indico event page. At the top left is the EOS logo (a stylized orange cube icon) followed by the text "EOS 2023 Workshop". To the right of the logo is the date "24–27 Apr 2023", the location "CERN", and the time zone "Europe/Zurich timezone". On the far right is a search bar with the placeholder "Enter your search term" and a magnifying glass icon.

