EOS Basic Concepts & Design

EOS Workshop 2021

Andreas-Joachim Peters
CERN IT-ST for the EOS team

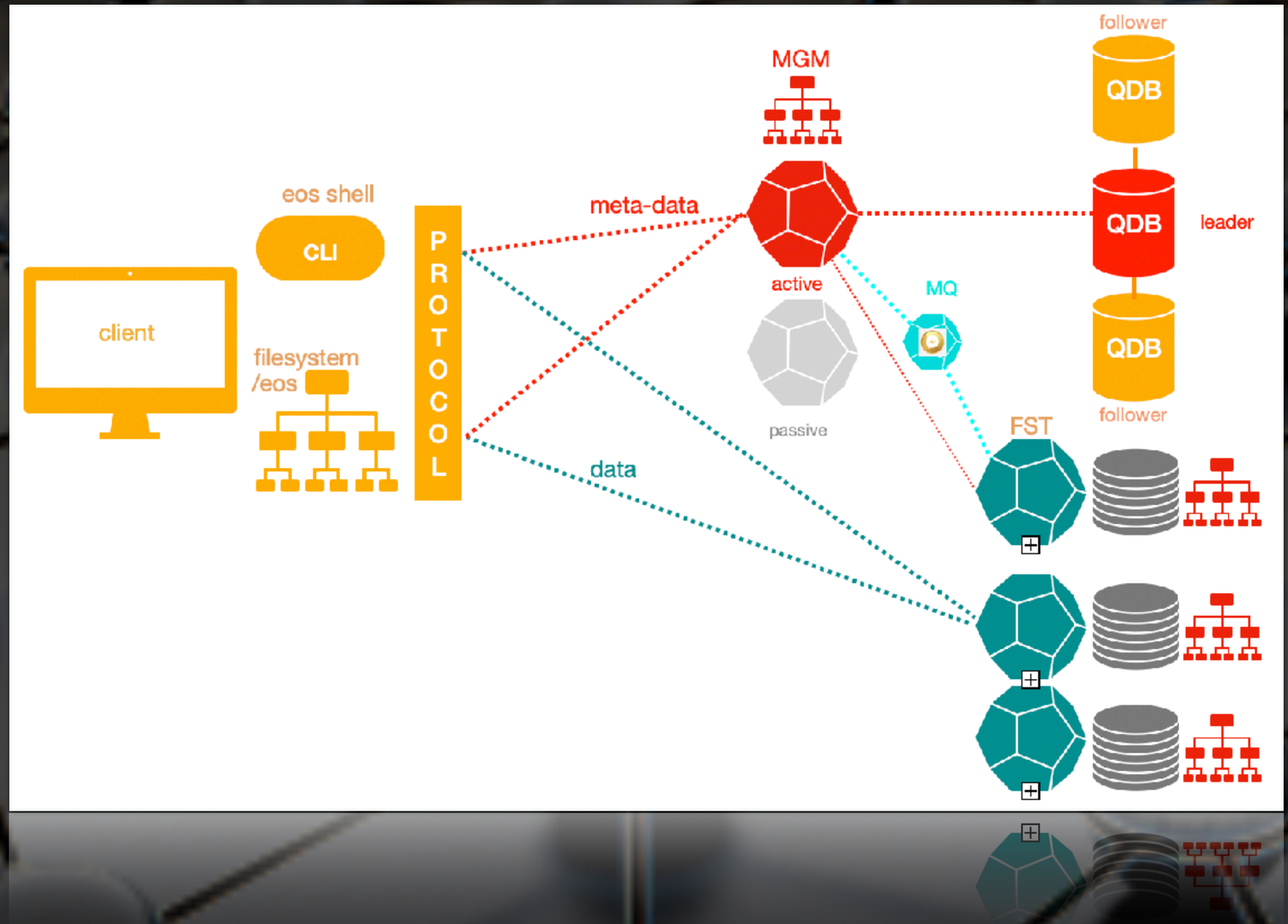
Current Architecture

framework

XRootD

components

CLIENTs
MGM
MQ
FST
QuarkDB



IO path

meta-data
data

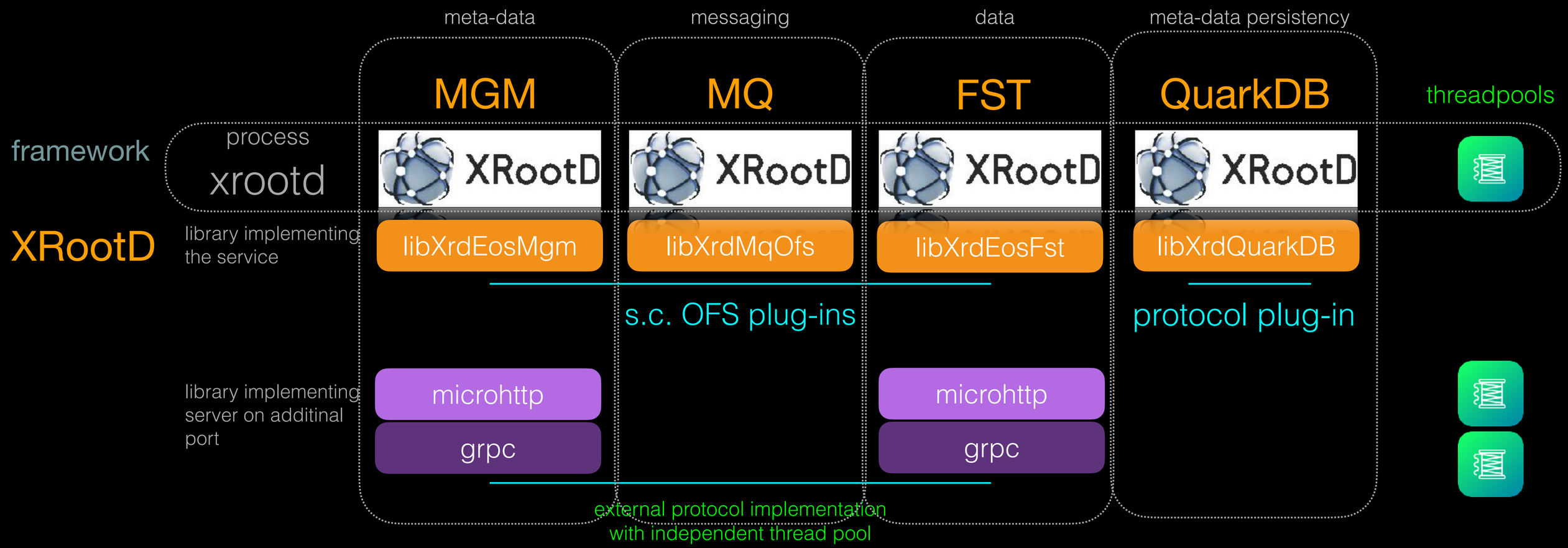
protocols

sync
async

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

XRootD Framework

xrootd.org



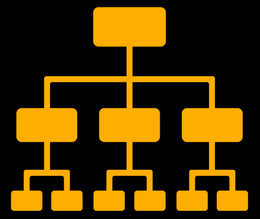
there are **additional plug-ins** used in MGM,MQ and FST services
for **authentication, authorization**, additional native protocols plugins like http …

Hierarchical and Object Namespace

- **MGM** service provides a **hierarchical namespace** starting at / (inode 1)
  - directories and files are arranged in a tree structure with parent child relation
    - children of directories are files or directories
    - directories and files have a numerical ID (inode number = object ID called *fid*)
  - bi-directional lookup
    - path → inode /eos/foo/bar => inode 1 [/] → inode 2 [eos] → inode 3 [foo] → inode 100 [bar]
    - inode → path inode 100 [bar]→ inode 3 [foo]→ inode 2 [eos]→ inode 1 [/]= /eos/foo/bar
  - historical complication inode != id because in-memory namespace had overlapping id namespace for files and directories and filesystems don't allow the same id for a file and directory (VFS)

name	id	parent id	type
/	1	1	dir
eos	2	1	dir
foo	3	2	dir
bar	100	3	file

- **FST** services use **flat inode namespace** (object storage approach) for files
  - flat node namespace is mapped into pseudo hierarchy
    - fid → path fid 100 stored with name / <prefix> / *hex*(fid / 10000) / *hex* (fid) e.g. **`/data/01/00000000/00000a84`**



Namespace and QuarkDB

- MGM service uses a **configurable amount of memory** to cache file and directory meta-data using an **LRU mechanism**
- file and directory meta-data are **persisted in QuarkDB**
- MGM uses a **write-back mechanism** to send writes to QuarkDB (async)
- Every meta-data access in the MGM uses a **prefetch mechanism** to make sure meta-data is in the meta-data cache **before** taking a **namespace lock**
  prefetch takes ±0.5 .. few ms

```
# ------------------------------------------------------------------------------
ALL      File cache max num                20000000
ALL      File cache occupancy              13106502
ALL      In-flight FileMD pre-fetch        0
ALL      Container cache max num           2500000
ALL      Container cache occupancy         2436582
ALL      In-flight ContainerMD pre-fetch   0
```

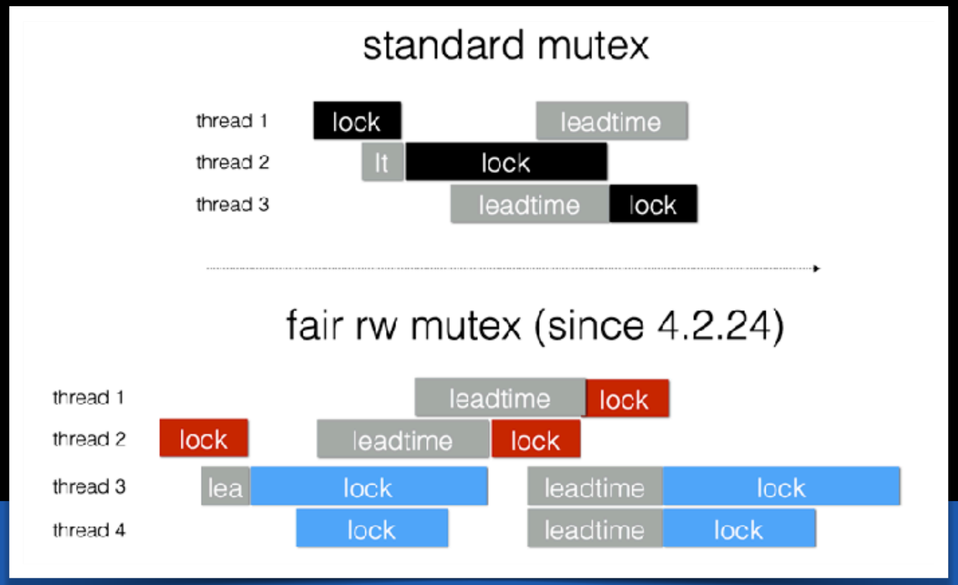
cache

cache

bash> eos ns

Namespace Locking

- A hierarchical structure requires locking when the hierarchy is changing
- example: mv /eos/foo/file /eos/bar/file
  - file is at any moment attached to foo or bar, but neither invisible nor visible in both directories
- EOS uses a **global fair RW lock** inside the MGM service

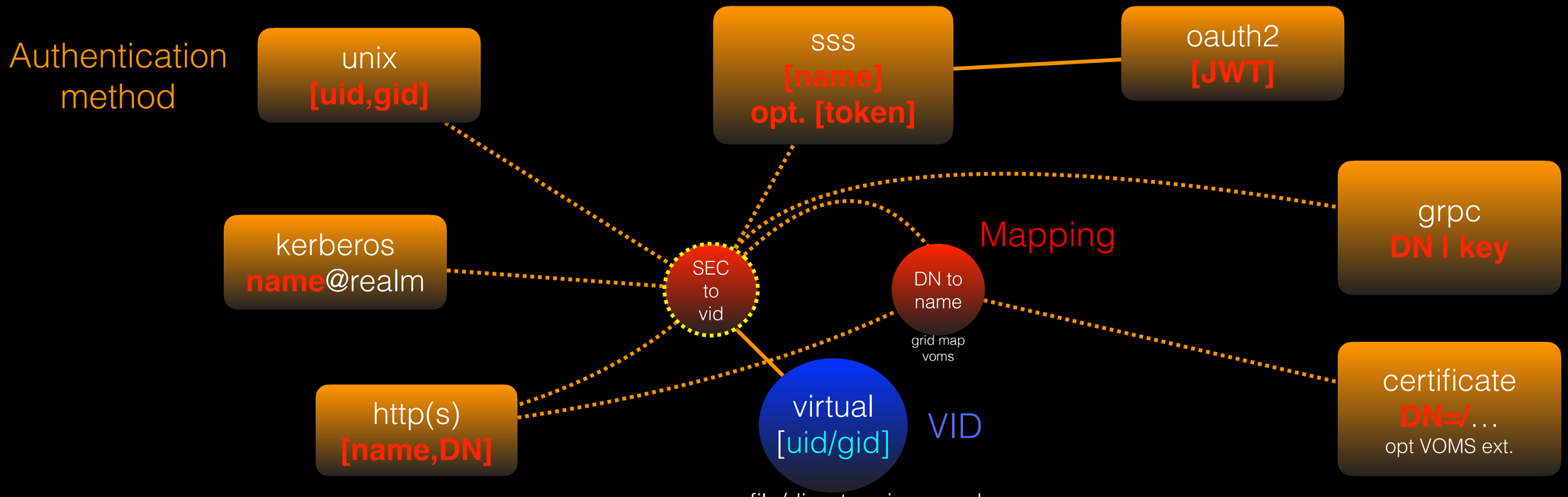


- writers operate exclusive
- readers go in parallel
- future readers queue behind waiting writers

Result:
performance is depending on ratio read:write MD ops

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]



every file/directory is owned
by some [uid,gid], never by a name, certificate or token

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

VID

convenience

Virtual Identity Mapping

- mapping is done from security context [XrdSecEntity] created by a connection based on the authentication method to VID

```
/*----------------------------------------------------------------------------*/
void
Mapping::IdMap(const XrdSecEntity* client, const char* env, const char* tident,
               VirtualIdentity& vid, bool log)
```

- mapping can be static or dynamic
  - static: map all clients using UNIX authentication to a static VID [100,100]
  - dynamic: map all kerberos names using the UNIX getpwd mechanism
    e.g. user foo is mapped in LDAP to VID [1234,1234]

```
[root@ajp ~]# eos vid ls | grep unix
tident:"unix@facebook":gid => root
tident:"unix@facebook":uid => root
unix:"<pwd>":gid => nobody
unix:"<pwd>":uid => nobody
```

- mapping can rely on external plug-ins (e.g. VOMS, OAUTH2)

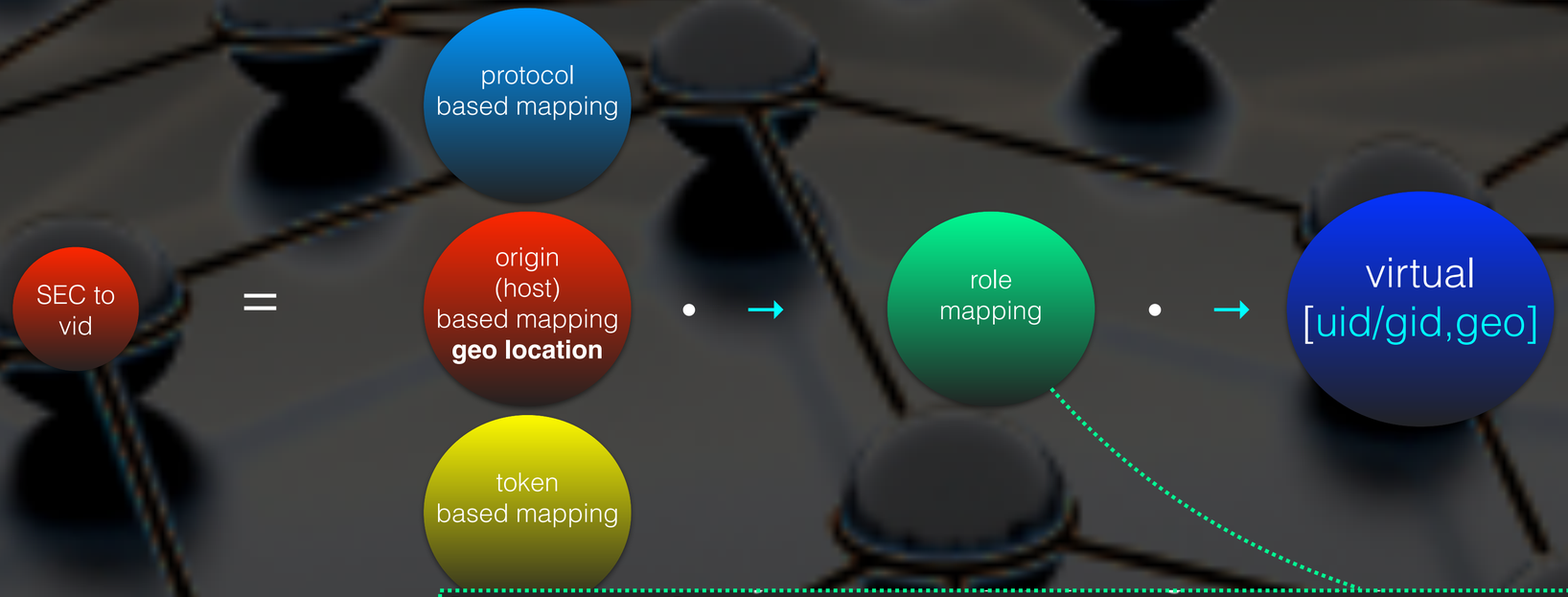
- VIDs can be granted to be **sudo'er**
  - every **sudo'er** can add to a request a role VID, for whom a given request is executed

```
[root@ajp ~]# eos vid ls | grep sudoer
sudoer                    => uids(daemon,admin)
```

- The **sudo'er** functionality can be also granted based on the client hostname/IP
  - a host which has sudo'er privileges is called a **gateway**

Virtual Identity Mapping

```
/*----------------------------------------------------------------------------*/
void
Mapping::IdMap(const XrdSecEntity* client, const char* env, const char* tident,
               VirtualIdentity& vid, bool log)
```




```
[root@ajp ~]# eos -r 1000 1000 whoami
Virtual Identity: uid=1000 (2,3,99,1000) gid=1000 (4,99,1000) [authz:sss] sudo* ho
```

```
[root@eoshome-ns-i01-00 (mgm:master mq:master) ~]$ eos ns stat | grep IdMap
 all IdMap                          1.01 G   1307.50   1208.85   1174.41   1181.54      0.05     ± 0.02
```

executions rate 5s 5m 1h 24h ms variation

Virtual Identity Mapping YOU

How does EOS see me?

```
[root@ajp ~]# eos whoami
Virtual Identity: uid=0 (0,2,3,99) gid=0 (0,4,99) [authz:sss] sudo* host=localhost domain=localdomain geo-location=ajp
```

roles | protocol | sudo | host, domain, geo location

Who can EOS see?

```
[root@ajp ~]# eos who -a
auth   : sss                       := 1 sessions
user   : root                      := 1 sessions
client : root                      := root@localhost                ( sss) [ localhost                                ] { XRoot    } 0s idle time
```

Virtual Identity Special VIDs

- VID root [0,0,sss] can do everything

```
[root@ajp ~]# eos whoami
Virtual Identity: uid=0 (0,2,3,99) gid=0 (0,4,99) [authz:sss] sudo* host=localhost domain=localdomain geo-location=ajp
[root@ajp ~]# eos rm -rf /
```

- VID adm [3,4] can do chmod,chown,quota,ls,read/write … but not impersonate
- VID daemon[2,2,sss] is used by FSTs to authenticate to MGM

- VID nobody[99,99] is your destiny if you are not mapped

Access Authorization

- after VID mapping the **access functionality** is called, which decides if your VID can currently access this EOS instance

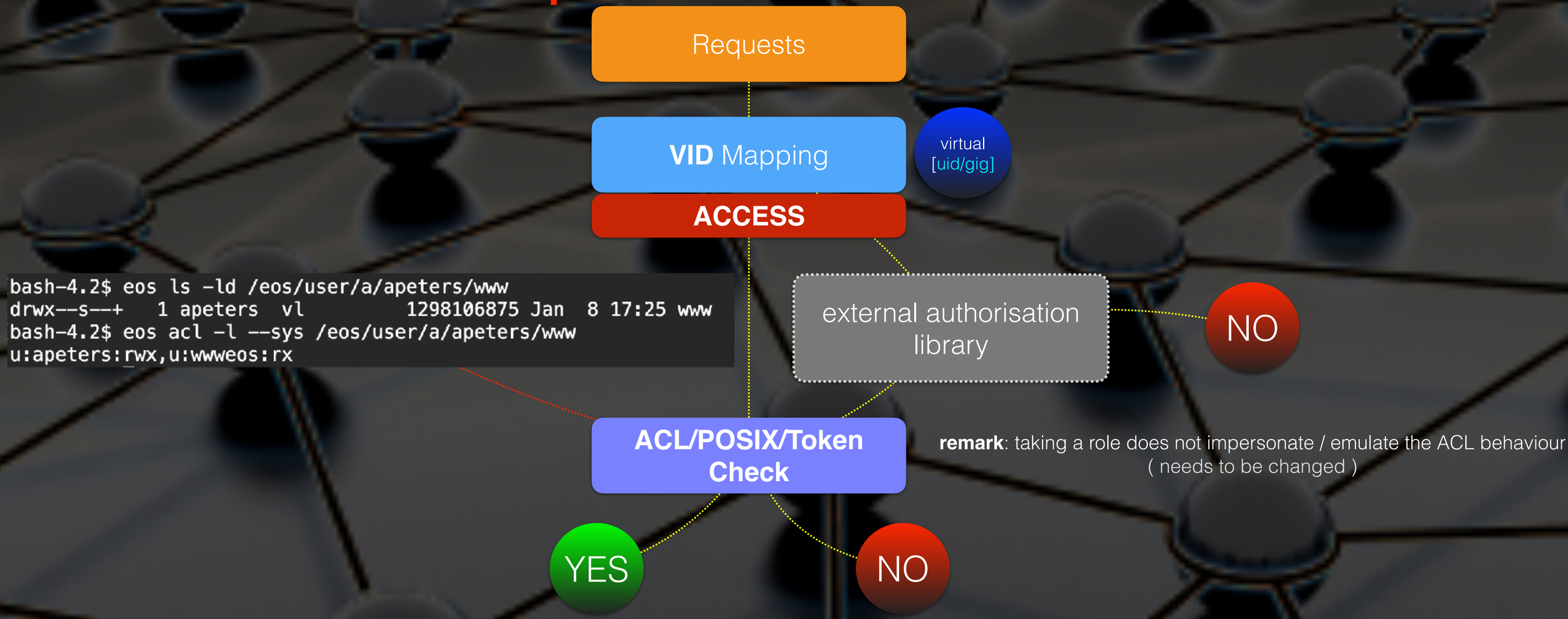
  - access can be expressed as **white or blacklist** (e.g. allow vs ban)
    - by **user, group, host** and **domain** name

  - access can be **stalled** based on configured **rate limits** for the requesting client

larger tails due seeks

t write-back (seq. disk IO)

Namespace Authorization

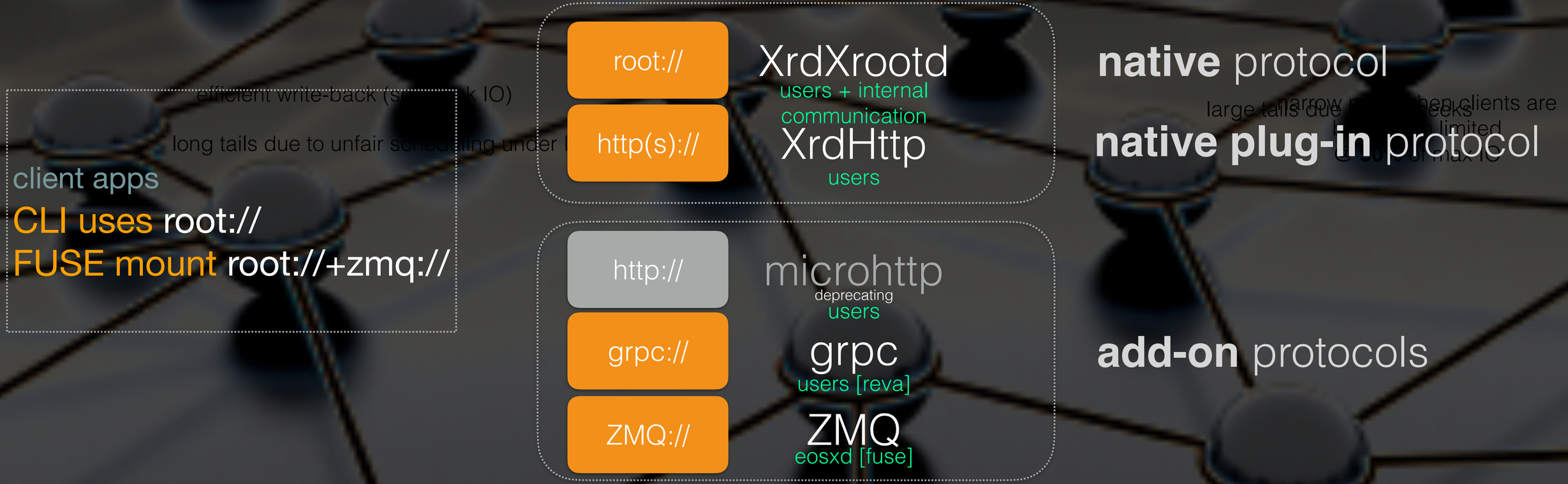



```
bash-4.2$ eos ls -ld /eos/user/a/apeters/www
drwx--s--+   1 apeters  vl         1298106875 Jan  8 17:25 www
bash-4.2$ eos acl -l --sys /eos/user/a/apeters/www
u:apeters:rwx,u:wwweos:rx
```

remark: taking a role does not impersonate / emulate the ACL behaviour (needs to be changed)

https://eos-docs.web.cern.ch/configuration/permission.html

Supported Protocols



client apps

CLI uses root://

FUSE mount root://+zmq://

Quota

- thee types of quota
  user - match by user id
  group - match by group id
  project - match all writers


```
┌─> Quota Node: /eos/dev/test/
```

user	used bytes	logi bytes	used files	aval bytes	aval logib	aval files	filled[%]	vol-status	ino-status
100	0 B	0 B	0	1.00 TB	1.00 TB	1.00 M	0.00 %	ok	ok
2000	0 B	0 B	0	1.00 TB	1.00 TB	1.00 M	0.00 %	ok	ok
adm	0 B	0 B	0	1.00 TB	1.00 TB	1.00 M	0.00 %	ok	ok
daemon	0 B	0 B	0	1.00 TB	1.00 TB	1.00 M	0.00 %	ok	ok
nobody	4.10 KB	4.10 KB	4	100.00 MB	100.00 MB	1.00 M	0.00 %	ok	ok
operator	0 B	0 B	0	1.00 TB	1.00 TB	1.00 M	0.00 %	ok	ok

group	used bytes	logi bytes	used files	aval bytes	aval logib	aval files	filled[%]	vol-status	ino-status
nobody	0 B	0 B	0	0 B	0 B	0	100.00 %	ignored	ignored

summary	used bytes	logi bytes	used files	aval bytes	aval logib	aval files	filled[%]	vol-status	ino-status
All users	4.10 KB	4.10 KB	4	5.00 TB	5.00 TB	6.00 M	0.00 %	ok	ok
All groups	0 B	0 B	0	0 B	0 B	0	100.00 %	ignored	ignored

- two optional quota settings
  - inode - How many files?
  - volume - How much data?

- quota is configured on quota nodes (directory + subtree) e.g /eos/foo/

- quota nodes are **not nested**
  - if /eos/foo/ and /eos/ is a quota node, /eos/foo/bar concerns only the deepest matching node /eos/foo/

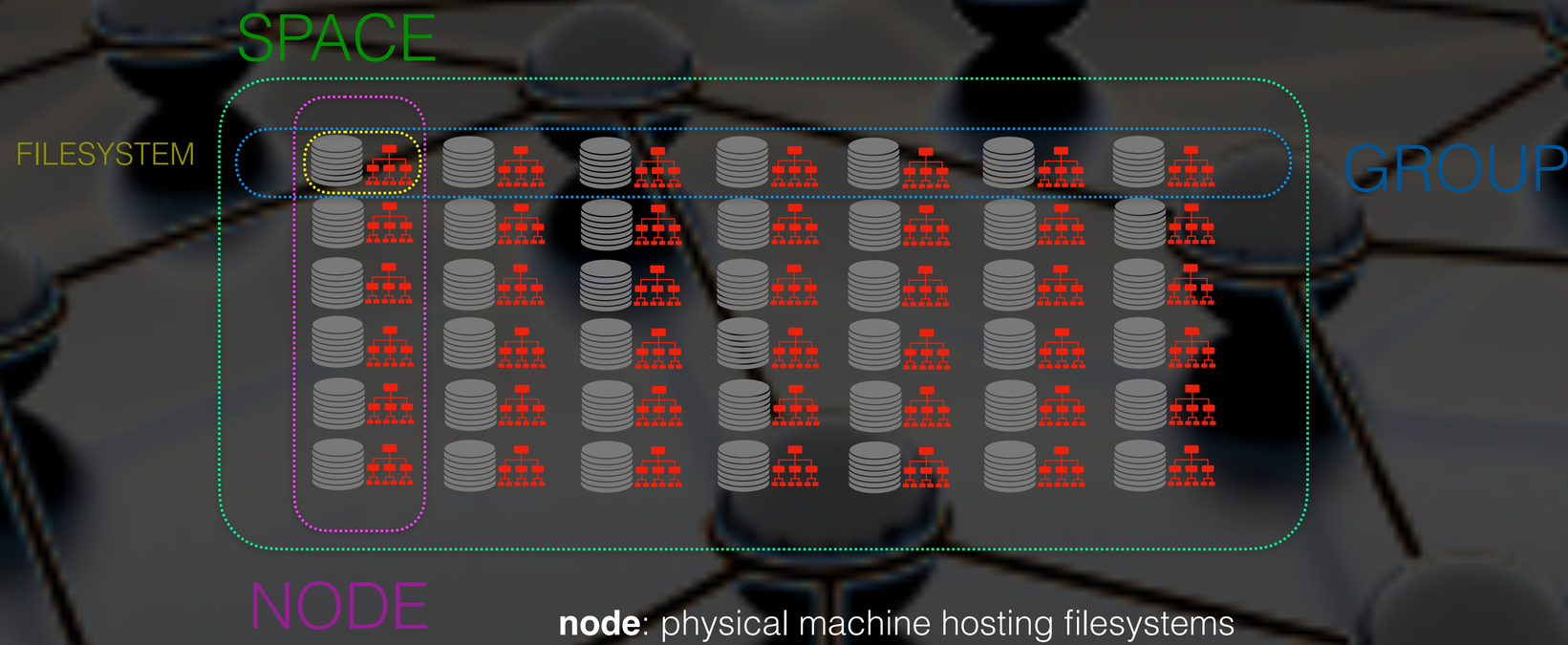
- quota can be en-/disabled for each SPACE (what is that?)

- quota defines usage of physical not logical space (it includes the volume including redundancy e.g. replication of files)

https://eos-docs.web.cern.ch/configuration/quota.html

EOS workshop 2021
1st to 4th of march @ VIRTUAL

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

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 Open Storage
About
Tech
Resources
Workshop
Service
News
Support
Git
Community
EVERYTHING YOU NEED
TO KNOW ABOUT
EOS Open Storage
WORKSHOP '21
LATEST V4.8.35
Install
Virtual Workshop 1.-5. March 2021
eos.web.cern.ch