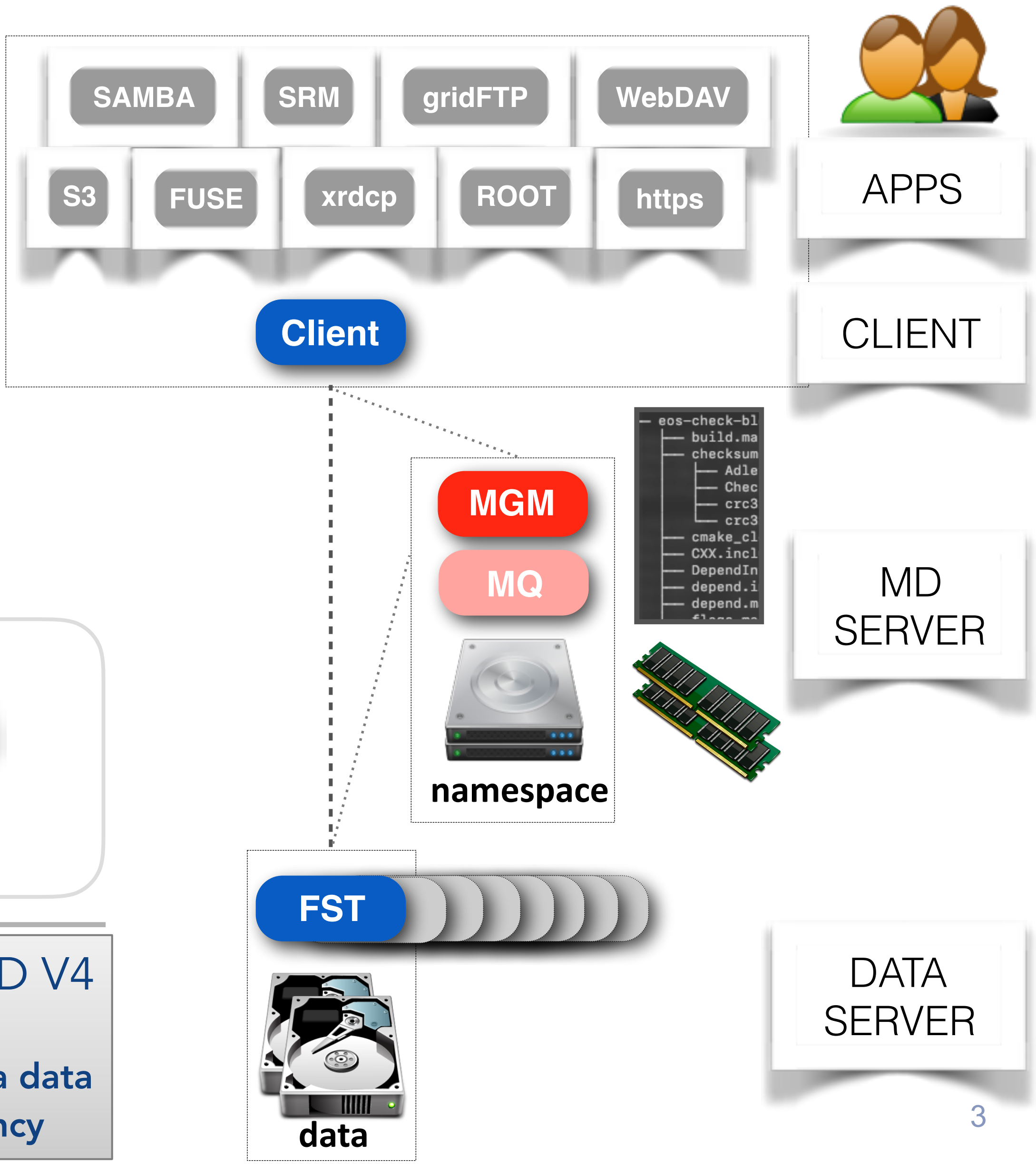
CERN

EOS and CERNBox:

deployment of the new namespace and the new eosuser architecture

Luca Mascetti
CERN IT Storage

EOS Architecture



```
─ eos-check-bl
  ├── build.ma
  ├── checksum
  │   ├── Adle
  │   ├── Chec
  │   ├── crc3
  │   └── crc3
  ├── cmake_cl
  ├── CXX.incl
  ├── DependIn
  ├── depend.i
  ├── depend.m
```

EOS Production Releases

Aquamarine

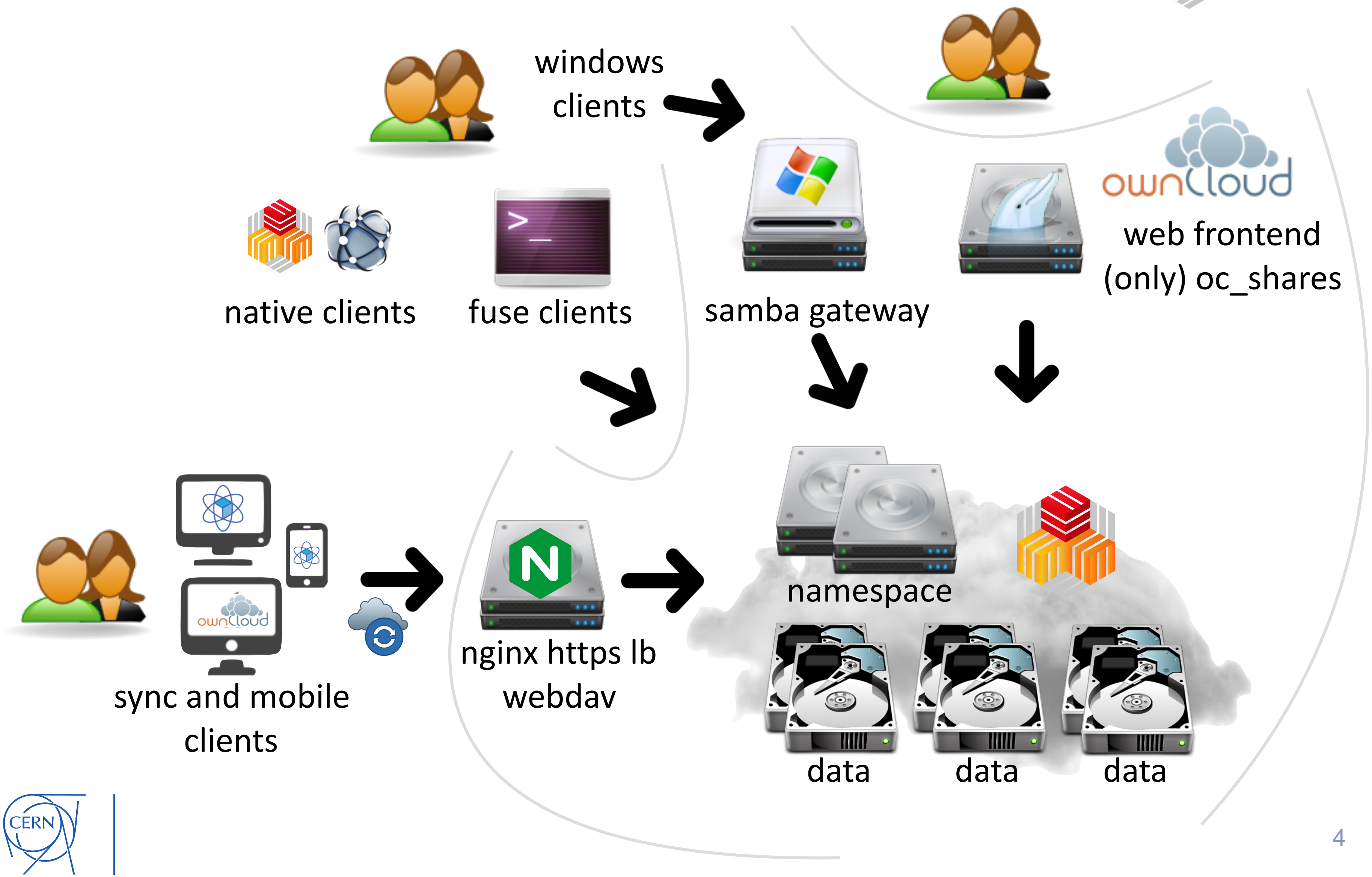
V 0.3.X

Citrine

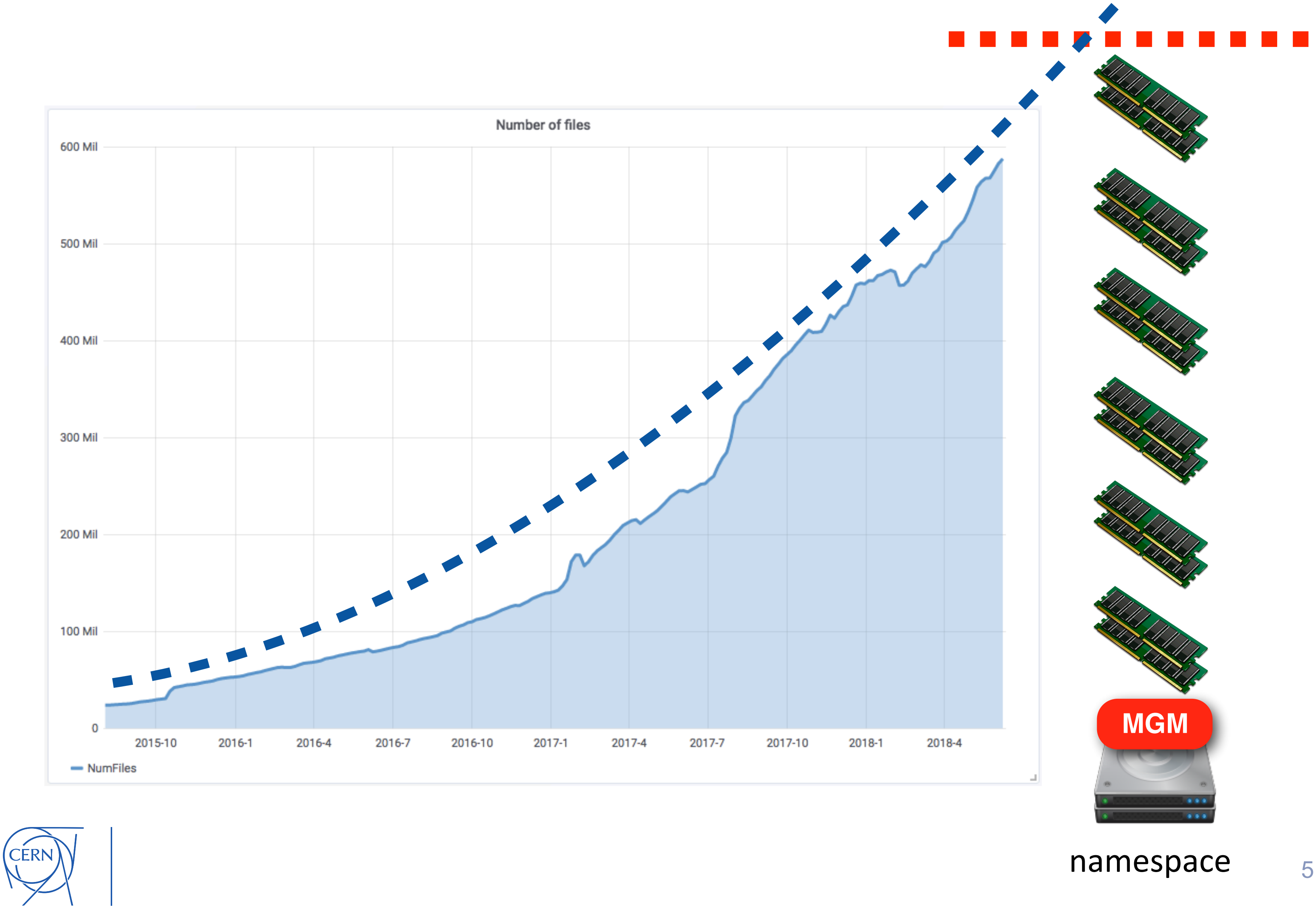
V 4.X

XRootD V3	XRootD V4
IPV4	**IPV6**
namespace in-memory	**plugins for meta data**
data on attached disks	**& data persistency**

EOSUSER a.k.a. CERNBox

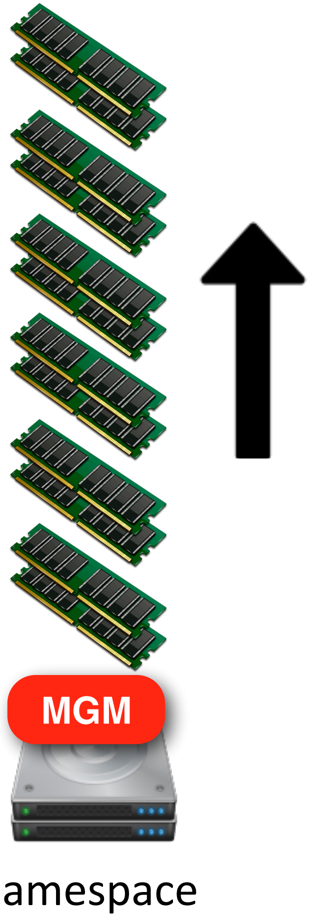


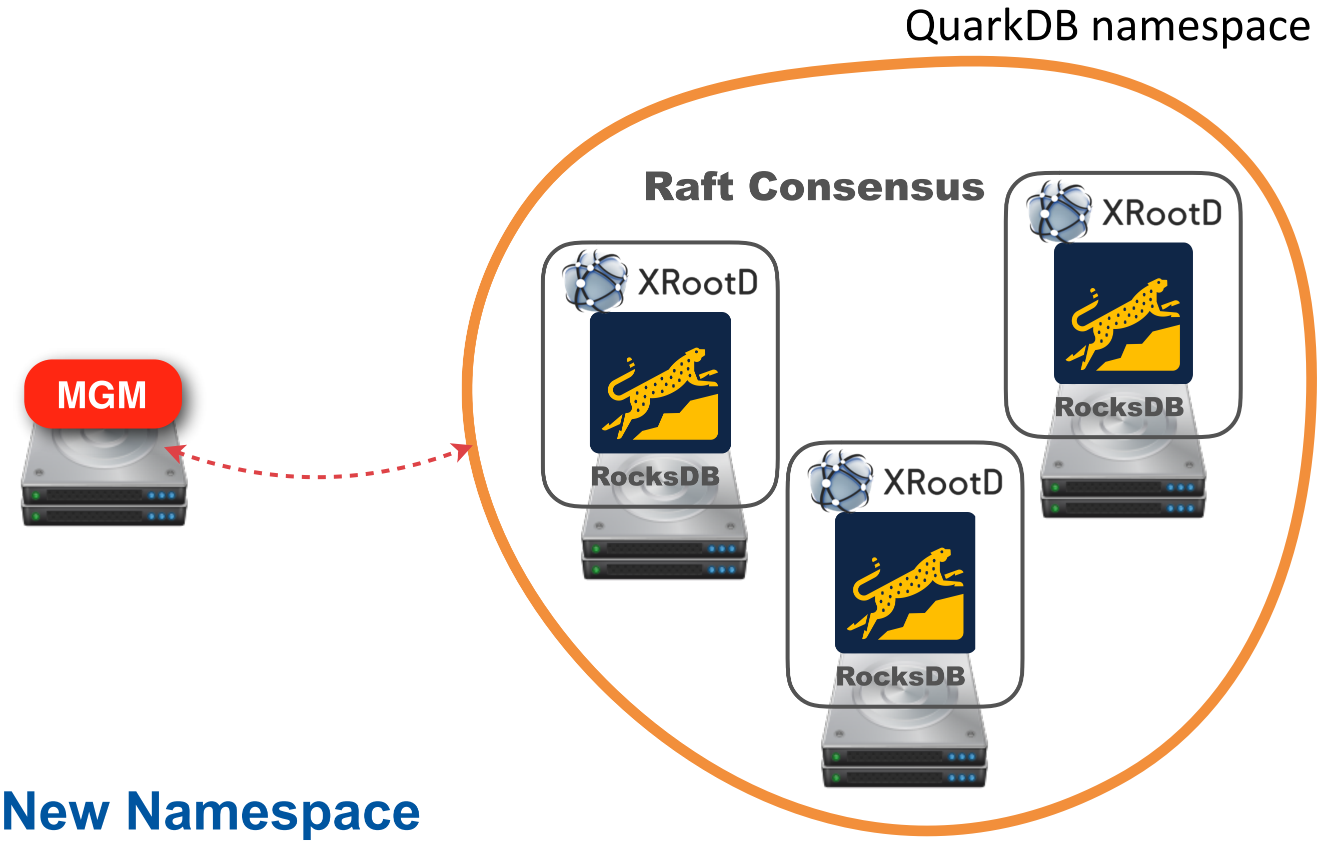
EOS Namespace Challenge



EOS Namespace Challenge

Shifting namespace paradigm: from scale-up to scale-out





New Namespace

- KV store resident on disk
- **very short restart time!!**
  - not based on #files and #dirs
- namespace caching in MGM memory

Solution 1: eosuser upgrade

Upgrade current production

Two steps upgrade:

1. upgrade from aquamarine to citrine
2. namespace conversion

From past experiences:

- very very very long downtime => just not acceptable
- instabilities in booting filesystems with millions of files

Solution 1: eosuser upgrade

Upgrade current production

Two steps upgrade:

1. upgrade from aquamarine to citrine
2. namespace conve[illegible]

From past experiences:

- very very very long downtime => j[illegible]ptable
- instabilities in booting filesystems with [illegible] files

QUALITY CONTROL • REJECTED •

Solution 2: eosuser2

Deploy a new empty instance with latest namespace technology

New deployment and migration:

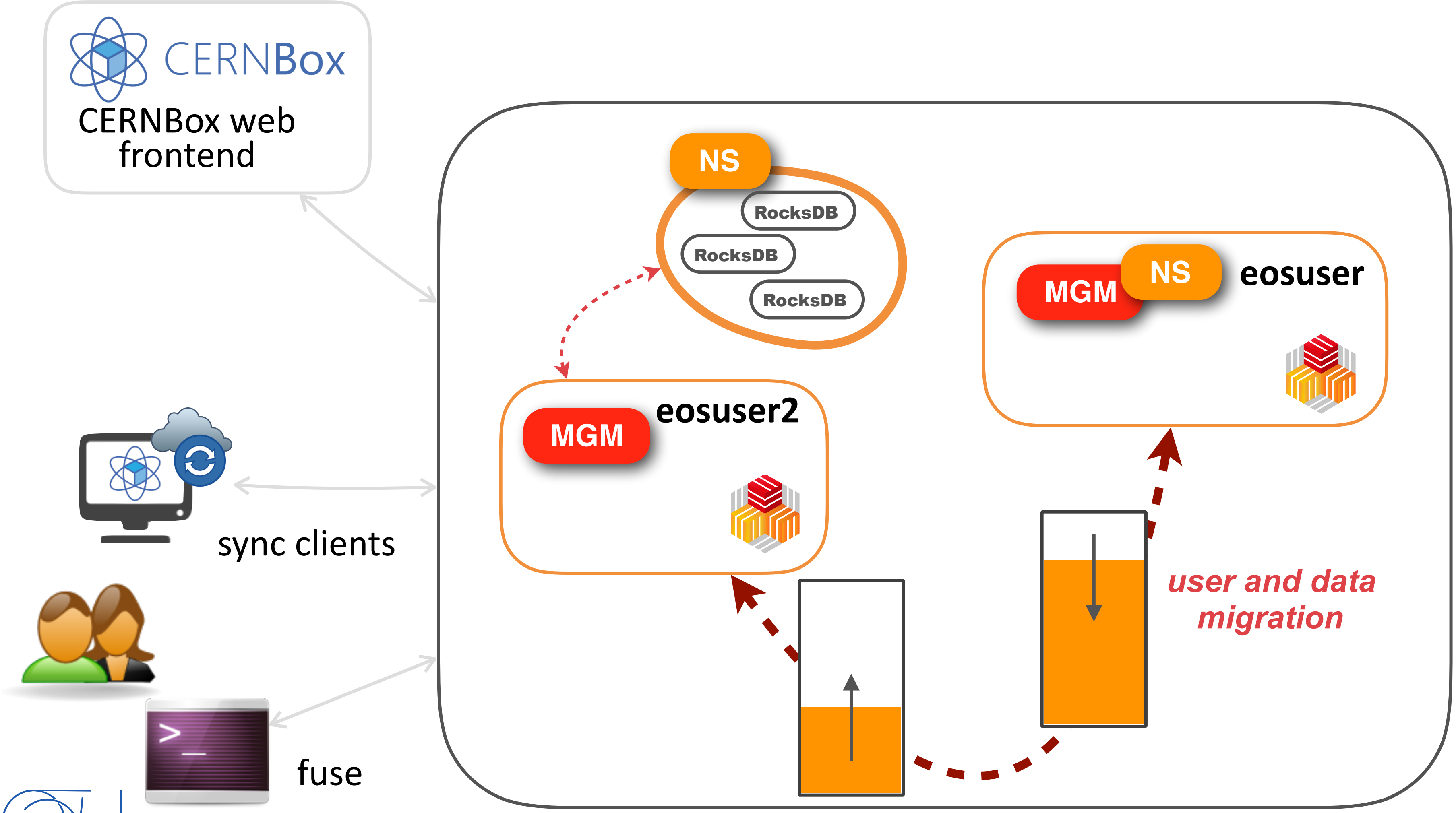
1. build a **new empty EOS instance**
    1. start immediately with QDB namespace
2. migrate gradually users

From past experiences:

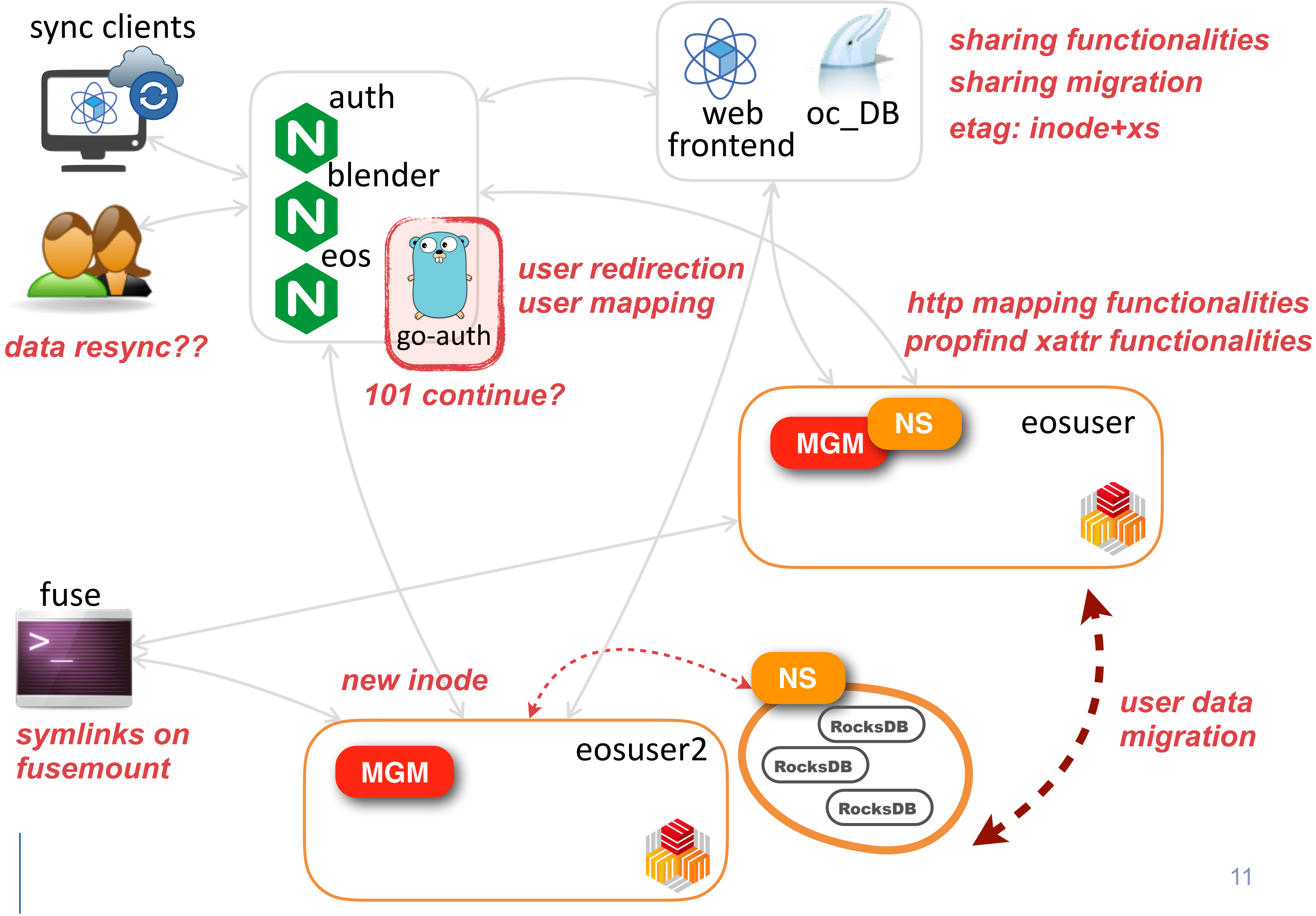
- migration needs to be transparent!!!
- no BIG-BANG! approach
- better load control over time
- better operations "feeling"

Solution 2: eosuser2

Deploy a new empty instance with latest namespace technology



FYI: Behind the scenes



sharing functionalities
sharing migration
etag: inode+xs

user redirection
user mapping

http mapping functionalities
propfind xattr functionalities

data resync??

101 continue?

new inode

symlinks on fusemount

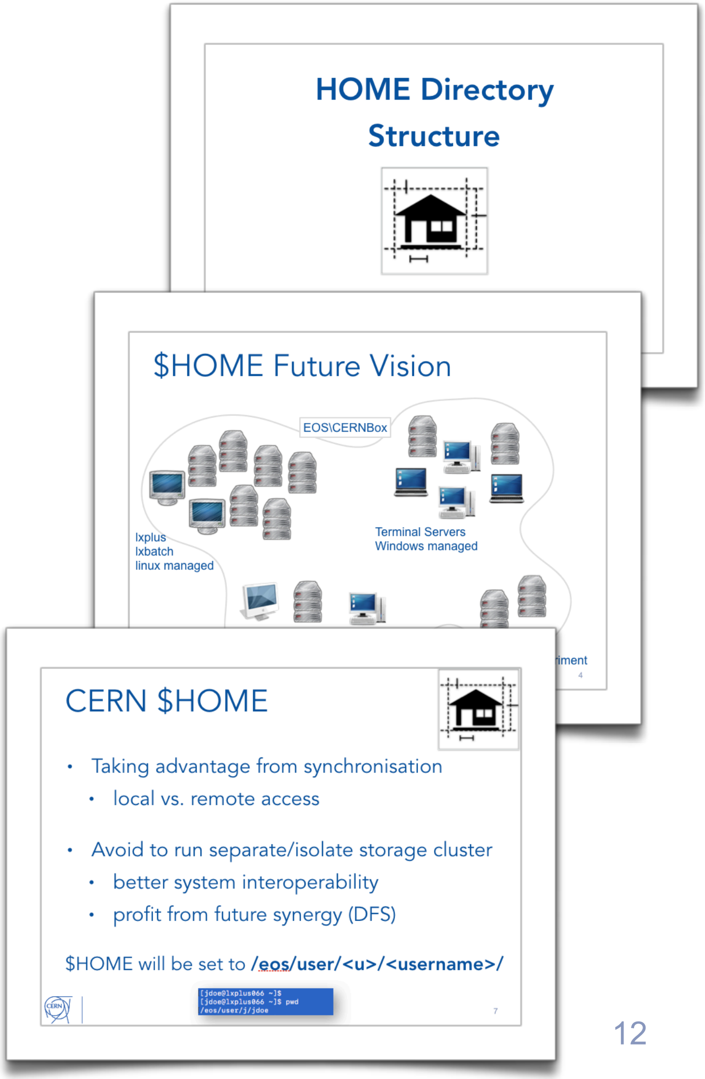
user data migration

Solution 2: eosuser2

Deploy a new empty instance with latest namespace technology

Some additional considerations:

- single instance for all users
- MGMs single point of failures
- Scale metadata performance
- difficult users isolation
- future big upgrade => big bang?
- “plane crash” effect
- Shared Desktop across devices
  - CERN $HOME future plans
  - DFS and linux home discussions



HOME Directory Structure

$HOME Future Vision

CERN $HOME

- Taking advantage from synchronisation
  - local vs. remote access
- Avoid to run separate/isolate storage cluster
  - better system interoperability
  - profit from future synergy (DFS)

$HOME will be set to **/eos/user/<u>/<username>/**

```
[jdoe@lxplus066 ~]$
[jdoe@lxplus066 ~]$ pwd
/eos/user/j/jdoe
```

Solution 2: eosuser2

Deploy a new empty instance with latest namespace technology

Some additional considerations:

- single instance for all users
- MGMs single point of failures
- Scale metadata performance
- difficult users isolation
- future big upgrade => big bang
- "plane crash" effect
- Shared Desktop across devices
  - CERN $HOME future plans
  - DFS and linux home discussions

QUALITY CONTROL • REJECTED •

HOME Di tory

Struc

HOME Future Vision

CERN $HOME

- Taking advantage from synchroni tion
  - local vs. remote access
- Avoid to r arate/isolate storage c ter
  - better system interoperability
  - profit from future synergy (S)

$HOME w set /eo er/ u>/<username>/

```
[jdoe@lxpl ]$
[jdoe@lxpl ~]$ pw
/eos/user
```

7

Solution 3: ...

Divide et Impera

Solution 3: eoshome (running out of names...)

Deploy multiple empty instances with latest namespace technology

Architectural review, new deployment and migration:

1. build a **multiple empty EOS instance**
    1. start immediately with QDB namespace
2. add a level of indirection
3. support system expansion and reduction
4. migrate gradually users

From past experiences:

- migrations need to be transparent!!!
- no BIG-BANG! approach
    - gradual (future) software roll-out
- better load control over time
- better operations “feeling”
- better user isolation
- less load/stress per instance

Solution 3: eoshome (running out of names…)

Deploy multiple empty instances with latest namespace technology

Architectural review, new deployment and migration:

1. build a **multiple empty EOS instance**
   1. start immediately with QDB namespace
2. add a level of indirection
3. support system expansion and reduction
4. migrate gradually users

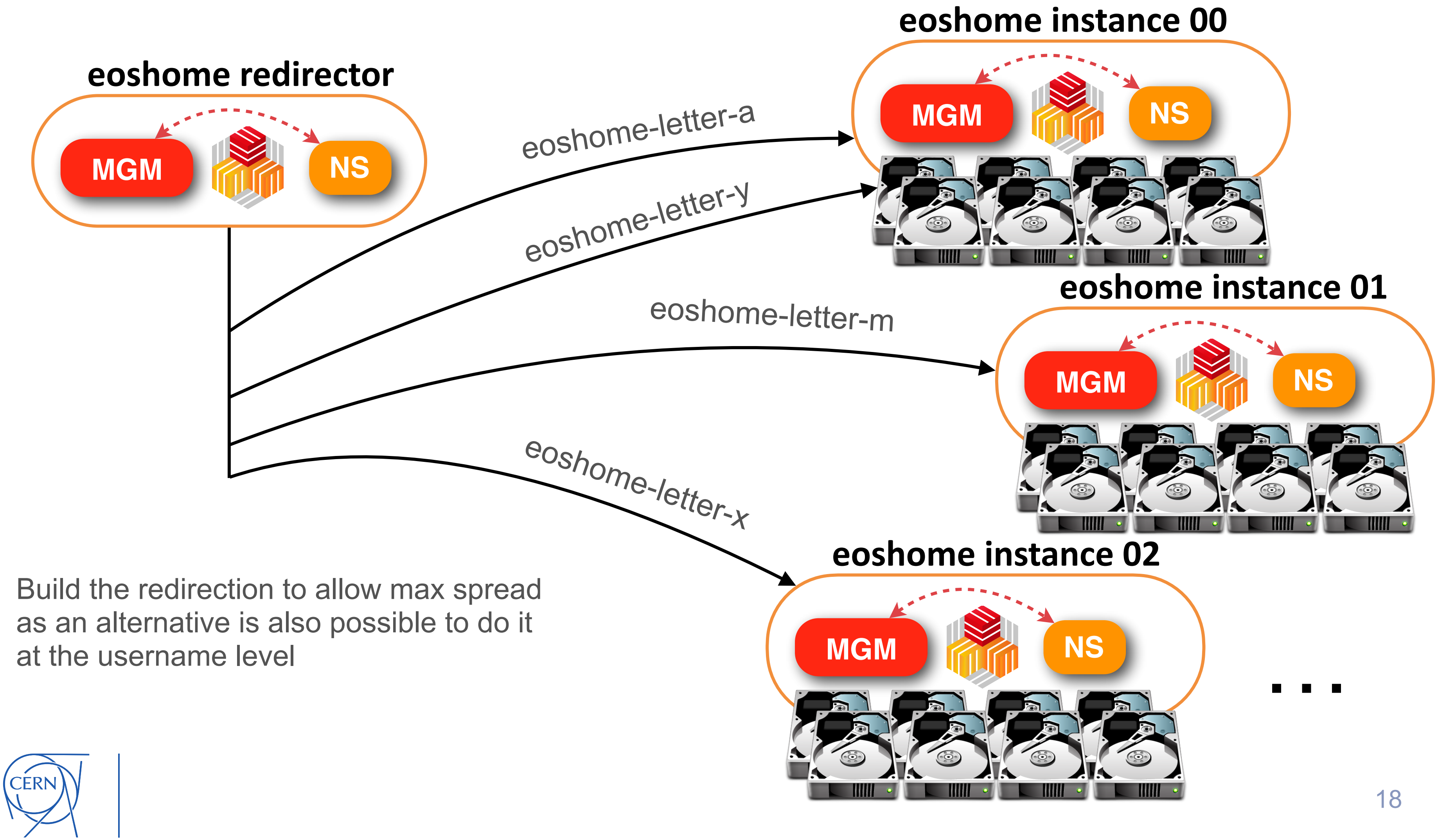
From past experiences:

- migrations need to be transparent!!!
- no BIG BANG! approach
  - gradual (future) software roll-out
- better load control over time
- better operations “feeling”
- better user isolation
- less load/stress per instance

QUALITY CONTROL APPROVED

Solution 3: eoshome

Deploy multiple empty instances with latest namespace technology

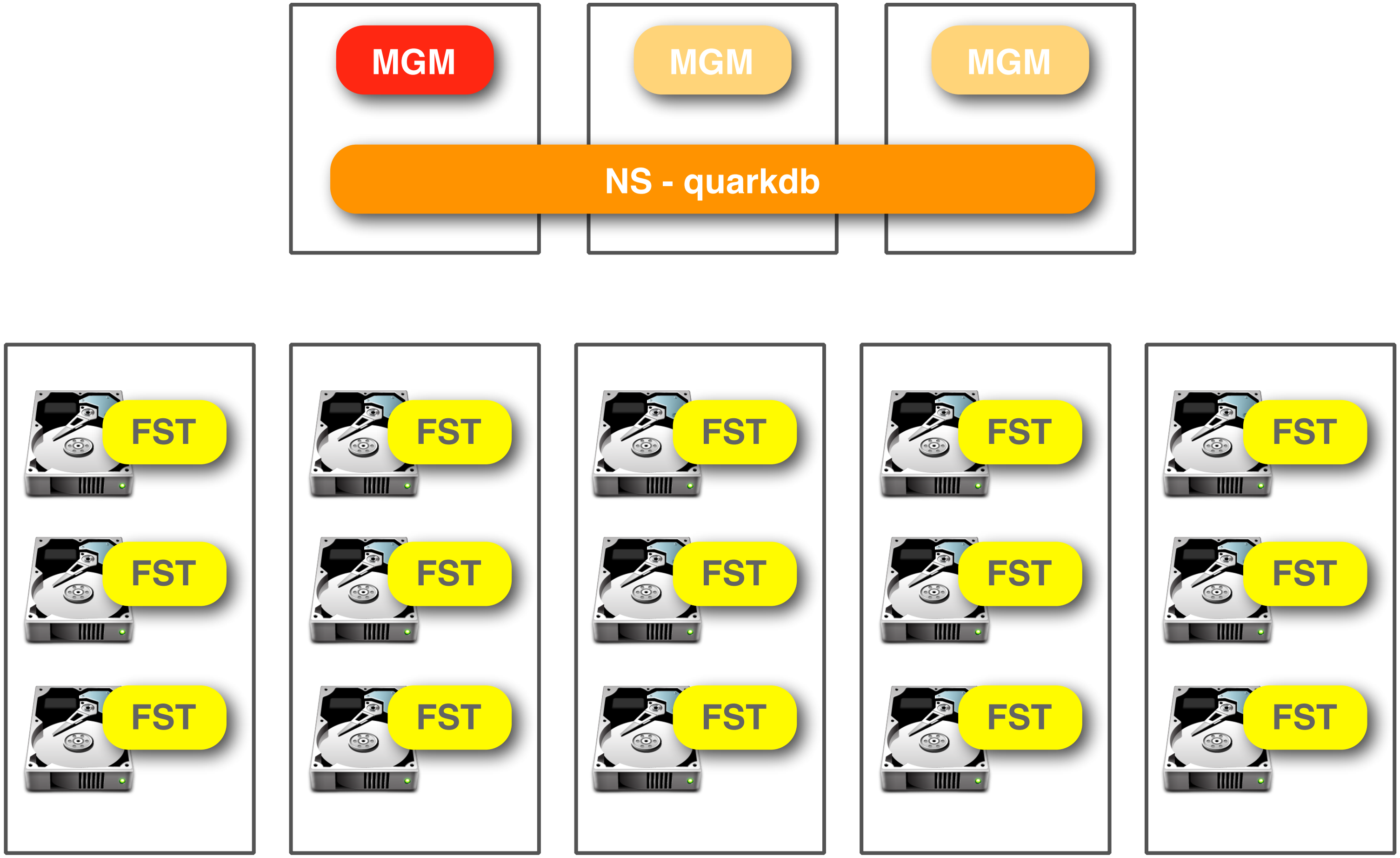


Build the redirection to allow max spread as an alternative is also possible to do it at the username level

Solution 3: eoshome

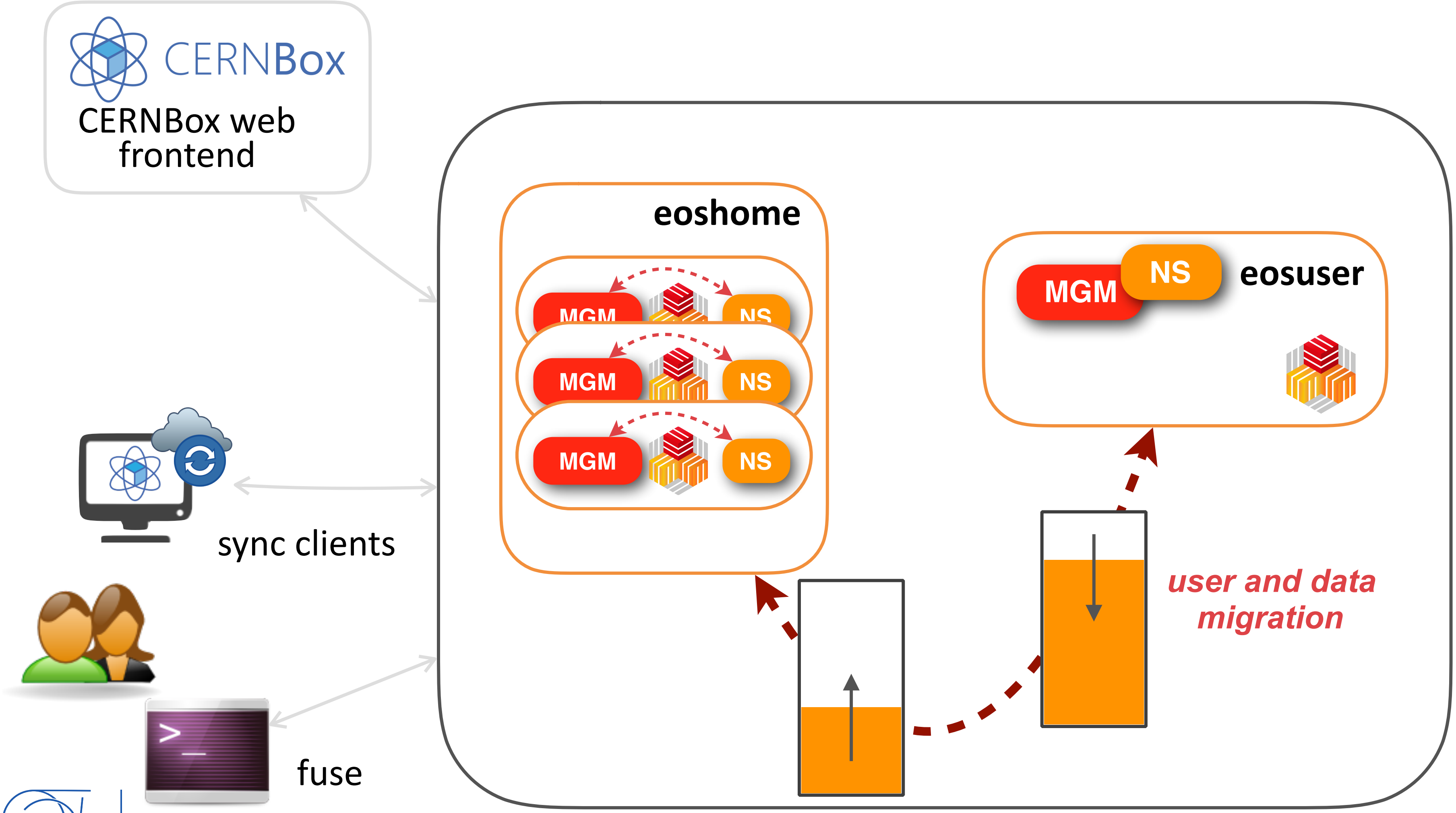
Deploy multiple empty instances with latest namespace technology

eoshome instance XY



Solution 3: eoshome

Deploy multiple empty instances with latest namespace technology

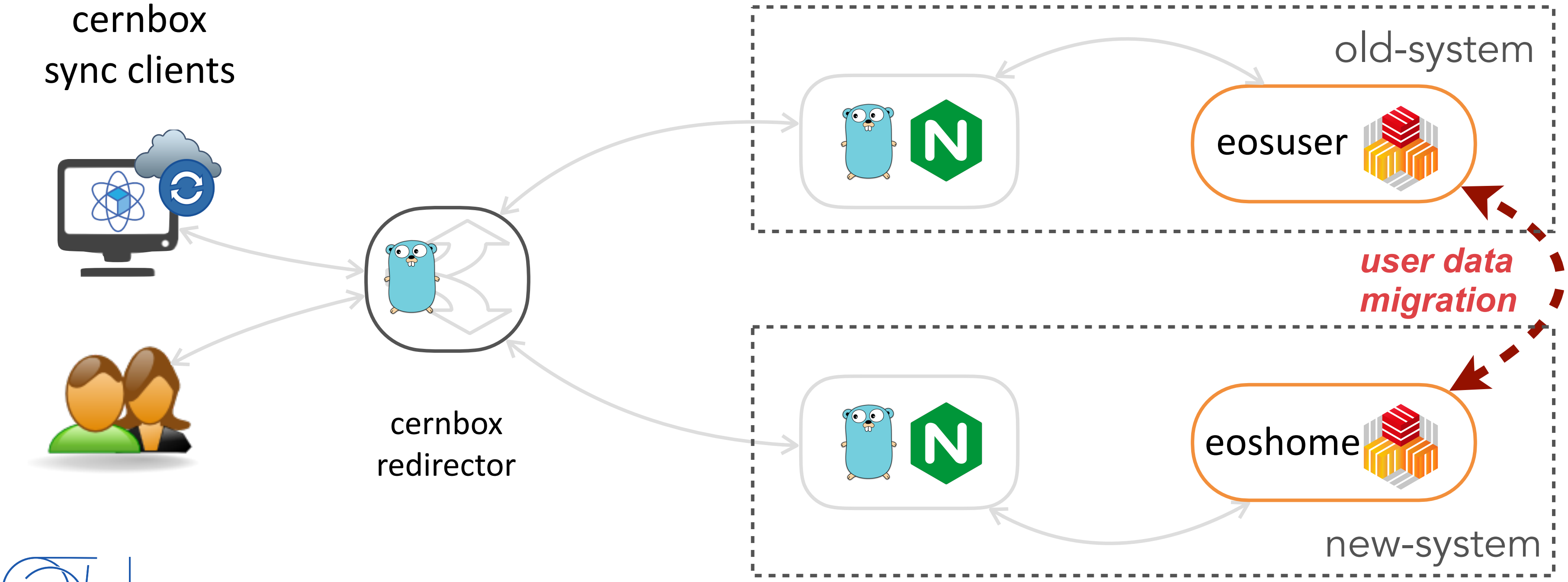


Solution 3: eoshome

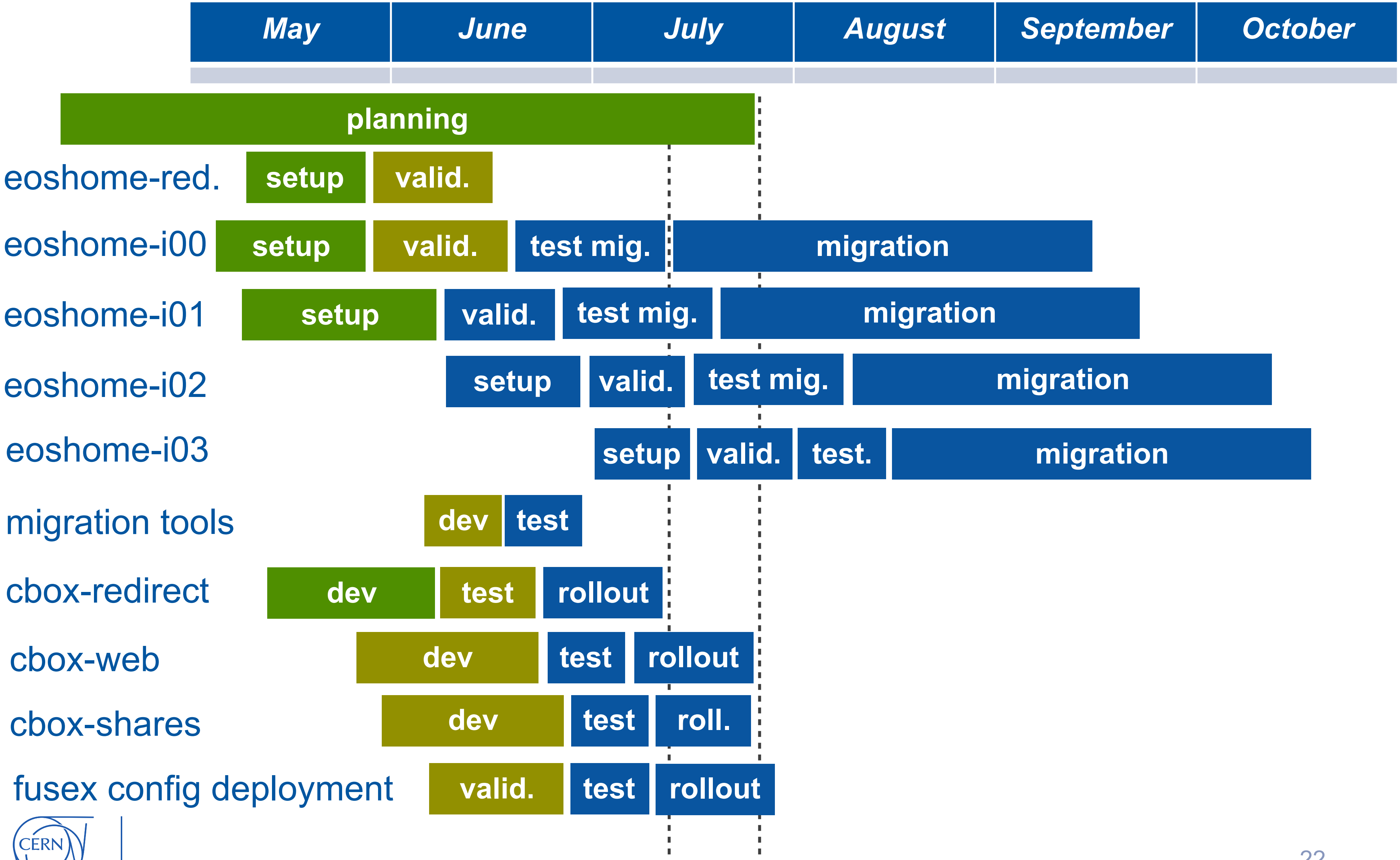
Deploy multiple empty instances with latest namespace technology

Migration scenario similar to **Solution 2**

- same requirements on the CERNBox side
- same requirements on the migration tools



Current Status and Roadmap



May | June | July | August | September | October

planning

eoshome-red. setup valid.

eoshome-i00 setup valid. test mig. migration

eoshome-i01 setup valid. test mig. migration

eoshome-i02 setup valid. test mig. migration

eoshome-i03 setup valid. test. migration

migration tools dev test

cbox-redirect dev test rollout

cbox-web dev test rollout

cbox-shares dev test roll.

fusex config deployment valid. test rollout

Summary and Outlook

Lots of hard work ahead!

General improvement of **EOS\CERNBox** architecture

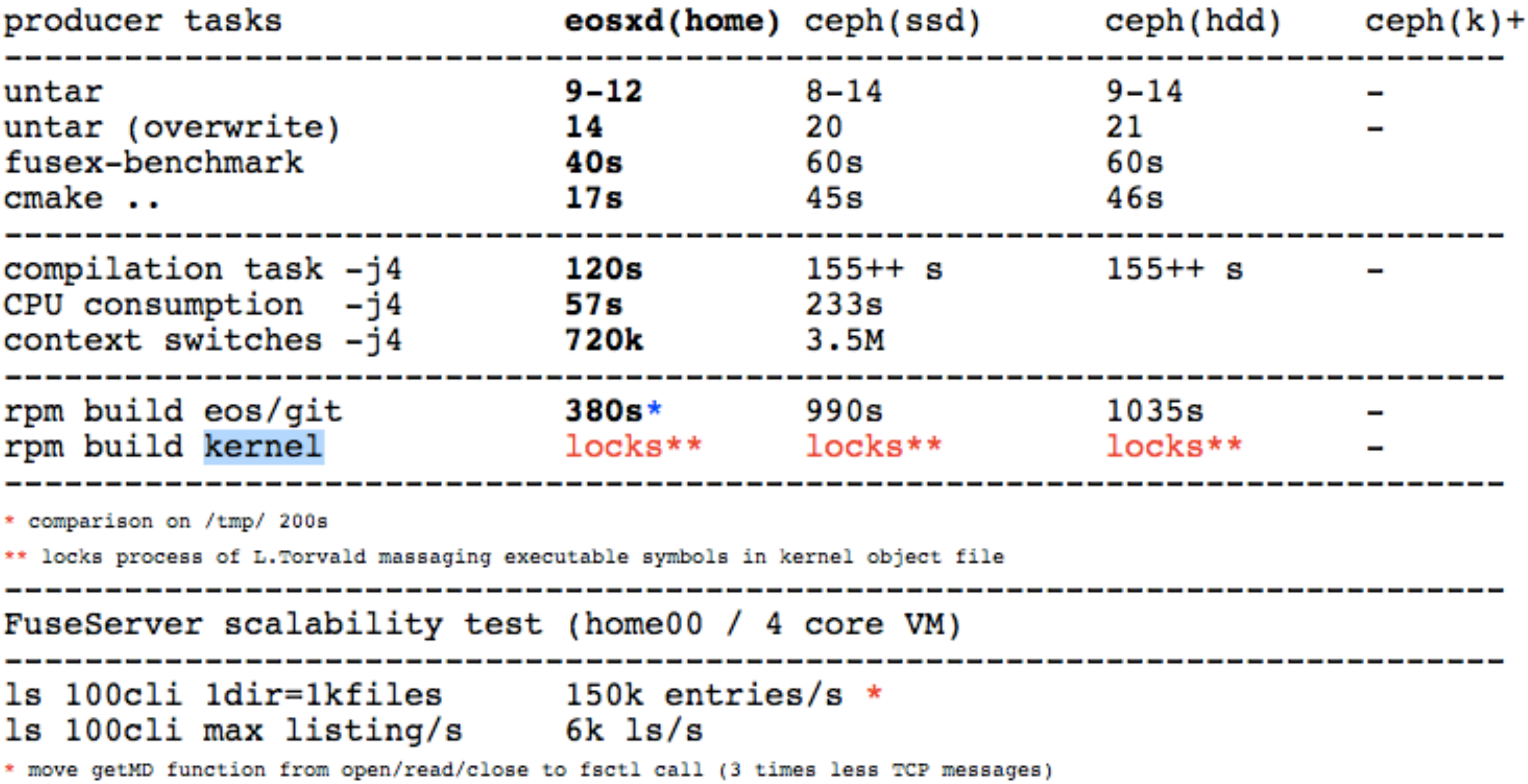
- removing SPOFs
- improving metadata performance
- reducing drastically downtimes
  - less user impacted
  - almost zero restart time
- flexibility to scale up and out at the same time
- removing big-bang upgrades
  - simplify small scale testing and software rollout

Thanks for the attention!

www.cern.ch

Questions?

Andreas J.P. EOSXD Benchmark



producer tasks	**eosxd(home)**	ceph(ssd)	ceph(hdd)	ceph(k)+
untar	**9-12**	8-14	9-14	-
untar (overwrite)	**14**	20	21	-
fusex-benchmark	**40s**	60s	60s	
cmake ..	**17s**	45s	46s	
compilation task -j4	**120s**	155++ s	155++ s	-
CPU consumption -j4	**57s**	233s		
context switches -j4	**720k**	3.5M		
rpm build eos/git	**380s***	990s	1035s	-
rpm build kernel	locks**	locks**	locks**	-

* comparison on /tmp/ 200s

** locks process of L.Torvald massaging executable symbols in kernel object file

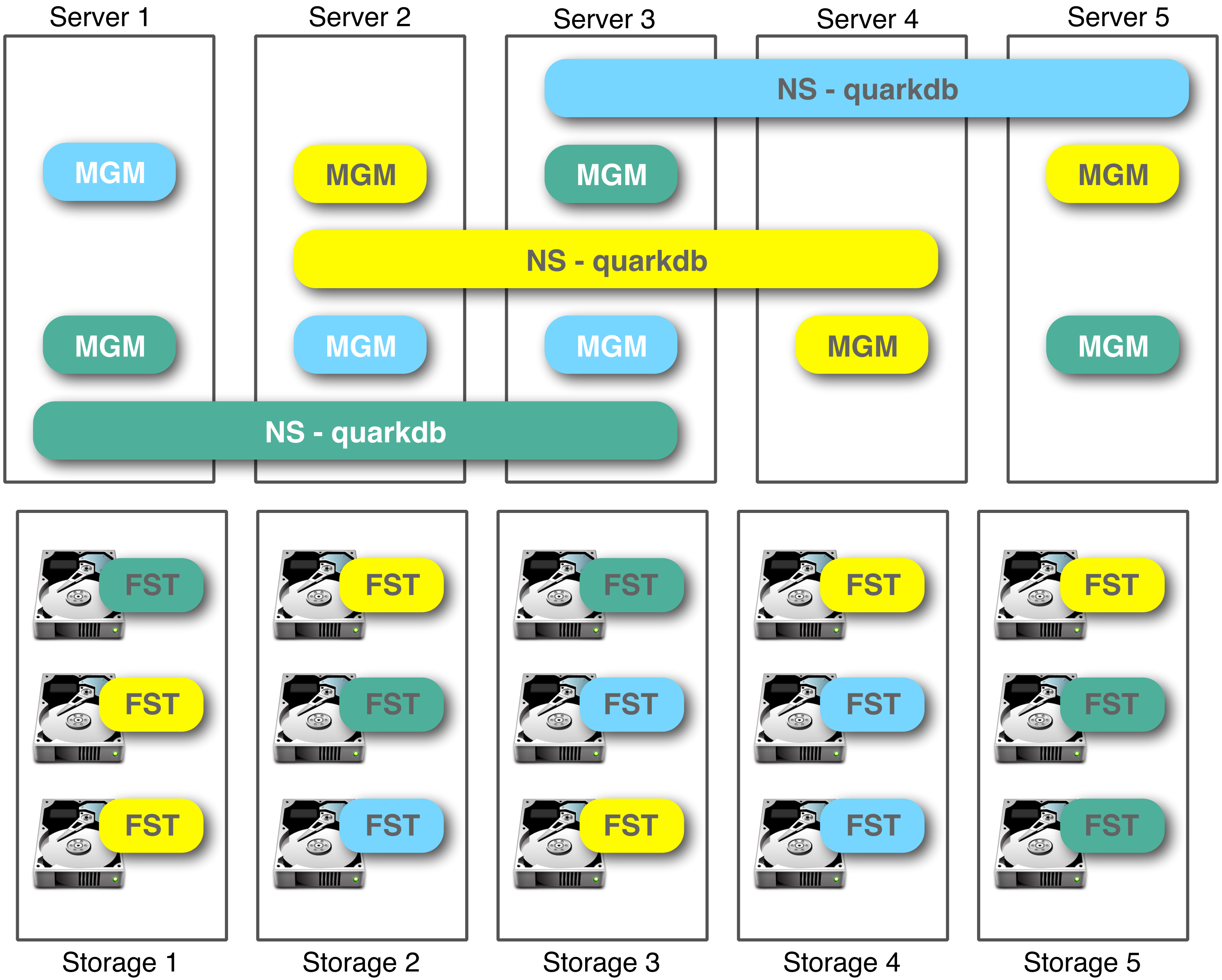
FuseServer scalability test (home00 / 4 core VM)

ls 100cli 1dir=1kfiles	150k entries/s *
ls 100cli max listing/s	6k ls/s

* move getMD function from open/read/close to fsctl call (3 times less TCP messages)

stable benchmarks on 4 core VM towards EOS-MGM CEPH-MDS @0.3ms RTT (default mount) on idle instances (repeated many times on several days/instances)

mixed eoshome instances over hw


Server 1
Server 2
Server 3
Server 4
Server 5
NS - quarkdb
MGM
MGM
MGM
MGM
NS - quarkdb
MGM
MGM
MGM
MGM
MGM
NS - quarkdb
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
FST
Storage 1
Storage 2
Storage 3
Storage 4
Storage 5