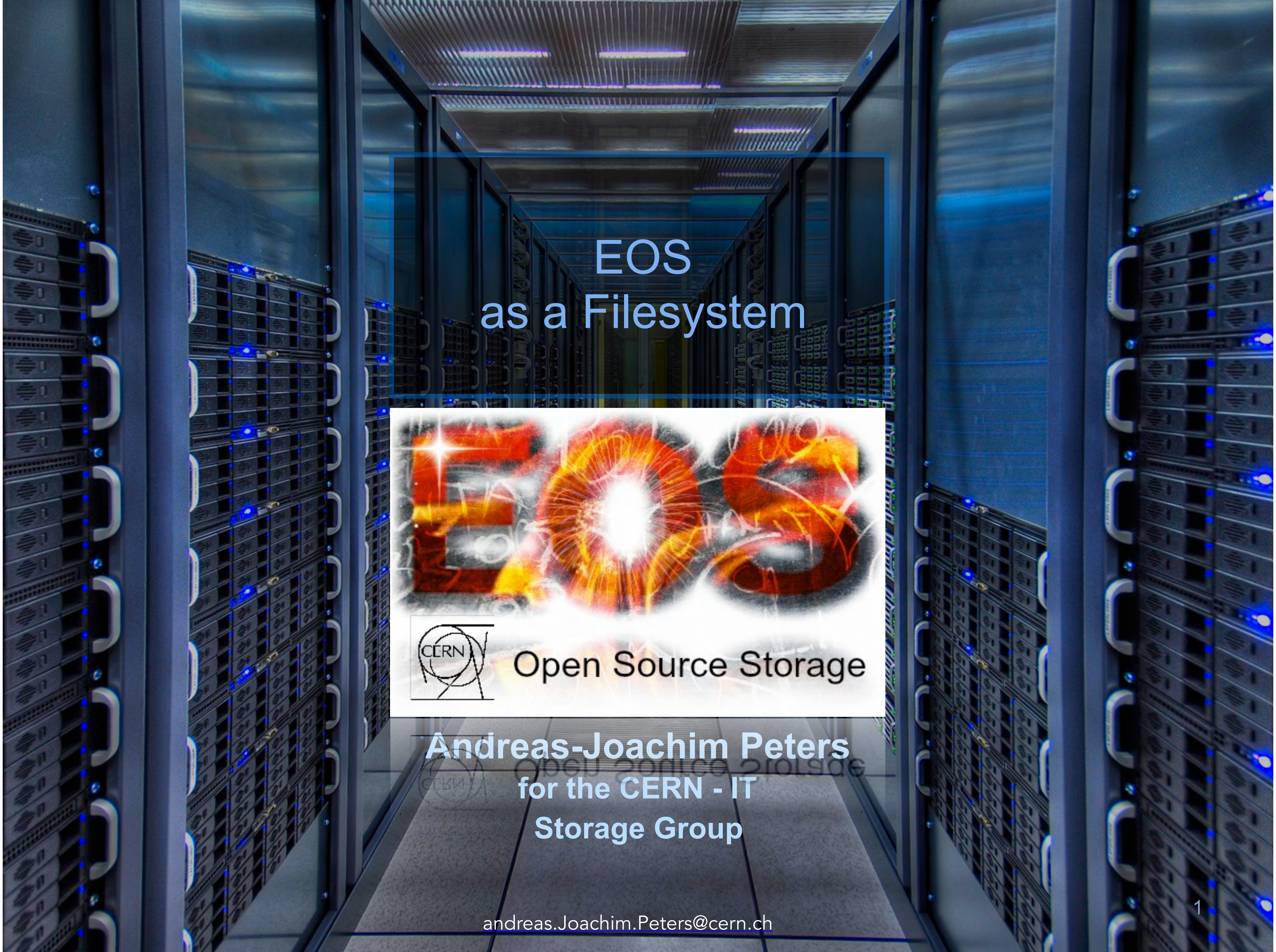


EOS
as a Filesystem

EOS

CERN Open Source Storage

Andreas-Joachim Peters
for the CERN - IT
Storage Group

andreas.Joachim.Peters@cern.ch

We want filesystems.

What we are used to ….

… or how to use EOS as a filesystem

History of FUSE in EOS

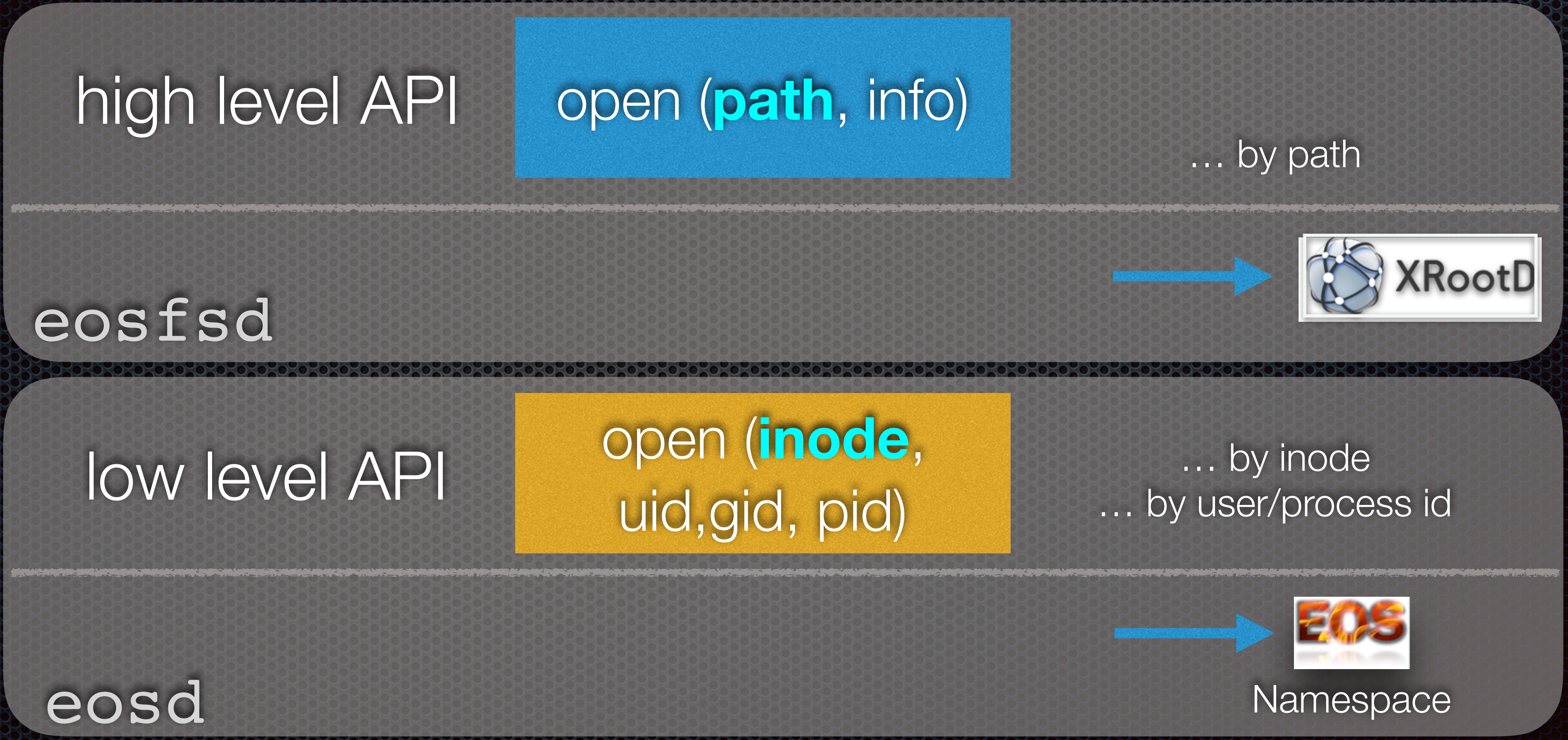
EOS FUSE - history

- two FUSE clients were available since four years
  - **eosfsd** - individual single user mount (krb5/gsi) - high-level API
    - used on lxplus (atlas, cms, public users…)
  - **eosd** - shared multi-user mount (trusted/sss) - low-level API
- implementation was adoption of old *xrootdfs* FUSE implementation
  - using high-level API in C
  - *Summary*: it worked only for simple POSIX use cases, modest performance - not high priority for active development in EOS in the past

FUSE APIs

FUSE

high vs low level API



high level API

open (**path**, info)

… by path

`eosfsd`

XRootD

low level API

open (**inode**, uid,gid, pid)

… by inode
… by user/process id

`eosd`

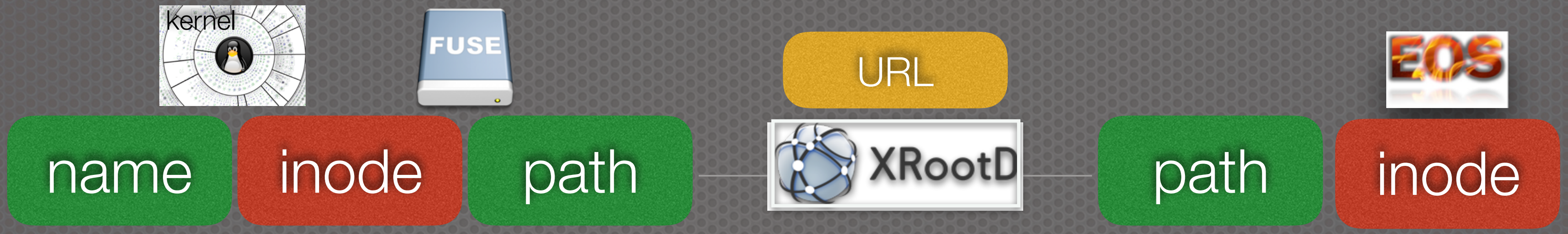
EOS

Namespace

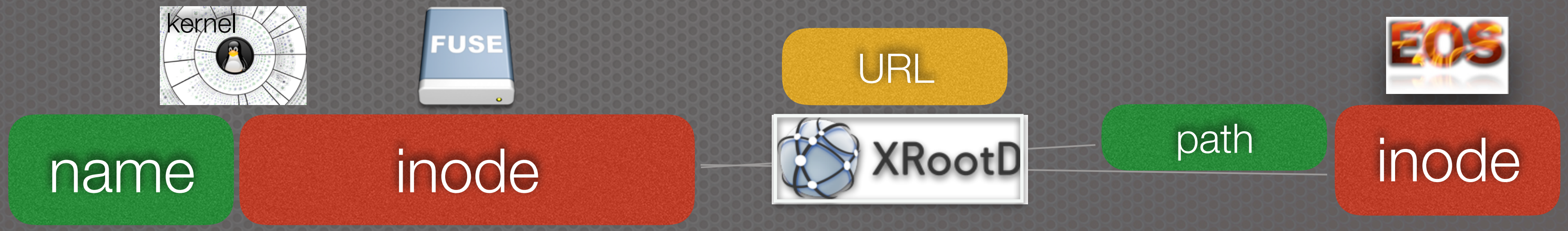
FUSE

high vs low level API

high level API



low level API



Difficulty: be consistent in inode/path translation

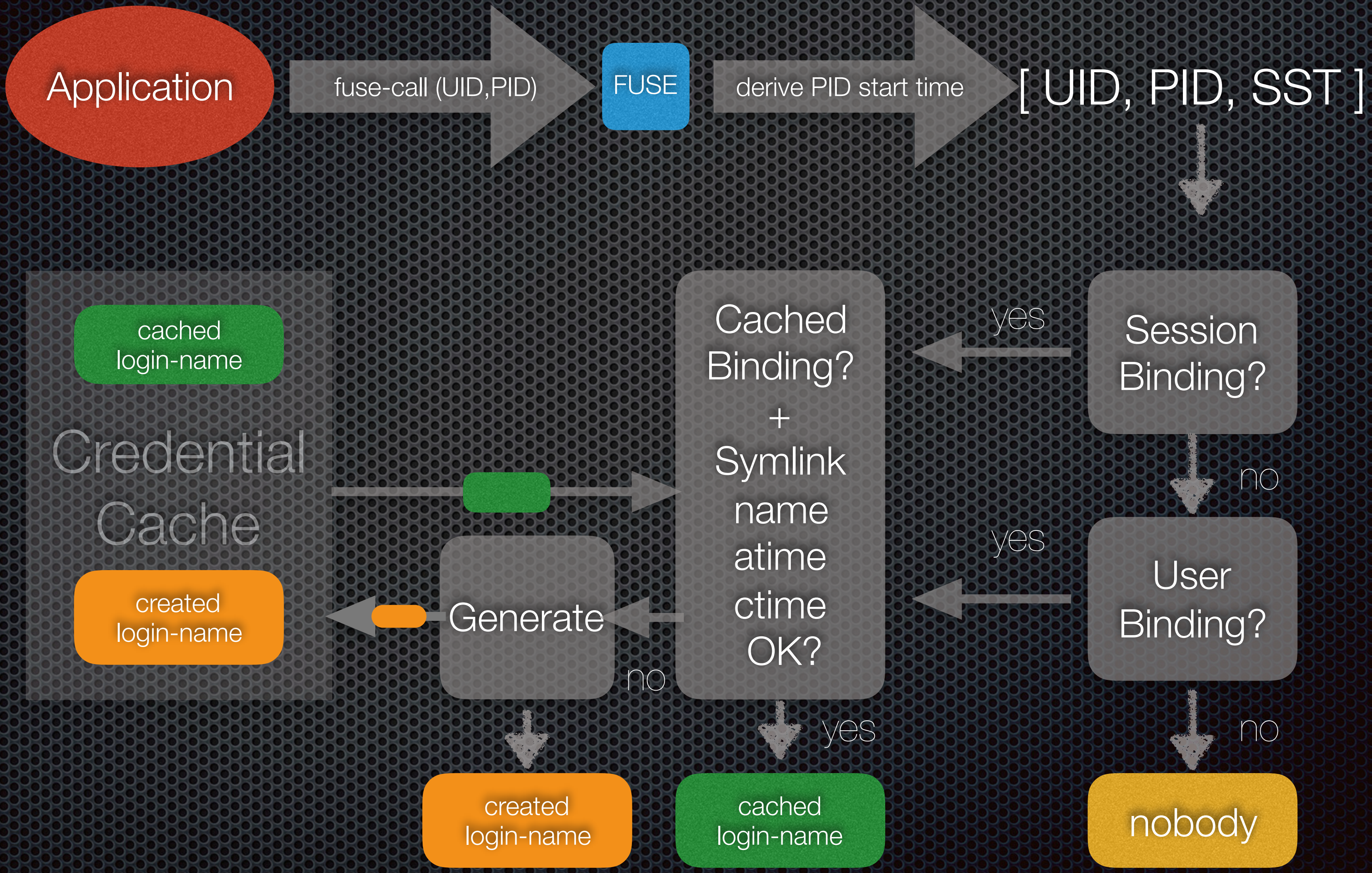
Challenges in FUSE

A first challenge ...

- we need a **security mechanism** as in AFS via FUSE
  - see next talk!

FUSE Client Mapping Logic

uid/pid → login-name



The second challenge ...

- a filesystem does a *hell* of *meta* data operations
  - e.g. a compilation of XRootD does 1.2 M system calls, 440k open, 140k stat, 145k read, 70k write calls
    (A) with a remote FS:
    800s IO time (assuming 1ms latency)
    (B) with the local FS:
    0.5s IO time

- users expect the performance of (B)

- users expect it never fails and does everything like and even better than a local disk

In the beginning there was darkness ...

Where we started ...

... you could not compile anything

... then it took 18min to compile the XRootd example vs. 4min with a local disk

... in parallel compilation it took 22s with a local disk vs. 9 min with EOS FUSE ...

Refactoring EOS FUSE

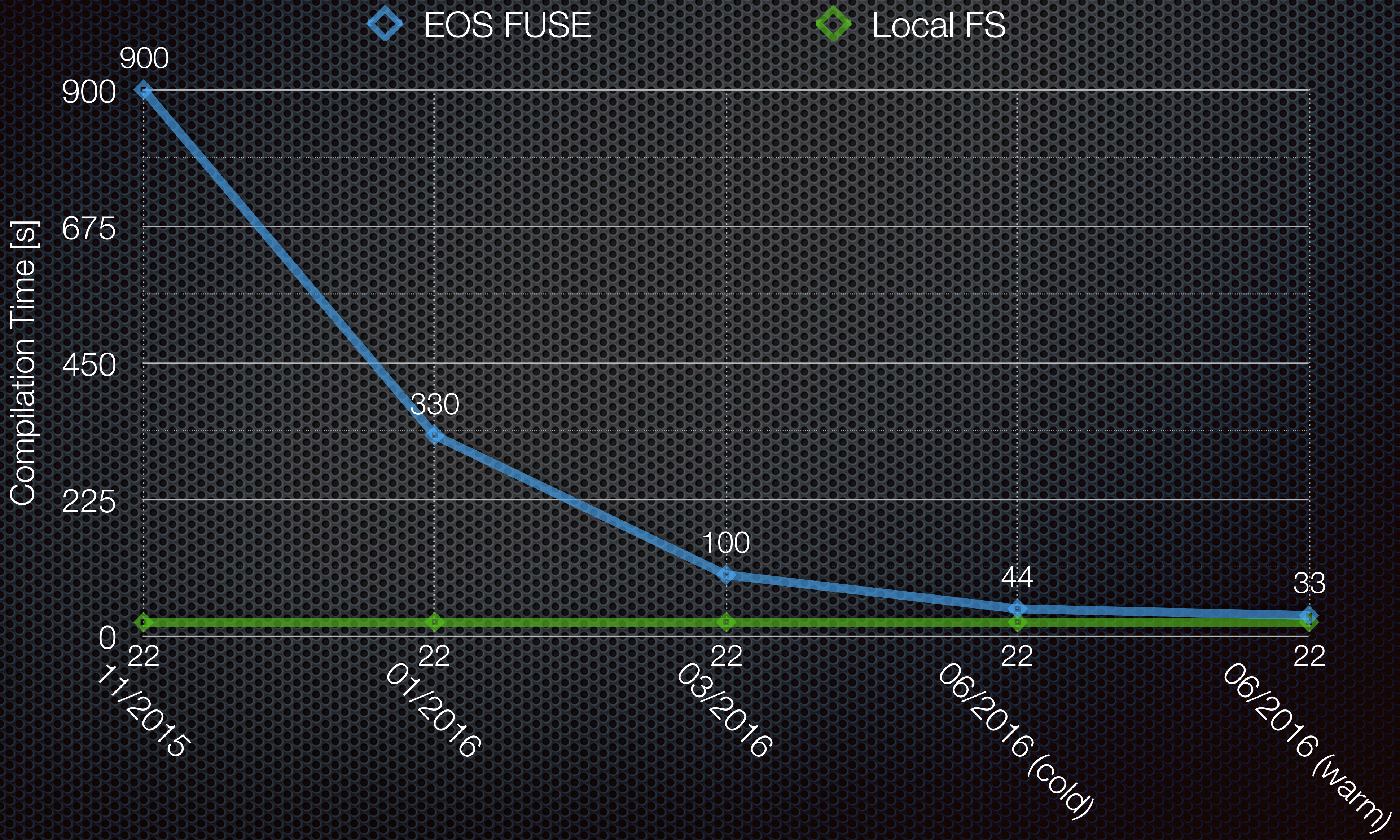
- Refactoring Ingredients
  - **parallelize**
  - avoid sync => use **async** where possible
  - avoid small IOs - **aggregate** IOs
  - reduce roundtrips - **bulk** operations
    - readdir + N x stat = one call instead of (N+1)
  - **recover** errors

Refactored EOS FUSE

- evolved **eosfusebind** in CITRINE since **11/15**
- introduced **asynchronous open** mechanism in CITRINE **1/16**
- joined BERYL/CITRINE FUSE implementation in **2/16**
- pure **C++** implementation **3/16**
  - reused work by Justin Salmon (FUSE template)
  - reused work by Michal Simon (Rados FUSE)
  - use negative stat cache of kernel
  - path name encoding
  - server announces features to client
- introduced restore & repair functionality **5/16**
- provided Mac OS X package **5/16**
- performance and (mtime) consistency improvements **3/16-today**
  - good momentum by /eos task force and external feedback from JRC & Aarnet

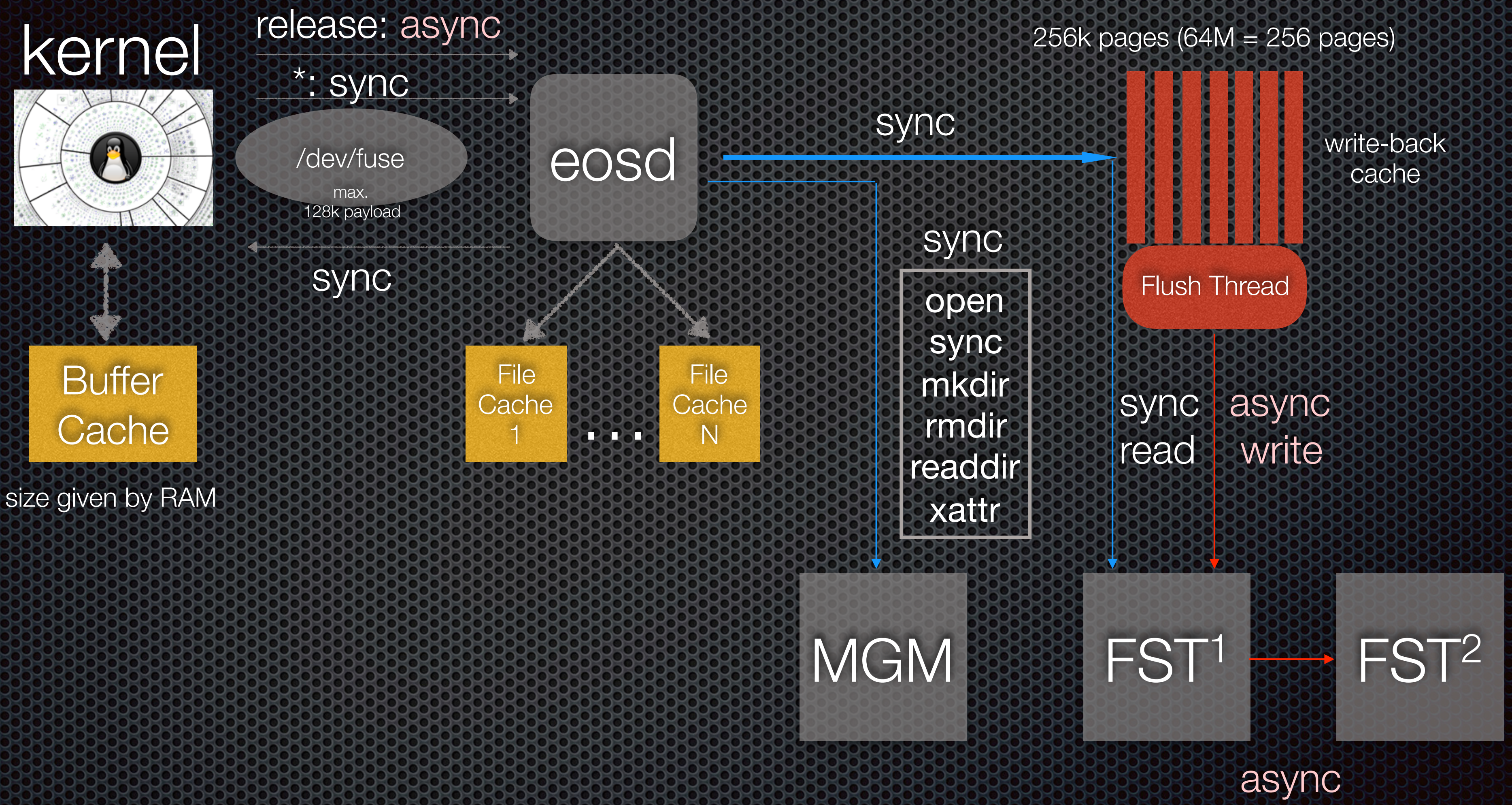
Performance Improvements

XRootD compilation benchmark



Current Implementation

libfuse 2.x



kernel

release: async

*: sync

/dev/fuse

max.
128k payload

sync

eosd

Buffer Cache

size given by RAM

File Cache 1

. . .

File Cache N

sync

sync

open
sync
mkdir
rmdir
readdir
xattr

256k pages (64M = 256 pages)

write-back cache

Flush Thread

sync read

async write

MGM

FST^1

FST^2

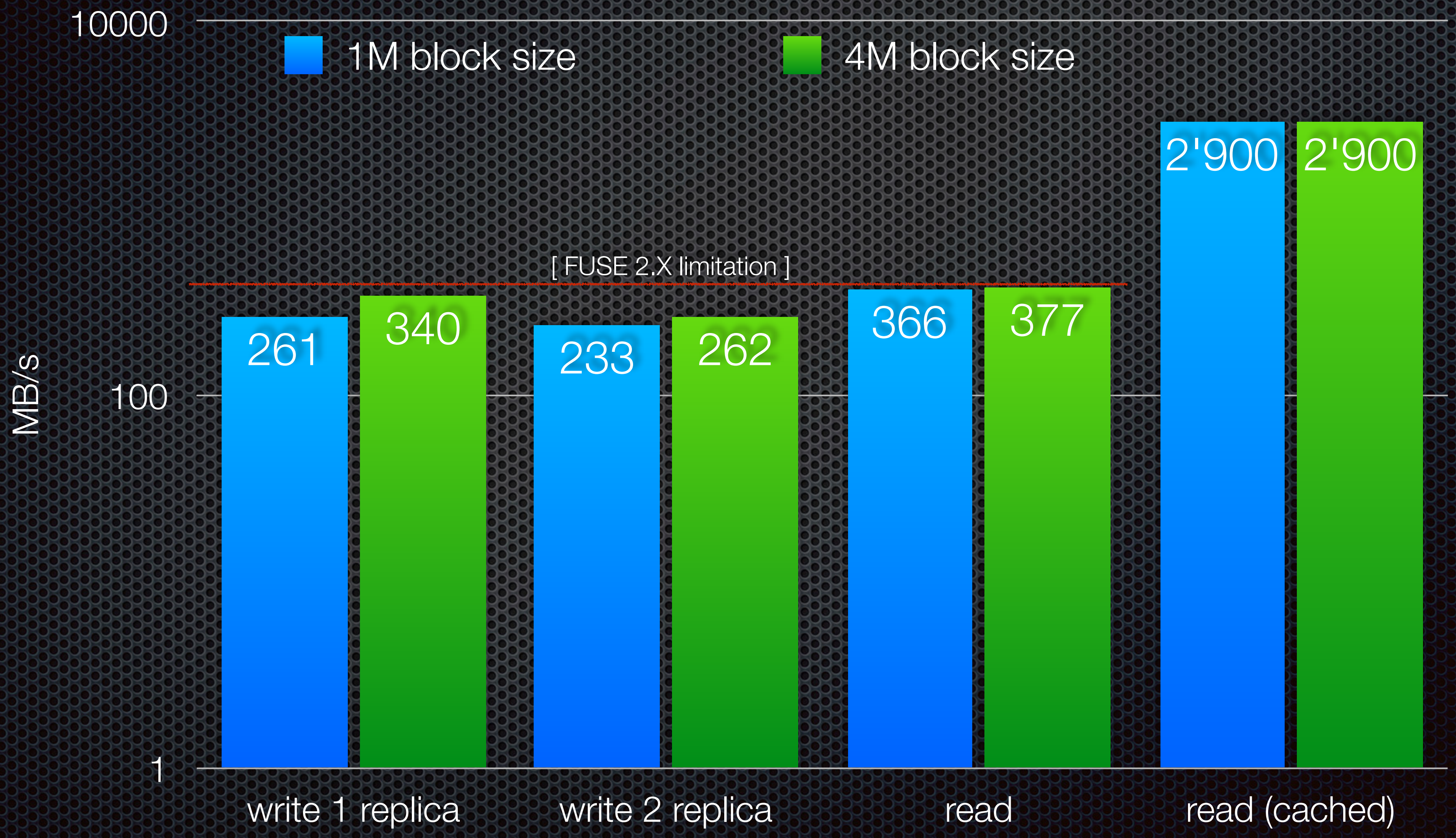
async

libfuse 3.x

uses linux buffer cache as write-back cache - requires new kernel - can be patched for large IOs 128k => 1M for 2 GB/s throughput

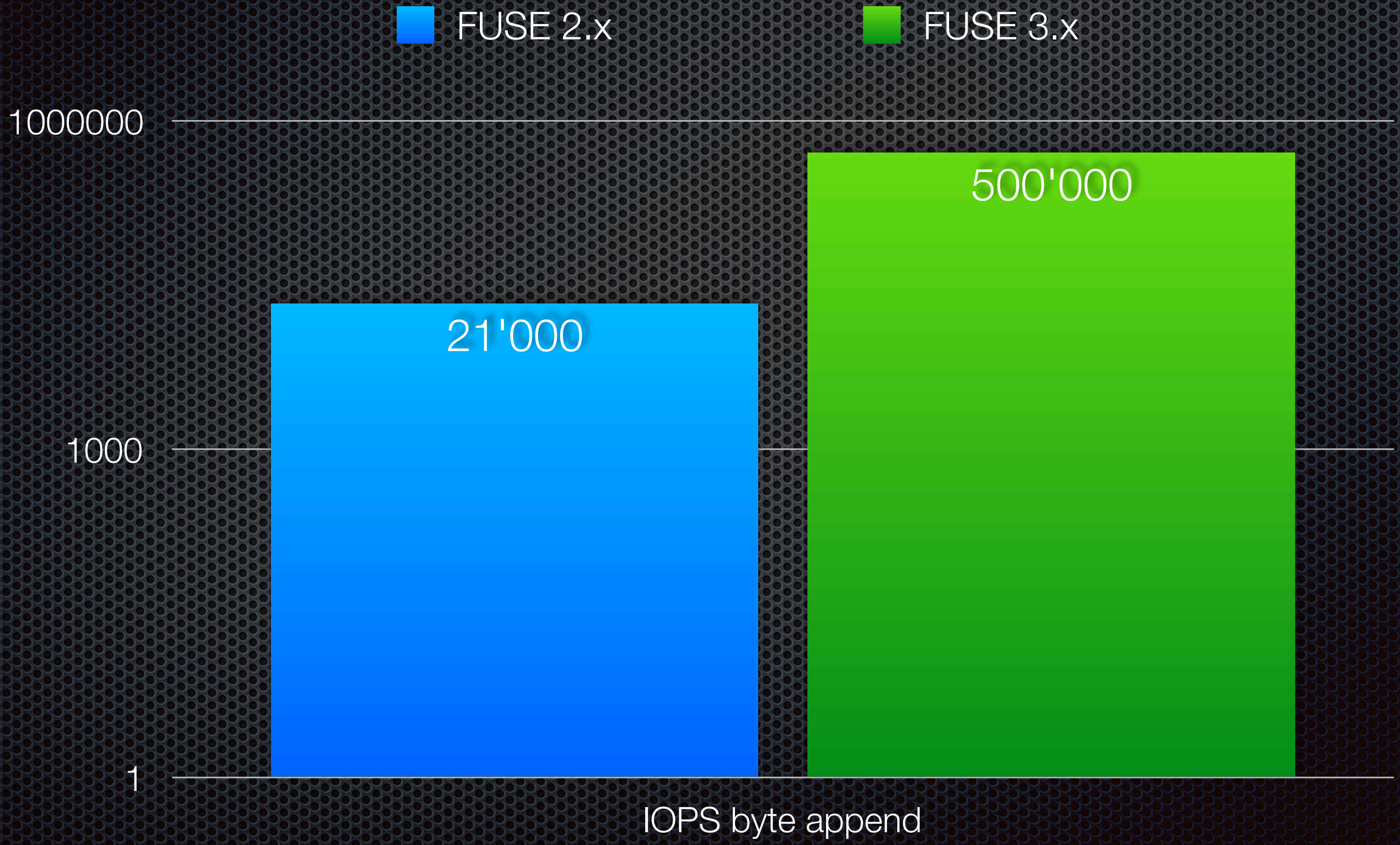
FUSE - Throughput

default 64M write-back cache - 10GE client

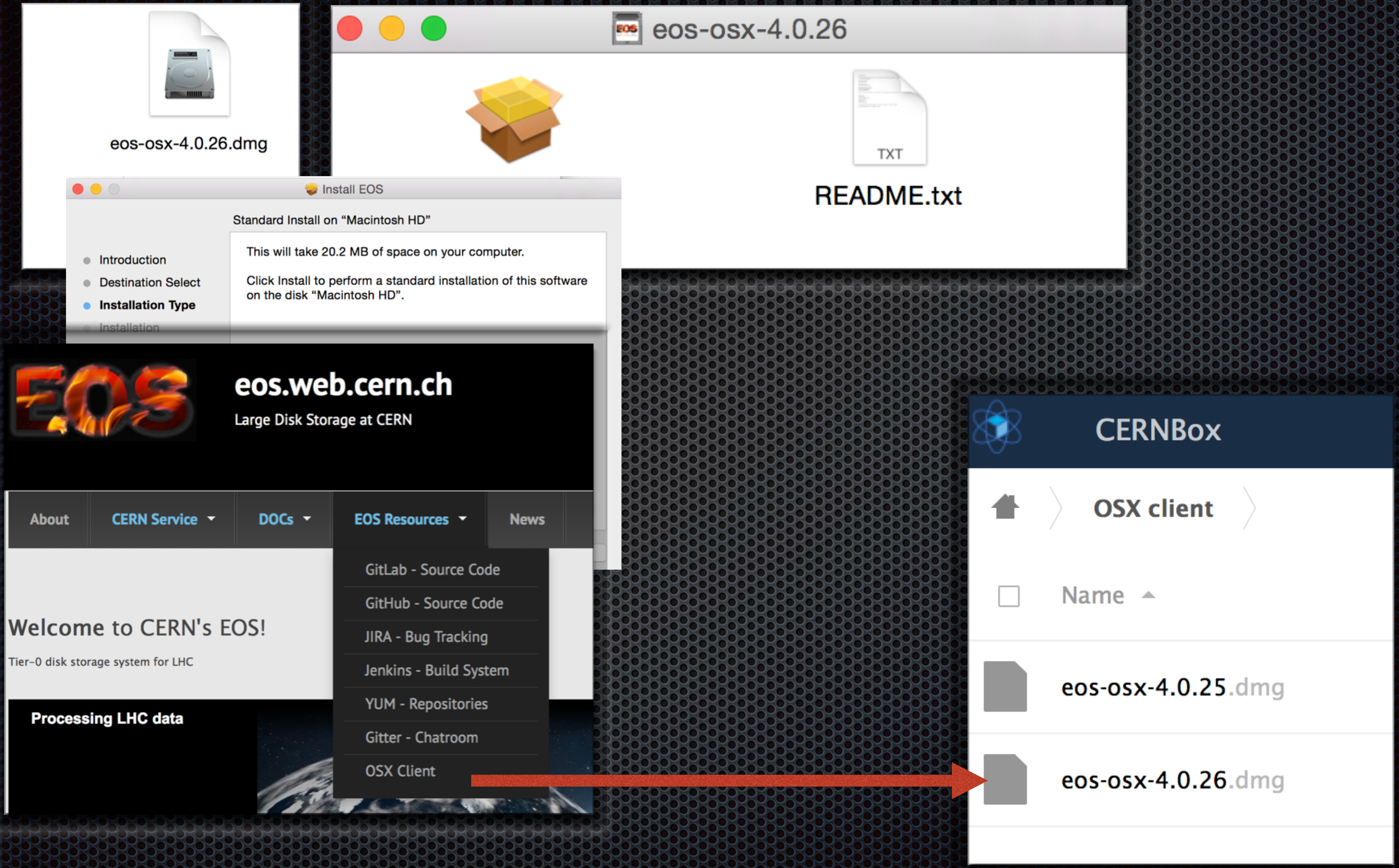


FUSE - IOPS

file fragments < cache fragments



FUSE Mac OSX Client



Standard Install on “Macintosh HD”

This will take 20.2 MB of space on your computer.

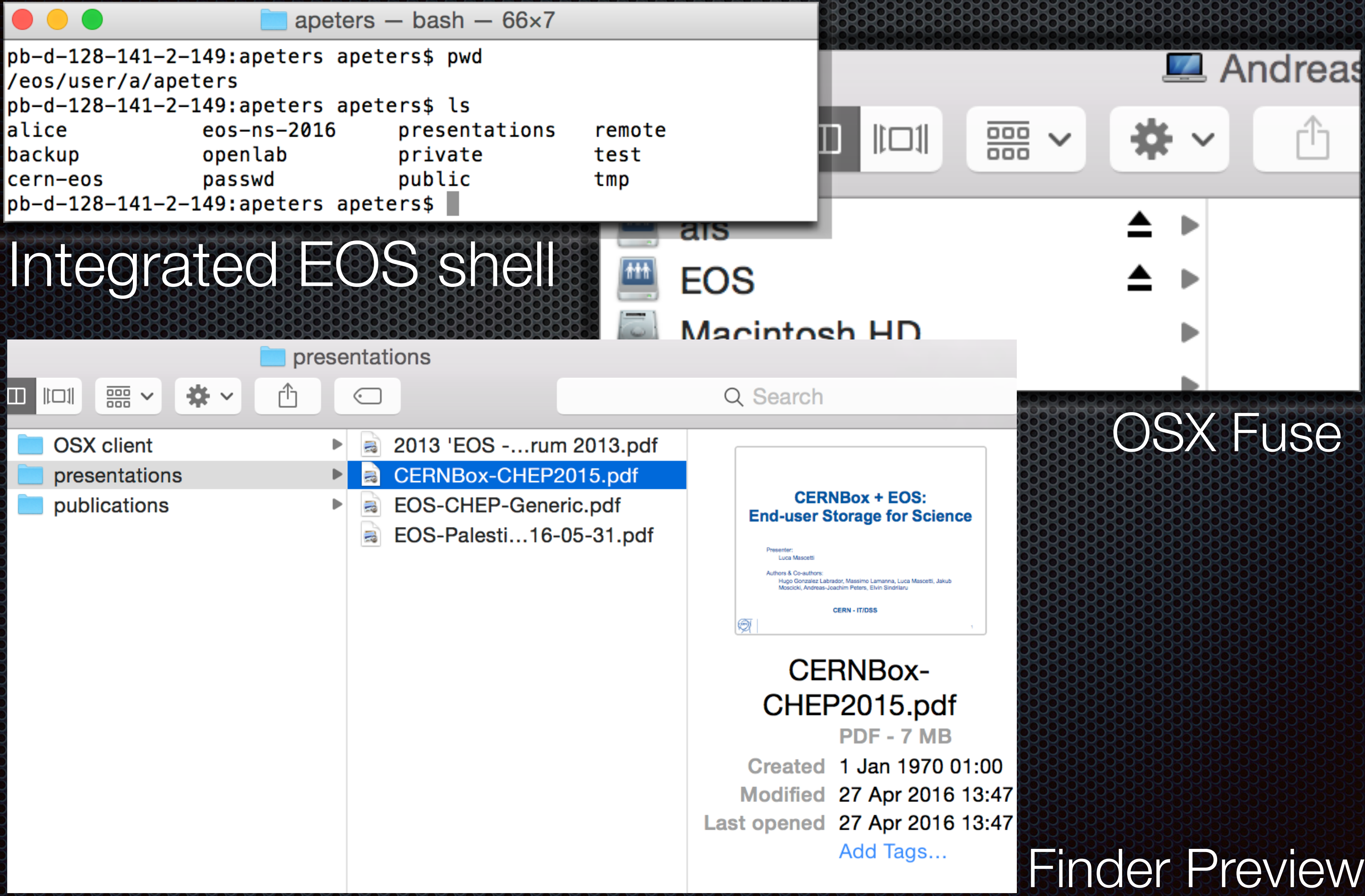
Click Install to perform a standard installation of this software on the disk “Macintosh HD”.

Welcome to CERN's EOS!

Tier–0 disk storage system for LHC

Find it on eos.web.cern.ch

FUSE Mac OSX Client



```
pb-d-128-141-2-149:apeters apeters$ pwd
/eos/user/a/apeters
pb-d-128-141-2-149:apeters apeters$ ls
alice           eos-ns-2016     presentations   remote
backup          openlab         private         test
cern-eos        passwd          public          tmp
pb-d-128-141-2-149:apeters apeters$
```

Integrated EOS shell

OSX Fuse

Finder Preview

issues on the way

- **mtime consistency** - any violation of modification time consistency was visible to rsync, emacs, vim, vi etc.
- **memory leaks** - FUSE daemon is a long-running daemon, repetitive tests reveal quickly leaks - sometimes leaks were only triggered under certain timing/error conditions - we observe still a **long-term memory increase** when running under CIFS
- **inode - name consistency** - rename of open files
- **stability** - improved significantly - thanks David!

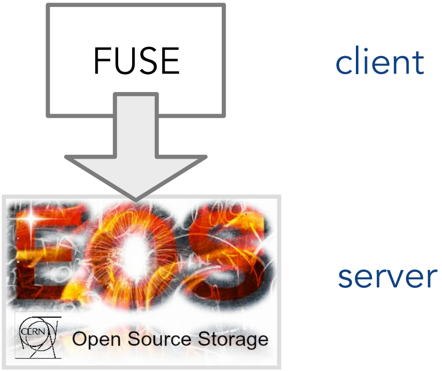
/eos at CERN (see DevOps talk)

- currently **client deployed on**
  - lxplus, lxbatch cluster
  - on **SWAN** services
- BUT: there are known limitations considering consistency & performance which finally lead to the plan to **implement a third generation** in the way a filesystem works

A third generation for EOS FUSE - eosxd

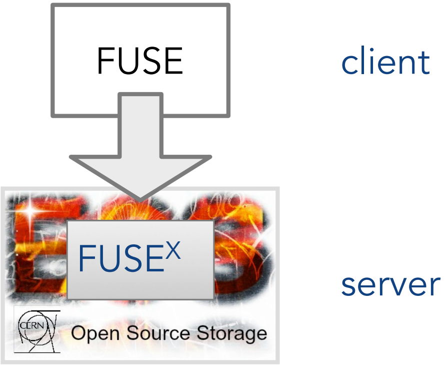
Architectural Change

V2 implementation



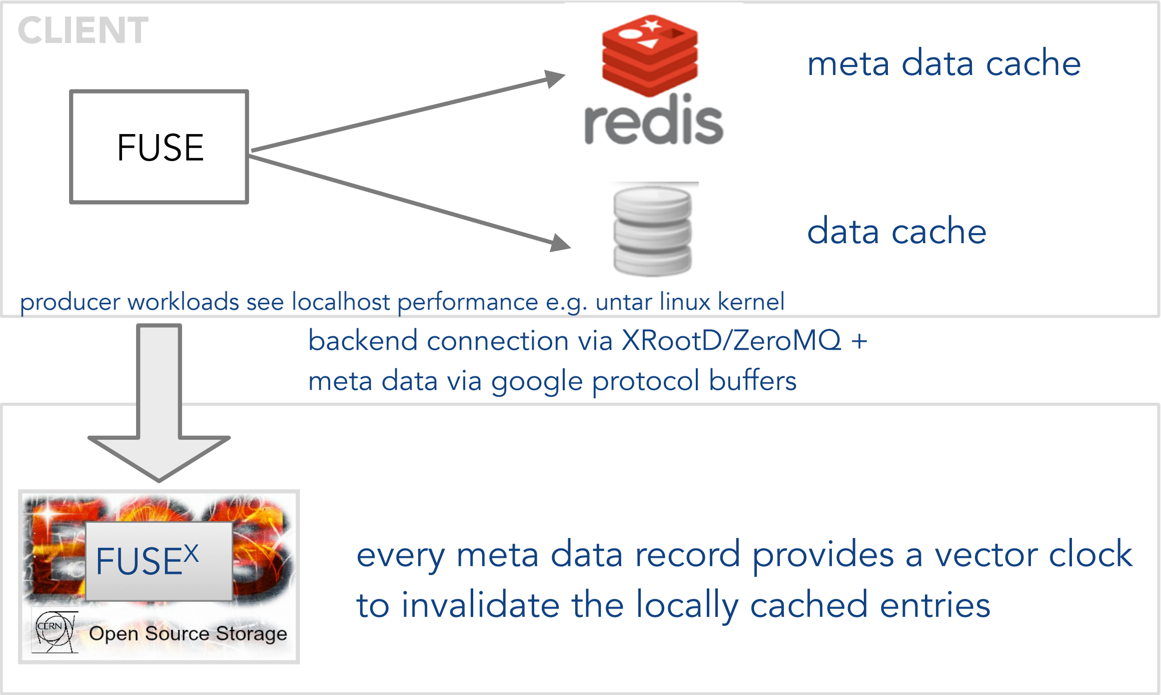
FUSE filesystem implemented as **pure client side** application without dedicated server side support.

V3 implementation



Dedicated server-side support providing a fully asynchronous server->client communication, leases, locks, file inlining, local meta-data and data caching

Architectural Change



meta data cache

data cache

producer workloads see localhost performance e.g. untar linux kernel

backend connection via XRootD/ZeroMQ +
meta data via google protocol buffers

every meta data record provides a vector clock
to invalidate the locally cached entries

Development Phases

- **Phase 1**
  - standalone front-end implementing persistent client side meta- an data cache
    - simplified configuration ✓
    - guaranteed local consistency & performance ✓
    - kernel NFS4 compatible ✓
- **Phase 2**
  - fully asynchronous protocol between client-server in both directions ✓
  - client heartbeats & server-initiated eviction ✓
  - meta-data vector clocks ✓

Development Phases

- **Phase 3**
  - meta-data upstream connection & small file handling
- **Phase 4**
  - large file & client cache handling
- **Phase 5**
  - locks & leases
- **Phase 6**
  - enabling client-side kernel cache & up-calls
- **Phase 7**
  - drop ZMQ and use XRootD SSI2

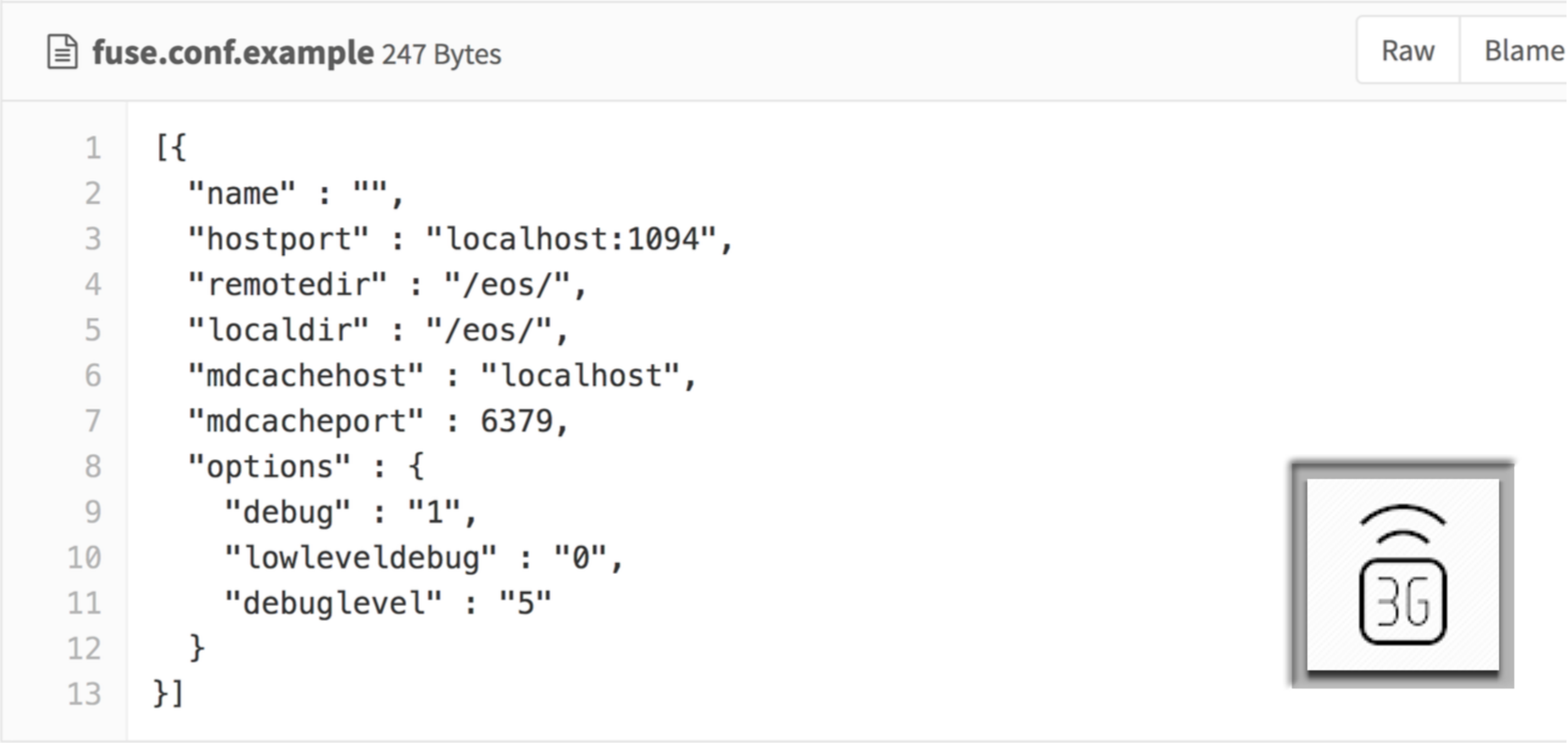
now

→ 1st release (Q2/17)

enabling optional

Phase 1 - simpler configuration

/etc/eos/fuse[.$name].conf

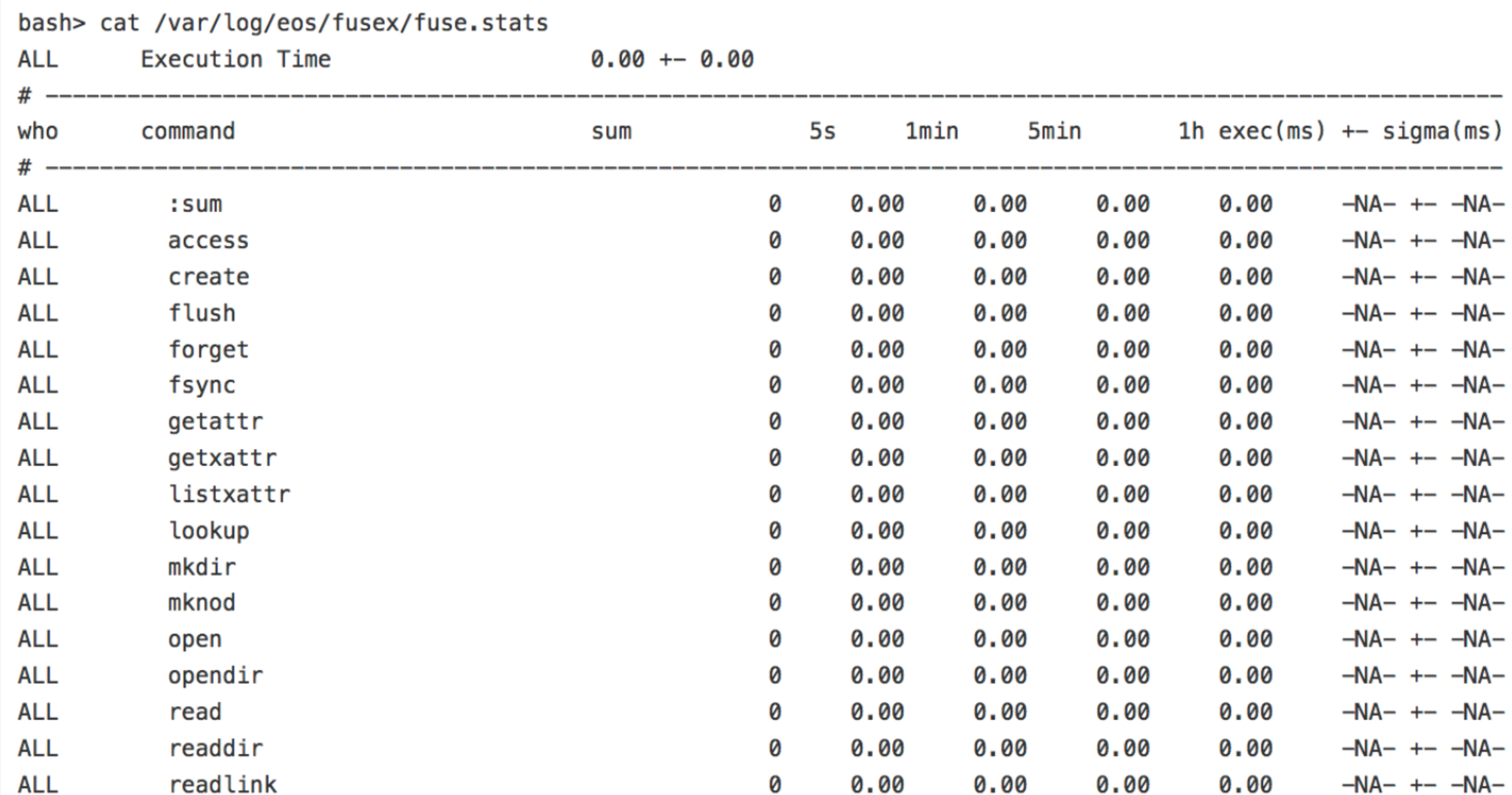


fuse.conf.example 247 Bytes

```
[{
  "name" : "",
  "hostport" : "localhost:1094",
  "remotedir" : "/eos/",
  "localdir" : "/eos/",
  "mdcachehost" : "localhost",
  "mdcacheport" : 6379,
  "options" : {
    "debug" : "1",
    "lowleveldebug" : "0",
    "debuglevel" : "5"
  }
}]
```

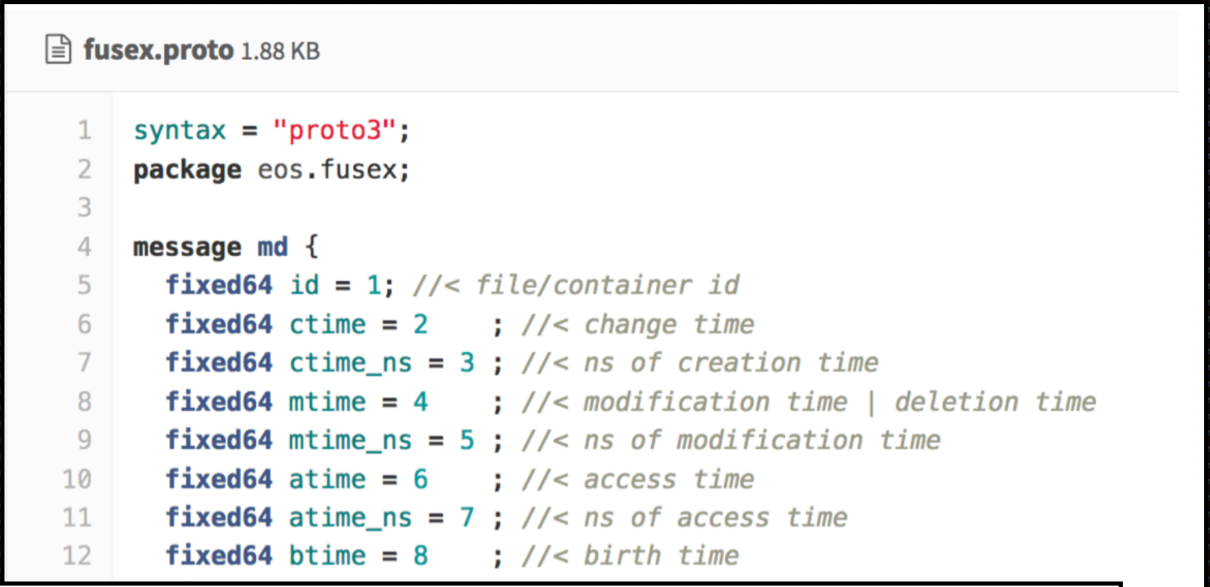
Phase 1 - real-time stats

/etc/eos/fuse[.$name].stats



```
bash> cat /var/log/eos/fusex/fuse.stats
ALL      Execution Time                   0.00 +- 0.00
# ----------------------------------------------------------------------------------------------------------
who      command                          sum             5s      1min      5min        1h exec(ms) +- sigma(ms)
# ----------------------------------------------------------------------------------------------------------
ALL        :sum                                       0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        access                                     0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        create                                     0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        flush                                      0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        forget                                     0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        fsync                                      0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        getattr                                    0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        getxattr                                   0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        listxattr                                  0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        lookup                                     0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        mkdir                                      0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        mknod                                      0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        open                                       0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        opendir                                    0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        read                                       0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        readdir                                    0     0.00      0.00      0.00      0.00      -NA- +- -NA-
ALL        readlink                                   0     0.00      0.00      0.00      0.00      -NA- +- -NA-
```

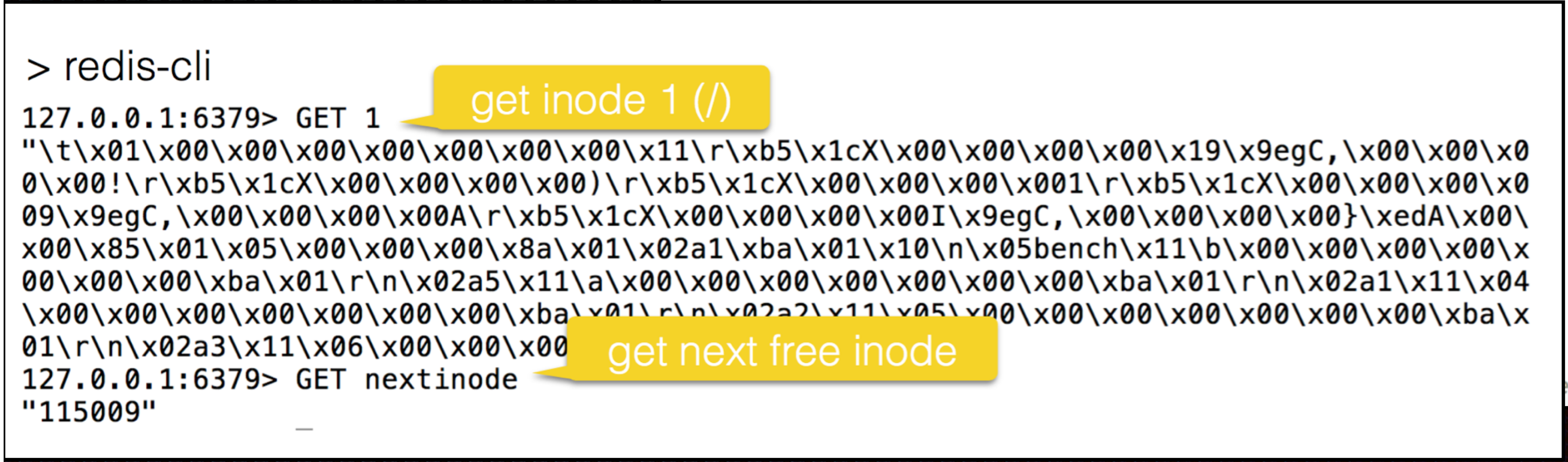
Phase 1 - protobuf MD Redis cache



fusex.proto 1.88 KB

```
syntax = "proto3";
package eos.fusex;

message md {
  fixed64 id = 1; //< file/container id
  fixed64 ctime = 2    ; //< change time
  fixed64 ctime_ns = 3 ; //< ns of creation time
  fixed64 mtime = 4    ; //< modification time | deletion time
  fixed64 mtime_ns = 5 ; //< ns of modification time
  fixed64 atime = 6    ; //< access time
  fixed64 atime_ns = 7 ; //< ns of access time
  fixed64 btime = 8    ; //< birth time
```



```
> redis-cli
127.0.0.1:6379> GET 1
"\t\x01\x00\x00\x00\x00\x00\x00\x00\x11\r\xb5\x1cX\x00\x00\x00\x00\x19\x9egC,\x00\x00\x0
0\x00!\r\xb5\x1cX\x00\x00\x00\x00)\r\xb5\x1cX\x00\x00\x00\x001\r\xb5\x1cX\x00\x00\x00\x0
09\x9egC,\x00\x00\x00\x00A\r\xb5\x1cX\x00\x00\x00\x00I\x9egC,\x00\x00\x00\x00}\xedA\x00\
x00\x85\x01\x05\x00\x00\x00\x8a\x01\x02a1\xba\x01\x10\n\x05bench\x11\b\x00\x00\x00\x00\x
00\x00\x00\xba\x01\r\n\x02a5\x11\a\x00\x00\x00\x00\x00\x00\x00\xba\x01\r\n\x02a1\x11\x04
\x00\x00\x00\x00\x00\x00\x00\xba\x01\r\n\x02a2\x11\x05\x00\x00\x00\x00\x00\x00\x00\xba\x
01\r\n\x02a3\x11\x06\x00\x00\x00
127.0.0.1:6379> GET nextinode
"115009"
```

get inode 1 (/)

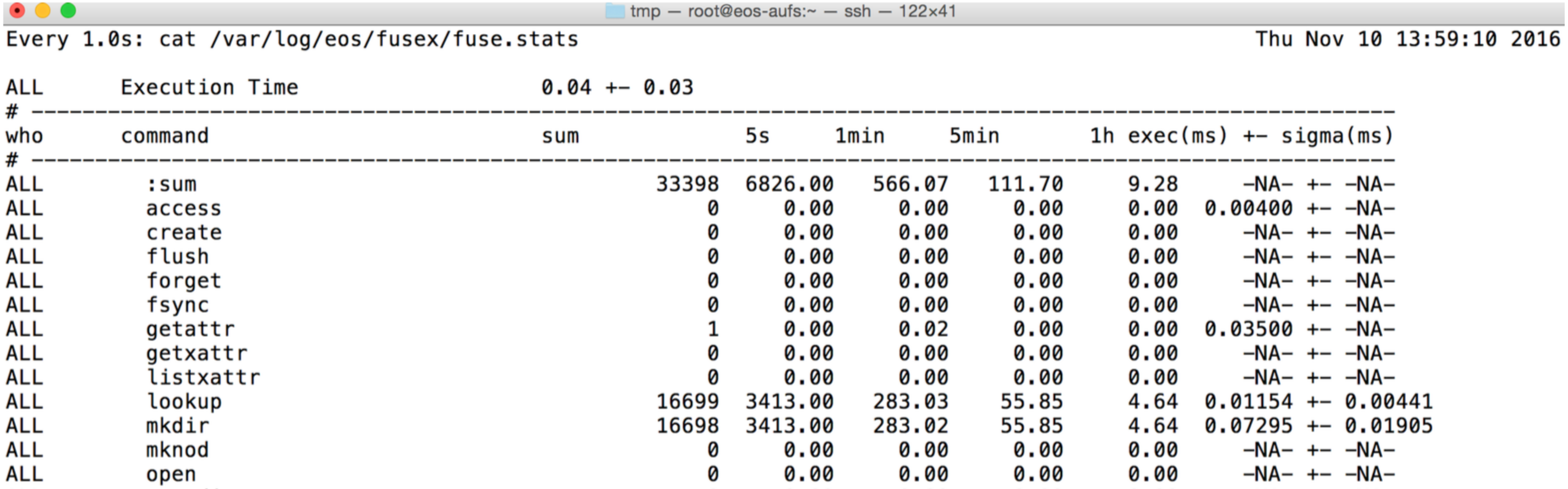
get next free inode

100 x (untar(300) ; rm -rf)

30.000 creates + deletes

31SEC FUSE POOL
33SEC C++ POOL

C++ for (i=0;i< 100000; i++) {mkdir (i)}



```
Every 1.0s: cat /var/log/eos/fusex/fuse.stats                                                     Thu Nov 10 13:59:10 2016

ALL        Execution Time                   0.04 +- 0.03
# ----------------------------------------------------------------------------------------------------------------
who        command                          sum             5s       1min       5min       1h exec(ms) +- sigma(ms)
# ----------------------------------------------------------------------------------------------------------------
ALL          :sum                                 33398  6826.00    566.07     111.70       9.28      -NA- +- -NA-
ALL          access                                   0     0.00      0.00       0.00       0.00   0.00400 +- -NA-
ALL          create                                   0     0.00      0.00       0.00       0.00      -NA- +- -NA-
ALL          flush                                    0     0.00      0.00       0.00       0.00      -NA- +- -NA-
ALL          forget                                   0     0.00      0.00       0.00       0.00      -NA- +- -NA-
ALL          fsync                                    0     0.00      0.00       0.00       0.00      -NA- +- -NA-
ALL          getattr                                  1     0.00      0.02       0.00       0.00   0.03500 +- -NA-
ALL          getxattr                                 0     0.00      0.00       0.00       0.00      -NA- +- -NA-
ALL          listxattr                                0     0.00      0.00       0.00       0.00      -NA- +- -NA-
ALL          lookup                               16699  3413.00    283.03      55.85       4.64   0.01154 +- 0.00441
ALL          mkdir                                16698  3413.00    283.02      55.85       4.64   0.07295 +- 0.01905
ALL          mknod                                    0     0.00      0.00       0.00       0.00      -NA- +- -NA-
ALL          open                                     0     0.00      0.00       0.00       0.00      -NA- +- -NA-
```

Performance Phase 1 Implementation

tar xvf linux-4.9.tar.xz

56 SEC

60038 files/dirs

Summary

- During the last year the second FUSE generation implementation made considerable progress
- EOS **FUSE2 performance** and stability has been **significantly improved** last year - strong security model is working
  - it does not have the look&feel of AFS (latency etc.)
  - difficult to have low latency & consistency
- EOS **FUSE2** has non-resolvable issues considering consistency and performance baselines - way to go is redesign =>
- EOS **FUSE3**
  - adding server side support and asynchronous server-to-client communication
  - reimplementation on the way - benchmarks promising
  - you can contribute with testing to its success!

tenki
hvala
ขอขอบคุณคุณ
takk
спасибо
kam sah hamnida
dank je
дзякуй
dziękuję
bedankt
הדות
dhanyavadagalu
tack
gracias
blagodaram
mèsi
xièxie
tanemirt
rahmet
enkosi
mochchakkeram
trugarez
djere dieuf
danke
kop khun krap
faafetai lava
arigatô
shukriya
ありがとう
kia ora
dankon
děkuji
manana
diolch
akun
grazzi
gràçie
kiitos
spas
ačiū
mamnun
barka
thank you
dhanyavad
gratias ago
tau
dankie
शुक्रिया
sulpáy
tapadh leat
chnorakaloutioun
gràcies
આભાર
teşekkür ederim
bayarlalaa
σας ευχαριστώ
obrigada
sagolun
murakoze
taiku
mahalo
didi madloba
chokrane
rahmat
sukriya
obrigado
dakujem
misaotra
welalin
mercé
najis tuke
terima kasih
謝謝
köszönöm
اركش
asante
grazie
nandri
mersi
sobodi
mauruuru
merci
nanni
vinaka
matondo
cảm ơn bạn
go raibh maith agat
paldies
ngiyabonga