Ins & Outs of EOS

Elvin Sindrilaru
on behalf of the EOS Team

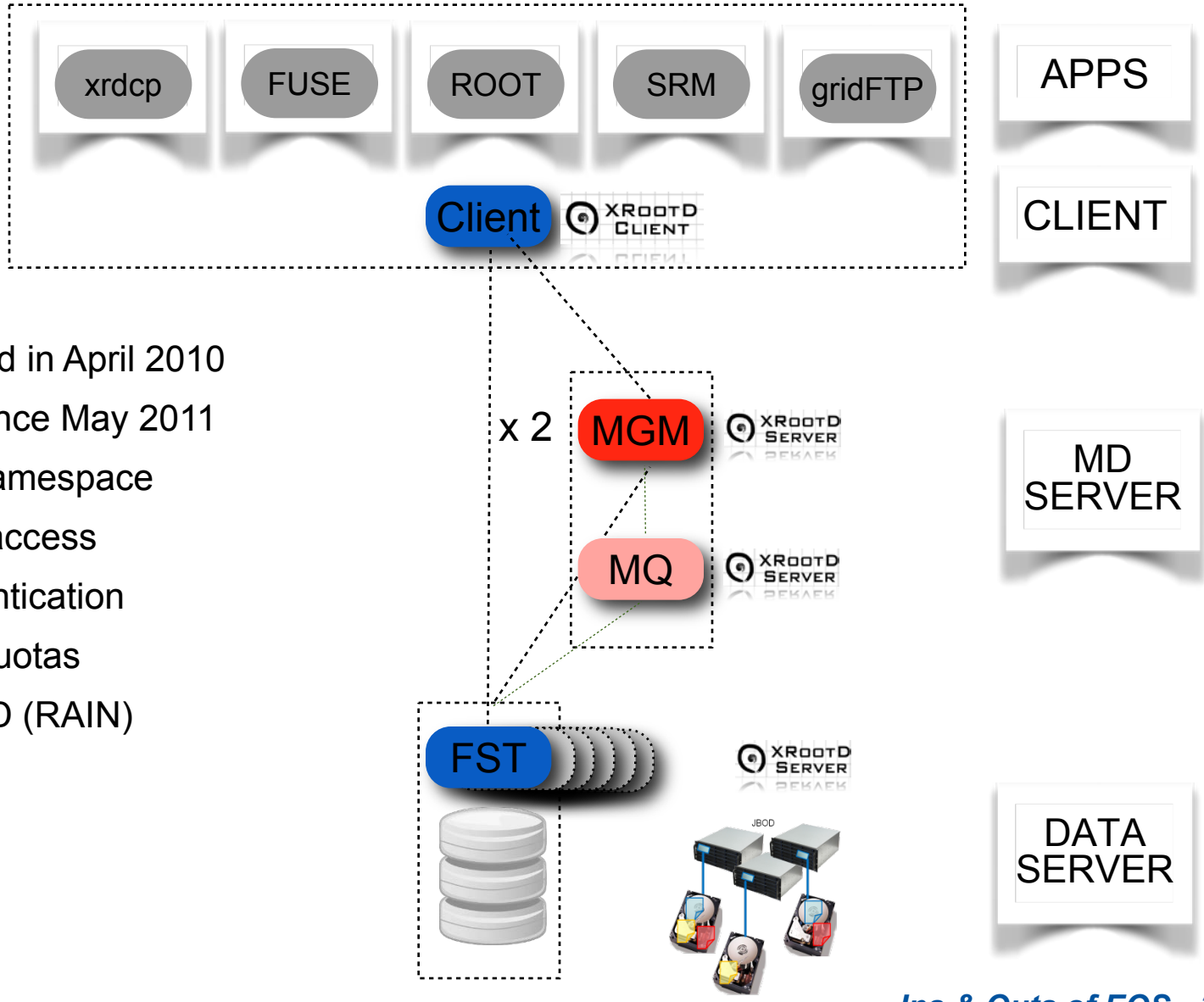
XRootD Workshop - UCSD
29.01.2015

CERN IT Department
CH-1211 Geneva 23
Switzerland
www.cern.ch/it

Outline

- EOS architecture and status update
- New features
  - Archive tool
  - Vector reads and RAIN layouts
  - XrdCl plugin for RAIN and vector reads
  - Authentication delegation
  - Geo-scheduling
- R&D and future directions
- Summary

EOS architecture



- Project started in April 2010
- Production since May 2011
- In-memory namespace
- Low latency access
- Strong authentication
- User/group quotas
- Network RAID (RAIN)
- Tunable QoS

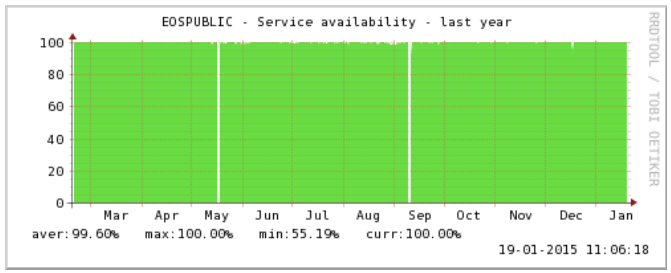
EOS status & new features

- EOS current production release **0.3 – Beryl**
  - Master/slave failover
  - Recycle bin + new ACLs
  - RAIN layouts
  - HTTPS/WebDav interface

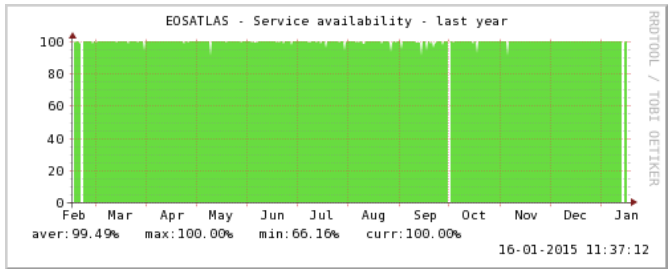
- Next major release **0.4 – Citrine** – 2015
  - Based on XRootD 4
  - Vector read support
  - Geo-scheduling
  - Archiving tool
  - Scalable authentication font-end

EOS service availability last year

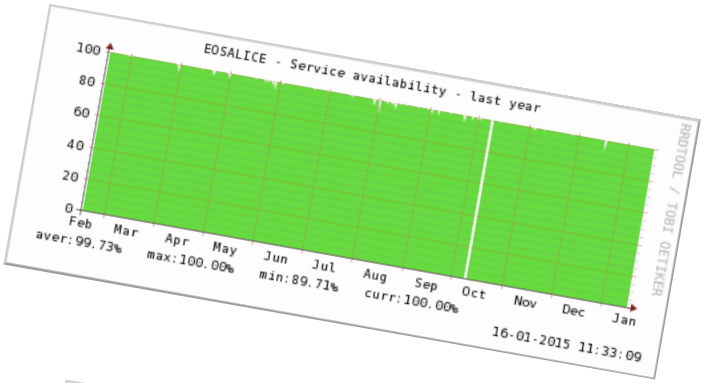
- **ALICE** – most stable **> 99.73%**
  - no SRM, no KRB5/GSI with ALICE Authz
- **PUBLIC** – lots of users **> 99.60%** - KRB5/GSI

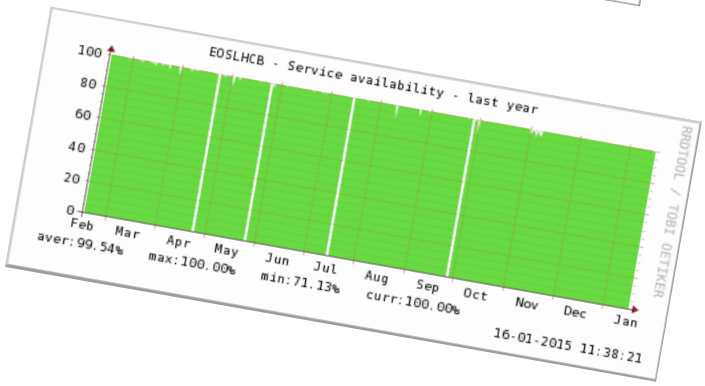


- **LHCb** – least number of files **> 99.54%**
  - SRM, KRB5/GSI
- **ATLAS** – most disks **> 99.49%** - SRM, KRB5/GSI

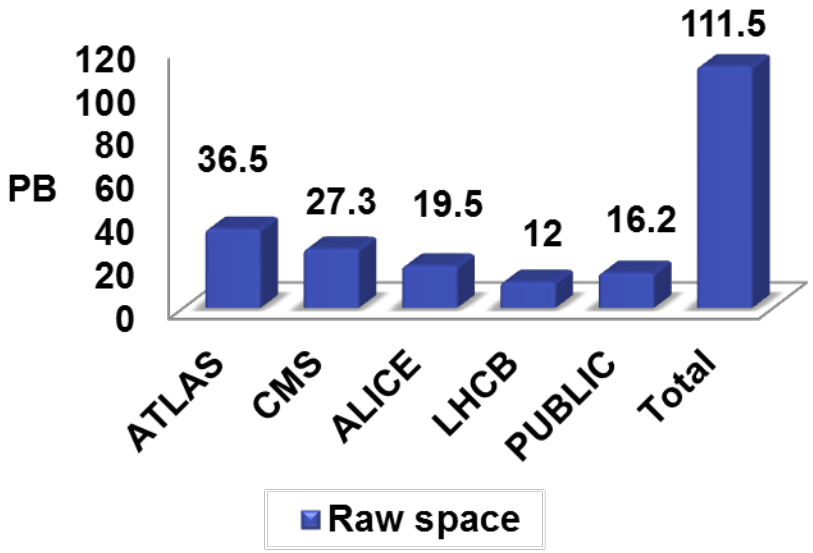


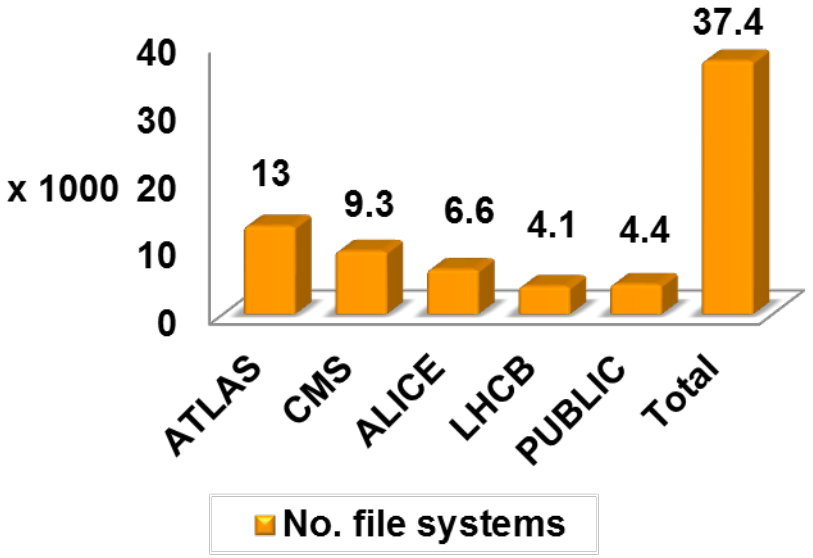
- **CMS** – mostly SRM issues **> 98.92%** - SRM, KRB5/GSI

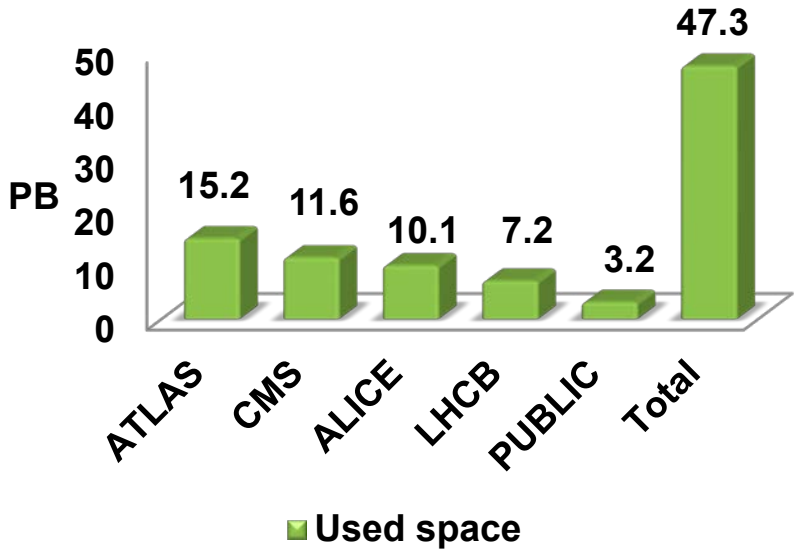


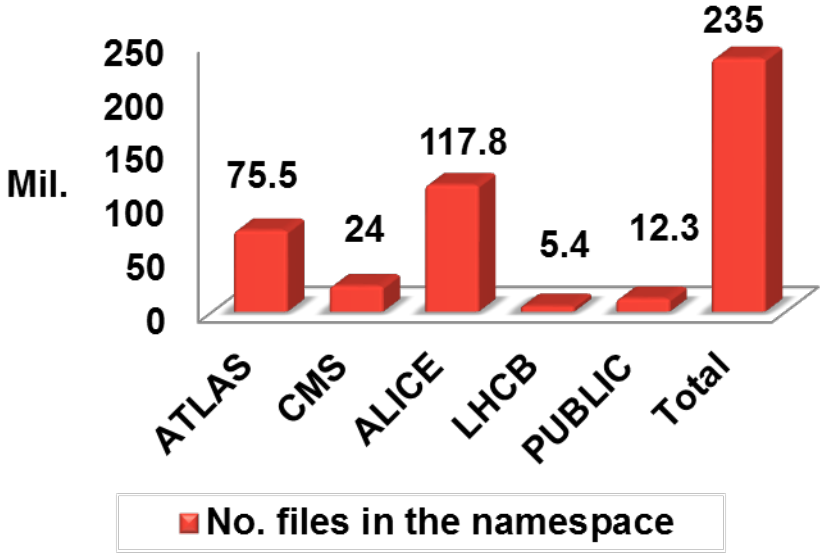


EOS service "size"





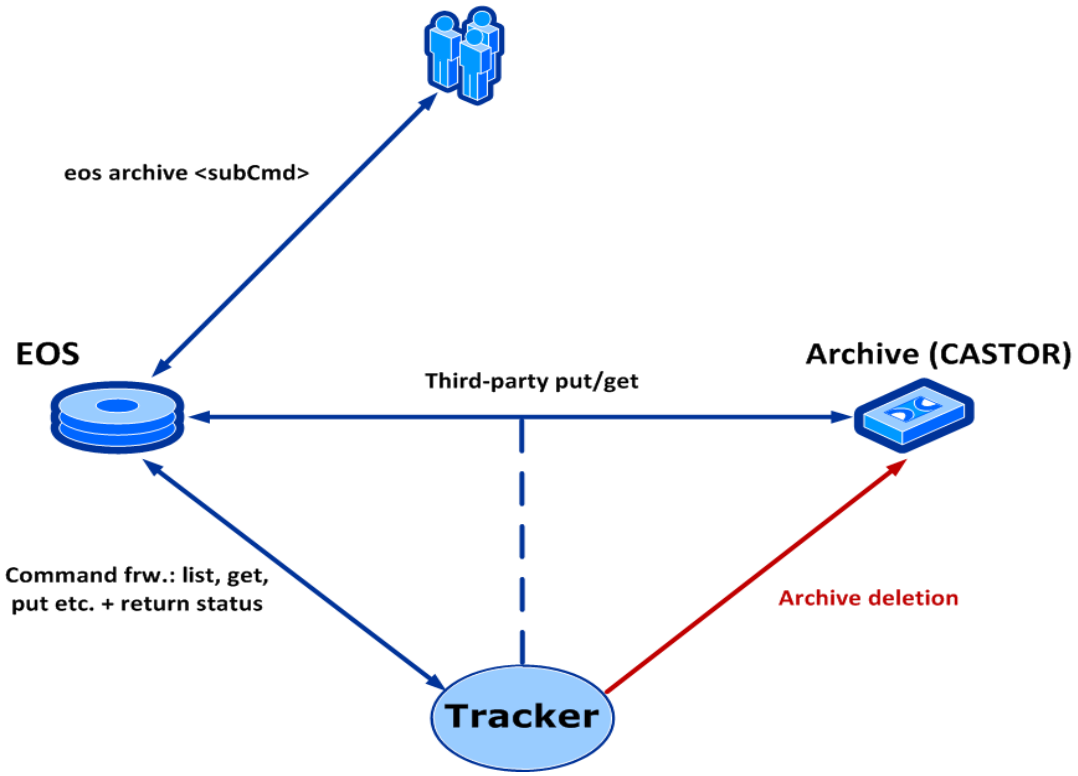




Why an archiving tool?

- EOS **quota** limit - free up storage space in EOS for users' online activities
- Spare users from developing *ad hoc* archiving solutions
- Manage **efficiently** the movement of data between disk and archive storage

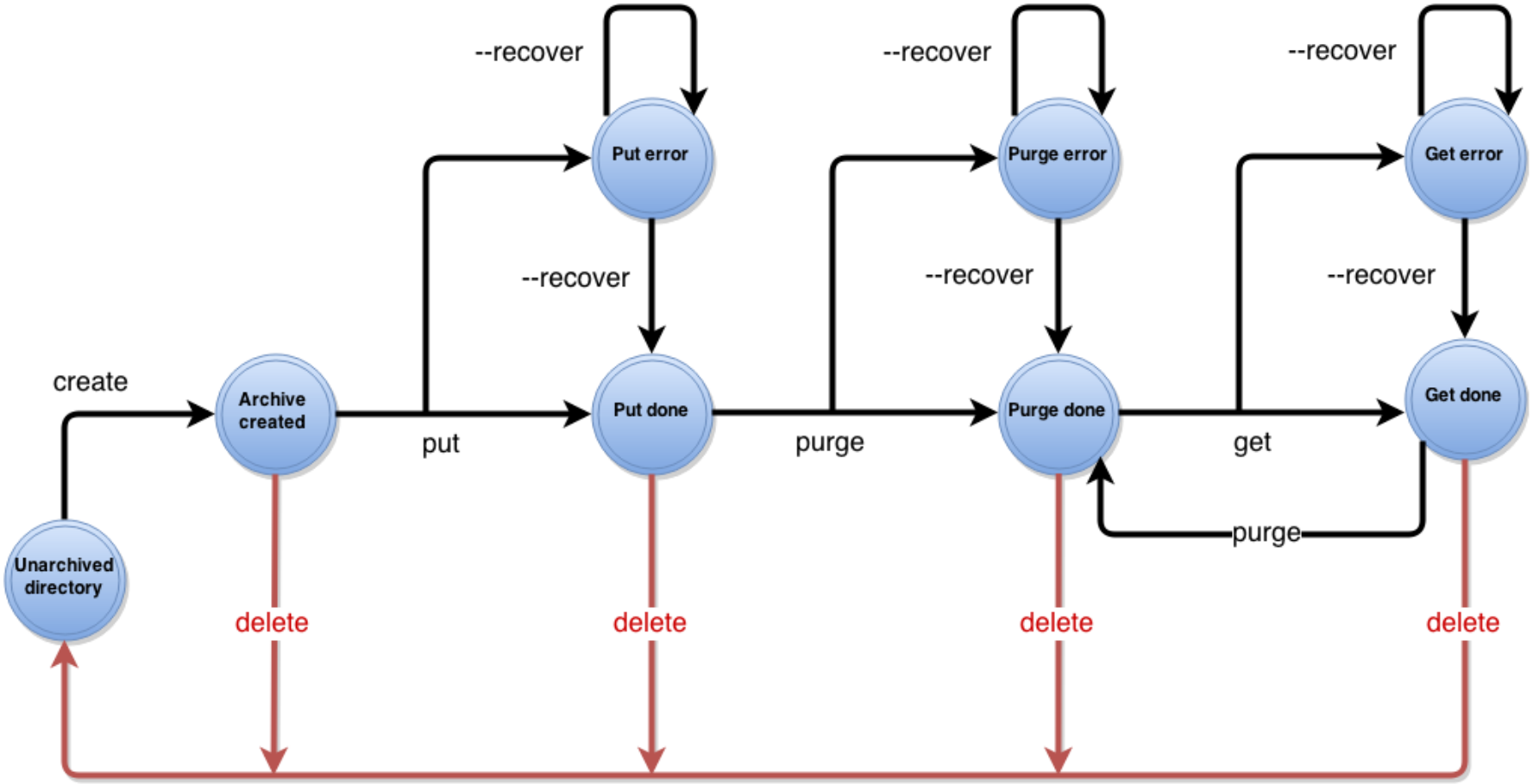
- Archive creation means:
  - EOS sub-tree freeze
  - No updates/writes of data or metadata are allowed
  - Uses **XRootD Third-Party Copy** in parallel (4.1)
  - Deletion of an archive is an **admin** command



Archive and log files

- **Archive file – contains entries in JSON format**
  - Header (source, destination, archive size etc.)
  - Directory/file EOS metadata information in JSON format
  - *Never* modified during the lifetime of an archive
  - Can be used *in the future* to get information about the contents of the archive
  - Archive *"get"* restores files in the original layout (2 replicas, RAIN etc.)
- **Archive log file – archive.log**
  - Summary of the last executed transfer
  - Hints to why a transfer has failed
  - Users can/should access it in case of errors

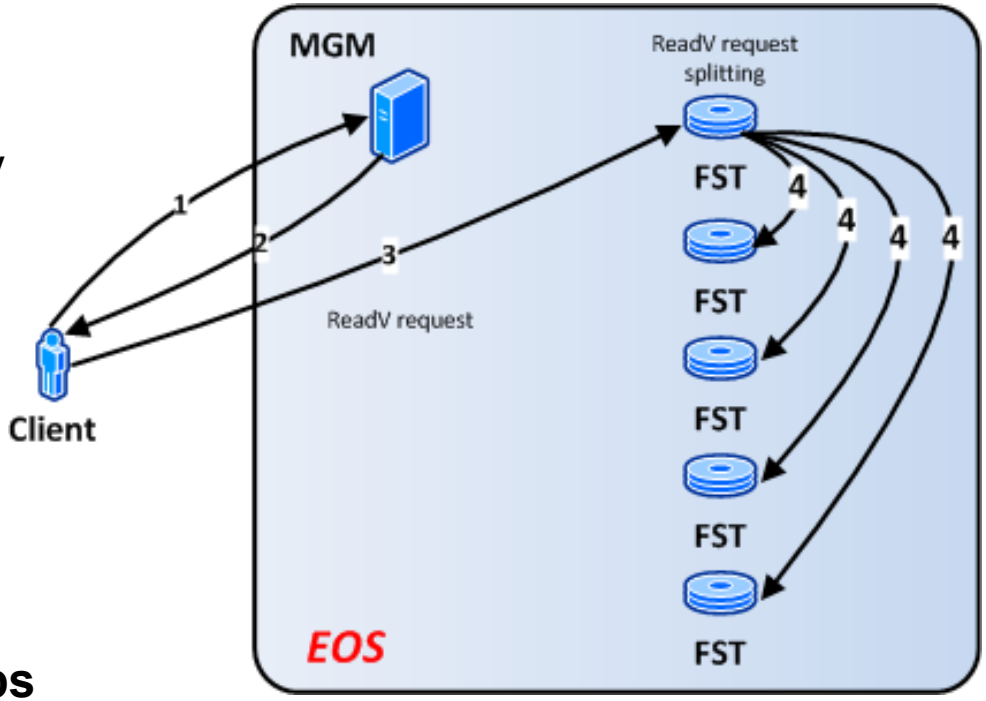
Archiving workflow



Vector reads and RAIN layouts (1)

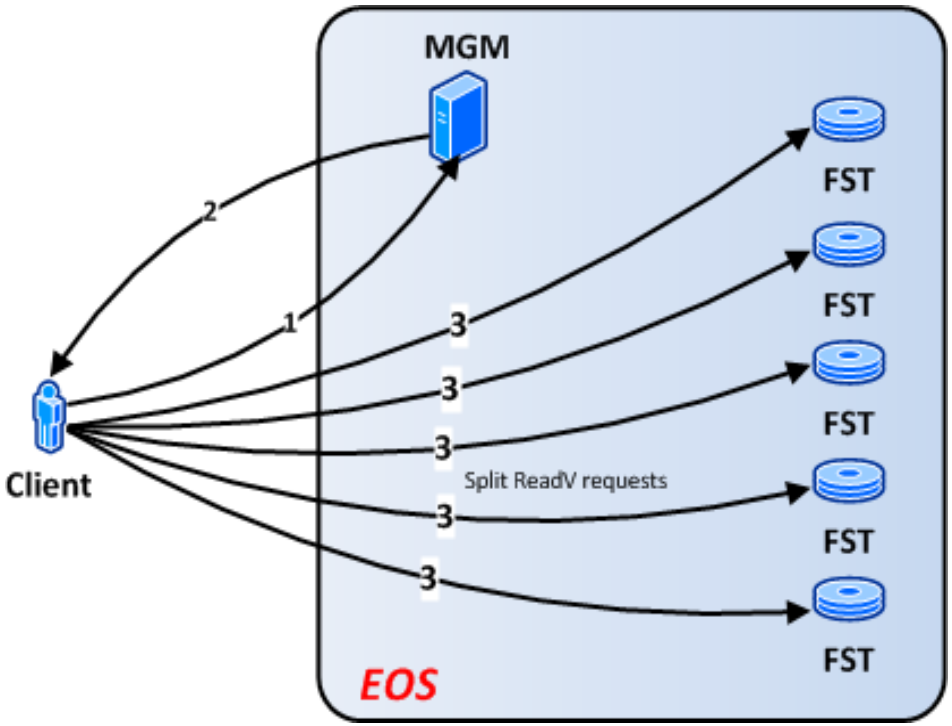
- **XRootD 4** supports natively vector reads
- EOS extends this concept to cover **RAIN** (Redundant Array of Independent Nodes) layouts

- **Gateway mode**
  - Vector reads split at the entry point
  - Individual reads are at least the size of the block checksum – **spot errors**
  - Direct **benefits for ROOT jobs** using the vector read API



Vector reads and RAIN layouts (2)

- **Parallel mode**
  - Vector reads split already at the client
  - Individual reads are at least the size of the blockxs
  - No double copy overhead

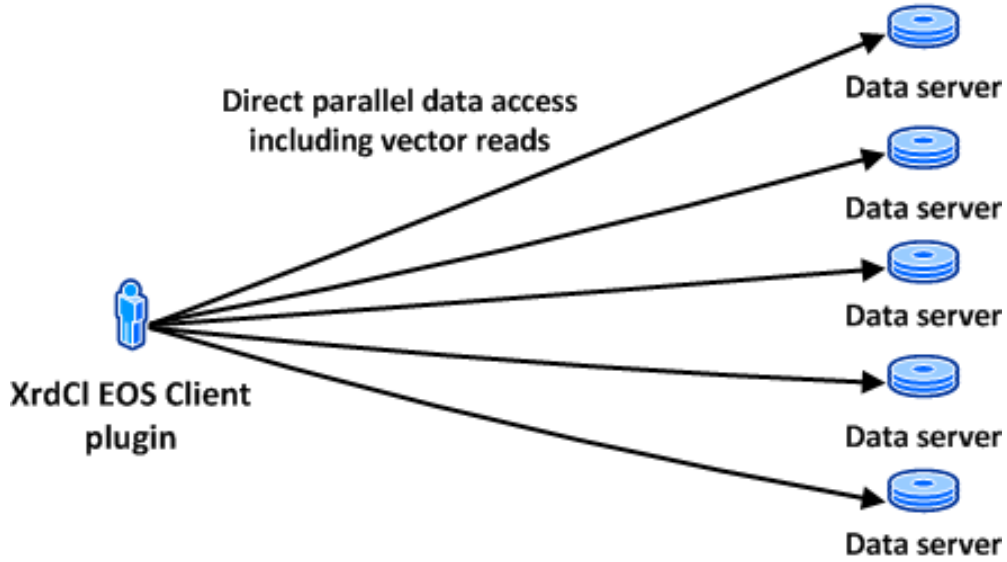


- Parallel mode without vector reads is already available in:
  - **eoscp** – EOS copy command
  - **FUSE**

- With XRootD 4 this functionality can also be implemented as an **XrdCl plugin**

XrdCl plugin for RAIN layouts

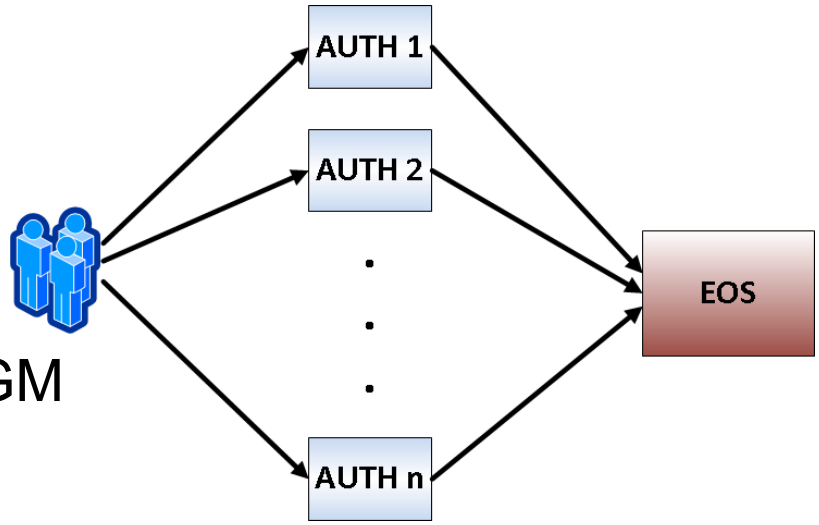
- XrdCl client plugin-in extending only the **File interface**



- **Some of the benefits:**
  - Contact the stripe servers directly
  - Move CPU intensive reconstruction operations to the client
  - Completely transparent for the layers above
  - Plugin library can be distributed separately from the core XRootD

EOS authentication delegation

- One authentication error can crash the whole EOS namespace and address scalability with a huge number of clients O(1000)
- **Solution?**
  - **Decouple** the authentication step from the rest of the metadata operations
  - All requests are forwarded to EOS using a **separate communication channel**



- Authentication done in one of the **AUTH** instances (XRootD)
- **No authentication** step at the MGM
- Avoid **CPU intensive operations** at the MGM

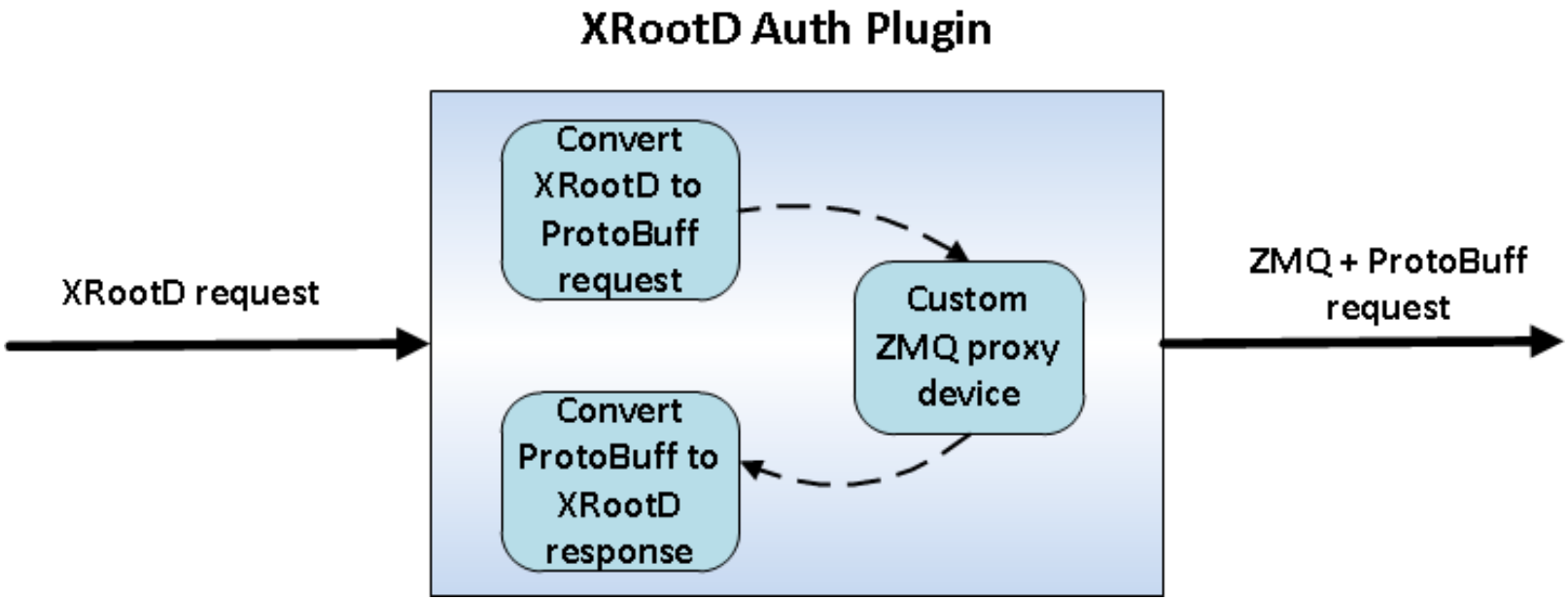
Auth. delegation - implementation

- **ZMQ** used for communication between AUTHx and EOS MGM
- Message serialization: **Google ProtocolBuffers** - forward and backward compatible
- Each object has a ProcolBuffers representation:
  - XrdSecEntity ------> XrdSecEntityProto
  - XrdOucErrInfo ------> XrdOucErrInfoProto etc.

- Each request type has its own ProtocolBuffer representation:
  - Directory open ------> DirOpenProto
  - Read from file ------> FileReadProto etc.

- Response object has the same structure for all requests

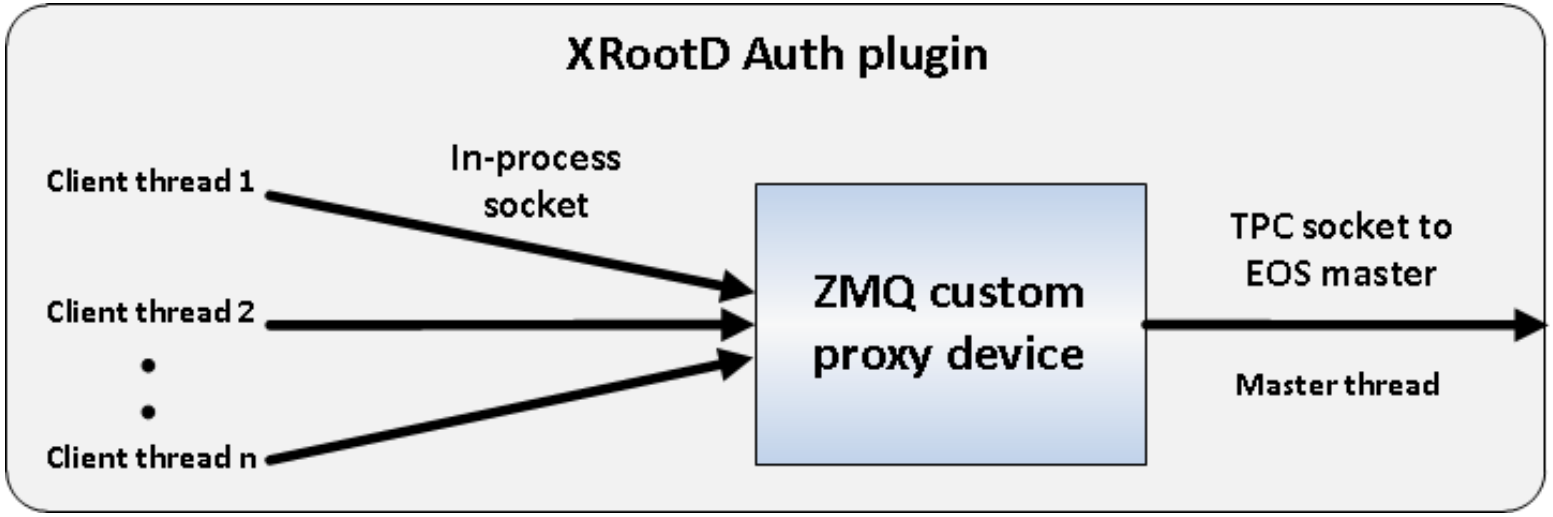
Auth. extension – front-end



- AUTH is an **XRootD server** with a modified OFS layer
- ZMQ – "***inter-process***" sockets inside Auth Plugin
  - ***TCP*** socket to contact the MGM
- Communication pattern: **REQ -> ROUTER -> DEALER**

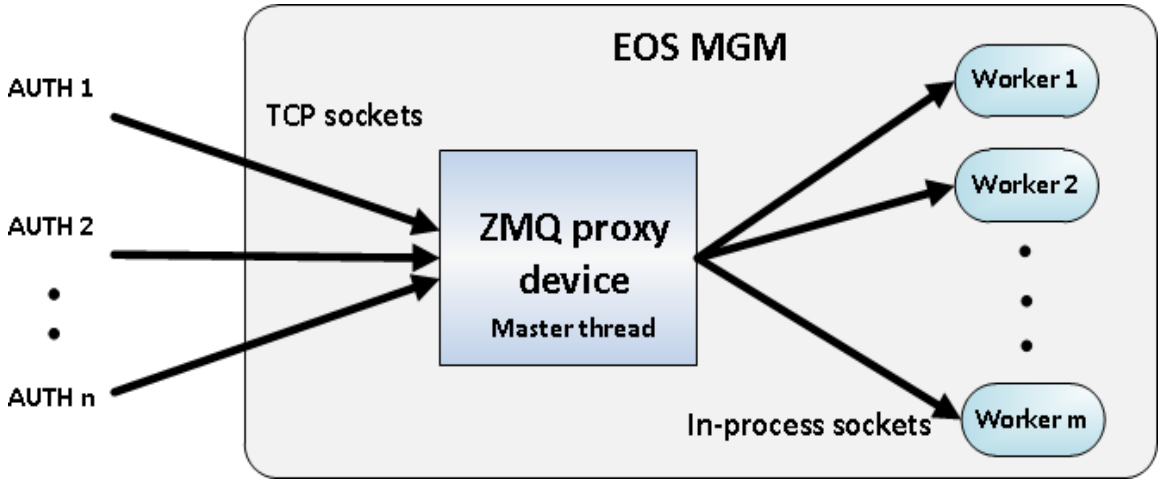
Auth. extension – plumbing

- In-process sockets are reused between client requests
- ZMQ **proxy device** forwards requests to current master



- Dealing with **stateful operations**:
  - **UUID** at the Auth instance: “IP_addres:object_ptr_value”
  - UUID sent along with the request message
  - MGM uses **mapping** from UUID to FS object (file/directory)

Auth extension – back-end



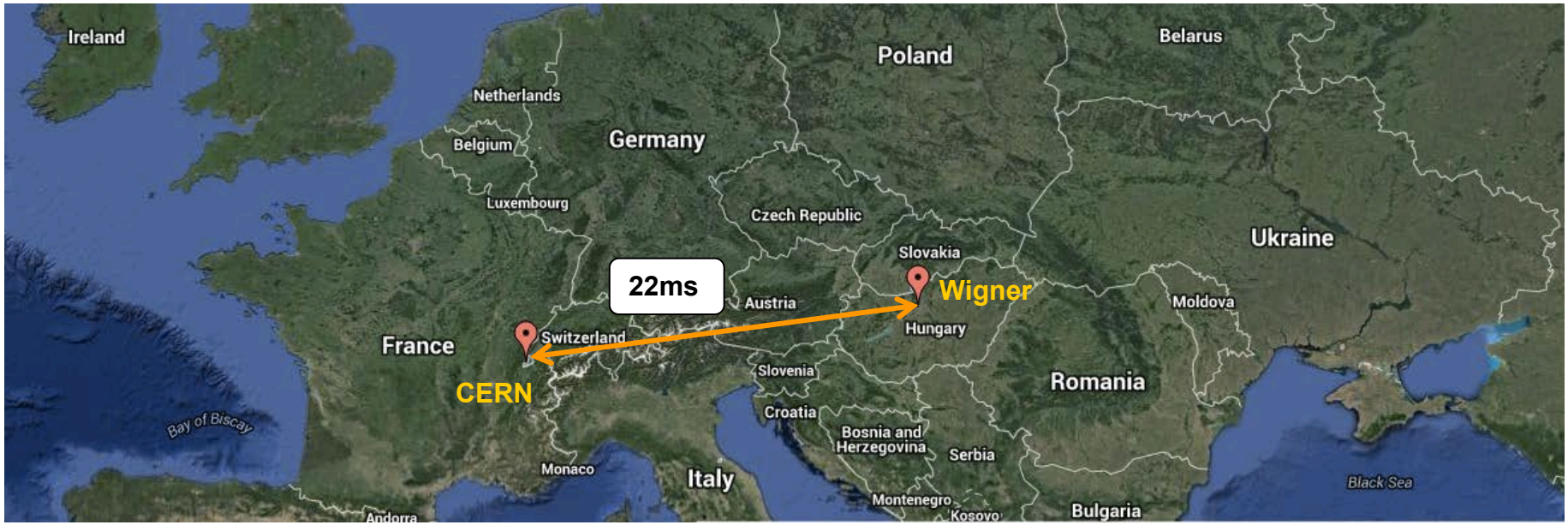
- Default **ZMQ proxy device** forwards requests to worker threads
- Communication pattern: **ROUTER -> DEALER -> REP**
- Single client from same LAN, doing 10k stats:

Operation	Avg. duration	Rate
Direct EOS stat	380 µs	2.6 KHz
EOS AUTH stat	600 µs	1.6 KHz

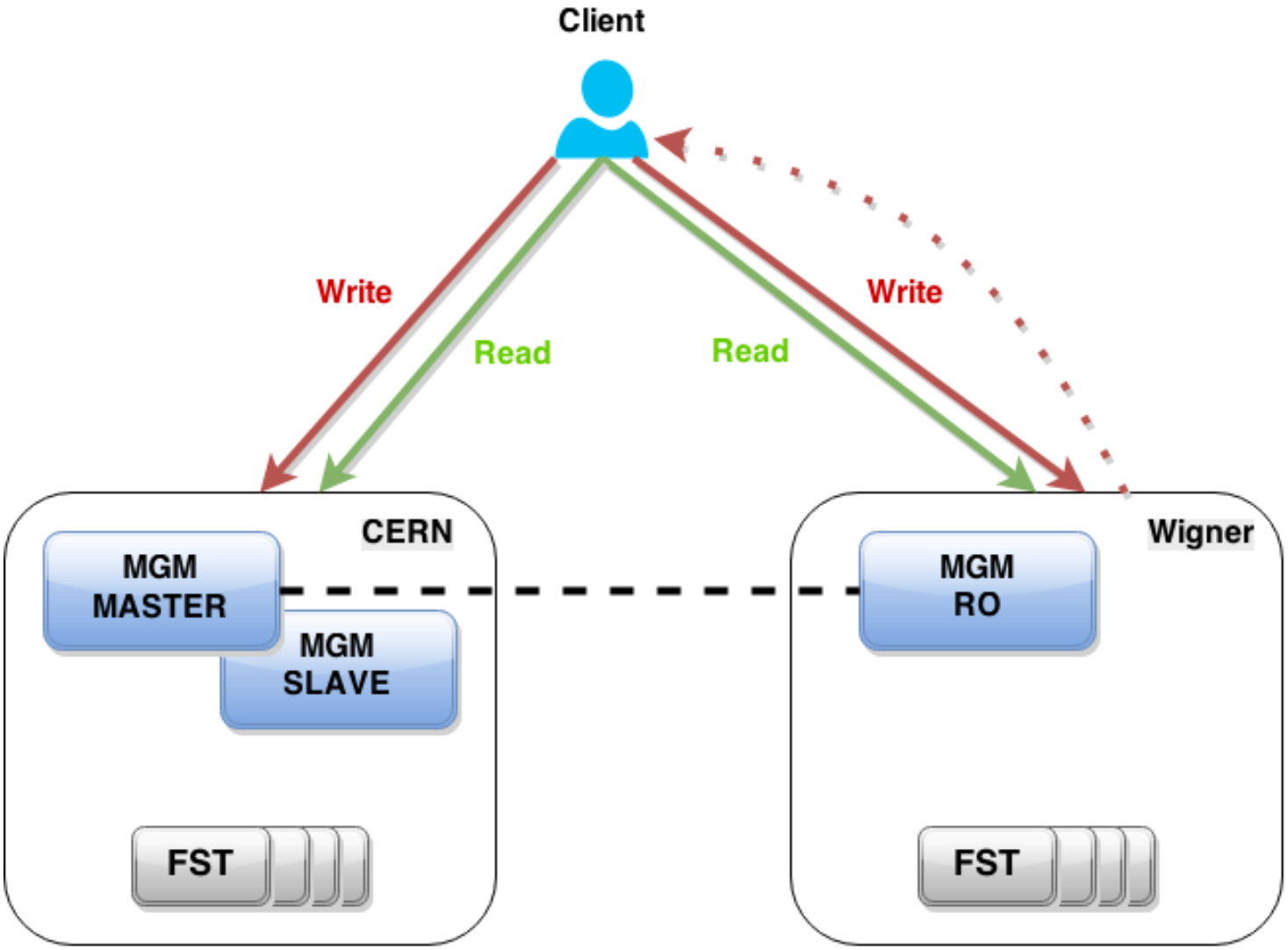
- Increased individual latency, but gained scalability

Wigner computer center (HU)

- New **remote CC in Hungary** which will be around **40%** the size of the CERN CC
- Disk servers at CERN & Wigner will be added to the same EOS instance

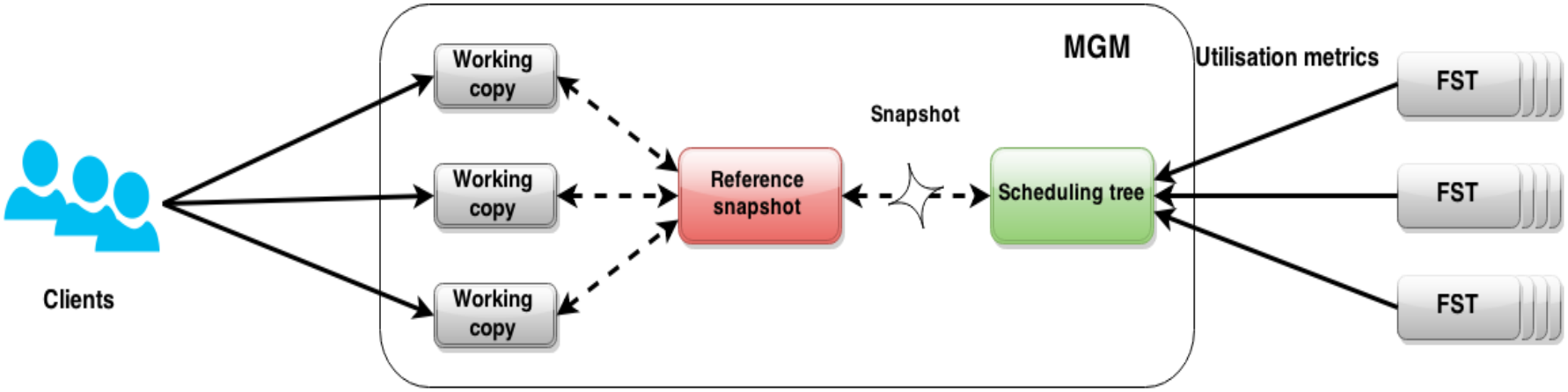


EOS CERN-Wigner deployment



EOS geo-scheduling

- **Goal:** Make placement and access operations **geotag-aware**
- Example of geotags:
  - **<ROOT>:site1:rack100**, **<ROOT>:site2:rack25** etc.
- New option for eos commands: **-g <n>**
  - Aggregates displayed information along the geotree down to depth <n>
- New **geosched** command shows the internal state and parameters of the GeoTree used for scheduling



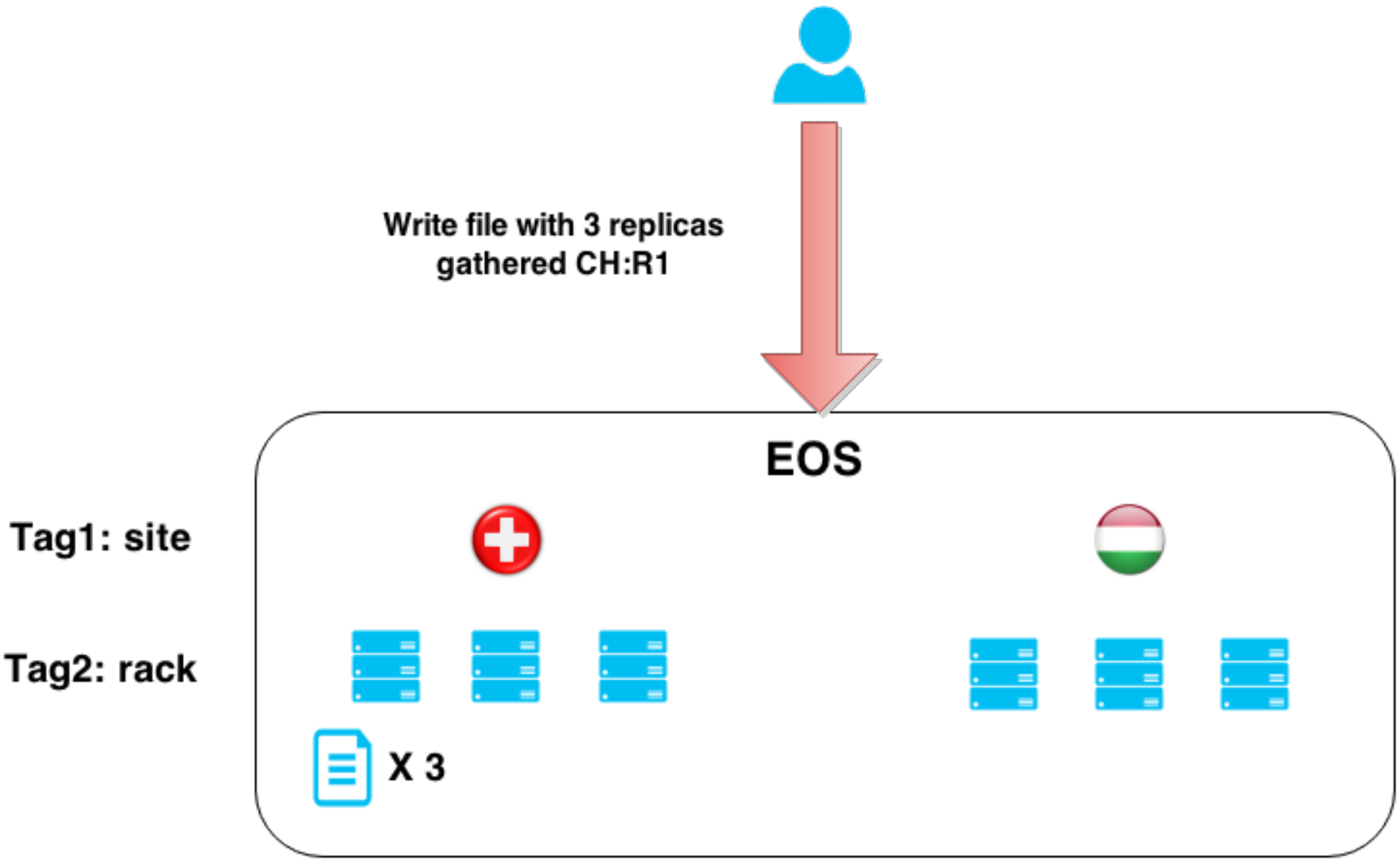
EOS placement policies

- Control the tradeoff between speed, availability and safety
- Can be set as an **extended attribute** at directory level:

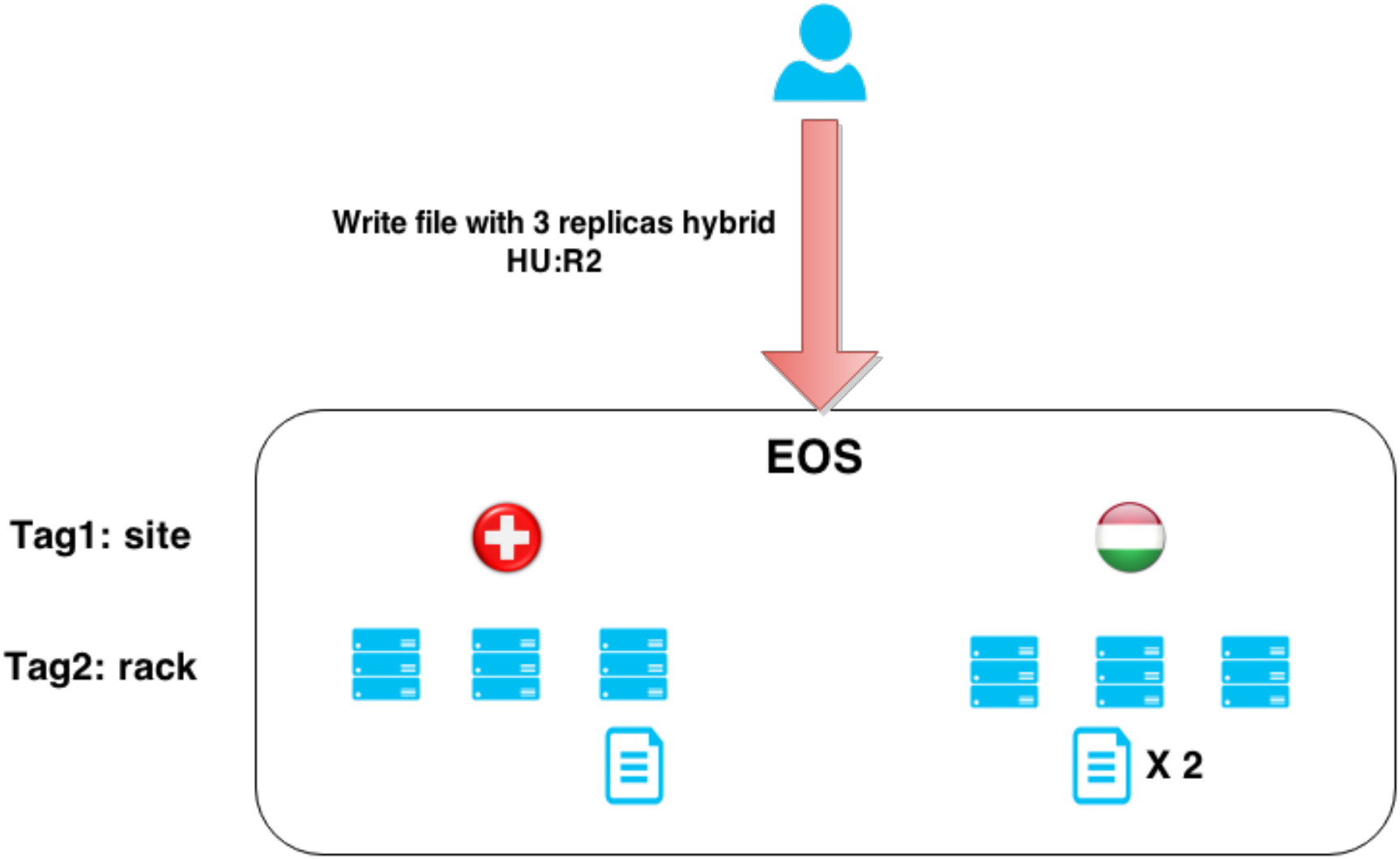
  sys.forced.plctply
- Three types of policies: **scattered, gathered and hybrid**

	Gathered tag1:tag2	Hybrid tag1:tag2	Scattered
Replica	All as close as possible to tag1:tag2	All - 1 around tag1:tag2, one as far as possible	All as scattered as possible
RAIN	All as close as possible to tag1:tag2	All – num_parity around tag1:tag2, num_parity as far as possible	All as scattered as possible

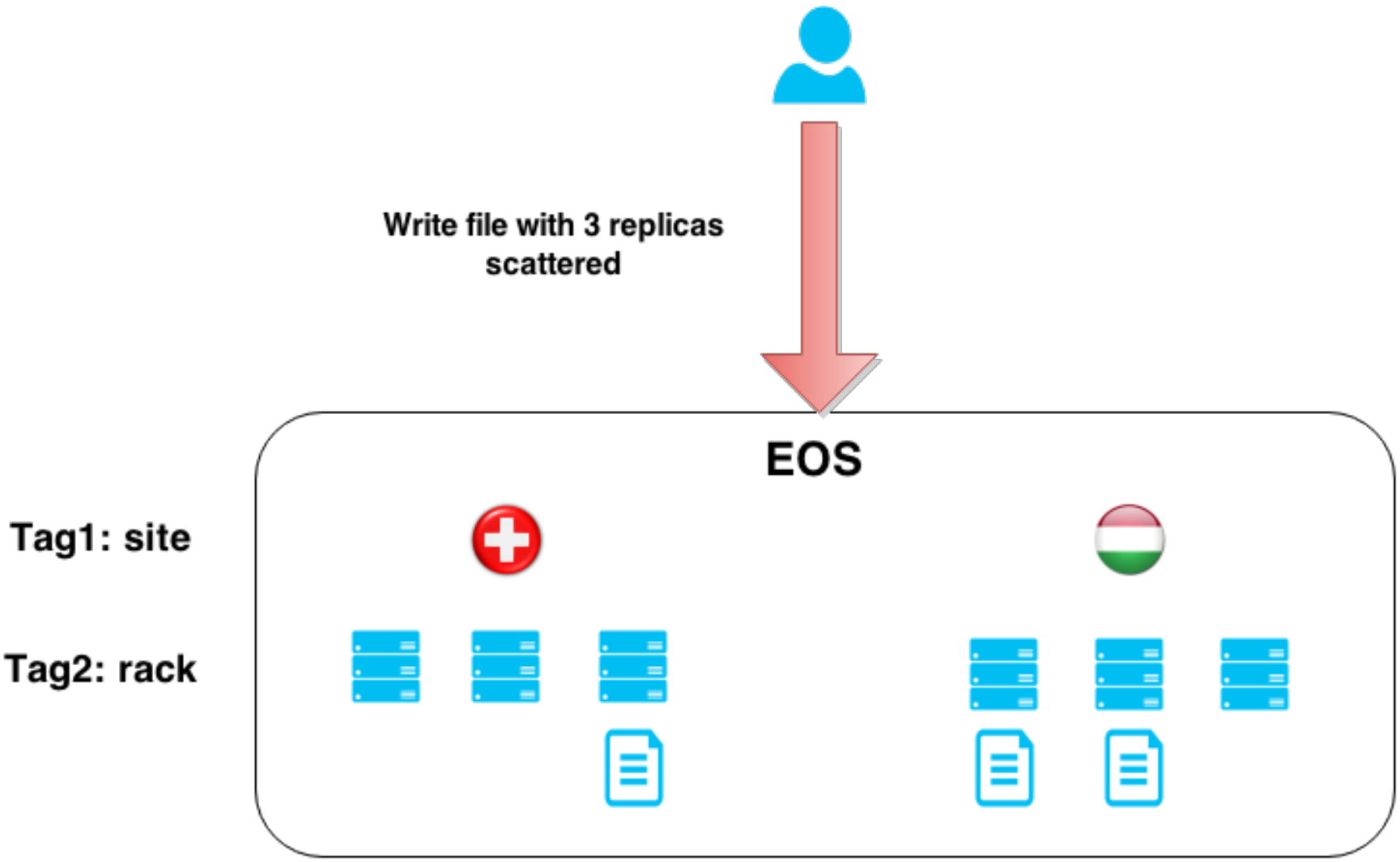
Placement policies – example(1)



Placement policies – example(2)

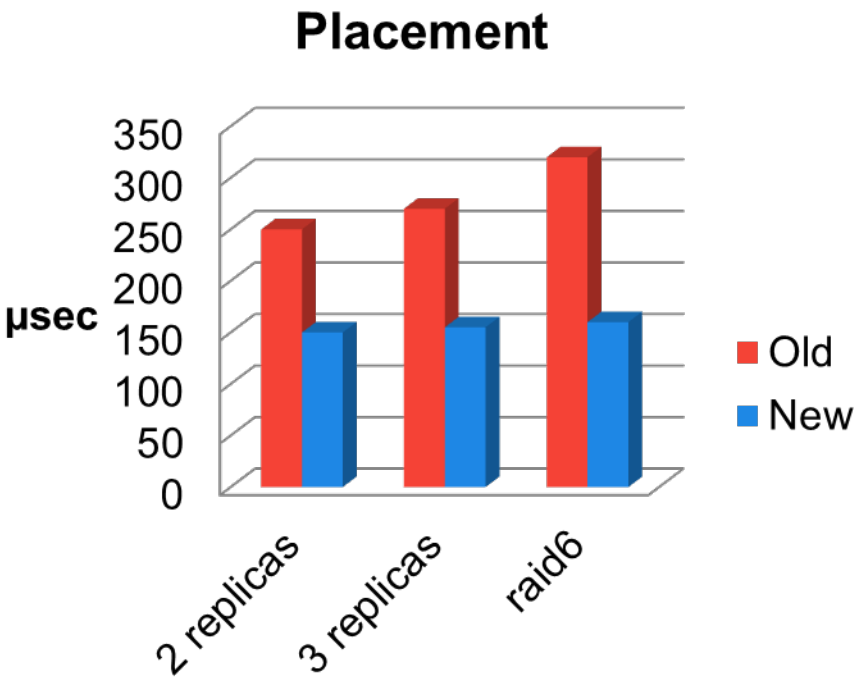


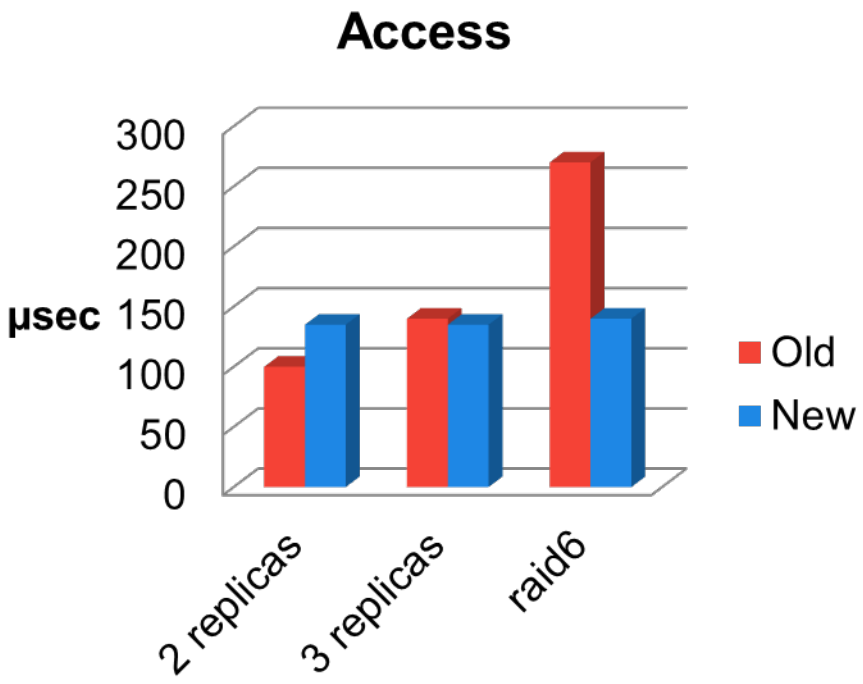
Placement policies – example(3)



EOS geo-scheduling – performance

- Impact on performance is overall **negligible**, giving better results in the majority of the use-cases





IT-DSS XRootD Auth Plug-In

CERN IT Department

xrootd-auth-change-id

- use XRootD like NFS server applying POSIX permissions & ACLs from locally mounted filesystem

- files are accessed and stored with the mapped identity of the XRootD client

- switches only filesystem ID of each XRootD thread

- works only on Linux

https://github.com/cern-eos/xrootd-auth-change-uid

CERN
IT Department

EOS - Diamond R&D - 1st generation

Information https://github.com/cern-eos/eos-diamond/wiki/1-Introduction

CERN IT Department

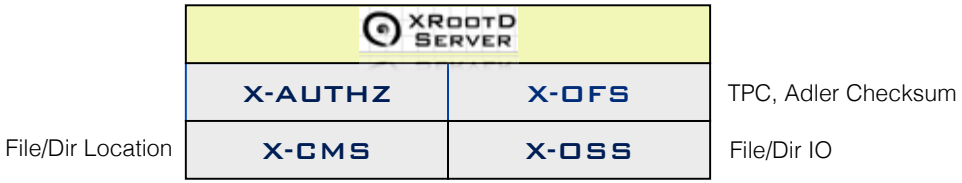
1st generation implementation

- files & POSIX namespace stored on CEPH
  - implemented by RadosFS

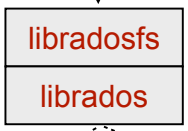
- POSIX permission model
- parallel IO (variable chunk size)
- support for EC pools (only seq. uploads)
- tested with XRootD & HTTP protocol

Diamond

XRootD



CERN IT-DSS R&D Project Joaquim Rocha



low-level File/Dir IO

- 112 file-store OSDs with 2 rep(data) 3 rep(meta)
Test 1 created **85 million** files in 4 days (250 Hz) using 320 ROOT clients creating 16k files - IOPS bound - no failures

Test 2 wrote 1.2 GB/s using 320 ROOT clients with 16M files until OSDs were full - no failures

OSDs in Data Pools

OSDs in Metadata Pools

https://github.com/cern-eos/eos-diamond

EOS - Diamond R&D - 2nd generation

CERN IT Department

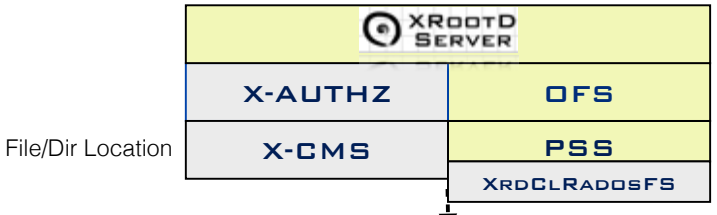
2nd generation implementation
- use standard OFS & PSS
- bridge to CEPH done with XrdCl plug-in

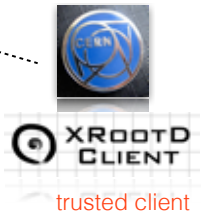
XrdClRadosFS

- allows authenticated access via XRootd Server
- allows direct parallel IO without proxy from trusted client (batch nodes) using directly XrdClRadosFS

DRAFT

XRootD





OSDs in Data Pools

OSDs in Metadata Pools

https://github.com/cern-eos/XrdClRadosFS

EOS
Possible Change in Access Model @ CERN during 2015

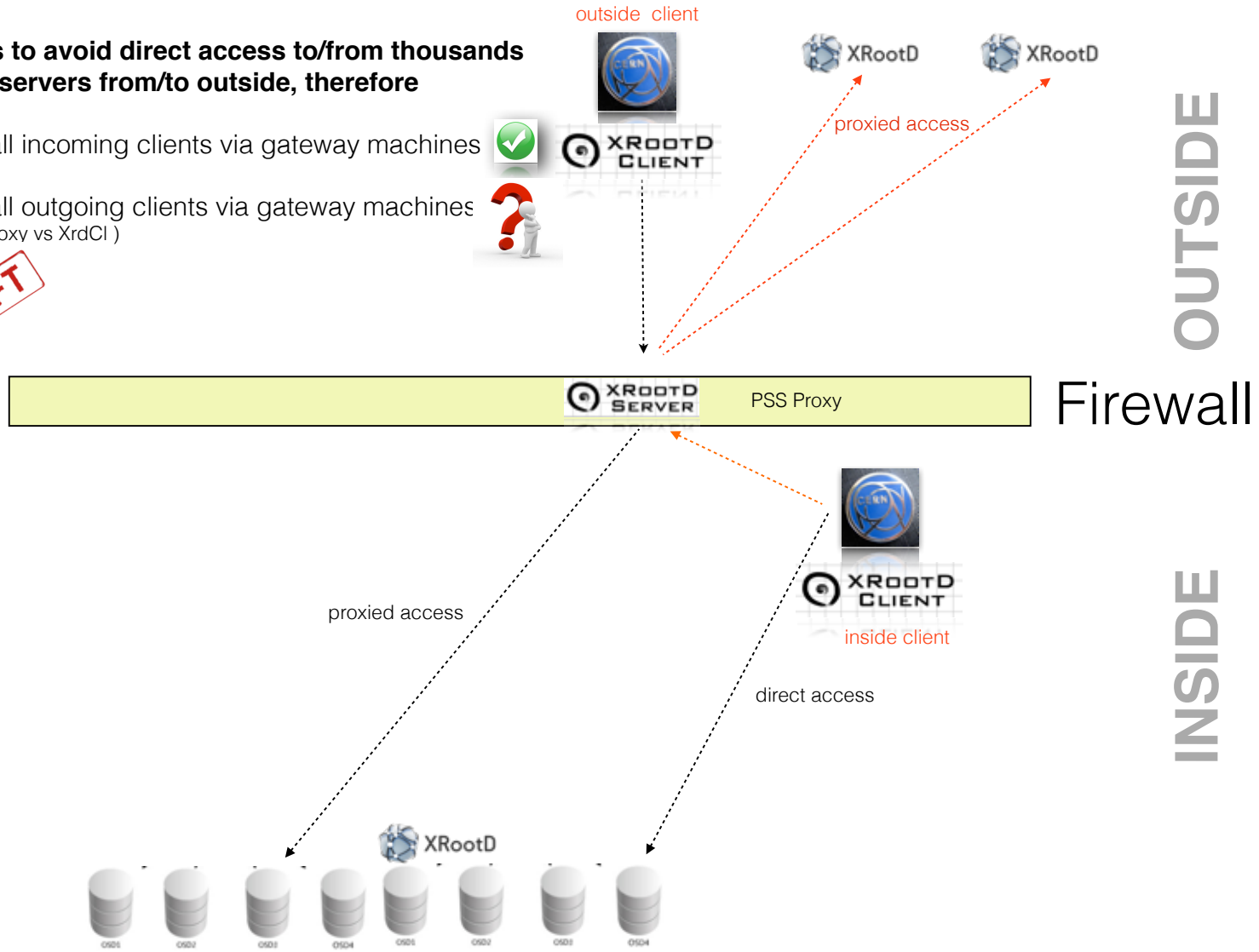
CERN IT Department

IT wants to avoid direct access to/from thousands of disk servers from/to outside, therefore

- proxy all incoming clients via gateway machines

- proxy all outgoing clients via gateway machines
(socks4 proxy vs XrdCl)

DRAFT



Summary

EOS - CERN Disk Storage System > 100 PB

- Offers archive functionality to save data on tape
- Brings IO improvements by supporting vector reads and extending this functionality to RAIN layouts
- Addresses scalability by using authentication delegation
- Capitalizes on the new CC in Wigner and makes data locality transparent to the user with the help of geo-scheduling