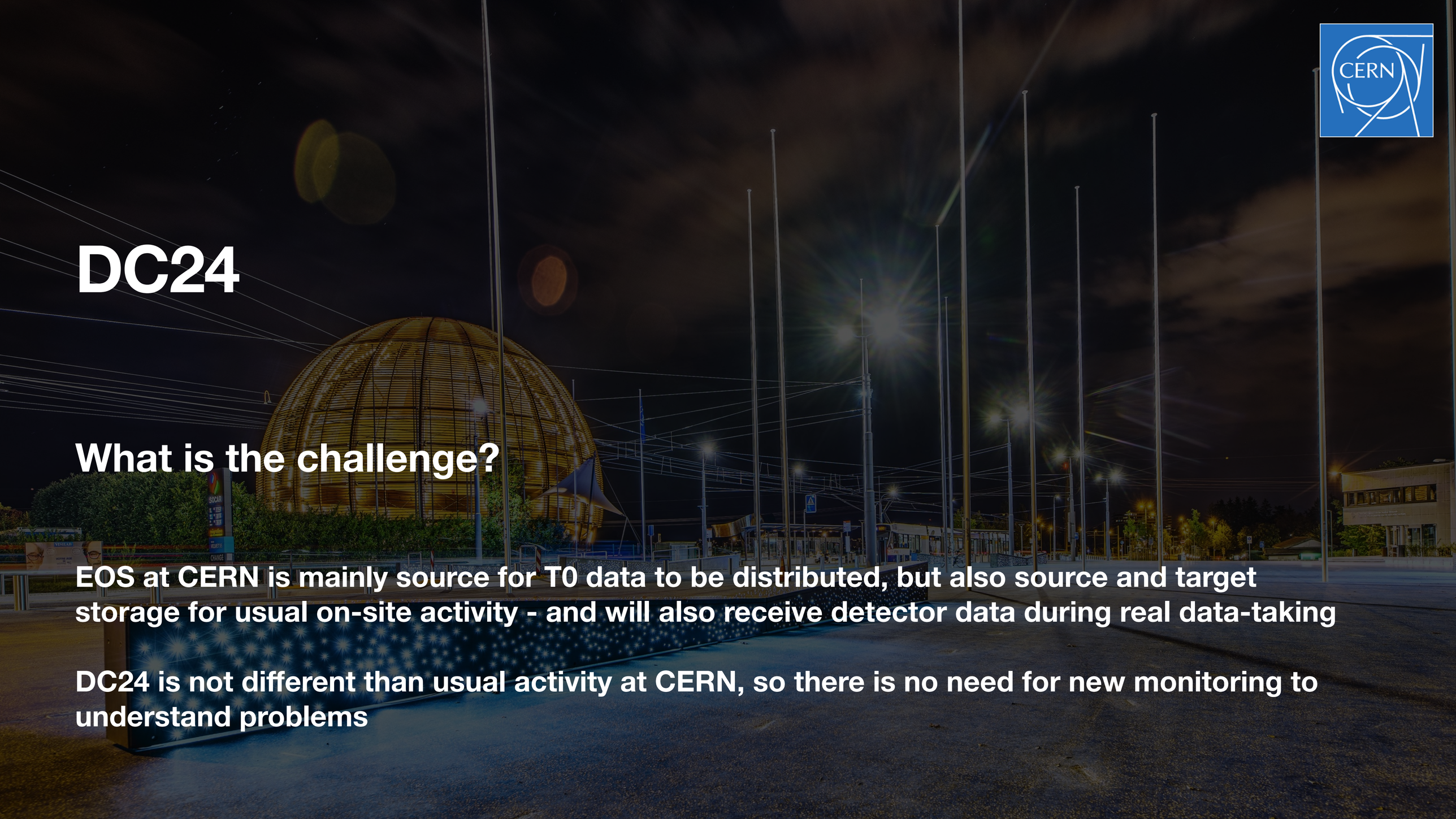

CERN
Accélérateur de science

DC24 Status EOS

DC24 Data Challenge 2024 Workshop 9-10.November 2023

Dr. Andreas-Joachim Peters for the EOS project



DC24

What is the challenge?

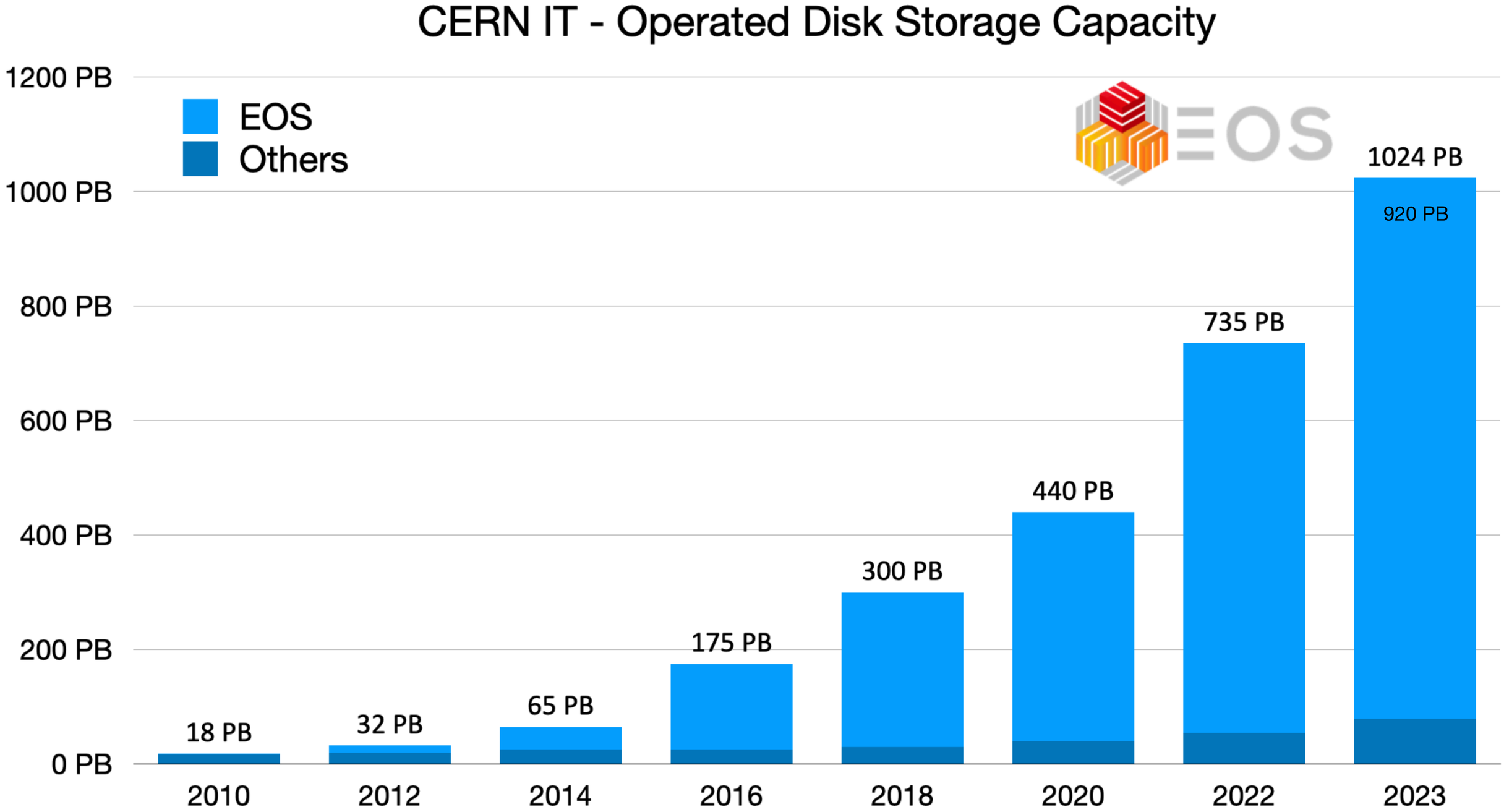
EOS at CERN is mainly source for T0 data to be distributed, but also source and target storage for usual on-site activity - and will also receive detector data during real data-taking

DC24 is not different than usual activity at CERN, so there is no need for new monitoring to understand problems

EOS Disk Service for Physics

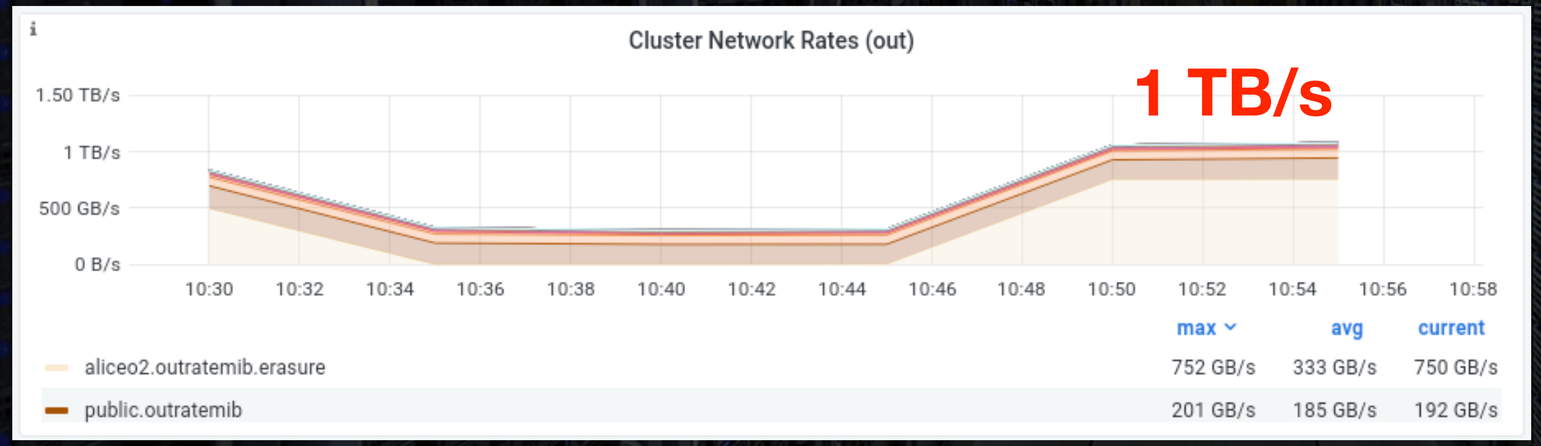
eos.web.cern.ch

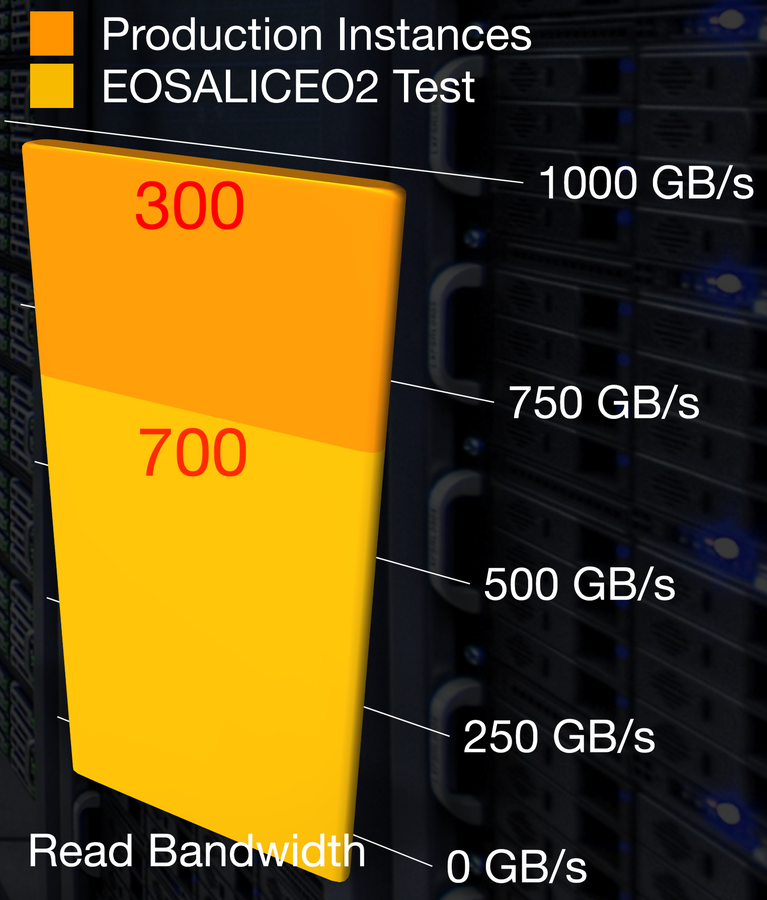
EOS


CERN IT - Operated Disk Storage Capacity
EOS
Others
EOS
1200 PB
1000 PB
800 PB
600 PB
400 PB
200 PB
0 PB
18 PB
32 PB
65 PB
175 PB
300 PB
440 PB
735 PB
1024 PB
920 PB
2010
2012
2014
2016
2018
2020
2022
2023

CERN EOS as Data Source for DC24

Artificial Read Test on EOSALICEO2 Summer 2023



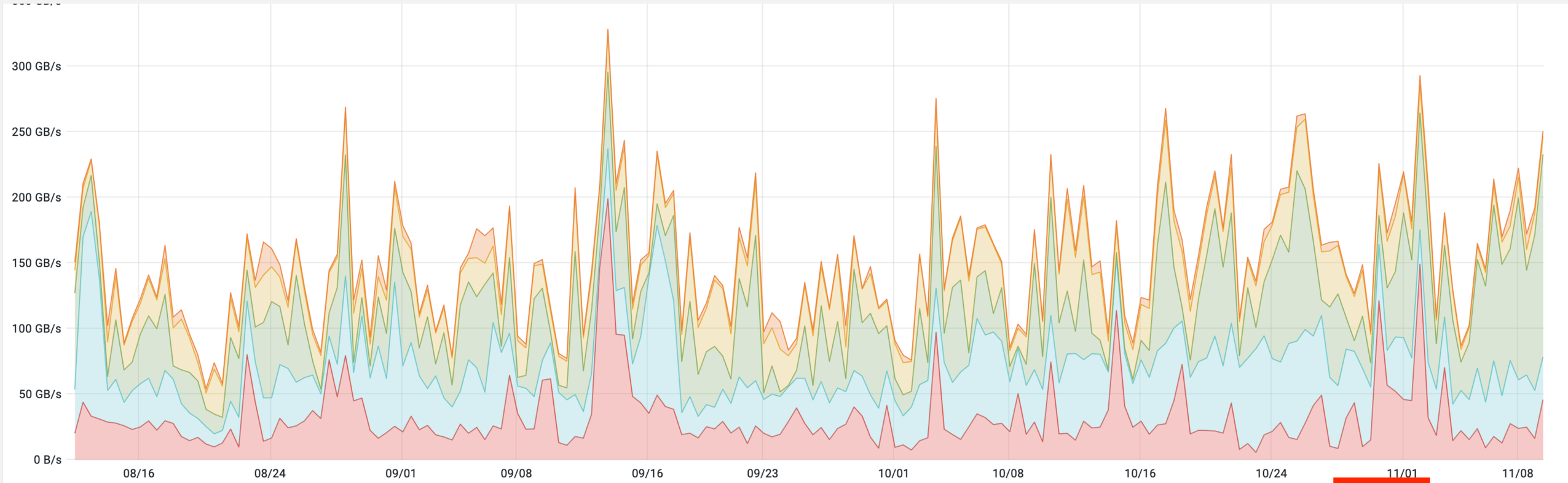


EOS Disk should not be a bandwidth bottleneck!

Avg. file transaction rates 100-500Hz (OpenRead) per instance are normal (production peak 4kHz)

CERN EOS as Data Source for DC24

- Hardware in LHC instances ages back 0-10 years
  - this won't be the majority of hardware running in Run-4 (hopefully) - aiming for O2 performance for other LHC instances



	max	avg	current
public.outratemib	199 GB/s	31.9 GB/s	45.7 GB/s
cms.outratemib	156 GB/s	42.4 GB/s	32.6 GB/s
alice.outratemib	155 GB/s	44.4 GB/s	155 GB/s
atlas.outratemib	70.1 GB/s	28.2 GB/s	13.8 GB/s
lhcb.outratemib	25.3 GB/s	5.02 GB/s	3.84 GB/s

Peak Read rates from production instances

Internal EOS Monitoring

Meta-Data and IO prioritisation

- we have a wide range of handles to modify meta-data rates by user, group or application tags - low-impact DC24 no MD challange
- we have approximate handles to limit max stream bandwidth by user, group or application tags, have IO priorities on each disk, direct IO etc …
- by DC24 we will have a new global **io-limit** interface in place - high-impact DC24 is about bandwidth
- biggest challenge for us is probably to guarantee DC24 share against usual activities of ‘others’ …
  - this cannot be solved by network configuration because many resources share the same infrastructure at CERN

Internal EOS Monitoring

File Creation/Update/Deletion Reports

File Creation/Update Reports allow to distinguish missing activity from slow IO and to compute **IO efficiency** per file access.

ot	time spent in ms to open the file
ct	time spent in ms to close a file (includes waiting for async writes and checksumming)
rt	time spent in ms waiting for disk reads
rvt	time spent in ms waiting for disk reads for vector reads
wt	time spent in ms waiting for disk writes
lrt	time spent in ms waiting for layout reads
lrvt	time spent in ms waiting for layout vector reads
lwt	time spent in ms waiting for layout writes
iot	time spent in total from open to close
idt	idle time from open to close (where no open, close, read,readv or write happens)

In total there are over 70 parameters reported per open/close sequence

Deletion Reports

TAG	Description
log	uuid to correlate log entries
host	FST host name
fid	file id of the file deleted
fsid	filesystem id where the file is deleted
del_ts	timestamp when the deletion message was generated
del_tns	timestamp in ns when the deletion message was generated
dc_ts	change timestamp of the deleted file
dc_tns	change timestamp in ns of the deleted file
dm_ts	modification timestamp of the deleted file
dm_tns	modification timestamp in ns of the deleted file
da_ts	access timestamp on local disk of the deleted file
da_tns	access timestamp on local disk in ns of the deleted file
dsize	size of the file before deletion
sec.app	always: deletion

Internal EOS Monitoring

Meta-Data and IO prioritisation

io-limit allows to **regulate bandwidths** consumed by user/group or applications **almost in real-time** with a 10s feedback-loop

One knob per instance …

DC24 Priority

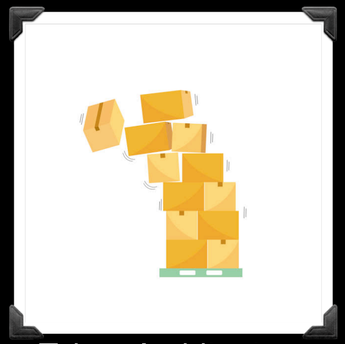
type	id	key	range	current	limit	scaler
app	atlasexport	rbytes	1min	7.46 GB/s	10.00 GB/s	1.00
app	atlast0	rbytes	1min	71.09 GB/s	80.00 GB/s	1.00
app	atlast0	wbytes	1min	17.77 GB/s	20.00 GB/s	1.00

XRootD Monitoring “Old” & “New”

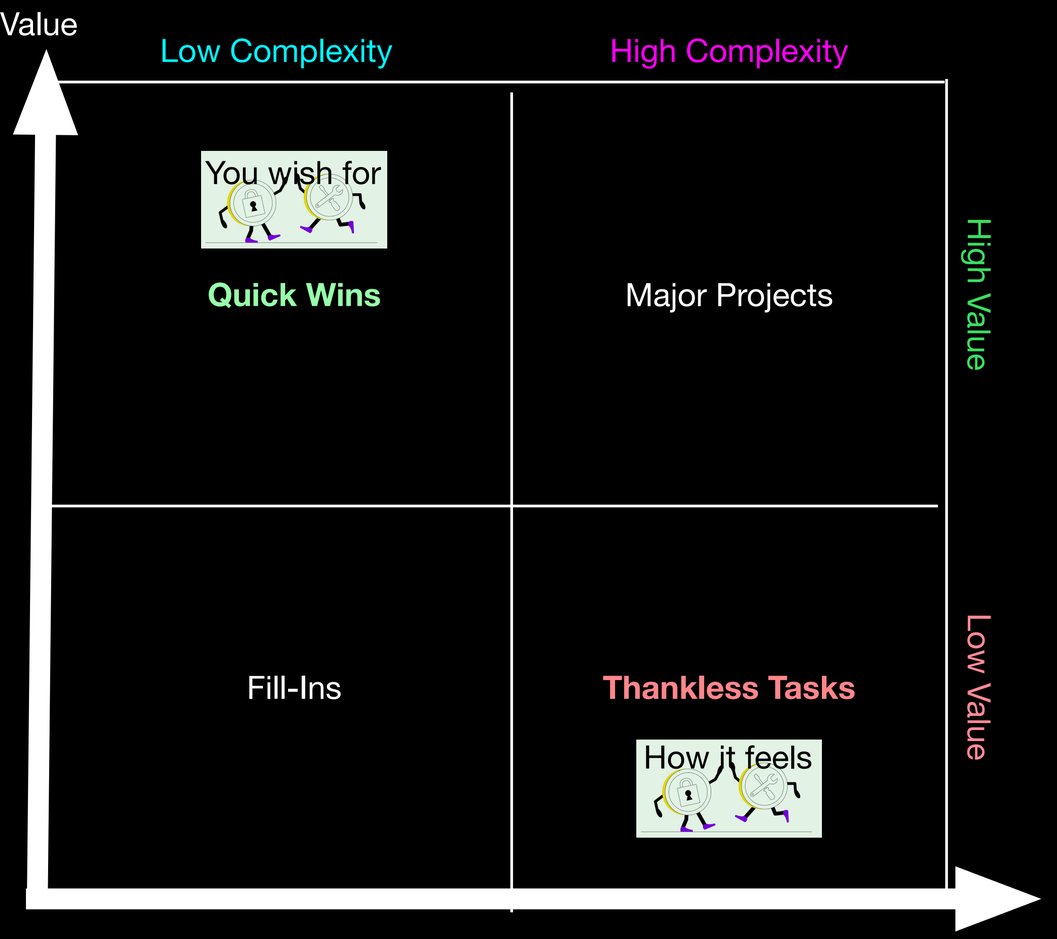
ATLAS	CMS	ALICE	LHCB	PUBLIC
atlas-xrdmon-collector.cern.ch:9331 wlcg-xrootd-shoveler-atlas.cern.ch:9994	cms-xrdmon-collector.cern.ch:9331 wlcg-xrootd-shoveler-cms.cern.ch:9995	aliendb2.cern.ch:9930 wlcg-xrootd-shoveler-alice.cern.ch:9993	lhcb-xrdmon-collector.cern.ch:9330 wlcg-xrootd-shoveler-lhcb.cern.ch:9996	not configured

Tokens for DC24

are we there yet?



Token Architecture



Mail Thread 2017 about Token TPC

Andreas-Joachim Peters <andreas.joachim.peters@cern.ch> Jul 21, 2017, 10:20 AM

to Alejandro, Mario, Brian, Fabrizio, Wei, Andrew, Doug, Hiro, Kaushik, Vincent, Oliver, Cedric

Hi all,

this how ALICE since 2003 authorizes access to XrootD storage (via ALICE tokens issued centrally in this case). The URI mechanism is protocol independent, which is great point! For the HTTP case the proposed model is a very good starting point!

Maybe we should come up with a description of a 'default' token. One could consider this industry standard here:

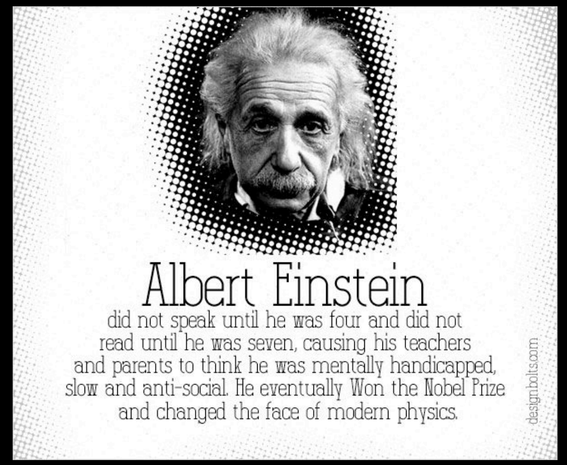
https://jwt.io/

The mechanism should not be stateful on the receiving end, the token should be self describing and might restrict the usage by time and network partition etc ... in the case of AWS this is already there.

With this mechanism you can implement 3rd party pull, push and bridging and I think that would be a very good step forward!

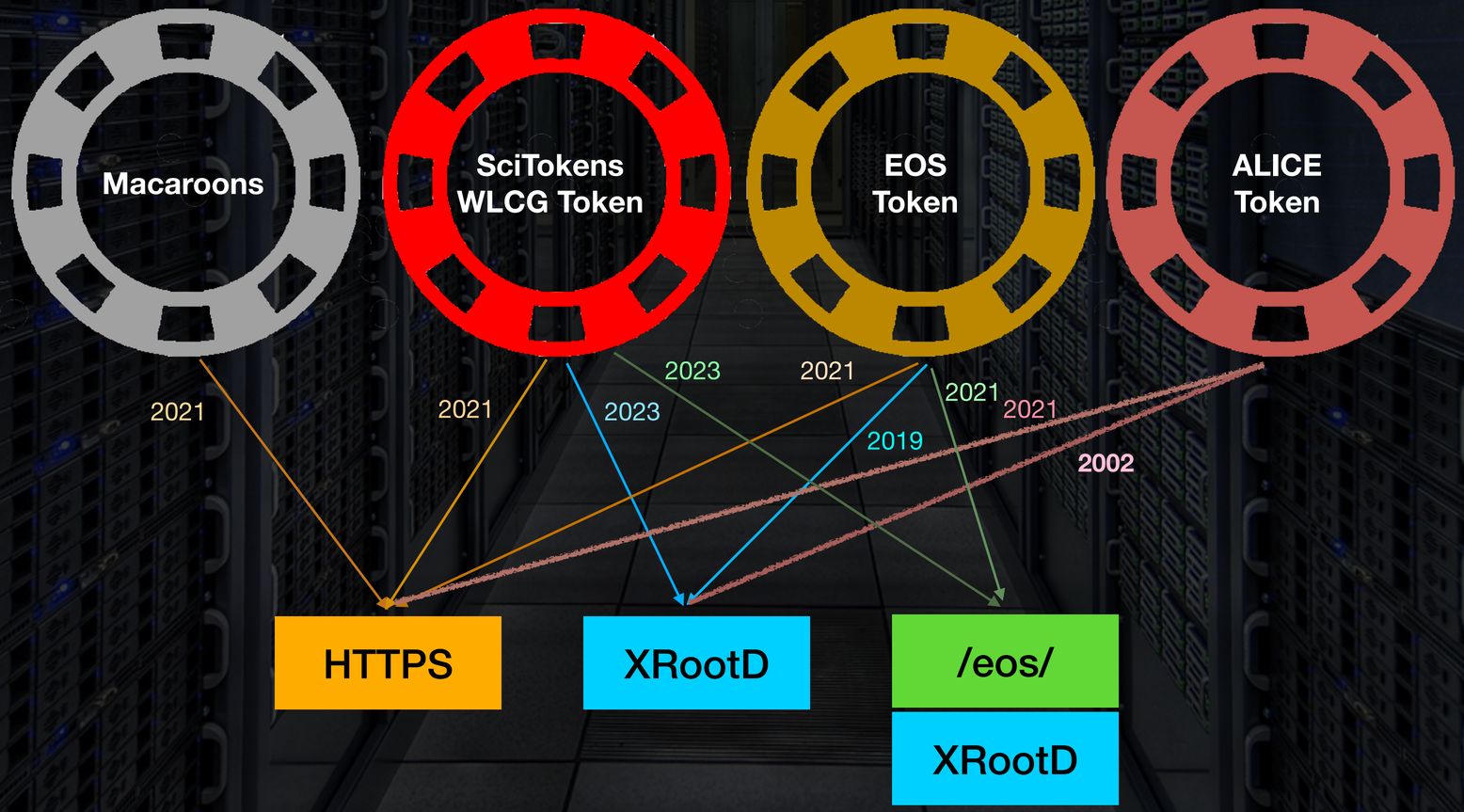
Cheers Andreas.

1st Token FTS Transfer after 6 years



Token Flavours & Protocols

EOS Support (CERN Deployment)



Token Support for DC24

Current WLCG Token Issuer configuration in EOS instances at CERN

ATLAS	CMS	LHCB	PUBLIC	ALICE
OSG-Connect dteam001	OSG-Connect dteam001	OSG-Connect dteam001	OSG-Connect dteam001	ALICE Token aliprod
IAM ATLAS atlas001	IAM CMS cmsprod		CILogon	

Token Mapfiles for ATLAS/CMS

```
[
  {"path": "/eos/atlas/atlasscratchdisk", "result": "atlas001", "comment": "Owner of the ATLASSCRATCHDISK area"},
  {"path": "/eos/atlas/atlasdatadisk", "result": "atlas003", "comment": "Owner of the ATLASDATADISK area"}
]
```

```
[{"path": "/eos/cms/store", "result": "cmsprod", "comment": "Owner of the EOSCMS store data"}]
```

General Token Issues

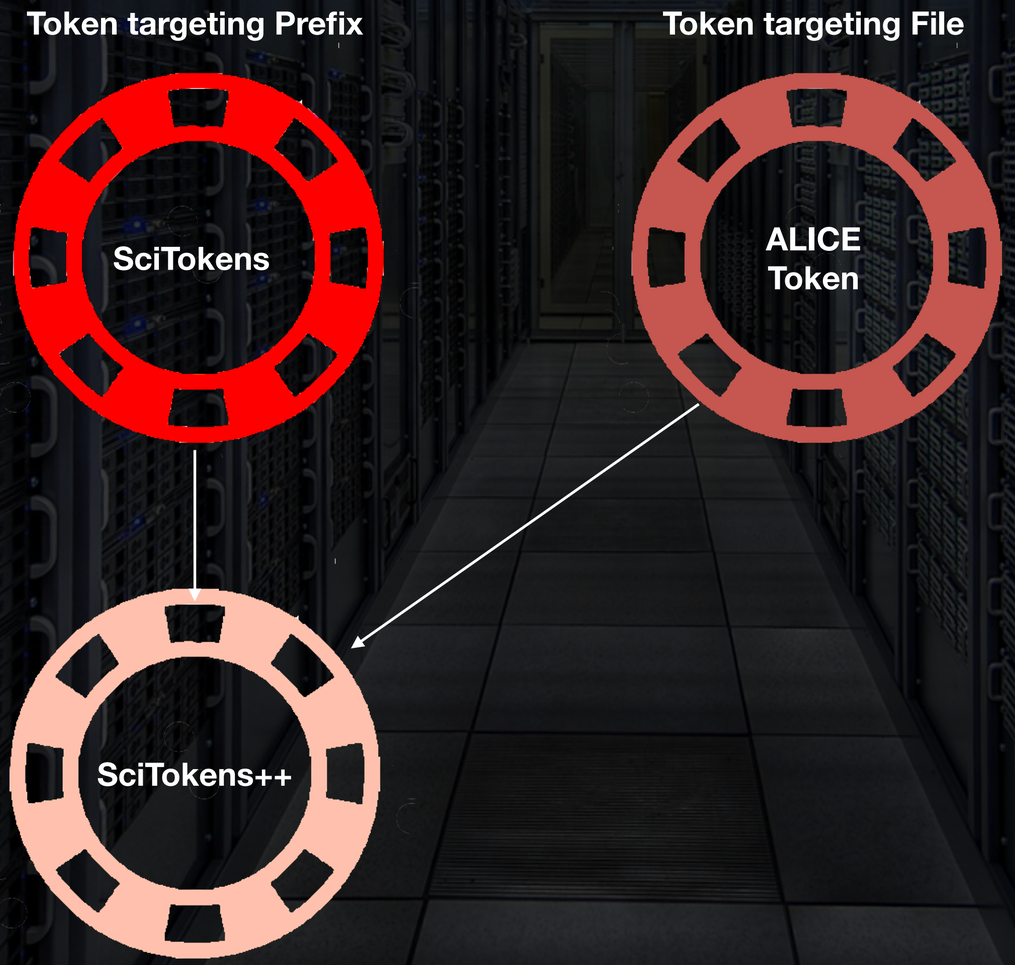
- mixture of **ZTN** and **ALICE Token** in the same instance currently **not working** because **ZTN** enforces **TLS** and ALICE clients do not have CAs configured
  - problem in Vienna site
  - additionally there are old software versions which don't support tokens/TLS and they have to continue to work
  - requires **XRootD** server enhancements since old clients cannot be changed

- **XRootD** protocol has many more API calls than **HTTPS** and we will support all of them with token authorisation only in upcoming 5.2 EOS releases
  - Token support with Tape REST API missing - no specification
  - Same for generic API calls - use common sense/pragmatism to map them

- We have never seen a production benchmark for file **open/s** using WLCG tokens via HTTPS or XRootD (e.g. we had trouble with TLS in general)
  - while we have a good understanding/experience of bandwidth bottlenecks and resource competition within experiments

Can we converge token libraries?

Co-existince is complicating deployment & implementation

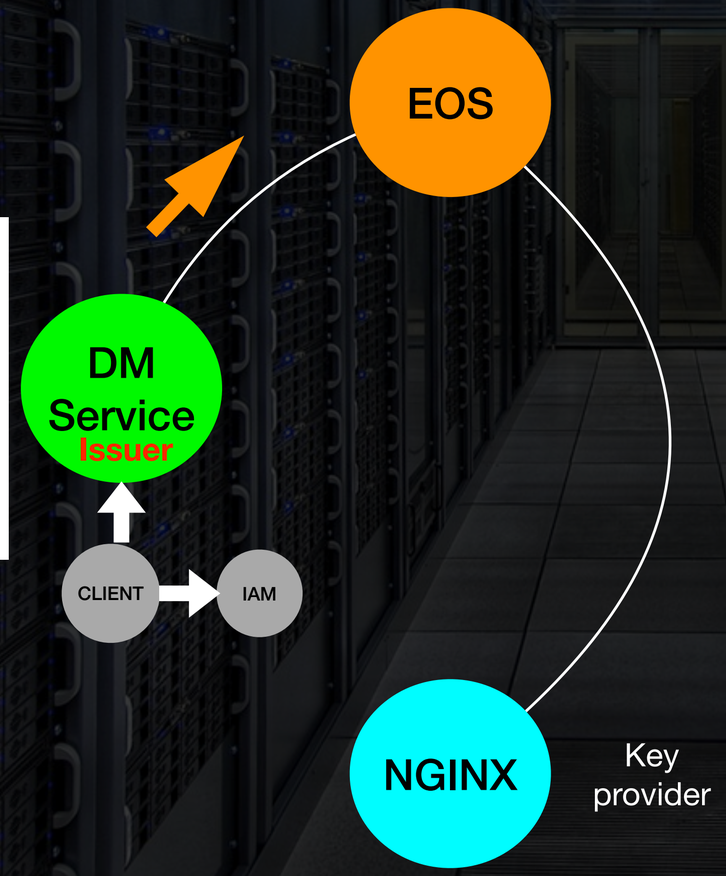


ALICE plug-in is low-maintance & issue free
- Token format unchanged since 21 years
- but we did
  - Two OpenSSL API Changes
  - Scalability Improvements
    Multithread->Multiprocess

“OpenSSL API changes are painful.“

A half day of **Token R&D** Token=signed URL

Example: SciTokens for ALICE




```
/etc/xrootd/scitokens.cfg:

[Issuer engine]
issuer = https://engine.cern.ch/
base_path = /
map_subject = False
default_user = alice
```

```
{
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "exp": 1692865940,
  "iat": 1692865340,
  "iss": "https://engine.cern.ch/",
  "jti": "9eb170b6-e3c9-41f8-9b31-6807e43bac57",
  "nbf": 1692865340,
  "scope": "storage.read:/eos/alice/file",
  "sub": "aliprod",
  "wlcg.ver": "1.0"
}
```

SciToken CLI creates token with kHZ

- no IAM dependency
- no IAM bottleneck-coupling
- no token lifecycle needed
- simple

```
[root@engine.cern.ch# cat /usr/share/nginx/html/jwk
{"keys":[{
    "kty": "EC",
    "use": "sig",
    "crv": "P-256",
    "kid": "ALICE",
    "x": "OknmzRxVuW3-Im6FBswnf3JsVWoNEgjOIGC6RKhreXY",
    "y": "euy6nuOZKjEDUPxRQDMUOlgeu26jgnx5QXHhGIAEvaU",
    "alg": "ES256"
}]}
```

```
root@engine.cern.ch]# cat /usr/share/nginx/html/.well-known/openid-configuration
{"jwks_uri":"https://engine.cern.ch/jwk"}
```

A half day of **Token R&D** Token=signed URL

- if the IAM model does not scale or is identified as an unfortunate architectural choice,
  data management services (RUCIO) / file cataloges (ALICE/LHCB) might issue JWT token for individual files
  without any IAM involvement since users are authenticated already to these services,
  these services are trusted and they are actually the source of ACLs (authorization) or the ACL
  arrives embedded in the IAM token of the client

  - this improves security since the scope is reduced to the minimum
    (single file) - there are no tokens circulating which can read/write or delete **every** GRID file

  - there is no need for complicated token exchange and refresh workflows (no token lifecycle management)

- model works out of the box by configuring a new token issuer in storage targets
  and provisioning public key service on service side (subdir creation policy to be checked in XRootd/DCache)

- possible improvement in SciToken library
  - library does only know prefixes, cannot scope to a single object (file/container)
    - the scope /cms/grid/ allows to create
      `/cms/grid/higgs.root`
      `/cms/grid/dir1/higgs.root/higgs.root`
      `/cms/gridhiggs.root aso …`

Summary & Outlook

- we didn’t identify any particular obstacles with EOS@CERN for DC24 or external installations
  - but work to do be done to homogenise tokens with all protocols in multi-VO setups and keep backward compatibility with ‘old clients’ (ZTN vs ALICETK)

- we will try to use the new **io-limit** interface during DC24 and make sure services are upgraded to the required version at CERN-T0

- propose to run a 1-byte file transfer challenge in 2024 to see bottlenecks nobody likes to talk about

- interested to contribute small improvements to SciToken library to support signed URL model & phase-out ALICE token library

- if **IAM** token approach fails for **DM**, there is a low-hanging fruit as fall-back architecture which is possibly simpler, more robust, more secure and proven to work well since 21 years in production for one LHC experiment

Thanks for your attention!

EOS