New experimental XRootD plug-ins & R&D

Dr. Andreas-Joachim Peters - CERN IT-SD-PSS

XROOTD WORKSHOP LYON 2026

A short overview of four experimental plugins and two R&D projects. Each topic has a dedicated presentation with full detail; these are attached but not presented here.

SECURITY

1 XrdSecoidc

2 XrdSecssh

OFS ACCESS

3 XrdHttp

4 XrdClJournalCache

PROTOCOL & BACK-END R&D

5 cern-nfs

6 aios

Authentication vs authorization

WHAT IT ADDRESSES

These plugins cover two common ways users authenticate today: OIDC/IAM tokens and SSH keys. They authenticate the client and set `XrdSecEntity`. They do not authorize paths; that remains with `ofs.authlib`. Both require TLS. OIDC works on `root://` and `https://`; SSH only on `root://`.

	XRDSECOIDC	XRDSECSSH
Identity	Bearer JWT validated against the issuer JWKS	SSH public key or OpenSSH user certificate
Secret on the wire	The token, over TLS	Signature of nonce + host + fingerprint
Protocols	`root://` and HTTPS	`root://` only
Access permissions	Later: Acc (e.g. `XrdAccToken`)	Later: the usual Acc / `acc` stack

TAKE-AWAY Both plugins authenticate only. Acc authorizes paths. TLS is required.

OIDC compared to ZTN / SciTokens

WHAT IT ADDRESSES

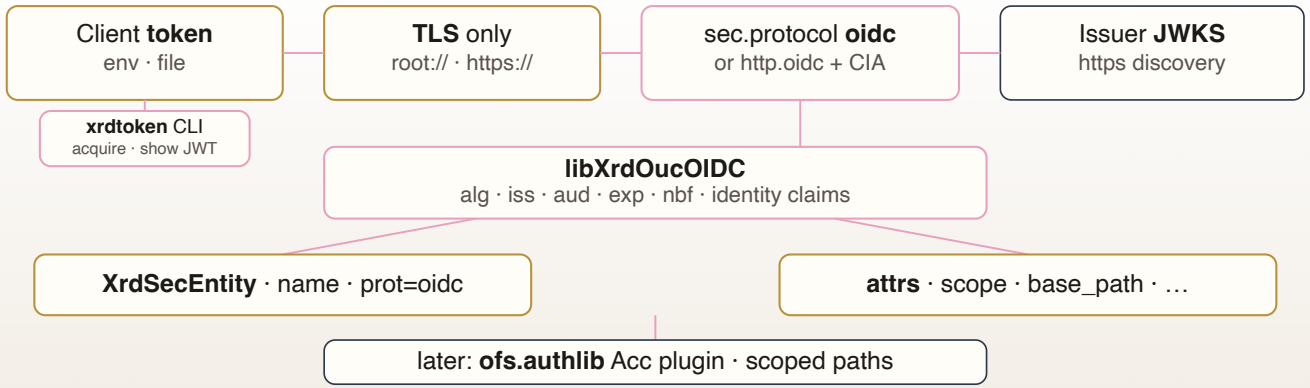
Lets a user log in to XRootD with a token from CERN SSO or a VO IAM, on both `root://` and HTTPS.

	ZTN + SCITOKENS	THIS STACK
Identity	SciTokens helper inside `ztn`	`sec.protocol oidc`
Access permissions	Same AccSciTokens library	`ofs.authlib` Acc plugin
HTTPS	No ZTN login in XrdHttp	`http.oidc` + CIA → the same plugin
Scopes in the JWT	Consumed inside SciTokens	Copied onto `XrdSecEntity` only

TAKE-AWAY This plugin could replace ZTN and SciTokens.

Architecture: root:// and HTTPS

Validation is local after JWKS is fetched. There is no token-introspection call on the hot path.



TAKE-AWAY One JWKS validator for both protocols. No introspection on the hot path.

Issuers and identity mapping

Claims become `XrdSecEntity.name` and attributes. The plugin does not interpret `storage.read:/alice`; that is the role of Acc.

NAME

Default claim order

1. `preferred_username`
2. `upn`
3. `username`
4. `name`
5. `sub`

`-identity-claim` replaces the list. `-forced-identity-claim` is per issuer. Forced `email` requires `[email-map]`.

ATTRIBUTES

Exported for Acc

- `-entity-claim scope` → `token.scope`
- JSON arrays become spaces
- `wlcg.ver` and other safe claim names
- Per-issuer `base_path` (one absolute path)
- Repeatable `restricted_path` as a JSON array

IMPLEMENTED

- `sec.protocol oidc` + HTTPS via `http.oidc`
- 65 unit tests, packaging

NOT IMPLEMENTED

- Revocation, EdDSA

	CERN SSO (KEYCLOAK)	WLCG IAM
Role	Login / identity	VO-scoped storage tokens
Typical claims	`preferred_username`, `email`, `sub`	`scope`, `wlcg.ver`, `sub`
Path authz	No `storage.*` scopes	Yes, with `XrdAccToken`

TAKE-AWAY Secoidc maps claims only. Acc authorizes paths. IAM tokens may carry storage scopes; CERN SSO does not.

SSH-key authentication over TLS

WHAT IT ADDRESSES

Lets a user log in to XRootD with an SSH key or user certificate, over TLS, without a password.

AUTHN

Challenge / response

The server sends a nonce. The client signs it and sends the response. Sets `XrdSecEntity.name`. Makes no path decision.

TRUST

Keys or CA

A `keys-file` maps raw keys to users. A `ca-keys-file` validates user certificates. Either source is sufficient.

BINDING

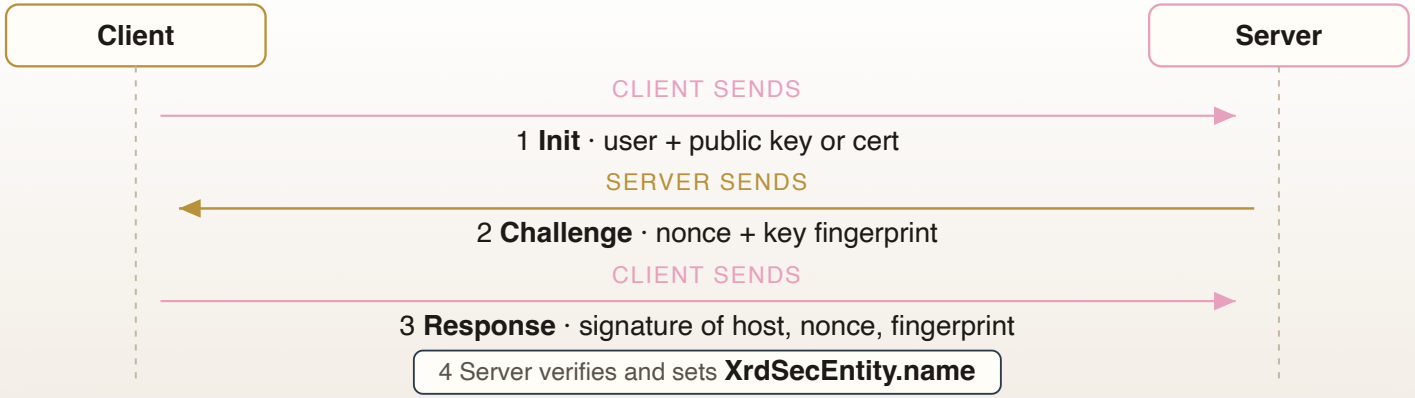
Host in the signature

The client signs the hostname it connected to. A redirect target cannot reuse that login on another server.

TAKE-AWAY SSH keys or user certificates on TLS `root://`. The host name is part of the signed payload.

The server challenges; the client responds

Two round-trips after TLS. The server picks the random nonce. The client proves it holds the private key. The key never leaves the client.



TLS is mandatory. The nonce is per connection and single-use (`-nonce-ttl`). List load-balancer names with `-hostnames`.

TAKE-AWAY The server sends the challenge. The client sends the response. The signature is valid only for that host.

Mapping, revocation, and status

POLICY

- Principal map + `-principal-as-user`
- `-deny-users` defaults to `root`
- Hot-reloaded revocation list (binary KRLs are not supported)
- Empty principals are not treated as a wildcard
- Client: PEM / OpenSSH file / ssh-agent

IMPLEMENTED

- `sec.protocol ssh` + TLS gate + host binding
- 68 unit tests, packaging

NOT IMPLEMENTED

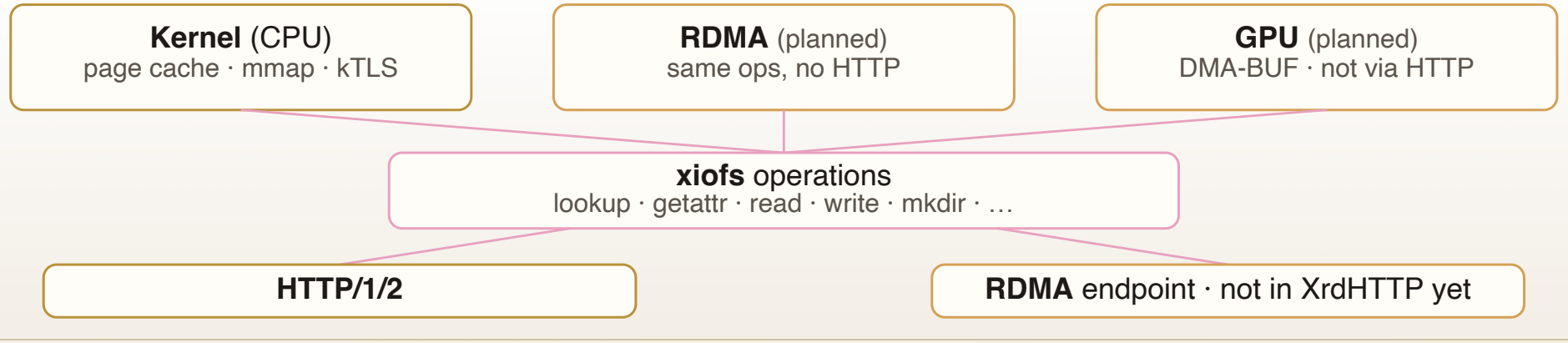
- TLS channel binding; ECDSA / FIDO; HTTPS handshake

TAKE-AWAY No HTTPS, ECDSA, or TLS channel binding yet. This plug-in could serve use cases of sss authentication.

XrdHttp extensions

WHAT IT ADDRESSES

Extends the HTTP plug-in to use a standard, widely deployed header parser, to support HTTP/2 via nghttp2, and ultimately to provide kernel, RDMA, and GPU filesystem access.



TAKE-AWAY HTTP is implemented. RDMA and GPU are planned and not yet on the wire.

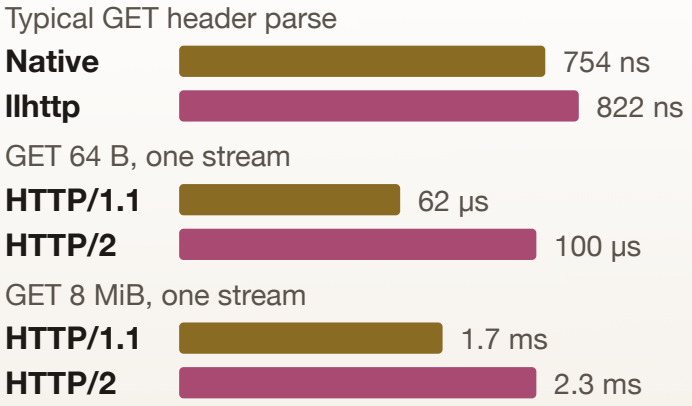
Changes versus master

The GET / PUT / HEAD / DELETE, body, TPC, and checksum implementations in the request handler are unchanged. Header parsing was replaced so that HTTP/1 is bounded and shares one request object with HTTP/2.

	MASTER	THIS BRANCH
HTTP/1 header parser	line scanner	llhttp (or `http.parser legacy`)
HTTP/2	none	nghttp2, then the same `XrdHttpReq`
Kerberos on HTTP	none	`http.auth krb5` SPNEGO
Partial write	PUT replaces the file	PATCH with `Content-Range`
POSIX over HTTP	none	XIOFS: CLI, FUSE, kernel module

MICROBENCH

Parser and HTTP/2



Native header parsing remains faster; the switch was not made for speed.

TAKE-AWAY Same `XrdHttpReq` path. New: parser, HTTP/2, Kerberos, PATCH, XIOFS.

XIOFS kernel client

IMPLEMENTED

llhttp, HTTP/2, Kerberos, PATCH / PROPPATCH / LINK, FUSE, kernel + kTLS.

NOT IMPLEMENTED

RDMA, GPU-direct (ioctl stub), in-kernel HTTP/2, parallel Bridge.

NEXT

Propose the parser, HTTP/2, and Kerberos for master. XIOFS as the follow-up.

TAKE-AWAY The kernel mount uses HTTP/1.1 + kTLS. Propose HTTP/2 first; XIOFS as the follow-up.

Client range journal

WHAT IT ADDRESSES

A journaling client-side cache reduces the cache space required for repeated identical runs (e.g. analysis) to the minimum amount of data the application actually reads. This reduces read amplification in filesystems and keeps replayed I/O fully sequential.

WHERE

Client, proxy, or shared FS

XrdCl plugin for `xrdcp`, ROOT, PSS, XrdHttp, or `xjcd`. The journal can reside on a shared high-performance filesystem.

WHAT

Exact ranges

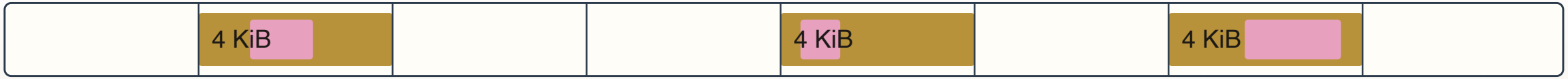
A hit requires the *full* request to be present in the journal. A miss is fetched and appended. A write resets that file's journal. Reports IOPS and an optional I/O profile.

RELATION

XCache (XrdPfc)

XCache is a shared, server-side block cache with prefetch. The two are complementary.

Original file · 4 KiB blocks · read amplification



Gold: bytes read from the file or shared FS. Pink: the ranges the application requested (unaligned, < 4 KiB).

Journal · exact pieces, no 4 KiB rounding

1.2 + 0.7 + 2.0 KiB stored. Gaps in the file are not filled.

TAKE-AWAY Exact ranges, no prefetch. On a shared filesystem the cache is shared and read amplification is eliminated.

JournalCache versus XCache

	JOURNALCACHE	XCACHE (XRDPFC)
Role	XrdCl plugin, optional HTTP forwarder	Server OSS cache in front of PSS
Unit	Exact read ranges	Fixed blocks (default 256 KiB)
Prefetch	None	Adjacent blocks
Sharing	One flock writer per file; the journal can reside on a shared filesystem serving many nodes	Many clients on one node
I/O profile	Reports IOPS and key metrics on request	gstream / ResourceMonitor
HTTP freshness	Cache-Control, ETag, and 304 via the HTTP extension	`pfc.httpcc`

JOURNALCACHE

`XRD_PLUGIN=libXrdClJournalCachePlugin-5.so`

Sparse rereads (ROOT branches). Closed HTTP forwarder for `/https://origin/path` (`xjcd` + `xjc allow-origin`). The HTTP extension honours Cache-Control and can return 304. Reports IOPS and can print an I/O profile with key metrics on request.

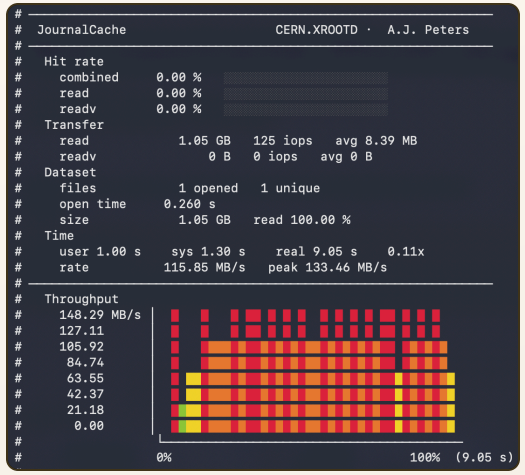
XCACHE

Shared site cache in front of a federation. Many jobs, sequential reads, production purge. The two can be combined.

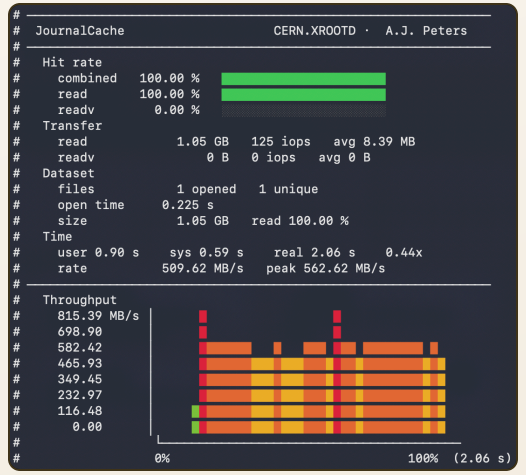
TAKE-AWAY Sparse client rereads: JournalCache. Shared site cache: XCache.

I/O profile from the plugin

UNCACHED



CACHED



TAKE-AWAY Same 1.05 GB read: 0% hit in 9 s, then 100% hit in 2 s. The most useful integration is probably a simple switch in ROOT for end-user analysis.

NFSv4 front-end for XRootD

WHAT IT ADDRESSES

Exports an XRootD namespace as a standard Linux NFS mount, enabling native kernel mounts in Linux environments. It is a complete open-source implementation of NFSv4.1 in C++, written from scratch. Originally this has been developed to add NFSv4.1 to EOS, but since it has a plug-in VFS it can be added to XRootD as well.

CHEP 2026: Adding NFS 4.2 as a Strategic Protocol for EOS

PROTOCOL

NFSv4.0 and 4.1

Userspace NFS server. Full 4.0 compounds. Optional minor=1 sessions and pNFS flexfiles MDS. AUTH_SYS. Kernel clients mount with `vers=4.1`.

LIBRARY

libcernnfs

The daemon is a thin CLI around `cernnfs::ServerBuilder`. The same server can run in-process in any XRootD-derived process.

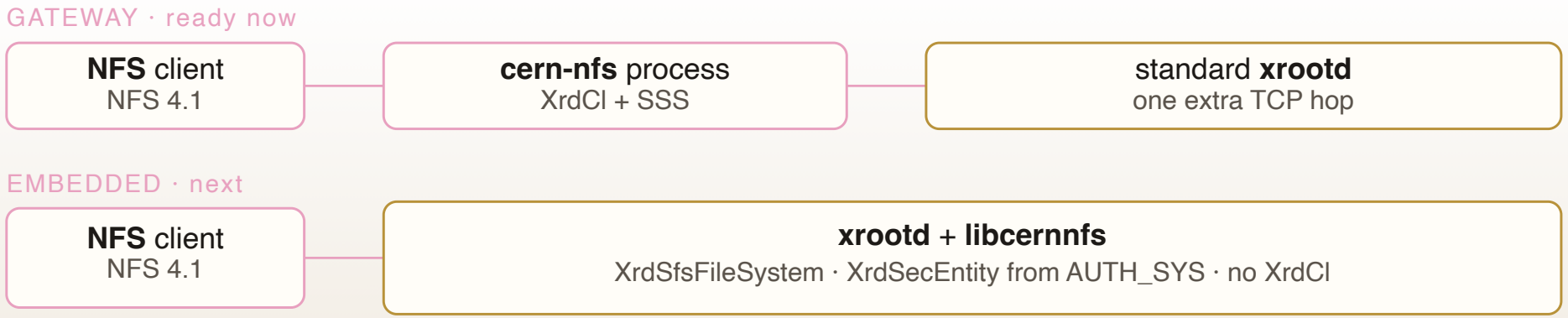
CLIENT

Kernel NFS mount

Ordinary Linux mounts. No eosxd, no XrdPosix, no per-job plugin. Batch nodes, containers, and laptops mount it directly.

TAKE-AWAY The NFS core is backend-agnostic. XRootD is one VFS: XrdCl (gateway) or XrdSfs (embedded).

Gateway and embedded



Gateway

```
cern-nfs --nfs-minor 1 \
  --xrd-sss-keytab=/etc/cern-nfs/sss.keytab \
  root://xrd-host:1094//export
sudo mount -t nfs4 -o vers=4.1,port=2049 \
  cern-nfs-host:/ /mnt/xrd
```

Embedded

No SSS, no second process, no extra hop. Once `ofs` is initialised, pass the live `XrdSfsFileSystem*` to CERN-NFS and listen on port 2049 from a worker thread.

TAKE-AWAY Same NFS server. The gateway re-exports over the network; the embedded variant calls `XrdSfs` in-process.

Identity, pNFS, and status

NFS AUTH_SYS
sec=sys · sec=krb5

IdentityScope
SSS loginid or XrdSecEntity

ofs.authorize
same checks as root:// clients

READY

- NFSv4.0 / 4.1, `libcernnfs`, RPM
- `sec=sys` and `sec=krb5`
- Gateway: XrdCl + SSS
- pNFS flexfiles with a configured DS list

IN PROGRESS

- Embedded `XrdSfs` mapping
- Per-request AUTH_SYS identity
- Locate-driven pNFS mirrors

PNFS

- XRootD Locate becomes the flexfiles DS list
- The kernel client reads directly from the node that holds the data
- An EOS branch already tests MGM/FST in-process

TAKE-AWAY XRootD stays authoritative for ACLs. The gateway is usable today; the native plugin is the same server running in-process.

aiosd

C++20 cluster object store (private project)

WHAT IT ADDRESSES

Stores data as objects on a small cluster and presents that cluster to XRootD as a filesystem back-end.

CLUSTER

Gossip + map

`aiosd` peers join over signed TCP++ gossip. Each node rebuilds the same cluster map (epoch, targets, classes) from advertised capacity.

PLACE

Class-scoped ring

`place(oid, class, n)` walks a vnode ring. Primary is first in the acting set. A membership change remaps about 1/N of objects.

DURABILITY

Replica or EC

Replica fan-out, or erasure coding (`k+m`, XOR or ISA-L). Layout and storage class are chosen per object version at write time. No pools.

TAKE-AWAY Placement is per object, not per pool. Experimental. Test data only.

Store bench on local NVMe

`aios-store-bench`, filesystem bodies, one node. 1 MiB objects, 4 KiB I/O, 8 threads, 4000 ops, no fsync.

AIOS-STORE-BENCH

```
aios % ./build/aios-store-bench --root /var/tmp/aios-bench \
    --mode fs --op all --threads 8 --count 4000 \
    --large-size 1048576 --io-size 4096 --no-fsync
[aios][info] object store open … shards=16 inline_max=65536
=== filesystem bodies size=1048576 io=4096 threads=8 ===
  put      4000 ops    2.390 s    1674 ops/s  1673.93 MiB/s
  get      4000 ops    1.299 s    3080 ops/s  3080.27 MiB/s
  del      4000 ops    0.176 s   22737 ops/s     0.00 MiB/s
  rwrite   4000 ops    6.802 s     588 ops/s     2.30 MiB/s
  rread    4000 ops    0.052 s   76510 ops/s   298.87 MiB/s
  append   4000 ops   15.606 s     256 ops/s     1.00 MiB/s
```

THROUGHPUT

Whole-object and range read

get 3.08 GiB/s

put 1.67 GiB/s

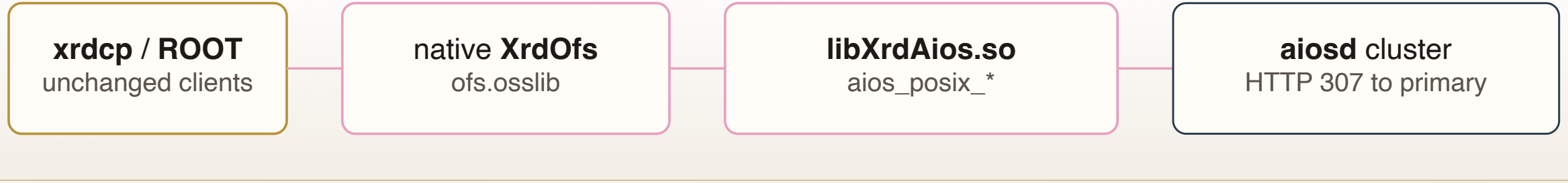
rread 299 MiB/s

4 KiB rwrite 588 ops/s · append 256 ops/s

p50 get 1.8 ms · put 3.7 ms · rread 0.07 ms

TAKE-AWAY Local NVMe, no fsync: get 3.1 GiB/s, put 1.7 GiB/s. Not a cluster number.

XRootD OSS plugin



IDENTITY

Local passwd map

- Each operation reads `XrdSecEntity.name`
- `getpwnam_r` sets the POSIX caller
- Unknown names are rejected (fail closed)
- Mapped uid 0 is POSIX root

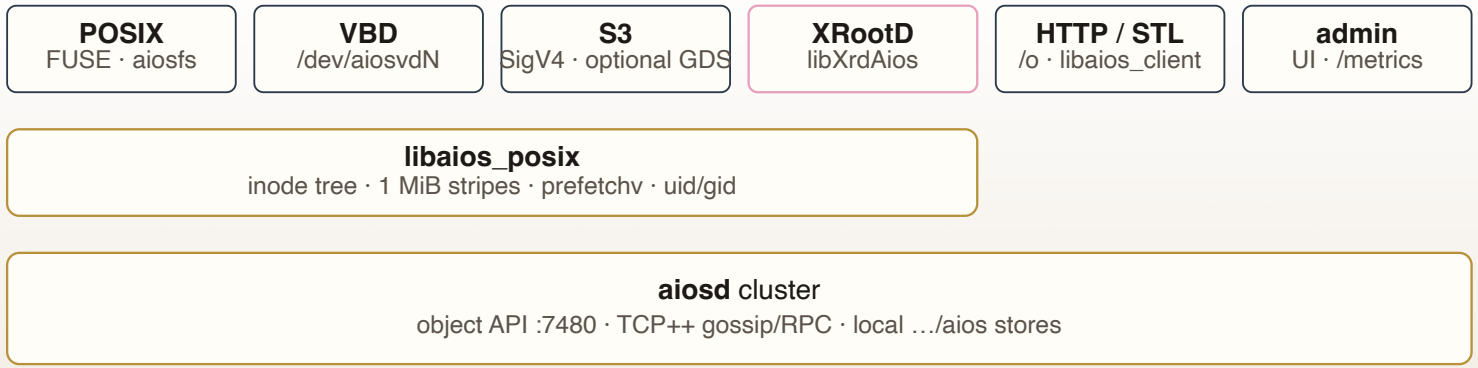
I/O

ReadV on the kernel interface

- Sparse-range prefetch: `ioctl(AIOS_IOC_PREFETCHV)` / `aios_posix_prefetchv`
- Up to 256 inline file ranges as one backend batch
- Same UAPI on FUSE and kernel aiosfs (`kernel/aiosfs_uapi.h`)
- `libXrdAios ReadV` calls it before filling the vector
- TPC and AIO stay in native XrdOfs

TAKE-AWAY The gateway is an ordinary xrootd. `ReadV` uses the same kernel prefetch UAPI as FUSE and aiosfs.

Frontends



IN AIOS TODAY

- Gossip, CH placement, replica / EC
- HTTP object API, S3, POSIX, FUSE
- `libXrdAios` OSS + ReadV prefetch

NOT IN LIBXRDAIOS

- An XRootD-native object API
- Proxy FSctl

STILL R&D

- Production auth beyond passwd + HMAC
- Identity hardening (opener versus later operations)

TAKE-AWAY Experimental object store. XRootD sees a POSIX tree because the OSS plugin mounts `libaios_posix`.

PLUGIN	READY TO PROPOSE	STILL R&D
XrdSecoidc	Authn on `root://` and HTTPS; Acc authorizes	Revocation, EdDSA
XrdSecssh	SSH keys / user certs on TLS `root://`	Channel binding, HTTPS
XrdHttp	llhttp, HTTP/2, Kerberos, PATCH	XIOFS follow-up; RDMA / GPU later
JournalCache	Client range journal; closed HTTP proxy	Simple ROOT switch for analysis
cern-nfs	Gateway NFSv4 in front of a standard xrootd	In-process XrdSfs plugin
aios	-	OSS plugin on an experimental object store

TAKE-AWAY NFS and AIOS remain R&D.

These plug-ins and R&D projects are experimental but, in my view, ready for evaluation. The code requires another review pass after the branches have been squashed. Feedback is needed on which parts are useful for XRootD now, and on who would like to participate in the R&D that is not yet ready to land.

PLUGIN	REPOSITORY	BRANCH
XrdSecoidc	github.com/cern-eos/xrootd	XrdSecoidc
XrdSecssh	github.com/cern-eos/xrootd	XrdSecssh
XrdHttp	github.com/cern-eos/xrootd	XrdHttp2
XrdClJournalCache	github.com/cern-eos/xrootd	XrdClJournalCachePlugin
cern-nfs	gitlab.cern.ch/eos/cern-nfs	master
aios	github.com/apeters1971/aios	master

TAKE-AWAY Needed: project priorities for master, and interested people / testers for the R&D.