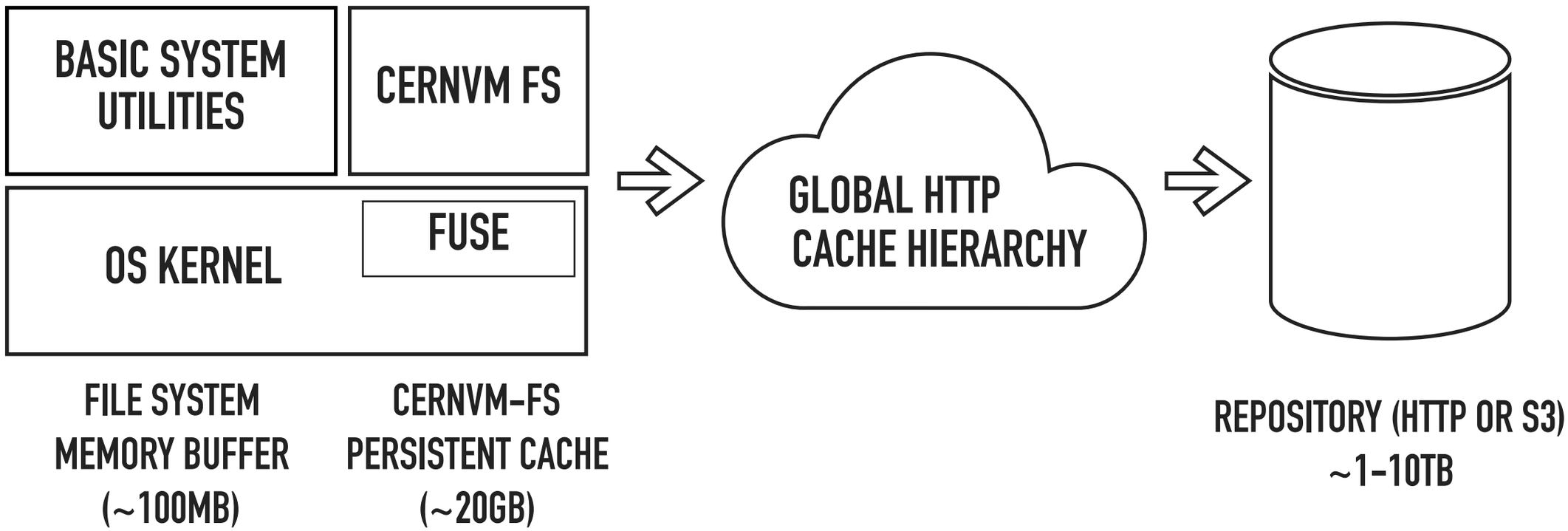
RADU POPESCU, CERNVM-FS TEAM

CERNVM-FS: BEYOND SOFTWARE DISTRIBUTION

GENERIC COMPONENTS OF THE ESCIENCE INFRASTRUCTURE ECOSYSTEM, OCT 2018, AMSTERDAM

A FILE SYSTEM APPROACH TO DISTRIBUTING SOFTWARE



- FUSE based, independent mount points, e. g. /cvmfs/sft.cern.ch
- Clients have a read-only view; single writer into repository
- Immutable, content-addressed storage
- HTTP transport, access and caching on demand

SOME STATS

- ~100 000 clients on the WLCG
- Largest repositories: O(10^8) number of files, O(TB) content size
- 85 monitored repositories
- Platforms:
  - x86_64, i386, ARM (aarch64)
  - Linux: RHEL, Ubuntu LTS, Debian, SLES
  - macOS
  - Experiment: Raspberry Pi, RISC-V!
- Latest version: CernVM-FS 2.5.1

SOFTWARE VS DATA

There are differences when storing and distributing data vs software:

- File size
- Number of files
- Access policies
- Storage location
- Change frequency

Large data distributions require a separate infrastructure, to avoid impacting the software distribution use case.

BEYOND SOFTWARE DISTRIBUTION WITH CERNVM-FS

- Conditions data distribution for ALICE and LHCb
- OSG's StashCache for working sets < 10 TB
- Using LIGO data on opportunistic resources in OSG

EXTERNAL DATA DISTRIBUTION

A set of features that allow the distribution of a namespace of an existing data repository (ex: LIGO):

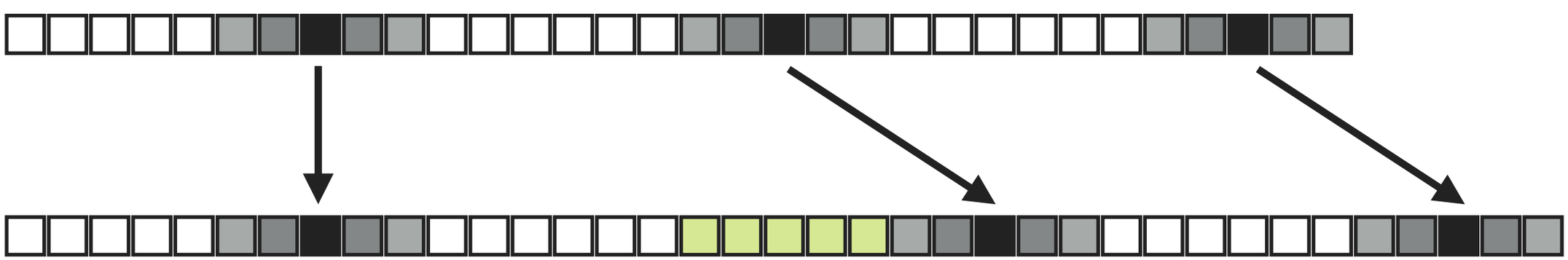
- Grafting (only store file metadata)
- Uncompressed files
- HTTPS access (with authorisation plugins)
- A separate infrastructure needs to be operated for this!

AUTHORISATION HELPER PLUGINS

- External processes communicate with CernVM-FS FUSE module over stdin/stdout
- Grant or deny read access to processes based on uid, gid, and “membership”
- Support for X.509 proxy certificates
- SciTokens support is coming in CernVM-FS 2.6.0, thanks D. Weizel!

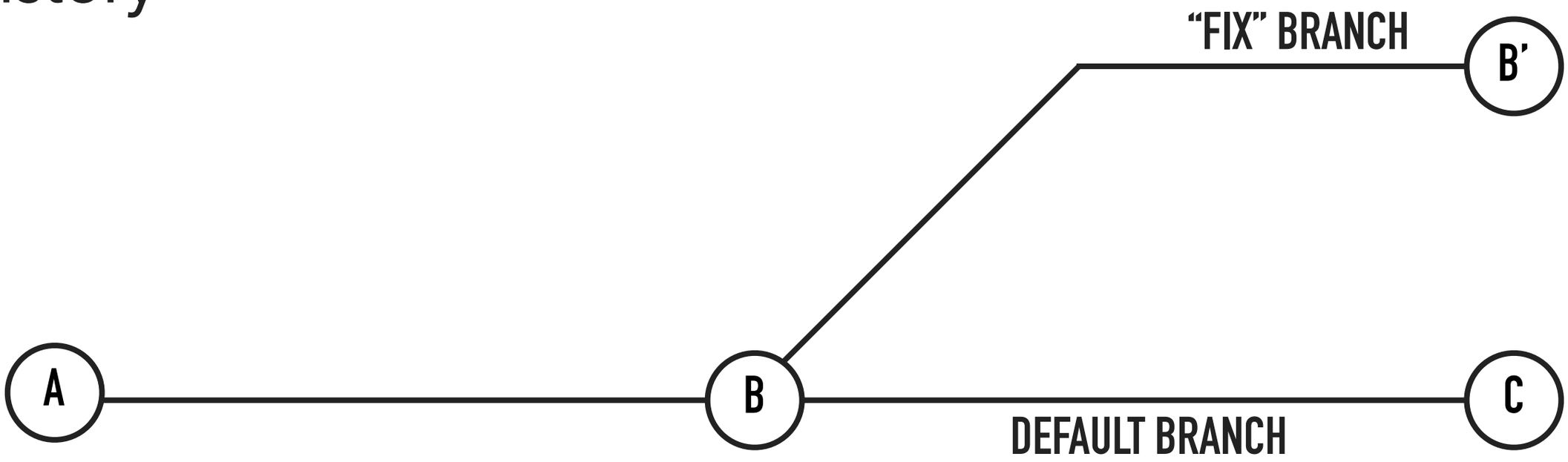
FILE STORAGE IN CVMFS

- Files are stored as compressed, content-addressed chunks
- Max chunk size can be configured
- Rolling checksum algorithm maximises chunk reuse



VERSIONING FEATURES

- Snapshots: All the snapshots in a repository's history are accessible through the .snapshots virtual directory (similar to ZFS)
- Branching: organise repository snapshots into a non-linear history



THE SNAPSHOT DIRECTORY

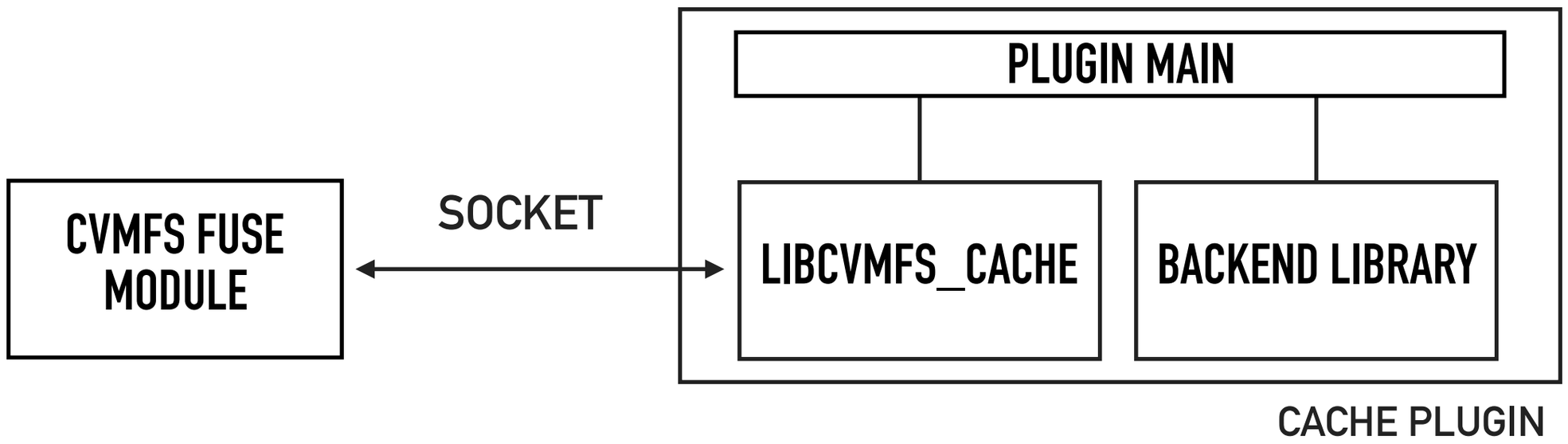
```
λ ls -al /cvmfs/cernvm-prod.cern.ch/.cvmfs/snapshots
total 381
dr-xr-xr-x 97 cvmfs staff   97 Sep 12 10:15 .
dr-xr-xr-x  3 cvmfs staff   97 Sep 12 10:15 ..
drwxr-xr-x  7 cvmfs staff 4096 Jan 13  2014 HEAD
drwxr-xr-x  4 cvmfs staff 4096 Jan 13  2014 cernvm-system-3.1.0.0
drwxr-xr-x  4 cvmfs staff 4096 Jan 13  2014 cernvm-system-3.1.1.0
drwxr-xr-x  4 cvmfs staff 4096 Jan 13  2014 cernvm-system-3.1.1.1
drwxr-xr-x  4 cvmfs staff 4096 Jan 13  2014 cernvm-system-3.1.1.2
```

CLIENT CACHE

- The CernVM-FS FUSE client caches content-addressed blocks locally
- Cache can be configured to suit different use-cases or environments: disk cache, external cache plugin, tiered cache
- At CSCS Lugano, CernVM-FS is running on Piz Daint (Cray XC40/50, #6 Top500), using disk cache on GPFS

EXTERNAL CACHE PLUGIN API

- A cache plugin is an external process which communicates with the main CernVM FS client process through a socket (UNIX or network), using a well defined protocol



CACHE PLUGIN API

- The plugins support different operating environments, such as diskless compute nodes in HPC
- Current plugins: in-memory cache, RamCloud (low latency key-value store), XRootD
- There is a library for developing new plugins

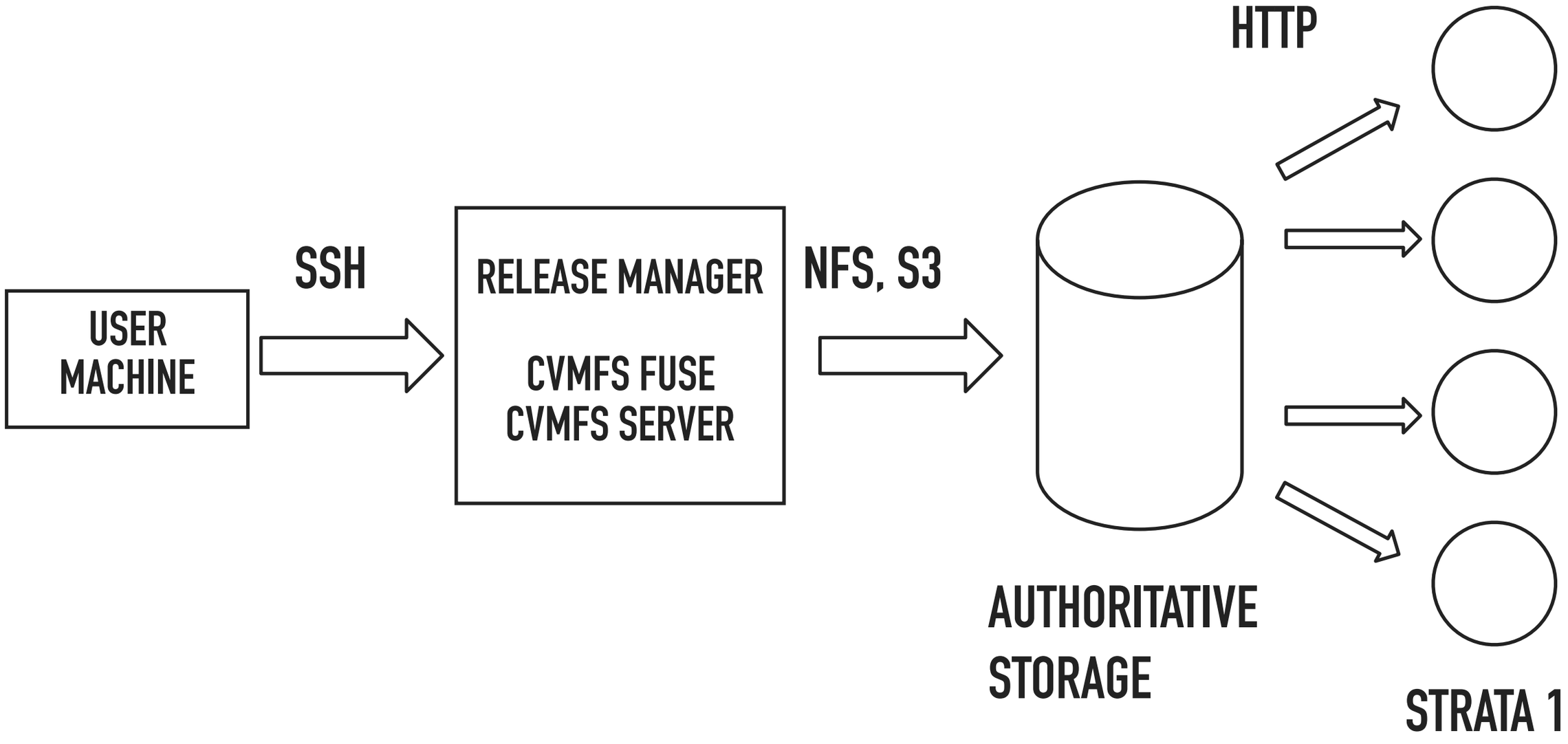
TARBALL INGESTION (IN CERNVM-FS 2.6.0)

- Allows to directly publish the contents of an archive

```
# cvmfs_server ingest -t archive.tar sft.cern.ch
```

- Good performance, avoid passing through disk
- No OverlayFS needed
- No way to run any scripts in the same transaction
- A core component for container image distribution (unpacked.cern.ch)
- Thanks, Simone Mosciatti!

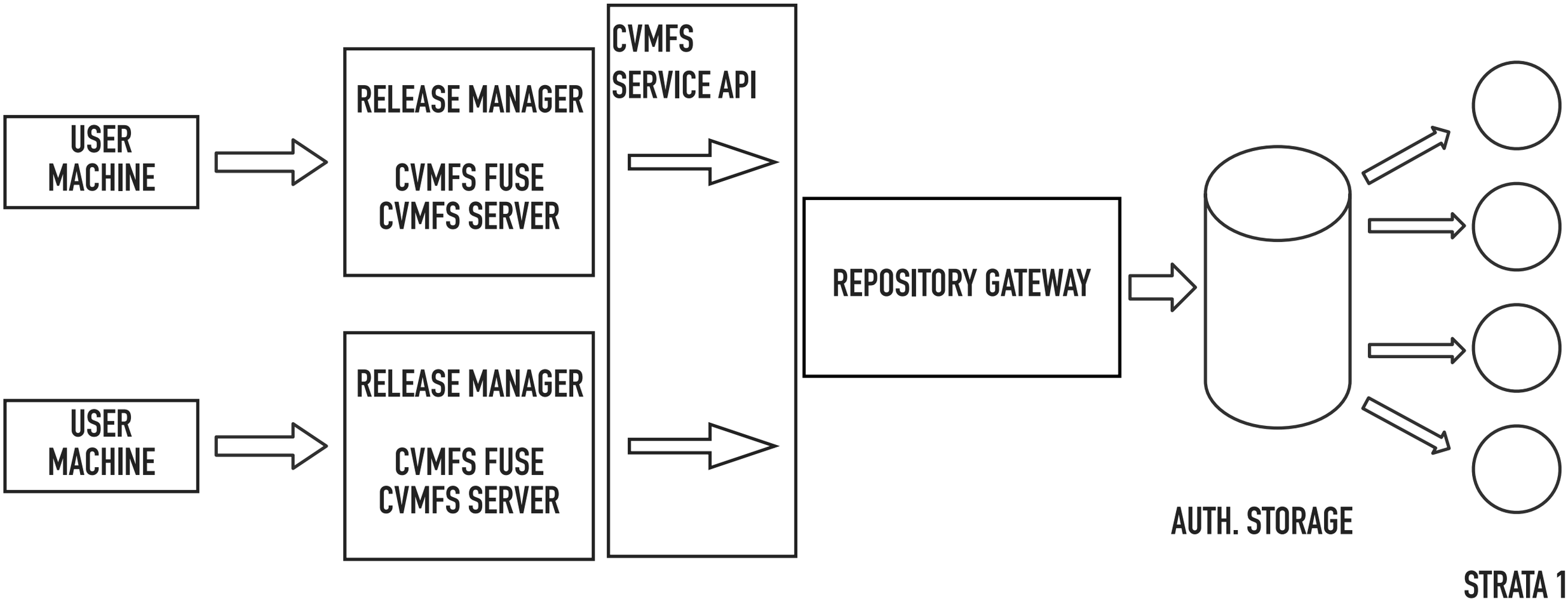
SINGLE PUBLISHER PER REPOSITORY



DISADVANTAGES

- No support for concurrent writing
- Shell access needed on the machine with direct access to the repository storage
- No fine grained access control
- Possible performance issues for very large change-sets

MULTIPLE PUBLISHERS



HORIZONTALLY SCALING THE PUBLICATION PROCESS

- Scale-out to multiple release manager machines, process changes concurrently
- New use cases possible, such as multi-tenant repos, containerised release manager
- More flexibility, but at an increased maintenance cost for the user; optional

S3 REPOSITORY BACKEND

- In addition to locally-mounted volumes, CernVM-FS can use S3-compatible object stores as repository storage backend
- Available S3 object stores: AWS S3, Google Cloud Storage, Ceph (CERN S3 Service), Minio (for testing)
- Advantages: availability, scalability, uploading with multiple streams
- Compatible with CernVM-FS repository gateway

XCACHE AS AN HTTP PROXY (EXPERIMENT)

- Xcache is an XRootD configuration that provides a high-performance file proxy
  - Accessed using XRootD or HTTP, ingests from XRootD or HTTP (new ingestion plugin)
- With HTTP ingestion plugin it can be inserted non-intrusively between CernVM-FS strata and clients
- Use cases: high-performance site-level cache, better than Squid for large files?
- All the pieces available in XRootD 4.9

SUMMARY

- CernVM-FS has a few useful features in a data distribution context
- Most features are orthogonal, can be mixed and matched
- CernVM-FS is not meant to replace existing data distribution solutions like XRootD, but complement them
- S3 object stores are becoming a preferred storage backend, we aim to have the best integration possible
- Looking forward to CernVM-FS 2.6.0 in Q1 2019

THANK YOU!