

# CernVM: Status and Plans

Perspective of the CERN development team -  
CernVM Workshop 2024



Sep 16th 2024, CernVM Workshop  
Valentin Völkl for the CVMFS Development Team at  
CERN

# Preliminaries

- CernVM(-FS) originated at CERN and the core development team is hosted here
  - Naturally the mission of CERN and the requirements of the recognized experiments drive the future of the project
- ... but has an increasingly wider user community
  - Which benefits the project!
    - Early efforts on “[Taking the C out of CVMFS](#)” - still supported
  - The feedback process is less structured
    - This workshop is a fantastic opportunity!

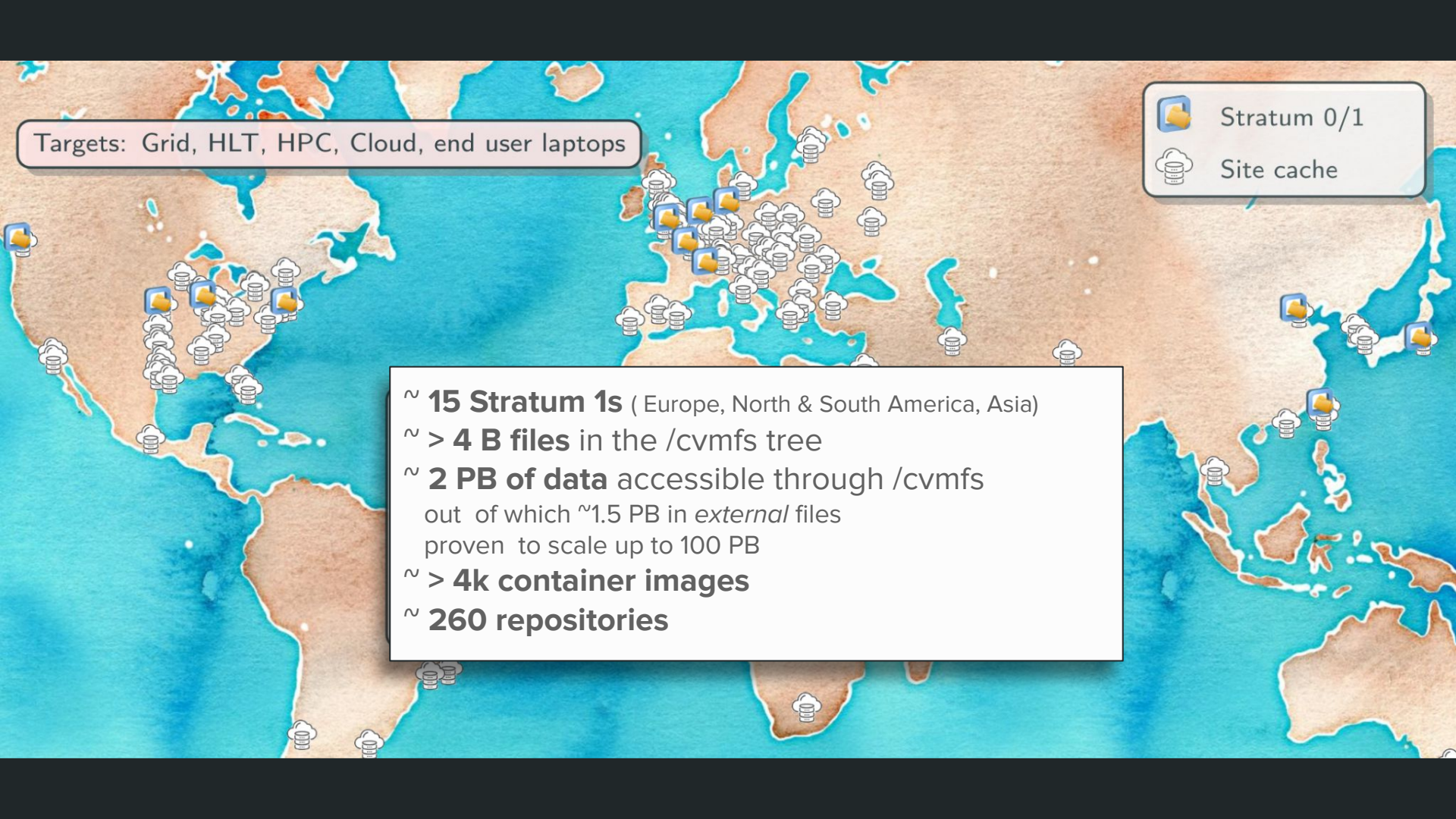
A world map with a light beige and blue color scheme. Numerous lightbulb icons are scattered across the map, primarily concentrated in Europe and North America. A semi-transparent white box is overlaid on the map, containing text.

## Key users:



- LHC & smaller CERN experiments
- Euclid, Jump Trading (contractual partners)
- Other scientific communities & industry (e.g., EESSI, LIGO, SKA, LSST, Roche, etc.)

## Key stakeholders:

- Experiments & end users: producers and consumers of data
- Site operators: focus on smooth operations, low-maintenance effort
- Stratum 1 operators: donate resources to the WLCG/cvmfs operations
- Developers: SFT, Jump Trading, Fermilab, community (“cvmfs-contrib”)

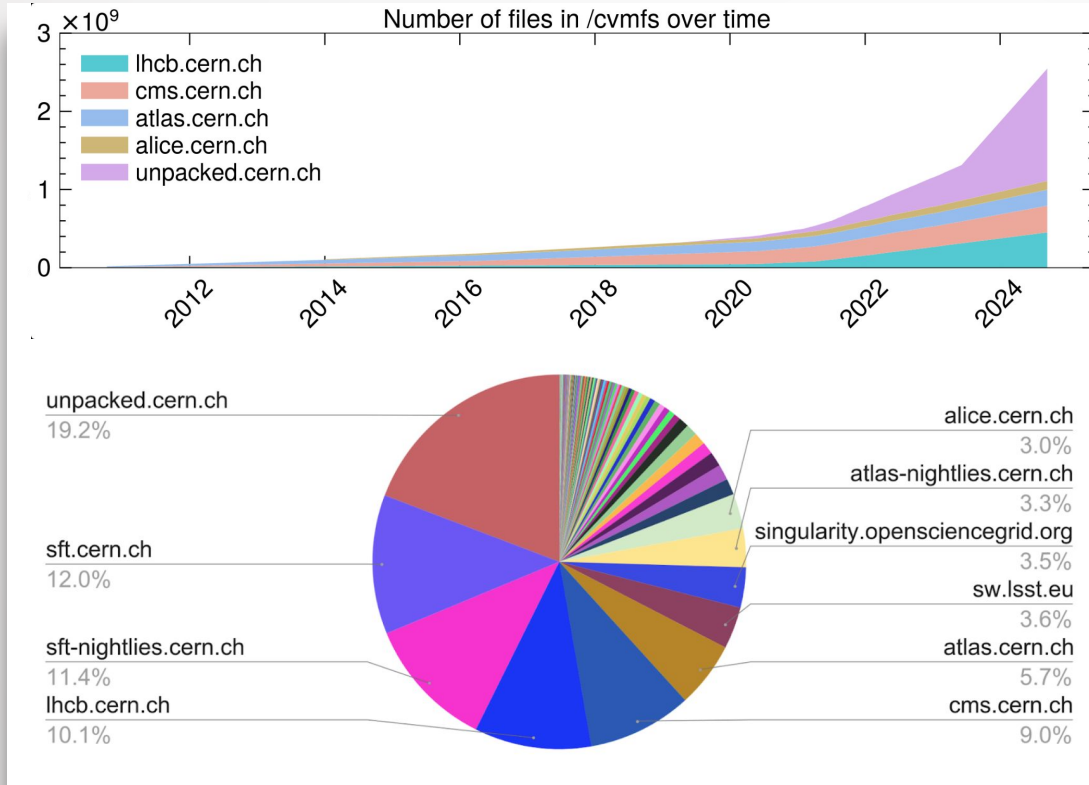
A world map with a watercolor-style background. The map is populated with two types of icons: blue folders with yellow arrows (representing Stratum 0/1) and stacks of white disks (representing Site cache). The Stratum 0/1 icons are concentrated in North America, Europe, and East Asia. Site cache icons are more widely distributed across all major landmasses. A legend in the top right corner identifies these icons. A text box in the top left lists targets: Grid, HLT, HPC, Cloud, and end user laptops. A large text box in the center provides statistics for the Stratum 1s.

Targets: Grid, HLT, HPC, Cloud, end user laptops

 Stratum 0/1  
 Site cache

~ **15 Stratum 1s** ( Europe, North & South America, Asia)  
~ **> 4 B files** in the /cvmfs tree  
~ **2 PB of data** accessible through /cvmfs  
out of which ~1.5 PB in *external* files  
proven to scale up to 100 PB  
~ **> 4k container images**  
~ **260 repositories**

# CVMFS in numbers



~ **15 Stratum 1s**

~ **> 4 B files** in the /cvmfs tree

~ **2 PB of data** accessible through /cvmfs  
out of which ~1.5 PB in *external* files  
proven to scale up to 100 PB

~ **> 4k container images**

~ **260 repositories**

- Backed by S3(+CEPH) or local storage
- Thanks to IT-Storage and the operators who expertly manage this infrastructure!

# Current status: CVMFS 2.11 (Released 2023)

See [full changelog](#) for more details:

Release Notes for CernVM-FS 2.11.0

Overview

Getting Started

Client Configuration

Setting up a Local Squid Proxy

## Improvements and changes

- [client] Re-use the file descriptor for a file already open in the local cache (#3067)
- [client] Add support for symlink kernel cache through CVMFS\_CACHE\_SYMLINKS (#2949)
- [client] Add telemetry framework to send performance counters to influx (#3096)
- [client] Add streaming cache mode through CVMFS\_STREAMING\_CACHE=yes (#3263, #2948)
- [client] Add CVMFS\_STATES\_CACHE\_TIMEOUT parameter to cache status results (#3015)

# Reference-counting Cache Manager

- `CVMFS_CACHE_REFCOUNT`: fixes a long-standing issue with many processes concurrently reading the same files; impacted ALICE in particular
  - cvmfs would open new file descriptors for the same files, sometimes reaching the system limit
  - Can be worked around, but requires effort on sys-admin side
- New cache manager mode keeps a table with references to file descriptors in memory and no longer duplicates them
  - Small overhead, but should not exceed a few MB of memory
- Already enabled for ALICE in the `cvmfs-config.cern.ch` repository, will become default in 2.12
  - But needs 2.11.2 in order to avoid having to increase the cvmfs file descriptor limit

## 2.11 Improvements in **Logging**

... are crucial, as many errors are hard to reproduce by the developers. Many small improvements were added in 2.11:

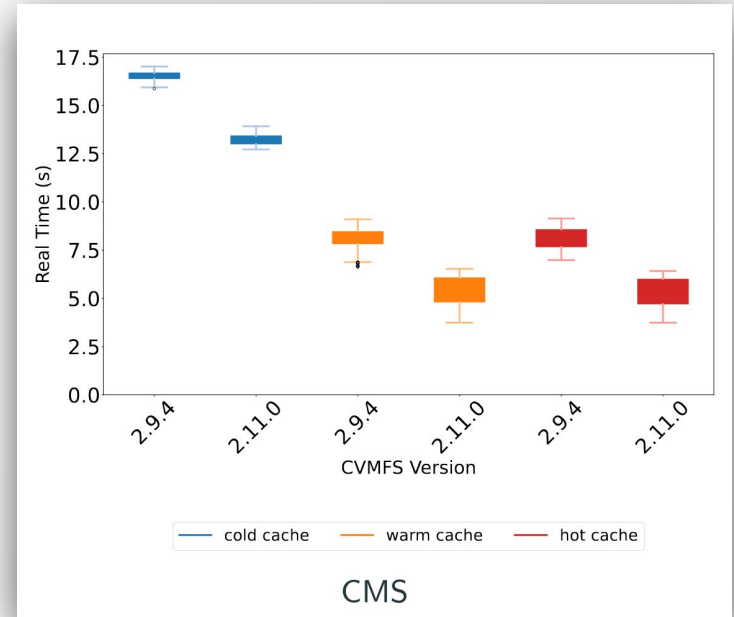
- Debug mode now preserved across `cvmfs_config` reload
- Debug output of internal Curl queries now available in the debug log
- Dedicated “\*.mount” log files for the mount helper
- Re-store core-file generation after credentials drop
- Improve logging of FUSE I/O errors

2.11 also introduces client telemetry that can be used with InfluxDB ([link](#)) and custom http tracing headers



# Performance improvements for caching

- Page Cache Tracker: Much better use of kernel page cache (already in 2.10)
- CVMFS\_SYMLINK\_CACHE possible on new enough FUSE/Kernel versions
  - Requires libfuse 3.10+
  - And kernel in rhel8+
- `Statfs` caching



See [CHEP 2023](#) for more details

# Improvements for external data

External Data: used by LIGO, osgstorage/stashcache, and (privately) by Jump Trading

- `CVMFS_CACHE_STREAMING`: 2.11 introduces a new “streaming” cache manager mode. This bypasses the cache completely, except for catalogs. Useful in a very special set of circumstances, mostly not for the software distribution usecase
- Protected extended attributes: allows to restrict xattrs by uid

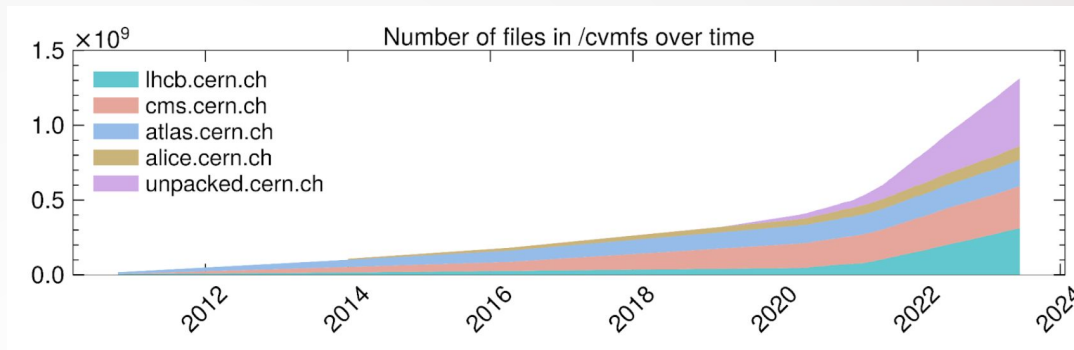
# Containers

- CVMFS provides tooling to unpack, store and distribute containers, with *unpacked.cern.ch* being the biggest repository:

```
~$ ls /cvmfs/unpacked.cern.ch/registry.hub.docker.com/cmssw/cs8\:x86_64-d20211124
afs  build  dev      etc    lib64  mnt    proc    sbin      sys  var
bin  cvmfs  environment  home  lost+found  opt    root    singularity  tmp
boot data  eos      lib    media  pool   run     srv        usr
```

- *Apptainer* can directly launch the container from this root file system.
- The same benefits from using CVMFS apply! Leading to:
  - Drastically faster container **startup** times
  - Automatic **cache management** of container images on the worker nodes

# Outlook: unpacked.cern.ch



- Very useful bridge to container deployment model
  - And lower-barrier entry to cvmfs publishing
- Many improvements that will be included in 2.12, following successful summer student project
  - REST API
  - Major refactor
- Can possibly free up some space by garbage-collection campaign

# Outlook on development new features in next releases

- File Bundles
  - Groups downloads of files that are accessed together
  - Can improve interactive access
- Container tools and ephemeral write shell
  - Helm charts
- Zstd compression (See Laura's talk)

# Further outlook

- CVMFS is (mainly) used for deploying software, in distributed computing environments
- Future trends in industry will affect us
  - Containers have been the biggest change so far
    - Integrated very well with CVMFS!
  - Future technologies will probably not completely replace the old, but add new possibilities
- Increased usage of HPC resources is a clear trend in the HEP community
- AI/ML seems to be most likely to change how “software” looks on disk

# Packaging

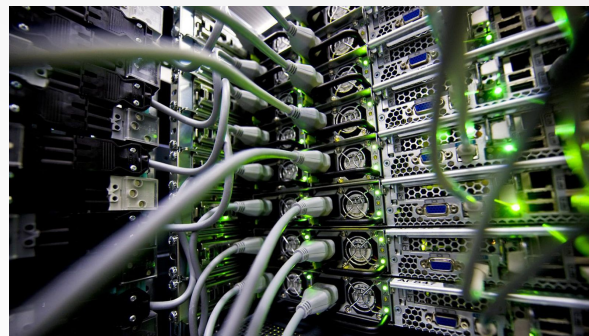
```
yum install cvmfs
apt install cvmfs
```

- Providing pre-built packages and yum/apt repositories seems to be appreciated
- cvmfs-prod and cvmfs-testing
  - Plan to add cvmfs-devel
- Target firstly:
  - RHEL(-clones) and Debian
  - MacOS for the laptop usecase (no server tools).
  - Open to adding new ones!
- **Goal:** get packages into upstream repositories for Debian and Fedora

Configuration Matrix	docker-i386	docker-x86_64	docker-aarch64
cc7	...	✓	✓
cc8	...	✓	✓
cc9	...	✓	✓
debian10	...	✓	...
debian11	...	✓	...
debian12	...	✓	✓
fedora38	...	✓	...
fedora40	...	✓	...
sles15	...	✓	...
ubuntu1804	✓	✓	...
ubuntu2004	...	✓	...
ubuntu2204	...	✓	✓
ubuntu2404	...	✓	✓
mac	...	...	...
container	...	✓	...

Please do consider using also the `cvmfs-testing` repositories!

# Further outlook: Hardware



- Increased parallelism (machines with  $> 128$  cores) introduces new challenges and bugs
  - Thanks to Jump Trading for reporting many of these, ahead of experiments!
  - Integrate further in test infrastructure
- Open to requests for new hardware target of packages (RISC-V, ppc ...)
  - RISC-V was already experimented with in 2018 with the HiFive Unleashed board

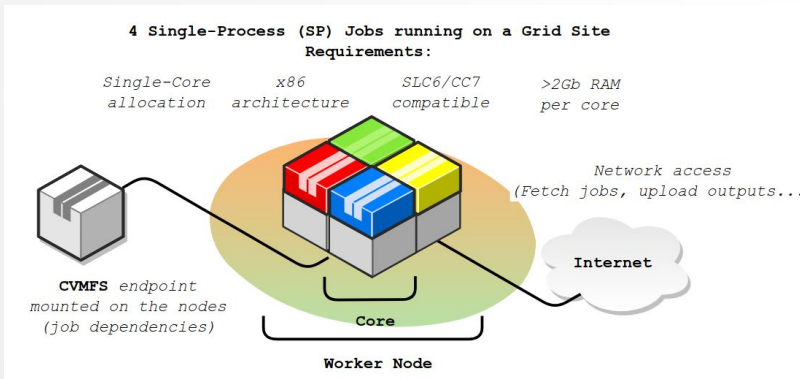
```
# file /usr/bin/cvmfs2
/usr/bin/cvmfs2: ELF 64-bit LSB executable, UCB RISC-V, version 1 (GNU/Linux),
dynamically linked, interpreter /lib/ld-linux-riscv64-lp64d.so.1, for GNU/Linux 4.15.0
```



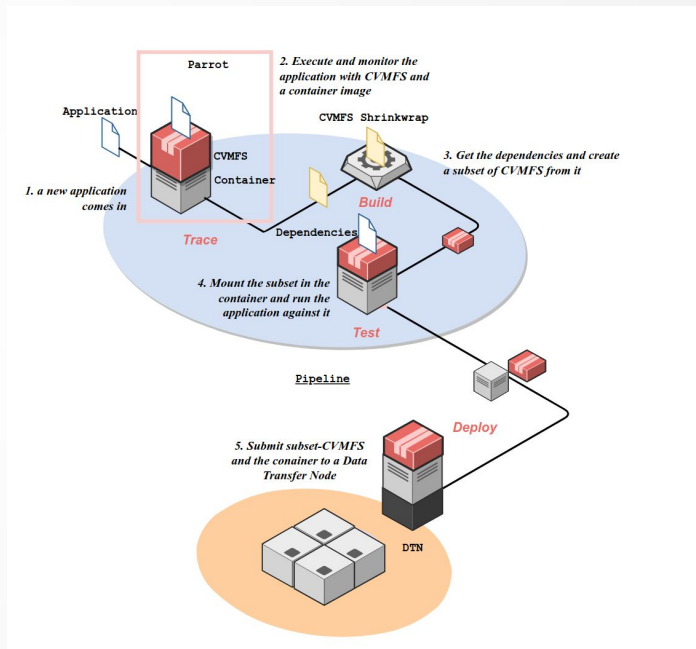
# Outlook HPC

HPC sites still impose many restrictions today. Workarounds for many configurations exist, but come at different levels of cost

Best case:

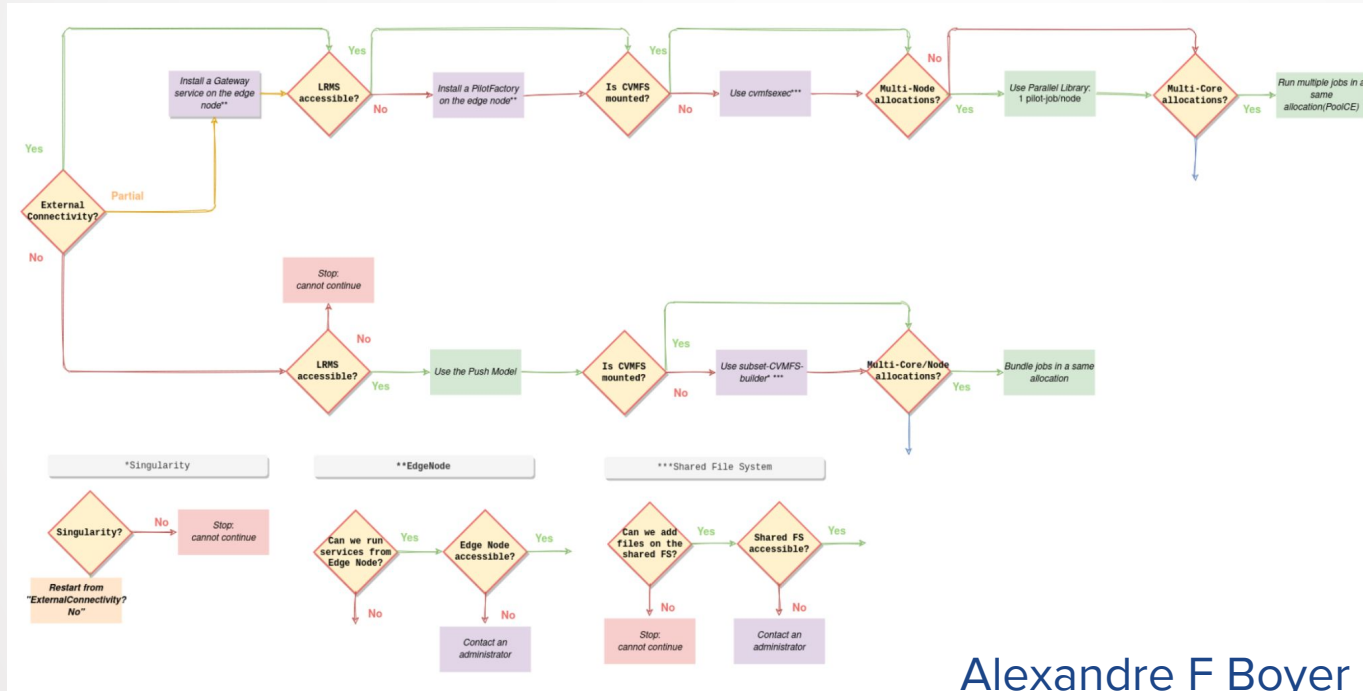


Worst case:



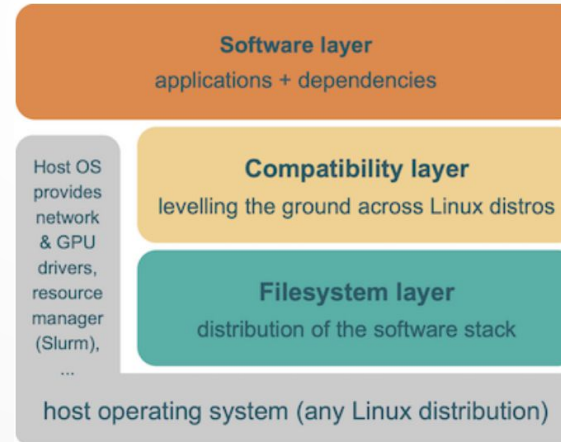
# HPC

Need to find custom solution depending on specific constraint



**HPC sites** can be a particular challenge, with many restrictions.

The CVMFS development team supports the EESSI project, which provides unified software installations to European HPC sites on CVMFS.



# Some focus points in further developments

1. CDN / Proxies / Caching
2. Throughput / performance
3. Availability / robustness / useability

## What would drive a new major release?

- Community interest in new and extended features that conflict with a stable cvmfs2
- Performance optimizations that require extensive changes to the architecture
- Possibility of deprecating features (NFS-exports ...)

# CVMFS for data distribution

- Mixed experience of existing usecases (LIGO, ...)
  - The combination with authentication has made debugging issues particularly complicated
- High-throughput application should likely use a different solution! (XRootD/EOS), if I/O is a major performance factor
- Nevertheless, other usecases exist where using CVMFS infrastructure can be helpful
- We will try to improve performance as possible within the limits of FUSE

# Sidenote: CVMFS development and programming languages

- Currently: C++ (03) for most of the core
- Golang in server / container components (gateway and unpacker)
- Bash for scripts / CLI / configuration

We are “stuck” with cxxstd 03 due to the hotpatching functionality. ABI incompatibility between binaries compiled with different standard means that that for an upgrade, all repositories would have to be remounted. There are ideas for doing further serialization that would allow updates.

CVMFS has previously experimented with new languages (Erlang in a first implementation of the gateway, Javascript). Rust could be a good candidate to use in places for a new version, but would not attempt a rewrite.

# CernVM Appliance



- Focus has clearly shifted to the CernVM-Filesystem
- However, CernVM still has two major users: the **OpenData initiative** and the **ATLAS High Level Trigger**
- CernVM-Five, a container-first platform of CernVM is already available as a prototype
  - Significantly improves maintainability
  - Can be useful in many scenarios; as “standard container image with cvmfs”

# Conclusion

- CVMFS has become an essential ingredient to distributed computing in HEP
- Thanks to good collaboration with grid-site operators and CERN IT-Storage!
- Prioritize reliability, robustness and performance in future developments
  - Some great ideas/new features are there already, but need to be made “production-ready”
- There is a lot of interest in a “data-delivery CVMFS”
  - Can have usecases in non-high-throughput scenarios
- Aim to continue workshops (every 18 months on average)