Introduction to the CernVM-File System

René Meusel, Jakob Blomer

Seminar at CNAF, 4th of December 2014

Agenda

1. What is **CernVM-FS**?

2. **Accessing Repositories:** CernVM-FS Client

3. **Updating Repositories:** CernVM-FS Server

4. From POSIX to CernVM-FS: **Internal Data Management**

5. **CernVM 3:** An Operating System hosted in CernVM-FS

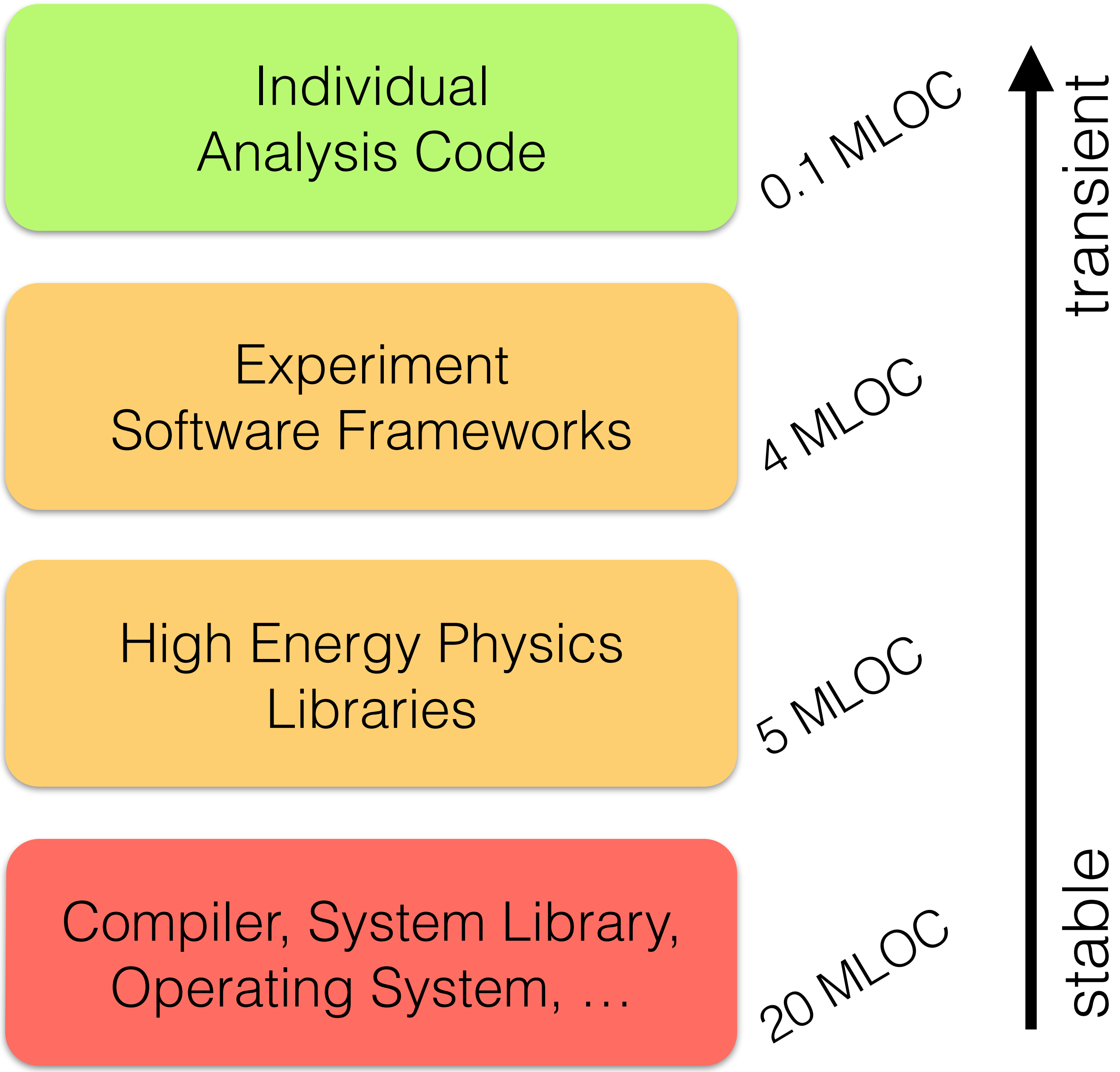
History of CernVM-FS

- Spin-off project of **CernVM 2**
  - Central idea: separation of virtual machine image provisioning and **HEP application software**
  - Using HTTP as transport was the only reliable solution in a diverse environment

---

- NFS shared software area and installation jobs reached scalability and feasibility limits
- PIC (Spain) and RAL (UK) pioneered the usage of CernVM-FS as a replacement technology
- Today: **CernVM-FS** is the preferred way of software distribution in the World wide LHC Computing Grid and other grid infrastructures

Characteristics of HEP Software Packages



Individual Analysis Code

Experiment Software Frameworks

High Energy Physics Libraries

Compiler, System Library, Operating System, …

0.1 MLOC

4 MLOC

5 MLOC

20 MLOC

transient

stable

- Frequent Updates
- Not a single binary - a development environment
- Hundreds of libraries, scripts, binaries, …; with sometimes unclear dependencies
- Hard to separate in modules
- *Not easily packagable*

Characteristics of HEP Software Packages

- **Millions** of file system objects
  (ATLAS Repository: 37M files; 6M directories; 8M symlinks)

- Usually **small file size**
  (ATLAS: average 70 kiB)

- High number of **duplicated files**
  (ATLAS: duplication factor of 9 (sic!))

- Globally **distributed compute resources**

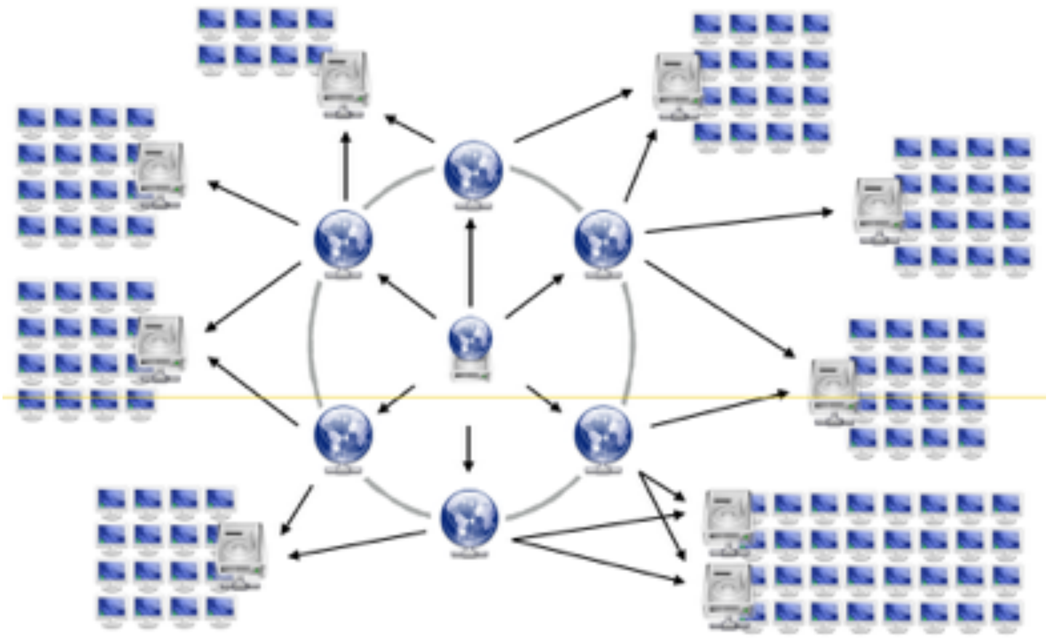
- Highly depends on a specific runtime environment

- Requires **long-term preservation** of software environment

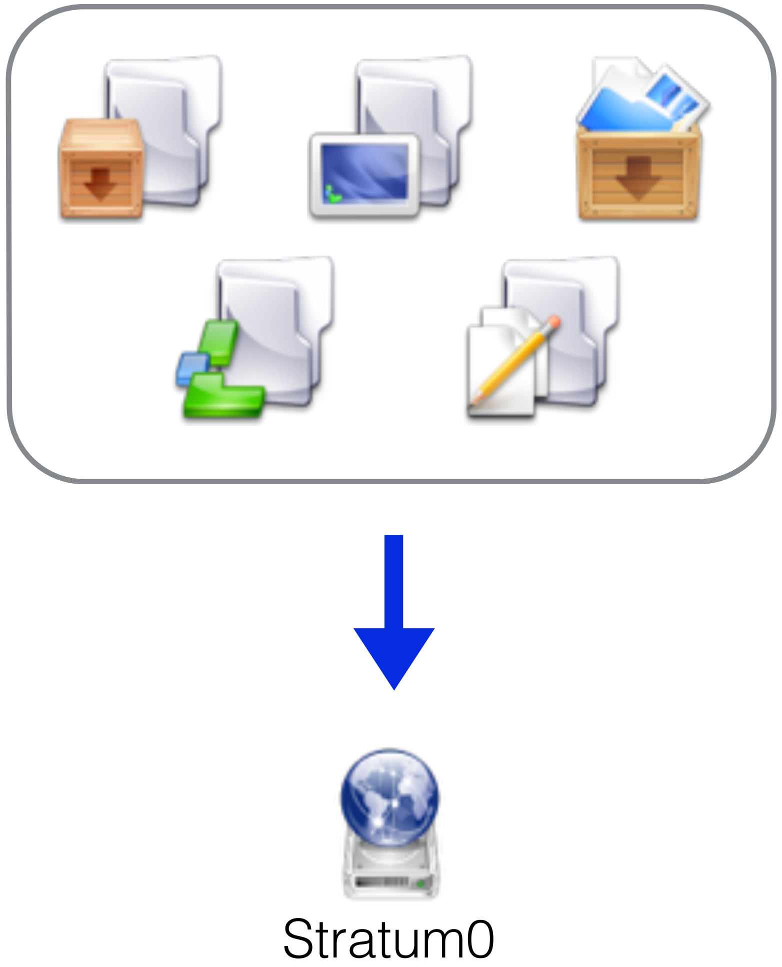
CVMFS in a Nutshell

What is CernVM File System?

- **Scalable software distribution system**
  - Infrequent atomic updates in a central location
  - Read-only access on the clients
  - Repository signed by a trusted release manager
- **HTTP based global data transfer**
  - Minimal protocol requirements
  - Aggressive hierarchical cache strategy
    - Assumption: Coherent working set on physically close nodes (cf. software vs. data distribution)
- **Accessible through a mounted file system (POSIX)**
  - FUSE module, NFS exported FUSE volume or Parrot

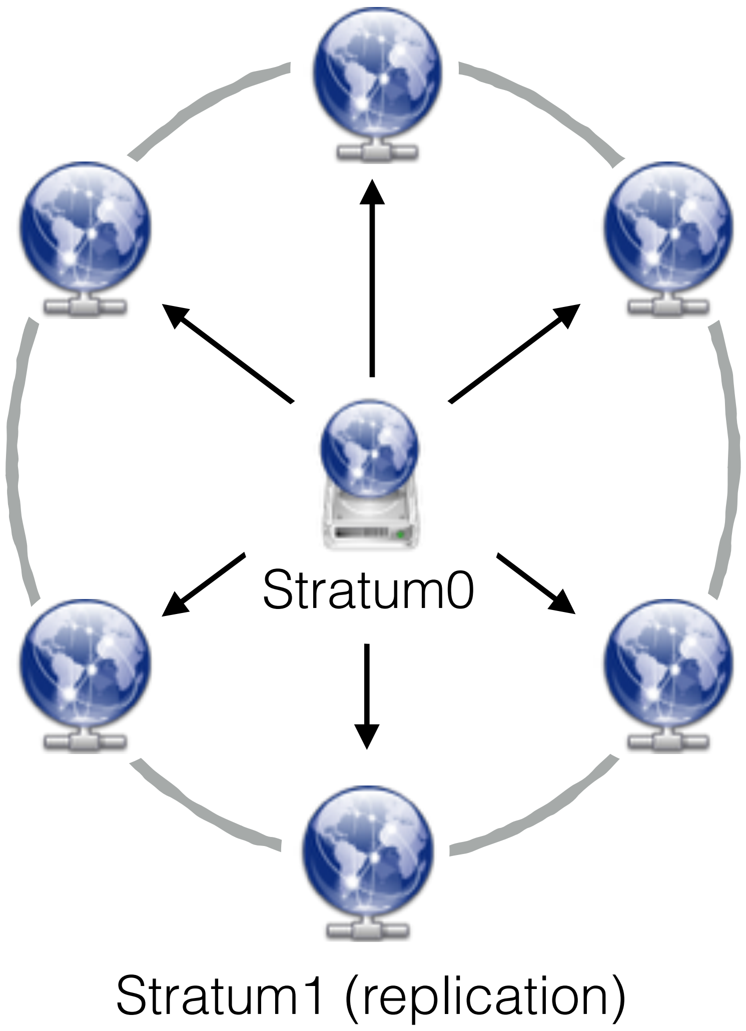


Distributing a Central Installation Worldwide

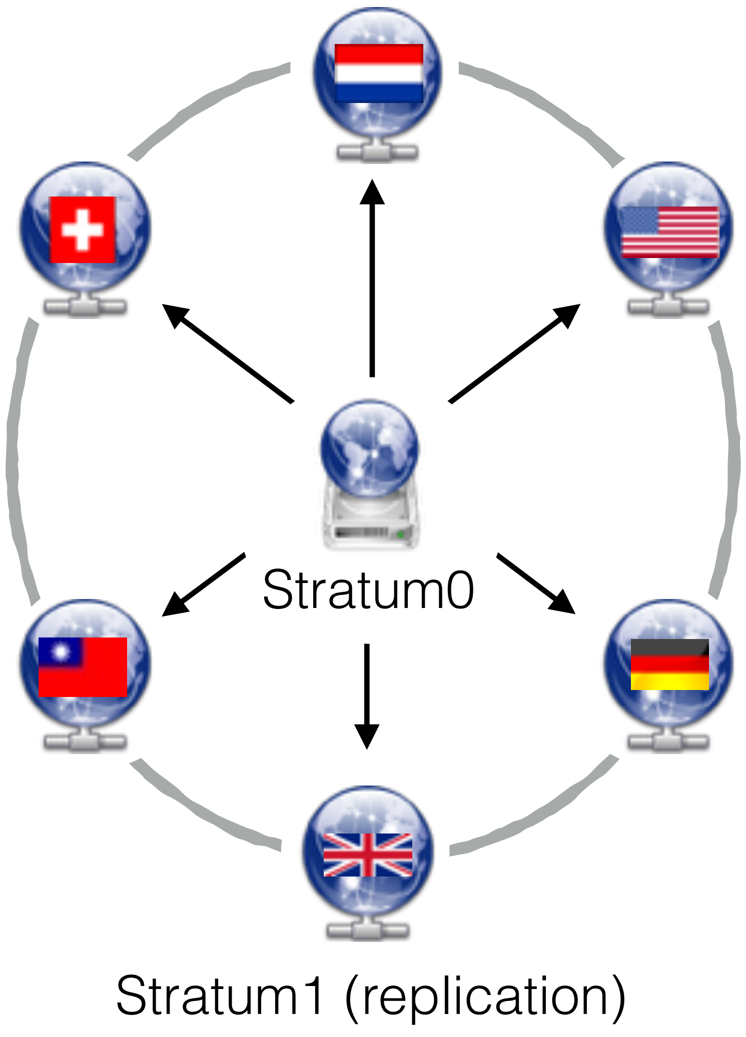


- Modifications happen on **Release Manager Machine** only
- File system snapshots on **Stratum 0** act as seed for distribution
- Globally distributed clients gain **on-demand read-only access**

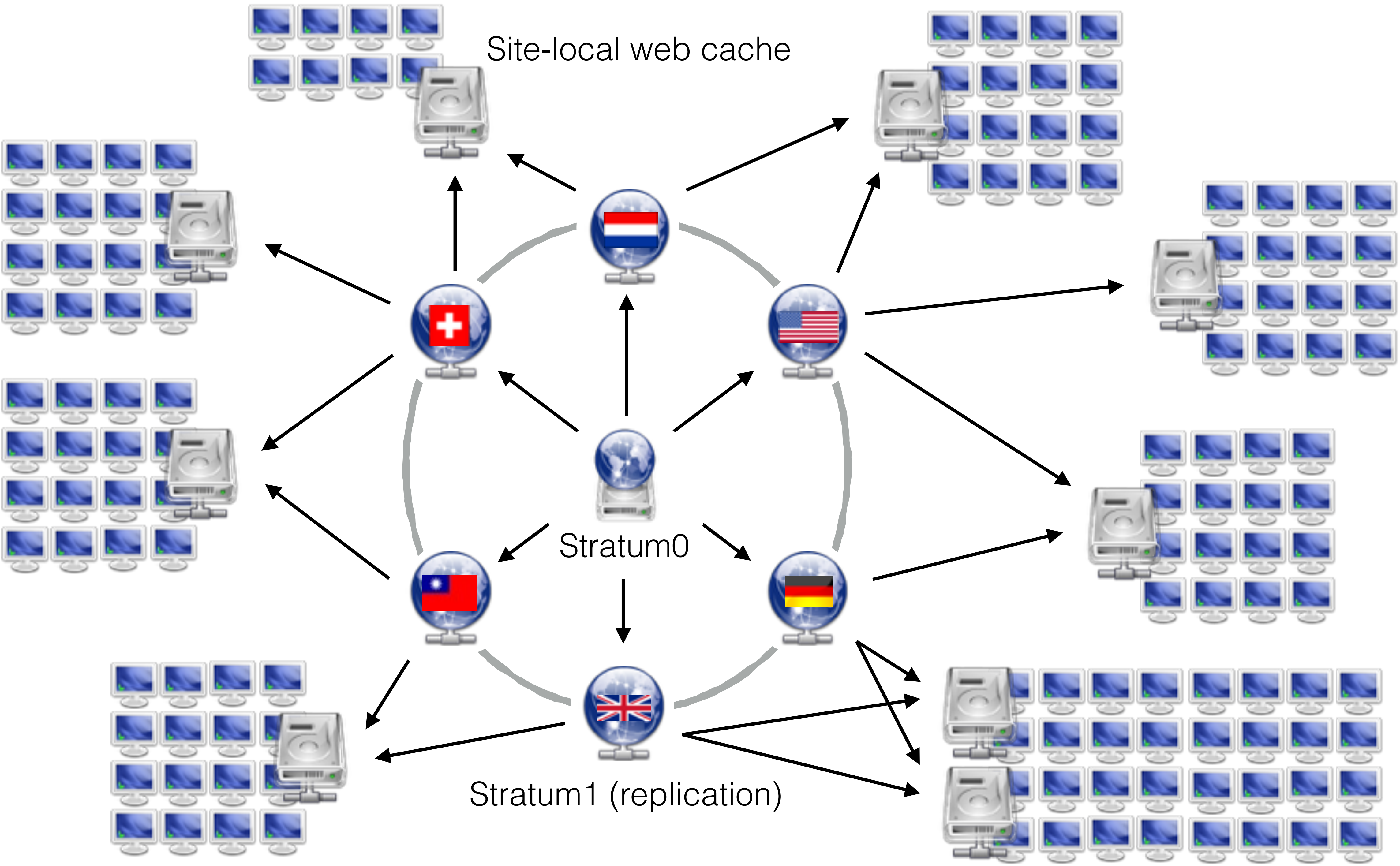
Distributing a Central Installation Worldwide



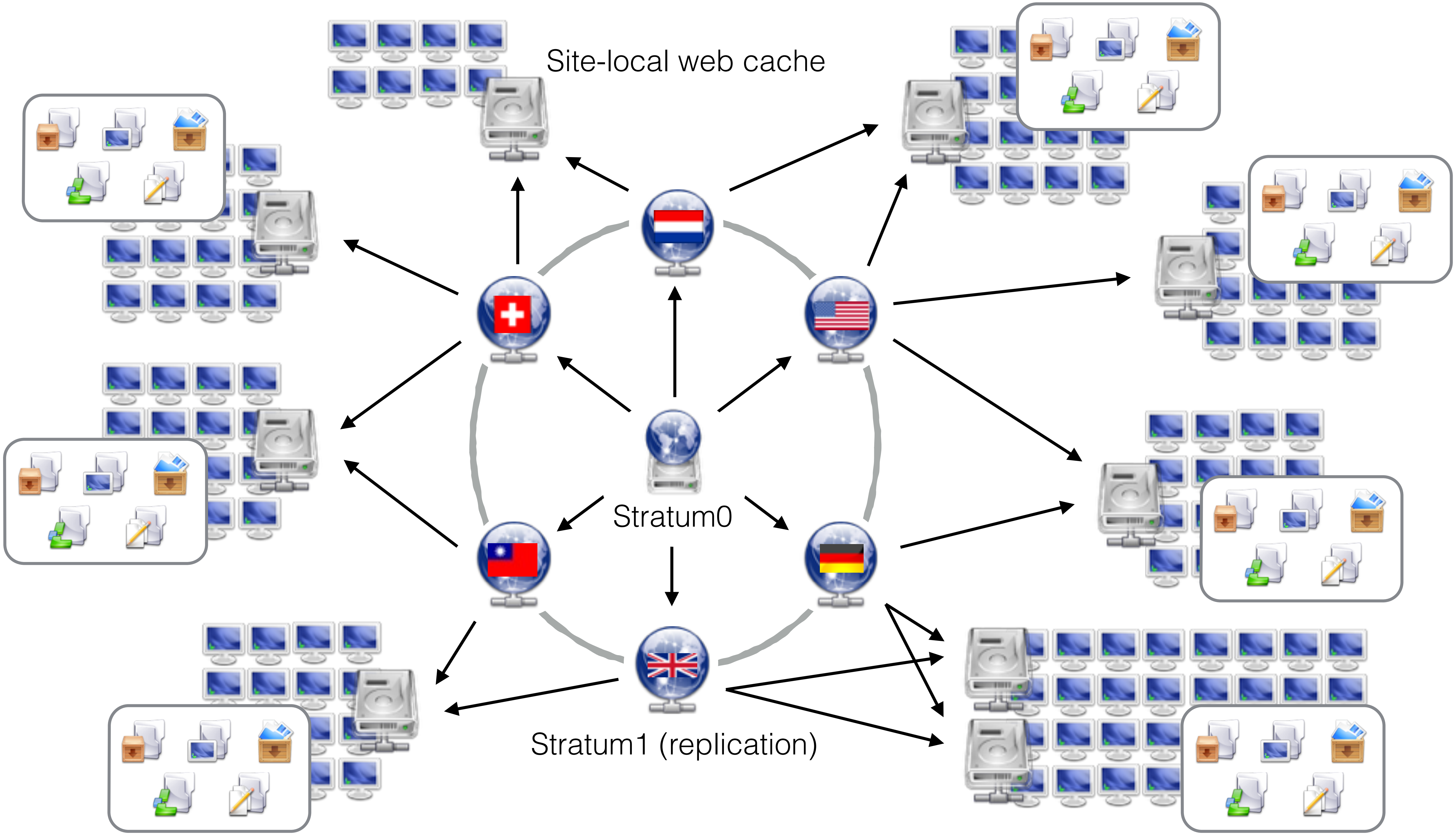
Distributing a Central Installation Worldwide



Distributing a Central Installation Worldwide



Distributing a Central Installation Worldwide



CERN-hosted Repository Statistics

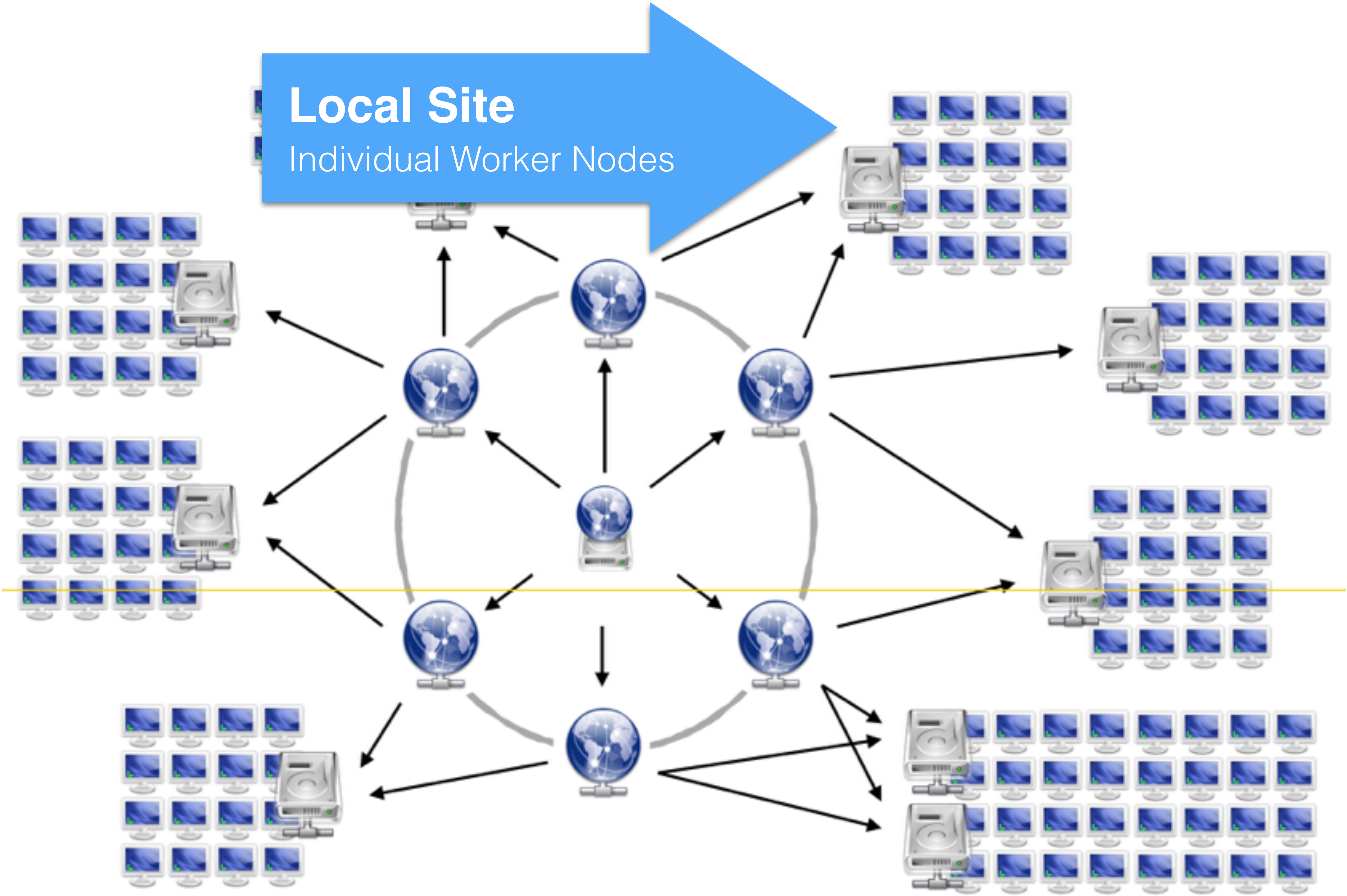
Repository	Files	Refer. Objects	Volume	ø File Size	
atlas.cern.ch	37'000'000	4'000'000	2.4 TiB	68.3 kiB	Mainly Software
cms.cern.ch	34'500'000	5’400'000	1.0 TiB	31.7 kiB	
lhcb.cern.ch	13’600'000	4'700'000	0.5 TiB	43.1 kiB	
alice.cern.ch	7'800'000	280'000	0.7 TiB	92.6 kiB	
ams.cern.ch	3'400'000	2'400'000	2.0 TiB	0.6 MiB	Software + Conditions Data
alice-ocdb.cern.ch	700'000	700'000	0.1 TiB	0.2 MiB	Conditions Data
atlas-condb.cern.ch	8'000	9'000	0.5 TiB	60.9 MiB	

- *Files* and *Volume* as saved in the CernVM-FS catalogs
- Actual number of *Referenced Objects* is compressed and de-duplicated
- Based on latest revision - no history involved

(Effective: November 2014)

CernVM-FS Client
Accessing Repositories

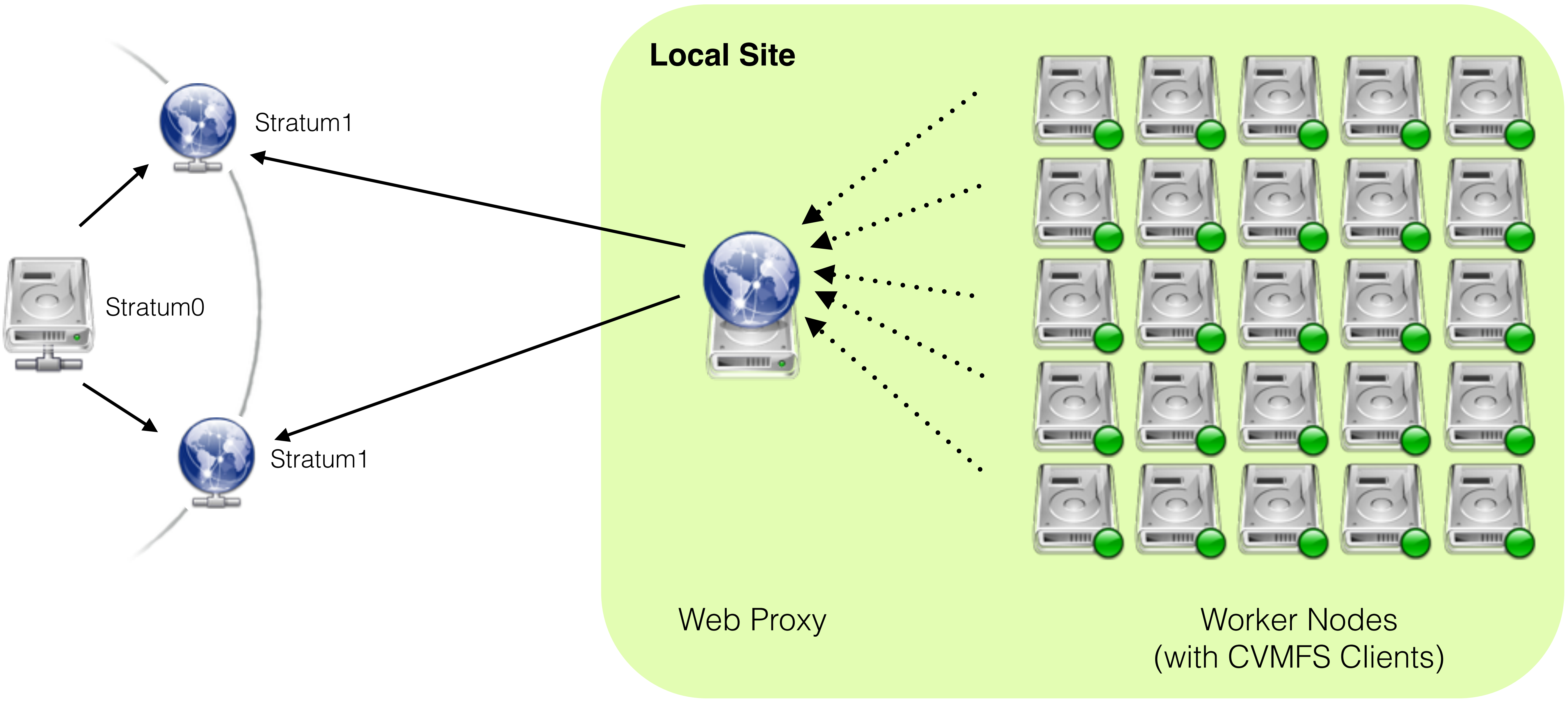
CernVM-FS Client Setup and Architecture



Local Site

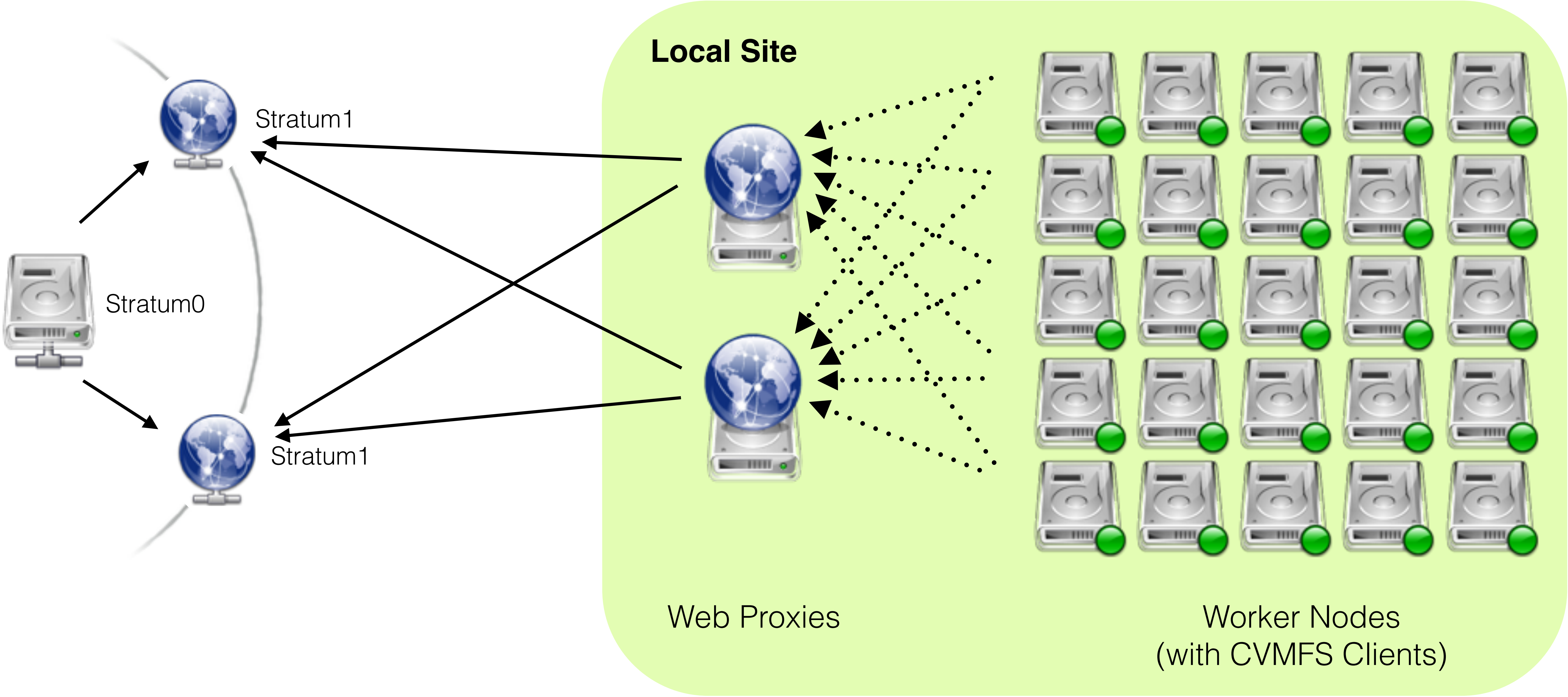
Individual Worker Nodes

CernVM-FS on Each Worker Node



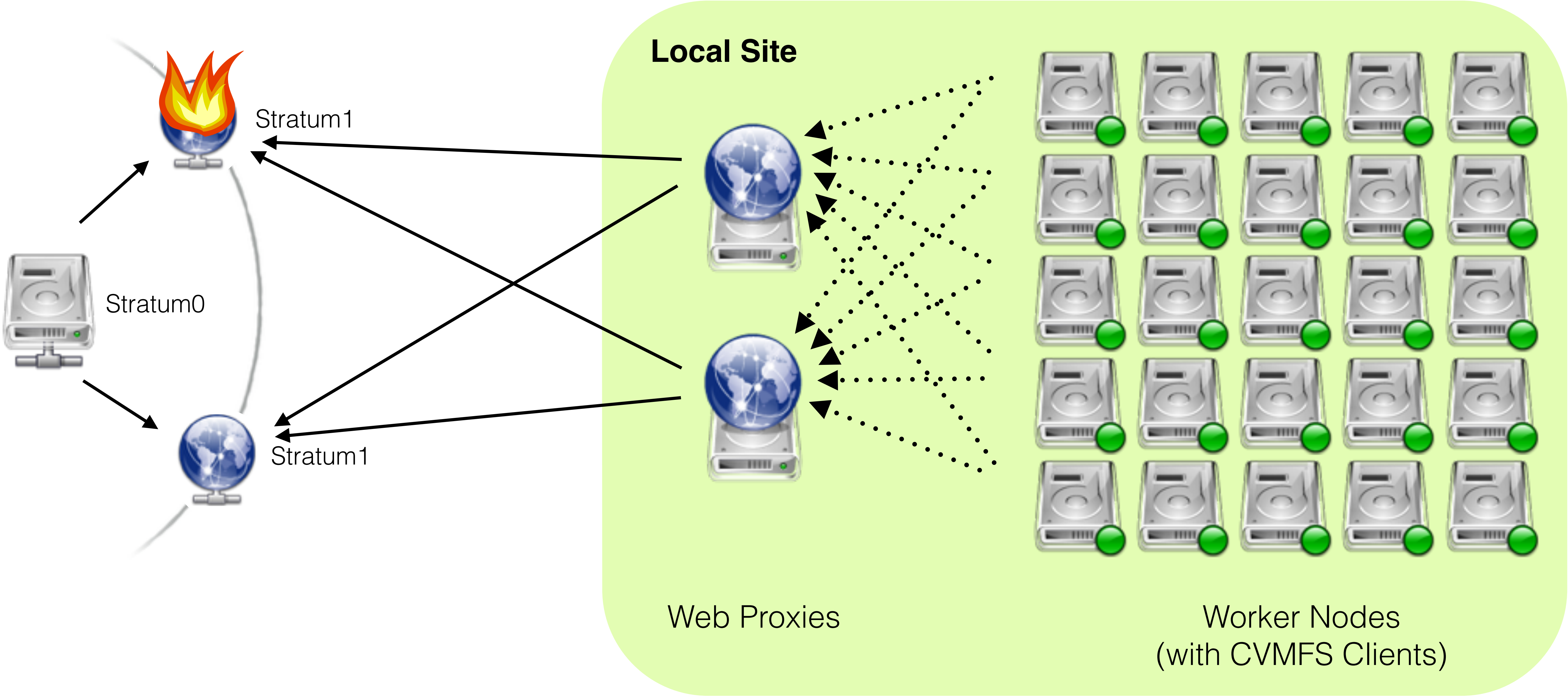
- CernVM-FS mounts as a FUSE module
- Most common approach in WLCG sites
- Local file system caches on each worker node

Scaling and Failover Scenario



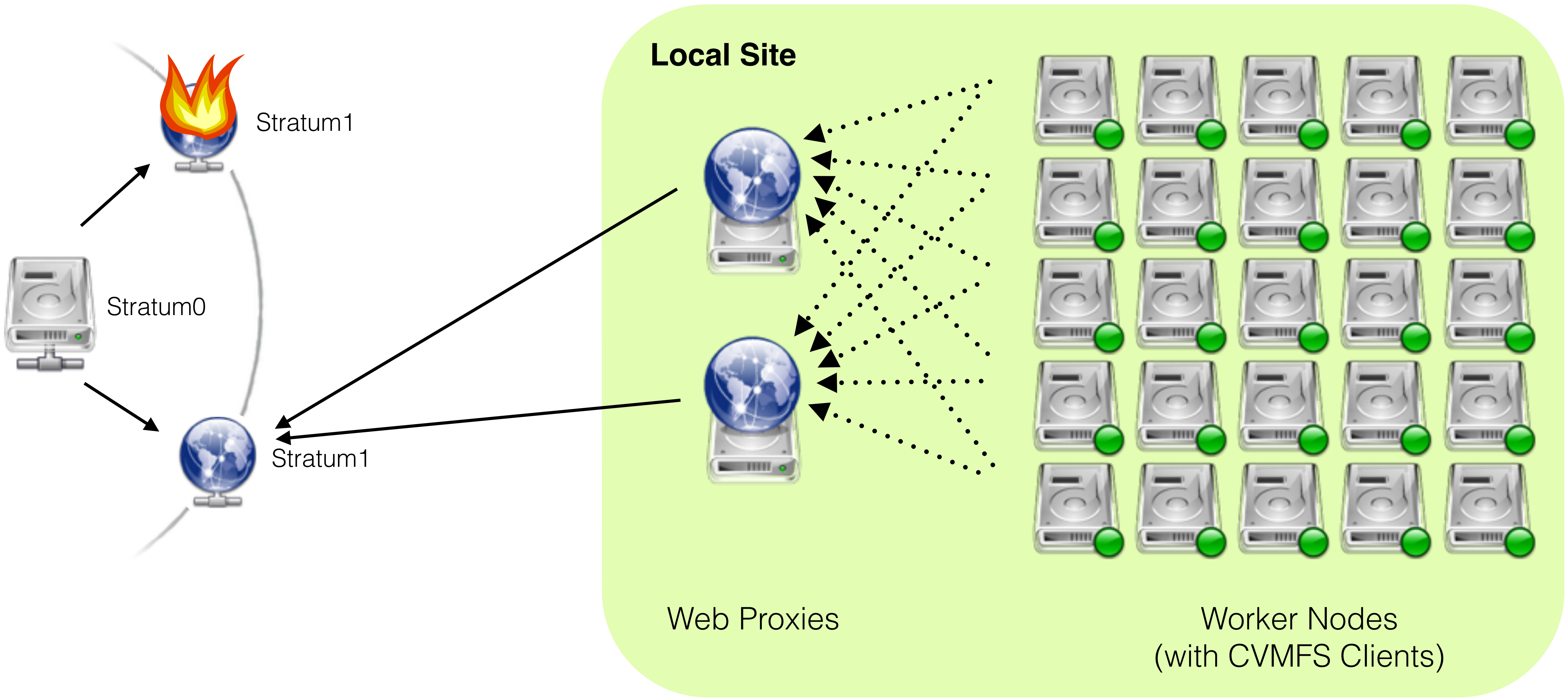
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

Scaling and Failover Scenario



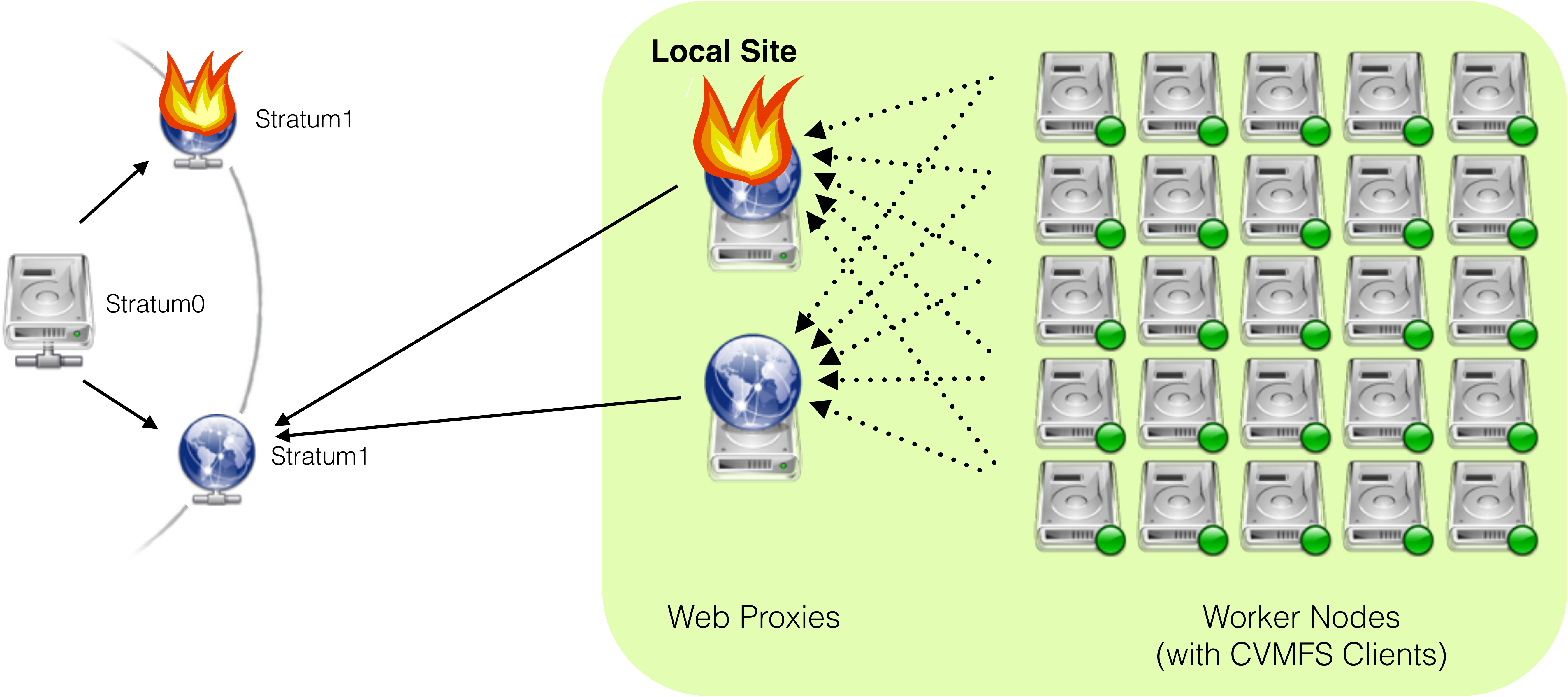
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

Scaling and Failover Scenario



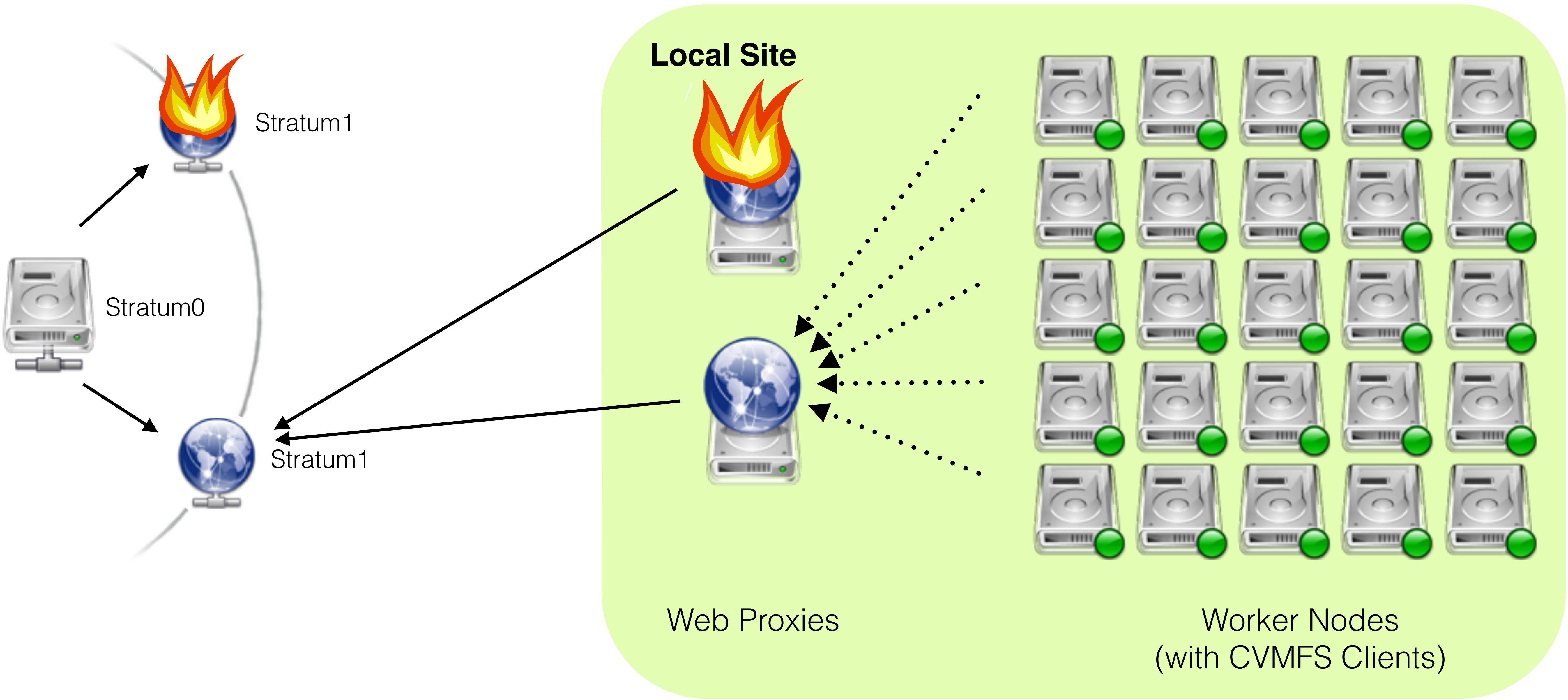
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

Scaling and Failover Scenario



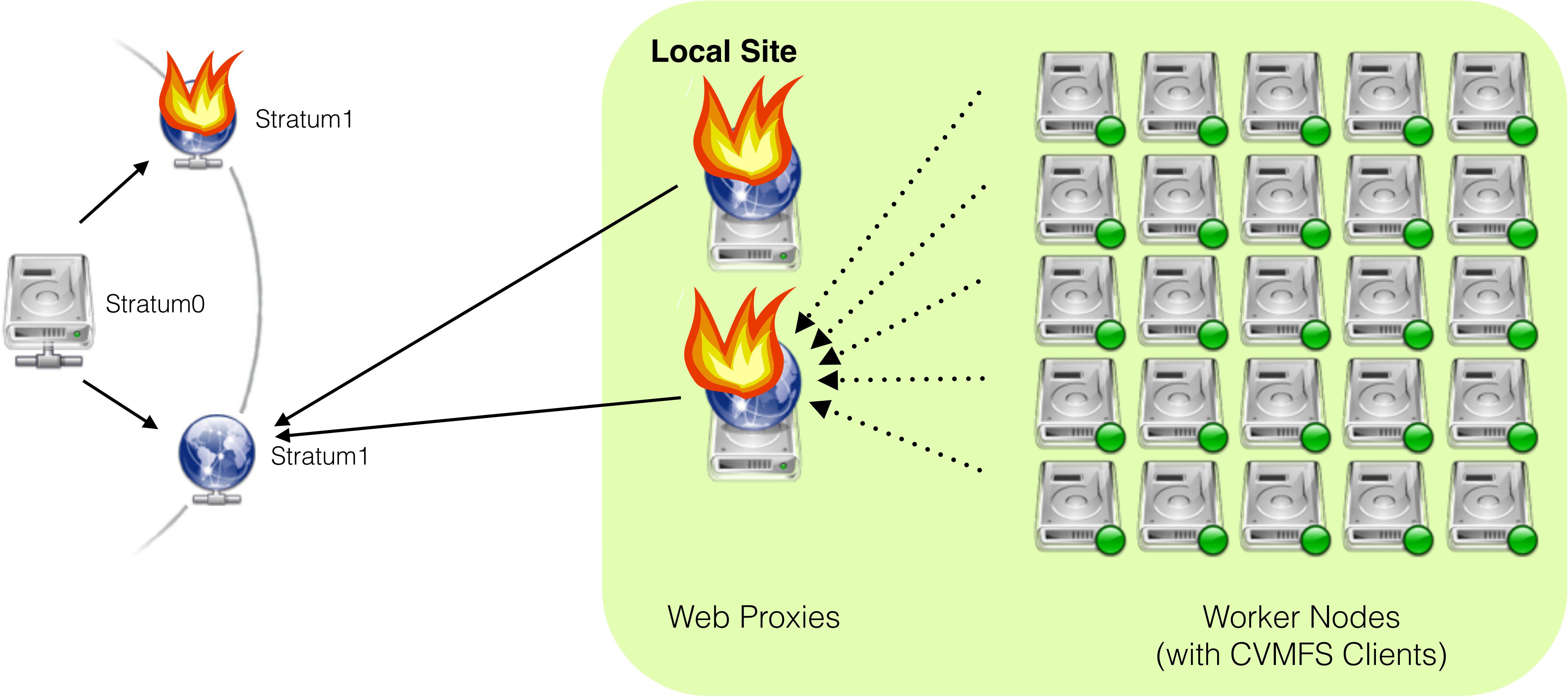
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

Scaling and Failover Scenario



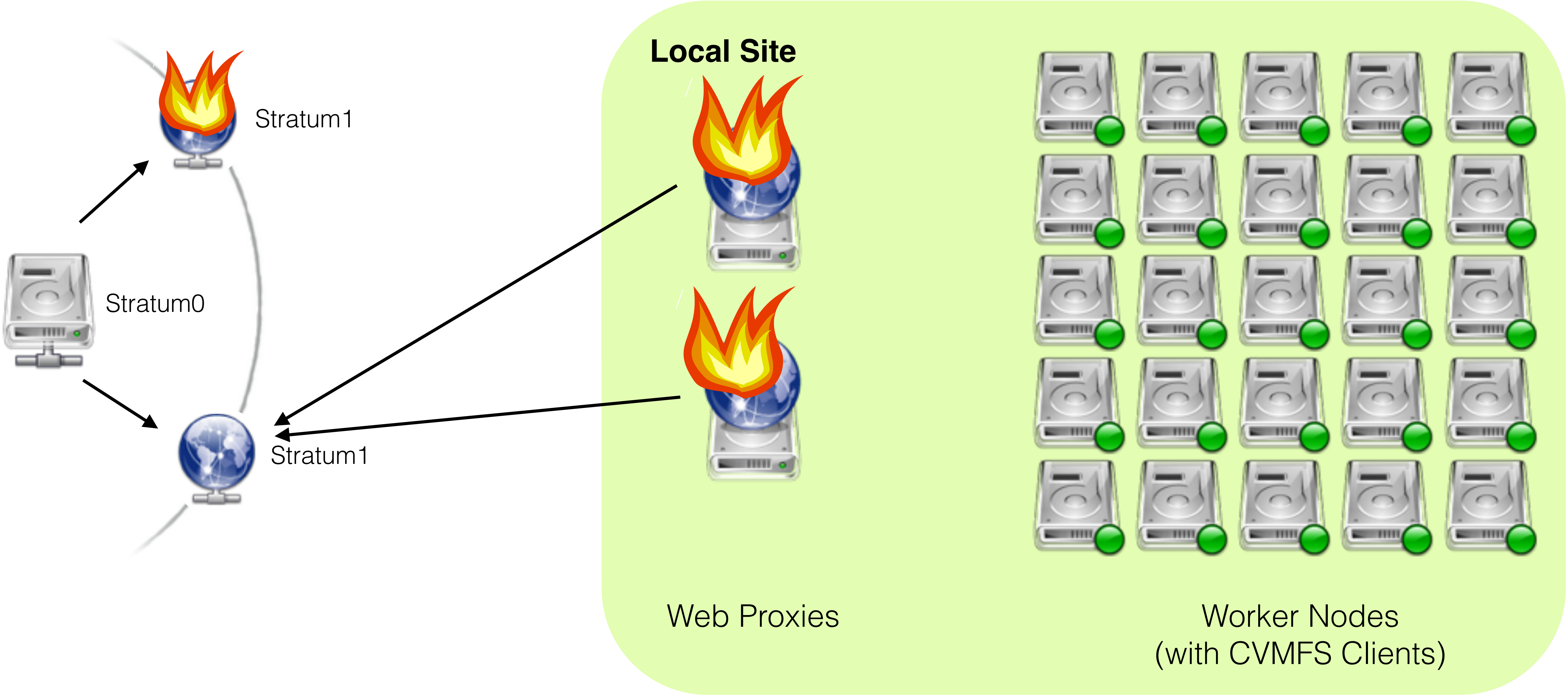
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

Scaling and Failover Scenario



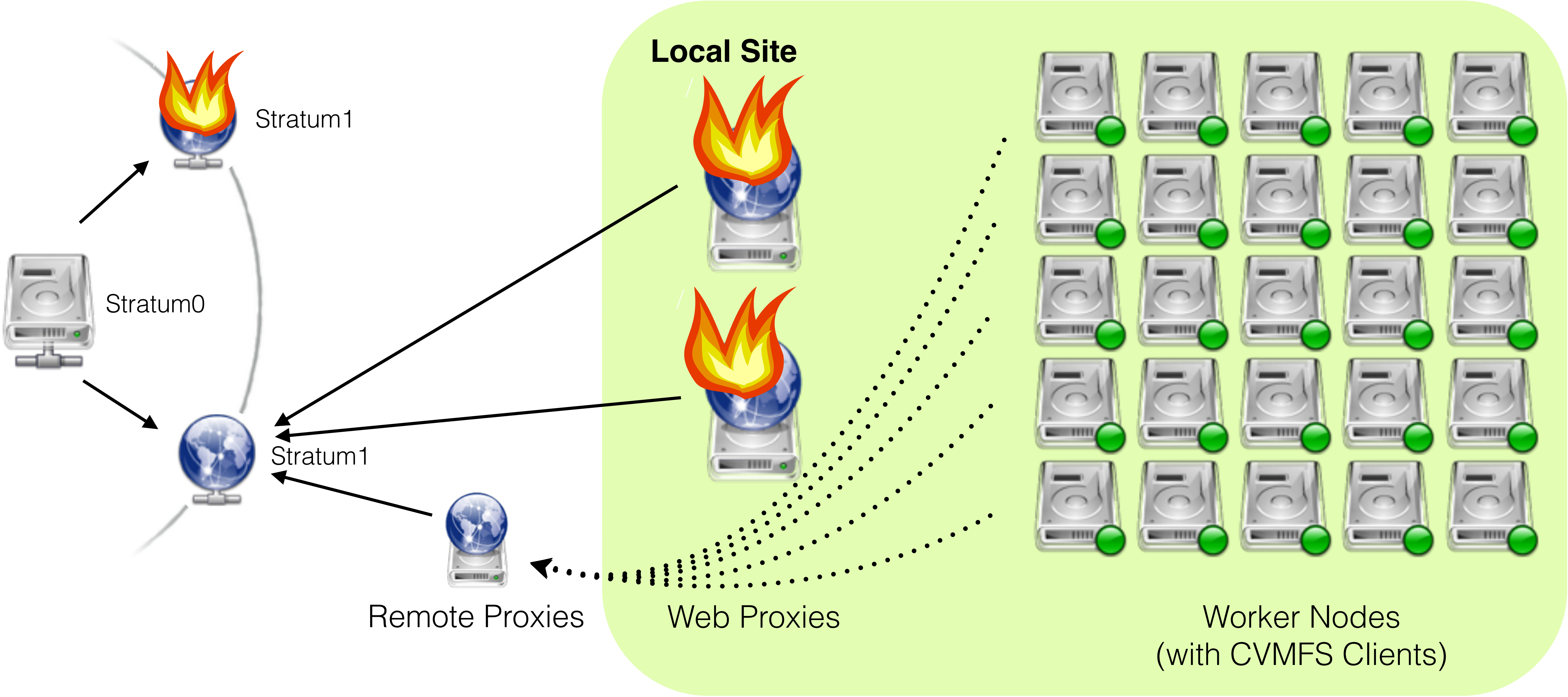
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

Scaling and Failover Scenario



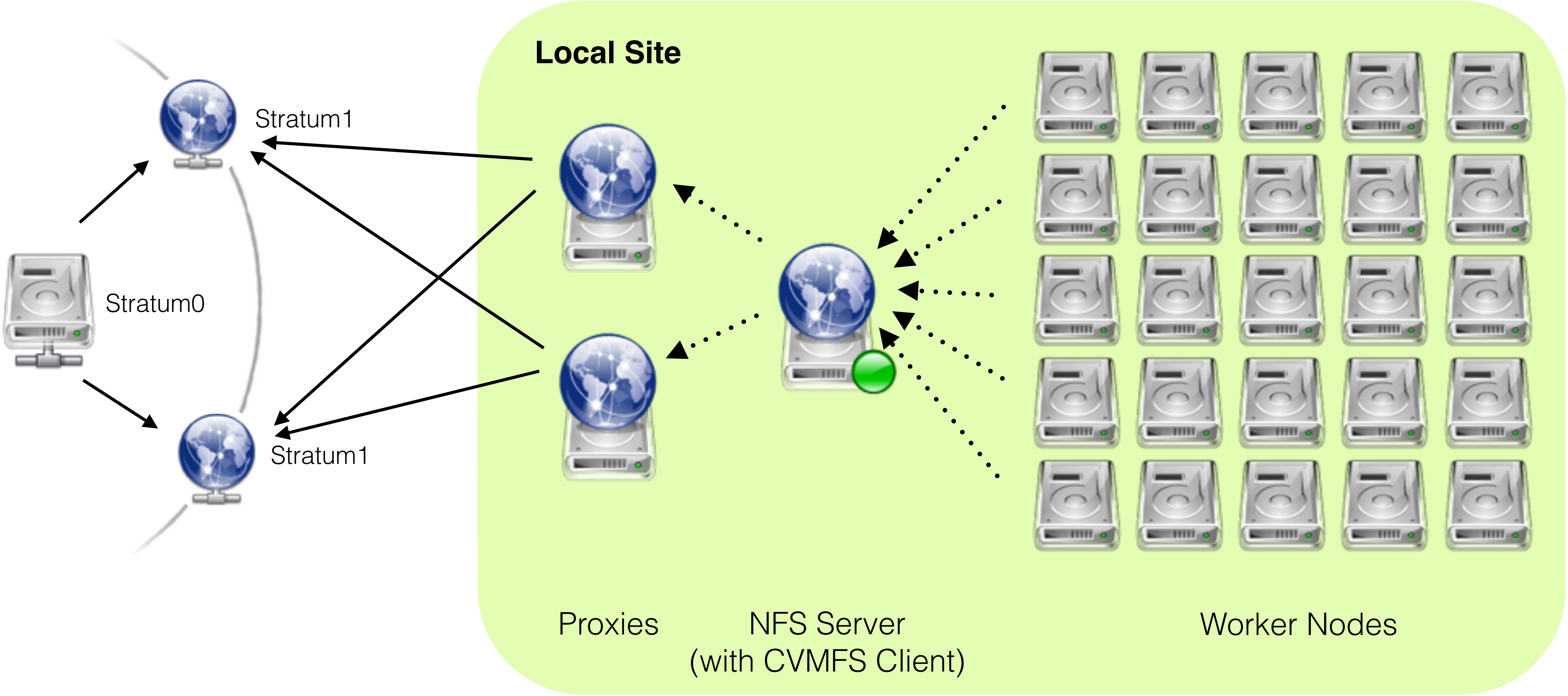
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

Scaling and Failover Scenario



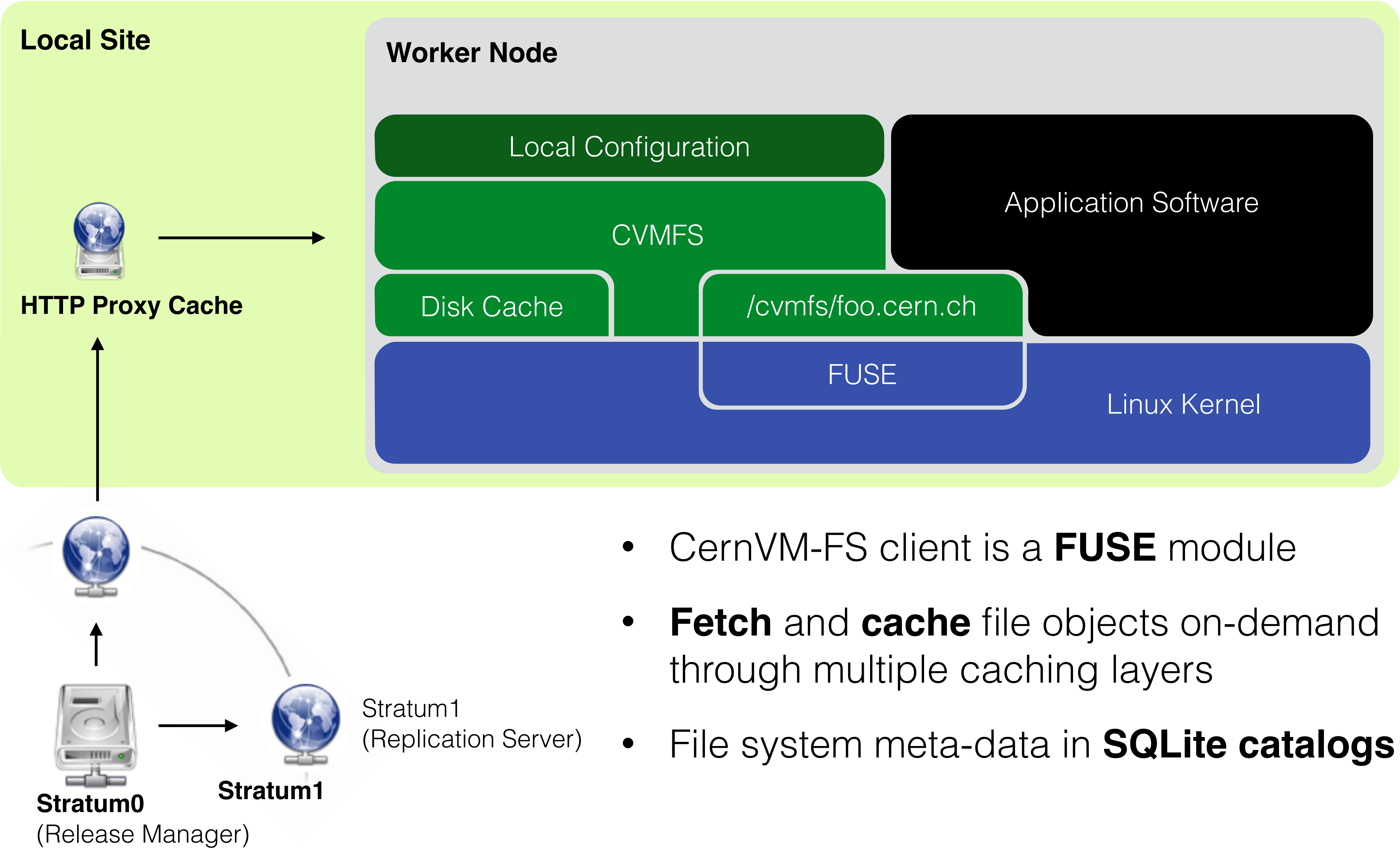
- Horizontal Scaling: Installing multiple web proxy caches
- Fail-over on local/remote proxies and Stratum 1 replicas
- Rule of Thumb: 1 proxy per 50-100 CernVM-FS clients

CernVM-FS through NFS Export



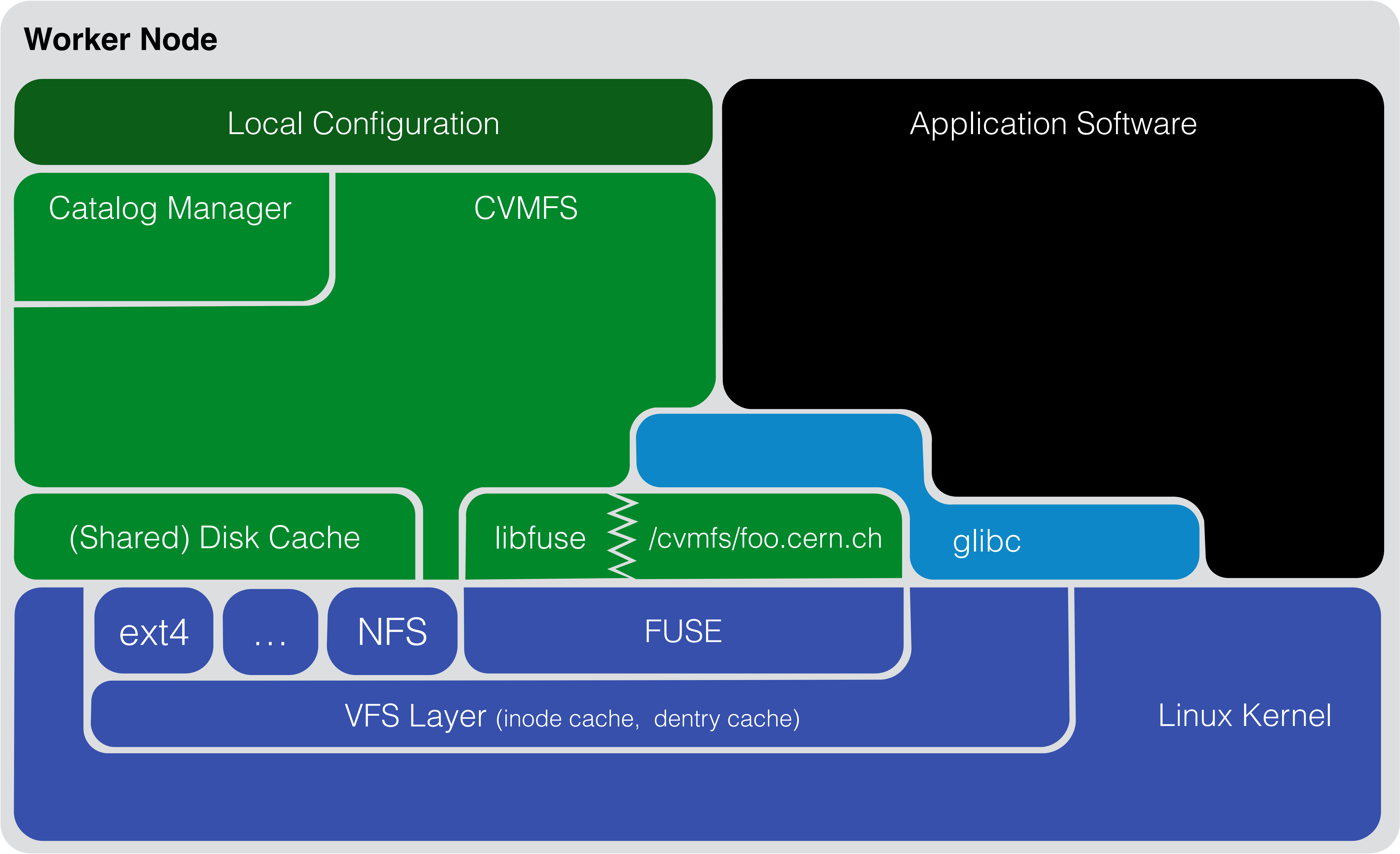
- NFS-exported FUSE module (bottle neck / single point of failure)
- Allows for diskless worker nodes
- DESY: serves 2k nodes with CernVM-FS over NFS

CernVM-FS Client Architecture

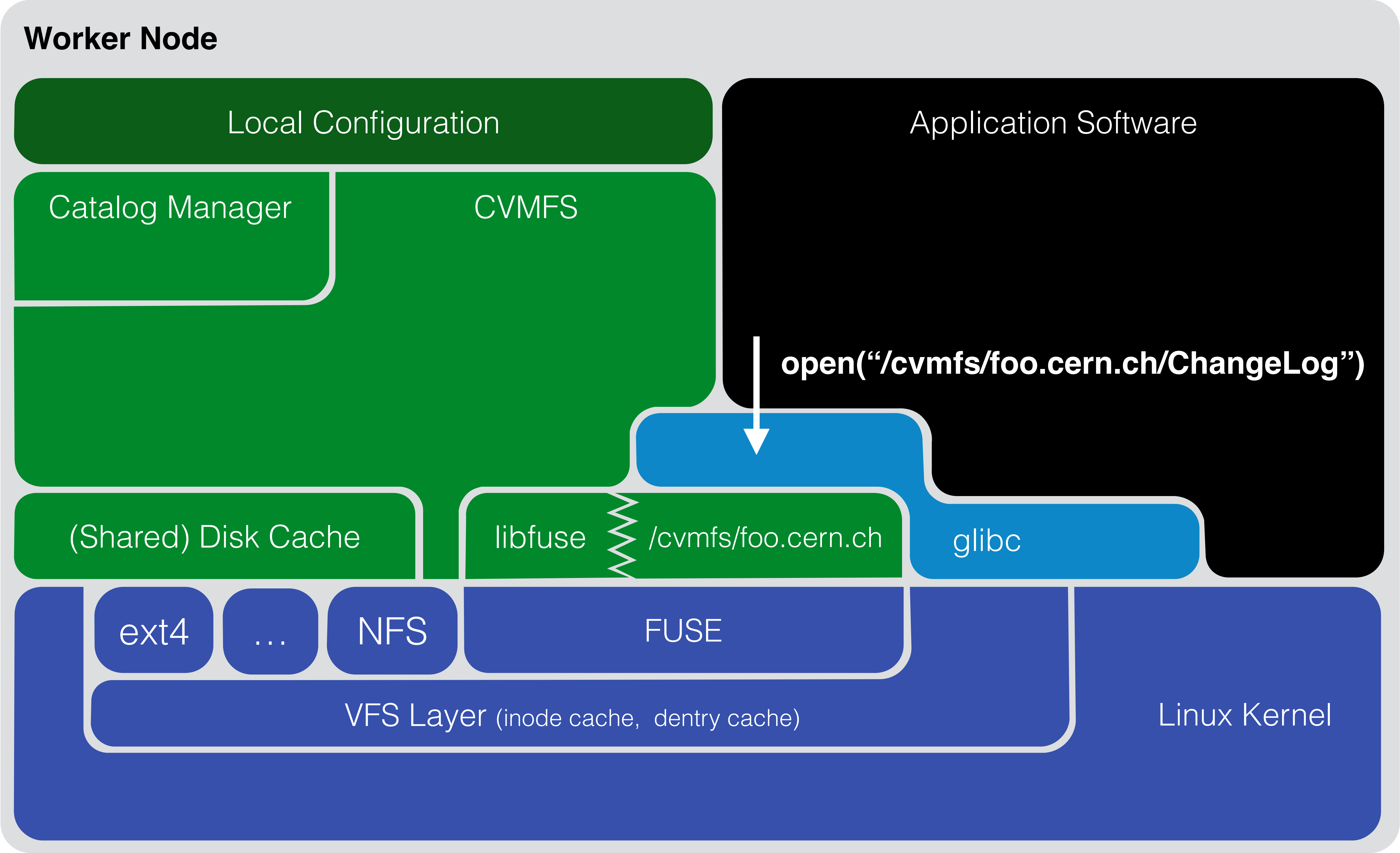


- CernVM-FS client is a **FUSE** module
- **Fetch** and **cache** file objects on-demand through multiple caching layers
- File system meta-data in **SQLite catalogs**

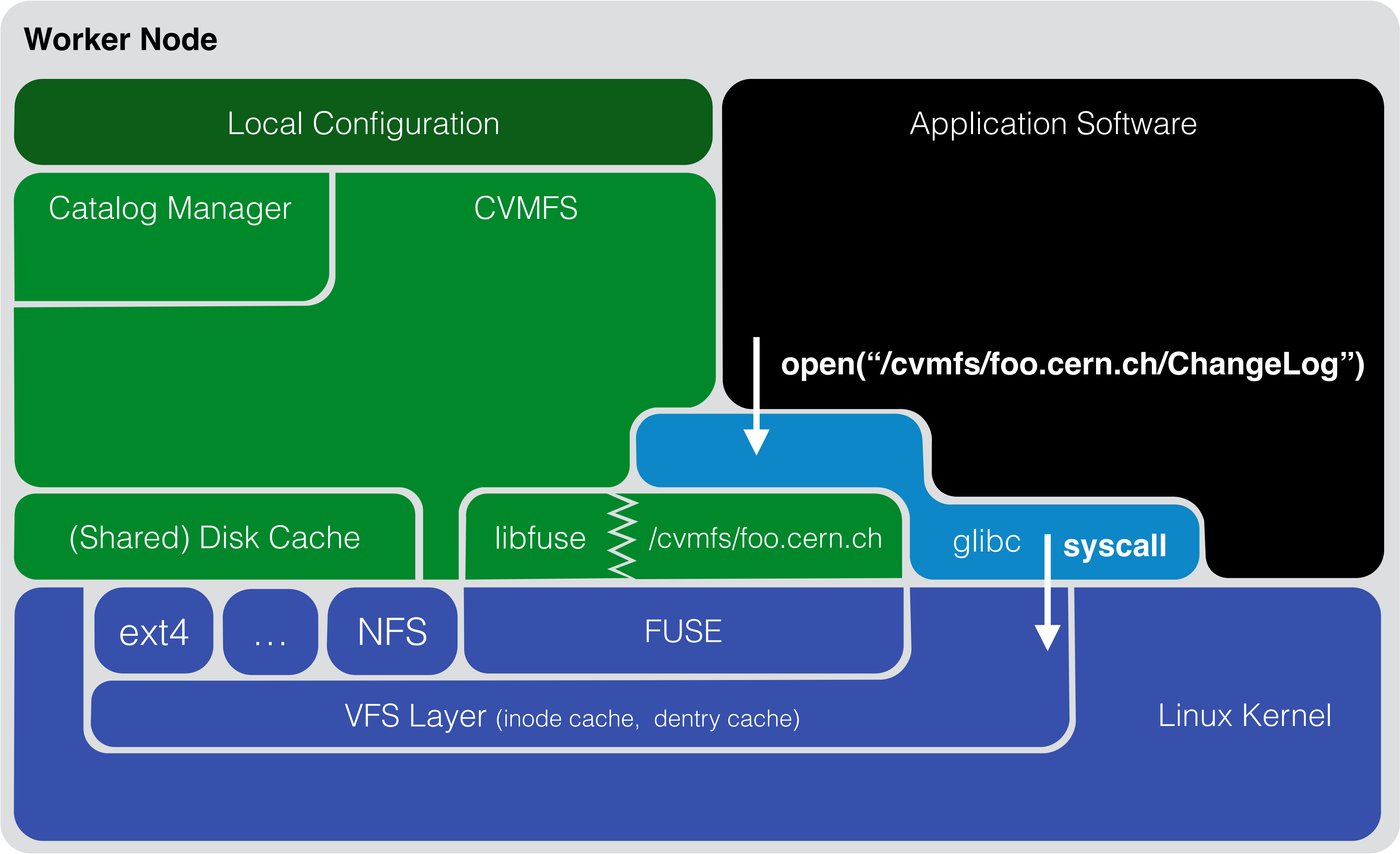
CernVM-FS Client - Control Flow



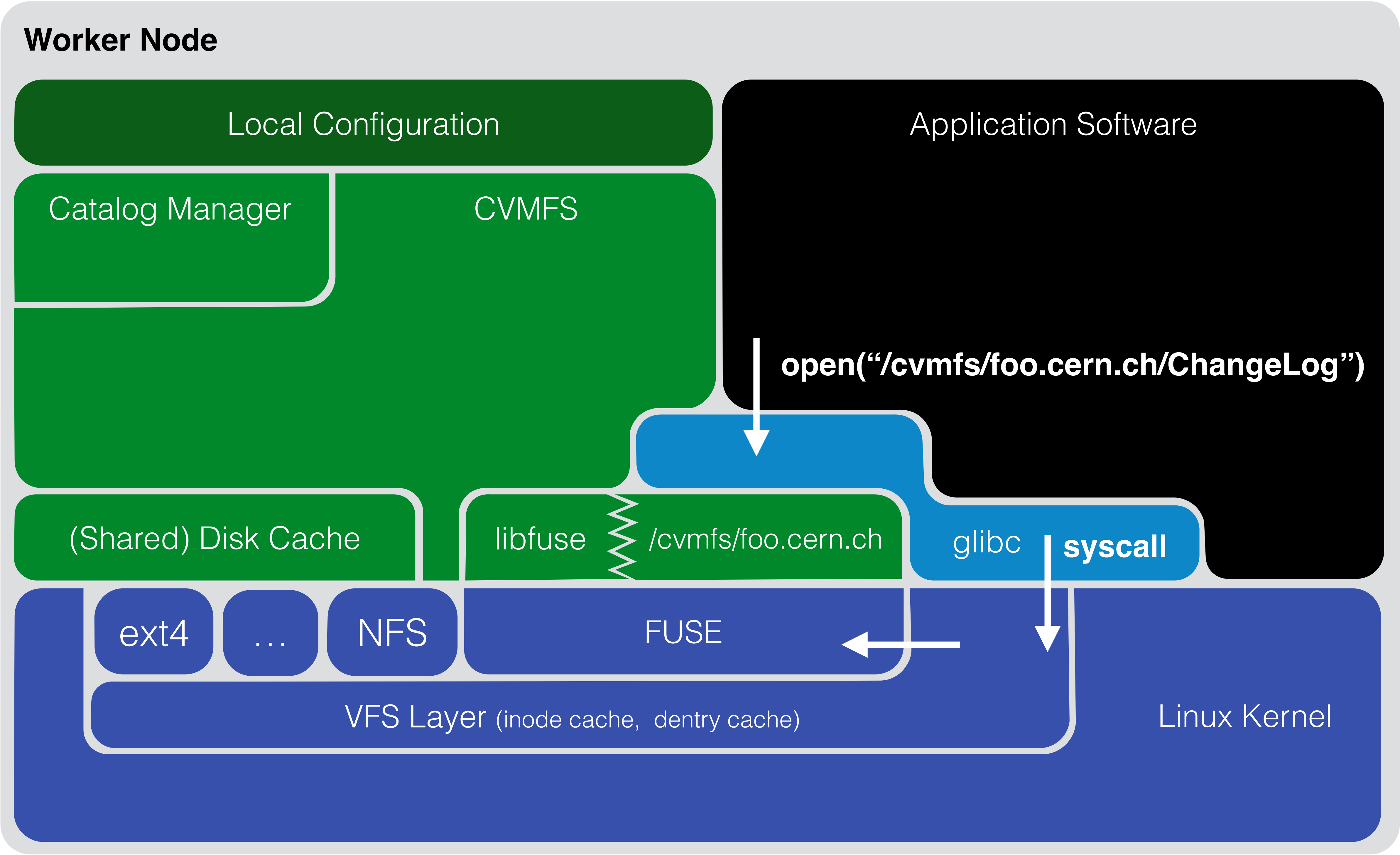
CernVM-FS Client - Control Flow



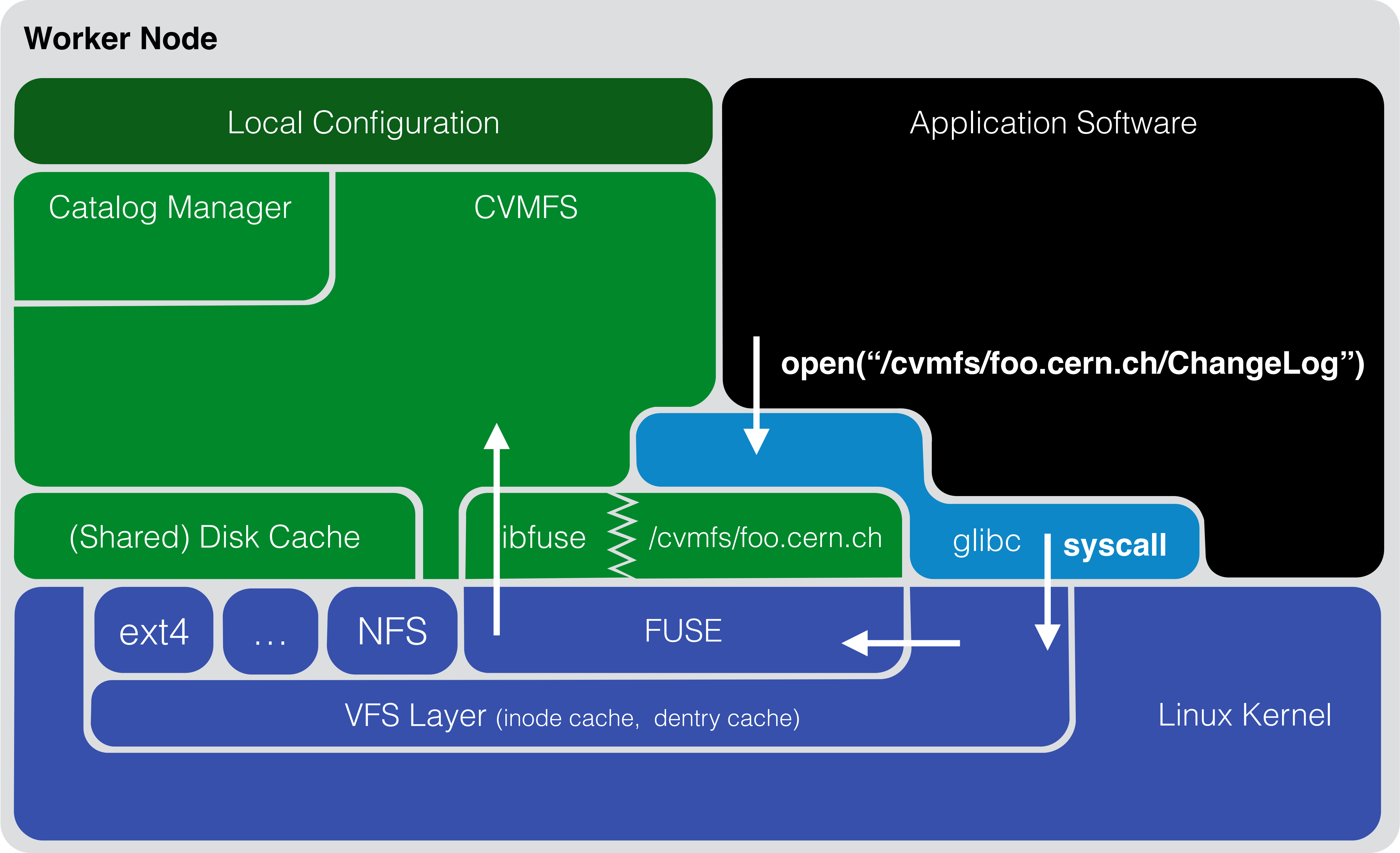
CernVM-FS Client - Control Flow



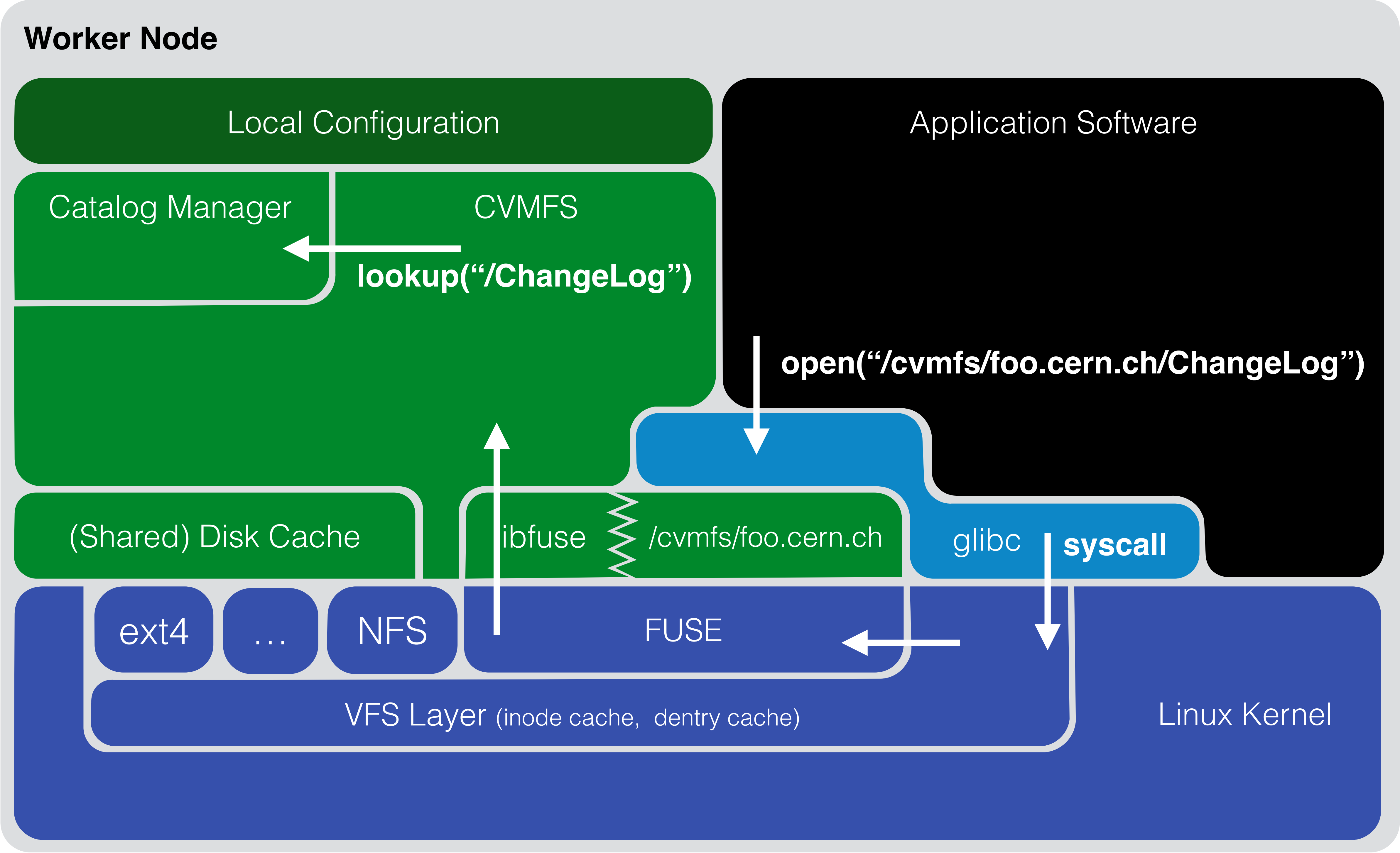
CernVM-FS Client - Control Flow



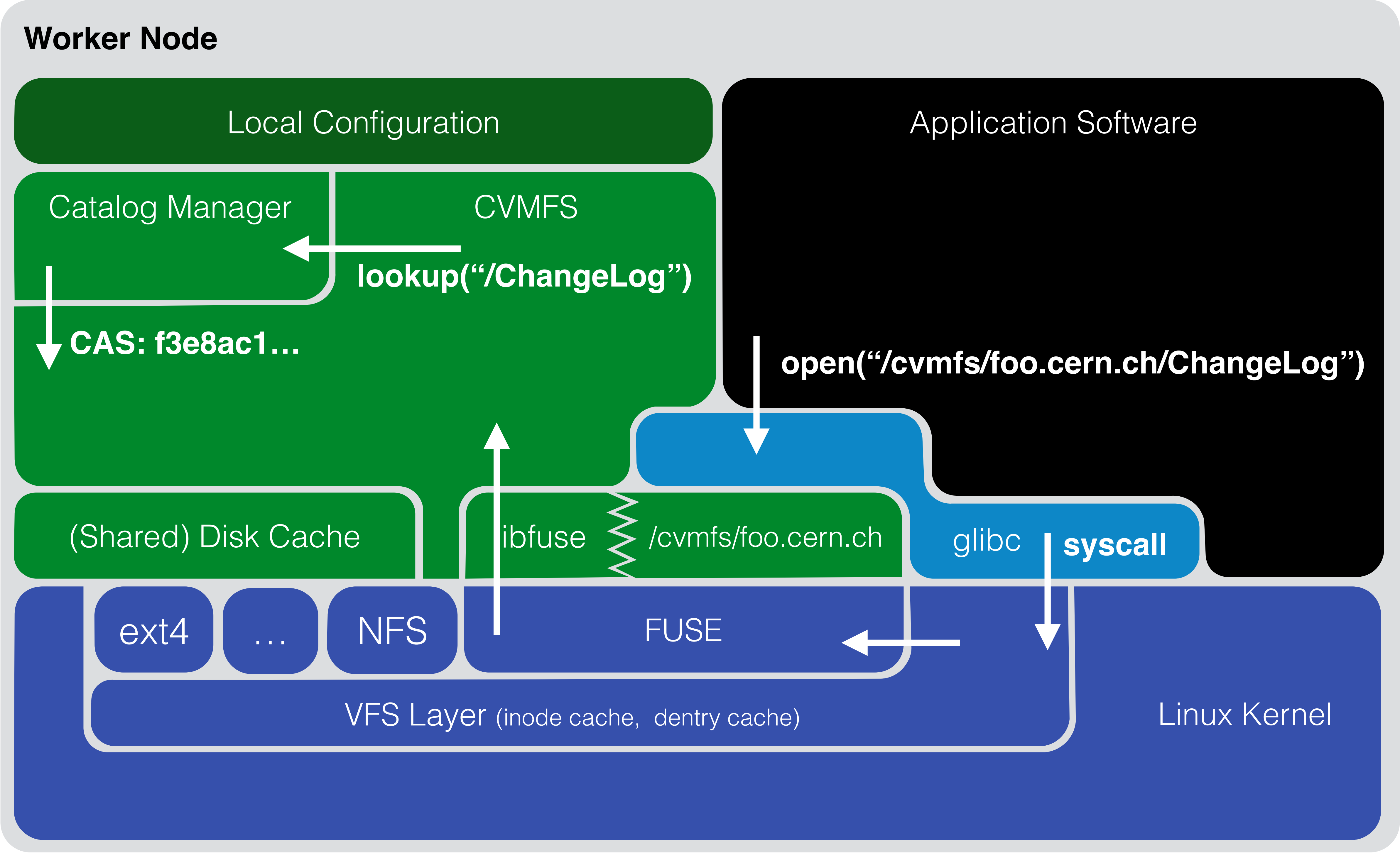
CernVM-FS Client - Control Flow



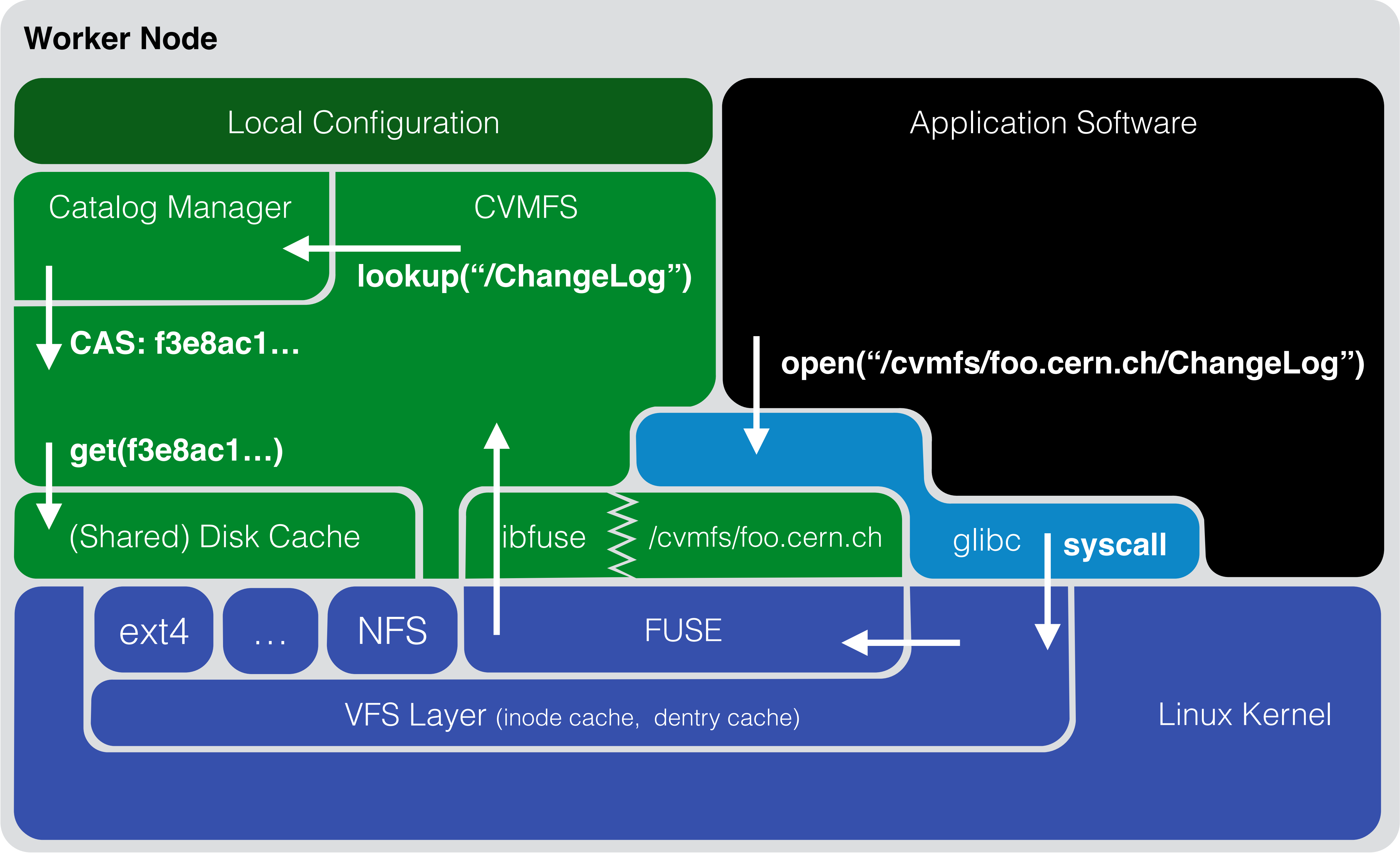
CernVM-FS Client - Control Flow



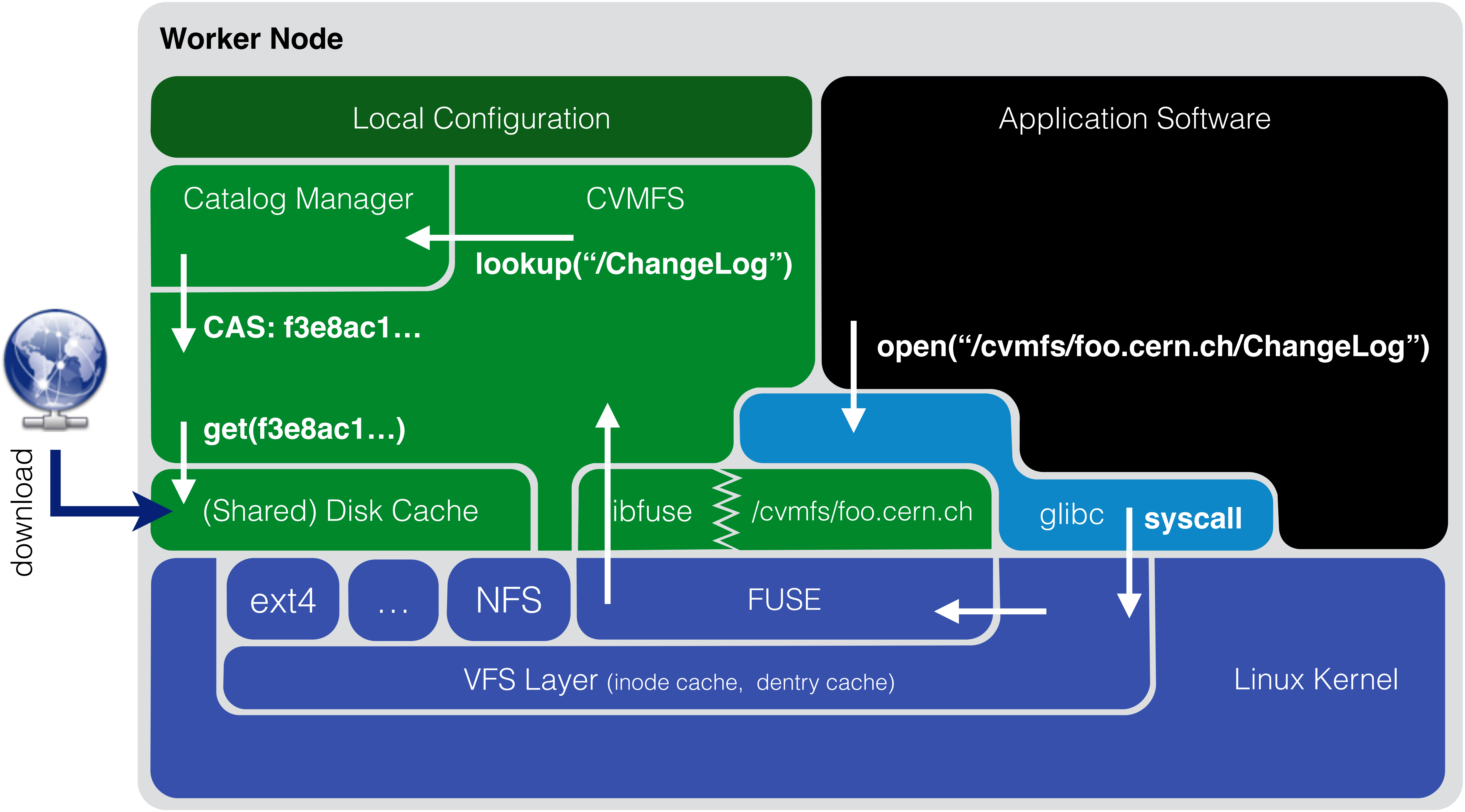
CernVM-FS Client - Control Flow



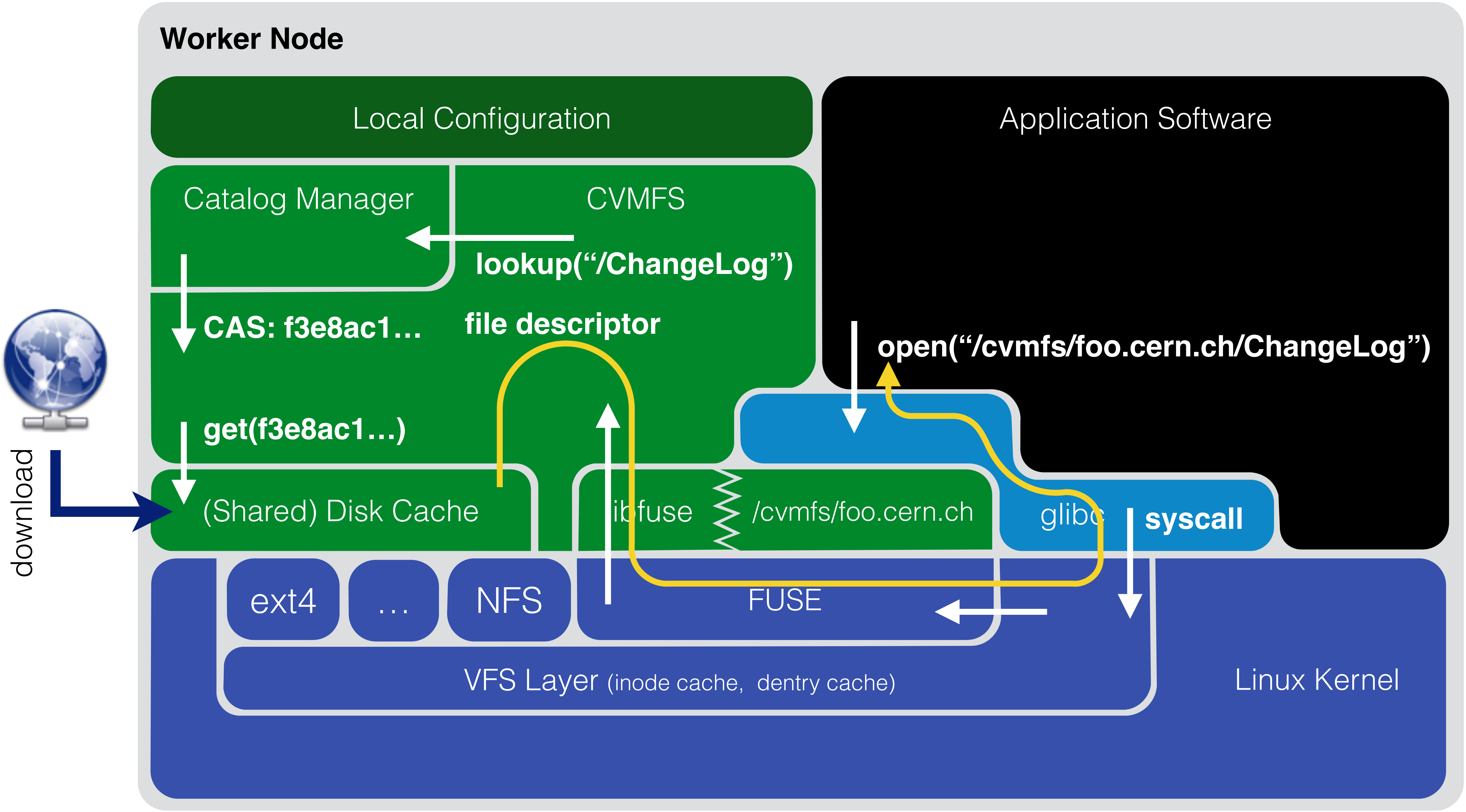
CernVM-FS Client - Control Flow



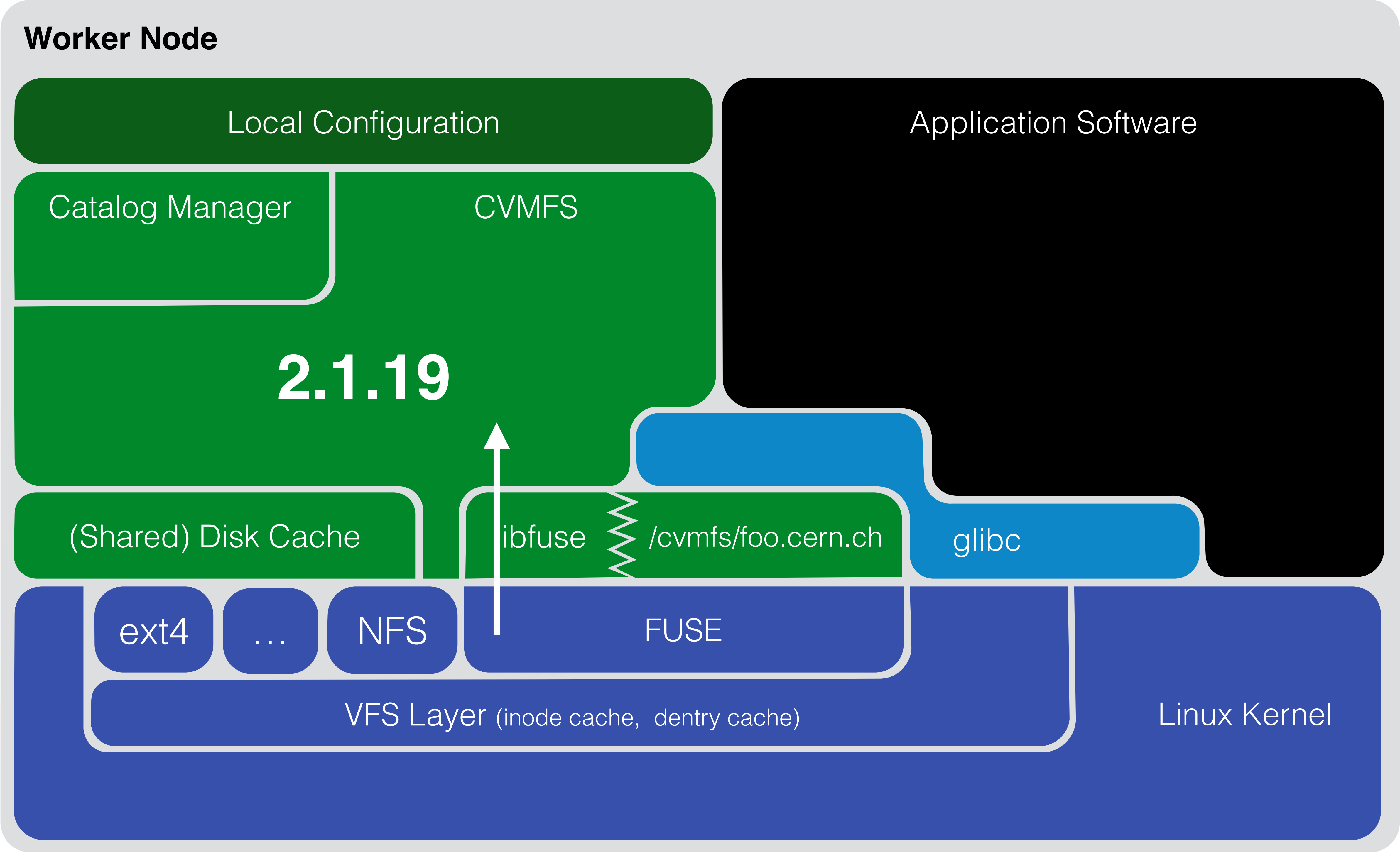
CernVM-FS Client - Control Flow



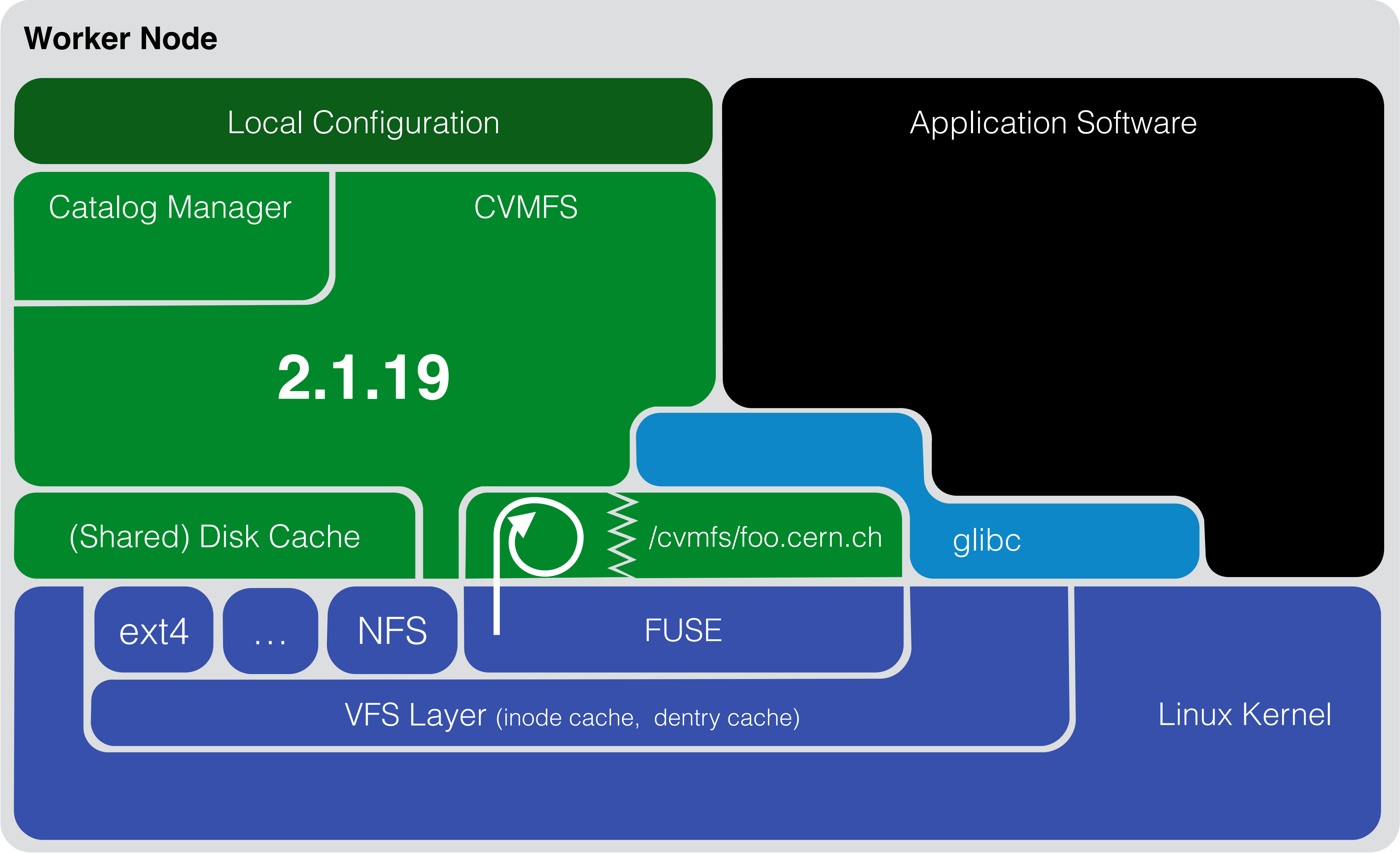
CernVM-FS Client - Control Flow



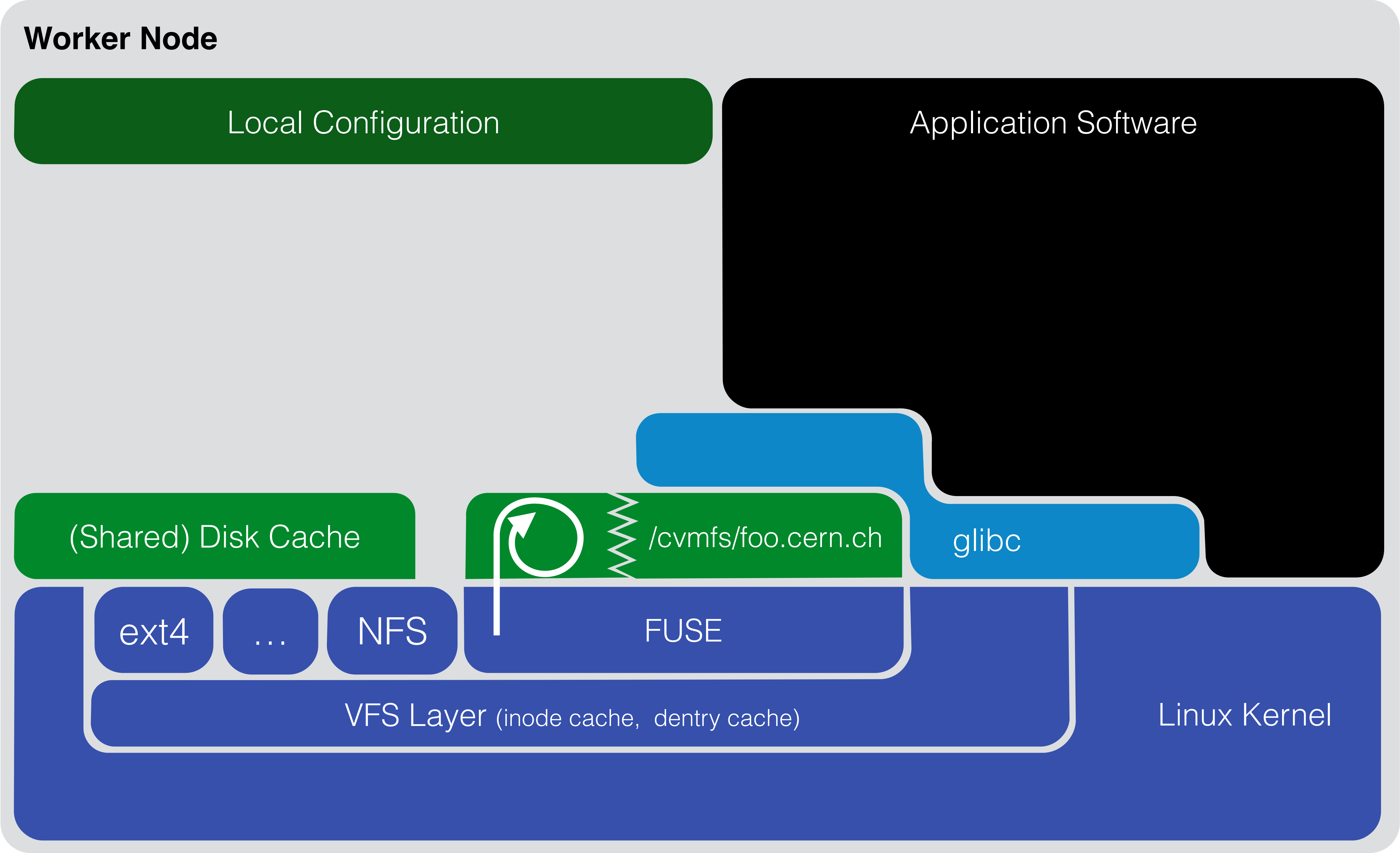
CernVM-FS Client - Hotpatching/Reloading



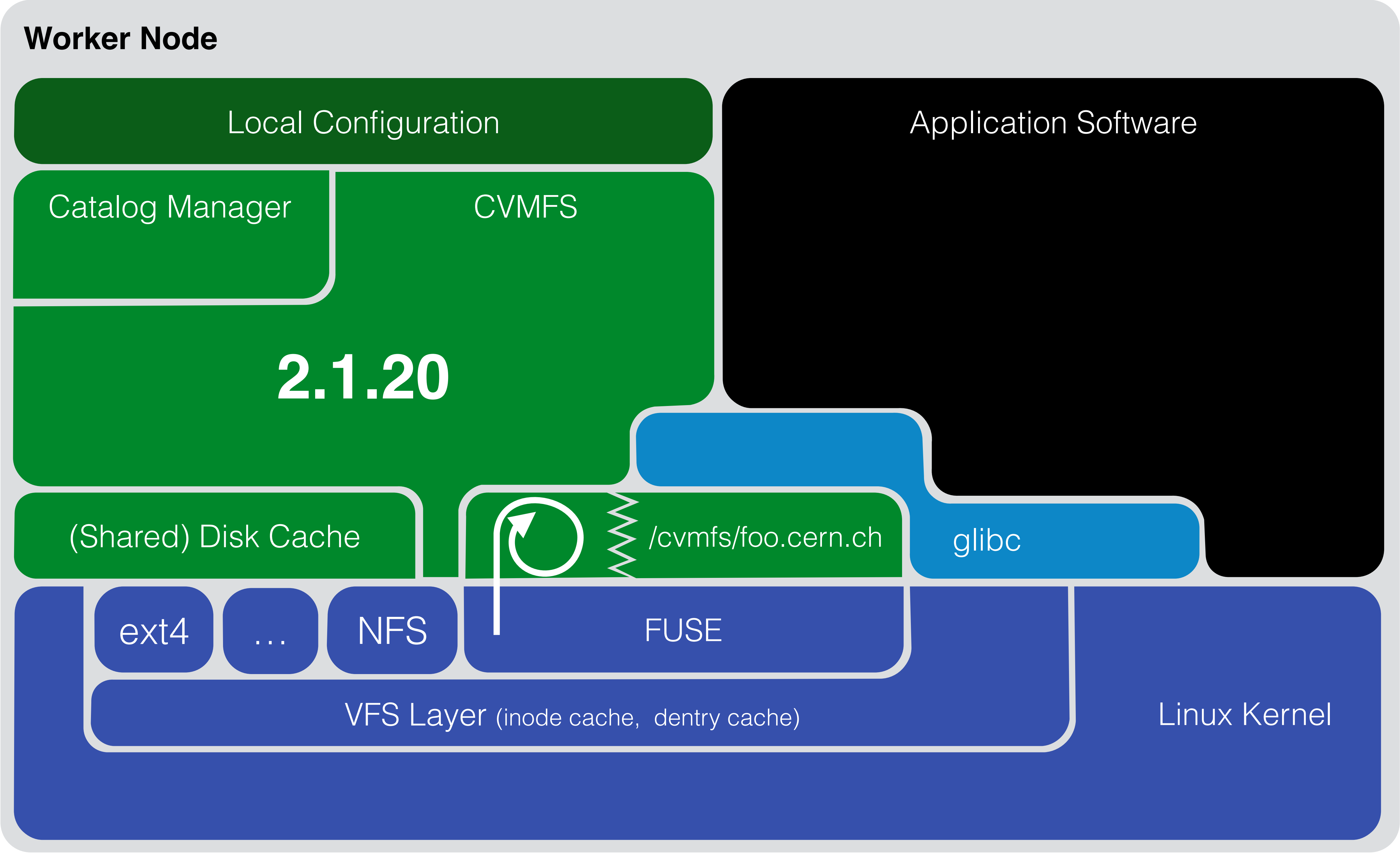
CernVM-FS Client - Hotpatching/Reloading



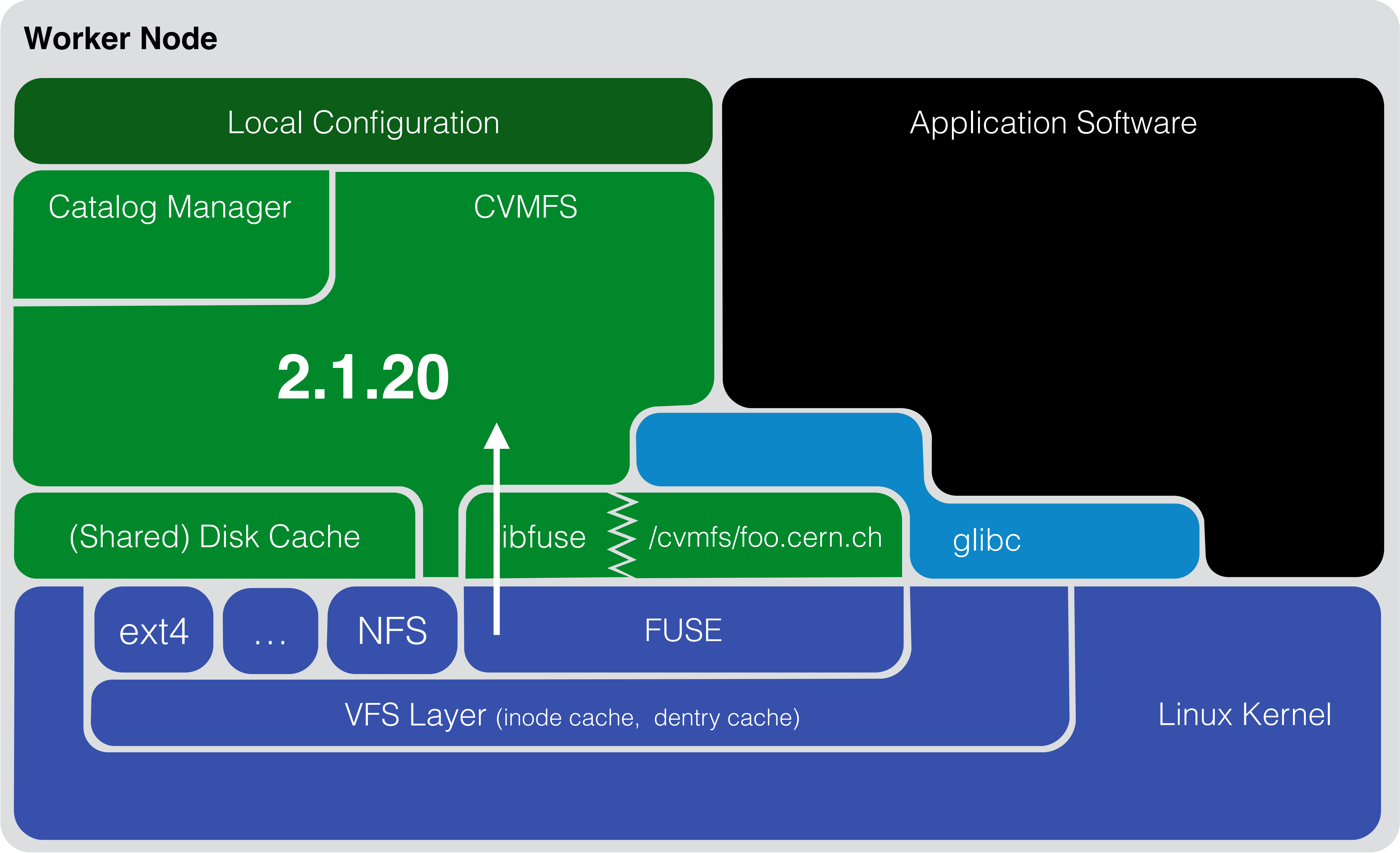
CernVM-FS Client - Hotpatching/Reloading



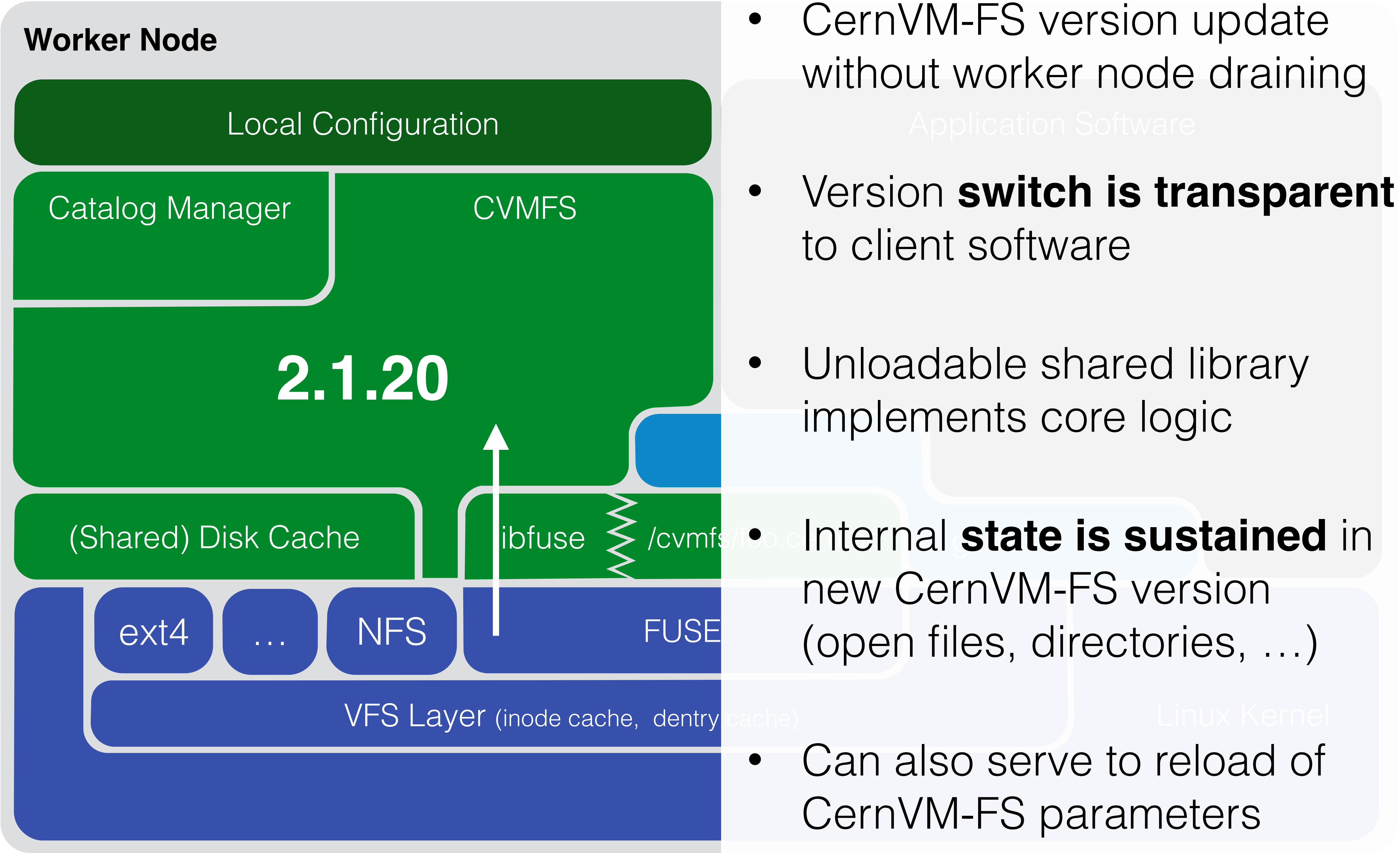
CernVM-FS Client - Hotpatching/Reloading



CernVM-FS Client - Hotpatching/Reloading



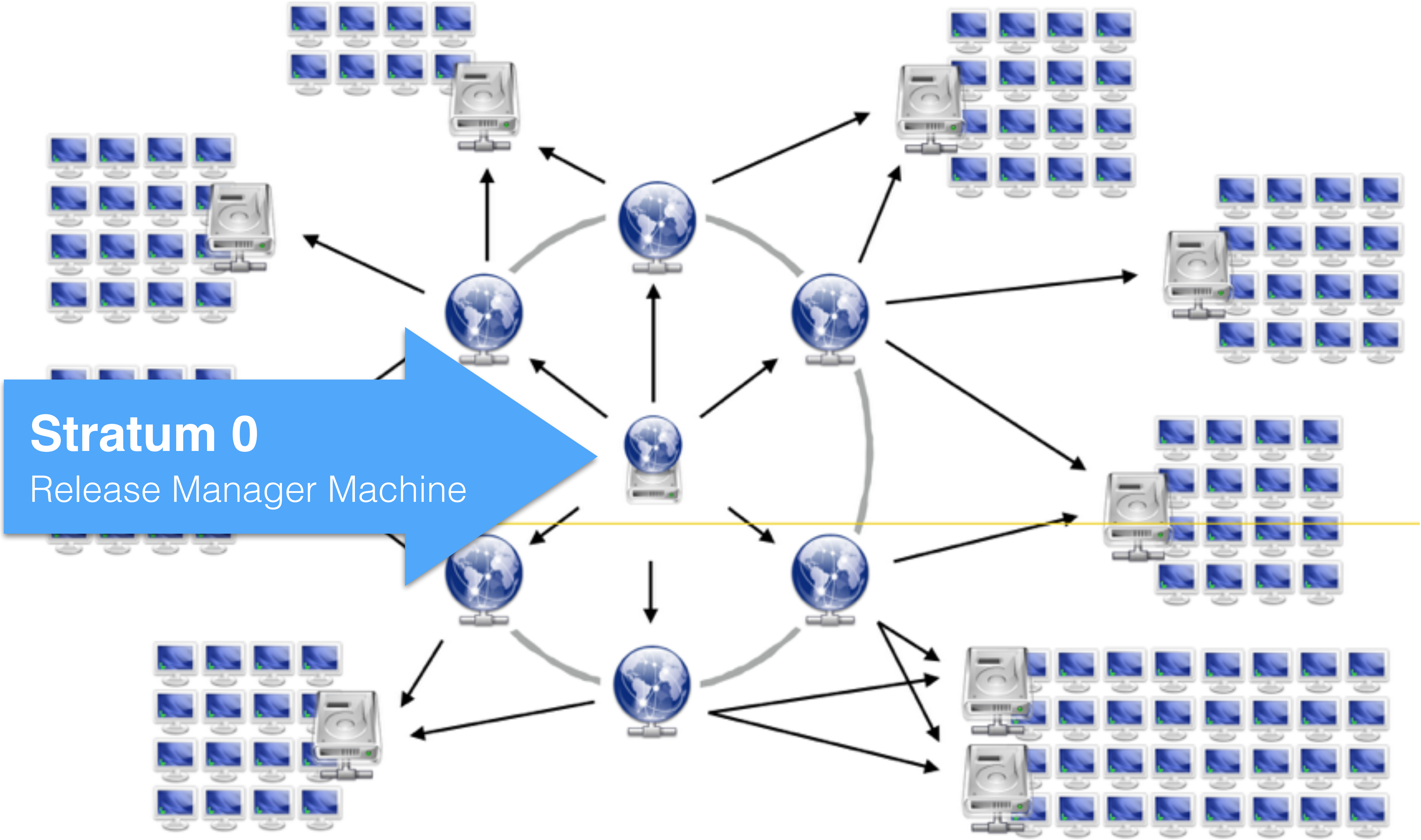
CernVM-FS Client - Hotpatching/Reloading



- CernVM-FS version update without worker node draining
- Version **switch is transparent** to client software
- Unloadable shared library implements core logic
- Internal **state is sustained** in new CernVM-FS version (open files, directories, …)
- Can also serve to reload of CernVM-FS parameters

CernVM-FS Server Setup and Usage

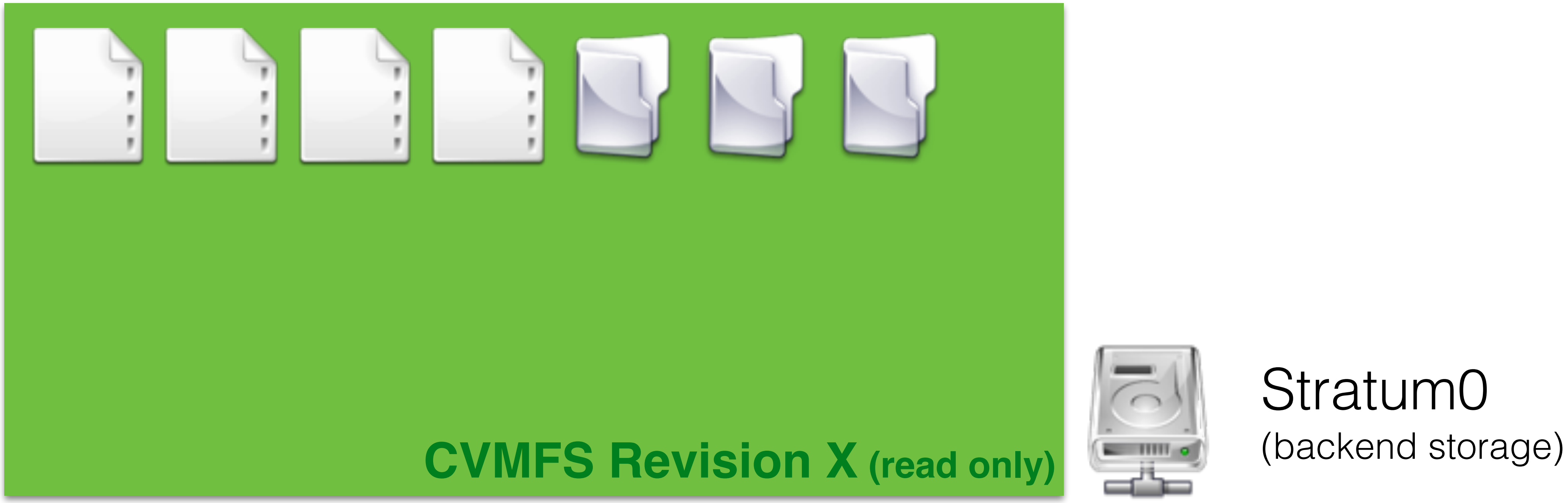
Release Manager Machine Internals



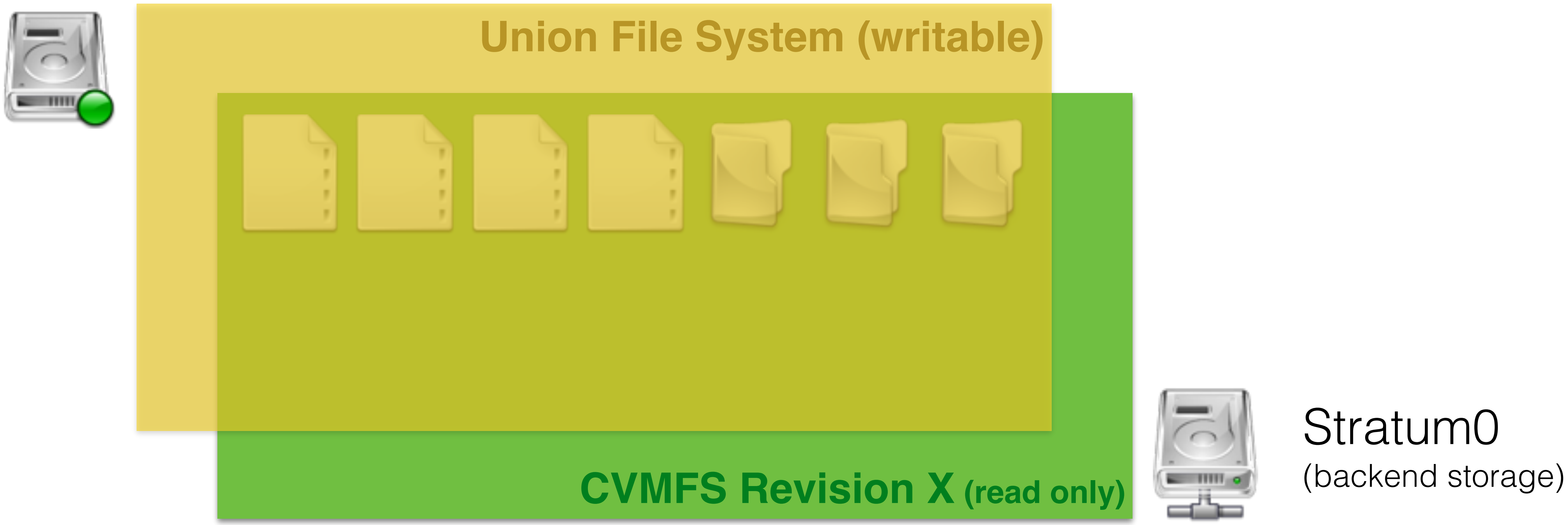
Updating a Repository

Stratum0
(backend storage)

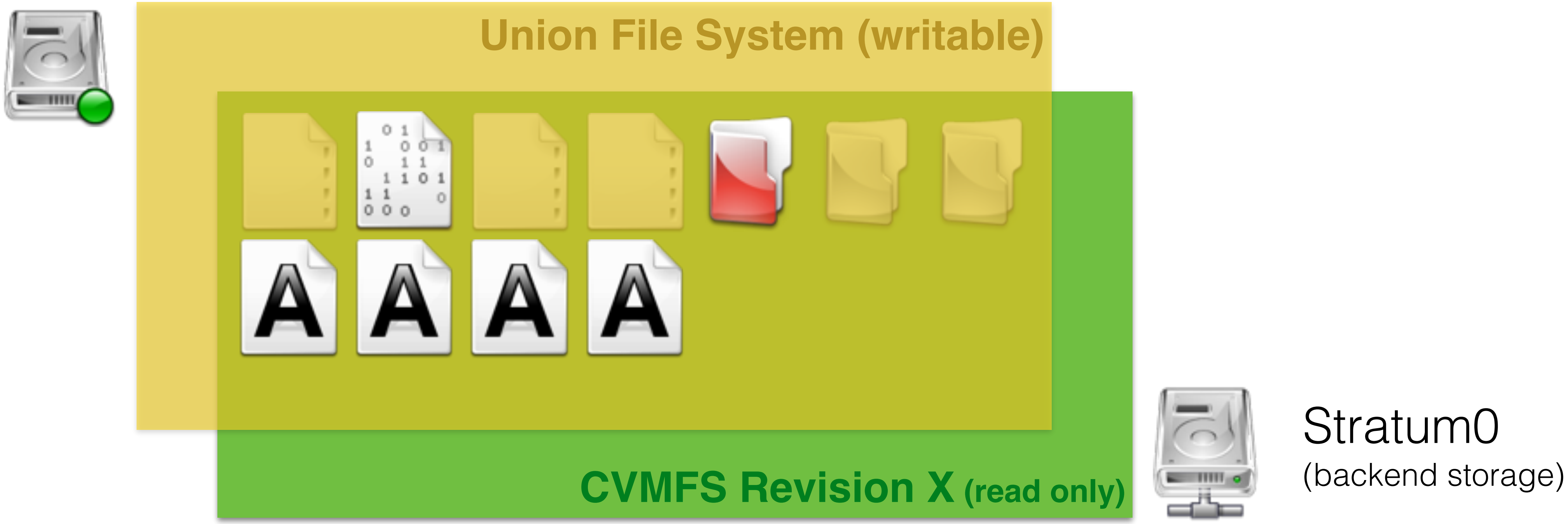
Updating a Repository



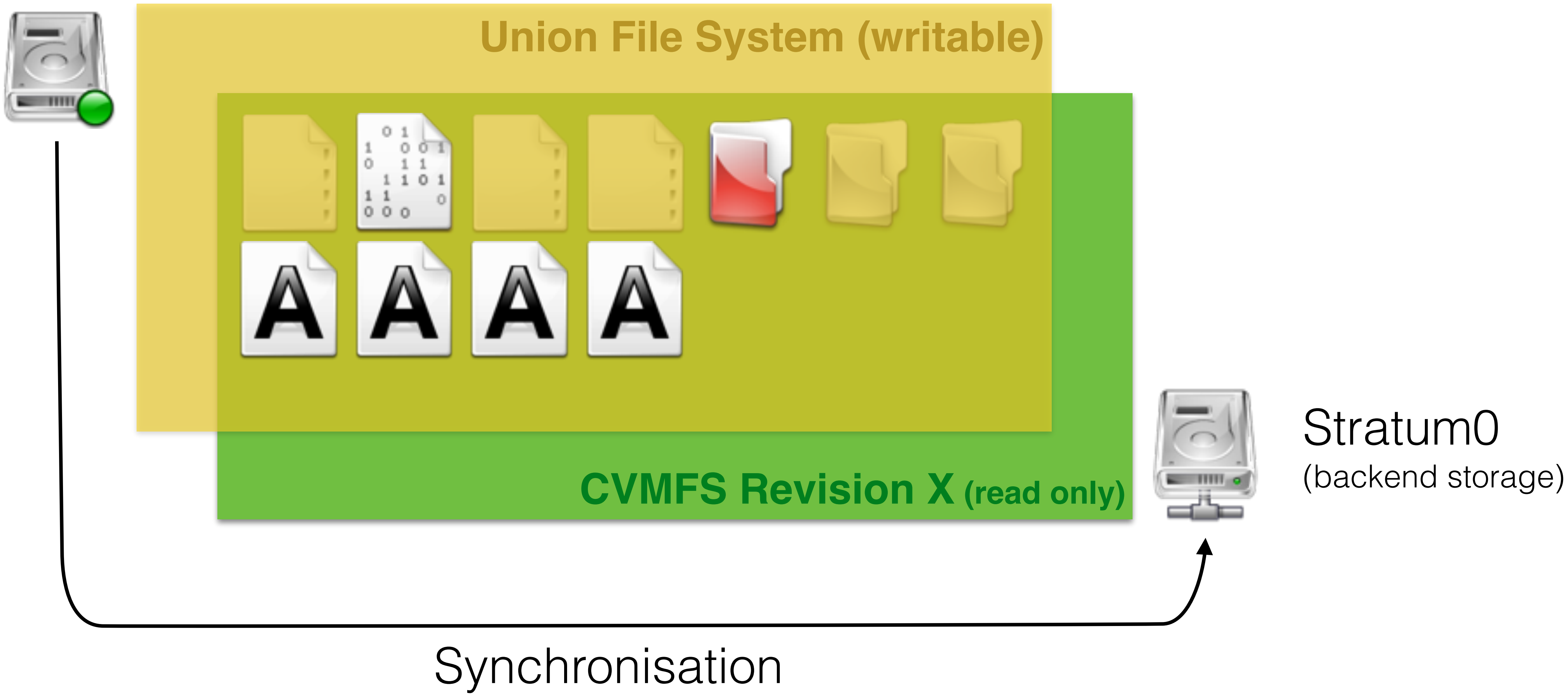
Updating a Repository



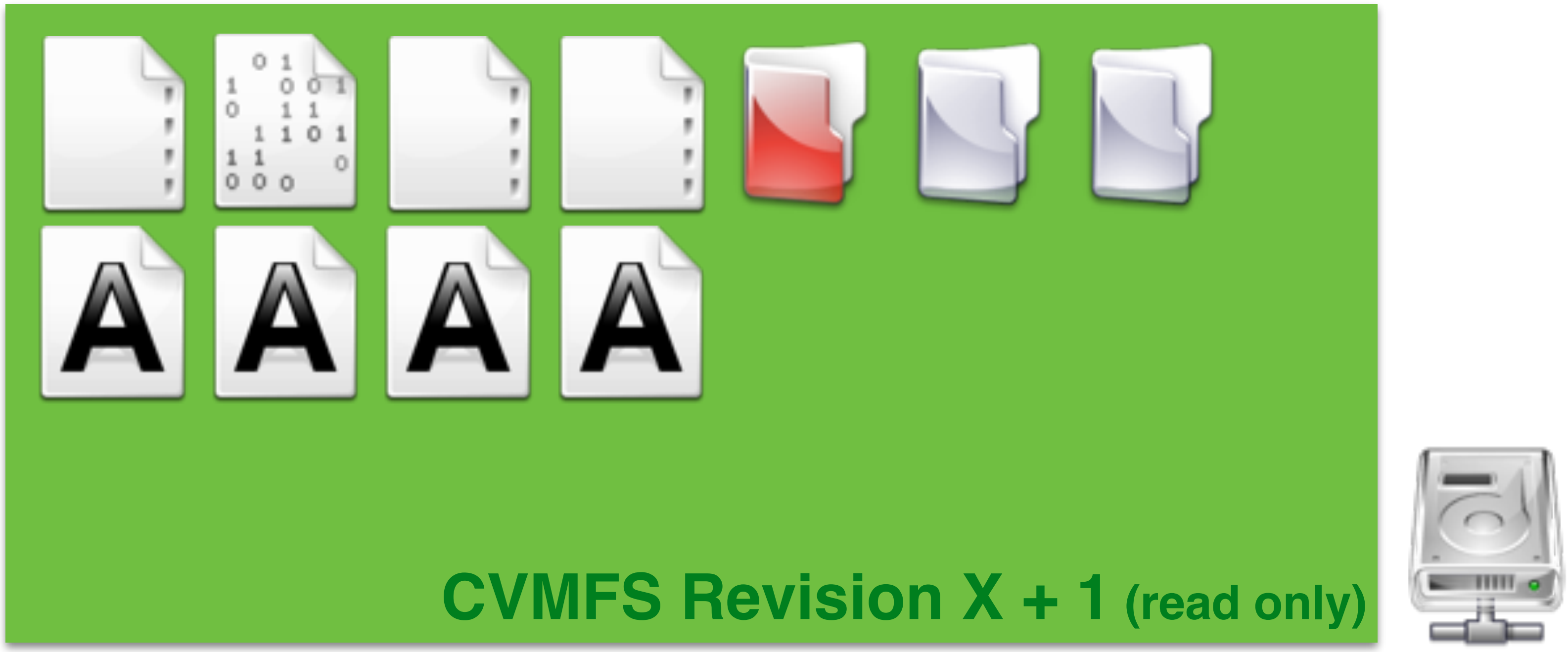
Updating a Repository



Updating a Repository



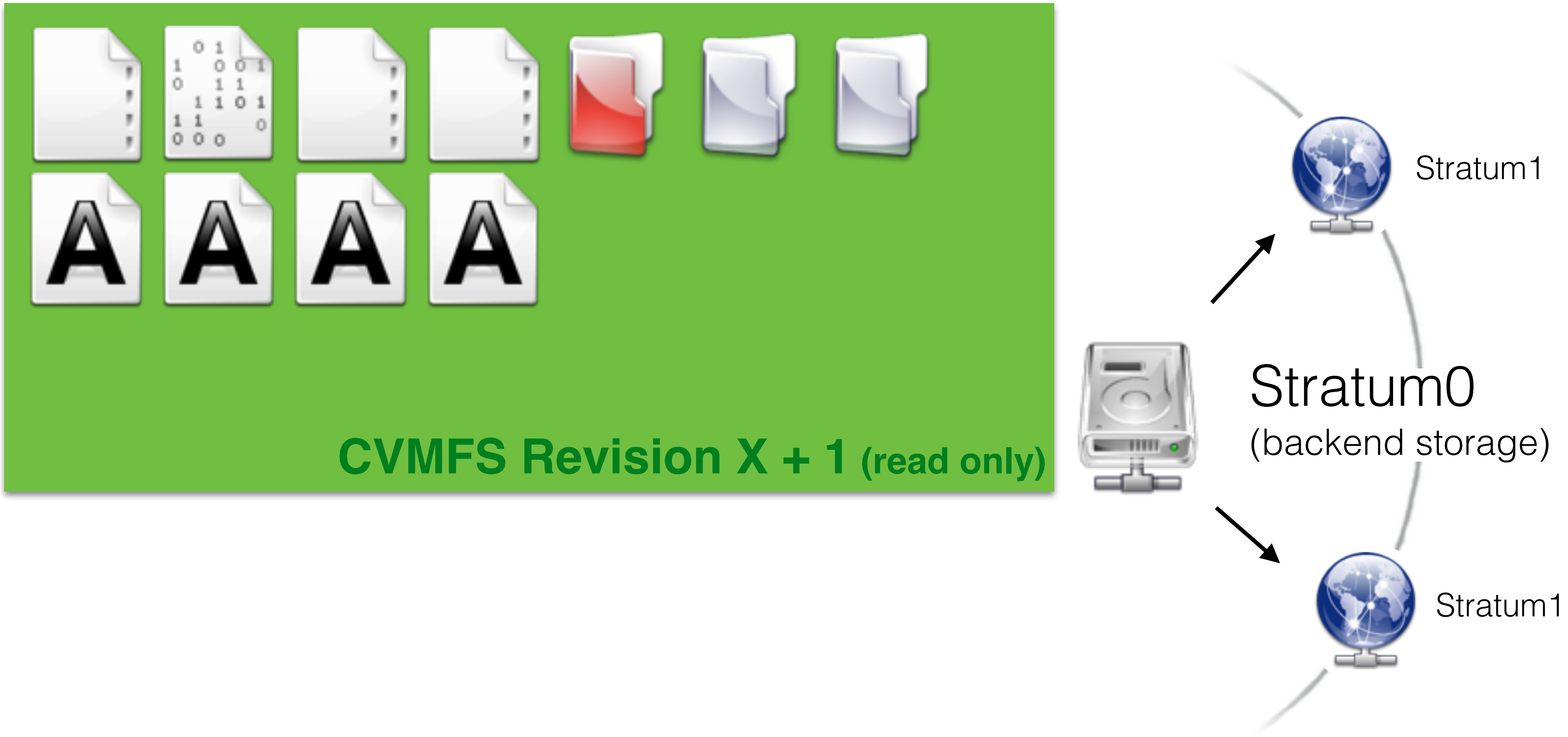
Updating a Repository



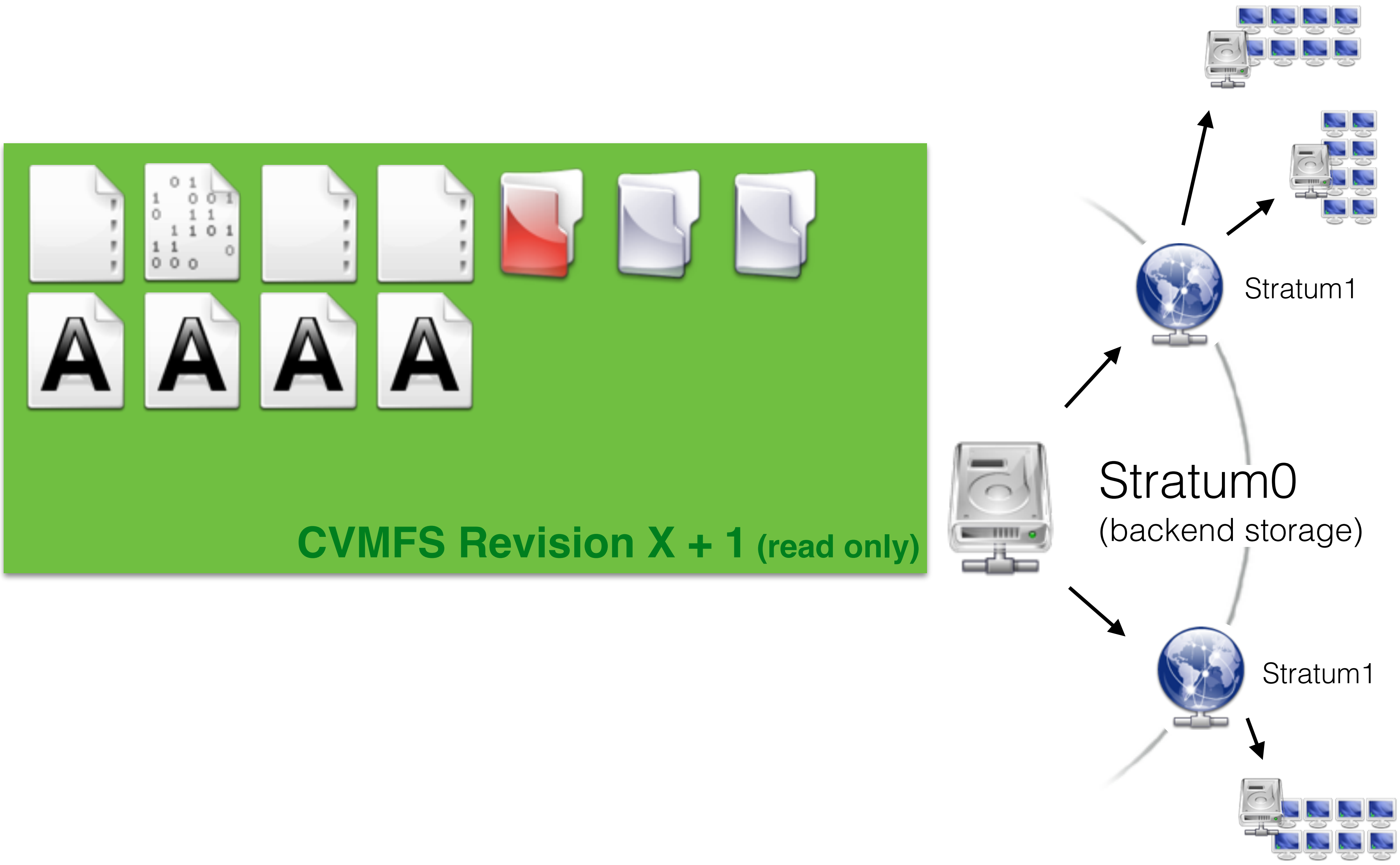
Stratum0

(backend storage)

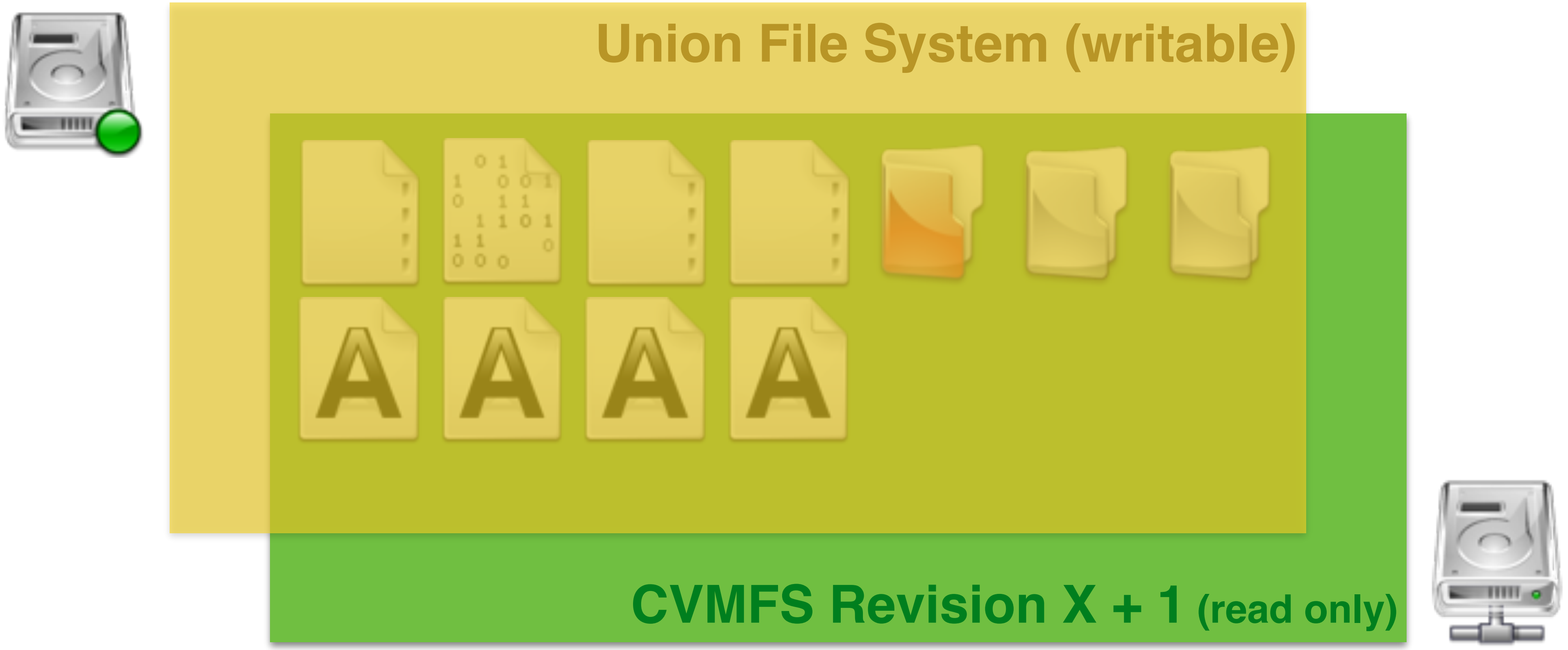
Updating a Repository



Updating a Repository

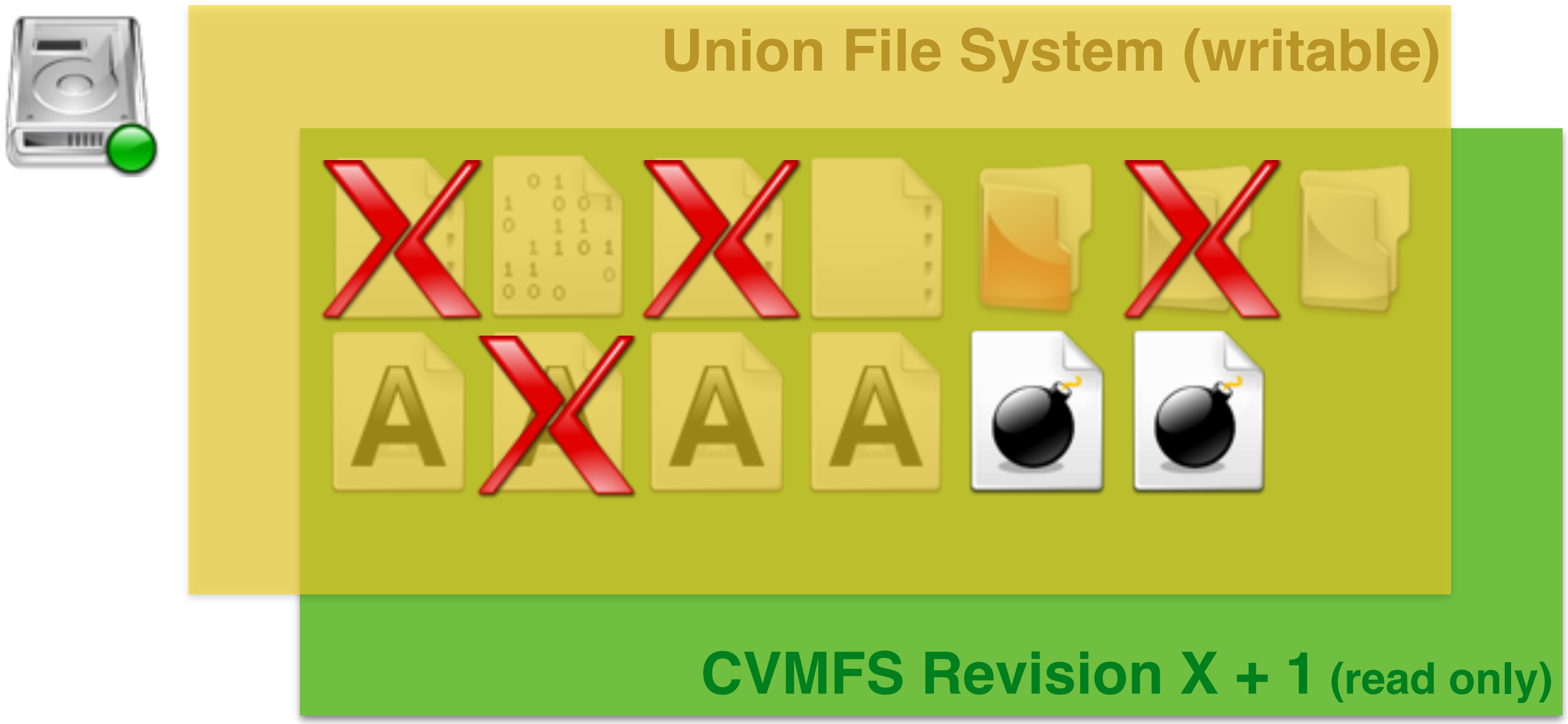


Updating a Repository


Union File System (writable)
CVMFS Revision X + 1 (read only)

Stratum0
(backend storage)

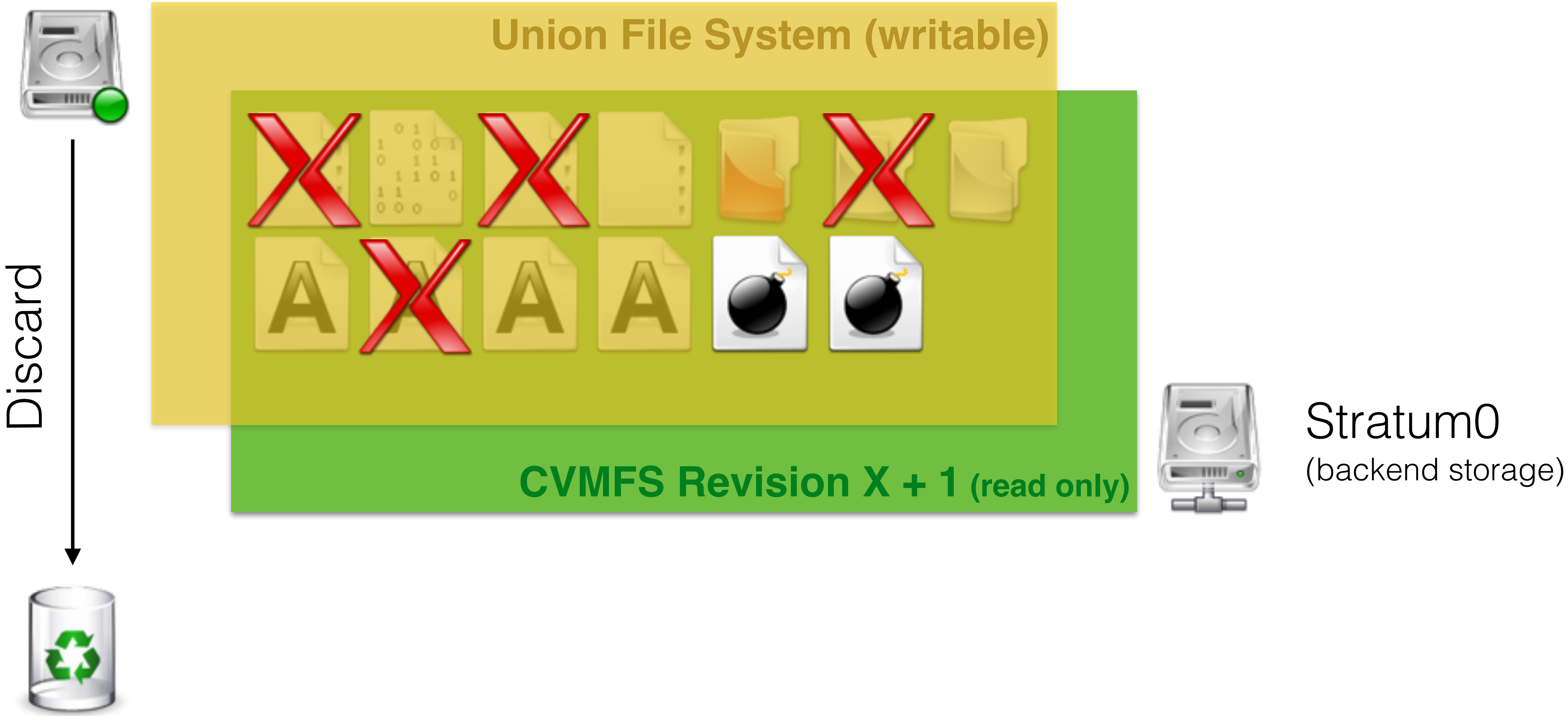
Updating a Repository


Union File System (writable)
A
A
A
A
CVMFS Revision X + 1 (read only)

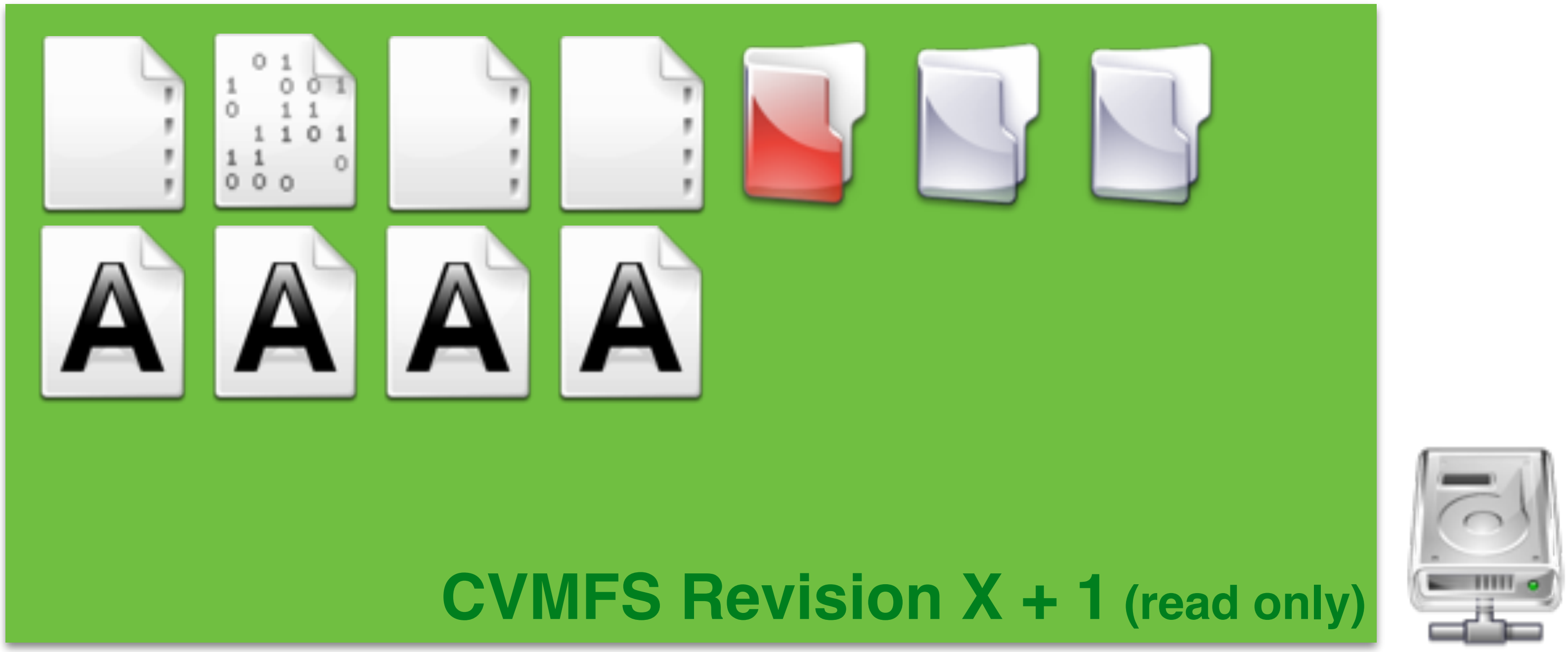
Stratum0

(backend storage)

Updating a Repository


Union File System (writable)
CVMFS Revision X + 1 (read only)
Discard
Stratum0
(backend storage)

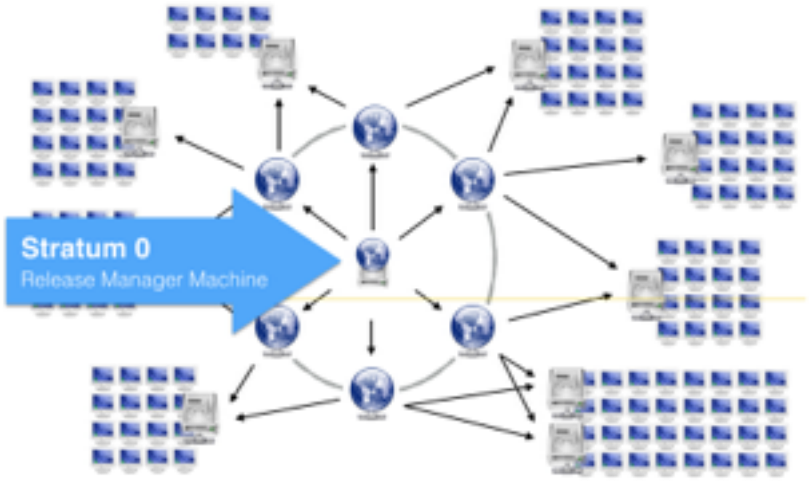
Updating a Repository



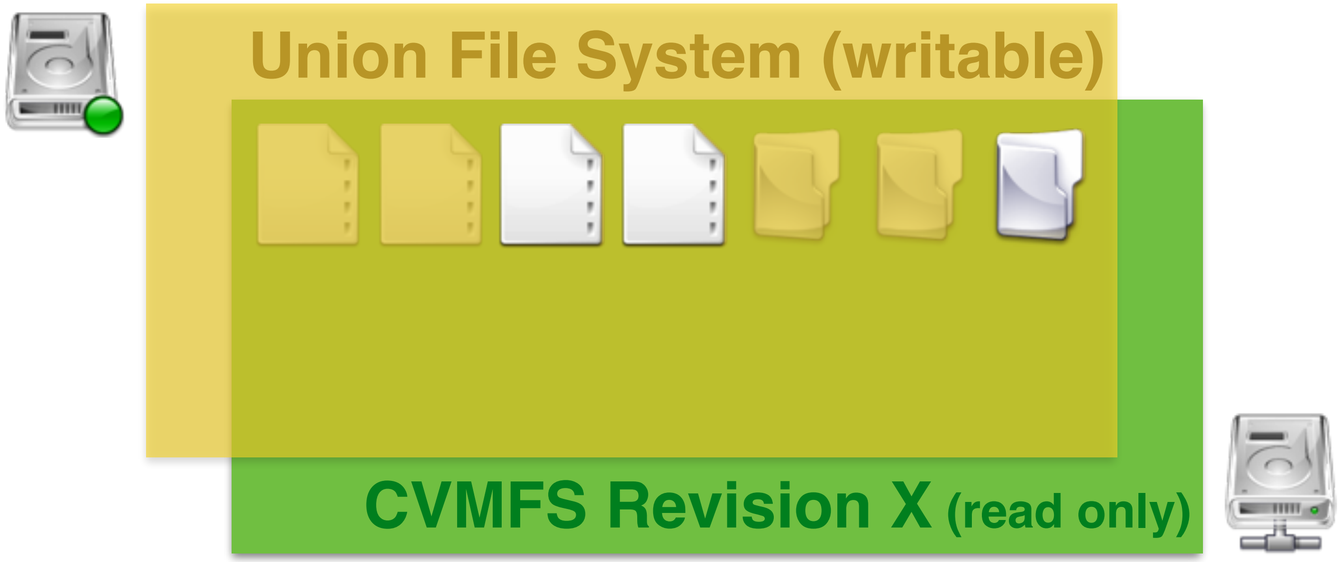
CVMFS Revision X + 1 (read only)

Stratum0
(backend storage)

CernVM-FS Server

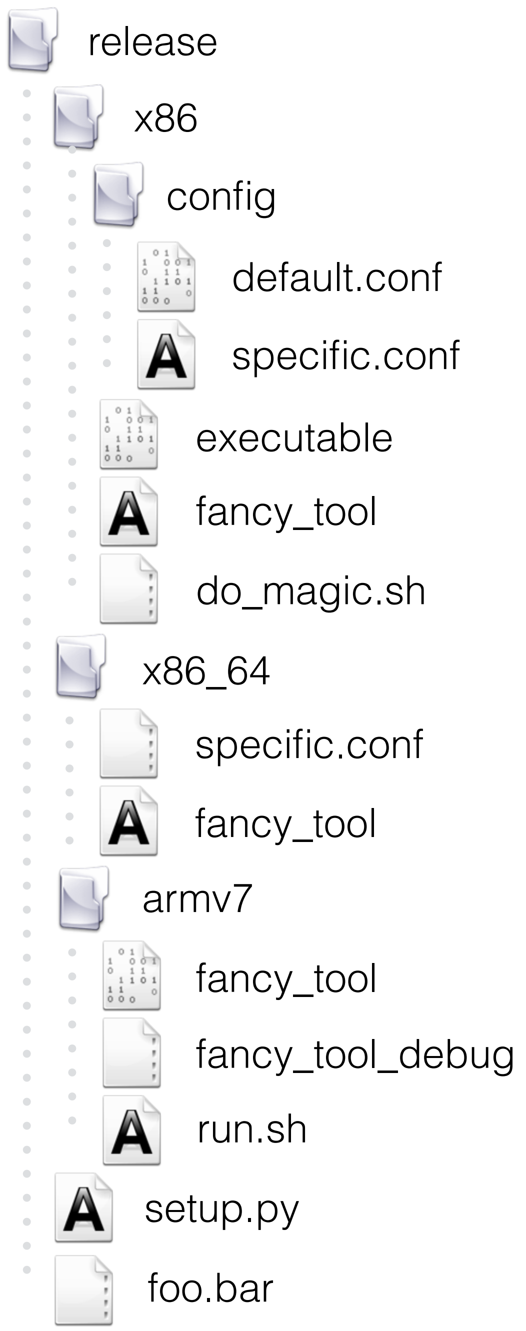


- Single **writable backend** of CernVM-FS
- Transactional publishing in **file system snapshots**
- POSIX-compliant read-write file system (copy-on-write semantics)
  - based on kernel-level union file system
  - aggregated change set in writable scratch area
- **Batch publishing** of snapshots
- **Historic snapshot** management
  - repository revisions stay available



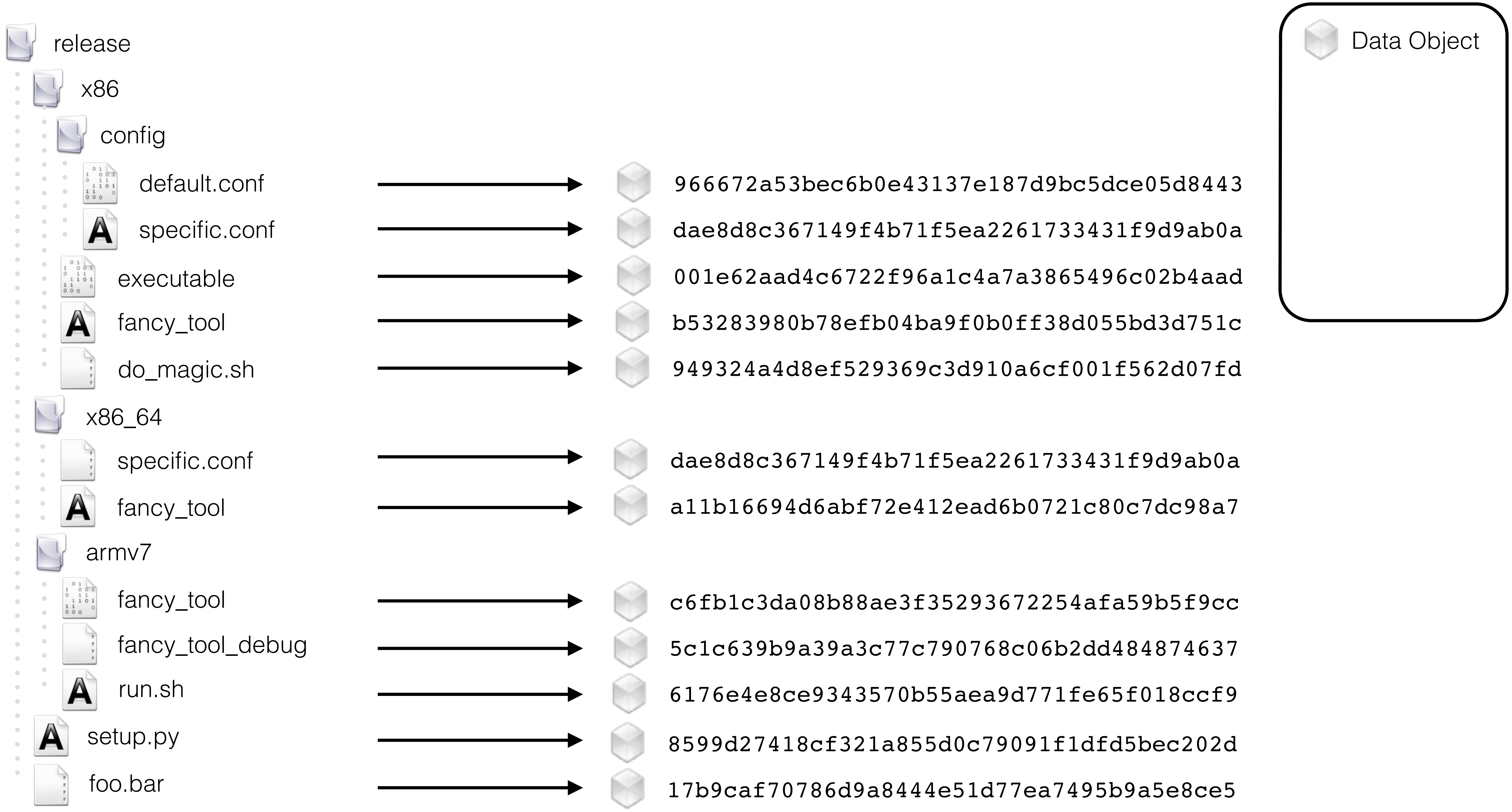
From POSIX
to CernVM-FS

From POSIX to Blob-Objects



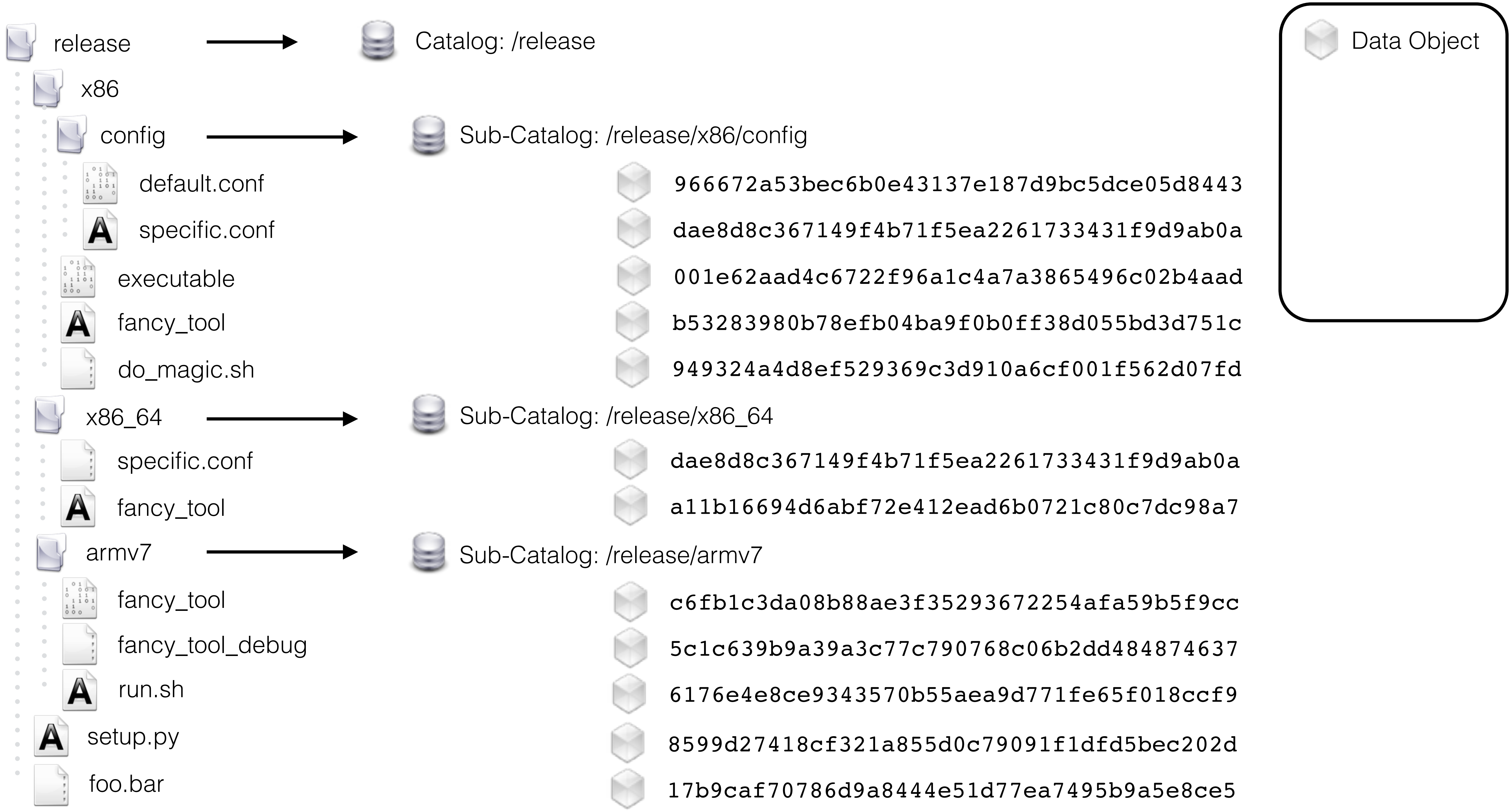
- release
  - x86
    - config
      - default.conf
      - specific.conf
    - executable
    - fancy_tool
    - do_magic.sh
  - x86_64
    - specific.conf
    - fancy_tool
  - armv7
    - fancy_tool
    - fancy_tool_debug
    - run.sh
  - setup.py
  - foo.bar

From POSIX to Blob-Objects



- release
  - x86
    - config
      - default.conf → 966672a53bec6b0e43137e187d9bc5dce05d8443
      - specific.conf → dae8d8c367149f4b71f5ea2261733431f9d9ab0a
    - executable → 001e62aad4c6722f96a1c4a7a3865496c02b4aad
    - fancy_tool → b53283980b78efb04ba9f0b0ff38d055bd3d751c
    - do_magic.sh → 949324a4d8ef529369c3d910a6cf001f562d07fd
  - x86_64
    - specific.conf → dae8d8c367149f4b71f5ea2261733431f9d9ab0a
    - fancy_tool → a11b16694d6abf72e412ead6b0721c80c7dc98a7
  - armv7
    - fancy_tool → c6fb1c3da08b88ae3f35293672254afa59b5f9cc
    - fancy_tool_debug → 5c1c639b9a39a3c77c790768c06b2dd484874637
    - run.sh → 6176e4e8ce9343570b55aea9d771fe65f018ccf9
  - setup.py → 8599d27418cf321a855d0c79091f1dfd5bec202d
  - foo.bar → 17b9caf70786d9a8444e51d77ea7495b9a5e8ce5

From POSIX to Blob-Objects



Data Object

release → Catalog: /release

x86

config → Sub-Catalog: /release/x86/config

default.conf 966672a53bec6b0e43137e187d9bc5dce05d8443

specific.conf dae8d8c367149f4b71f5ea2261733431f9d9ab0a

executable 001e62aad4c6722f96a1c4a7a3865496c02b4aad

fancy_tool b53283980b78efb04ba9f0b0ff38d055bd3d751c

do_magic.sh 949324a4d8ef529369c3d910a6cf001f562d07fd

x86_64 → Sub-Catalog: /release/x86_64

specific.conf dae8d8c367149f4b71f5ea2261733431f9d9ab0a

fancy_tool a11b16694d6abf72e412ead6b0721c80c7dc98a7

armv7 → Sub-Catalog: /release/armv7

fancy_tool c6fb1c3da08b88ae3f35293672254afa59b5f9cc

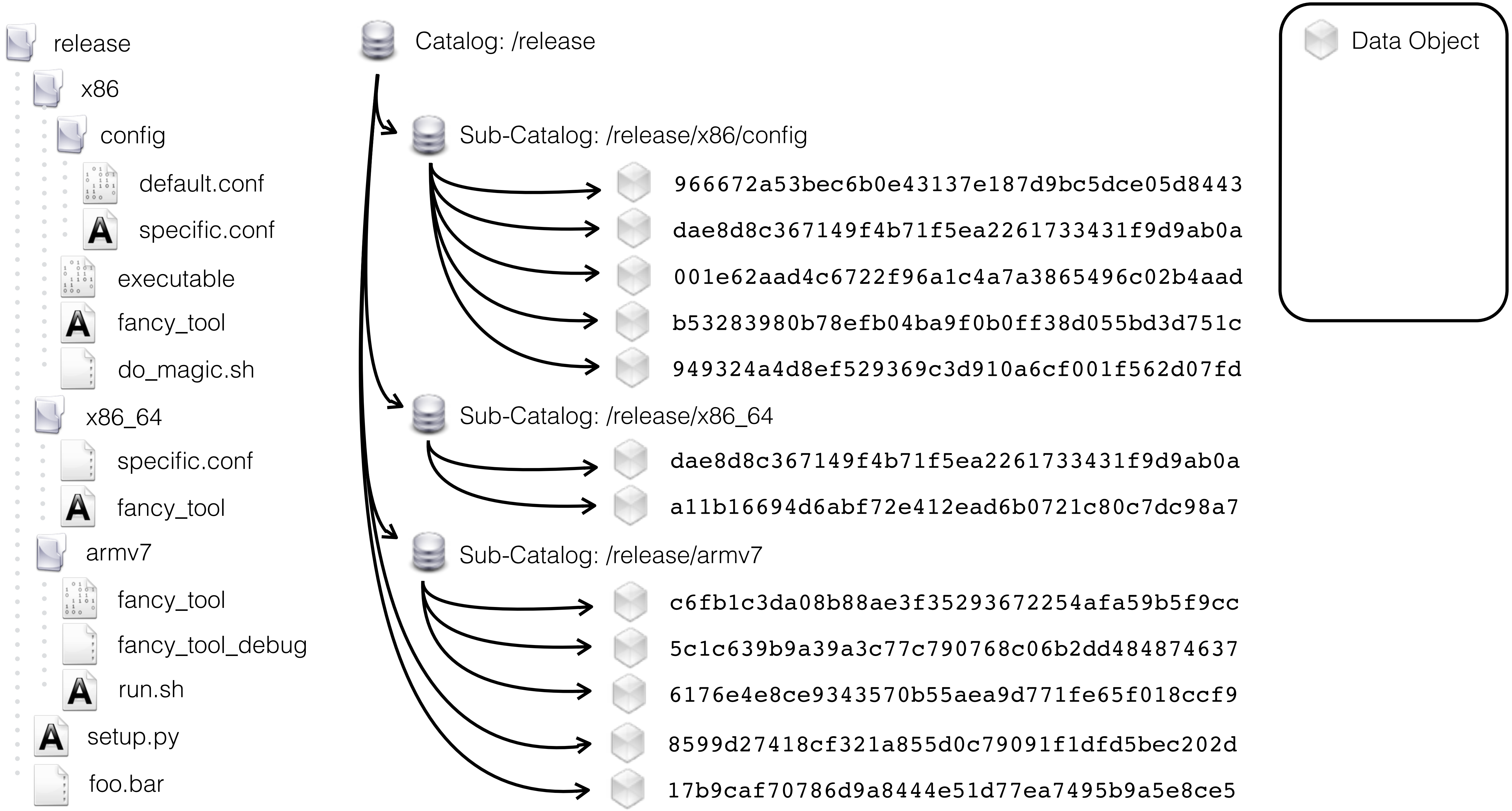
fancy_tool_debug 5c1c639b9a39a3c77c790768c06b2dd484874637

run.sh 6176e4e8ce9343570b55aea9d771fe65f018ccf9

setup.py 8599d27418cf321a855d0c79091f1dfd5bec202d

foo.bar 17b9caf70786d9a8444e51d77ea7495b9a5e8ce5

From POSIX to Blob-Objects



release
x86
config
default.conf
specific.conf
executable
fancy_tool
do_magic.sh
x86_64
specific.conf
fancy_tool
armv7
fancy_tool
fancy_tool_debug
run.sh
setup.py
foo.bar

Catalog: /release

Sub-Catalog: /release/x86/config

966672a53bec6b0e43137e187d9bc5dce05d8443
dae8d8c367149f4b71f5ea2261733431f9d9ab0a
001e62aad4c6722f96a1c4a7a3865496c02b4aad
b53283980b78efb04ba9f0b0ff38d055bd3d751c
949324a4d8ef529369c3d910a6cf001f562d07fd

Sub-Catalog: /release/x86_64

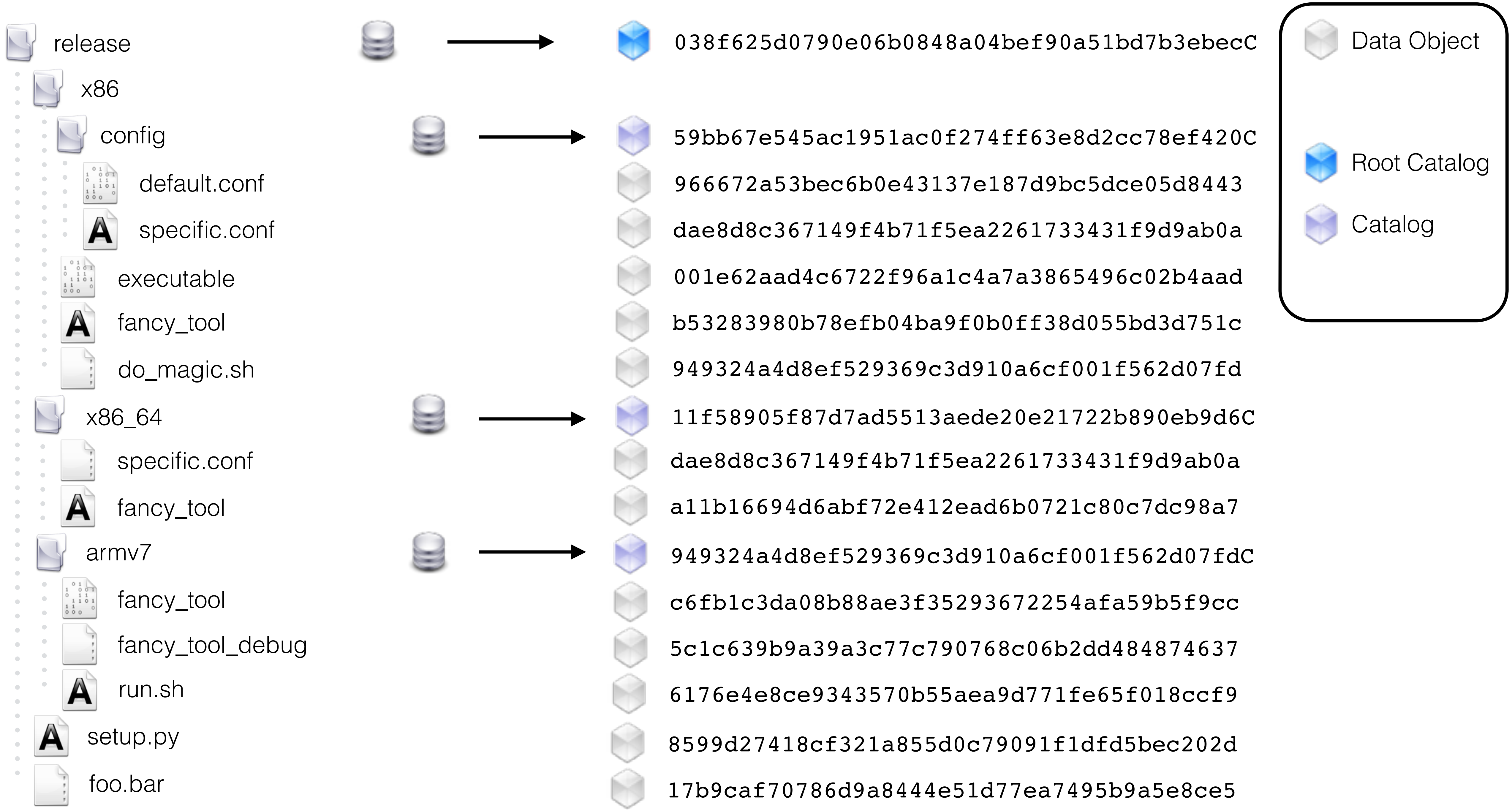
dae8d8c367149f4b71f5ea2261733431f9d9ab0a
a11b16694d6abf72e412ead6b0721c80c7dc98a7

Sub-Catalog: /release/armv7

c6fb1c3da08b88ae3f35293672254afa59b5f9cc
5c1c639b9a39a3c77c790768c06b2dd484874637
6176e4e8ce9343570b55aea9d771fe65f018ccf9
8599d27418cf321a855d0c79091f1dfd5bec202d
17b9caf70786d9a8444e51d77ea7495b9a5e8ce5

Data Object

From POSIX to Blob-Objects



release
x86
config
default.conf
specific.conf
executable
fancy_tool
do_magic.sh
x86_64
specific.conf
fancy_tool
armv7
fancy_tool
fancy_tool_debug
run.sh
setup.py
foo.bar

038f625d0790e06b0848a04bef90a51bd7b3ebecC
59bb67e545ac1951ac0f274ff63e8d2cc78ef420C
966672a53bec6b0e43137e187d9bc5dce05d8443
dae8d8c367149f4b71f5ea2261733431f9d9ab0a
001e62aad4c6722f96a1c4a7a3865496c02b4aad
b53283980b78efb04ba9f0b0ff38d055bd3d751c
949324a4d8ef529369c3d910a6cf001f562d07fd
11f58905f87d7ad5513aede20e21722b890eb9d6C
dae8d8c367149f4b71f5ea2261733431f9d9ab0a
a11b16694d6abf72e412ead6b0721c80c7dc98a7
949324a4d8ef529369c3d910a6cf001f562d07fdC
c6fb1c3da08b88ae3f35293672254afa59b5f9cc
5c1c639b9a39a3c77c790768c06b2dd484874637
6176e4e8ce9343570b55aea9d771fe65f018ccf9
8599d27418cf321a855d0c79091f1dfd5bec202d
17b9caf70786d9a8444e51d77ea7495b9a5e8ce5

Data Object
Root Catalog
Catalog

From POSIX to Blob-Objects

038f625d0790e06b0848a04bef90a51bd7b3ebecC

59bb67e545ac1951ac0f274ff63e8d2cc78ef420C

966672a53bec6b0e43137e187d9bc5dce05d8443

dae8d8c367149f4b71f5ea2261733431f9d9ab0a

001e62aad4c6722f96a1c4a7a3865496c02b4aad

b53283980b78efb04ba9f0b0ff38d055bd3d751c

949324a4d8ef529369c3d910a6cf001f562d07fd

11f58905f87d7ad5513aede20e21722b890eb9d6C

dae8d8c367149f4b71f5ea2261733431f9d9ab0a

a11b16694d6abf72e412ead6b0721c80c7dc98a7

949324a4d8ef529369c3d910a6cf001f562d07fdC

c6fb1c3da08b88ae3f35293672254afa59b5f9cc

5c1c639b9a39a3c77c790768c06b2dd484874637

6176e4e8ce9343570b55aea9d771fe65f018ccf9

8599d27418cf321a855d0c79091f1dfd5bec202d

17b9caf70786d9a8444e51d77ea7495b9a5e8ce5

Data Object

Root Catalog

Catalog

From POSIX to Blob-Objects

🟩 .cvmfspublished

🟦 038f625d0790e06b0848a04bef90a51bd7b3ebecC

🟪 59bb67e545ac1951ac0f274ff63e8d2cc78ef420C

⬜ 966672a53bec6b0e43137e187d9bc5dce05d8443

⬜ dae8d8c367149f4b71f5ea2261733431f9d9ab0a

⬜ 001e62aad4c6722f96a1c4a7a3865496c02b4aad

⬜ b53283980b78efb04ba9f0b0ff38d055bd3d751c

⬜ 949324a4d8ef529369c3d910a6cf001f562d07fd

🟪 11f58905f87d7ad5513aede20e21722b890eb9d6C

⬜ dae8d8c367149f4b71f5ea2261733431f9d9ab0a

⬜ a11b16694d6abf72e412ead6b0721c80c7dc98a7

🟪 949324a4d8ef529369c3d910a6cf001f562d07fdC

⬜ c6fb1c3da08b88ae3f35293672254afa59b5f9cc

⬜ 5c1c639b9a39a3c77c790768c06b2dd484874637

⬜ 6176e4e8ce9343570b55aea9d771fe65f018ccf9

⬜ 8599d27418cf321a855d0c79091f1dfd5bec202d

⬜ 17b9caf70786d9a8444e51d77ea7495b9a5e8ce5

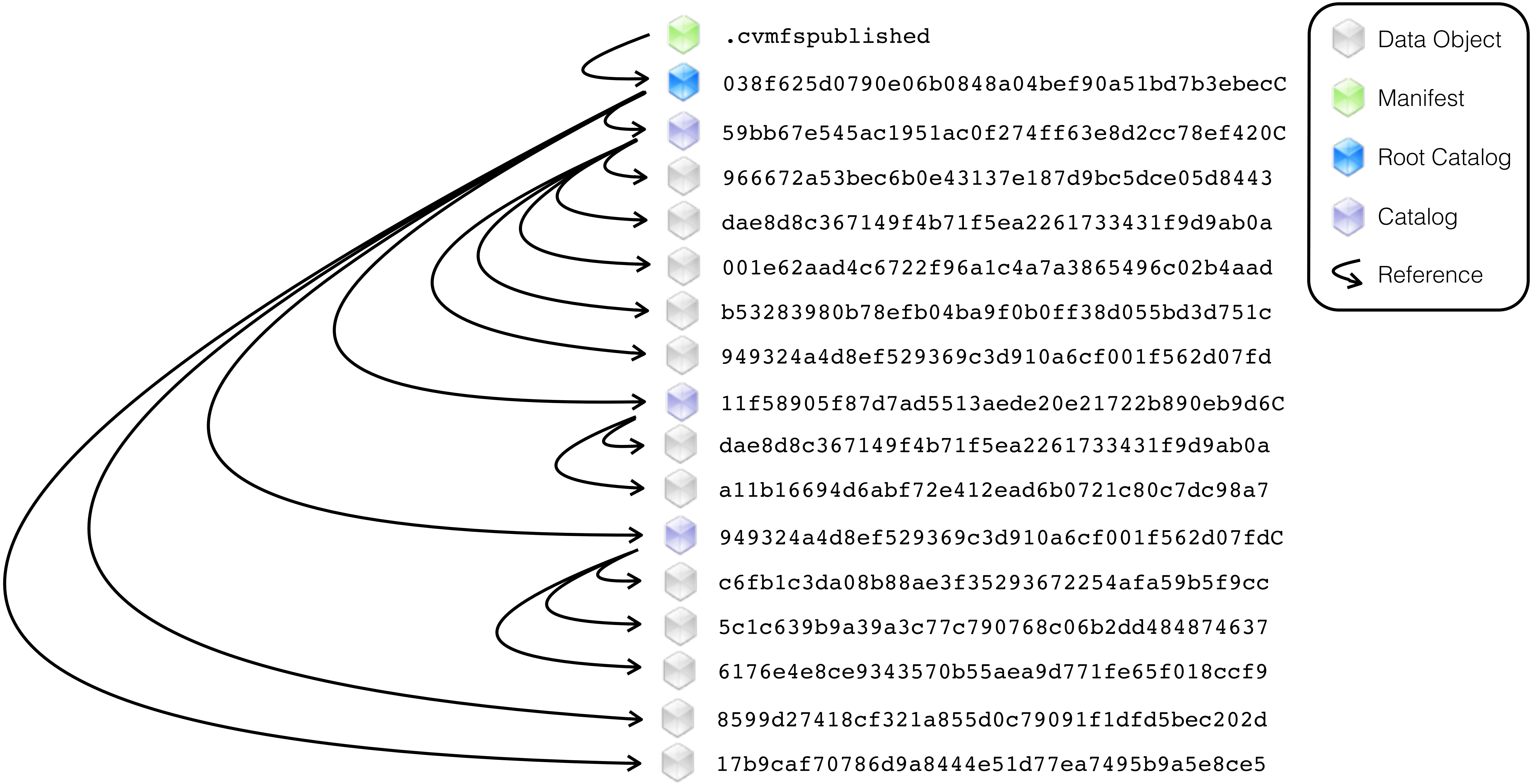
⬜ Data Object

🟩 Manifest

🟦 Root Catalog

🟪 Catalog

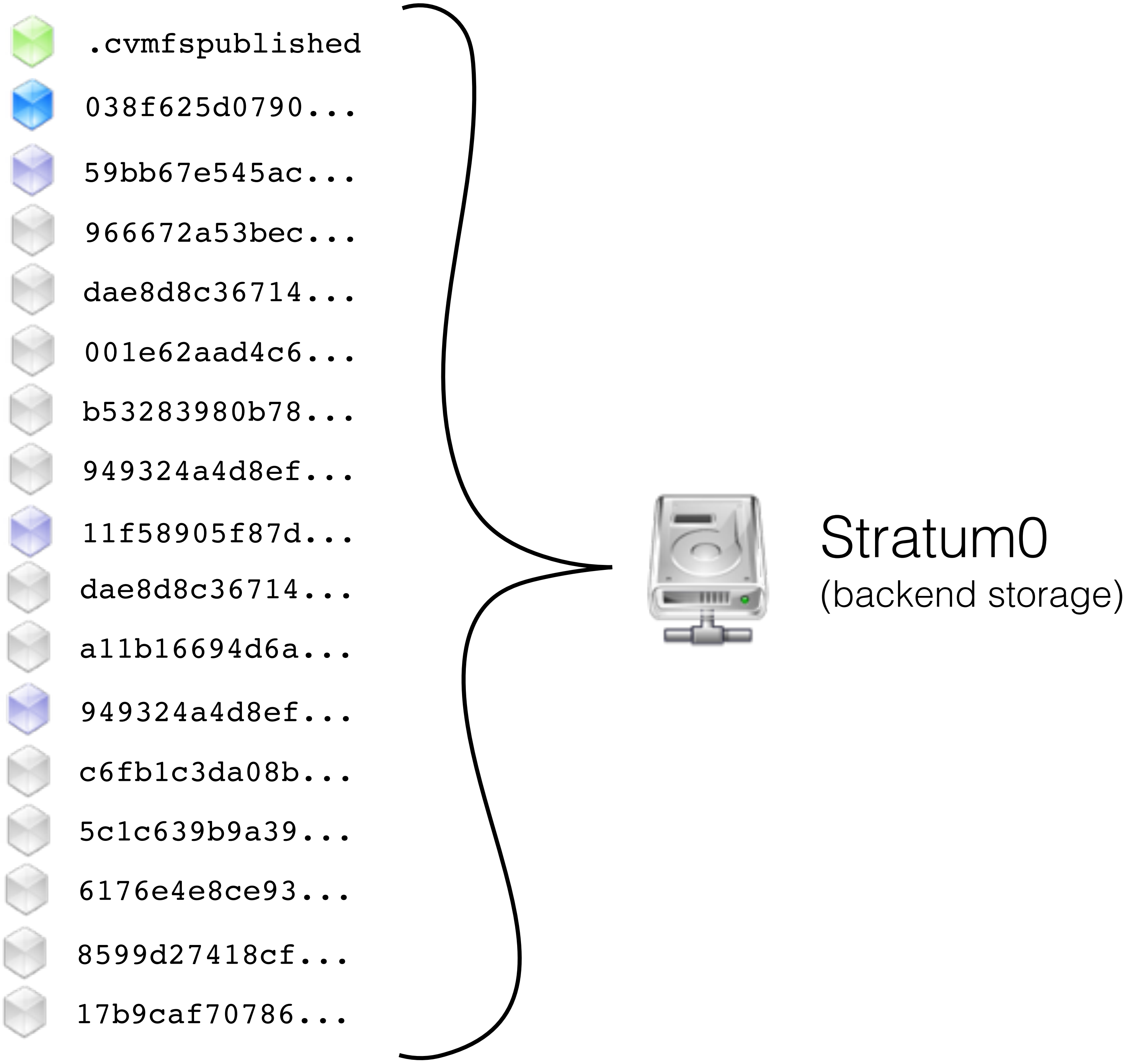
From POSIX to Blob-Objects



```
.cvmfspublished
038f625d0790e06b0848a04bef90a51bd7b3ebecC
59bb67e545ac1951ac0f274ff63e8d2cc78ef420C
966672a53bec6b0e43137e187d9bc5dce05d8443
dae8d8c367149f4b71f5ea2261733431f9d9ab0a
001e62aad4c6722f96a1c4a7a3865496c02b4aad
b53283980b78efb04ba9f0b0ff38d055bd3d751c
949324a4d8ef529369c3d910a6cf001f562d07fd
11f58905f87d7ad5513aede20e21722b890eb9d6C
dae8d8c367149f4b71f5ea2261733431f9d9ab0a
a11b16694d6abf72e412ead6b0721c80c7dc98a7
949324a4d8ef529369c3d910a6cf001f562d07fdC
c6fb1c3da08b88ae3f35293672254afa59b5f9cc
5c1c639b9a39a3c77c790768c06b2dd484874637
6176e4e8ce9343570b55aea9d771fe65f018ccf9
8599d27418cf321a855d0c79091f1dfd5bec202d
17b9caf70786d9a8444e51d77ea7495b9a5e8ce5
```

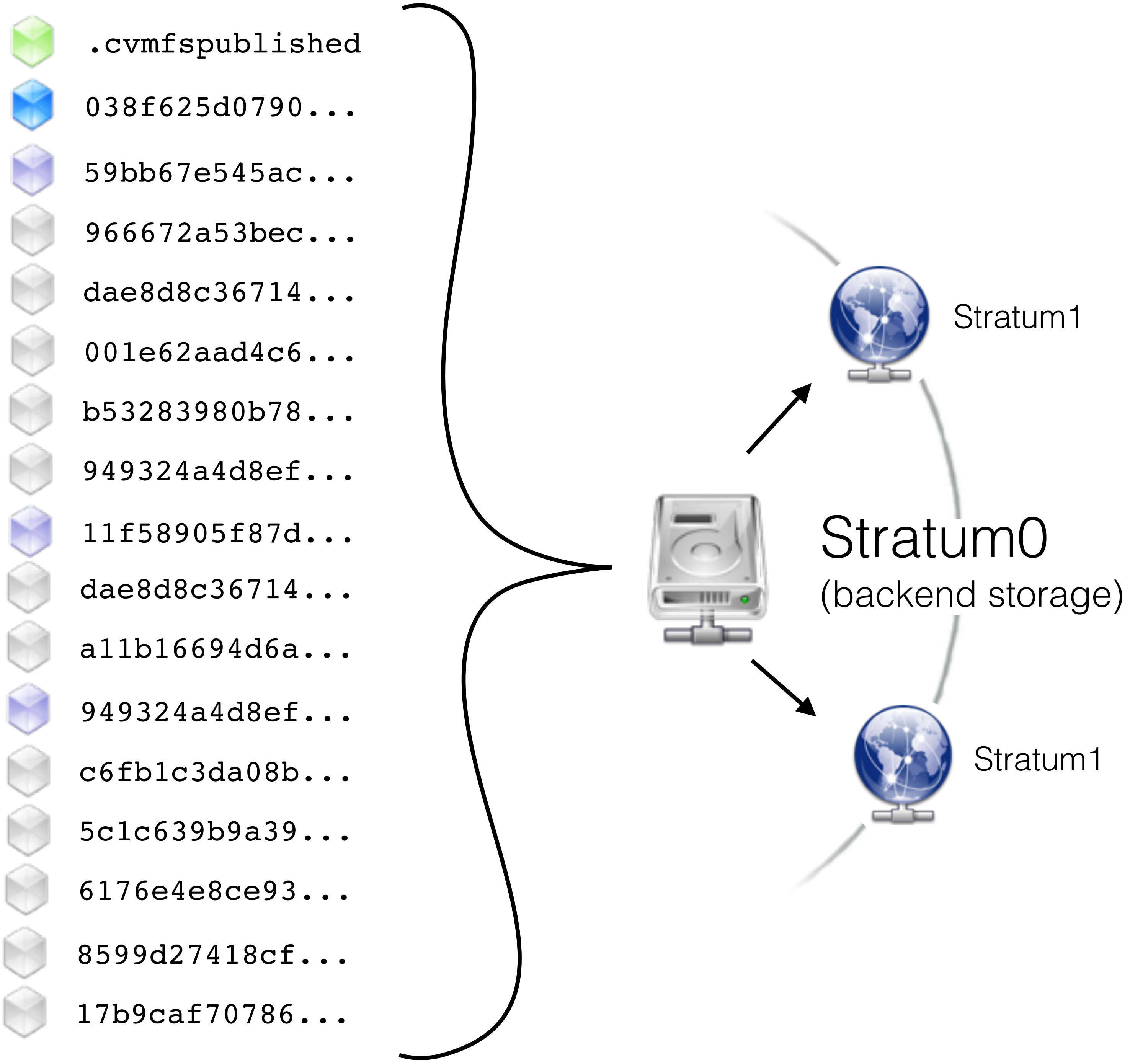
From POSIX to Blob-Objects

- **Hierarchy of File Catalogs**
  - File system meta-data, directories, symlinks, …
  - Content hashes of regular files
  - Root catalog is cryptographically signed
- **Content-Addressable Storage**
  - File de-duplication
  - Trivial file integrity checks
  - Insert-only semantic
- **Flat Namespace**
  - Perfect for HTTP caching



From POSIX to Blob-Objects

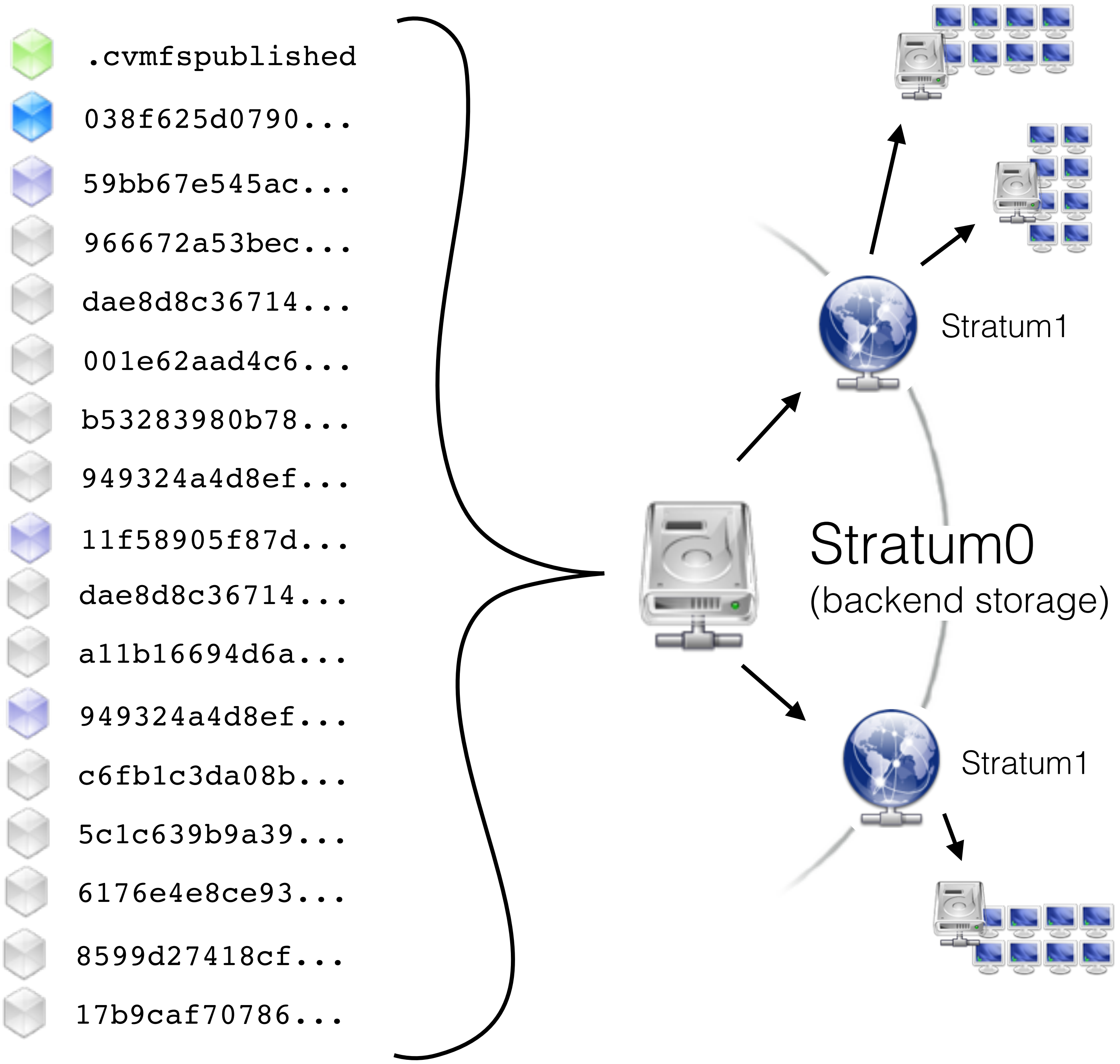
- **Hierarchy of File Catalogs**
  - File system meta-data, directories, symlinks, …
  - Content hashes of regular files
  - Root catalog is cryptographically signed
- **Content-Addressable Storage**
  - File de-duplication
  - Trivial file integrity checks
  - Insert-only semantic
- **Flat Namespace**
  - Perfect for HTTP caching



.cvmfspublished
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
dae8d8c36714...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...
17b9caf70786...

From POSIX to Blob-Objects

- **Hierarchy of File Catalogs**
  - File system meta-data, directories, symlinks, …
  - Content hashes of regular files
  - Root catalog is cryptographically signed
- **Content-Addressable Storage**
  - File de-duplication
  - Trivial file integrity checks
  - Insert-only semantic
- **Flat Namespace**
  - Perfect for HTTP caching



.cvmfspublished
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
dae8d8c36714...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...
17b9caf70786...

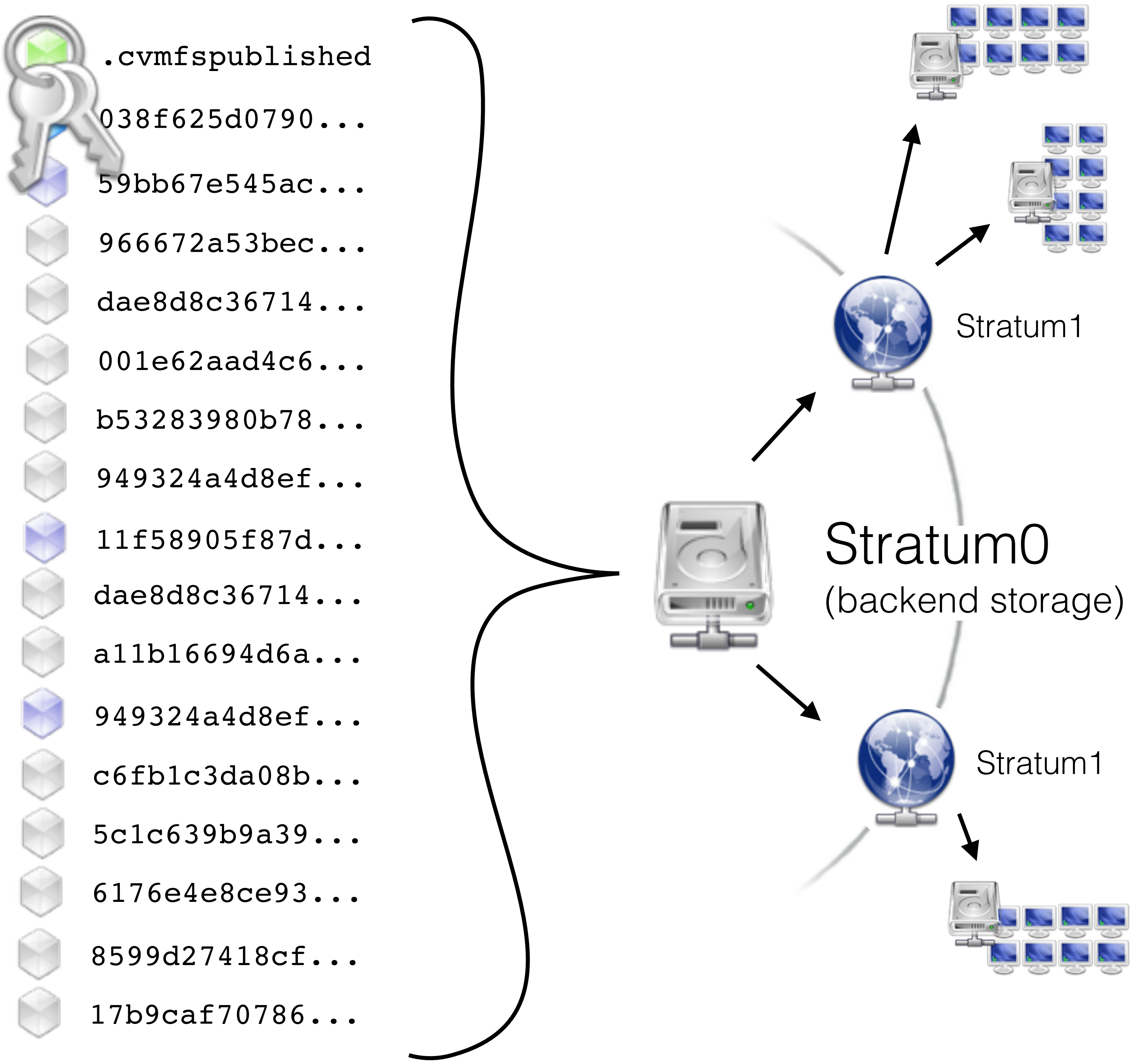
Stratum1

Stratum0
(backend storage)

Stratum1

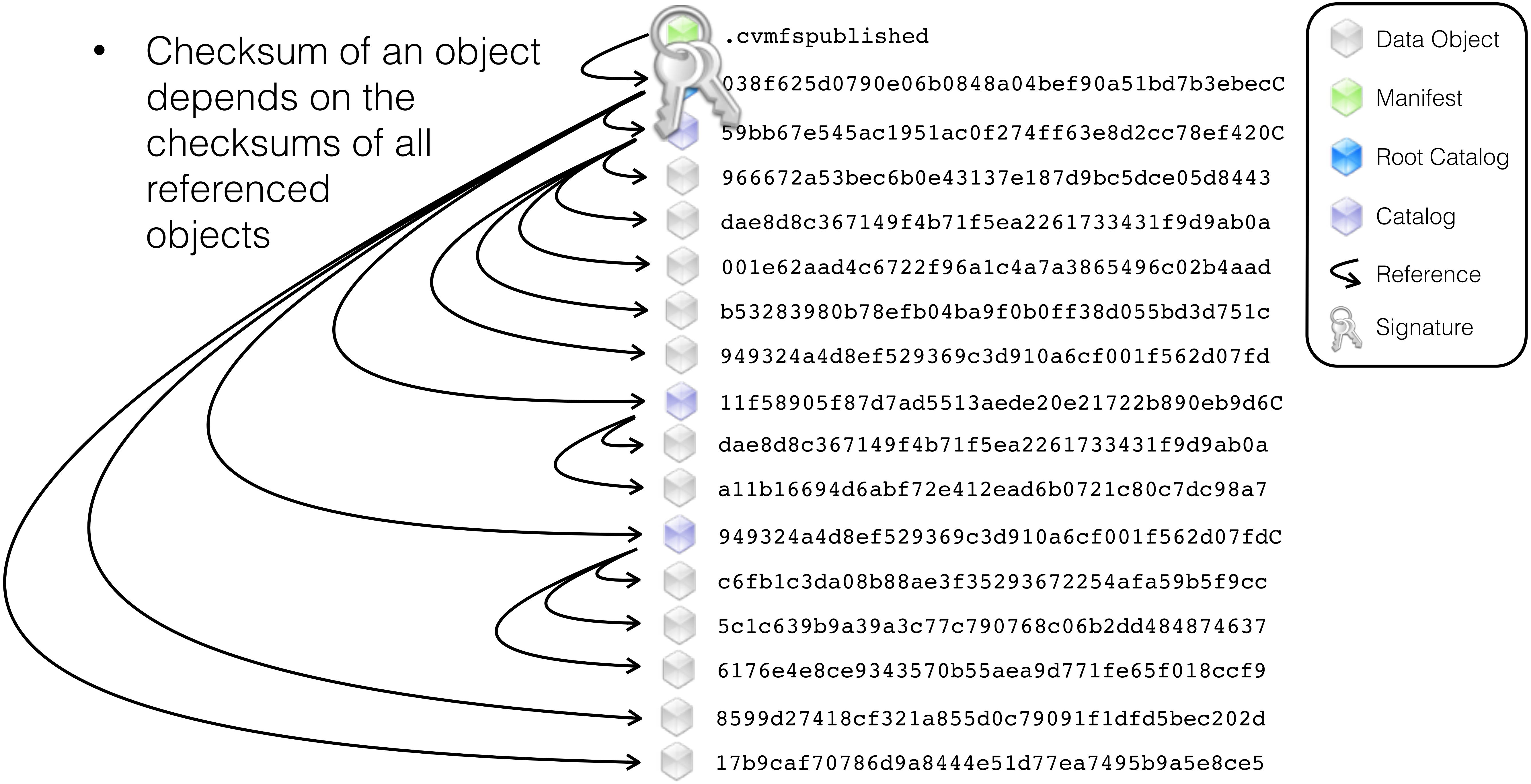
From POSIX to Blob-Objects

- **Hierarchy of File Catalogs**
  - File system meta-data, directories, symlinks, …
  - Content hashes of regular files
  - Root catalog is cryptographically signed
- **Content-Addressable Storage**
  - File de-duplication
  - Trivial file integrity checks
  - Insert-only semantic
- **Flat Namespace**
  - Perfect for HTTP caching



Integrity and Authenticity

- **Merkle Tree**
  - Checksum of an object depends on the checksums of all referenced objects



.cvmfspublished

038f625d0790e06b0848a04bef90a51bd7b3ebecC

59bb67e545ac1951ac0f274ff63e8d2cc78ef420C

966672a53bec6b0e43137e187d9bc5dce05d8443

dae8d8c367149f4b71f5ea2261733431f9d9ab0a

001e62aad4c6722f96a1c4a7a3865496c02b4aad

b53283980b78efb04ba9f0b0ff38d055bd3d751c

949324a4d8ef529369c3d910a6cf001f562d07fd

11f58905f87d7ad5513aede20e21722b890eb9d6C

dae8d8c367149f4b71f5ea2261733431f9d9ab0a

a11b16694d6abf72e412ead6b0721c80c7dc98a7

949324a4d8ef529369c3d910a6cf001f562d07fdC

c6fb1c3da08b88ae3f35293672254afa59b5f9cc

5c1c639b9a39a3c77c790768c06b2dd484874637

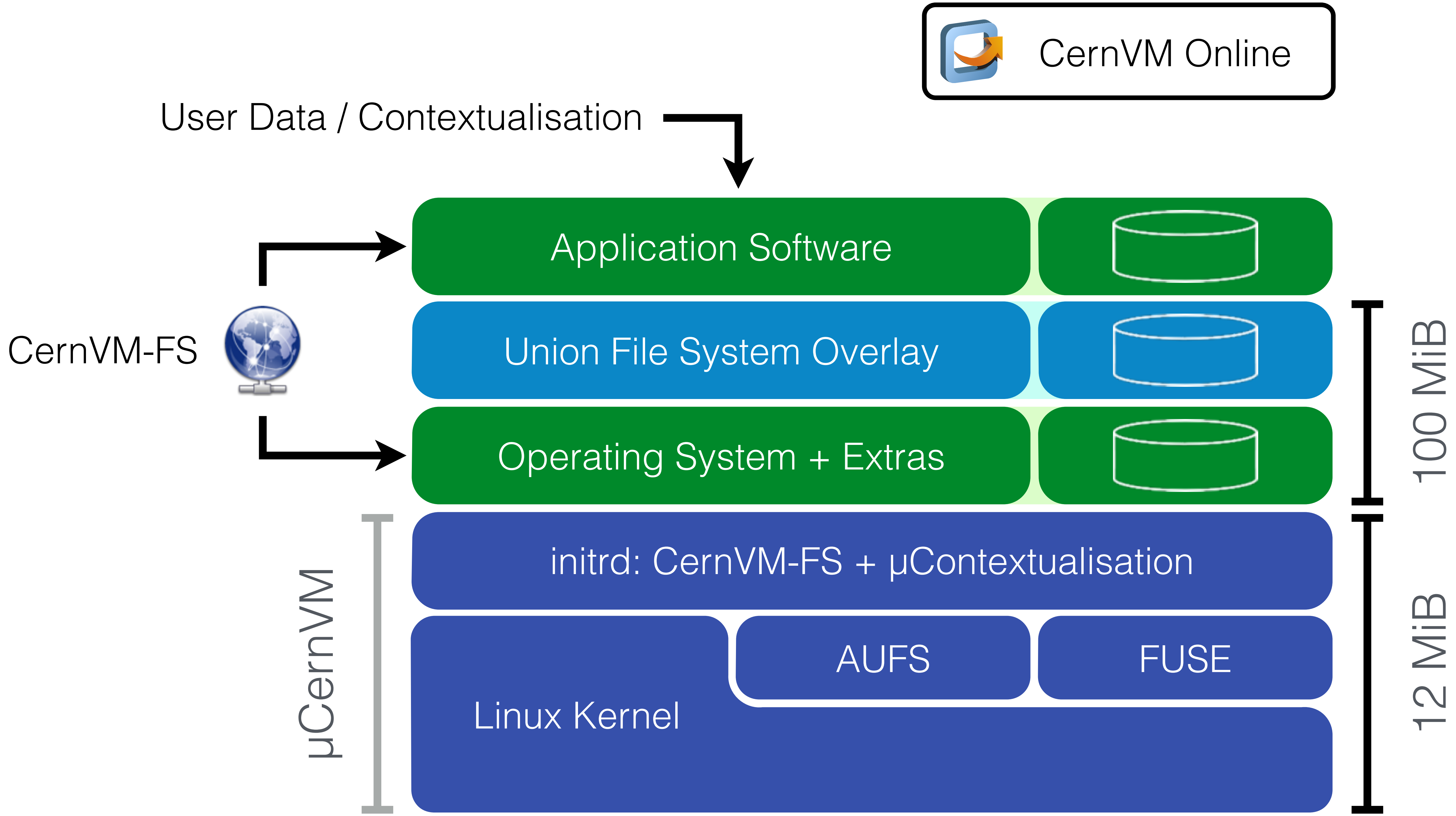
6176e4e8ce9343570b55aea9d771fe65f018ccf9

8599d27418cf321a855d0c79091f1dfd5bec202d

17b9caf70786d9a8444e51d77ea7495b9a5e8ce5

CernVM 3: Providing an Operating System on CernVM-FS

CernVM 3


CernVM Online
User Data / Contextualisation
CernVM-FS
Application Software
Union File System Overlay
Operating System + Extras
initrd: CernVM-FS + µContextualisation
AUFS
FUSE
Linux Kernel
µCernVM
100 MiB
12 MiB

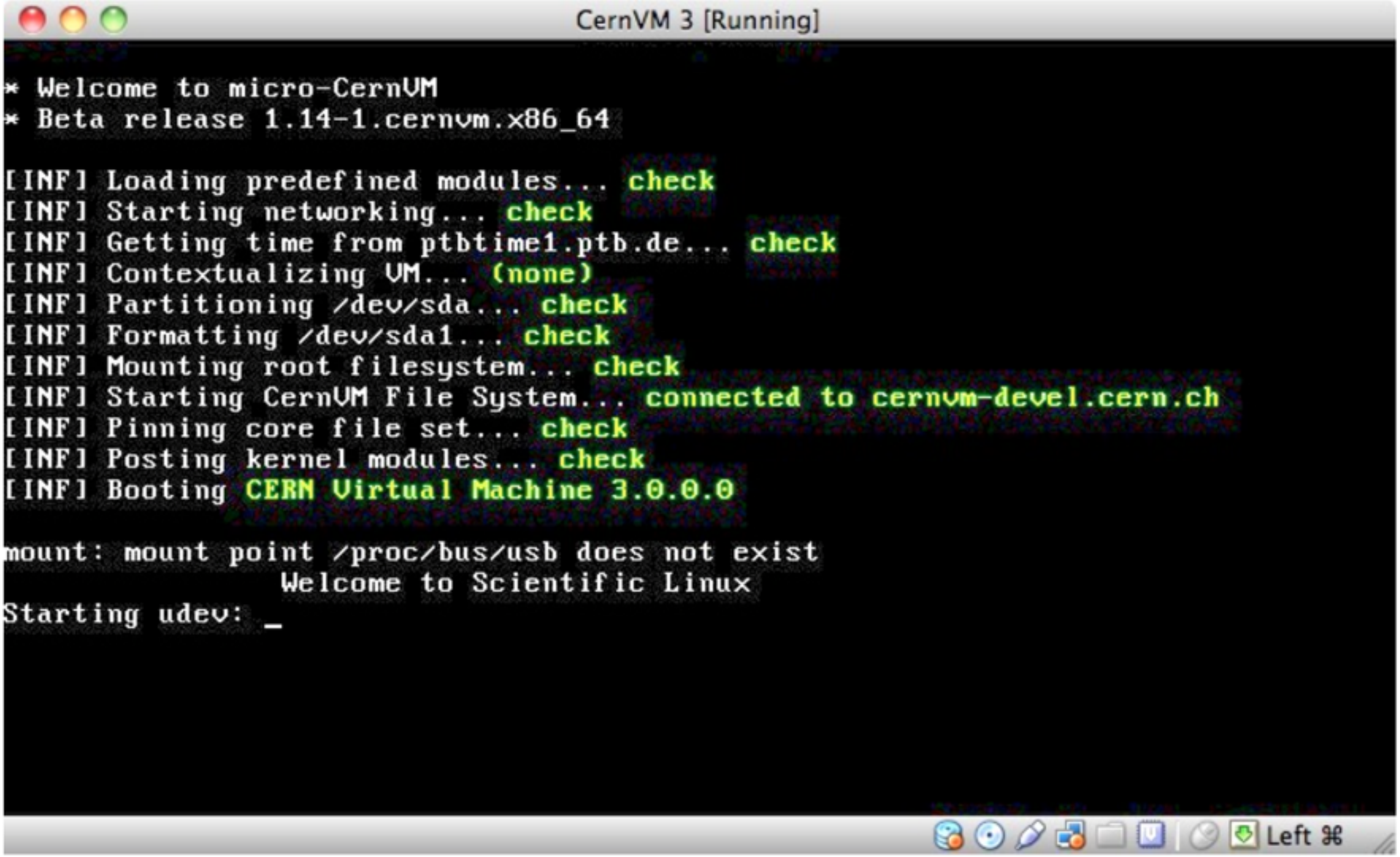
µCernVM Boot Loader

- CernVM Kernel: Linux Kernel 3.10 (long-term support - 2 years)
  - KSM, zRam, THP, cgroups, X32-ABI
  - AUFS, VMware drives, VBox drivers, OpenAFS
  - Minimal set of “virtualisation-friendly” device drivers
  - 8 MB binary (compared to 120 MB in SL6)

1. Execute SYSLINUX boot loader
2. Decompress and load Linux kernel
3. Decompress init ramdisk, execute customised `/init`

   1) Start networking
   2) Contextualise (supports EC2, OpenStack, OpenNebula, vSphere)
   3) [Partition,] [format and] mount scratch space
   4) Mount CernVM-FS (cernvm-prod.cern.ch)
   5) Mount AUFS root file system stack (copy-on-write)
   6) Change root file system and start operating system

Booting a CernVM 3

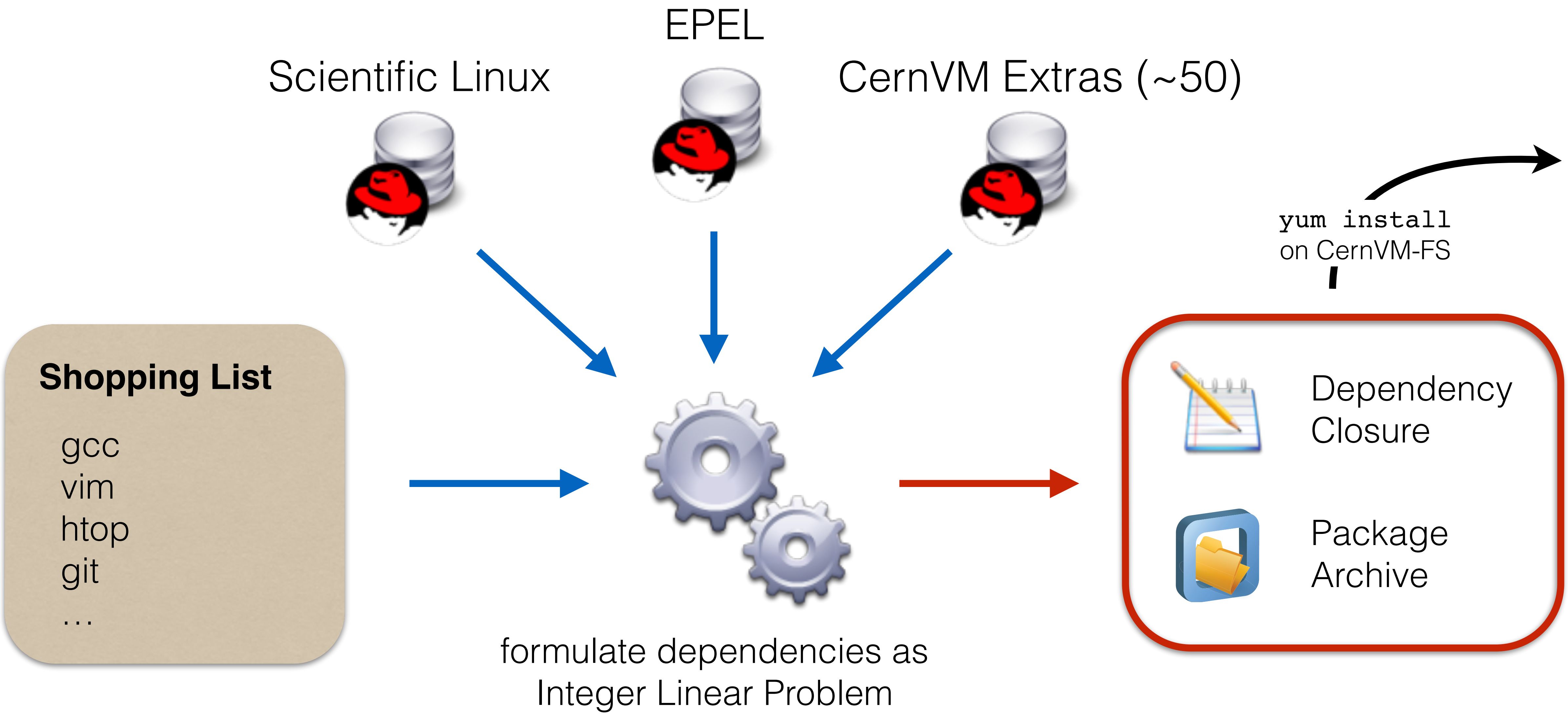


```
* Welcome to micro-CernVM
* Beta release 1.14-1.cernvm.x86_64

[INF] Loading predefined modules... check
[INF] Starting networking... check
[INF] Getting time from ptbtime1.ptb.de... check
[INF] Contextualizing VM... (none)
[INF] Partitioning /dev/sda... check
[INF] Formatting /dev/sda1... check
[INF] Mounting root filesystem... check
[INF] Starting CernVM File System... connected to cernvm-devel.cern.ch
[INF] Pinning core file set... check
[INF] Posting kernel modules... check
[INF] Booting CERN Virtual Machine 3.0.0.0

mount: mount point /proc/bus/usb does not exist
                Welcome to Scientific Linux
Starting udev: _
```

Build Process: Scientific Linux on CernVM-FS



Idea: automatically generate a fully versioned, closed package list from a “shopping list” of unversioned packages

Hypervisor / Cloud Controller Support

Hypervisor / Cloud Controller	Status
VirtualBox	✓
VMware	✓
KVM	✓
Xen	✓
Microsoft Hyper-V	✓
Parallels	⚡[1]
Openstack	✓
OpenNebula	✓
Amazon EC2	✓[2]
Google Compute Engine	✓[3]
Microsoft Azure	?
Docker	?

[1] Unclear license of the guest additions

[2] Only tested with ephemeral storage, not with EBS backed instances

[3] Only amiconfig contextualisation

Summary

- **CernVM-FS**
  - Global software distribution system
  - File system history preservation through snapshots
  - Replication and aggressive caching for scalability
  - Centrally installed software repository
  - On-Demand download

- **CernVM**
  - Tiny (20MB) virtual machine image that adapts
  - µCernVM + OS template on CernVM-FS + Contextualisation
  - Use Cases: IaaS, volunteer computing, long-term data preservation, development environment, open data appliance

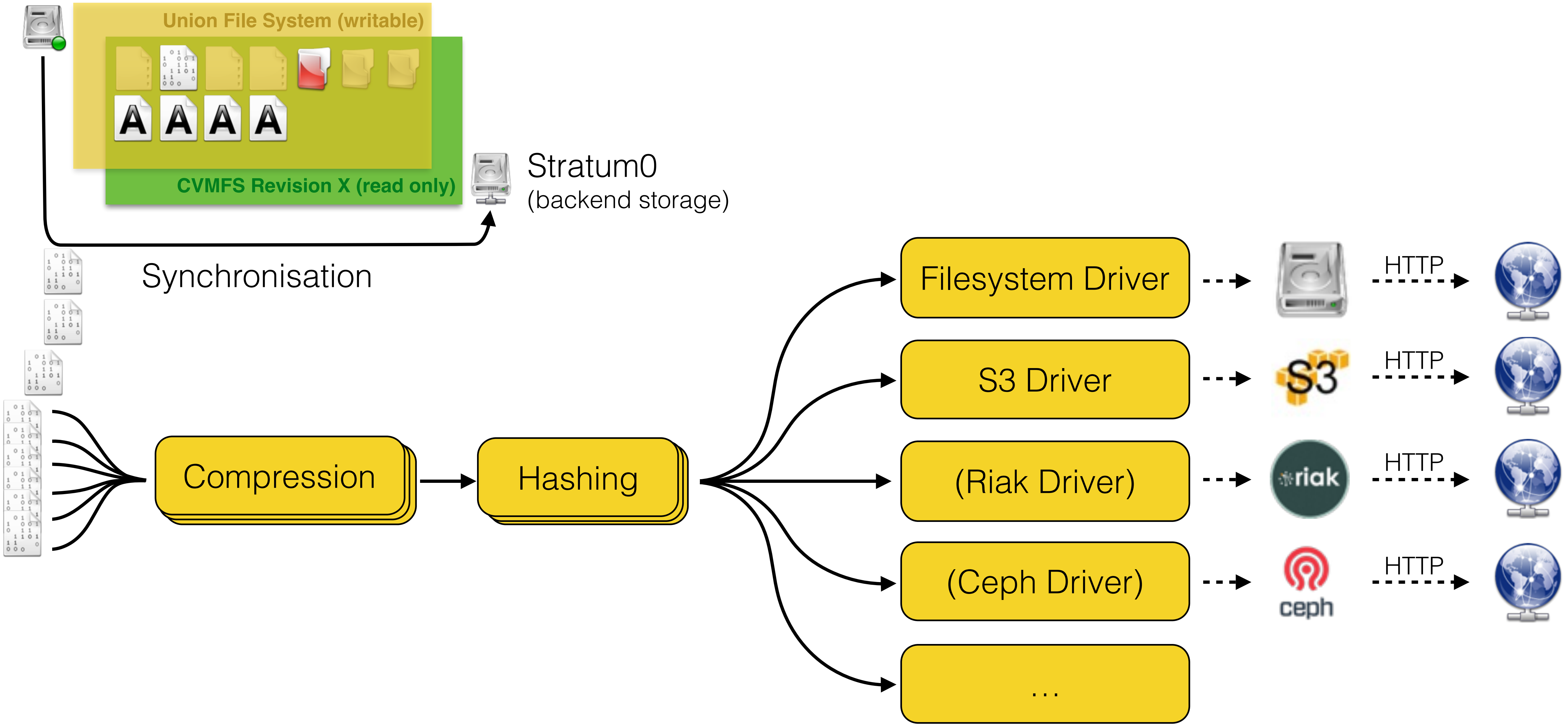
Pointers to Useful Further Resources

Fork me on GitHub

- **Documentation**
  - http://cernvm.cern.ch/portal/filesystem/techinformation
- **Download and Installation Instructions**
  - http://cernvm.cern.ch/portal/filesystem/downloads
- **Mailing Lists**
  - cvmfs-talk@cern.ch, cvmfs-testing@cern.ch, cvmfs-devel@cern.ch
  - cernvm-talk@cern.ch
- **Bug Tracker**
  - https://sft.its.cern.ch/jira/browse/CVM
- **Source Code** (CernVM-FS, Puppet Module)
  - https://github.com/cvmfs
  - https://github.com/cernvm
- **Nightly Builds**
  - https://ecsft.cern.ch/dist/cvmfs/nightlies/
- **CernVM as an Open Data Appliance**
  - http://opendata.cern.ch/VM

Alternative Storage Backends

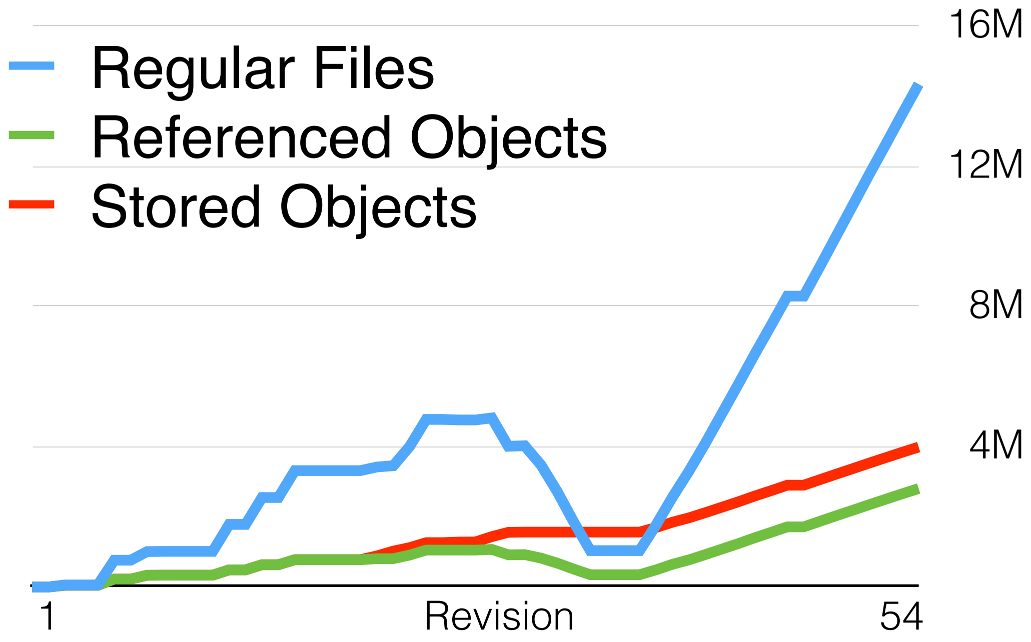
- “Plug-in” Architecture since CernVM-FS Server 2.1.17
  - Potential for adding alternative storage drivers (S3, Ceph, Basho Riak, OpenStack Swift, …)



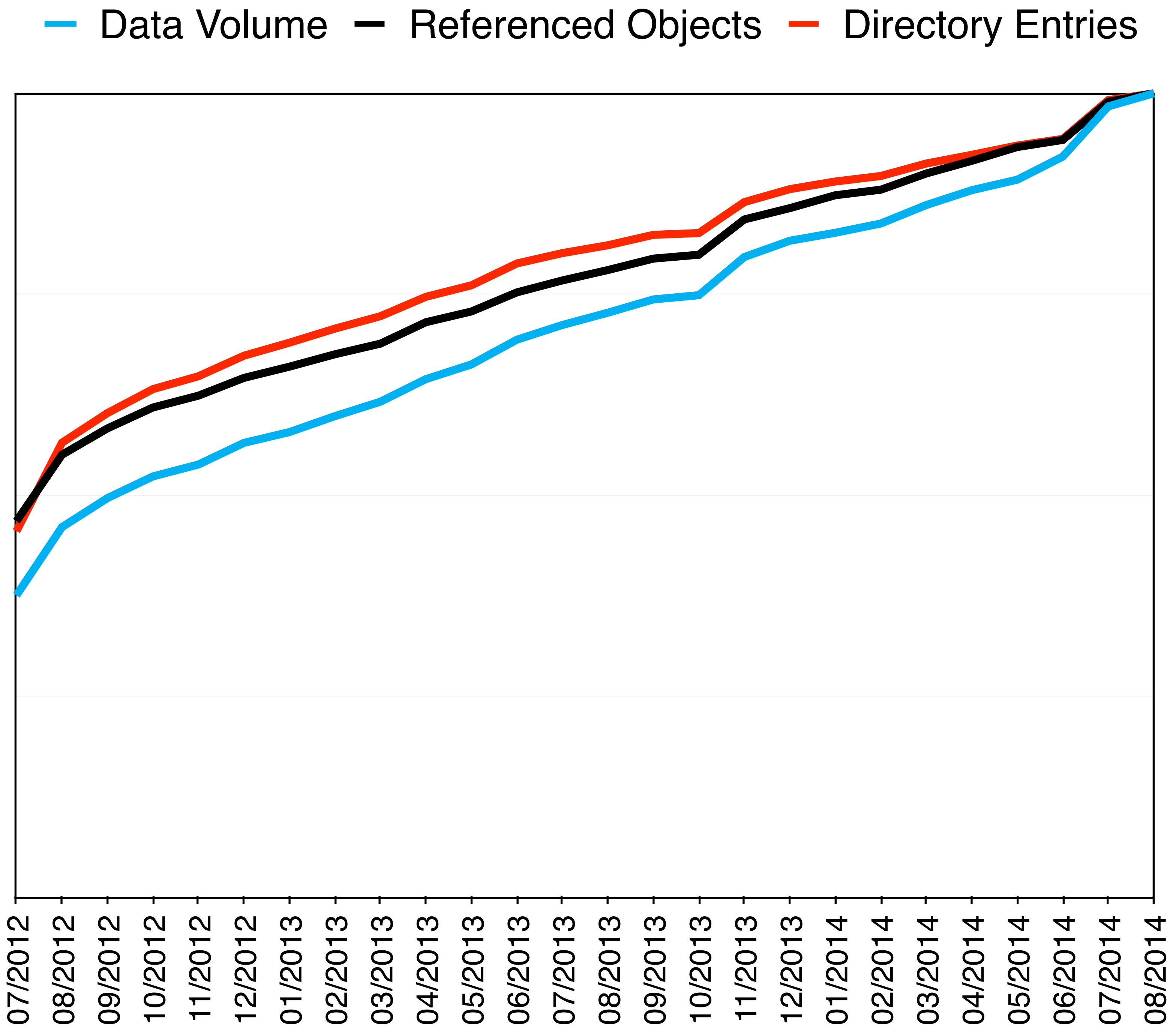
Garbage Collection on a CernVM-FS Server

- CernVM-FS initially designed as *insert-only* system
  - Historic snapshots stay reachable (long term preservation)
  - But: ever-growing backend storage volume

- Use-Case: Publishing of nightly integration build results
  - Requested by CMS and LHCb
  - Large amount of new files every day (f.e. LHCb: 1M files - 50 GiB)
  - Historic snapshots are of no interest
  - Garbage collection on revision level:
    - Sweep individual (old) snapshots
    - Sweep complete history

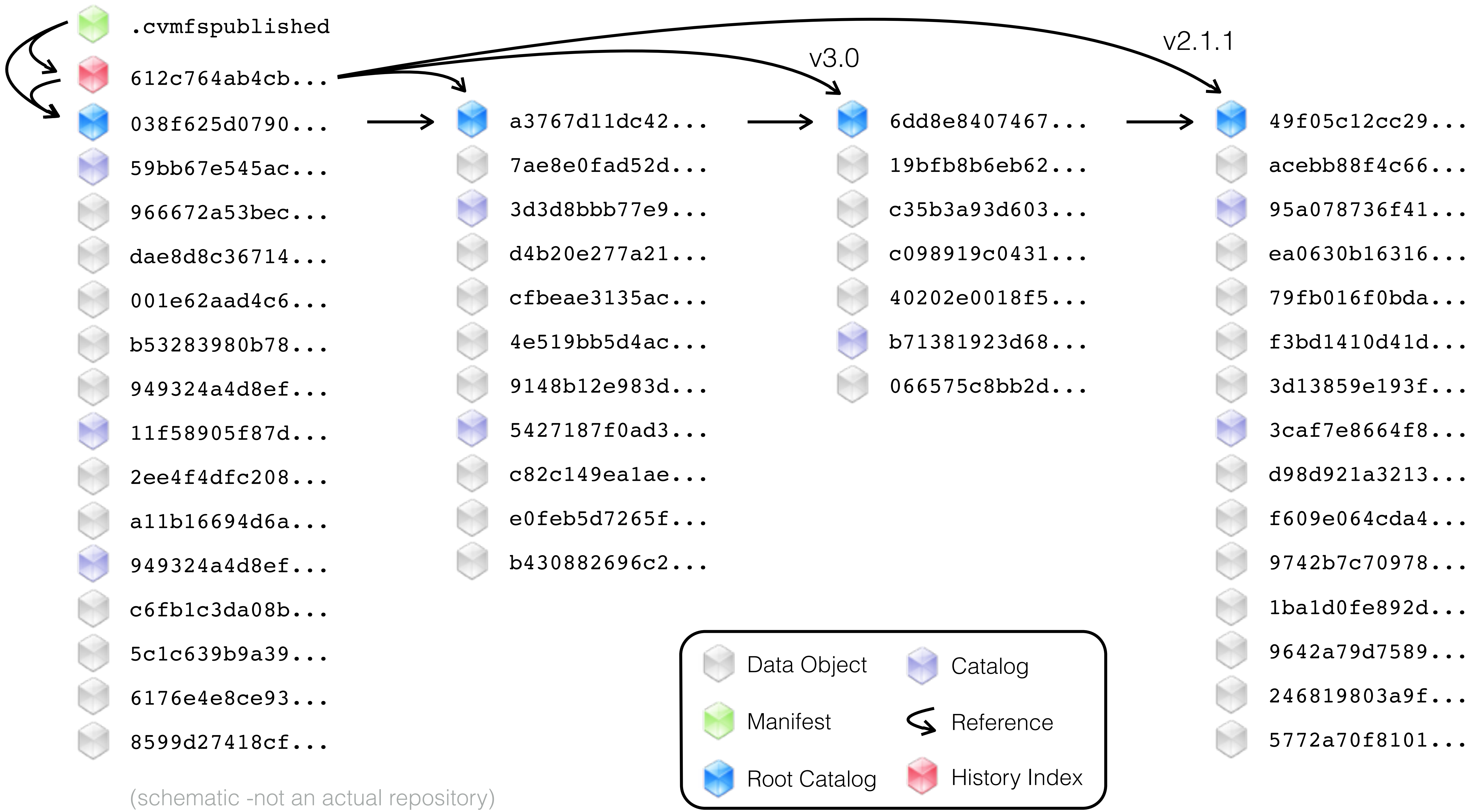


Growth Statistics for atlas.cern.ch



- Example Repository: **atlas.cern.ch**
- Size approximately doubled in two years
- Maximal values:
  - Data: 2.1 TiB
    Entries: 48.0 M
    Objects: ~3.8 M

Garbage Collection on CernVM-FS Servers



.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

v3.0

6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

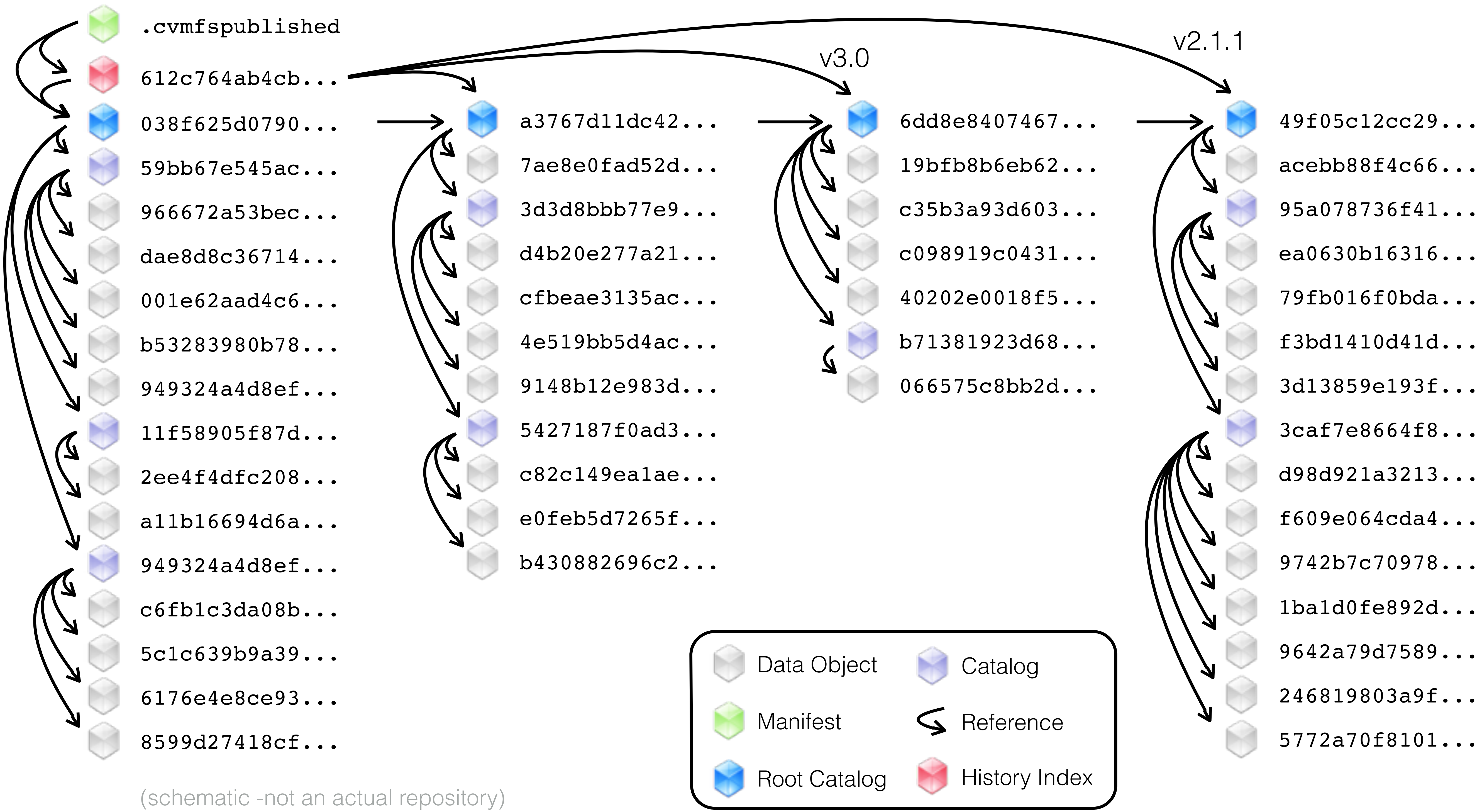
v2.1.1

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

Data Object
Catalog
Manifest
Reference
Root Catalog
History Index

(schematic -not an actual repository)

Garbage Collection on CernVM-FS Servers



.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

v3.0

6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

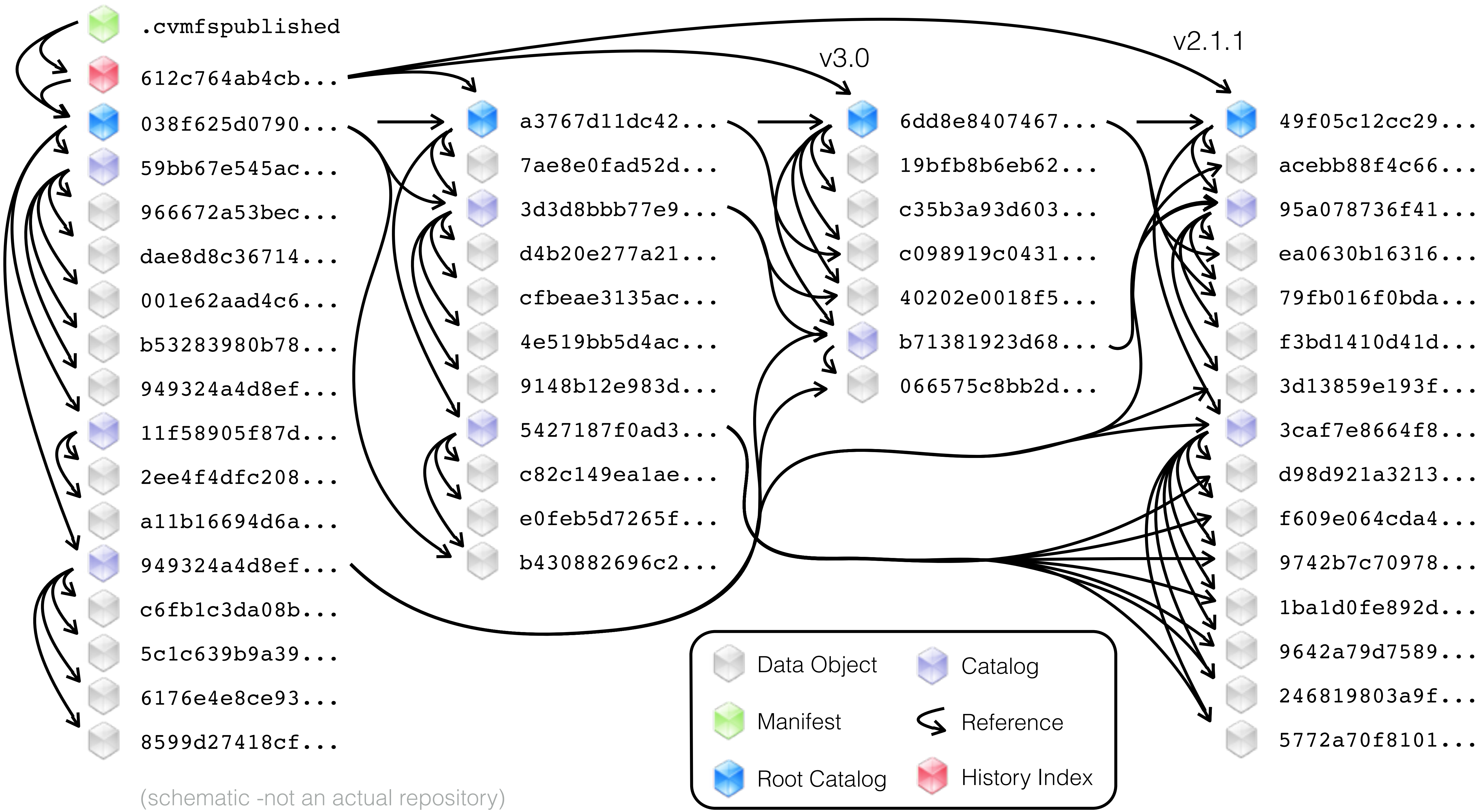
v2.1.1

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

Data Object
Catalog
Manifest
Reference
Root Catalog
History Index

(schematic -not an actual repository)

Garbage Collection on CernVM-FS Servers



.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

v3.0

6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

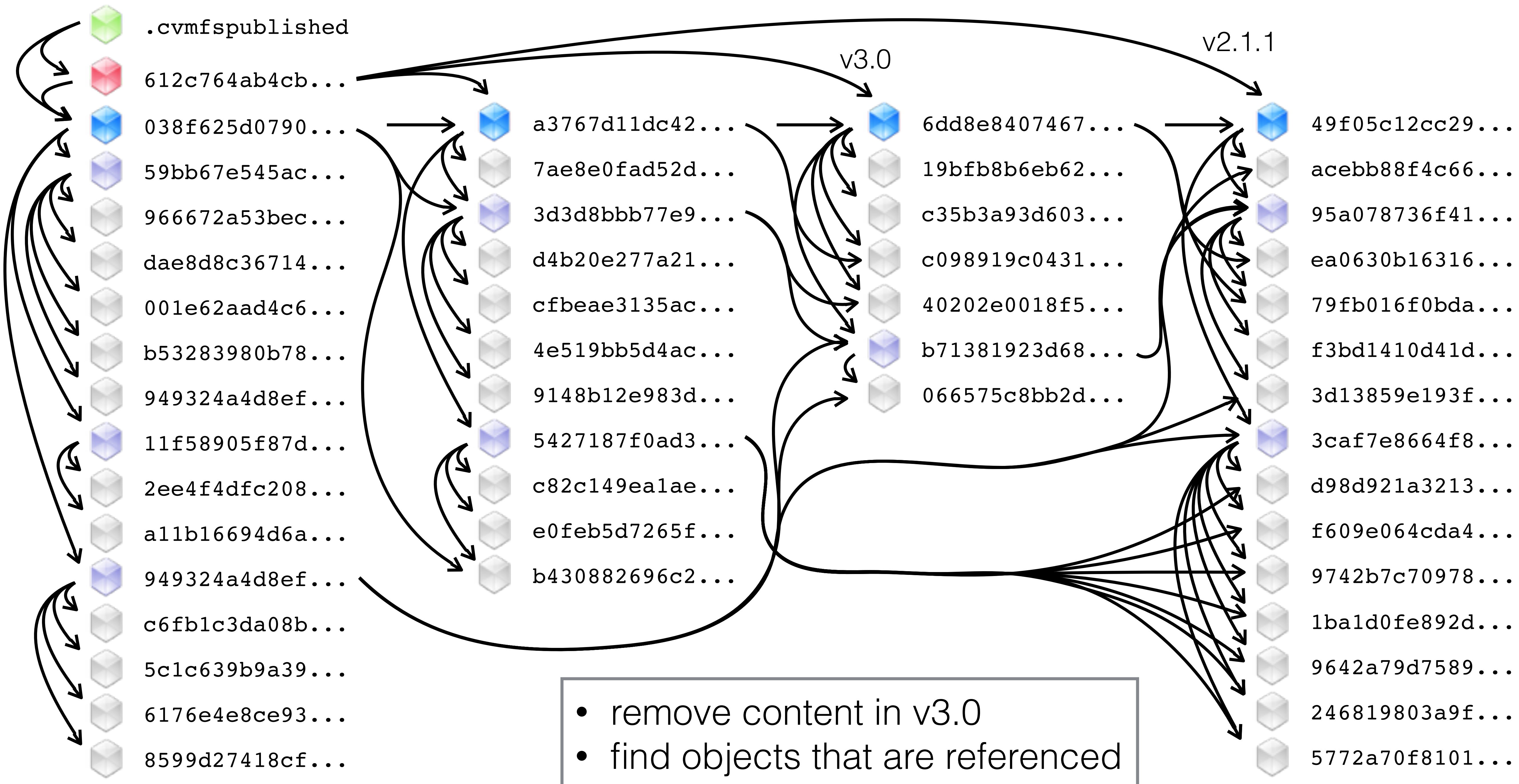
v2.1.1

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

Data Object
Catalog
Manifest
Reference
Root Catalog
History Index

(schematic -not an actual repository)

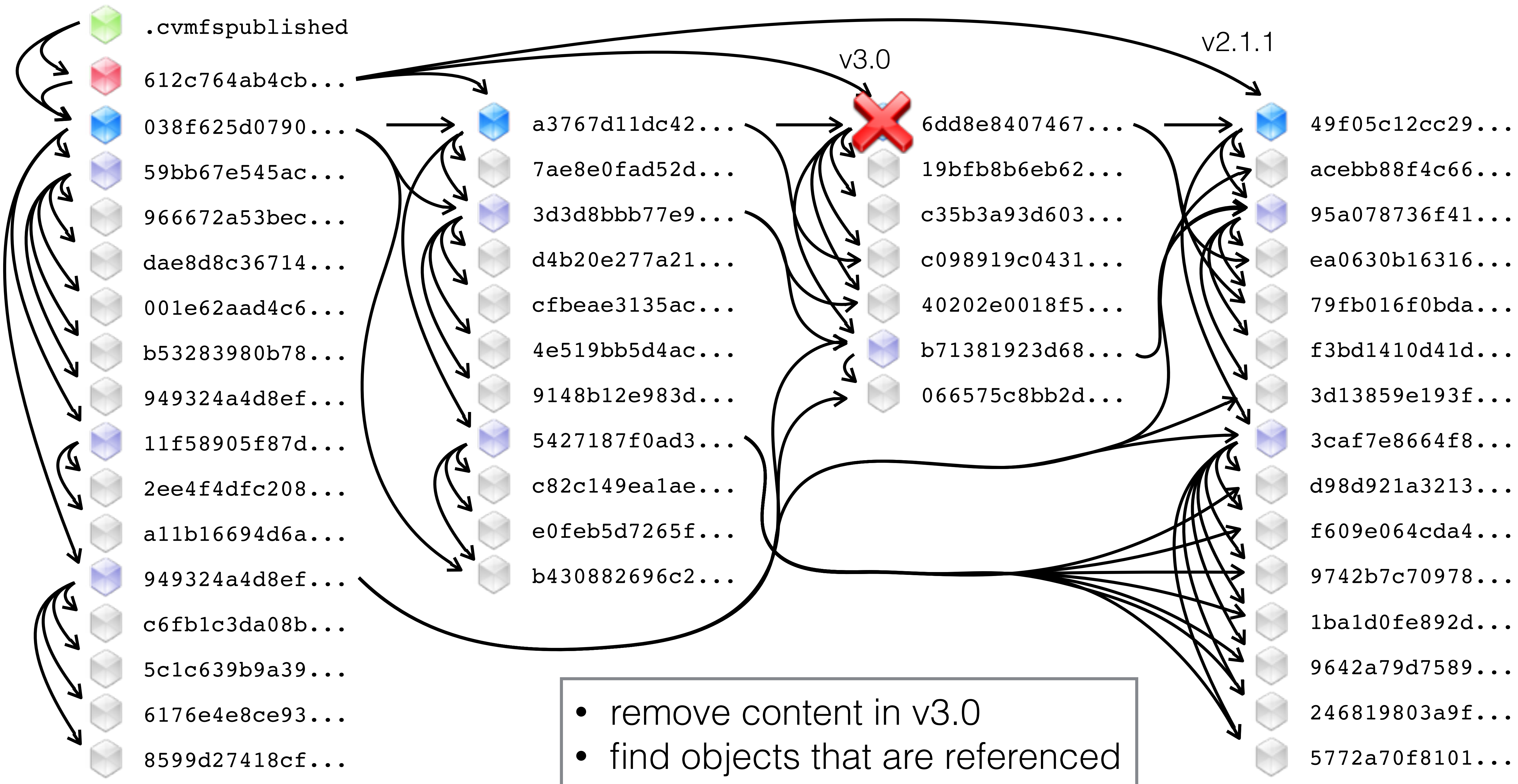
Garbage Collection on CernVM-FS Servers



- remove content in v3.0
- find objects that are referenced nowhere else

(schematic -not an actual repository)

Garbage Collection on CernVM-FS Servers



.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

v3.0

6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

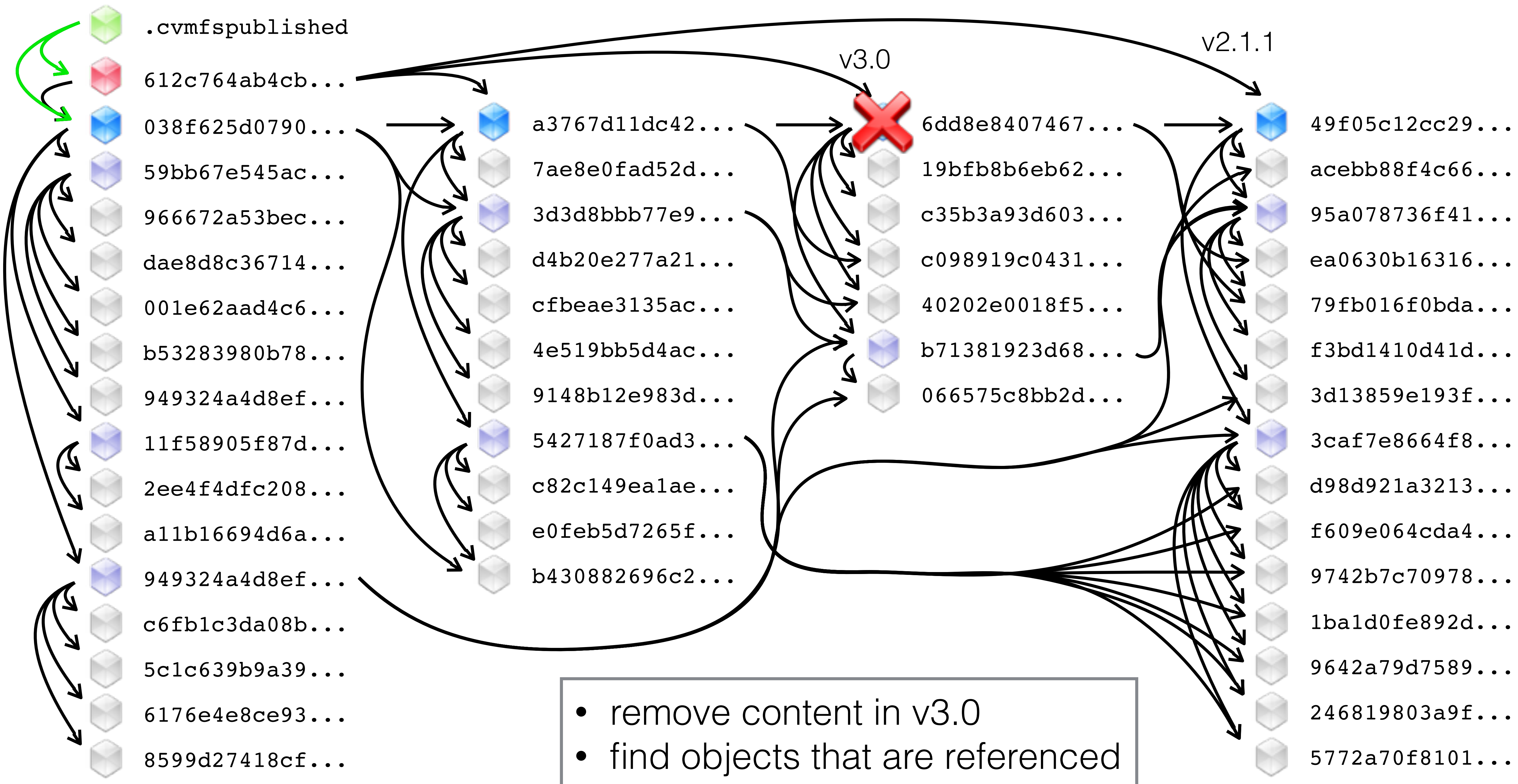
v2.1.1

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

- remove content in v3.0
- find objects that are referenced nowhere else

(schematic -not an actual repository)

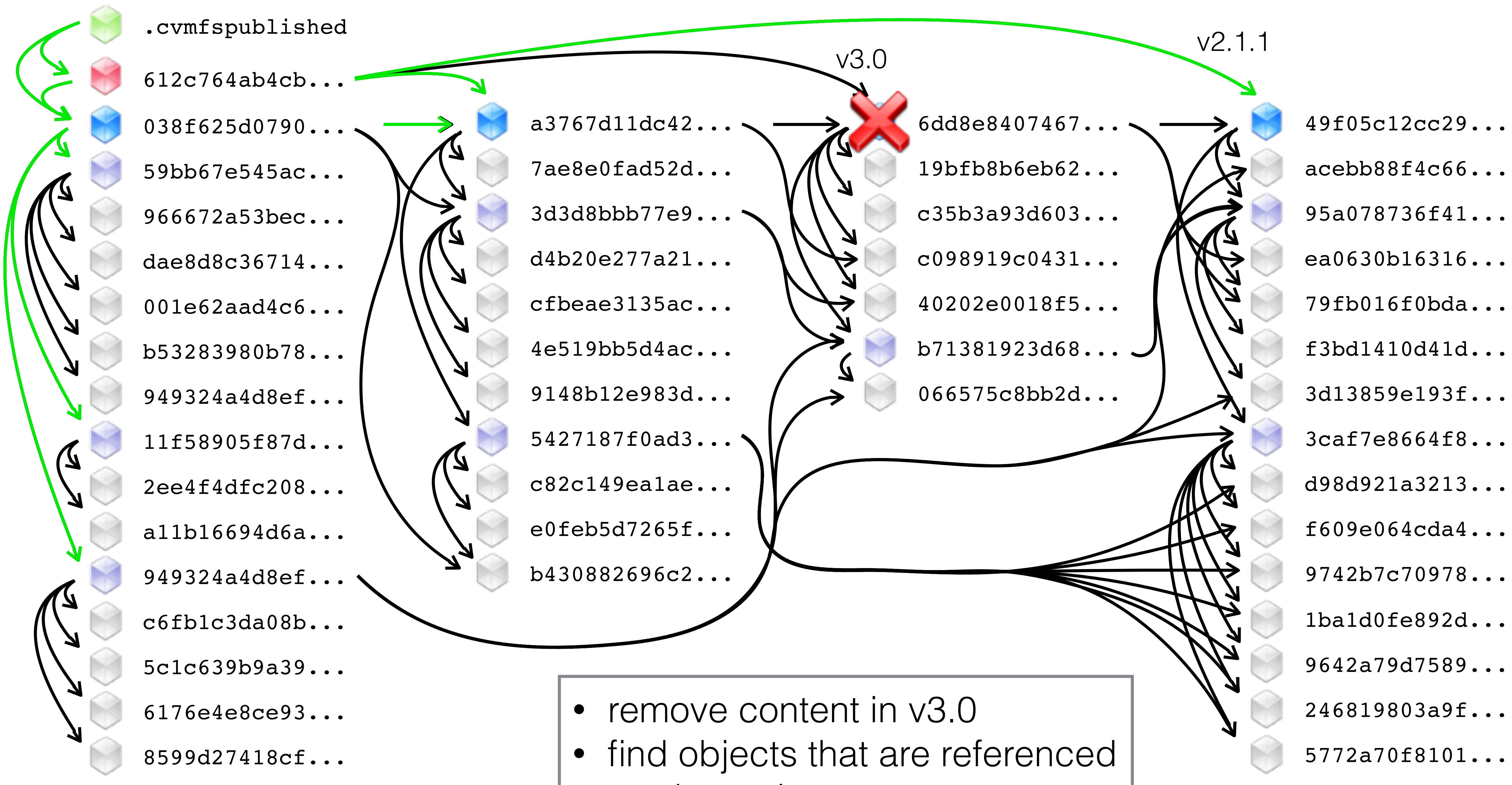
Garbage Collection on CernVM-FS Servers



- remove content in v3.0
- find objects that are referenced nowhere else

(schematic -not an actual repository)

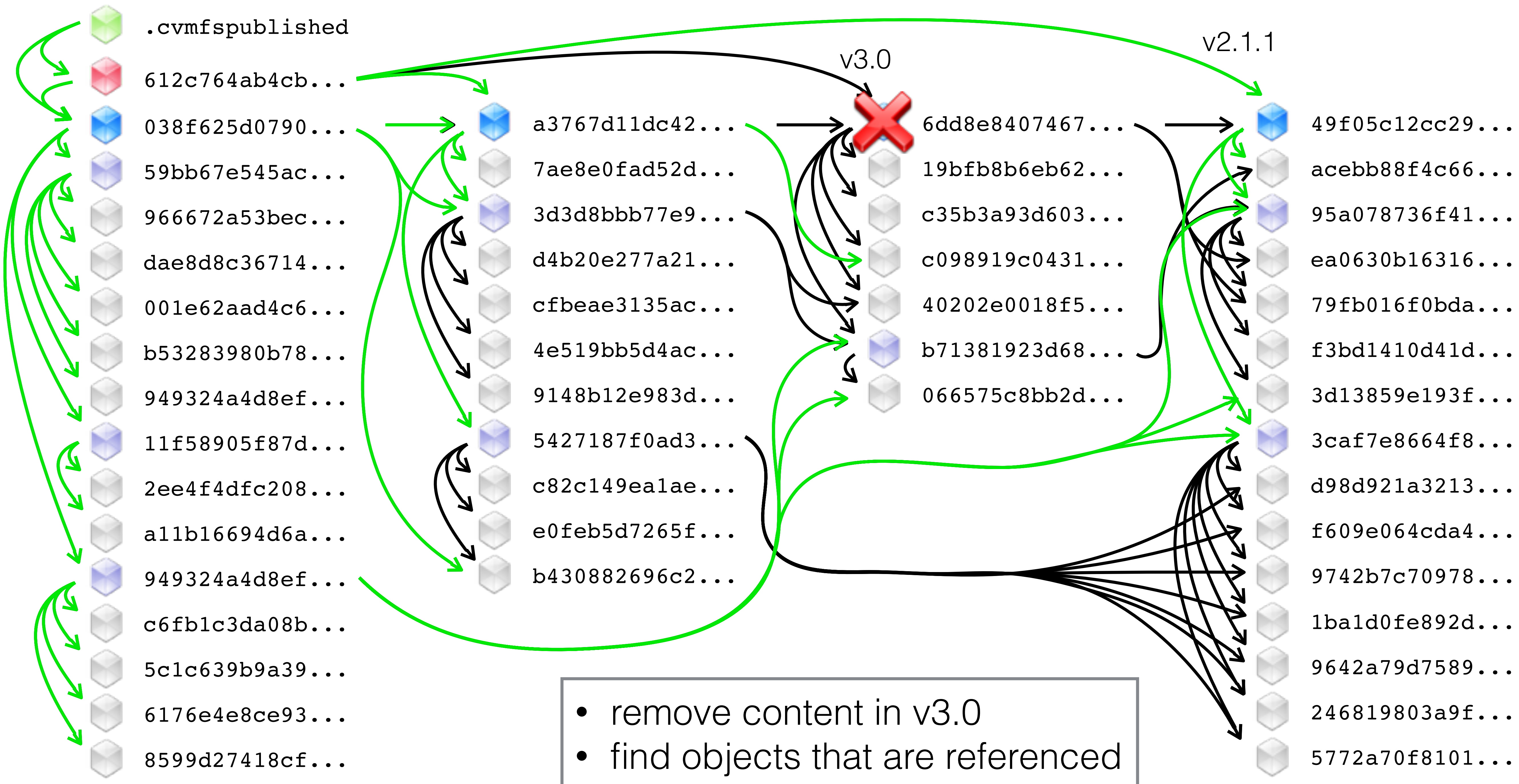
Garbage Collection on CernVM-FS Servers



- remove content in v3.0
- find objects that are referenced nowhere else

(schematic -not an actual repository)

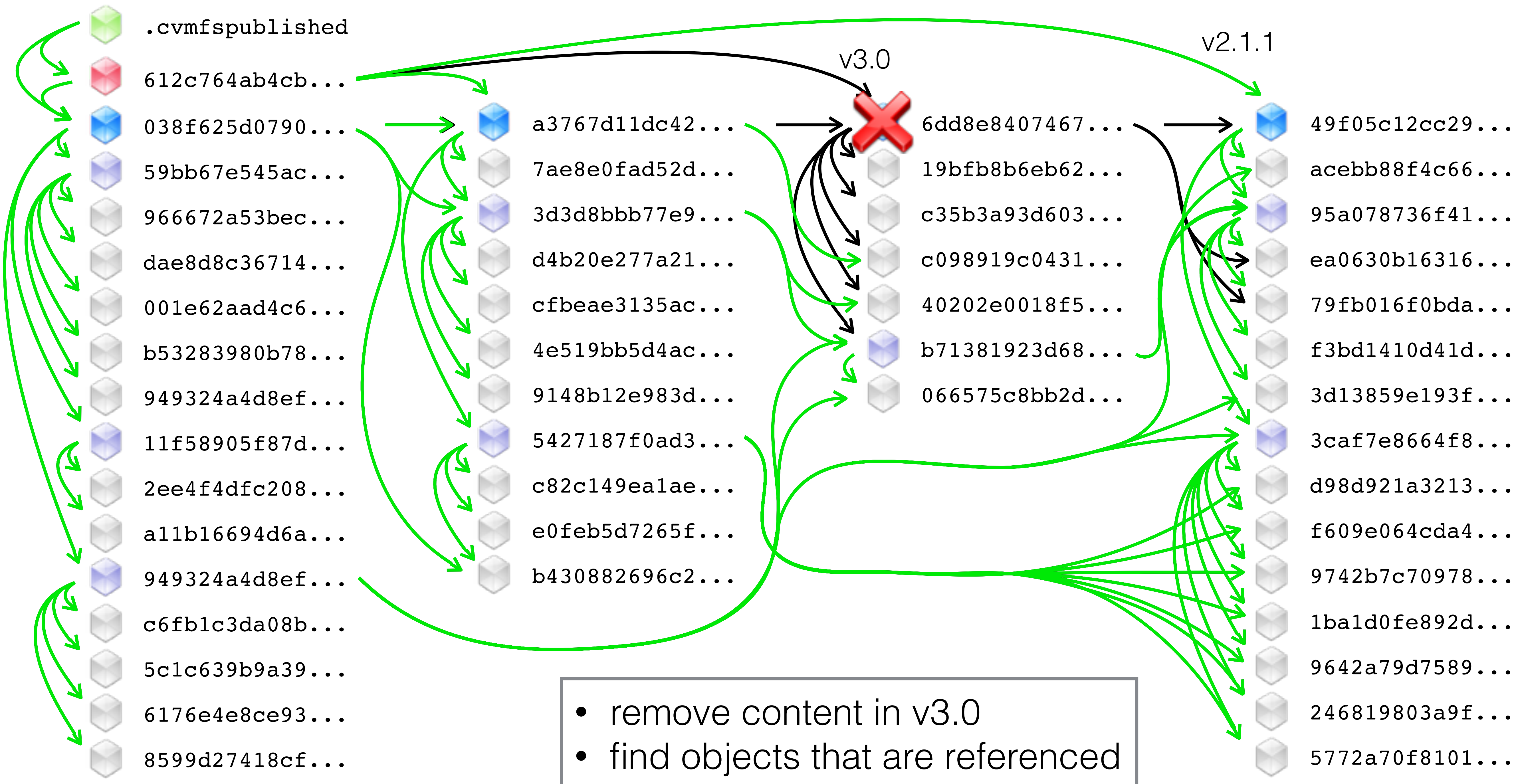
Garbage Collection on CernVM-FS Servers



- remove content in v3.0
- find objects that are referenced nowhere else

(schematic -not an actual repository)

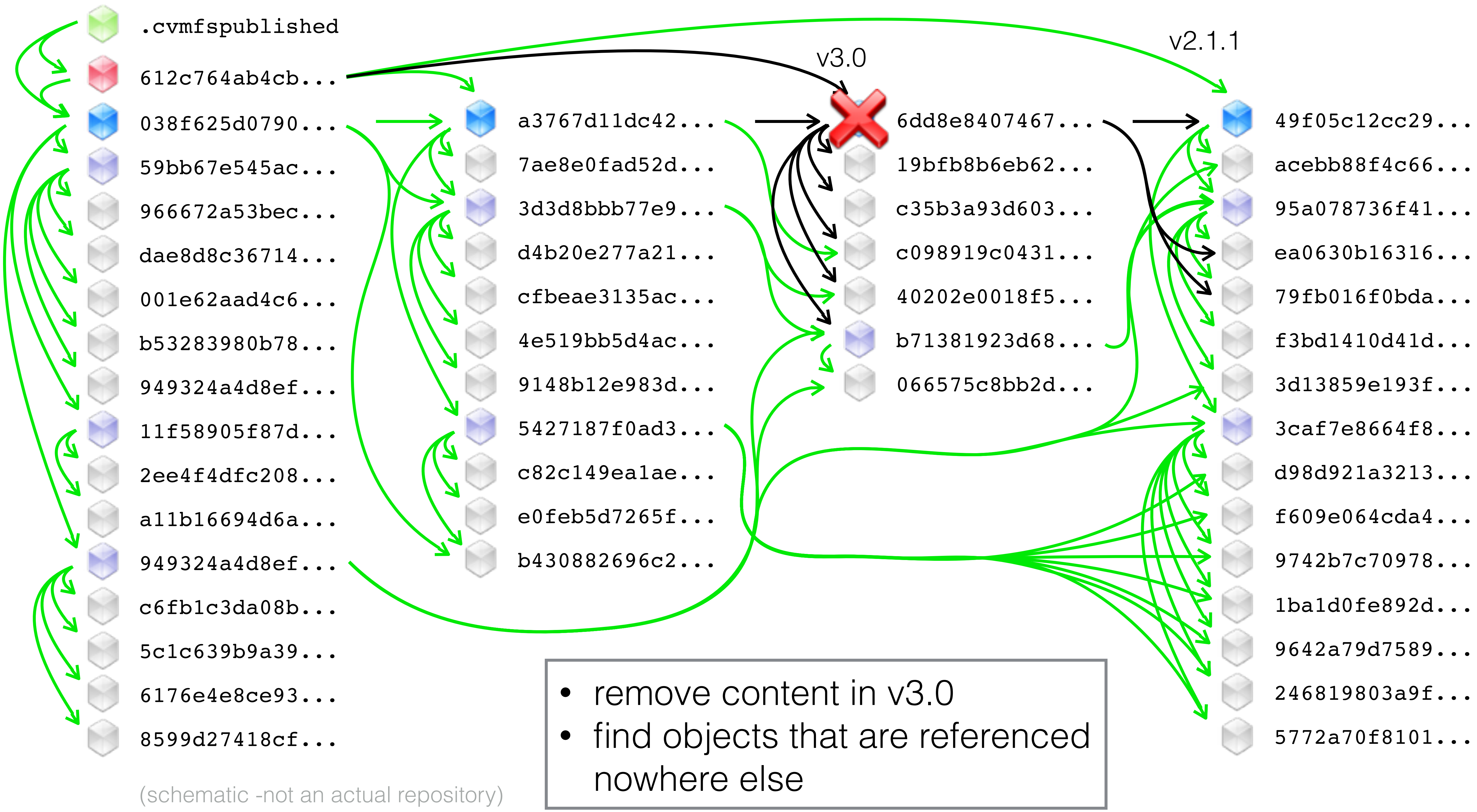
Garbage Collection on CernVM-FS Servers



- remove content in v3.0
- find objects that are referenced nowhere else

(schematic -not an actual repository)

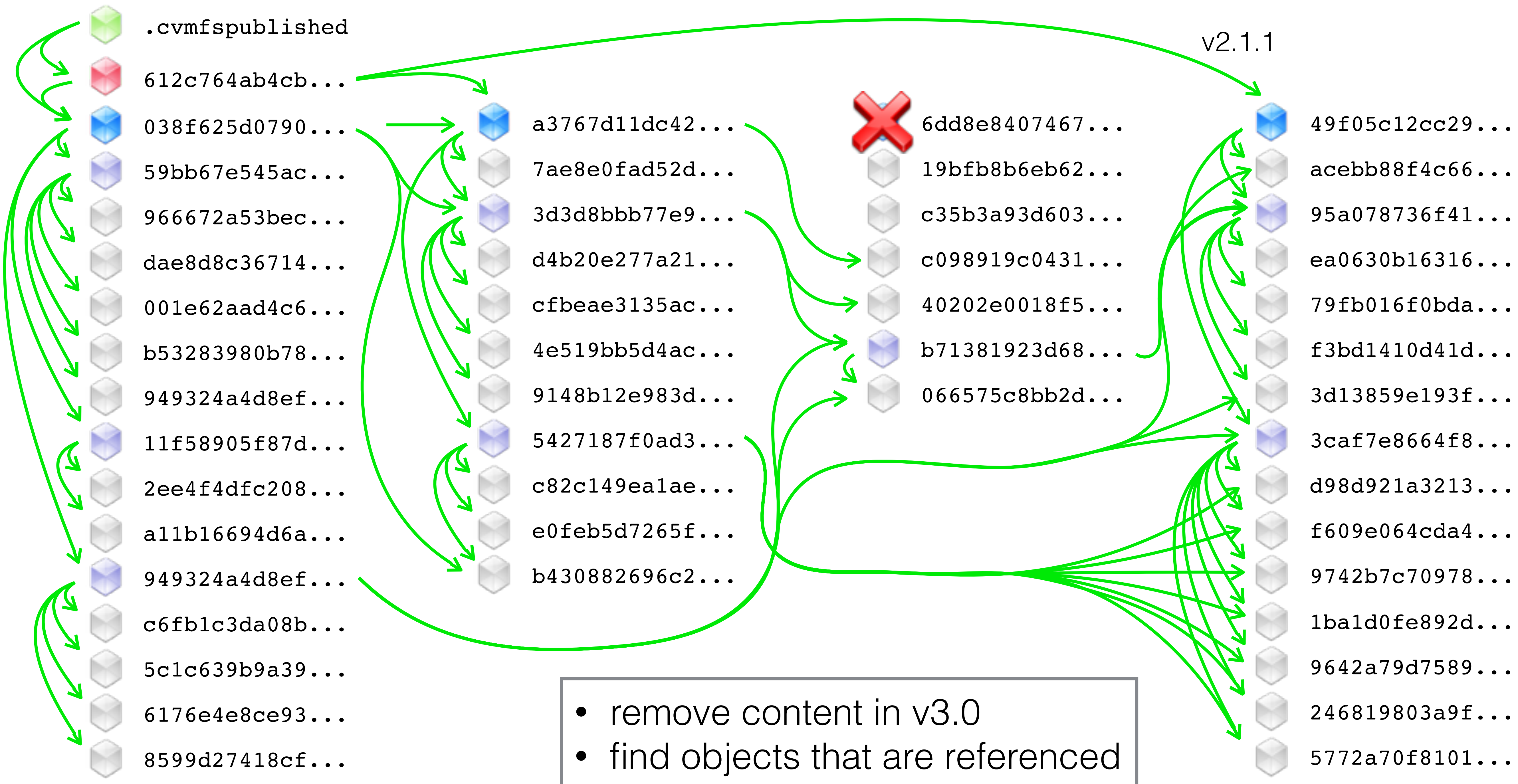
Garbage Collection on CernVM-FS Servers



(schematic -not an actual repository)

- remove content in v3.0
- find objects that are referenced nowhere else

Garbage Collection on CernVM-FS Servers



.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

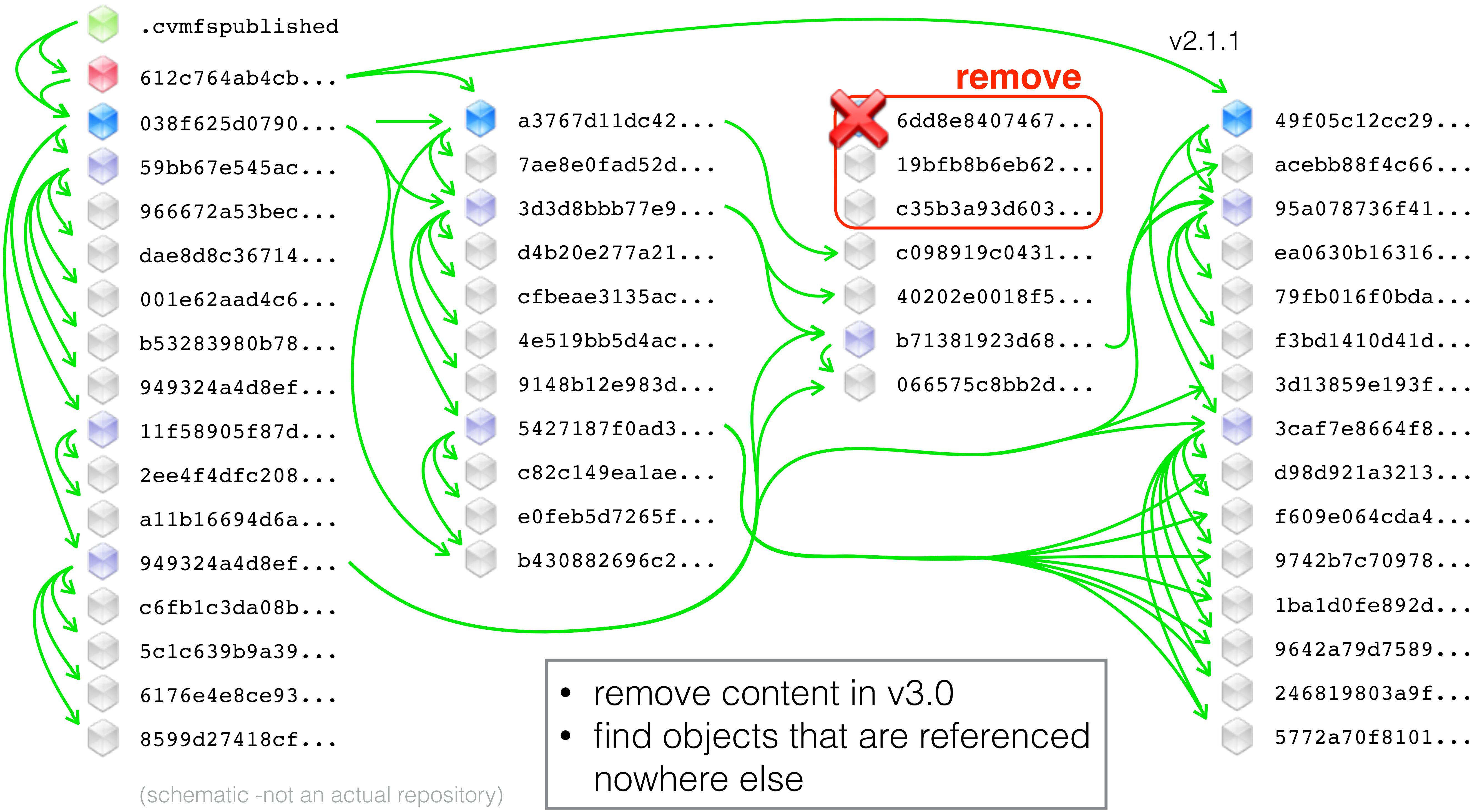
v2.1.1

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

- remove content in v3.0
- find objects that are referenced nowhere else

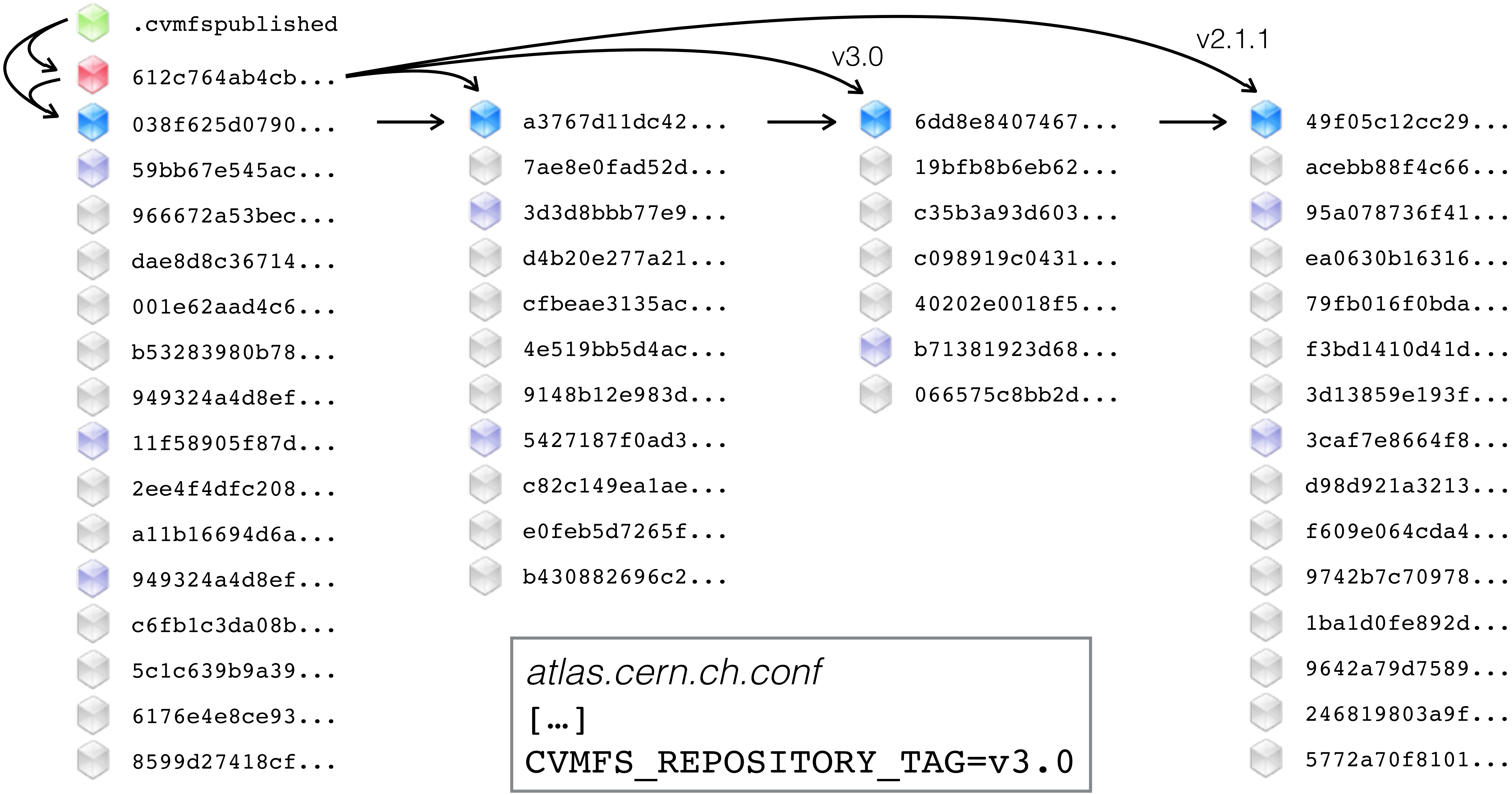
(schematic -not an actual repository)

Garbage Collection on CernVM-FS Servers



- remove content in v3.0
- find objects that are referenced nowhere else

CernVM-FS Snapshot History



.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

v3.0

6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

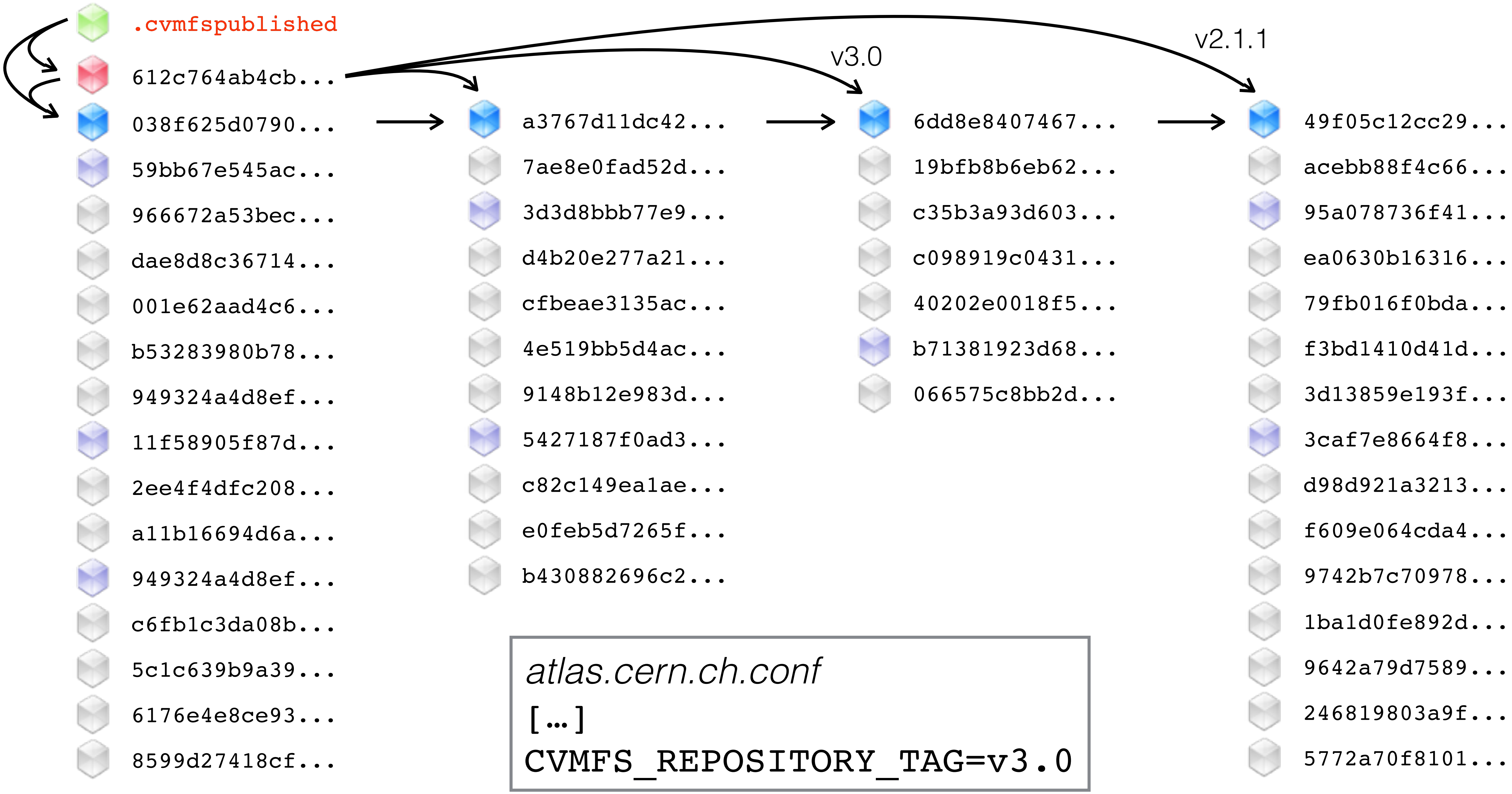
v2.1.1

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

atlas.cern.ch.conf

```
[…]
CVMFS_REPOSITORY_TAG=v3.0
```

CernVM-FS Snapshot History



.cvmfspublished

612c764ab4cb...

038f625d0790...

59bb67e545ac...

966672a53bec...

dae8d8c36714...

001e62aad4c6...

b53283980b78...

949324a4d8ef...

11f58905f87d...

2ee4f4dfc208...

a11b16694d6a...

949324a4d8ef...

c6fb1c3da08b...

5c1c639b9a39...

6176e4e8ce93...

8599d27418cf...

a3767d11dc42...

7ae8e0fad52d...

3d3d8bbb77e9...

d4b20e277a21...

cfbeae3135ac...

4e519bb5d4ac...

9148b12e983d...

5427187f0ad3...

c82c149ea1ae...

e0feb5d7265f...

b430882696c2...

v3.0

6dd8e8407467...

19bfb8b6eb62...

c35b3a93d603...

c098919c0431...

40202e0018f5...

b71381923d68...

066575c8bb2d...

v2.1.1

49f05c12cc29...

acebb88f4c66...

95a078736f41...

ea0630b16316...

79fb016f0bda...

f3bd1410d41d...

3d13859e193f...

3caf7e8664f8...

d98d921a3213...

f609e064cda4...

9742b7c70978...

1ba1d0fe892d...

9642a79d7589...

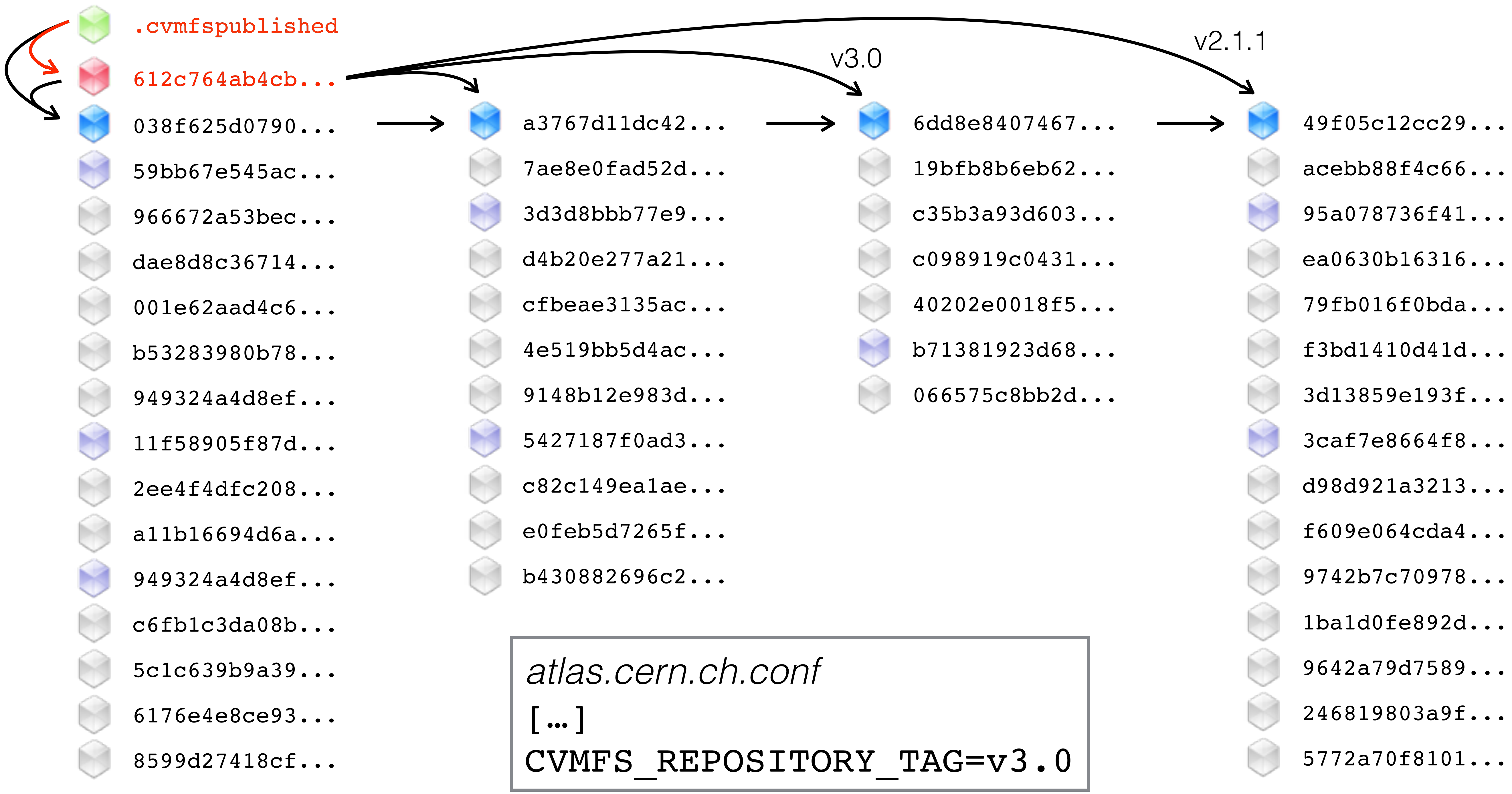
246819803a9f...

5772a70f8101...

atlas.cern.ch.conf

```
[…]
CVMFS_REPOSITORY_TAG=v3.0
```

CernVM-FS Snapshot History



v3.0

v2.1.1

.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

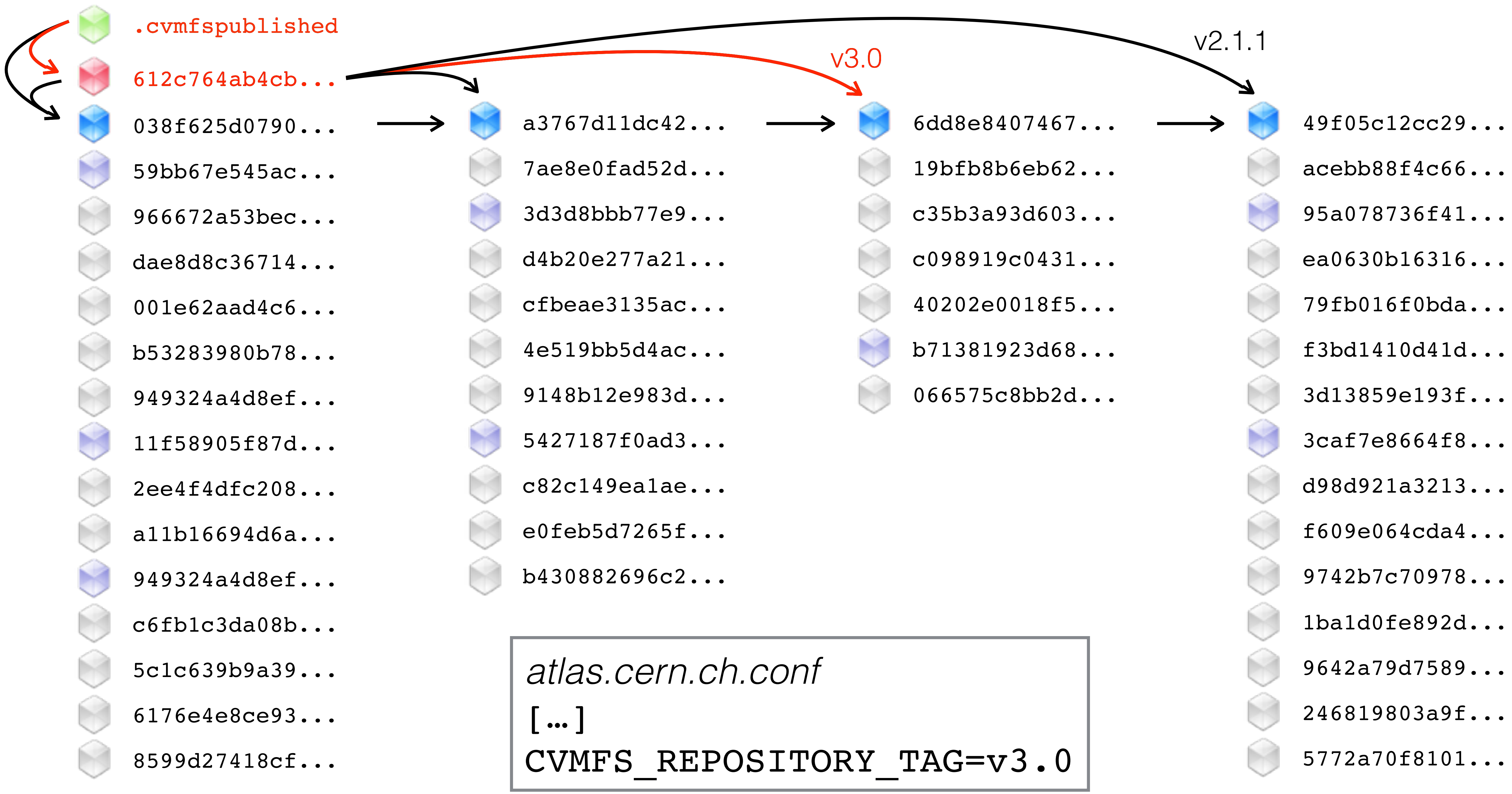
6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

atlas.cern.ch.conf

```
[…]
CVMFS_REPOSITORY_TAG=v3.0
```

CernVM-FS Snapshot History



.cvmfspublished
612c764ab4cb...
038f625d0790...
59bb67e545ac...
966672a53bec...
dae8d8c36714...
001e62aad4c6...
b53283980b78...
949324a4d8ef...
11f58905f87d...
2ee4f4dfc208...
a11b16694d6a...
949324a4d8ef...
c6fb1c3da08b...
5c1c639b9a39...
6176e4e8ce93...
8599d27418cf...

a3767d11dc42...
7ae8e0fad52d...
3d3d8bbb77e9...
d4b20e277a21...
cfbeae3135ac...
4e519bb5d4ac...
9148b12e983d...
5427187f0ad3...
c82c149ea1ae...
e0feb5d7265f...
b430882696c2...

v3.0

6dd8e8407467...
19bfb8b6eb62...
c35b3a93d603...
c098919c0431...
40202e0018f5...
b71381923d68...
066575c8bb2d...

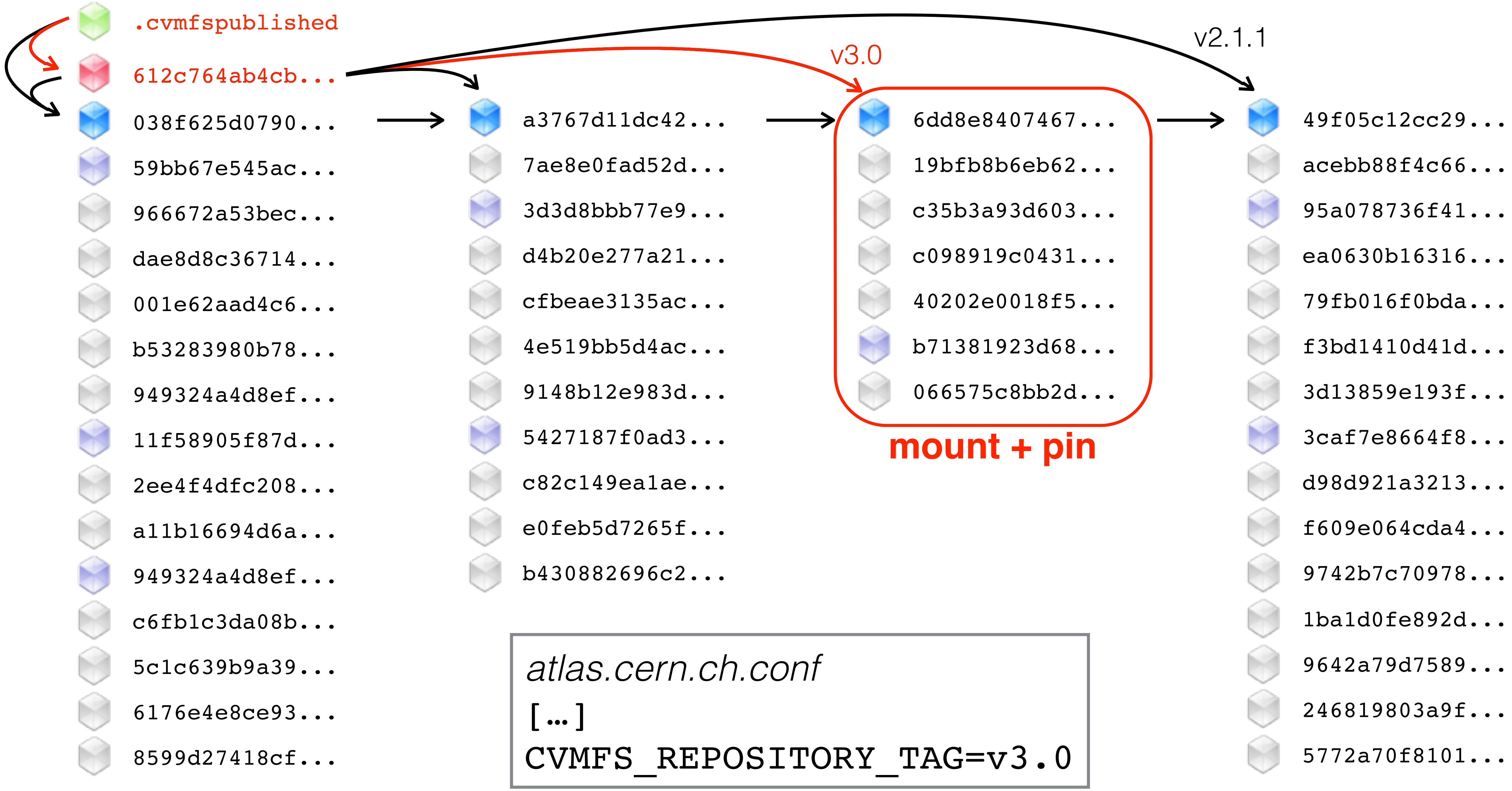
v2.1.1

49f05c12cc29...
acebb88f4c66...
95a078736f41...
ea0630b16316...
79fb016f0bda...
f3bd1410d41d...
3d13859e193f...
3caf7e8664f8...
d98d921a3213...
f609e064cda4...
9742b7c70978...
1ba1d0fe892d...
9642a79d7589...
246819803a9f...
5772a70f8101...

atlas.cern.ch.conf

```
[…]
CVMFS_REPOSITORY_TAG=v3.0
```

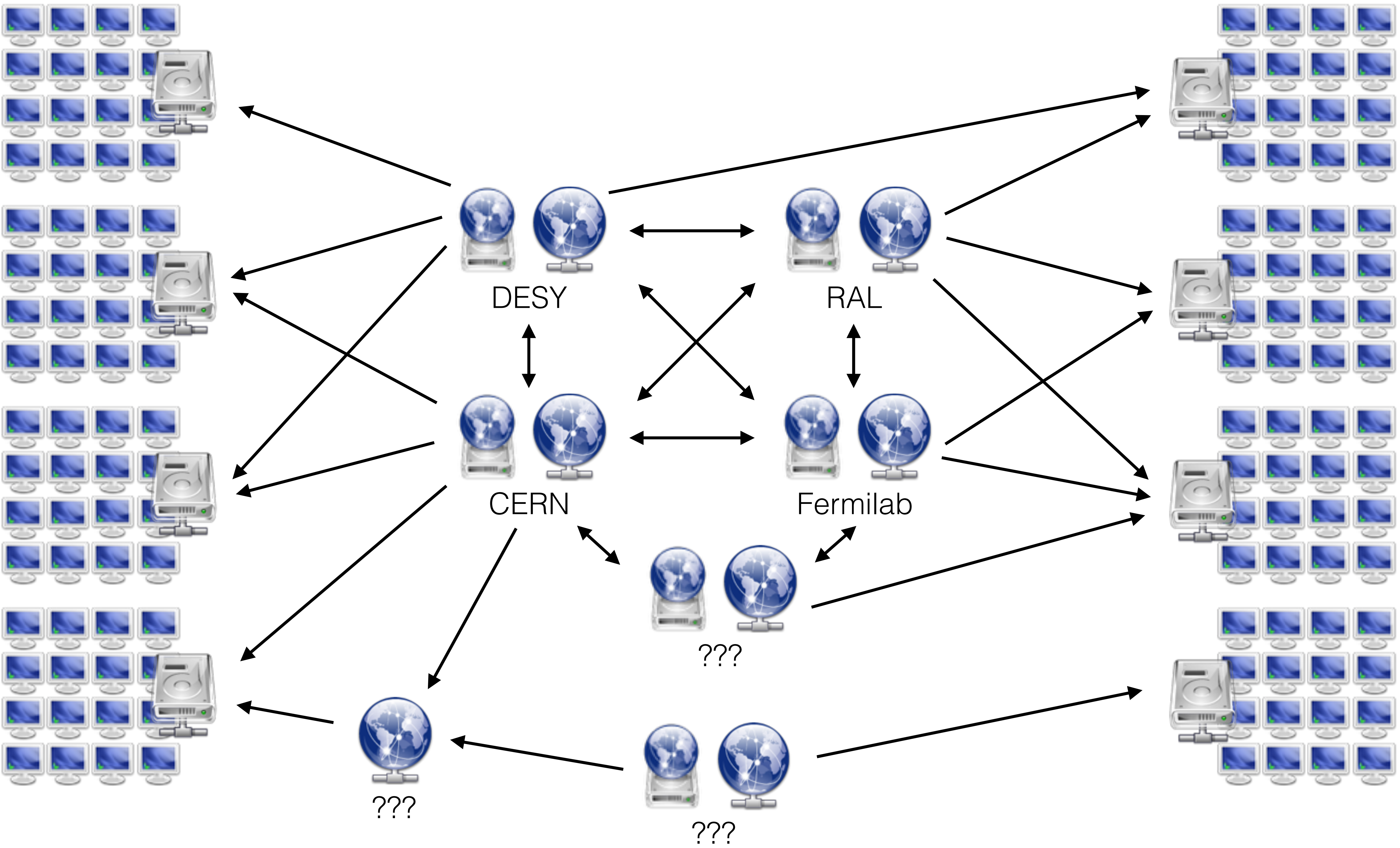
CernVM-FS Snapshot History



atlas.cern.ch.conf

```
[…]
CVMFS_REPOSITORY_TAG=v3.0
```

A Realistic CernVM-FS “Network”



New and Upcoming Features in CernVM-FS

- CernVM-FS on Parrot
  - Using multiple repositories concurrently with Parrot is unstable
  - Improved switching of repositories in *libcvmfs* (CernVM-FS 2.1.20)
  - Adapted Parrot connector is submitted to *cctools* project

- Web API on Stratum 1 servers (CernVM-FS 2.1.20)
  - Automatic Stratum 1 ordering (contribution by Dave Dykstra)
    - Clients send list of configured Stratum 1 URLs to one Stratum 1
    - List is sent back ordered by geographic distance to requester
    - Based on GeoIP database (www.maxmind.com)
  - Basis for push replication of repositories (as requested by ALICE)

New Features in CernVM-FS 2.1.x

Transactional Repository Updates

File System Snapshots

Snapshot History Database

Repository Rollbacks on Stratum 0

Parallel File Processing

Chunking of Large Files

Alternative Storage Backends

Multiple Repositories on one Installation Box

Aggregated Repository Statistics

Abandon ‘Shadow Directory’ on Installation Box

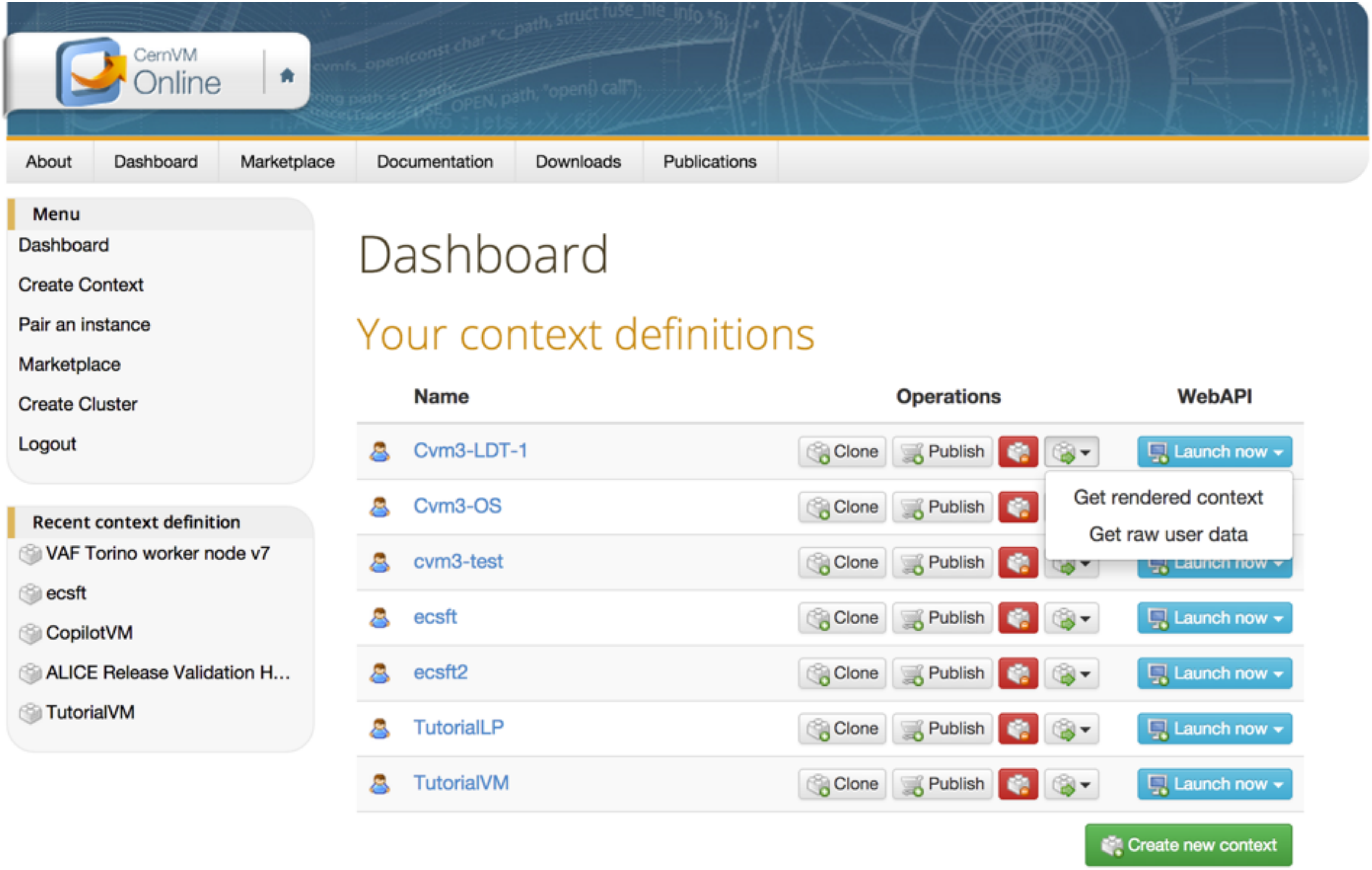
[…]

CernVM 3 - Details

- First production release (v3.1) on January 31st '14
- Current version 3.3 on May 27th '14
  - Based on SL 6.5, µCernVM 1.18 (kernel 3.10.44-74)
  - Contextualisation: amiconfig, cloud-init
    - Web portal (CernVM-Online[1]) with possibility to generate the user data file
  - Extras: HTCondor, ganglia, puppet, squid, xrootd, cloud clients
  - Integration with cloud-scheduler
  - cvm2ova tool to create custom OVA images
    - E.g. http://cernvm.cern.ch/releases/ROOT6.ova to run ROOT 6 on unsupported platforms

[1] http://cernvm.cern.ch/portal/online

CernVM Online



Dashboard

Your context definitions

Name	Operations	WebAPI
Cvm3-LDT-1	Clone Publish	Launch now
Cvm3-OS	Clone Publish	
cvm3-test	Clone Publish	Launch now
ecsft	Clone Publish	Launch now
ecsft2	Clone Publish	Launch now
TutorialLP	Clone Publish	Launch now
TutorialVM	Clone Publish	Launch now

Abstract

The CernVM-File System (CVMFS) is a snapshotting read-only file system based on HTTP to deliver centrally installed software to grids and clouds in a fast, scalable and reliable way. It is extensively used in the WLCG and gains adoption in various other grid infrastructures.

Contents of a CernVM-File System are centrally maintained on a so called release manager machine (CVMFS Server) constituting the single read/write location of the system. By separating file system meta data from actual file contents it creates a CernVM-FS repository that can be distributed as static HTTP content. Clients usually access these CernVM-FS repositories through a FUSE module that downloads individual files on-demand and caches them locally.

This talk is an introduction to CernVM-FS, focussing on the administrative perspective of both the CernVM-FS server and client. We will look at best practices of CernVM-FS client deployments in an existing computing centre and briefly overview the global software distribution setup utilised by the four main LHC experiments. Furthermore we will sketch how CernVM-FS internally handles repository contents and which assumptions on both file system content and distribution setup are made for scalability, performance and reliability.

As an important use case of CernVM-FS, we will take a glance at how CernVM 3 distributes a whole operating system on-demand, which simplifies and speeds up the deployment of virtual machines on cloud resources.

CERN