# Software Distribution with CernVM-FS

Jakob Blomer CERN, EP-SFT

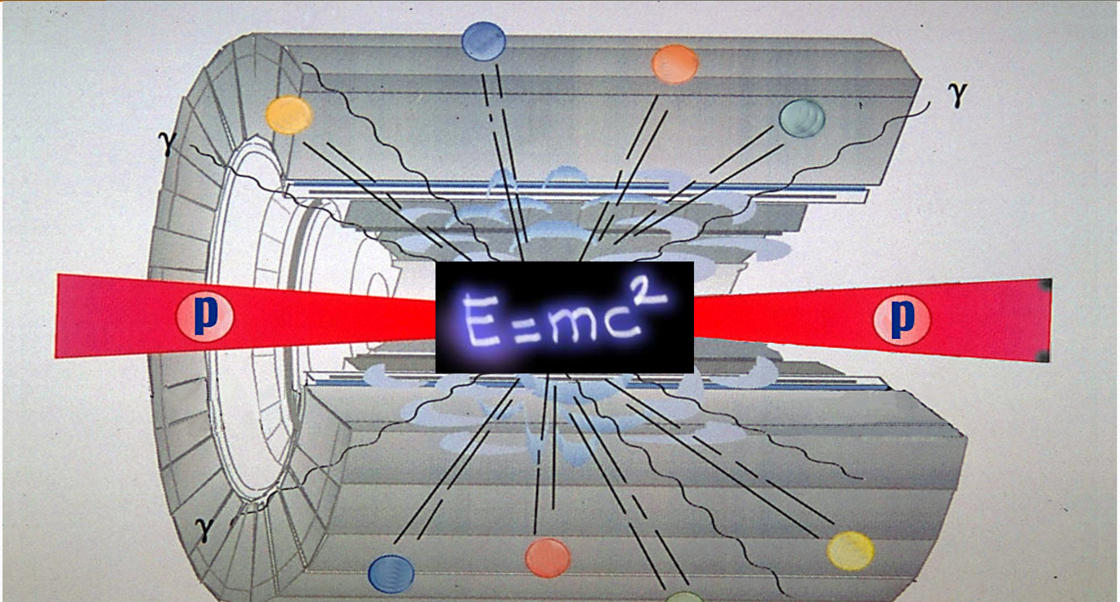
✉ jblomer@cern.ch

https://github.com/cvmfs/cvmfs

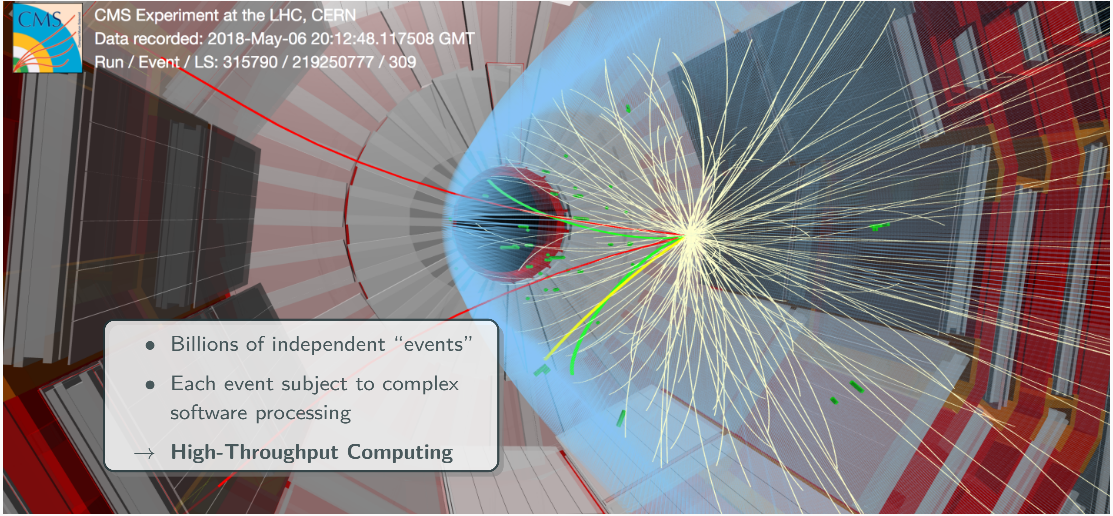
2019-05-17

# 1. Accelerate. . .

# 2. Collide. . .



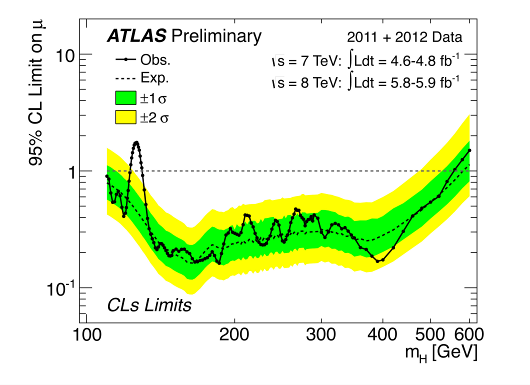
# 3. Measure. . .

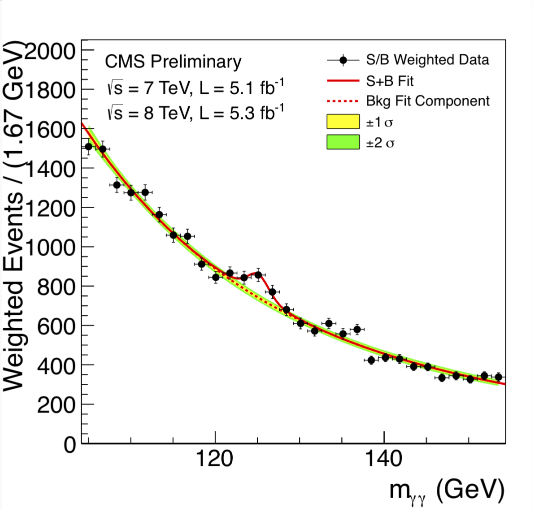

CMS
CMS Experiment at the LHC, CERN
Data recorded: 2018-May-06 20:12:48.117508 GMT
Run / Event / LS: 315790 / 219250777 / 309

- Billions of independent “events”
- Each event subject to complex software processing

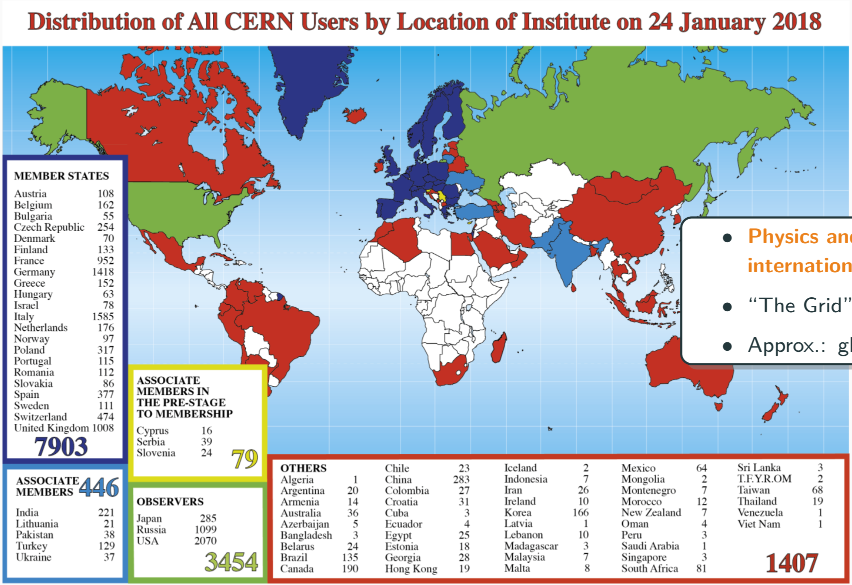
→ **High-Throughput Computing**

# 4. Analyze!


ATLAS Preliminary
2011 + 2012 Data
√s = 7 TeV: ∫Ldt = 4.6-4.8 fb⁻¹
√s = 8 TeV: ∫Ldt = 5.8-5.9 fb⁻¹
Obs.
Exp.
±1 σ
±2 σ
CLs Limits
95% CL Limit on μ
10
1
10⁻¹
100
200
300
400
500
600
m_H [GeV]


CMS Preliminary
√s = 7 TeV, L = 5.1 fb⁻¹
√s = 8 TeV, L = 5.3 fb⁻¹
S/B Weighted Data
S+B Fit
Bkg Fit Component
±1 σ
±2 σ
Weighted Events / (1.67 GeV)
2000
1800
1600
1400
1200
1000
800
600
400
200
0
120
140
m_γγ (GeV)

# Computing, Federated



Distribution of All CERN Users by Location of Institute on 24 January 2018

| MEMBER STATES | |
|---|---|
| Austria | 108 |
| Belgium | 162 |
| Bulgaria | 55 |
| Czech Republic | 254 |
| Denmark | 70 |
| Finland | 133 |
| France | 952 |
| Germany | 1418 |
| Greece | 152 |
| Hungary | 63 |
| Israel | 78 |
| Italy | 1585 |
| Netherlands | 176 |
| Norway | 97 |
| Poland | 317 |
| Portugal | 115 |
| Romania | 112 |
| Slovakia | 86 |
| Spain | 377 |
| Sweden | 111 |
| Switzerland | 474 |
| United Kingdom | 1008 |
| | 7903 |

| ASSOCIATE MEMBERS IN THE PRE-STAGE TO MEMBERSHIP | |
|---|---|
| Cyprus | 16 |
| Serbia | 39 |
| Slovenia | 24 |
| | 79 |

| ASSOCIATE MEMBERS | 446 |
|---|---|
| India | 221 |
| Lithuania | 21 |
| Pakistan | 38 |
| Turkey | 129 |
| Ukraine | 37 |

| OBSERVERS | |
|---|---|
| Japan | 285 |
| Russia | 1099 |
| USA | 2070 |
| | 3454 |

| OTHERS | |
|---|---|
| Algeria | 1 |
| Argentina | 20 |
| Armenia | 14 |
| Australia | 36 |
| Azerbaijan | 5 |
| Bangladesh | 3 |
| Belarus | 24 |
| Brazil | 135 |
| Canada | 190 |
| Chile | 23 |
| China | 283 |
| Colombia | 27 |
| Croatia | 31 |
| Cuba | 3 |
| Ecuador | 4 |
| Egypt | 25 |
| Estonia | 18 |
| Georgia | 28 |
| Hong Kong | 19 |
| Iceland | 2 |
| Indonesia | 7 |
| Iran | 26 |
| Ireland | 10 |
| Korea | 166 |
| Latvia | 1 |
| Lebanon | 10 |
| Madagascar | 3 |
| Malaysia | 7 |
| Malta | 8 |
| Mexico | 64 |
| Mongolia | 2 |
| Montenegro | 7 |
| Morocco | 12 |
| New Zealand | 7 |
| Oman | 4 |
| Peru | 3 |
| Saudi Arabia | 1 |
| Singapore | 3 |
| South Africa | 81 |
| Sri Lanka | 3 |
| T.F.Y.R.O.M | 2 |
| Taiwan | 68 |
| Thailand | 19 |
| Venezuela | 1 |
| Viet Nam | 1 |
| | 1407 |

- **Physics and computing: international collaborations**
- “The Grid”: ∼160 data centers
- Approx.: global batch system

# Software Delivery by a File System

❶ Provide uniform, consistent, and versioned POSIX file system access to `/cvmfs`

```
$ ls /cvmfs/cms.cern.ch
slc7_amd64_gcc700  slc7_ppc64le_gcc530  slc7_aarch64_gcc700  slc6_mic_gcc481
...
```

on **grids**, **clouds**, **supercomputers** and **end user laptops**

*read*

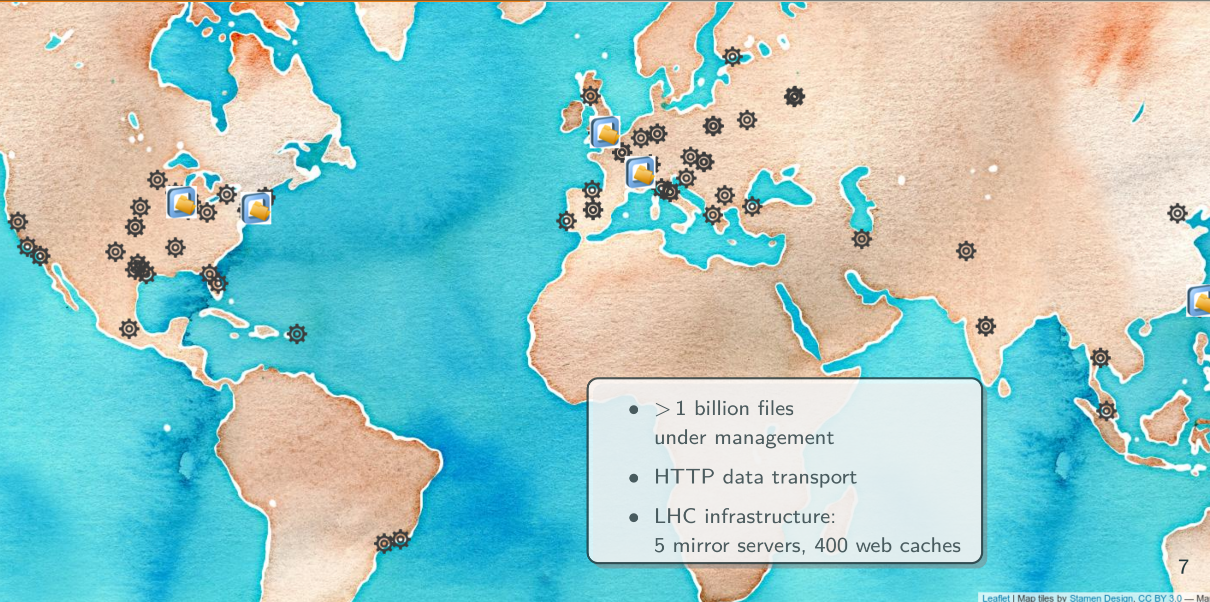
---

*publish*

❷ Populate and propagate new and updated content

- A few “software librarians” can publish into /cvmfs
- Transactional writes as in `git commit/push`

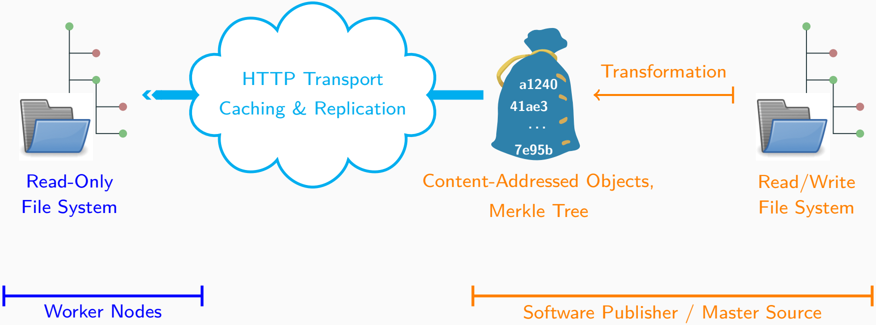
# Scale of Deployment



- > 1 billion files under management
- HTTP data transport
- LHC infrastructure: 5 mirror servers, 400 web caches

Leaflet | Map tiles by Stamen Design, CC BY 3.0 — Ma

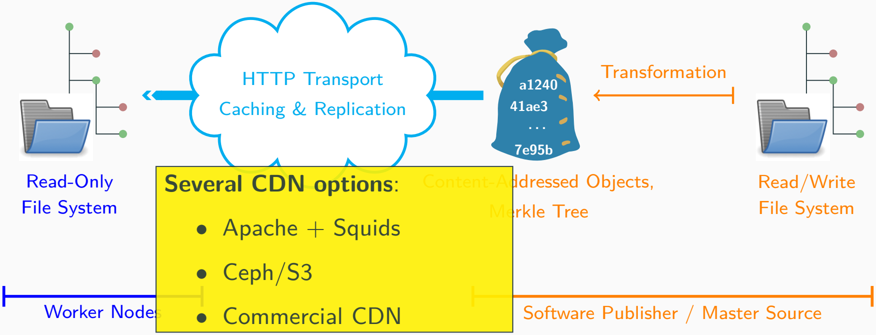
# Key Design Choices



- Reading and writing treated asymmetrically
- Immutable objects, stateless services

- HTTP transport + caching
- Consistency over availability

# Key Design Choices



HTTP Transport
Caching & Replication

Read-Only
File System

Content-Addressed Objects,
Merkle Tree

Transformation

Read/Write
File System

Worker Nodes

Software Publisher / Master Source

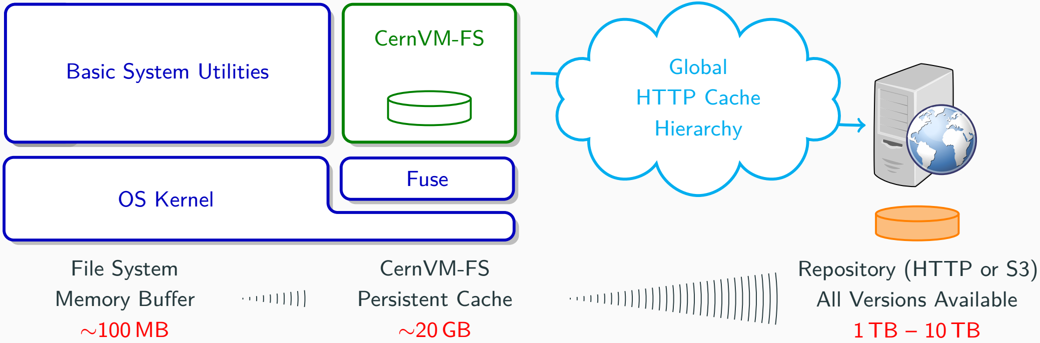
**Several CDN options:**

- Apache + Squids
- Ceph/S3
- Commercial CDN

- Reading and writing treated asymmetrically
- Immutable objects, stateless services

- HTTP transport + caching
- Consistency over availability

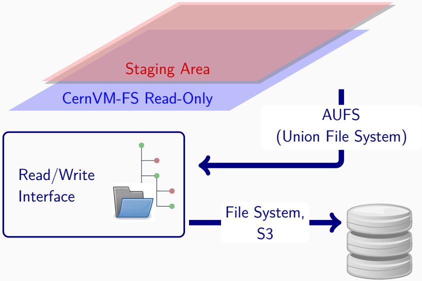
# Reading



- Fuse based, independent mount points, e. g. /cvmfs/atlas.cern.ch
- High cache effiency because entire cluster likely to use same software

# Writing

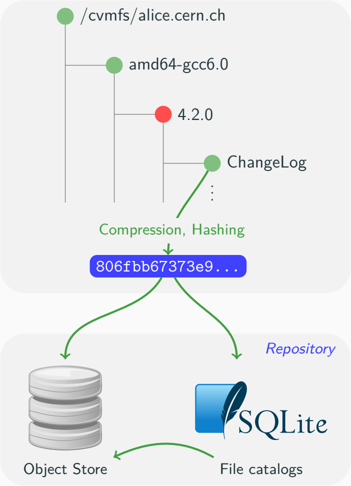
- Kernel-level union file system: AUFS, OverlayFS



**Publishing new content**

```
[ ~ ]# cvmfs_server transaction containers.cern.ch
[ ~ ]# cd /cvmfs/containers.cern.ch && tar xvf ubuntu1610.tar.gz
[ ~ ]# cvmfs_server publish containers.cern.ch
```

# Use of Content-Addressable Storage



## Object Store

- Compressed files and chunks
- De-duplicated

## File Catalog

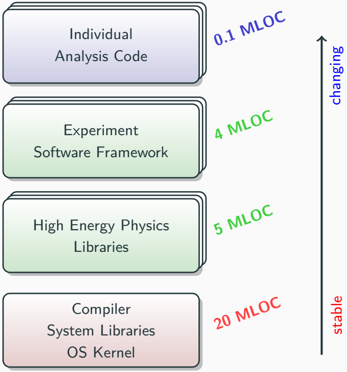
- Directory structure, symlinks
- Content hashes of regular files
- Digitally signed
  ⇒ integrity, authenticity
- Time to live
- Partitioned / Merkle hashes
  (possibility of sub catalogs)

⇒ Immutable files, trivial to check for corruption, versioning

# Why a file system?

# HEP Software Challenge

```
$ cmsRun DiPhoton_Analysis.py
```



Individual Analysis Code — 0.1 MLOC

Experiment Software Framework — 4 MLOC

High Energy Physics Libraries — 5 MLOC

Compiler System Libraries OS Kernel — 20 MLOC

changing

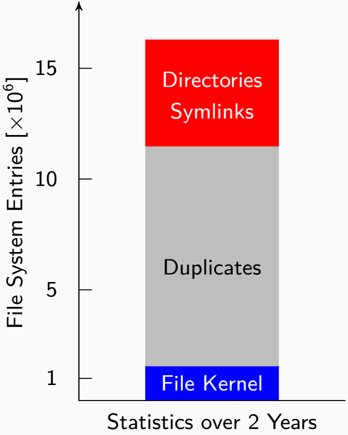
stable

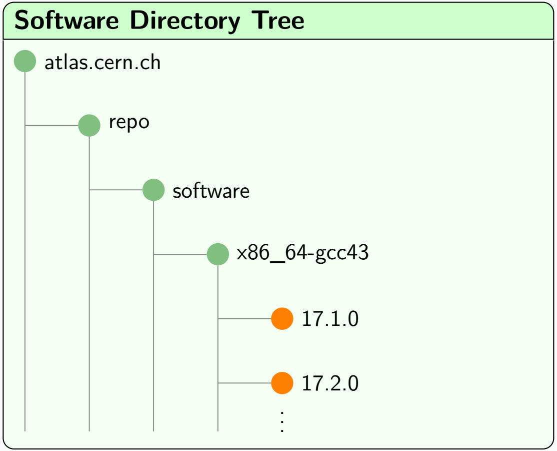
## Key Figures

- Hundreds of (novice) developers
- Hundred million binaries
- 1 TB / day of nightly builds
- ∼100 000 machines world-wide
- Daily production releases, remain available “eternally”

→ **too much for a packaging approach**

# Actual new content

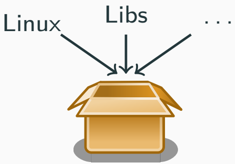




Between consecutive software versions: only ∼15 % new files

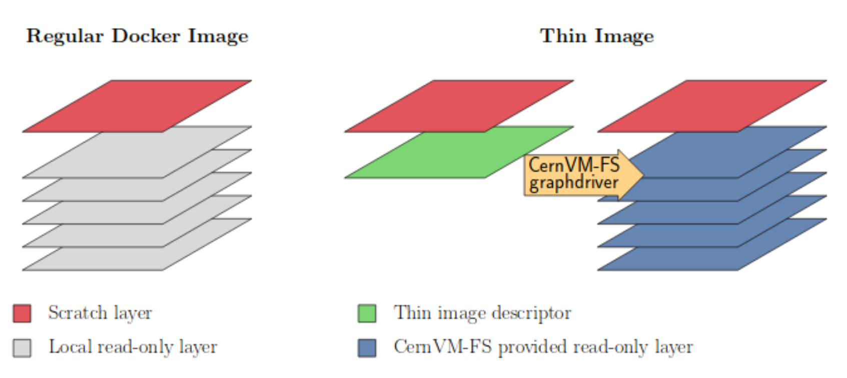
At runtime only tiny fraction of files actually accessed

# Container Amplification



- Containers are easier to create than to role-out at scale
- Ideally: containers for isolation and a software file system for distribution

# Custom Graph Driver / Snapshotter Approach



Ideally: native support for (some) unpacked layers on a read-only file sytem