



StashCache benchmarking on the IGWN infrastructure

a programmatic approach

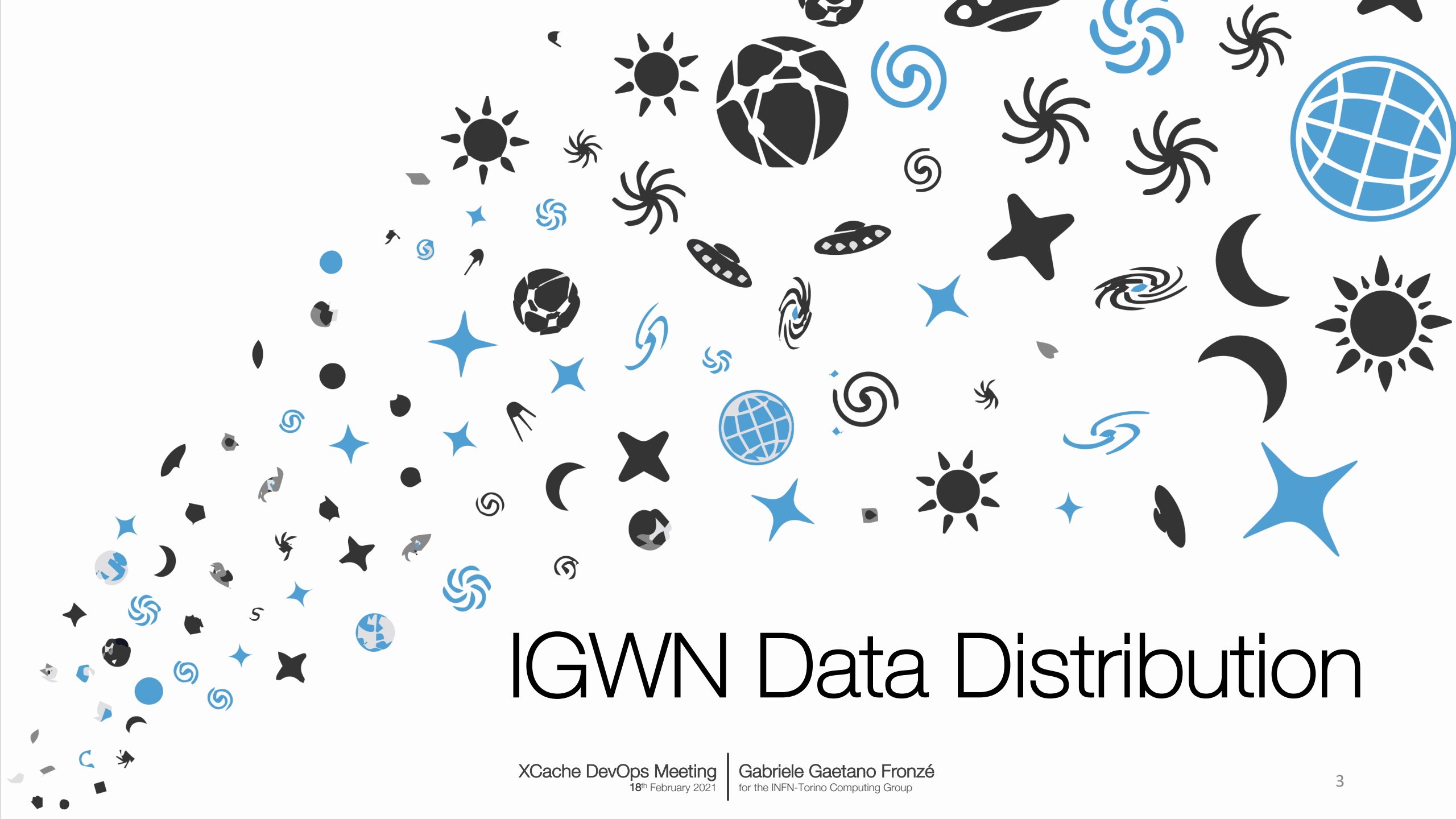
XCACHE DevOps Meeting
18th February 2021



Gabriele Gaetano Fronzé
for the INFN-Torino Computing Group

Summary

- IGWN Data distribution
- IGWN Data checker: a benchmarking tool



IGWN Data Distribution

Base bricks

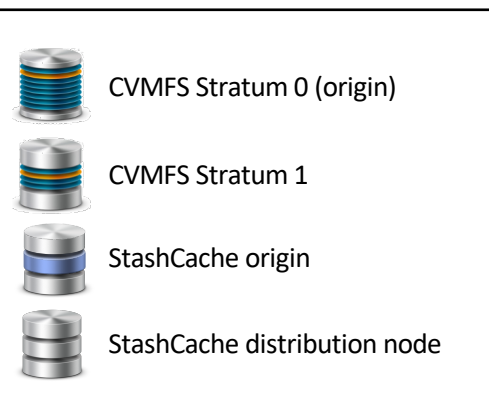
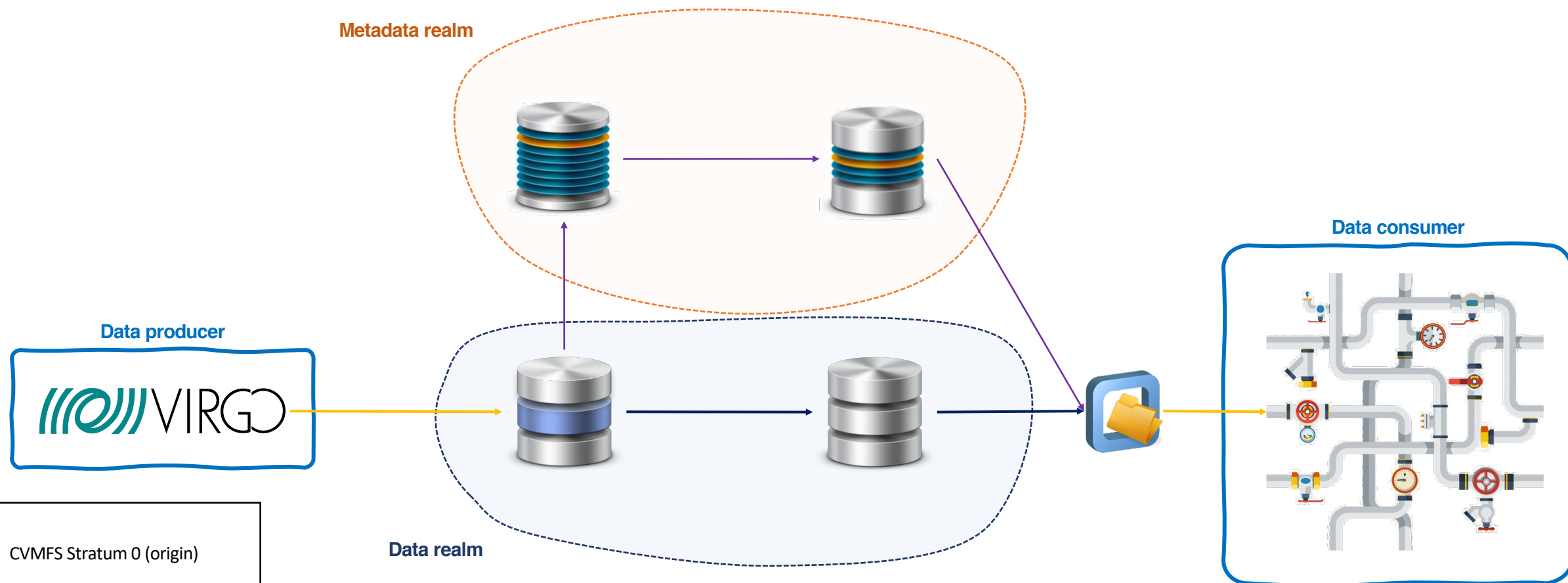


CVMFS = Data catalog

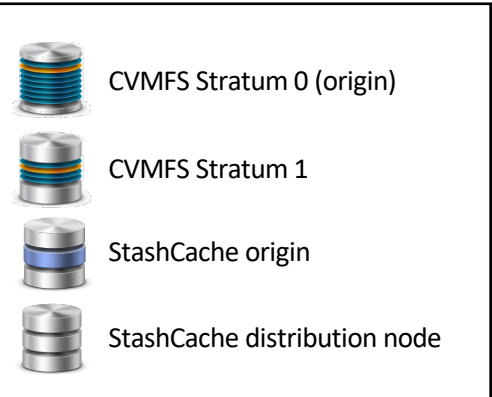
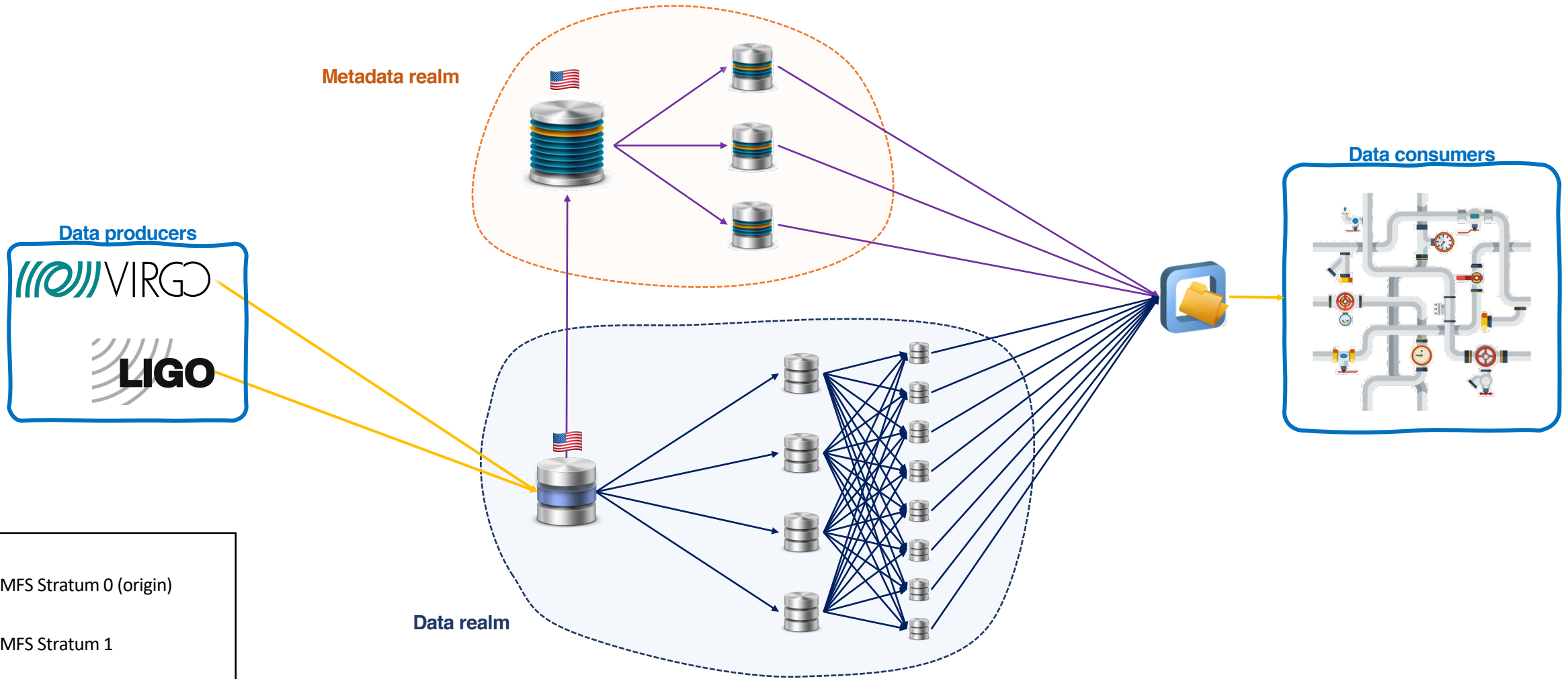


StashCache = Data storage

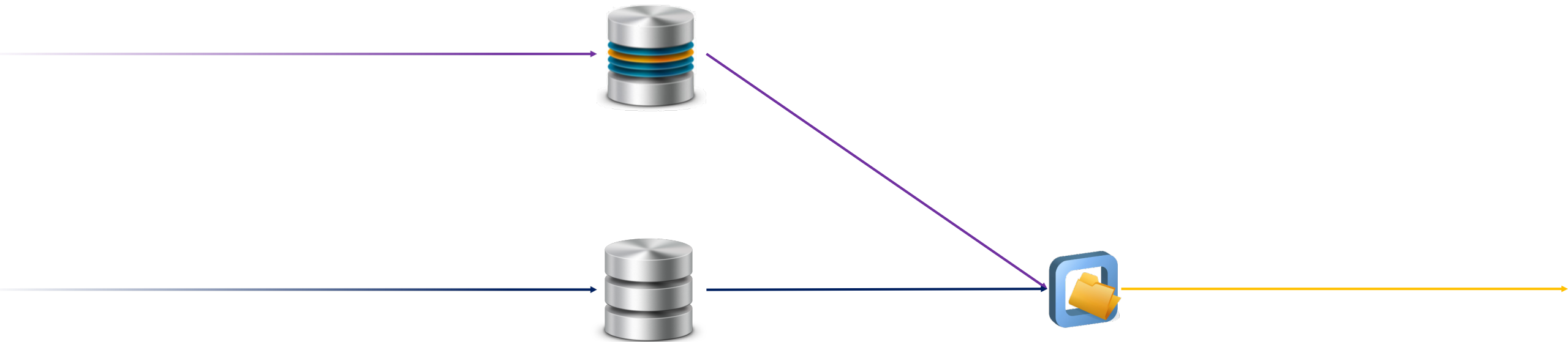
Base framework



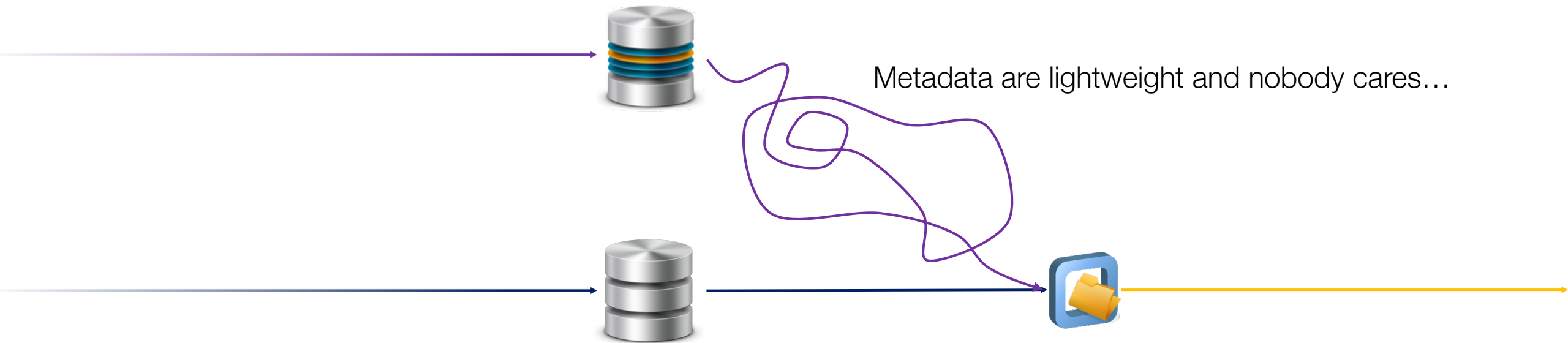
Complete framework



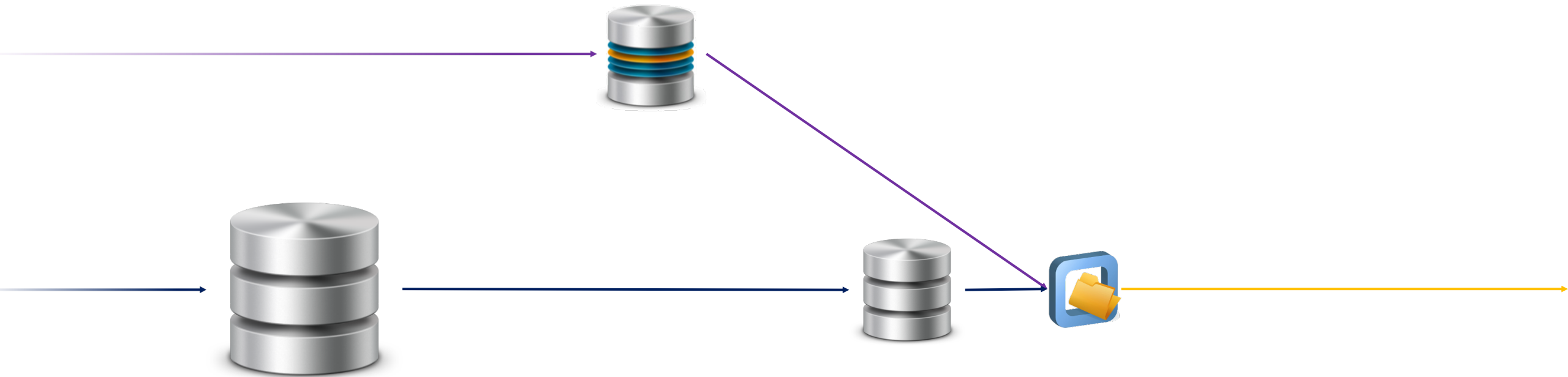
Last mile matters



Last mile matters

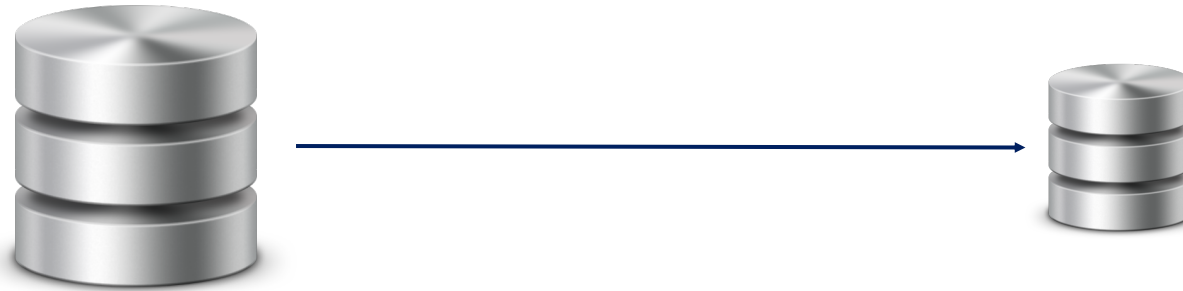


Last mile matters



Data are heavy! Keep it as close as possible!

Last mile matters



Computing centers
installed or are installing
local caches to improve data access



IGWN Data Checker

A data benchmarking tool

A tool for “last mile” measurements

A simple tool to:

- measure data access times
- measure file corruption probability
- asses local caches status

from the IGWN computing facilities

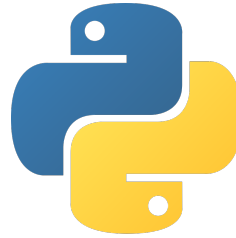


GitHub

<https://github.com/gabrielefronze/igwn-data-checker.git>

A tool for “last mile” measurements

Written in:



Generates:



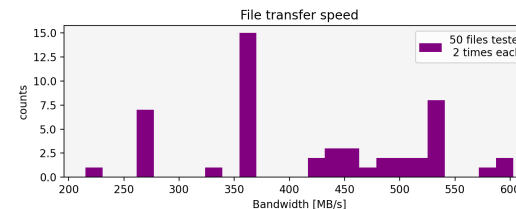
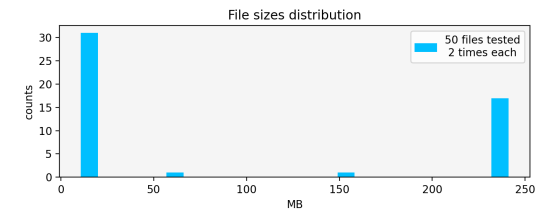
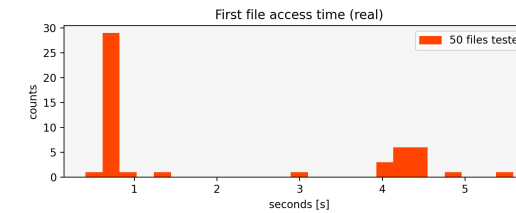
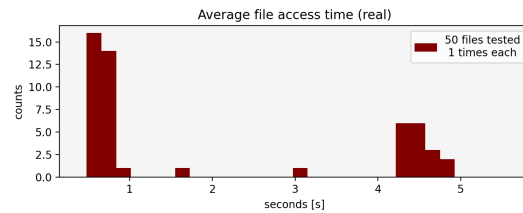
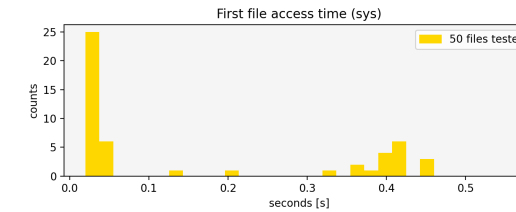
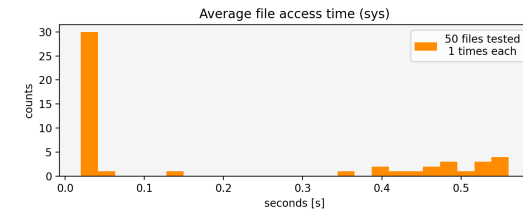
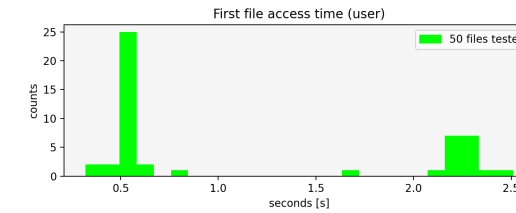
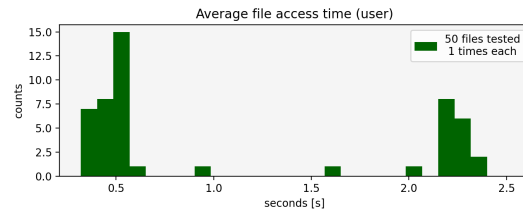
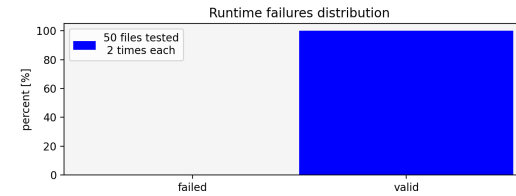
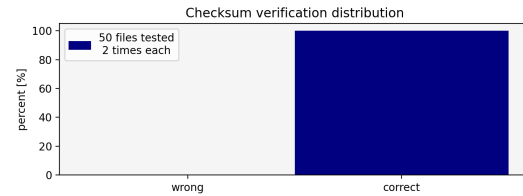
Runs on:



Example output

Checksum verification rate	Runtime failures rate
Average access user time	First access user time
Average access system time	First access system time
Average access real time	First access real time
File size distribution	Storage bandwidth

Example output



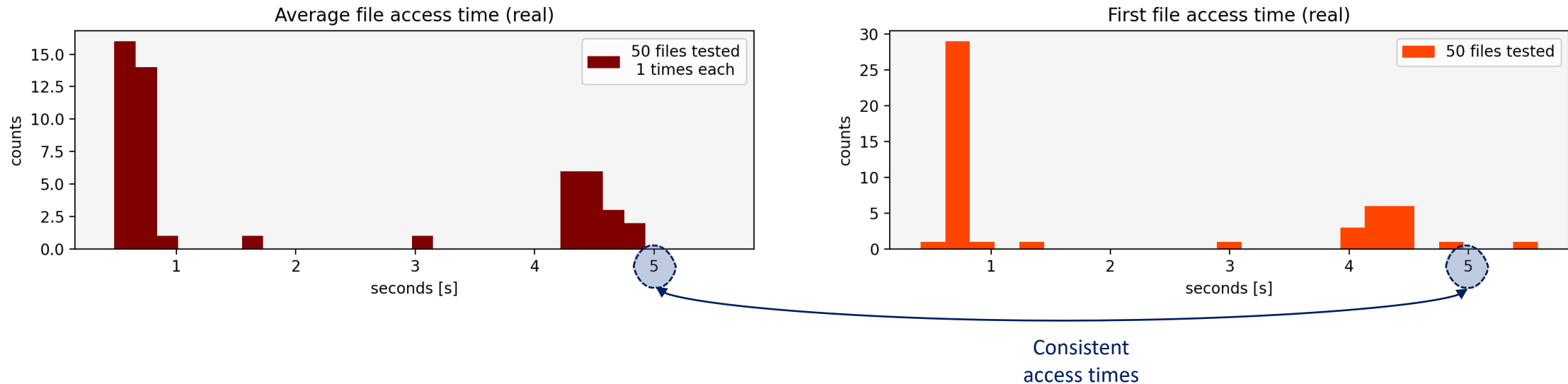
Cache miss-hit detection

When a cache is in place two things can happen when one tries to access a file:



Cache miss-hit detection

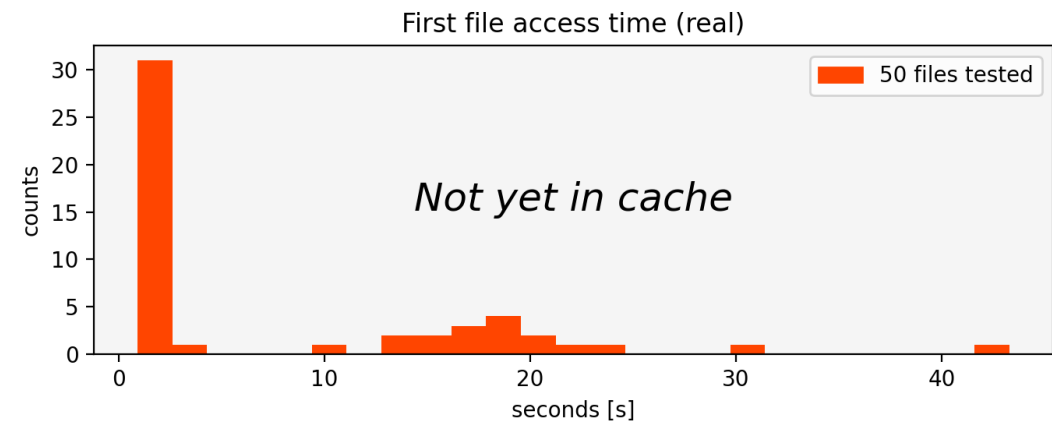
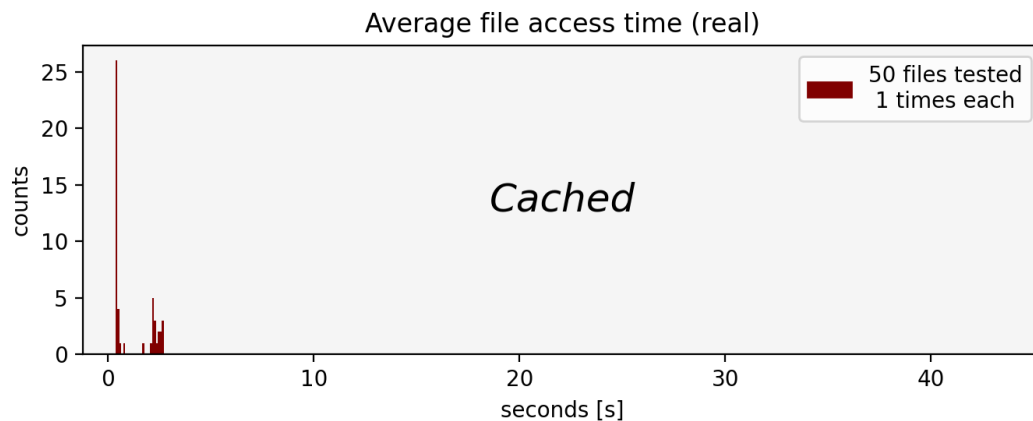
If data is already present in the nearest cache, the first access time should be consistent with the average of the following ones.



Cache miss-hit detection

Otherwise, the first access should result in a cache miss, triggering the cache to populate that entry.

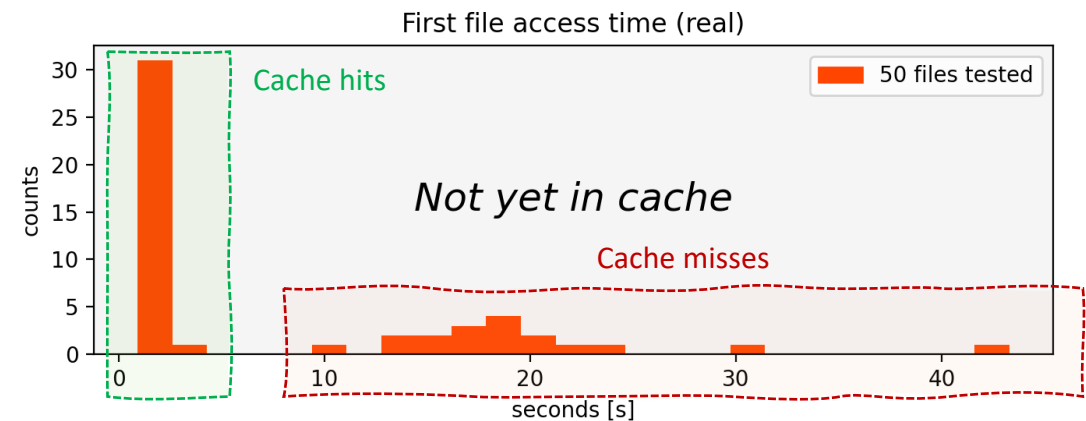
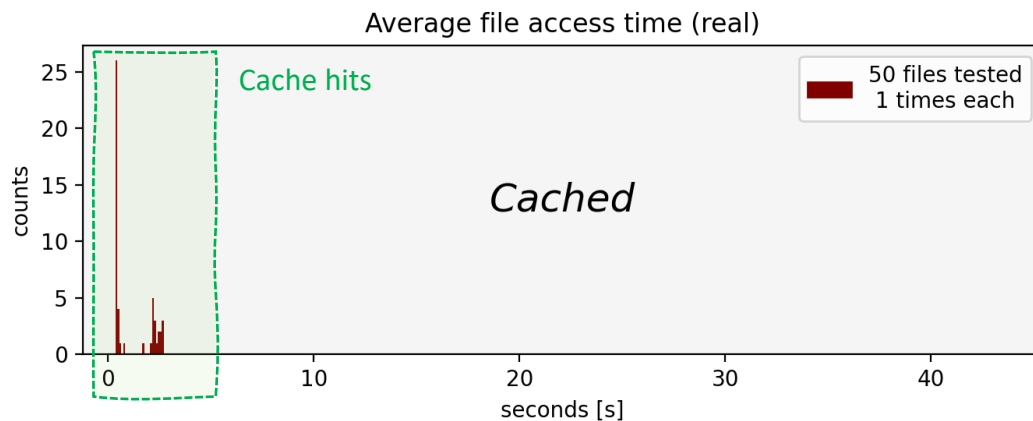
Following accesses will find the file in cache and be much faster!



Cache miss-hit detection

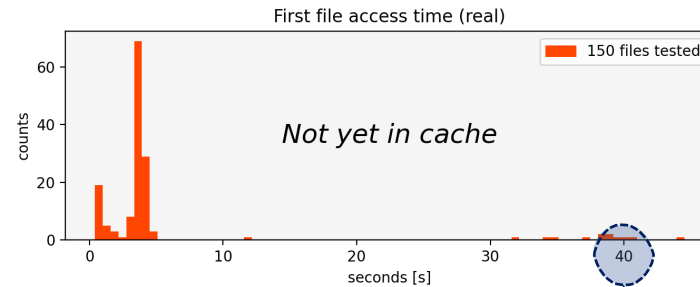
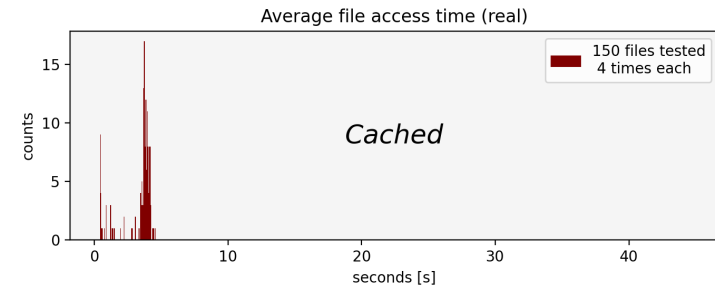
Otherwise, the first access should result in a cache miss, triggering the cache to populate that entry.

Following accesses will find the file in cache and be much faster!



Underperforming cache detection

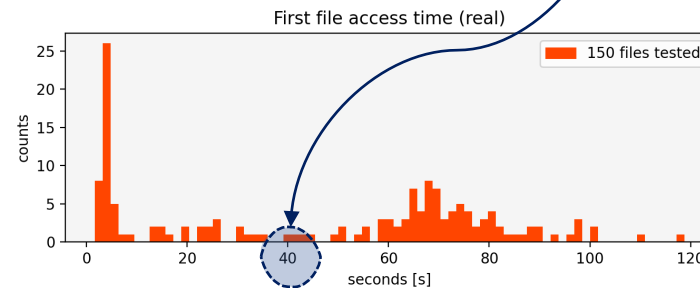
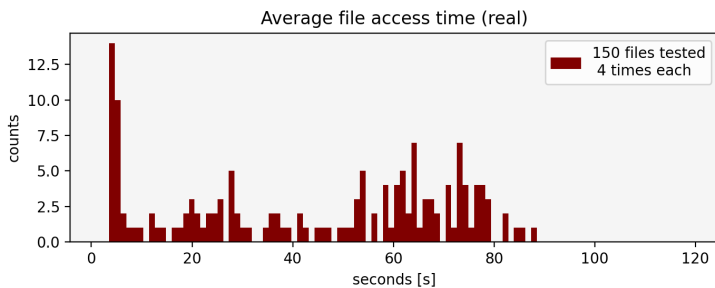
Comparing multiple local caches one can highlight problematic ones.



Both computing centers have a local cache.

The first one behaves as expected, some files are not yet cached but most of them results in a cache hit.

Following accesses are faster.

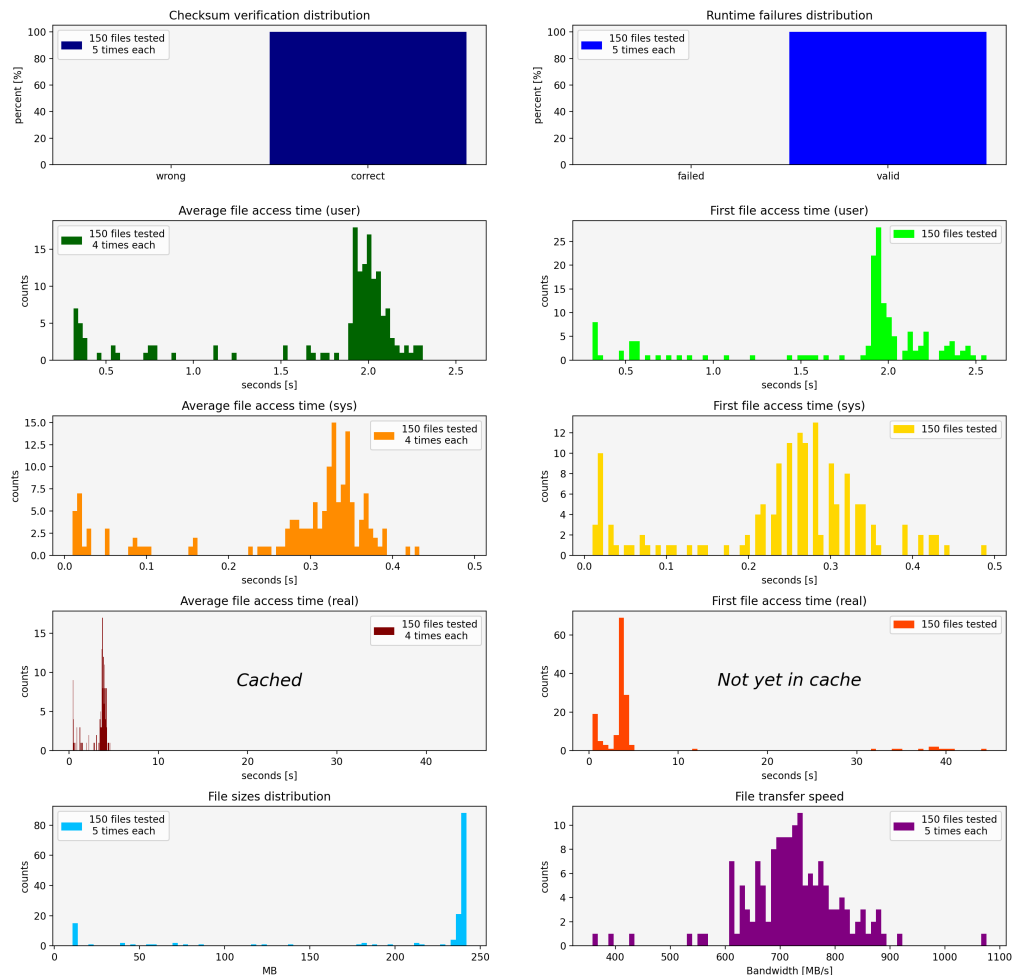


The second computing center shows slow access times (x3 wrt cache misses in the first) and following accesses are almost identical.

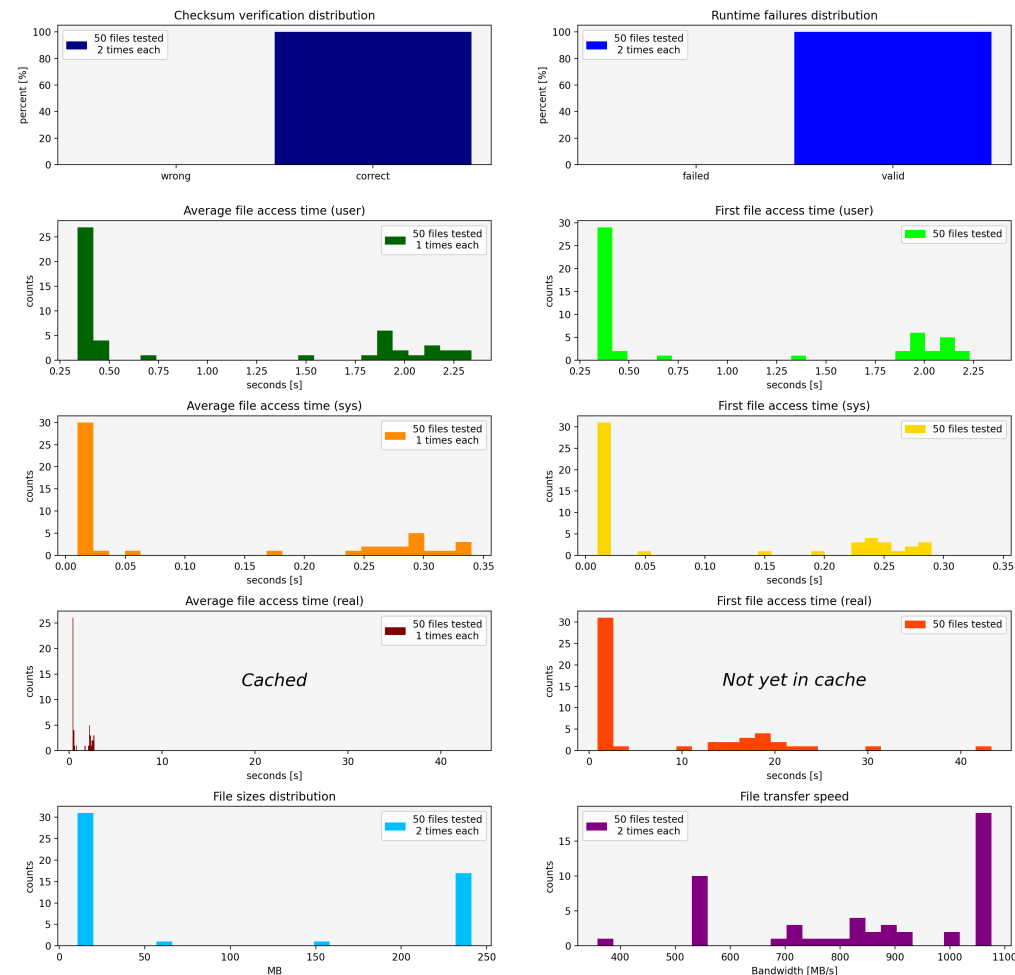
There is something wrong with the second CC cache to be addressed!

Preliminary @ CNAF and PIC

Input file: output-CNAF-LONG.json



Input file: output-PIC.json



Future plans

This tool can be packed in a suite of jobs
to be sent around periodically
to health-check our data distribution network
and
detect failures and other issues

**Thank you for
your attention!**

