

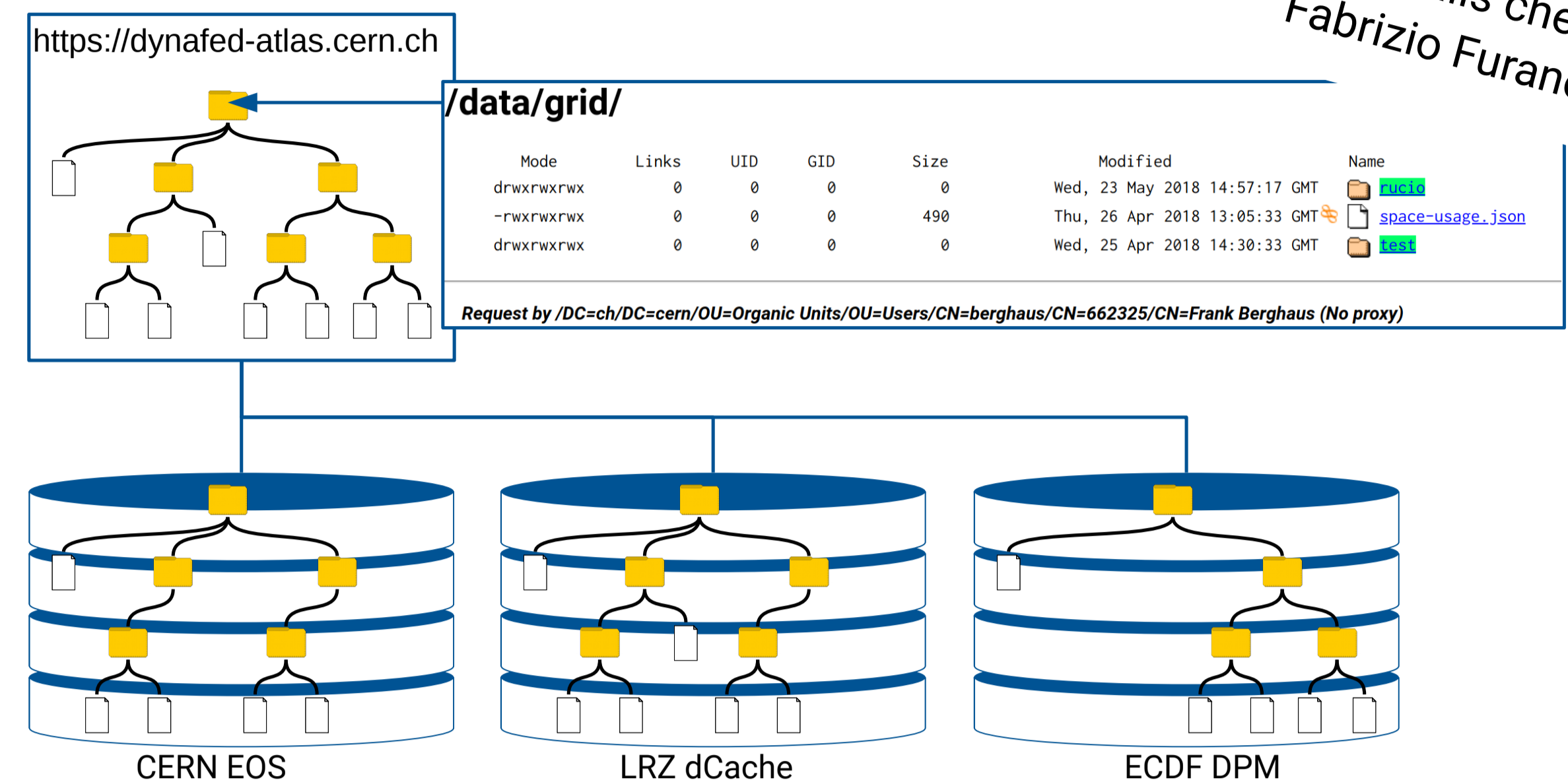
Introduction

Our goal is to run data-intensive applications on globally distributed opportunistic resources that have no local HEP storage. We want to use a data federation, such as Dynafed, to redirect the applications to the optimal¹ storage endpoint to retrieve input data or deposit output data.

(1) optimal in terms of geographic distance, network latency, system load.

Conceptual design

- Create the appearance of a unified namespace from distinct endpoints
- Provide file system catalogue with indexing on standard protocols:
 - HTTP, HTTP+WebDAV, NFS
- Redirect requests by IP address geographical location
 - On read/GET: redirects to nearest endpoint with a copy of requested file
 - On write/PUT: redirects to nearest writable endpoint
 - On delete: sends delete to all writable endpoints with file



For details check out Poster #69
Fabrizio Furano & Oliver Keeble

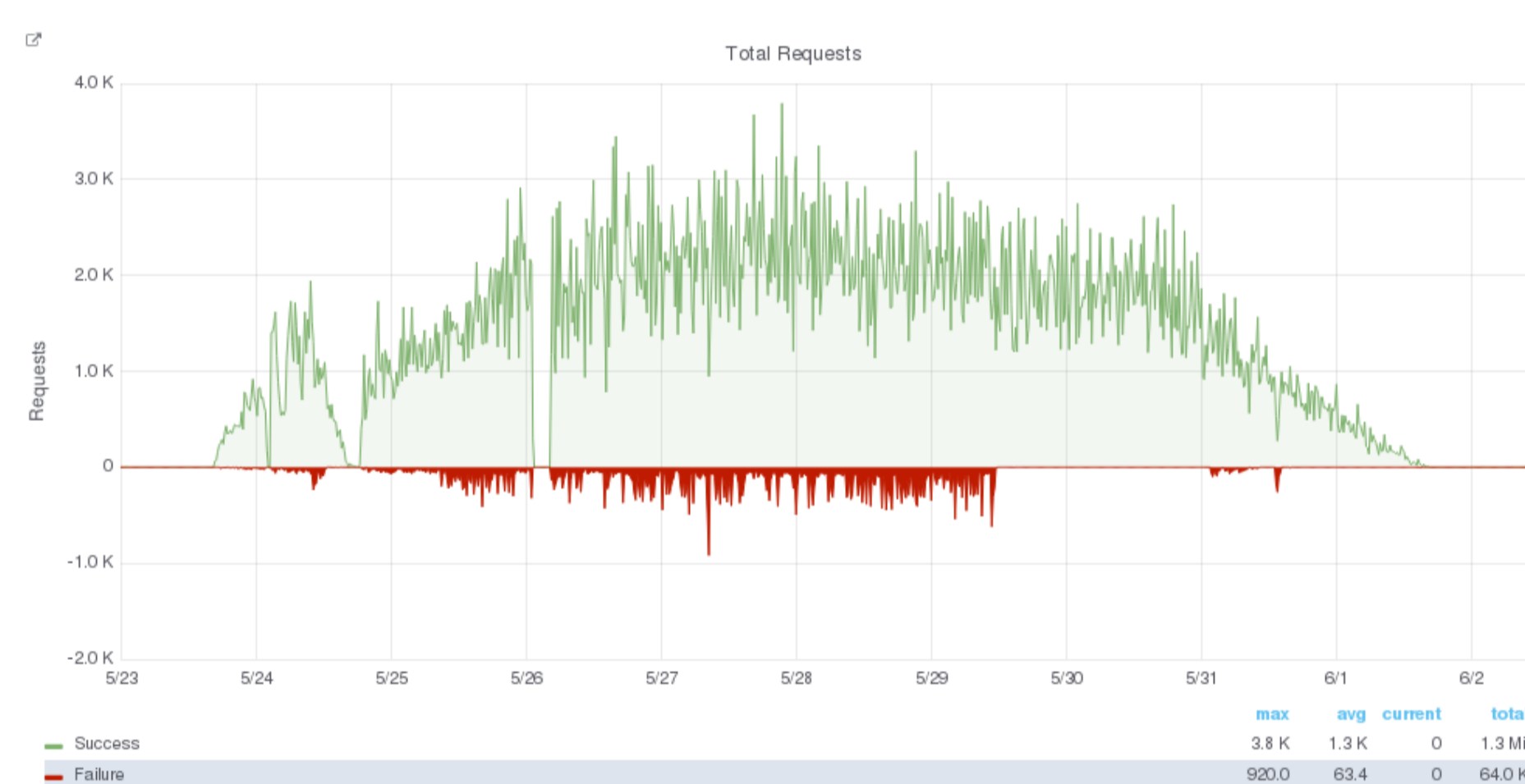
Dynafed and Rucio development

- **MOVE:** Rucio renames files after upload. Dynafed does not implement renaming because object stores do not guarantee this operation.
 - ✓ *Solution:* New Rucio protocol that does not rename files
- **MKCOL:** WebDAV requires directories. Object stores do not have directories.
 - ✓ *Solution:* On MKCOL Dynafed caches directory structure. When redirecting write request to endpoint, Dynafed creates directories as needed.
- **Checksum type:** ATLAS uses ADLER32 [RFC1950] while cloud storage providers only accept MD5 [RFC1321].
 - ✓ *Solution:* Rucio clients calculate MD5 for new files on upload.
- **Checksum mechanism:** Grid storage makes user responsible for file consistency: Want-Digest [RFC3230]. Cloud storage makes the provider responsible: Content-MD5 [RFC1544].
 - *Plan:* Support both schemes. FTS and GFAL support implemented.
 - ✓ *Workaround:* Ignore checksum for testing and development.

Integration with Rucio/ATLAS DDM

- Dynafed is used as a single storage element that supports only HTTP
- Rucio manages transfers to and from Dynafed using FTS
 - Dynafed may only be passive partner in 3rd party transfers
- Storage: CERN CephS3 & EOS, LRZ: dCache, ECDF: DPM
- Grid authentication and authorization using X.509 and VOMS:

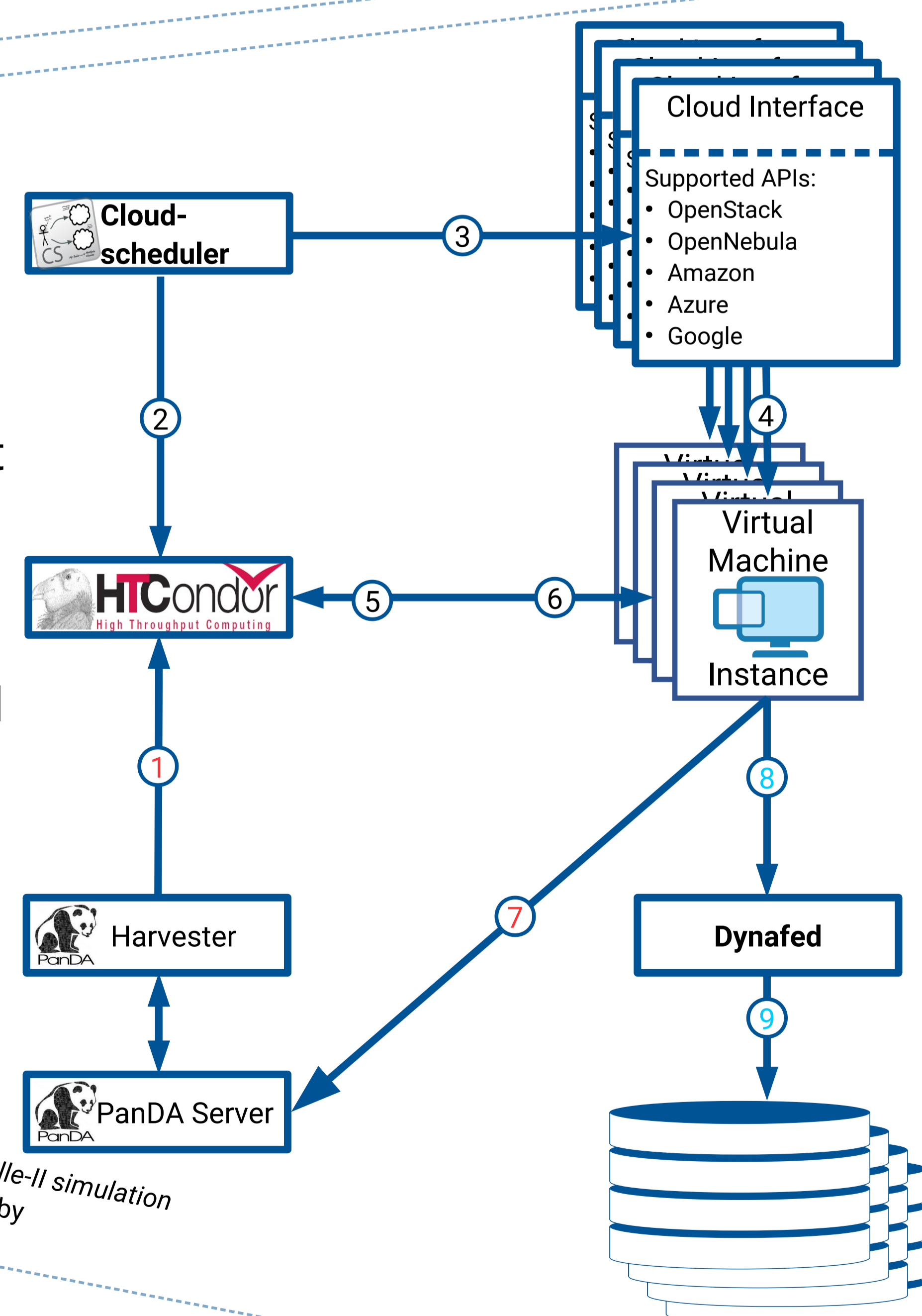
```
glb.allowgroups[]: "/atlas/*" /data rwl
glb.allowgroups[]: "/atlas/Role=production/*" /data rlwd
```



Evolution of the open-source data management system Rucio for LHC Run-3 and beyond ATLAS by Martin Barisits (T4 on Thu @11h)

Integration with PanDA/ ATLAS WMS

- PanDA queue associated with Dynafed Storage Element
- Rucio mover stages data in and out of worker node
 - Queue setup to test streaming
- Workload reads data from local file system
- Cloudscheduler provides resources on distributed cloud



- (1) Harvester submits pilot job from PanDA Queue to HTCondor
- (2) Cloudscheduler polls HTCondor and finds idle jobs
- (3) Cloudscheduler requests instances suitable for the jobs on IaaS Clouds
- (4) Cloud infrastructure provides requested instance
- (5) Instances Advertise to HTCondor
- (6) HTCondor sends pilot job to instance
- (7) Pilot job running on instance pulls workload from PanDA Queue
- (8) Rucio mover stages data from or to Dynafed
- (9) Dynafed redirects Rucio client to nearest storage

Using a dynamic data federation for running Belle-II simulation applications in a distributed cloud environment by Marcus Ebert (T4 Thursday @15h)

Testing Dynafed and HTTP

- Setup: dCache cluster and dynafed in same data centre
- Data: 1.5GB ROOT file from ATLAS tbar sample
- Test: Read p_T of all AntiKt4EMTopoJetAlternative
- Metric: Time for task to complete
- Mode: Streaming
- Conclusions: impact of Dynafed's redirection is small

Early work of Benjamin Rottler (U Freiburg) for qualification project

