



# Using dynafed with clouds in ATLAS and Belle II

Frank Berghaus

on behalf of the University of Victoria and TRIUMF

with lots of help from CERN-IT and ATLAS

# Introduction and motivation

Batch computing in a distributed cloud system

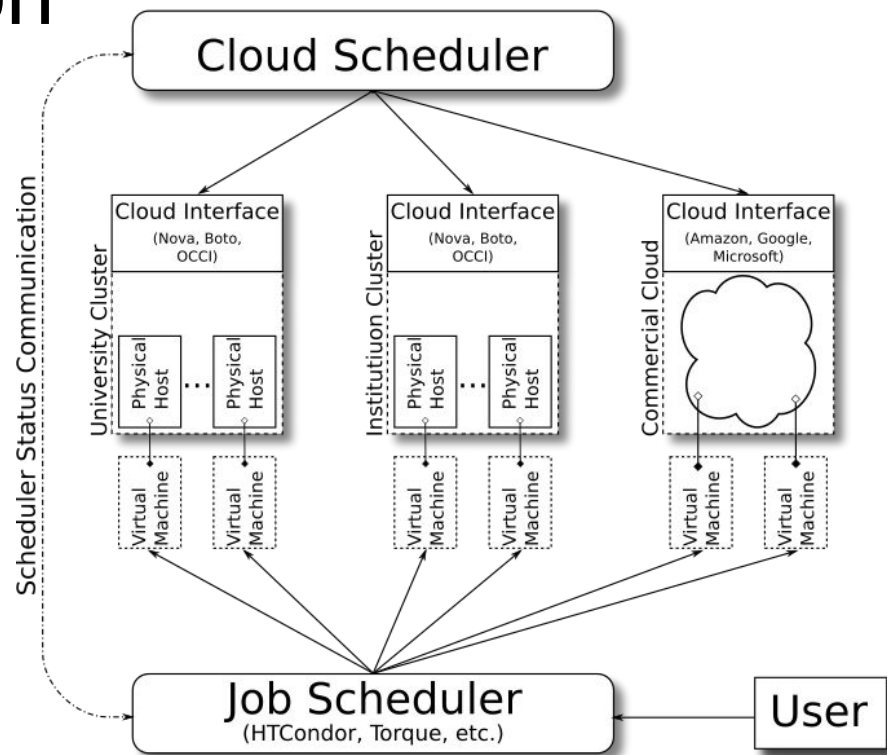
- Cloudscheduler: [github](#) & [arXiv](#)
- Users: Belle-II [DIRAC] & Atlas [PanDA]

Schedulers manage  $O(1k)$  cores in

- UVic -> Canada, USA, Germany, UK
- CERN -> Switzerland, Germany, UK, Austria

**CE:** HTCondor    **SE:** dCache (UVic), EOS (CERN)

Limited by remote access to storage





# Dynafed redirects to nearby storage

Want uniform configuration of all worker instances

Do not know location of worker a-priori

Cloud resources usually do not provide local SE - but often offer object storage

Use Dynafed as SE redirect direct workers to convenient storage



# Reality check

Too many tasks at once:

- Commissioning HTTP/WebDAV in experiments
- Integrating object storage
- Developing and debugging distributed site

Reduced task to **provide access to object stores through Dynafed**

# Technical details

```
[lcgdm-cbuild]
name=LCGDM Continuous Builds
baseurl=http://grid-deployment.web.cern.ch/grid-deployment/dms/
lcgdm/repos/el7/x86_64/
enabled=1
gpgcheck=0
protect=1
priority=20
```

```
[lcgutil-cbuild]
name=DMS Continuous Builds
baseurl=http://dmc-repo.web.cern.ch/dmc-repo/testing/el7/$basea
rch
enabled=1
gpgcheck=0
protect=1
priority=20
```

Version **1.5.X** from the LCGDM and DMS continuous build repositories.

Address dynafed using **davs** over https

Support 3rd party copy with **dynafed-tpc-gfa12**

Support checksum calculation with custom script [[gitlab](#)]

Disable macaroons:

- Interference with S3 signature
- Key can't be shared with grid storages

---

# Dynafed and ATLAS

*Note: test storages, not yet production*



# Dynafed instances for ATLAS

## CERN-EXTENSION\_CLOUDDISK

davs://dynafed-atlas.cern.ch/data/cloud/rucio

## IAAS\_SCRATCHDISK

davs://dynafed02.heprc.uvic.ca:8443/atlas/scratchdisk/rucio

Backed by s3.cern.ch

## CA-TRIUMF-DYNAFED\_DATADISK

davs://atlas-fed-fe1.triumf.ca/dynafed/atlas/atlasdatadisk/rucio

## CA-TRIUMF-DYNAFED\_SCRATCHDISK

davs://atlas-fed-fe1.triumf.ca/dynafed/atlas/atlasscratchdisk/rucio

Backed by atlas-fed-rgw01.triumf.ca

# CERN dynafed jobs

CERN-EXTENSION\_DYNAFED

Hammercloud sends functional tests

Cloudscheduler orchestrates workers at CERN

Pilot stages data in/out using rucio client API

Failure rate ~2.5%

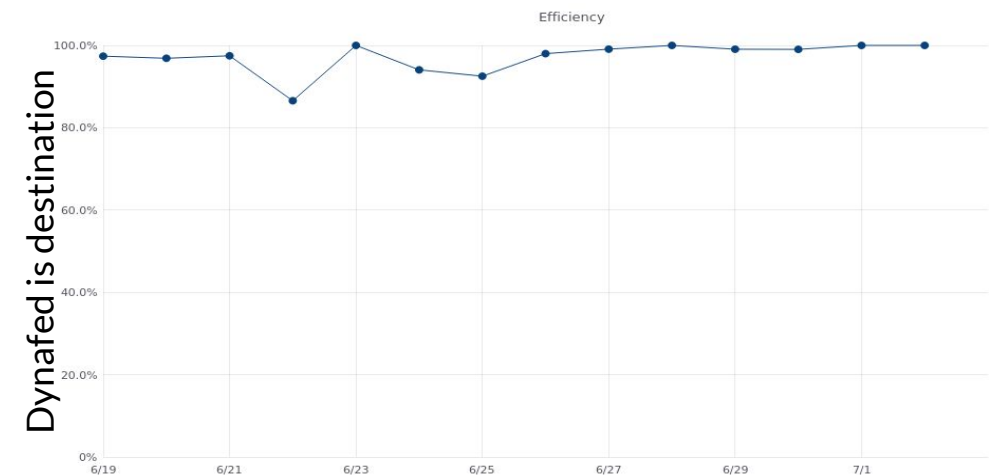




# CERN dynafed transfers

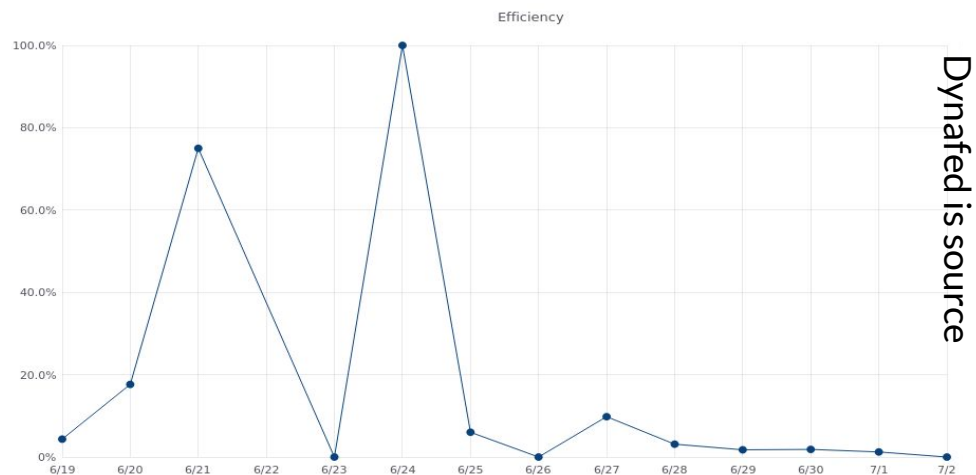
Rucio orchestrates functional transfers

[fts3-pilot.cern.ch](https://fts3-pilot.cern.ch) schedules copy



TPC: enabled but issues with gridsite delegation

**Rucio does not test getting out from Dynafed !**



# TRIUMF dynafed jobs

New ATLAS site for test:

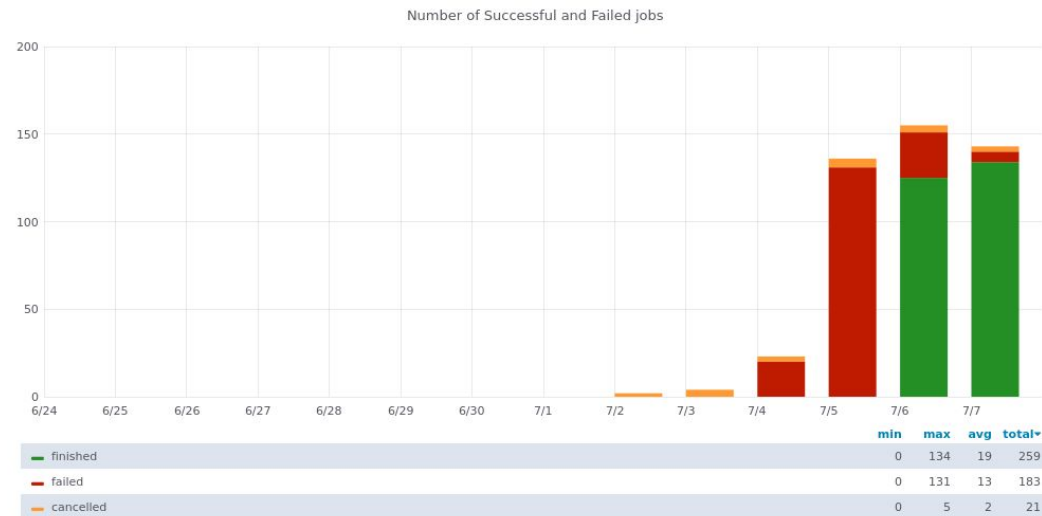
CA-TRIUMF-DYNAFED

Certification as grid site in progress

One worker node at TRIUMF

Receives FT job from Hammercloud and transfers from Rucio

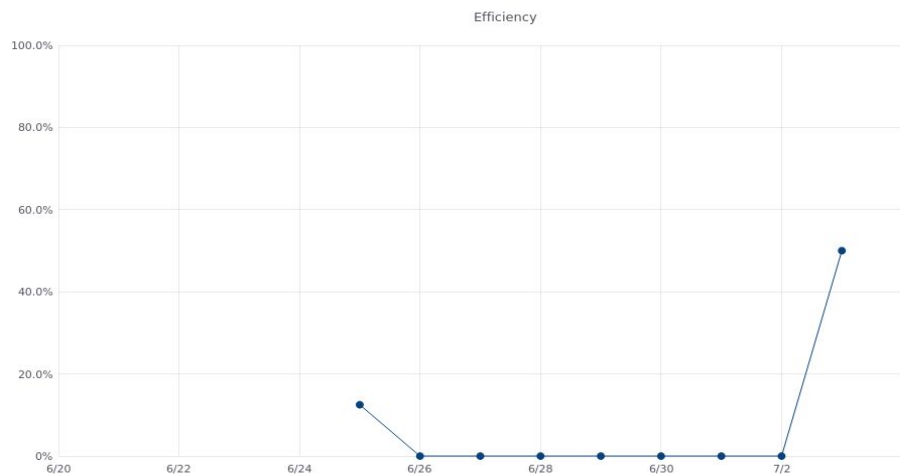
Pilot Rucio mover fails, gfal mover succeeds



# TRIUMF dynafed transfers

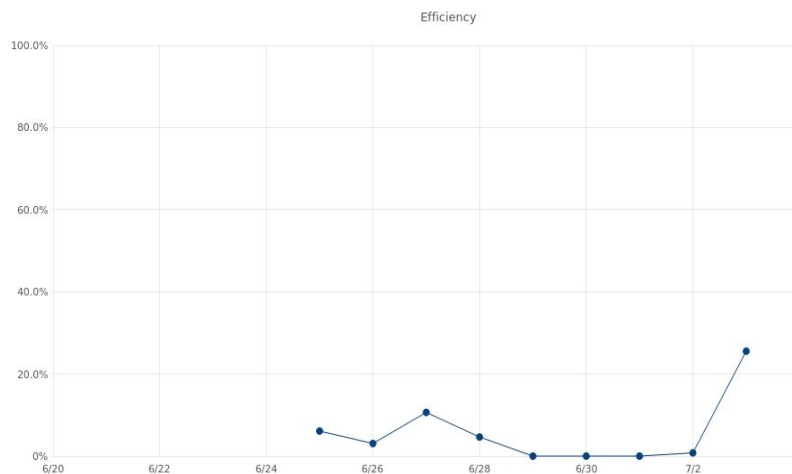
Rucio orchestrates functional transfers

[fts.usatlas.bnl.gov](https://fts.usatlas.bnl.gov) schedules copy



TPC enabled

Initial scaling issues with rados gateways





## As source in ATLAS DDM the last week

Destination \ Source	CA-TRIUMF-DYNAFED_DATADISK	CA-TRIUMF-DYNAFED_SCRATCHDISK	CERN-EXTENSION_CLOUDDISK	IAAS_SCRATCHDISK
CA-TRIUMF-DYNAFED_DATADISK	49%	46%	5%	0%
CA-TRIUMF-DYNAFED_SCRATCHDISK	93%	-	7%	0%
CA-VICTORIA-WESTGRID-T2_DATADISK	-	-	-	100%
CA-VICTORIA-WESTGRID-T2_SCRATCHDISK	-	-	-	100%
CERN-EXTENSION_CLOUDDISK	7%	78%	80%	-
FR-ALPES_SCRATCHDISK_DATA LAKES	-	-	0%	-
ROAMING	67%	-	-	-

## As destination in ATLAS DDM last week

[illegible]



# WebDAV only experience

Feedback from DESY and other dCache sites that ATLAS HTTP test transfers were killing their HTTP/WebDAV servers

This has been fixed in dCache, working setup documented here:

<https://twiki.cern.ch/twiki/bin/view/LCG/DCacheConfig>

---

# Dynafed and Belle-II



# Dynafed instances for Belle-II

## UVic-DATA-SE

- read: <https://dynafed02.heprc.uvic.ca:8443/belle>
- write: [srm://charon01.westgrid.ca/...](srm://charon01.westgrid.ca/)

## UVic-TMP-SE

- read: <https://dynafed02.heprc.uvic.ca:8443/belle>
- write: [srm://charon01.westgrid.ca/...](srm://charon01.westgrid.ca/)

Read from Dynafed via HTTP/WbDAV

Write to **UVic** dCache SRM

Backed by:

Grid SE: **UVic**, KEK, BNL, DESY, GridKA, Australia

S3: UVic, Arbutus, ComputeCanada, plus 6 Minio instances on various clouds



# Belle-II dynafed jobs

*past:* mount dynafed with gfalFS

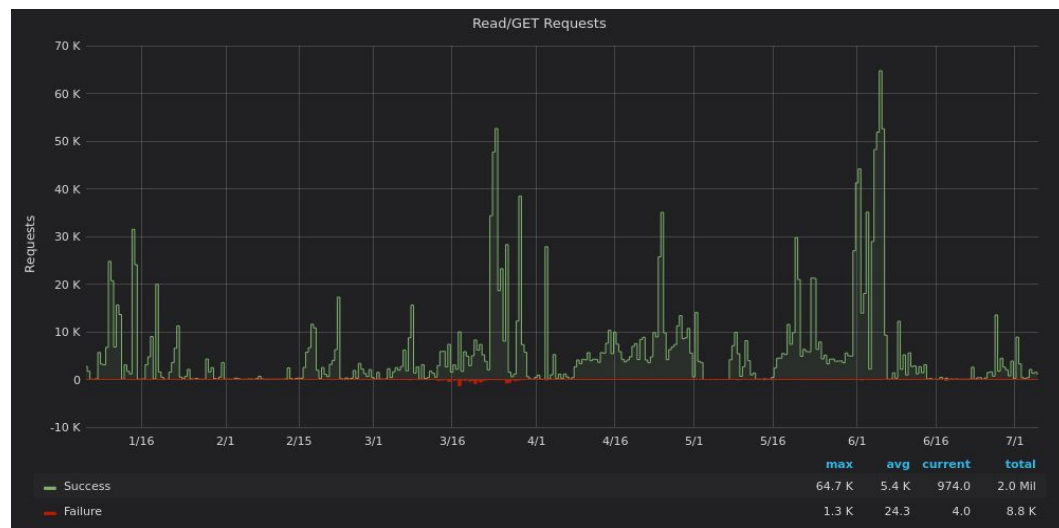
*present:* gfal2 adoption allows usage as SE

UVic dCache only writable endpoint

Read through WebDAV

Write through SRM

- *note:* other Belle-II storage only speaks SRM



# DOMA TPC on UVic dynafed

Manual data management to local cloud S3s

DIRAC tracks content of UVic dCache

3rd party copy is enabled and working but:

Multiple Apache processes means, check  
after may miss if landing on different process

Note: Belle-II uses SRM for transfers between SEs



---

**3<sup>rd</sup> Party Copy**

# CERN Dynafed

TPC will use Macaroon if given. Dynafed does not share secret with connected storage

⇒ can't issue valid macaroon

Gridsite bug: proxy delegation does not work when addressing storage using davs

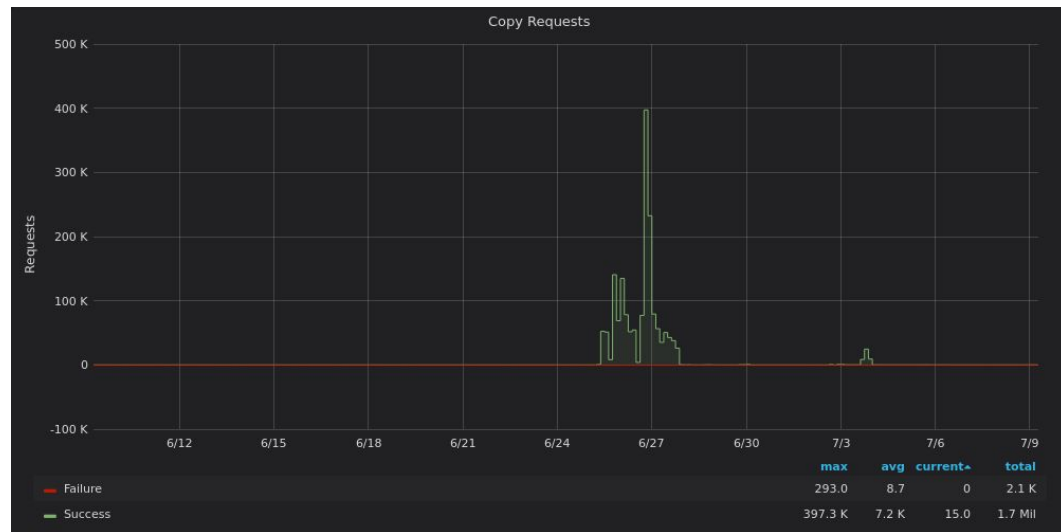
⇒ always use https [[davix patch](#)]



# TRIUMF Dynafed

Working since added to DOMA TPC tests and ATLAS DDM

Initial scaling issues for Rados gateways of backend storage are fixed



---

# Monitoring & accounting

# Dynafed storage stats

Python (3) package to track quota and free space

<https://pypi.org/project/dynafed-storagestats/>

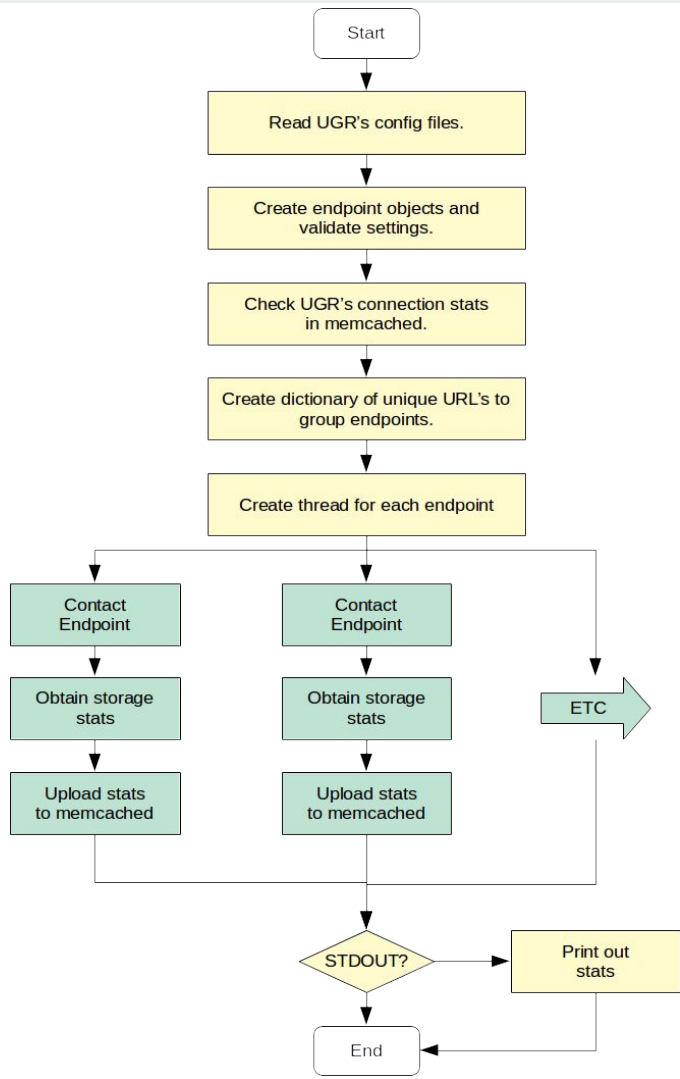
Adds locplugin configurations to manage calculation

Set dynafed to respect quota with `glb.minfreespace`

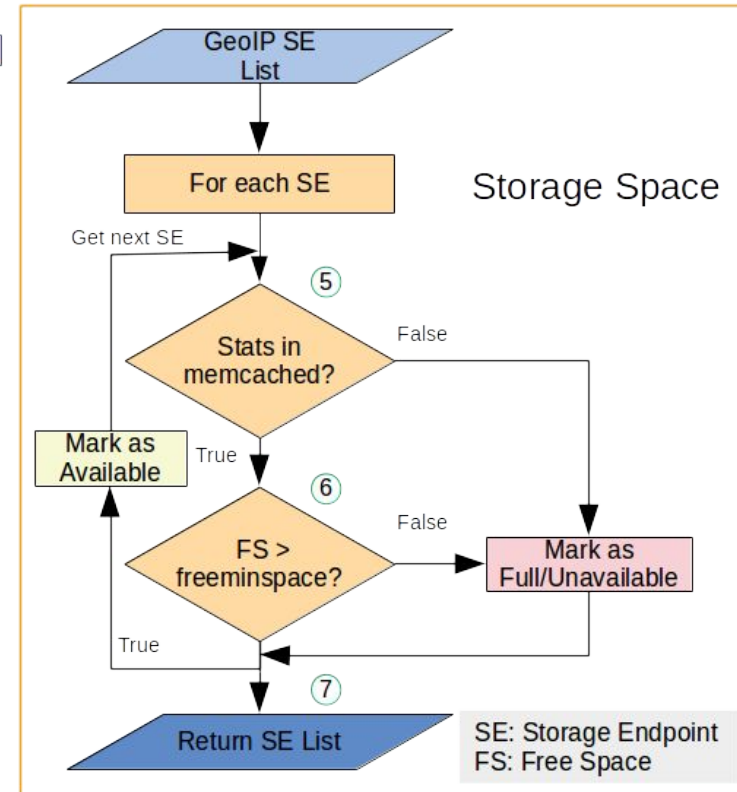
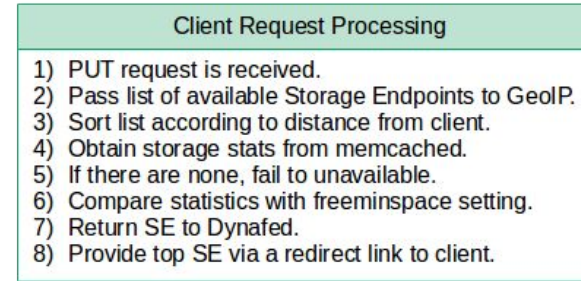
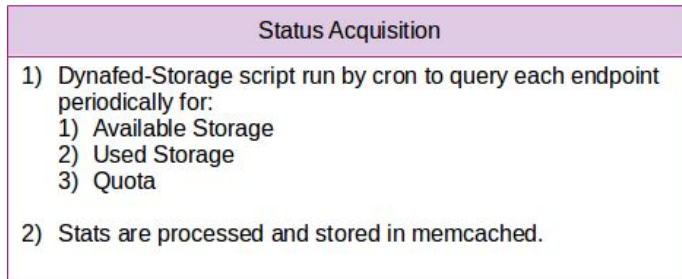
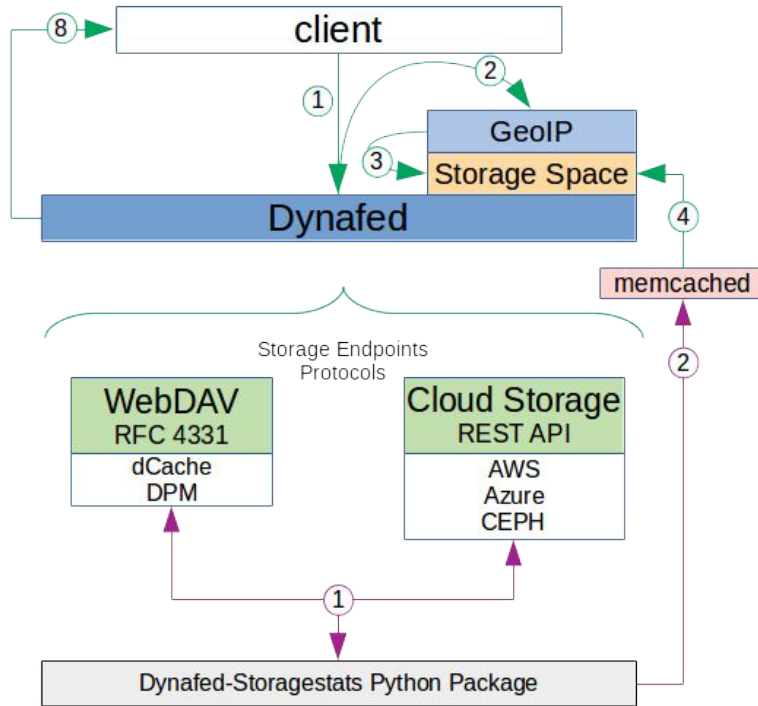
- Option per locplugin instead?

report functionality generates file dump and WLCG JSON

- BDII is going away, where else to report?
- CephS3: best to use rados admin interface



# Selecting a Storage Endpoint for Writing





# Beats, Elasticsearch, and Grafana

1. Filebeat parses http and ugr logs
2. Metricbeat collects http and system metrics
3. Execbeat collects stats from memcached

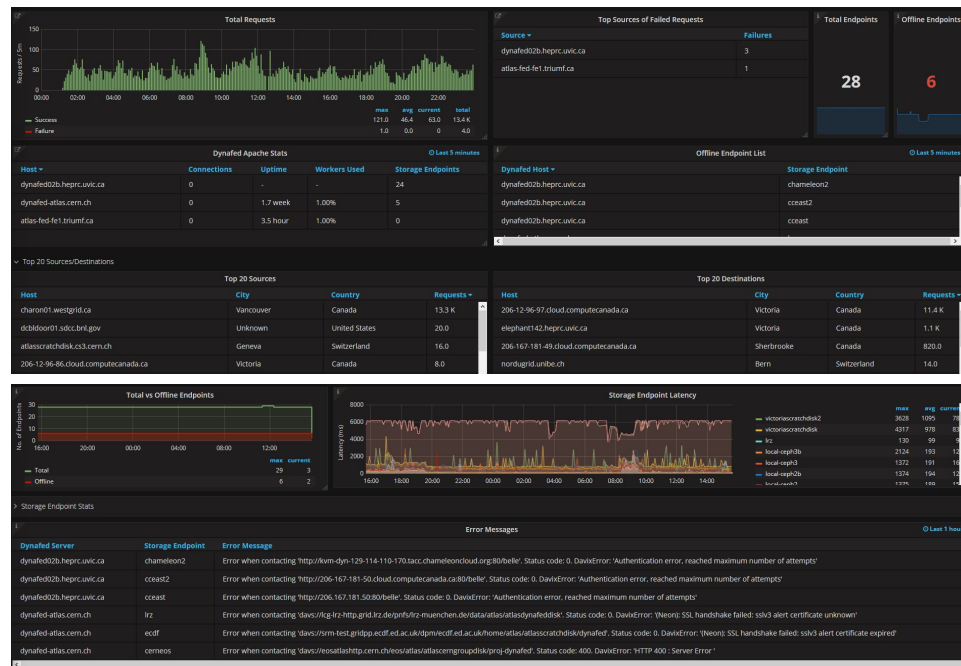
Beats report to Elasticsearch through Logstash

Grafana serves visualizations

Full configuration of system in the linked below

Ansible playbooks: <https://atlas-fed-git.triumf.ca/ffgalindo/atlas-fed-it>

Detailed overview: <https://heprc.blogspot.com/2019/06/monitoring-dynafed-with-elk.html>



---

# Ongoing work



# Short term plans with dynafed

Focus on using Dynafed to provide access to object storage

Implement checksum calculation everywhere: Dynafed becomes read and write endpoint at Belle-II

If dynafed is asked to be the active party in a TPC request, reliably stream data through Dynafed

Create scalable implementations of checksum calculation and 3<sup>rd</sup> party copy



# Scaling checksum calculation

## Issues for checksums of files on S3

- Downloading file means latency for large files
- Competes with dynamo for resources, esp.
  - disk
  - bandwidth

Proposal: store hashes in [object metadata](#)

Logic: Check for key, if exists use, else instruct a 4<sup>th</sup> party to download file, calculate hashes and add keys.

## System defined key

Content-MD5 = base64 encoded MD5 hash

Add user keys **x-amz-meta-[checksum name]** e.g.

x-amz-meta-adler32

x-amz-meta-crc32

question: base64 encode for consistency?



# Scaling 3<sup>rd</sup> party copy

Issues for checksums of files on S3

- Competes with dynafed for resources, esp.
  - disk
  - bandwidth
- Dynafed starts behaving like FTS

Proposal: send copy orders to a 4th party:

FTS or custom system?

Hook into checksum scaling:

Pull: use Content-MD5 mechanism to ensure file consistency on backend storage.

Issue Want-Digest request source for other checksums and add keys to object on S3



# Long term plans

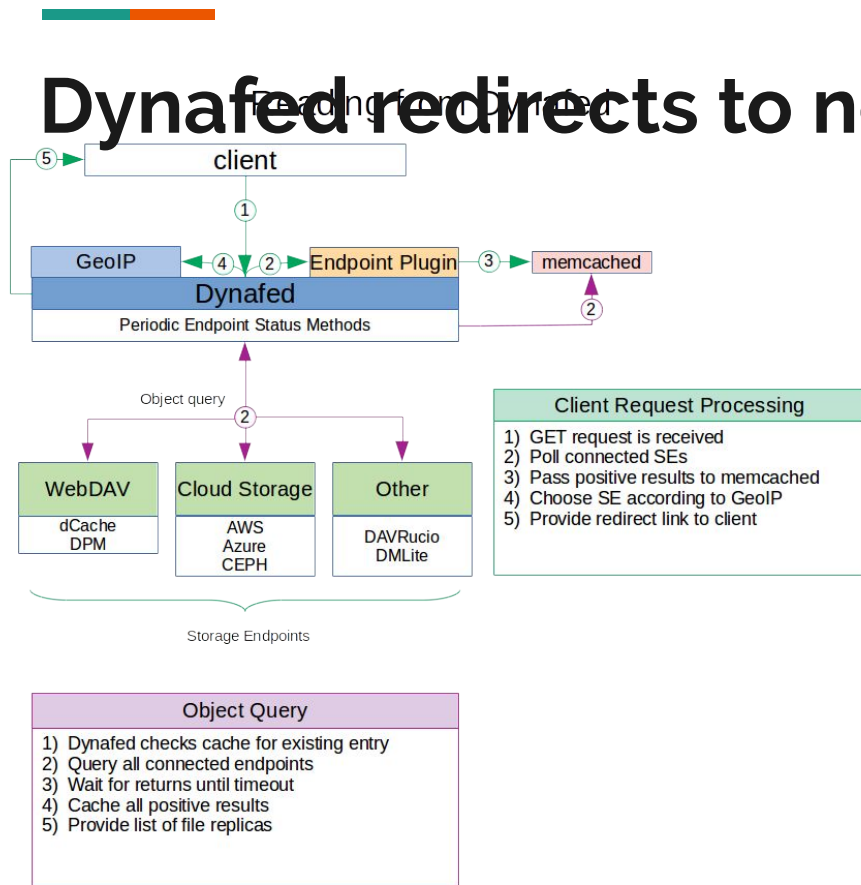
Federate multiple storages for write and read

Automate data replication between storage endpoints (based on popularity?)

# Thank you

---

# DynaFed: Redirect To Nearby Storage



- Dynafed redirects to close storage

- Operating three configurations:

- Belle-II at UVic:
  - R/O access (production)
- ATLAS at CERN:
  - R/W to cloud storage (dev)
  - R/W to grid storage (dev)

- Instances operated by others:



# Experience With ATLAS and Dynafed

- Early challenges:
  - MKCOL and MOVE not supported by Dynafed (not guaranteed on object stores), solutions:
    - Rucio protocol that does not rename files after upload
    - Dynafed allows MKCOL by creating directories in cache and making appropriate calls on file system endpoints
  - Rucio Replication to https/davs endpoints get stuck
    - Manually make calls to FTS and register data
- Early Success:
  - Functional tests run against Dynafed with CephS3 endpoint

# Rucio, Dynafed, and Checksums

- Rucio “TODO” fix causes jobs to fail: client can’t get **checksum**
- Mechanism:
  - Grid: User is responsible, Want-Digest [[RFC3230](#)]
  - Cloud: Provider is responsible, Content-MD5 [[RFC1544](#)]
- Algorithm
  - Grid: ADLER32 [[RFC1950](#)], for many reasons, can support others
  - Cloud: MD5 [[RFC1321](#)] (because it was there?)
- *Workaround*: Flag for Rucio not to request checksum from Dynafed

# Dynafed, FTS, and Proxies

- Rucio used legacy proxies with FTS
  - Some old SEs may not have migrated to RFC
  - Legacy proxy delegation through Dynafed to LRZ dCache were denied
  - Solution: switch to RFC proxy
- MKCOL bug:
  - Dynafed's MKCOL implementation [[slide 12](#)] broken for davs endpoints
  - Fixed in Dynafed 1.3.2
- Automated replication by Rucio to/from Dynafed works

# Debug & Development Cycle

- Rucio, Dynafed, and FTS developers are quick to respond and helpful
- conservative deployment policies → long development cycle
- Example: Rucio client & checksums
  - In March implemented behavior for ~~rucio client to respect~~ “verify\_checksum” flag
  - Debug: caught wrong exception (last week in ATLAS test pilots)
  - This week: implement correct exception
  - Next week: patch released
  - Few weeks: Wait for deployment of patch in cvmfs

Pull Request, Automated Tests, Peer Review, ...

# Victoria Dynafed for Belle-II

- Belle-II *is developing* gfal2 support for their DDM and WMS
  - Will allow direct usage of Dynafed as SE in the future
- Workaround:
  - Belle-II allows job configuration to access locally mounted volume
  - gfalFS provides fuse mount within Linux directory tree:
    - `gfalFS -s ${HOME}/b2data/belle davs://dynafed02.heprc.uvic.ca:8443/belle`
  - Jobs access Belle-II data from “local” directory `~/b2data/belle`
- gfalFS only needs Dynafed as access point for any cloud
  - Dynafed redirects gfalFS to closest endpoint via GeolP

# Victoria Dynafed for Belle-II

- Endpoints behind Belle-II Dynafed:
  - Compute Canada East (Minio via S3 API)
  - Compute Canada West (Minio via S3 API)
  - Amazon (S3)
  - Chameleon Cloud (Minio via S3 API)
  - Victoria Tier-2 SE (dCache, Victoria SE for Belle-II)
  - Victoria HEPRC Ceph (CephS3)

# DynaFed Belle-II Authentication & Authorization

- Based on [X.509](#) VOMS proxy
- Python-module based authentication
  - Implemented [grid-mapfile for authentication](#)
  - Plain text file read by python script for authorization, example:

```
/atlas atlas rlwd  
/belle belle rlwd  
/minio admin rldw  
/localCeph admin rlw
```

- Proxy sent to worker from DIRAC and used for gfalFS mount

# Belle-II Experience With gfalFS And Dynafed

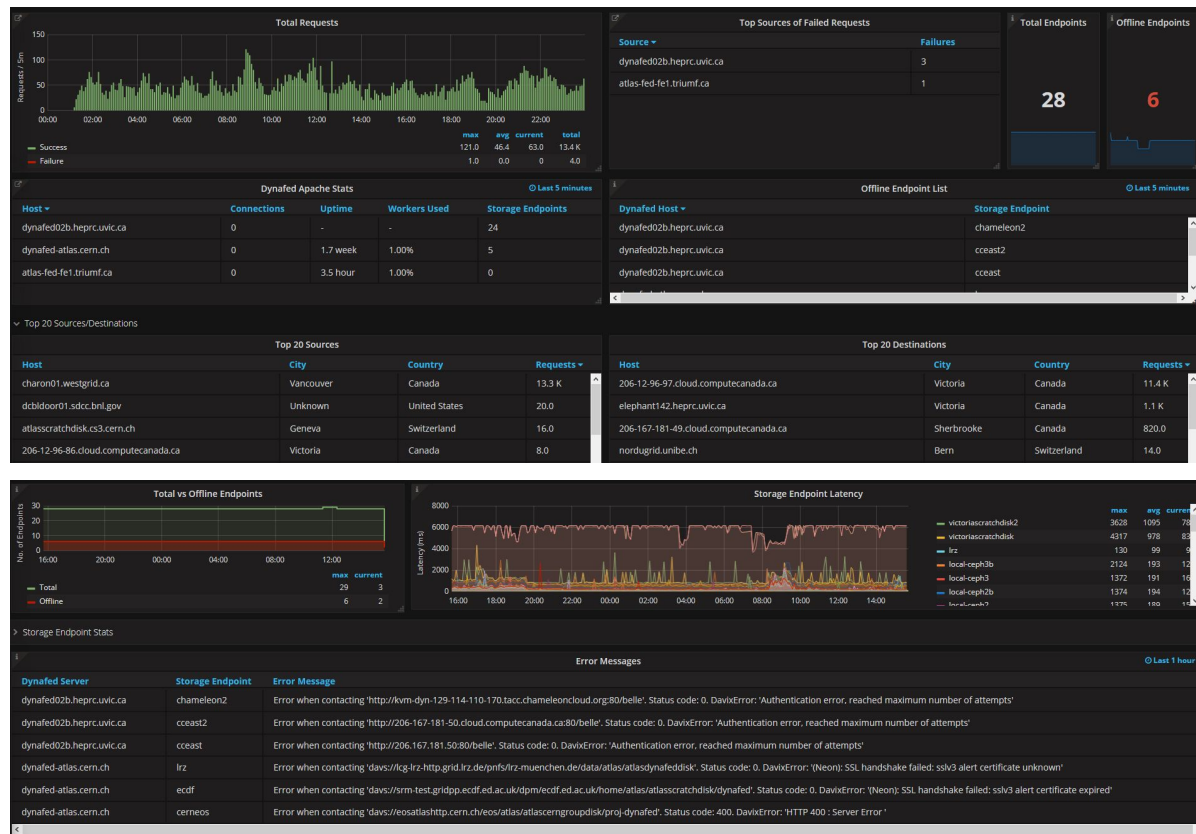
- **Load is balanced** across co-located storage endpoints
  - In recent MC campaign jobs each pulled one (1) of 20 input data sets - each ~5GB
  - 3000 parallel jobs → 30TB per day. Ran smoothly through Dynafed.
- **Easy and effective network usage**
  - Same configuration for all workers (6 separate clouds are used for Belle-II)
  - With same files used by many jobs network transfers stay local
- **Easy addition of new endpoints**
  - Added traditional Belle-II SEs while transferring new input data sets to our Endpoints



# Belle-II Experience With gfalFS And Dynafed

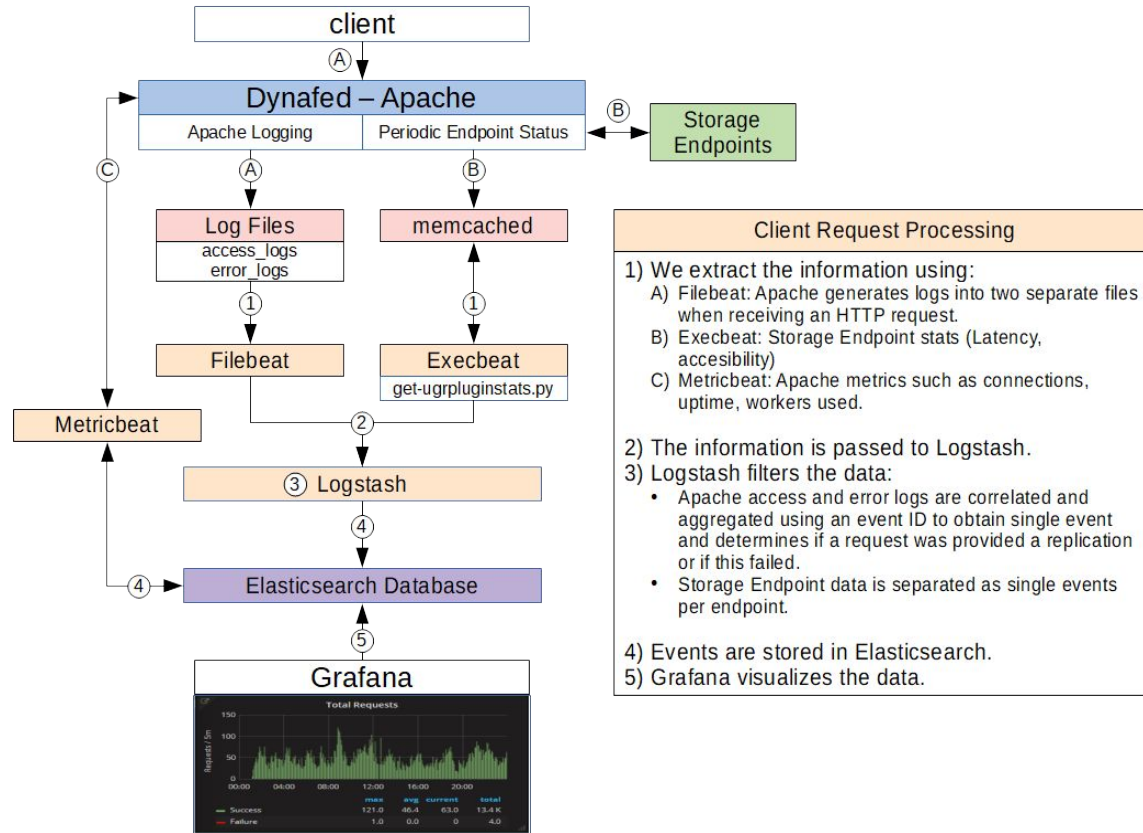
- System stability
  - Data is accessed locally with fail-overs observed when local storage becomes unavailable
    - *local* means nearby storage in Dynafed
  - Placing endpoints close to each other effectively balanced load
- In production for almost 1 year
- Segfaults of gfalFS (in libcrypto.so) since CernVM update
  - Intermittent issues, thus hard to debug
  - Appeared during Spectre/Meltdown patches
  - Observed when same endpoint is mounted multiple times on one

# Elasticsearch, Logstash, and Grafana



- Main Dashboard
  - Apache Metrics Stats
  - Online/Offline SE's
  - Top Sources and Destinations
  - Total Success/Fail Requests
- Storage Status Dashboard
  - Error messages
  - Online/Offline SE's
  - SE's Latency to Dynafed.

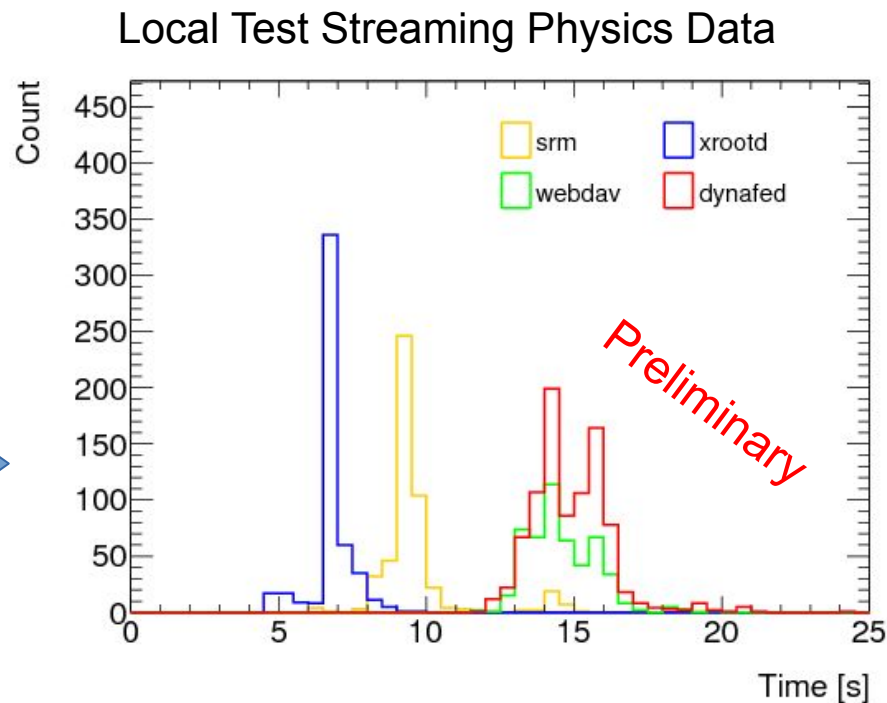
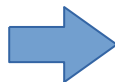
# Elasticsearch, Logstash, and Grafana



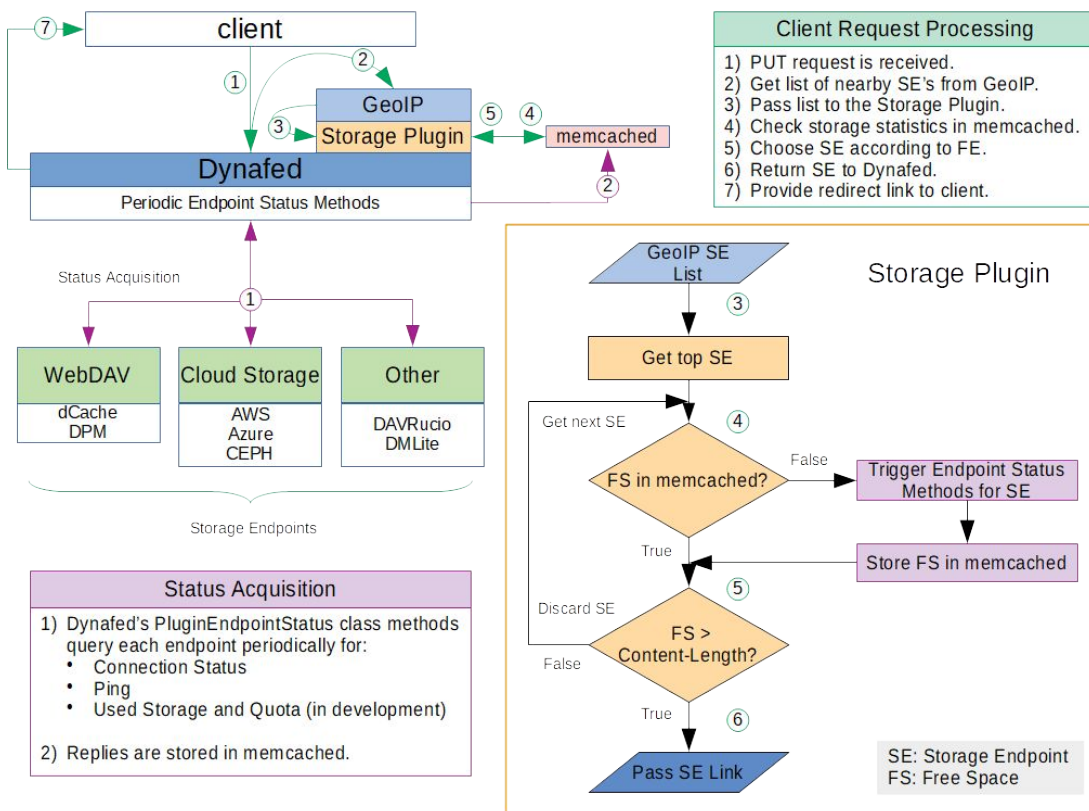
- Filebeat: Apache access and error log files
- Execbeat: Python script to extract stats from memcached
- Metricbeat: Apache metrics
- Grafana display for:

# Next Steps with Dynafed

- Integrate gfal2 into Belle-II
  - Next MC campaign needs 9TB, with 2GB per job
  - Full dataset cloud will be more challenging
- Run ATLAS functional tests (again)
- Benchmark Dynafed:
  - HTTP+WebDAV vs xrootd/srm
  - HTTP+WebDAV with Dynafed vs direct



# Next Steps: Dynafed Storage Plugin



- Issue with writing to Dynafed
  - Free space on endpoints unknown
- Need method to query usage and quota from endpoints
- Need common protocol for this information
- Commercial providers don't provide quota: we choose?