Cloud storage with the Dynafed data federator

Marcus Ebert
mebert@uvic.ca
on behalf of the HEP-RC UVic group:
Frank Berghaus, Kevin Casteels, Colson Driemel, Colin Leavett-Brown, Michael Paterson, Rolf Seuster, Randall Sobie,
Ryan Taylor (University of Victoria)
Fernando Fernandez Galindo, Reda Tafirout (TRIUMF)

- What is Dynafed
- Why do we want Dynafed
- Installations and tests at UVic
- Features, challenges, and things to work on...

What is Dynafed

- redirector for a dynamic data federation, developed by CERN-IT (fabrizio.furano@cern.ch)
  - for data transfers, client is redirected to a storage element with the data
  - this can be done depending on geographic location
    - storage elements closer to the job are preferred

What is Dynafed

- redirector for a dynamic data federation, developed by CERN-IT (fabrizio.furano@cern.ch)
  - for data transfers, client is redirected to a storage element with the data
  - this can be done depending on geographic location
    - storage elements closer to the job are preferred

Example:

file: https://dynafed02.heprc.uvic.ca:8443/belle/MC/release-00-09-00/DB00000265/BG15th/phase3/set14/BHWide.tgz
meta data: curl -k https://dynafed02.heprc.uvic.ca:8443/belle/MC/release-00-09-00/DB00000265/BG15th/phase3/set14/BHWide.tgz?metalink

What is Dynafed

- redirector for a dynamic data federation, developed by CERN-IT (fabrizio.furano@cern.ch)
  - for data transfers, client is redirected to a storage element with the data
  - this can be done depending on geographic location
    - storage elements closer to the job are preferred

Example:

file: https://dynafed02.heprc.uvic.ca:8443/belle/MC/release-00-09-00/DB00000265/BG15th/phase3/set14/BHWide.tgz
meta data: curl -k https://dynafed02.heprc.uvic.ca:8443/belle/MC/release-00-09-00/DB00000265/BG15th/phase3/set14/BHWide.tgz?metalink

on cc-east cloud (Sherbrooke):
http://206.167.180.208:80/belle/MC/...
https://gridftp02.clumeq.mcgill.ca:8443/webdav/belle/DATA/belle/MC/...
http://129.114.33.181:80/belle/MC/...
https://s3-uvic.dev.computecanada.ca/rjsBucket/belle/MC/...
http://elephant132.heprc.uvic.ca/mebucket/belle/MC/...

on cc-west cloud (Victoria):
https://s3-uvic.dev.computecanada.ca/rjsBucket/belle/MC/...
http://129.114.33.181:80/belle/MC/...
https://gridftp02.clumeq.mcgill.ca:8443/webdav/belle/DATA/belle/MC/...
http://206.167.180.208:80/belle/MC/...

What is Dynafed

- redirector for a dynamic data federation, developed by CERN-IT (fabrizio.furano@cern.ch)
  - for data transfers, client is redirected to a storage element with the data
  - this can be done depending on geographic location
    - storage elements closer to the job are preferred
- access through http(s)/dav(s) protocols
- can federate existing sites without configuration changes at sites
  - site needs to be accessible by http(s)/dav(s) (DPM, dCache, plain xrootd with plugin,...)
  - world wide distributed data can be made accessible under common name space and through a single endpoint
- X509/VOMS based authentication/access authorization can be used with dynafed
  - **http://HEPrc.blogspot.com** for grid-mapfile based authentication/authorization
    - different posts have also links to dynafed installation instructions in our TWiki
- can also **directly access S3 and Azure based storage**
  - no credentials visible to the client
  - preauthorized URL with limited lifetime is used to access files on the storage

Advantages of using S3 based storage

- easy to manage
  - no extra servers needed, no need for the whole Grid infrastructure on site (DPM, mysql, apache, gridftp, xrootd, VOMS information, grid-mapfile, accounting, ...)
  - just use private/public access key in central Dynafed installation

- no need for extra manpower to manage a grid storage site
  - small group with budget to provide storage but no manpower for it: Just buy S3 based xTB for y years and put the information into dynafed ---> instantly available to the Grid, no need to buy/manage/update extra hardware
  - if university/lab has already large Ceph installation --> just ask for/create a bucket, and put credentials in dynafed

- industry standard
  - adapted from Amazon by Open Source and commercial cloud and storage solutions
    - HPC, Openstack, Ceph, Google, Rackspace cloud storage, NetApp, IBM,...

- scalable
  - traditional local file storage servers based on traditional filesystems will become harder to manage/use with growing capacity needs, same for other “bundle” solutions (DPM,...)
  - raid5 dead, raid6 basically dead too, ZFS will get problems with network performance

Access to S3 based storage

glb.locplugin[]: libugrlocplugin_s3.so localceph2 2 http://elephant132.heprc.uvic.ca/mebucket/belle
locplugin.localceph2.xlatepfx: /belle /
locplugin.localceph2.s3.pub_key: <PUPLIC-KEY>
locplugin.localceph2.s3.priv_key: <PRIVATE-KEY>
locplugin.localceph2.writable: true
locplugin.localceph2.s3.signaturevalidity: 3600
locplugin.localceph2.s3.region: us-east-1
locplugin.localceph2.s3.alternate: true

Access to S3 based storage

glb.locplugin[]: libugrlocplugin_s3.so localceph2 2 http://elephant132.heprc.uvic.ca/mebucket/belle
locplugin.localceph2.xlatepfx: /belle /
locplugin.localceph2.s3.pub_key: <PUPLIC-KEY>
locplugin.localceph2.s3.priv_key: <PRIVATE-KEY>
locplugin.localceph2.writable: true
locplugin.localceph2.s3.signaturevalidity: 3600
locplugin.localceph2.s3.region: us-east-1
locplugin.localceph2.s3.alternate: true

---

gfal-copy davs://dynafed02.heprc.uvic.ca:8443/belle/datadisk/space-usage.json local-file.json

Process: Contact Dynafed --> Dynafed looks where file is ---> Dynafed gets authorized link -->
---> Dynafed redirects client to this link ---> Client access file direct on S3 through that link

Access to S3 based storage

glb.locplugin[]: libugrlocplugin_s3.so localceph2 2 http://elephant132.heprc.uvic.ca/mebucket/belle
locplugin.localceph2.xlatepfx: /belle /
locplugin.localceph2.s3.pub_key: <PUPLIC-KEY>
locplugin.localceph2.s3.priv_key: <PRIVATE-KEY>
locplugin.localceph2.writable: true
locplugin.localceph2.s3.signaturevalidity: 3600
locplugin.localceph2.s3.region: us-east-1
locplugin.localceph2.s3.alternate: true

---

gfal-copy davs://dynafed02.heprc.uvic.ca:8443/belle/datadisk/space-usage.json local-file.json

Process: Contact Dynafed --> Dynafed looks where file is ---> Dynafed gets authorized link -->
---> Dynafed redirects client to this link ---> Client access file direct on S3 through that link

http://elephant132.heprc.uvic.ca/mebucket/belle/datadisk/space-usage.json?X-Amz-Signature=2d8fc601379eb9e43dc16219a9da11452e8b7c0a22ad98186aaa4fe841b97e53&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=TJHJA902TSSJZ659E9D5%2F20171016%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20171016T061053Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host

Authentication/Authorization in Dynafed

- built-in VOMS proxy based authentication
  - in ugr.conf: *glb.allowgroups[]: belle /belle/ rl*
  - includes authentication and authorization
  - does not work in web browsers

- python-module based authentication possible
  - can be used for anything you want to use
  - exit code 0 = access granted
  - exit code 1 = access denied

- implemented usage of grid-mapfile for authentication
  - more info in our blogpost
  - plain text file used for authorization, read by the python module

Authentication/Authorization in Dynafed

- built-in VOMS proxy based authentication
  - in ugr.conf: *glb.allowgroups[]: belle /belle/ rl*
  - includes authentication and authorization
  - does not work in web browsers

- python-module based authentication possible
  - can be used for anything you want to use
  - exit code 0 = access granted
  - exit code 1 = access denied

- implemented usage of grid-mapfile for authentication
  - more info in our blogpost
  - plain text file used for authorization, read by the python module

Example of access file:

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

Authentication/Authorization in Dynafed

- built-in VOMS proxy based authentication
  - in ugr.conf: *glb.allowgroups[]: belle /belle/ rl*
  - includes authentication and authorization
  - does not work in web browsers

- python-module based authentication possible
  - can be used for anything you want to use
  - exit code 0 = access granted
  - exit code 1 = access denied

- implemented usage of grid-mapfile for authentication
  - more info in our blogpost
  - plain text file used for authorization, read by the python module

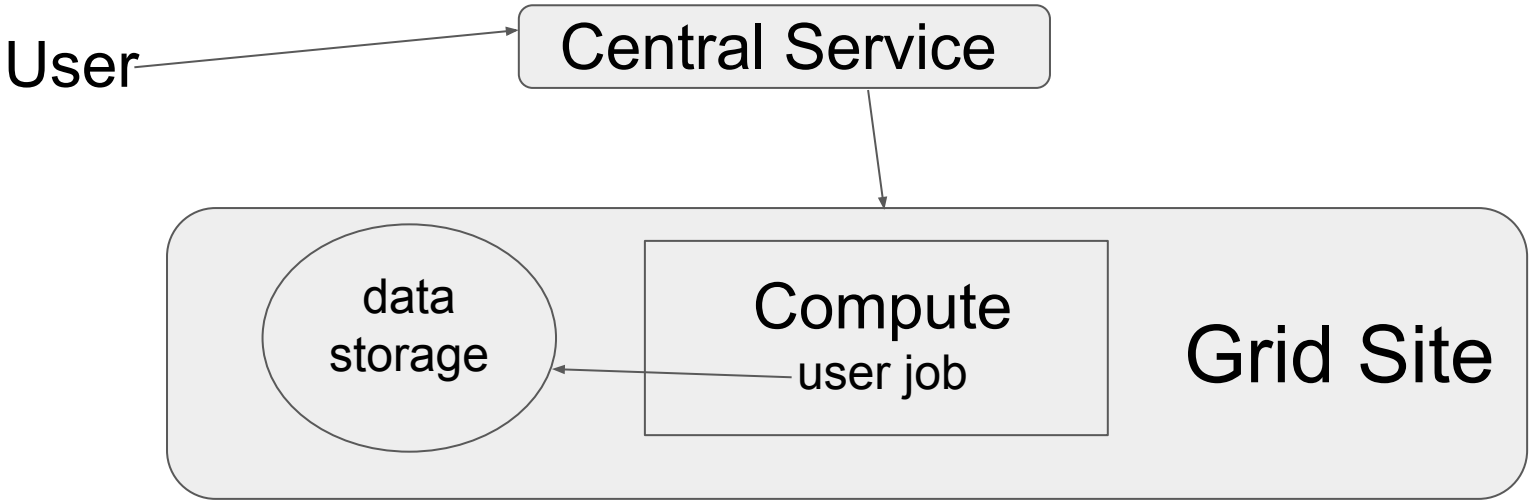
- also implemented alternative version of grid-mapfile usage without the need of a python module
  - you can read about it here

Why do we want (need) Dynafed...

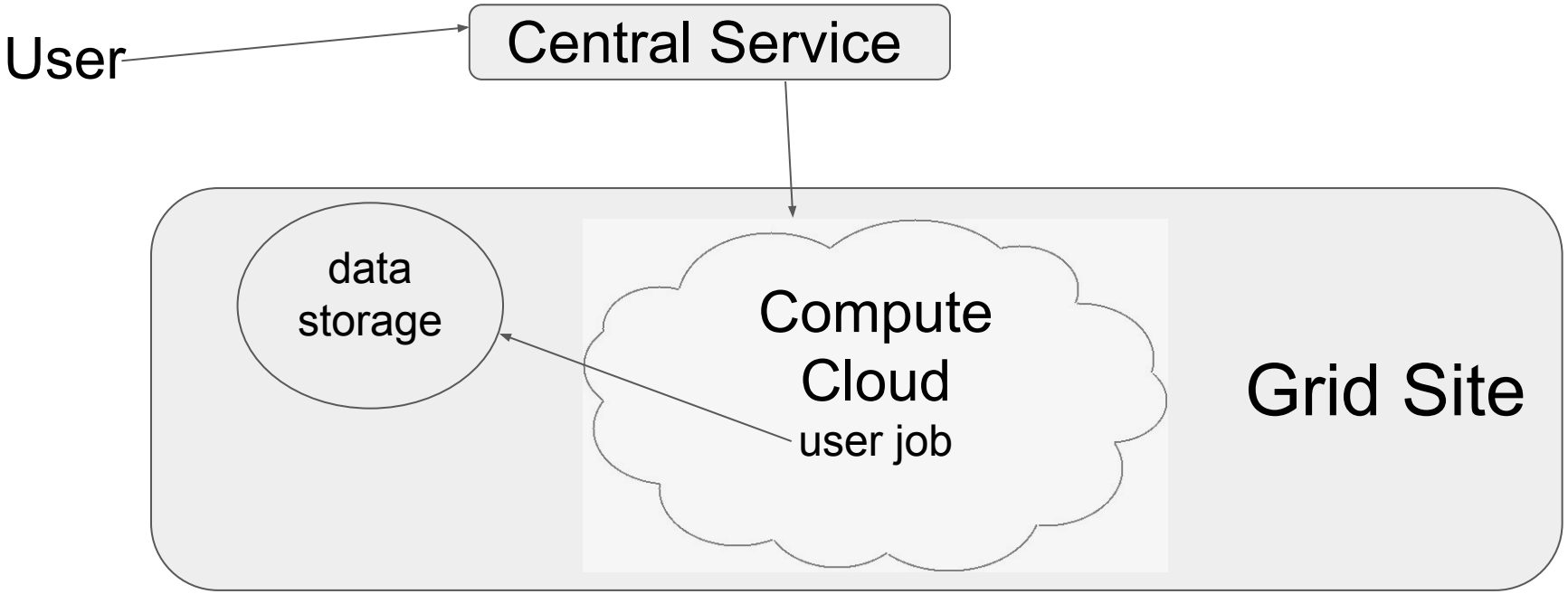
UVic Grid Site:

- Cloud computing using distributed cloud systems
- Clouds are distributed throughout Northern America and Europe right now
- Possibility to integrate other clouds anywhere
- CE: HTCondor + CloudScheduler
  https://indico.cern.ch/event/637013/contributions/2739289
- SE: so far traditional dCache site...

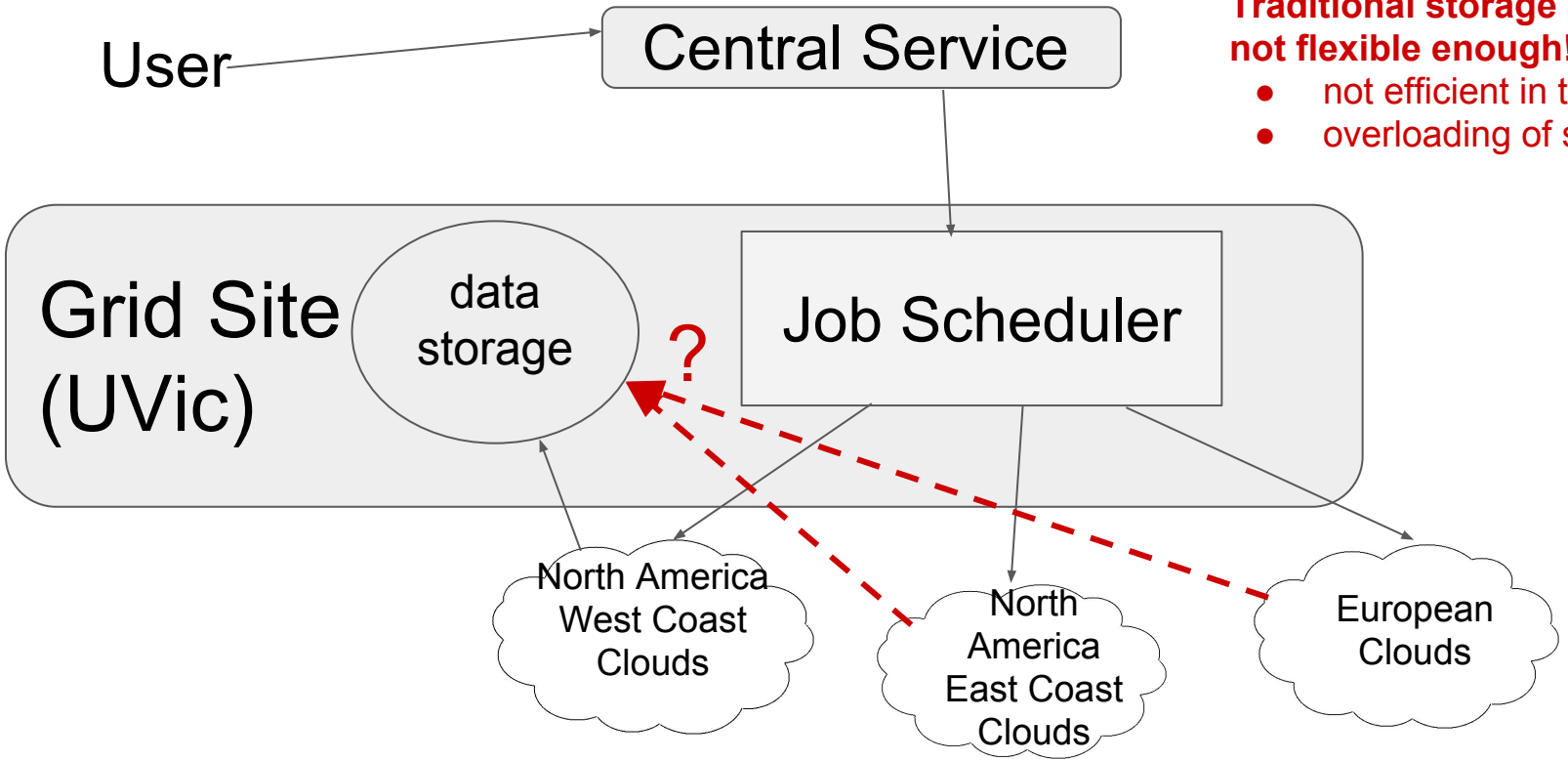
Traditional GRID site



Cloud computing for the GRID



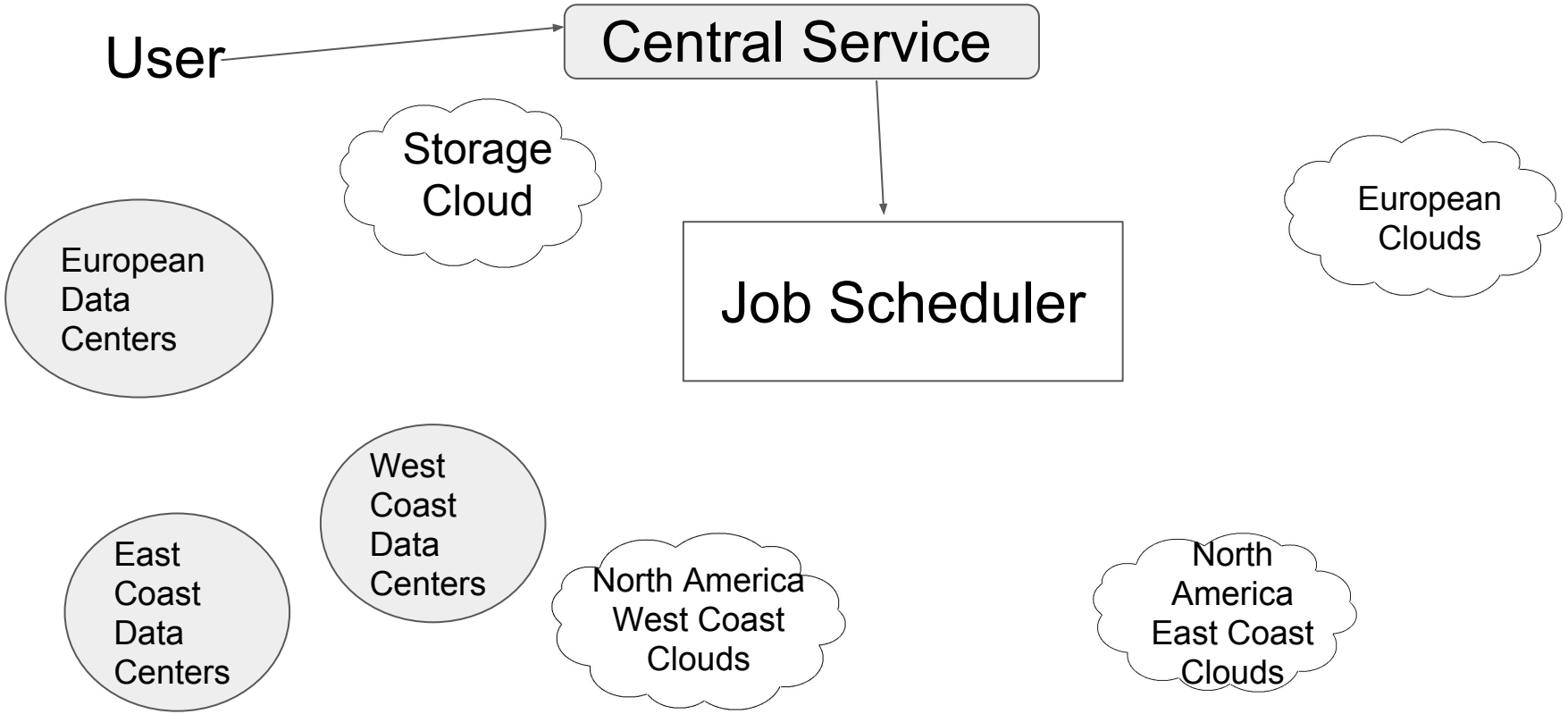
Cloud computing for the GRID



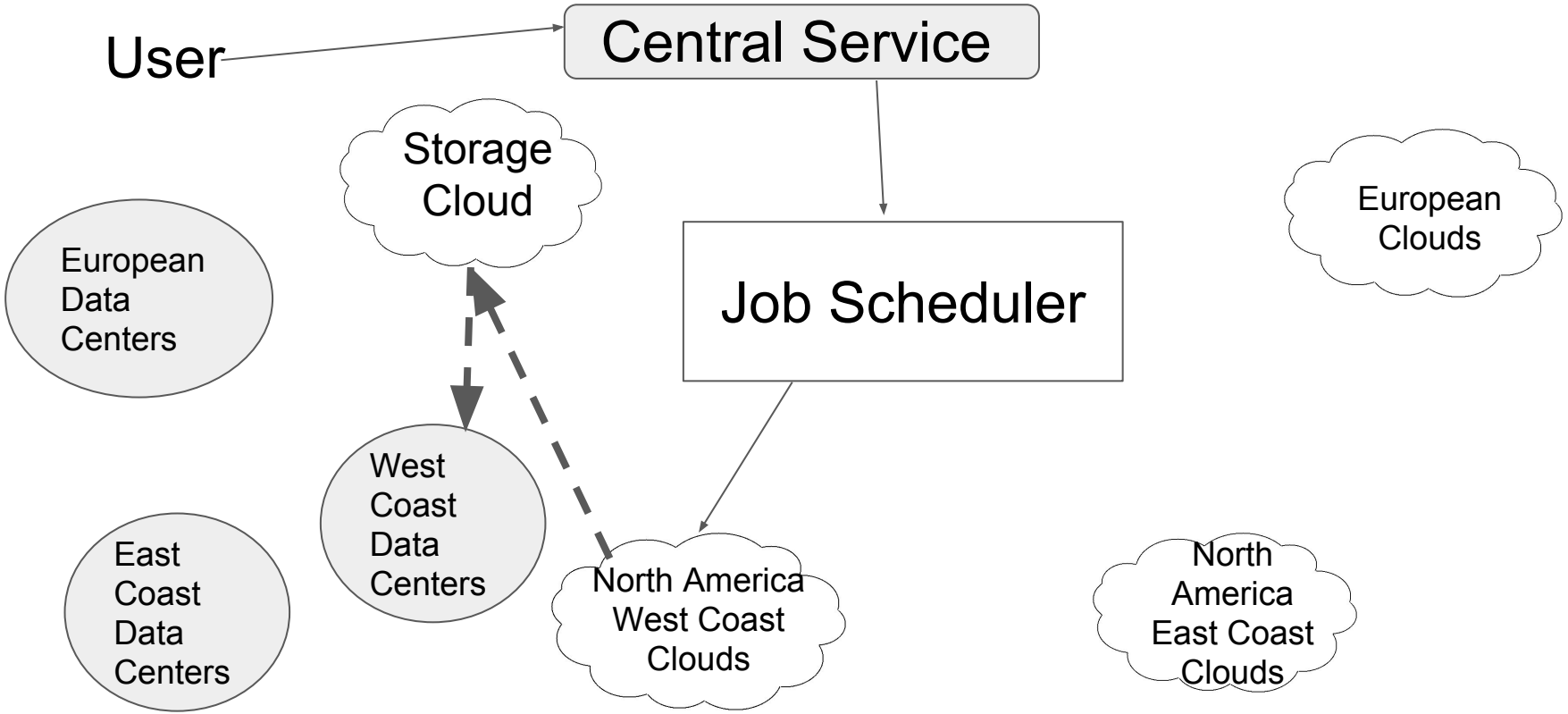
Problem:
Traditional storage systems are not flexible enough!

- not efficient in that situation
- overloading of single SE

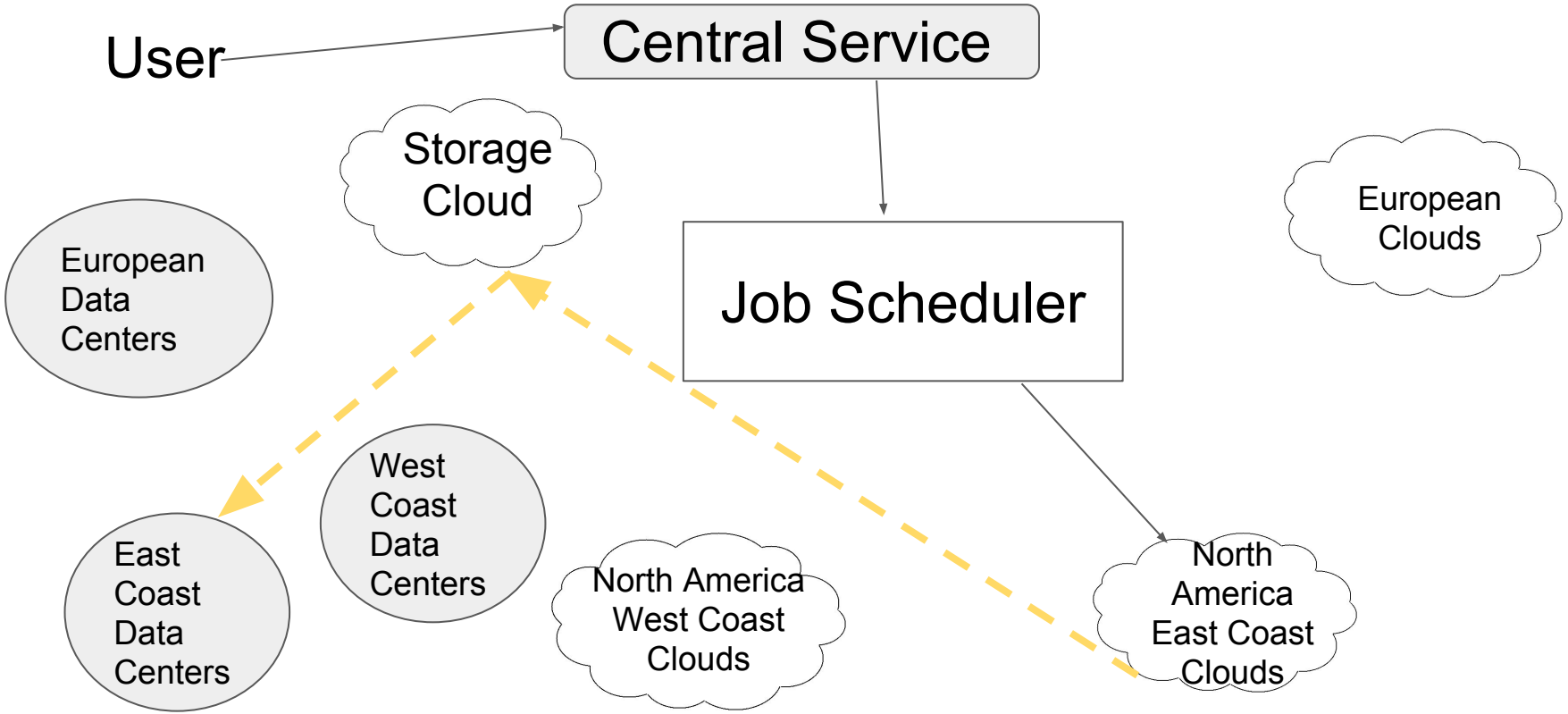
Cloud storage for the GRID



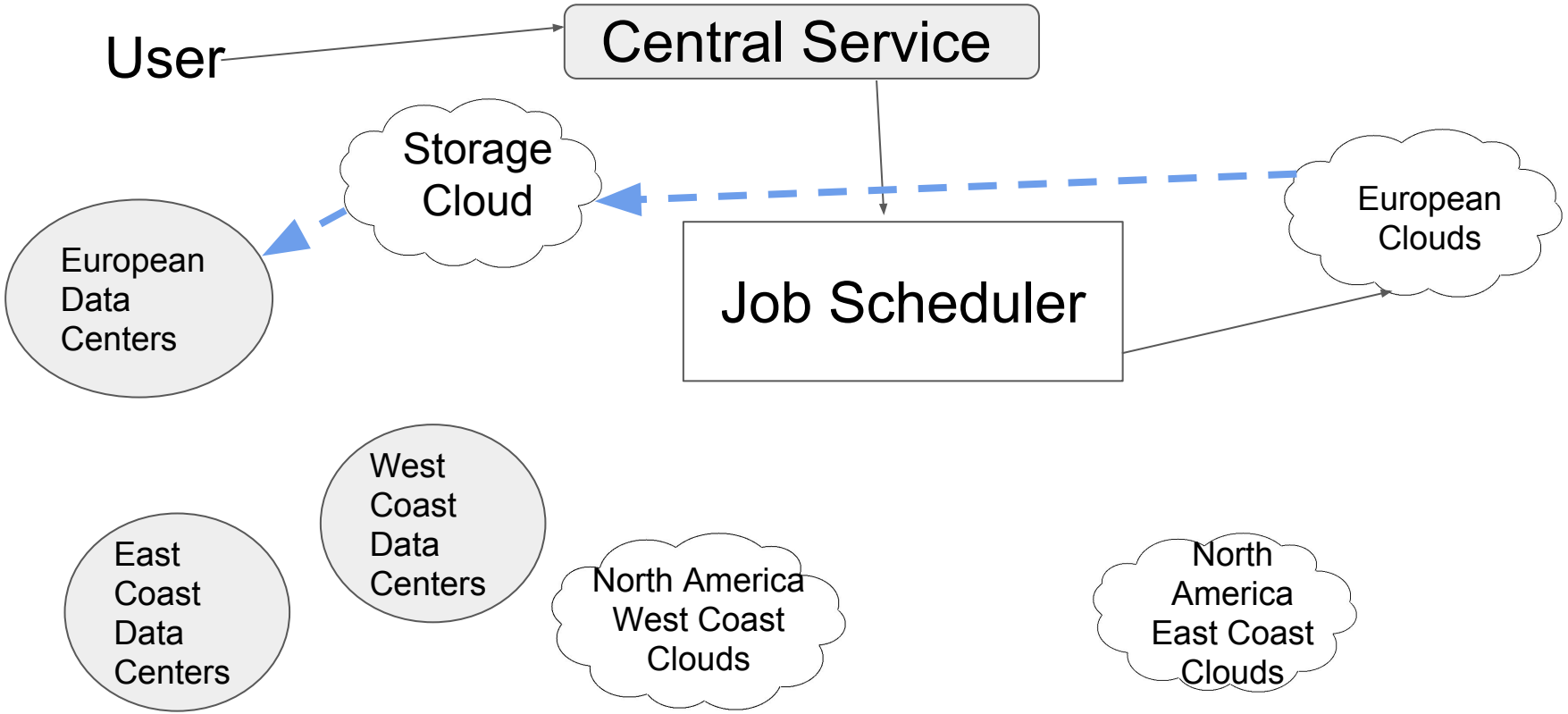
Cloud storage for the GRID



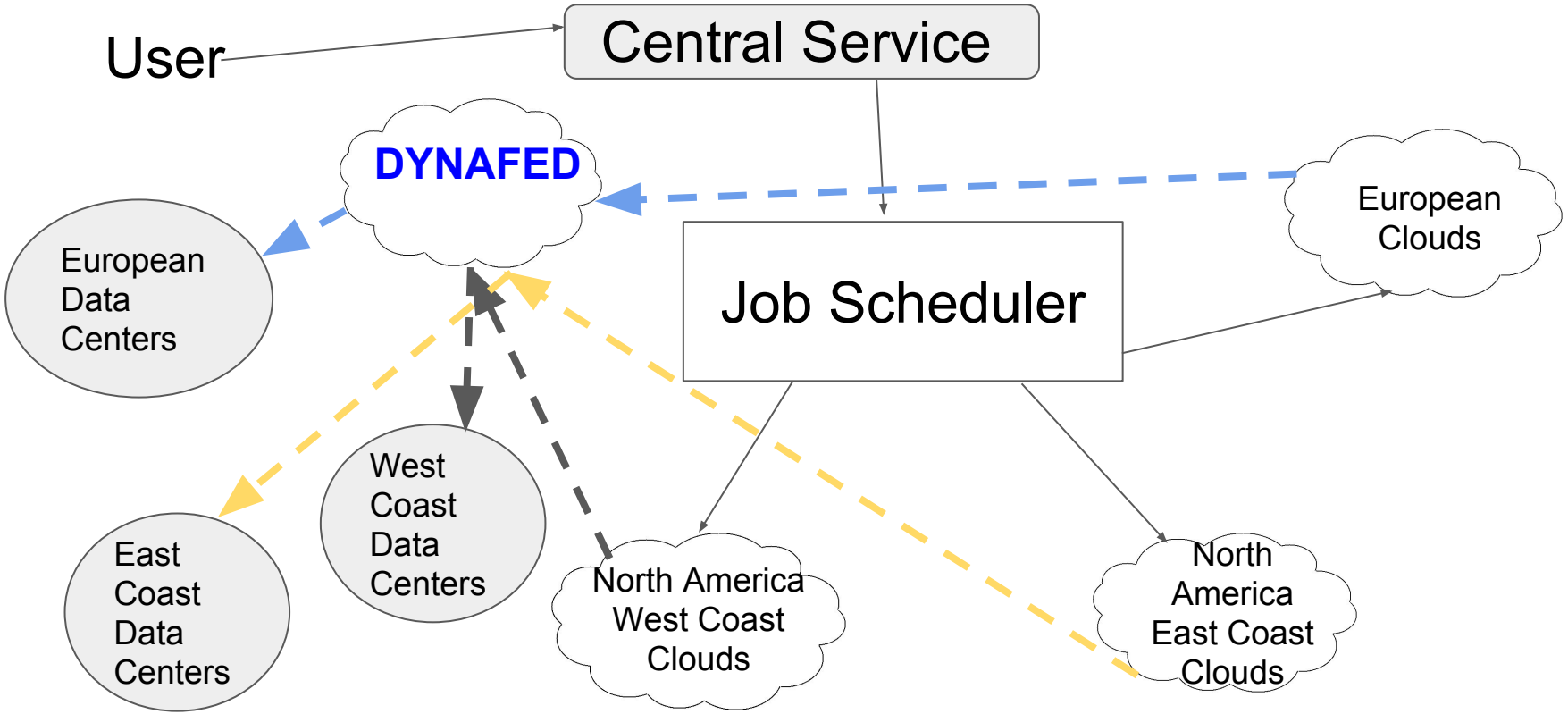
Cloud storage for the GRID



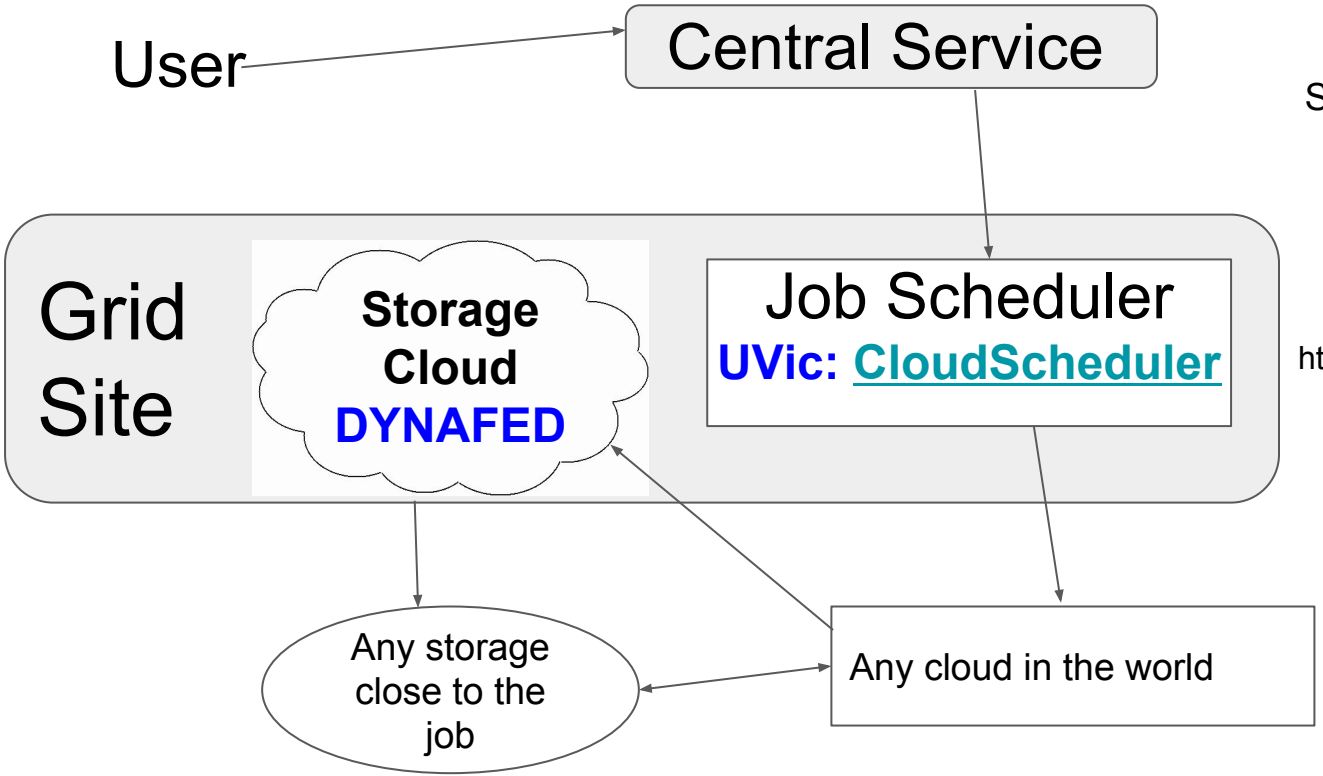
Cloud storage for the GRID



Cloud storage for the GRID



Distributed cloud-storage and cloud-compute for the GRID



Storage cloud not only **useful** for cloud jobs but **also for compute-only sites and experiments at a whole**!

CloudScheduler@HEPiX:
https://indico.cern.ch/event/637013/contributions/2739289

Dynafed@Victoria HEPRC group

- different test installations at CERN, Victoria, and TRIUMF
- so far mostly CEPH (S3 based) storage behind all installations for writing, ~O(50TB) but can be extended easily
  - federation of existing sites not used so far for writing
  - simple tools that provide a S3 API were very useful for testing: *https://www.minio.io/*

- storage in Victoria, TRIUMF, and CERN attached to each installation for testing
  - made it possible to test if the closest site is chosen depending on where a job runs

- CERN: enabled for Atlas automated test (HC) jobs, running well; working with davix/gfal/dynafed/Ceph developers and Atlas DDM; attached empty DPM to dynafed for writing of production jobs, **Thanks to Edinburgh GridPP site for test-DPM installation!**

- Victoria:
  - multi-VO enabled installation, can be used for Atlas and Belle-II access
  - used for distribution of Belle-II input files needed by all production jobs

- TRIUMF: testing ways of building and attaching CEPH clusters to Dynafed and log file analyses, e.g. Cipher influence on transfer speeds/CPU usage of https, monitoring of usage/redirection/...

Features, challenges, and things to work on...

Redirect and direct access with Dynafed

- client tools can get new redirect to another site if anything happens with an already established connection
  - site outage, network problems at a site,....

- root based tools can speak webdav and access data over network using dynafed
  - *TFile *f=TFile::Open("davs://dynafed.server:PORT/belle/path/to/file/file.root")*
  - uses external davix libraries

Possibility to open root files over the network with redirect to closest storage (on-site) and seamless switchover to other storage endpoints in case of problems!

Things we discovered in our tests

- files can not be renamed
  - not implemented in Dynafed
  - possibility of object stores behind Dynafed, which use hash computed from file name

- for 3rd party copies, Dynafed can not be the active party
  - Object store behind Dynafed can't initiate a transfer

works:
gfal-copy davs+3rd://dpm.server.org/path/file.root davs://dynafed.server.org/path/file.root
gfal-copy davs://dynafed.server.org/path/file.root davs+3rd://dpm.server.org/path/file.root

does not work:
gfal-copy davs+3rd://dynafed.server.org/path/file.root davs://dpm.server.org/path/file.root

in future gfal2 versions (already in davix-libs 0.6.6 and epel-testing):
gfal-copy davs://dpm.server.org/path/file.root davs://dynafed.server.org/path/file.root

- try all possible 3rd party copies with fallback
  - Source push to destination, if it fails then
  - Destination pull from source, if it fails then
  - Source -> Client machine -> Destination

Things we discovered in our tests

- Checksum handling for S3 different to what is used so far for Grid, different handling and algorithm
  - object stores support md5 while experiments use adler32

current Grid usage:

- client transfers file and asks for checksum
- storage endpoint saves file, calculates checksum, and sends checksum to client
- client calculates checksum from source file
- client compares both checksum and decides if transfer was successful
- client sends delete request to storage if checksums are different

current cloud usage:

- client transfers file to storage and sends a checksum together with the file
- storage endpoint saves file to object store, calculates checksum and compares with the original checksum
- storage endpoint does not keep the file on object store if checksums are different
- client gets only OK for successful storage of file if file was retrieved, stored, and checksum check at the storage was successful

client responsibility vs storage endpoint responsibility, adler32 vs md5

Things we discovered in our tests

- gfal-sum does not support S3 based storage through Dynafed
  - in contact with developers to have that included in davix-libs/gfal2-tools
  - same for checksum support in gfal-copy

- **3rd-party copies using webdav poorly supported by sites**
  - DPM sites miss some config options and directory used for proxy cache not created by default (/var/www/proxycache/ with correct owner and permissions)
  - dCache sites need to open another port (default: 8445)
  - plain xrootd site ? (Anyone?)

- WLCG http monitoring does not include 3rd party copy
  - Could this be included?

- gfalFS: mount whole federation as “normal” file system
  - makes federation accessible to standard tools
  - uses fuse
  - access through mount point still uses geo-location based access
  - using in production for Belle-II simulation jobs until http access in Belle-II Dirac is enabled

Challenges in federating existing sites

- Problems come from data management systems that were **developed without** Dynafed in mind
  - Atlas distributed data management doesn't work with federating existing sites so far
    - can only handle physical copies
    - files appearing through dynafed have no physical copy within dynafed
      - double count of files

- Belle-II distributed data management system in development
  - can take into account the special adjustments needed for Dynafed right from the beginning
  - interested in usage of Dynafed, not only as site-storage but also as experiment wide tool
  - checksum issue also a problem in usage for experiments, for now
    - hope to get this fixed soon

We are in contact with the davix/gfal2/dynafed developers and distributed data management groups for Belle-II and Atlas to make use of it.
Many bugs and initial problems are already fixed.

Summary and Outlook

- Dynafed is a good tool to utilize S3 based storage like Ceph for Grid usage without the need of any other Grid infrastructure or additional access servers
  - interesting for sites providing storage to experiments, as well as for experiments to integrate it into their DDM workflow

- still work needed to have Dynafed replace traditional site SE when using S3 based storage
  - checksum support in davix/gfal-tools or in Dynafed itself

- Dynafed can already be used for federating existing sites
  - experiments needs to put support into their DDM tools
  - in the meantime, successful usage for Belle-II production through fuse module, gfalFS

- good working with developers at CERN
  - through personal contact and through the dynafed users forum
    - dynafed-users-forum@cern.ch
    - https://groups.cern.ch/group/dynafed-users-forum/

- work on sites needed to support 3rd party copies through webdav
  - include in site monitoring?
  - need to test plain xrootd sites, if exist (anyone?)

Thank you!

additional information:
HEP-RC blog
Dynafed@CERN
CERN dynafed users forum
CERN dynafed support archive