ComputeCanada Cloud External Relationship Development Meeting

Cloud Usage for Workloads in High Energy Physics

Utilizing Distributed Clouds for Compute and Storage

Marcus Ebert
mebert@uvic.ca

on behalf of the High Energy Physics Research Computing 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 we do

- running compute workload for High Energy Physics experiments
  - **ATLAS** (CERN, Switzerland) and **Belle-II** (KEK, Japan) currently
  - large international collaborations with continuously demand for compute resources

- integrated into the Worldwide Grid Infrastructure
  - for experiments we are a "normal" grid site

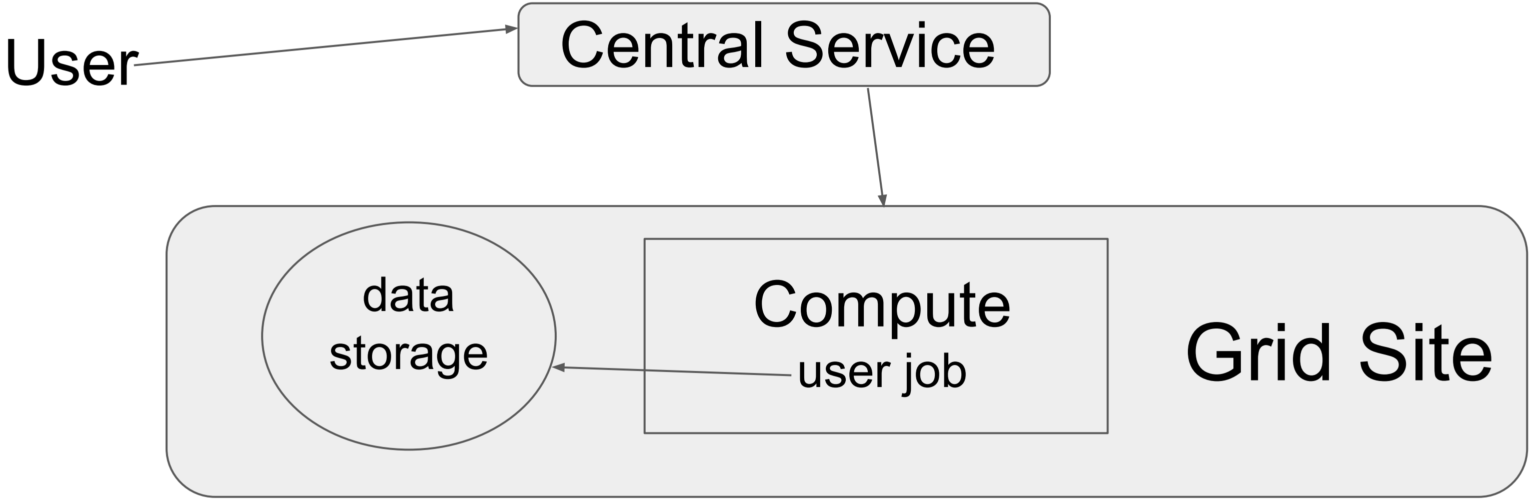
differences to a "normal" Grid site:

- we use VMs instead of bare meta l batch systems
- we run compute jobs not only locally in the same center where the data is
- dynamic VMs run in clouds at different locations

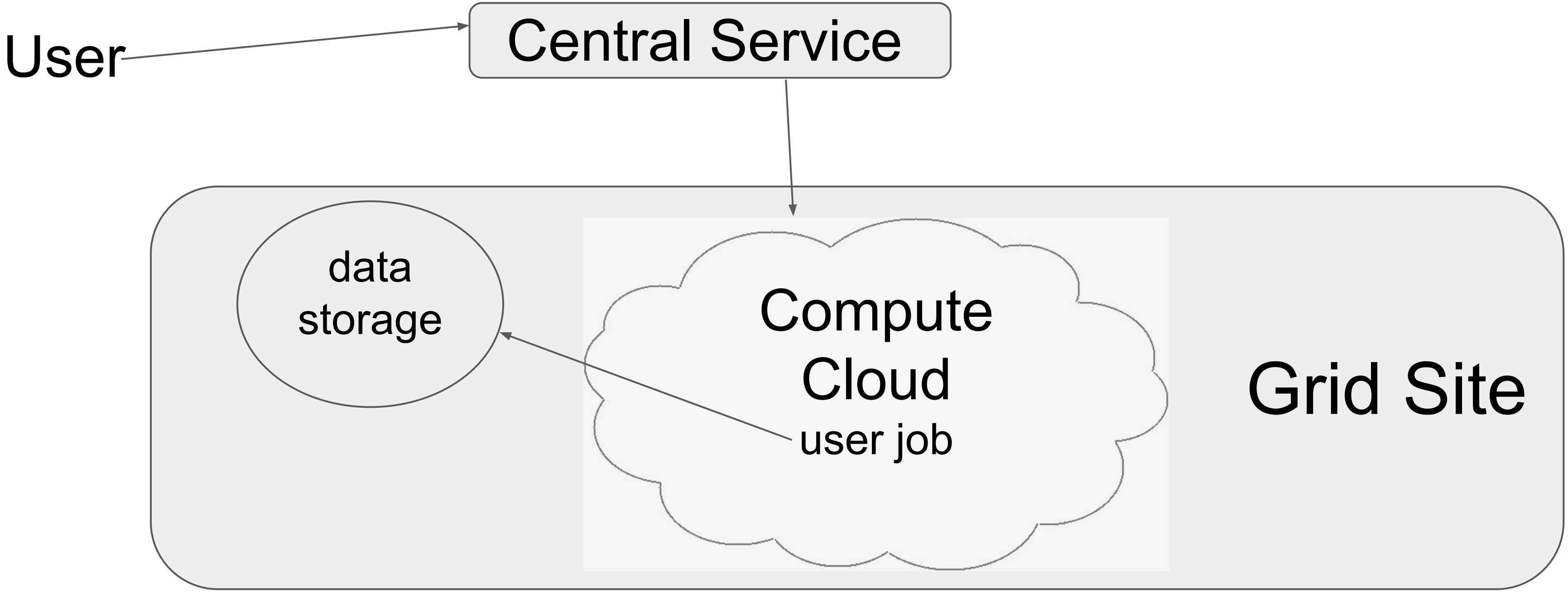
we need to handle:

- on-demand start/stop of VMs with correct resources
- image distribution
- uniform data handling

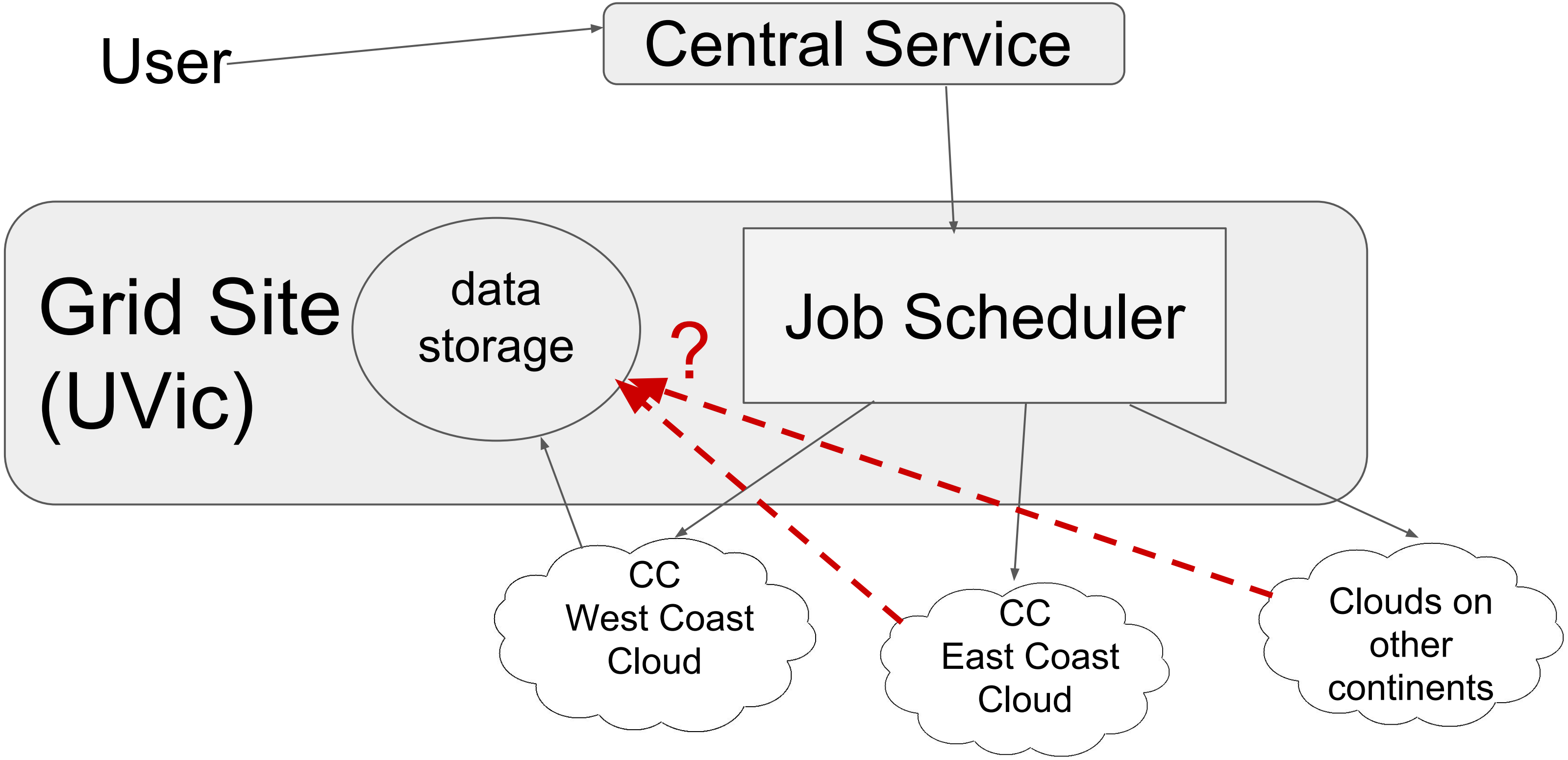
Traditional GRID site



Cloud computing for the GRID



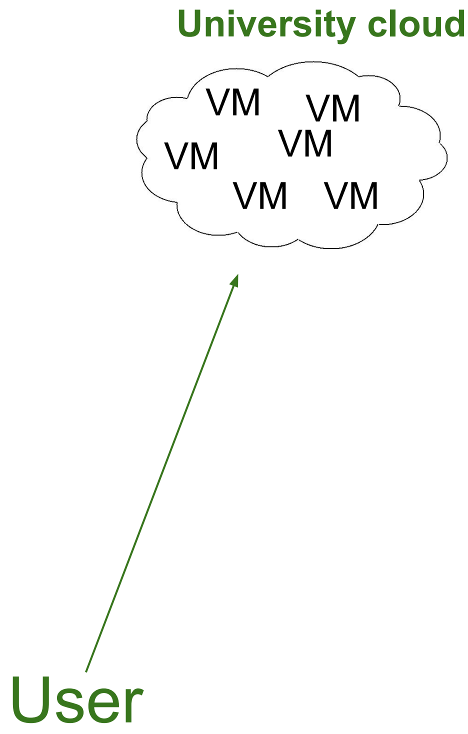
Cloud computing for the GRID



Distributed Cloud Compute

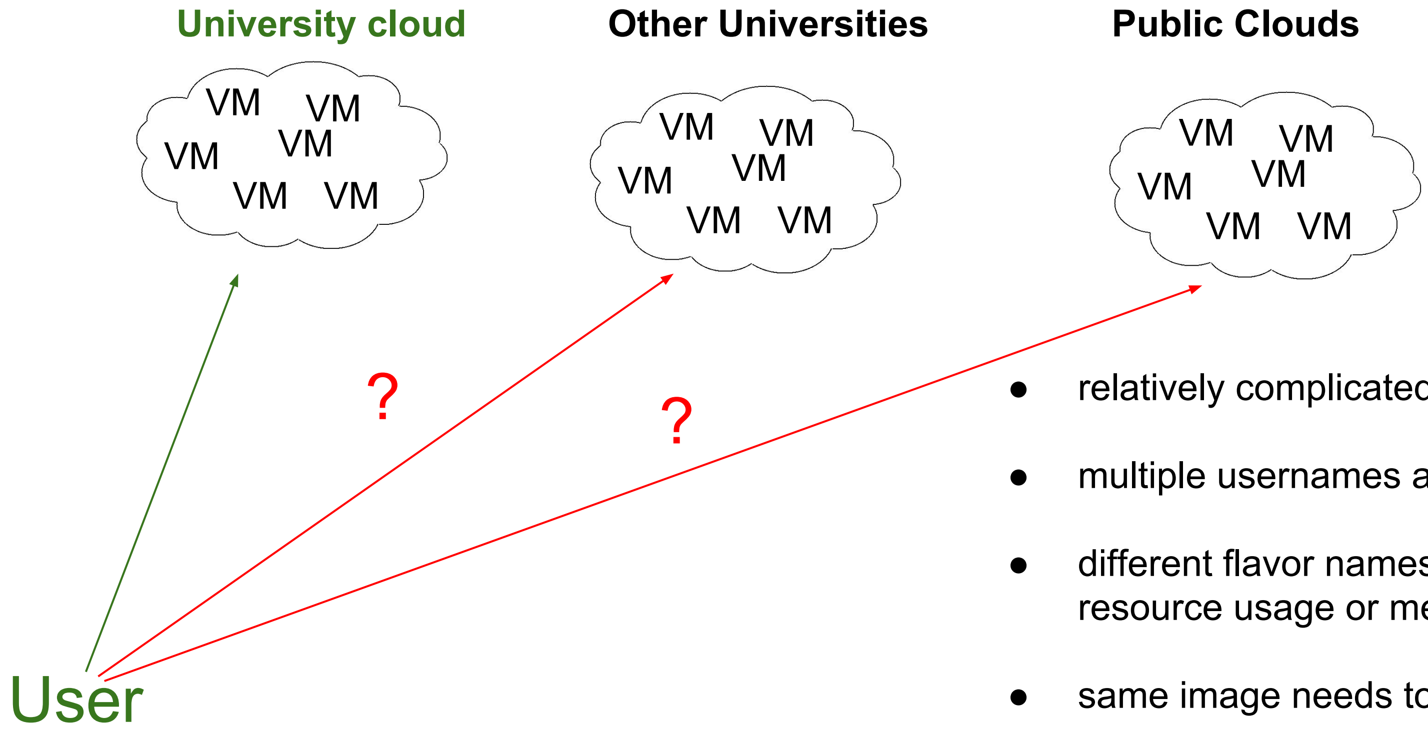
https://indico.cern.ch/event/637013/contributions/2739289

Single cloud compute



- relatively easy to handle
- just 1 set of user name, password, flavor names, and image

Multi-cloud compute



- relatively complicated to handle
- multiple usernames and passwords
- different flavor names with different resource usage or meanings
- same image needs to be everywhere

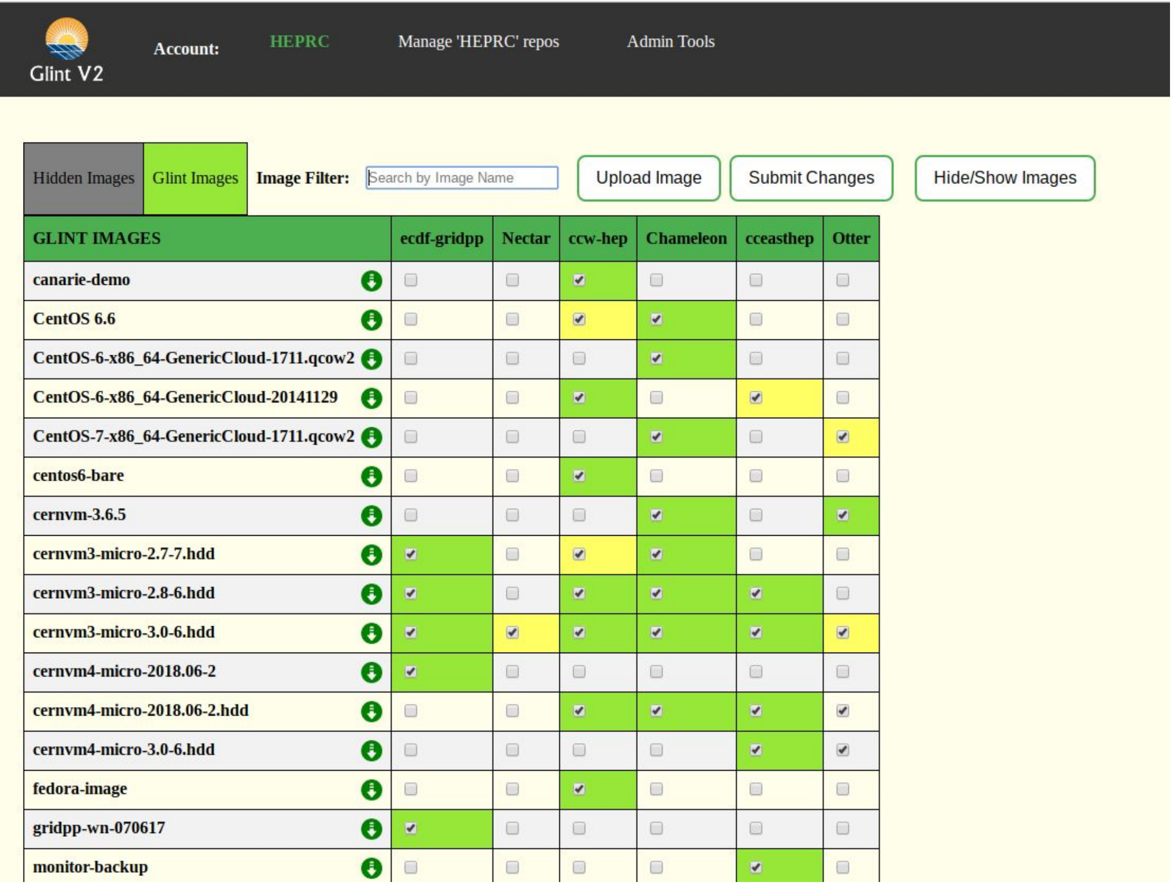
Multi-cloud compute at UVic

- developed program that takes care of VM start/termination: Cloudscheduler
- developed tool to distribute images across all clouds: Glint
- User does not need to know anything about clouds
- User only sees a batch system
  - HTCondor in our case
- developed Shoal to auto discover squids closest to the clouds
  - we make heavy use of CERNVM and CVMFS
- all used and developed software is Open Source and available on github

Cloudscheduler

- knows the allowed accounts and access URL to all used clouds
  - can use Openstack, OpenNebula, Amazon, Microsoft Azure, and Google Cloud
- can have defaults for flavor and image
  - one for all or for each cloud separate
  - can be overwritten by a job if needed
- queries batch system about idle jobs
  - are there idle jobs
  - what are the job requirements
- knows what resources are used and what resources can be used on all clouds
  - quota limits in cloudscheduler: configurable on the command line and config file
- when enough resources are available on a cloud then starts VMs that are needed by idle jobs
  - cloud-init to customize a VM
- when there are no idle jobs and VMs are idle too, shutdown unused VMs automatically
  - good on clouds where resources cost money
  - also needed to start new VMs with different flavors, depending on what a new job needs

Glint



- easy to use web interface where all supported clouds are visible
  - same tenants and accounts like in Cloudscheduler (Openstack only)
- possibility to upload an image to a cloud
  - e.g. from desktop through browser
- images that are on at least one cloud can easily be copied to all other clouds
  - in web interface just click a check box for that image on all clouds where it should be
- images can also be removed from clouds
  - just uncheck the box for that image on a cloud

Shoal

- Squid cache publishing and advertising tool
  - we need squid caches since all our VMs are running on CVMFS, and all use the same image no matter where they run

- consist of 3 parts
  - shoal server
  - shoal agent
  - shoal client

- **Shoal server :** lists all registered squids on the web and gives a list to the client, sorted by distance to the client using GeoIP DB
  - central machine, only one needed
  - we run it as a central service: http://shoal.heprc.uvic.ca/

- **Shoal agent:** will advertise a squid to Shoal server
  - needs to be installed on the squid that one want to be advertised

- **Shoal client:** will query the shoal server to get a list of close by squids
  - runs on VM that needs a squid for caching
  - changes CVMFS configuration on the VM to use the nearest squids
    - at startup of VM and then per cronjob at least twice a day

Multi-cloud compute at UVic

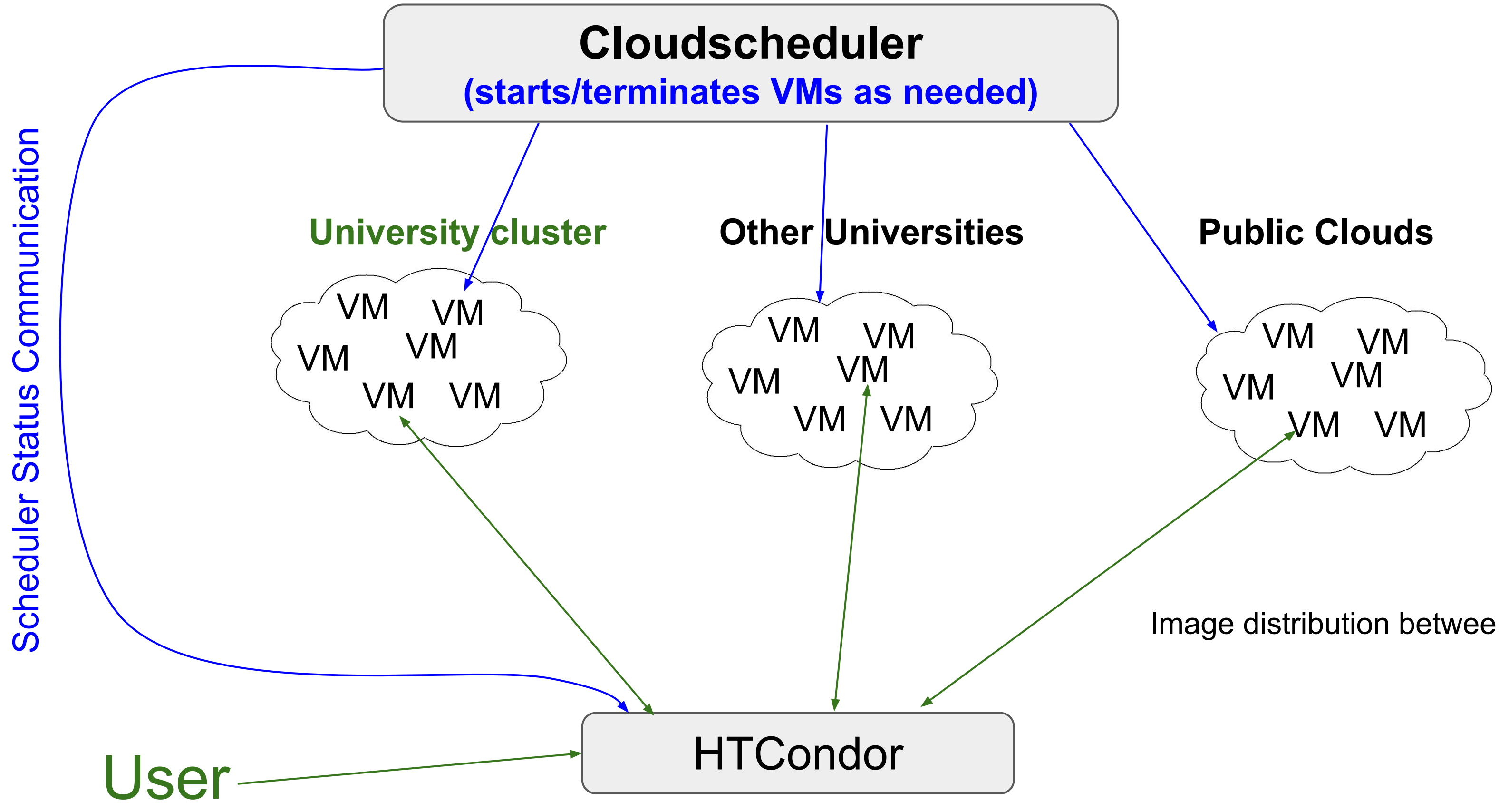


Image distribution between clouds: Glint

Multi-cloud compute at UVic

- run for several years successful with High Energy Physics workload
  - supporting the Atlas and Belle-II experiments within their Grid computing
  - usual workload:
    - transfer environment for the job to the worker node (VM)
    - get at least one input file from a centralized storage system
    - do some compute using the input file(s)
    - transfer all output to a centralized storage system

- currently using about 10 clouds
  - in Northern America and Europe
  - Australia in test mode

- about 5,000 cores used in parallel all the time
  - most cores located in the 2 Compute Canada clouds and at CERN

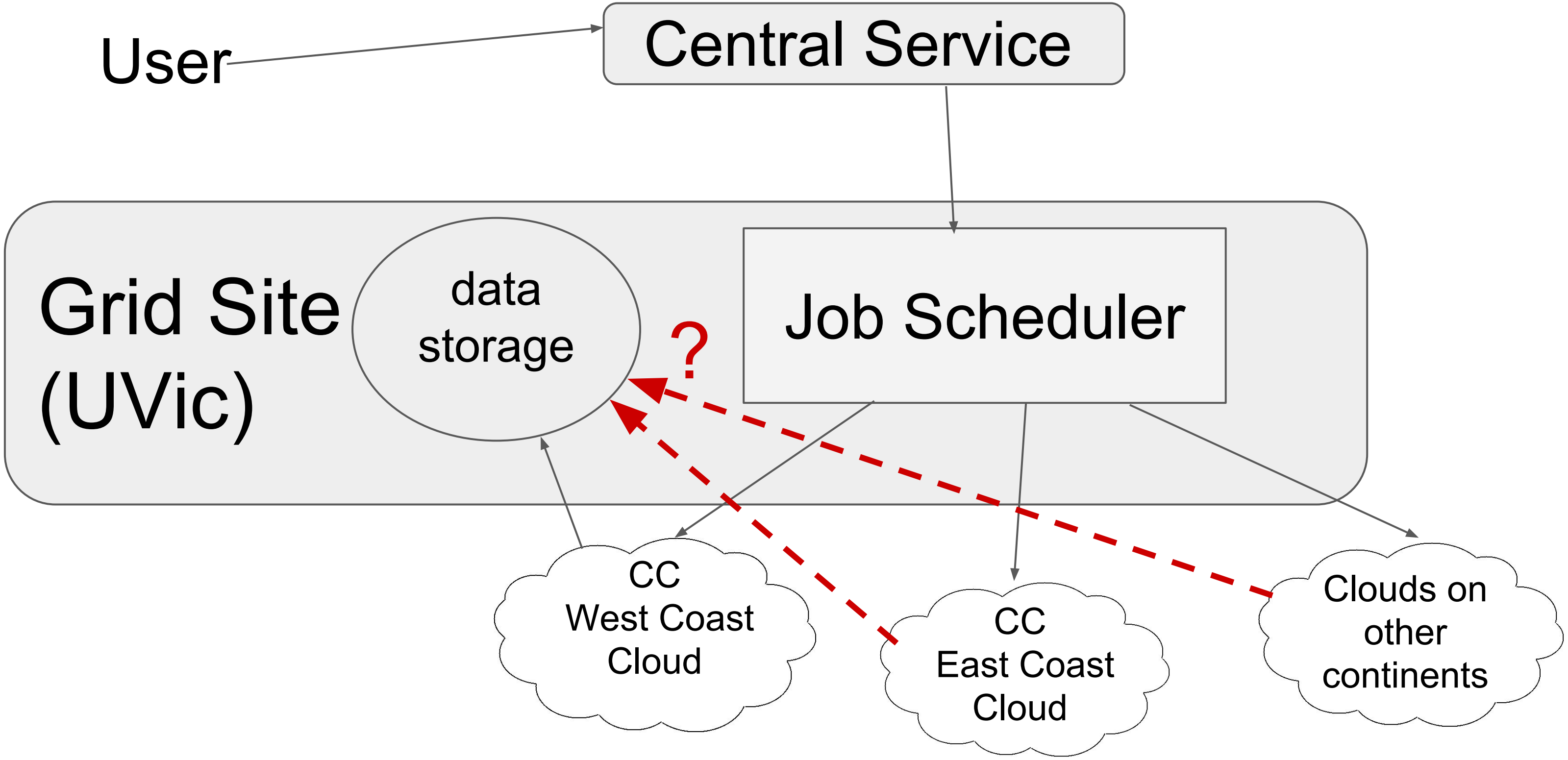
- **distributed compute works very well for us**

Data handling when the compute can be anywhere is a different story ...

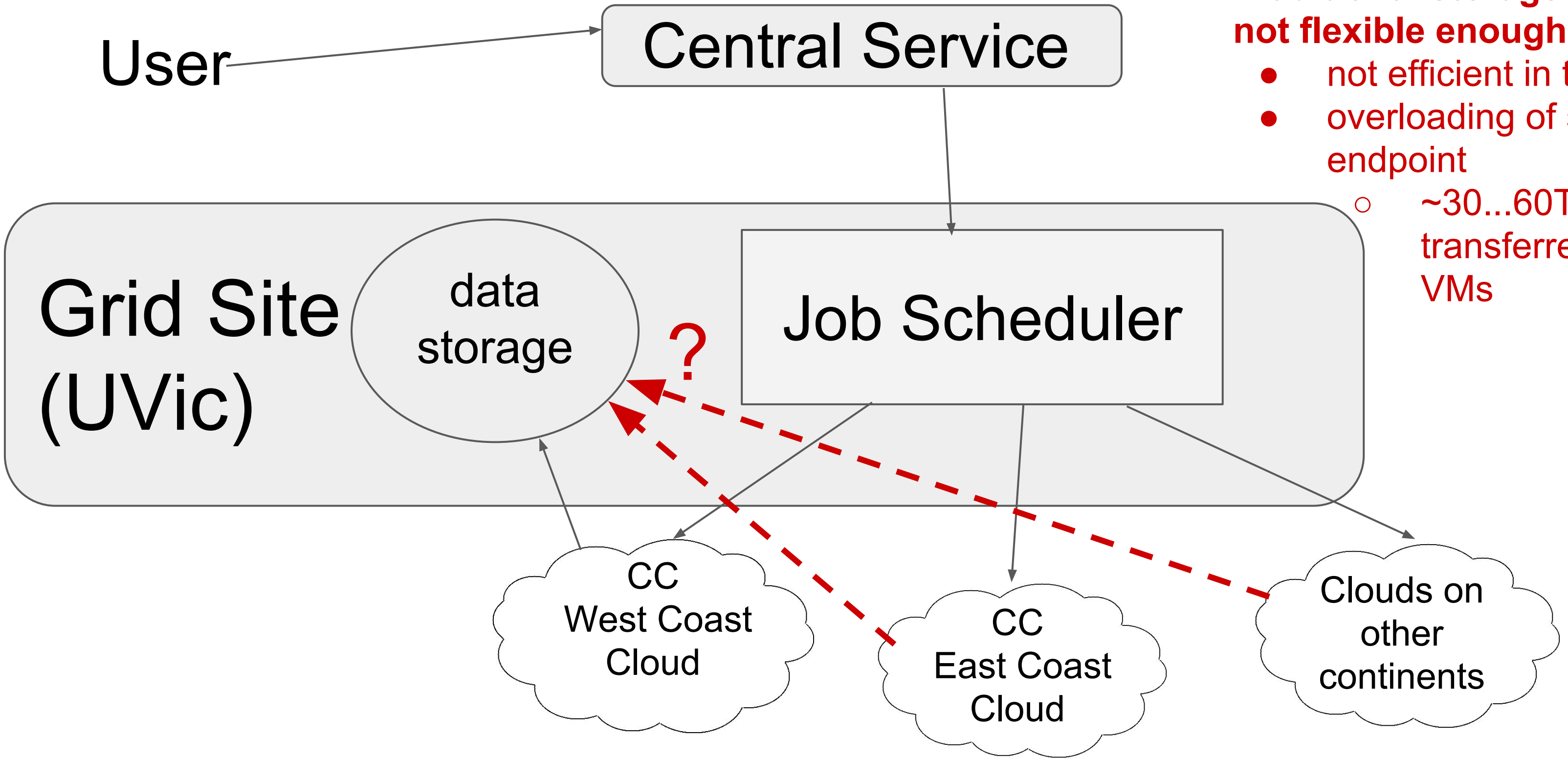
Distributed Cloud Storage

https://indico.cern.ch/event/637013/contributions/2739286

Cloud computing for the GRID



Cloud computing for the GRID

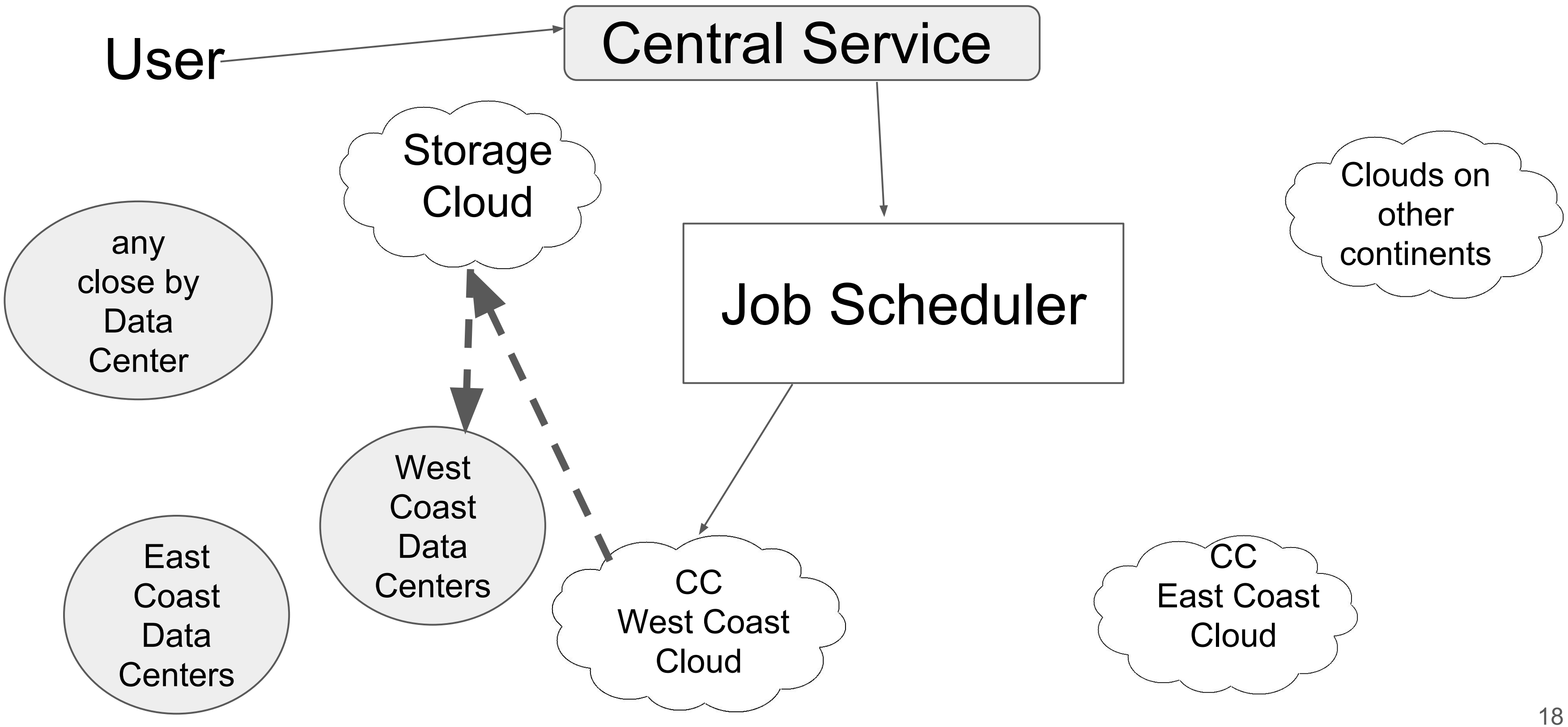


Problem:

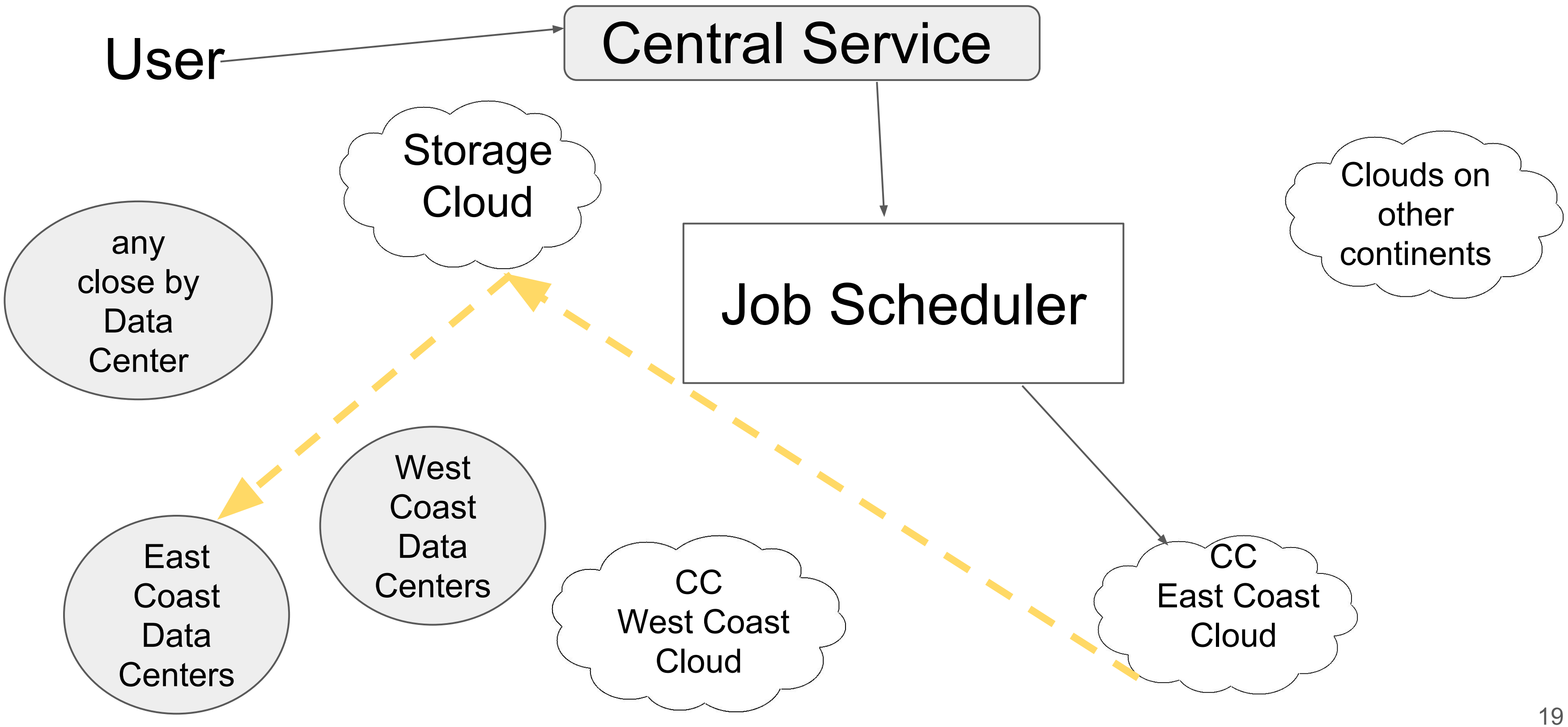
Traditional storage systems are not flexible enough!

- not efficient in that situation
- overloading of single storage endpoint
  - ~30...60TB/day transferred to/from the VMs

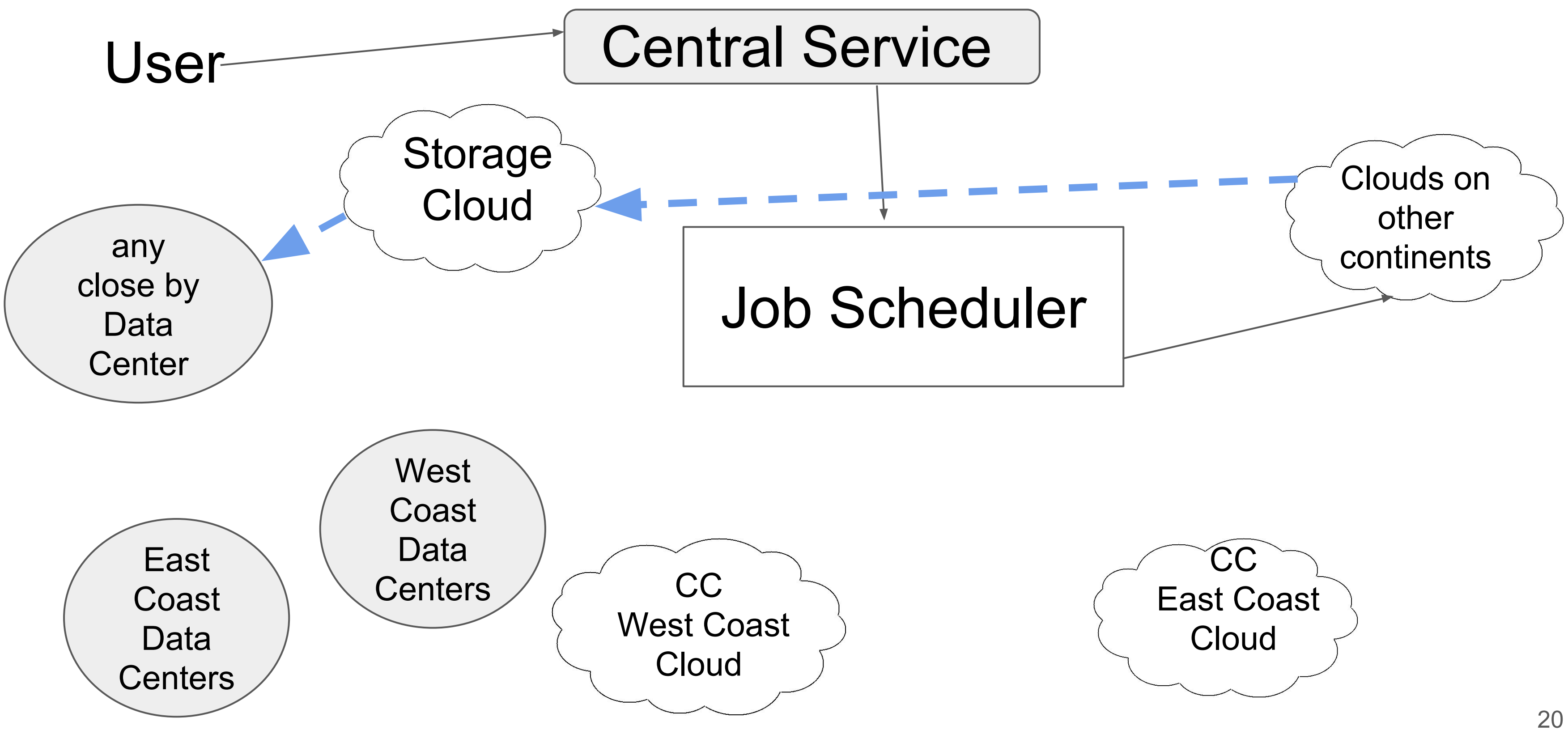
Cloud storage for the GRID



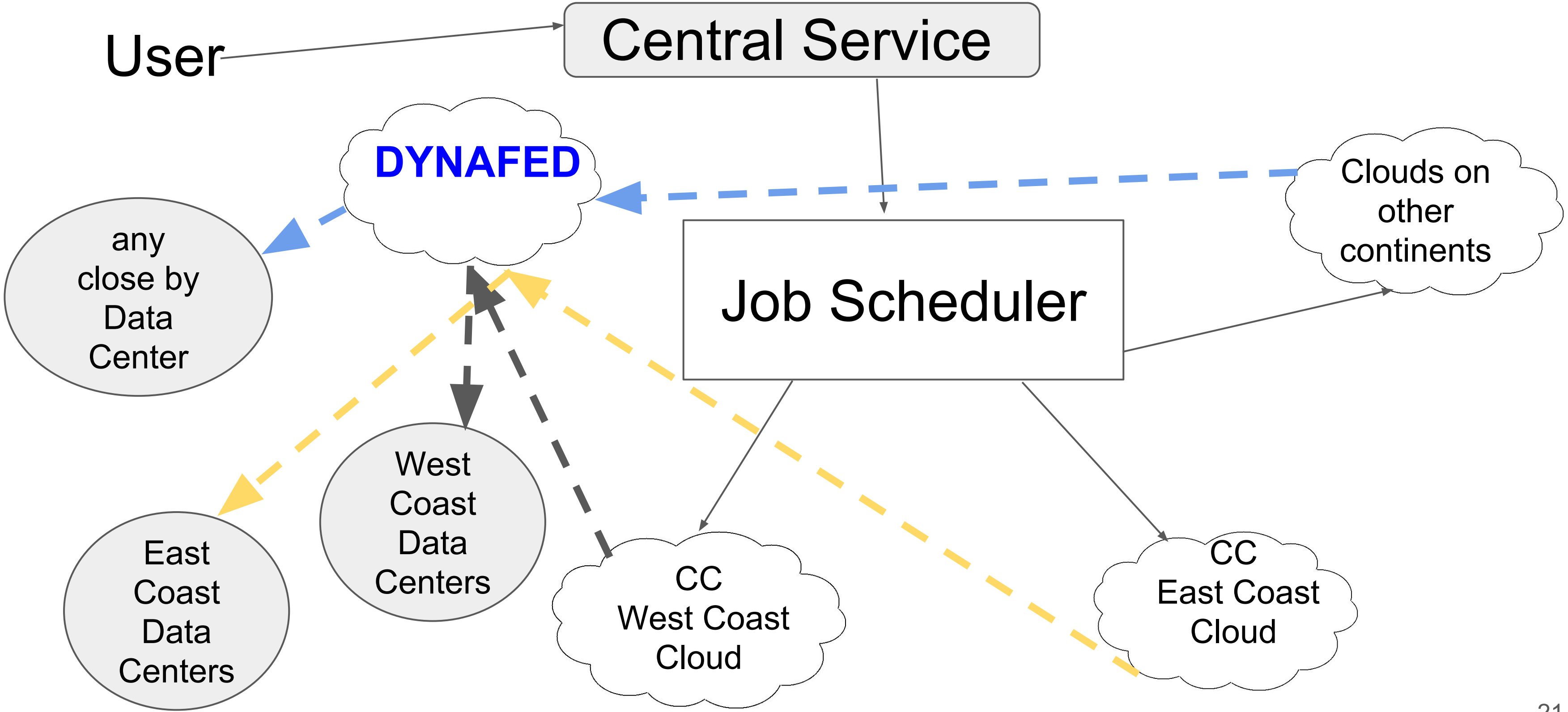
Cloud storage for the GRID



Cloud storage for the GRID



Cloud storage for the GRID



Dynafed

- developed by CERN IT
    - 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/

- redirector for a dynamic data federation
    - for data transfers, client is redirected to a storage element with the data

- access through http(s)/dav(s) protocols

- can directly access S3 and Azure based storage in addition to existing Grid storage
    - no credentials visible to the client
    - preauthorized URL with limited lifetime is used to access files on the storage

- X.509 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

Some features using Dynafed

- redirecting client to nearest site that has data or is enabled for writing data
  - uses GeoIP DB
  - in the future other characteristics could be added, like latency, bandwidth, or storage cost

- 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

- new sites can easily be added any time
  - administration of connected sites happens in Dynafed, not at a site

- gfalFS: tool to mount the whole data federation into a Linux file system
  - fuse based, but stable and reasonable fast
  - all VMs see same mount point and directory structure behind it, e.g. */mount/data/experiment/user/dir1/file1*
  - but each VM can get the data from a different endpoint when replicas across all endpoints exist
    http://heprc.blogspot.com/2017/12/mounting-federated-storage-cluster-as.html

Dynafed@Victoria HEPRC group

- running different installations
  - in production for Belle-II (through gfalFS and only for reading)
  - in testing for Atlas; expecting full production use in the next months
  - works very well, but for full production usage experiments need to change their frameworks

- behind Dynafed different kind of endpoints
  - existing Grid sites
  - own Ceph installation
  - minio based endpoints in VMs on different clouds
    - https://www.minio.io/

- multi-experiment enabled
  - same installation can be used for Atlas and Belle-II access
  - authentication and authorization controls who can access what

- want to demonstrate that this can work as a global distributed storage system
  - in the future needed when moving compute more and more to clouds and away from isolated sites
  - WLCG demonstrator project for future WLCG developments

Dynafed@Victoria HEPRC group

- running different installations
  - in production for Belle-II (through gfalFS and only for reading)
  - in testing for Atlas; expecting full production use in the next months
  - works very well, but for full production usage experiments need to change their frameworks

- behind Dynafed different kind of endpoints
  - existing Grid sites
  - own Ceph installation
  - minio based endpoints in VMs on different clouds
    - https://www.minio.io/

- multi-experiment enabled
  - same installation can be used for Atlas and Belle-II access
  - authentication and authorization controls who can access what

- want to demonstrate that this can work as a global distributed storage system
  - in the future needed when moving compute more and more to clouds and away from isolated sites
  - WLCG demonstrator project for future WLCG developments

To demonstrate that such distributed storage system can scale to a global system that can be used efficient and fault resistant, we need to have as many different endpoints as possible.

Dynafed@Victoria HEPRC group

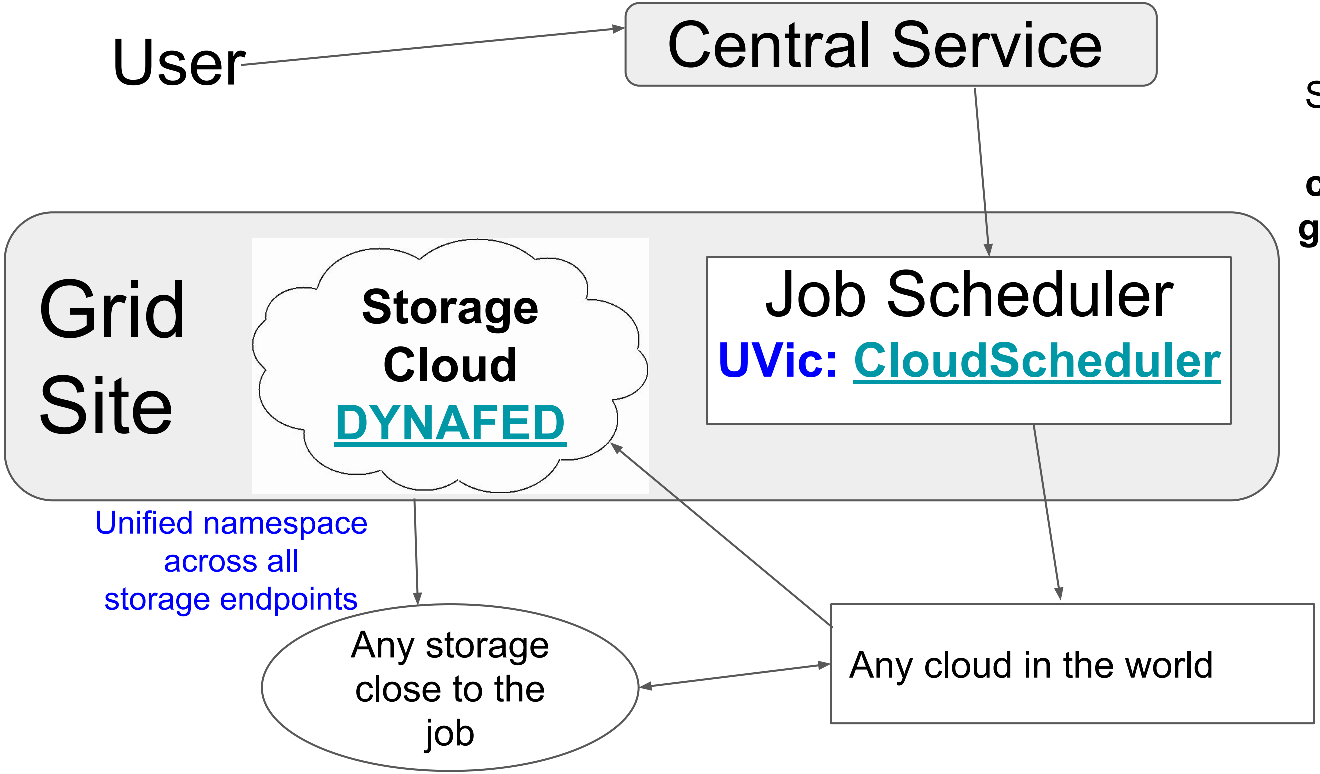
- most data comes from existing grid storage sites
- currently only our own Ceph storage in Victoria, ~15TB
- very small minio installations on the different clouds
  - running in Openstack VMs with a volume added
  - good for testing
    - but not much space and performance for large scale tests
- looking to expand to use other CEPH installations
  - at the order of 10s of TBs
  - not at a single location, but distributed across Canada
    - especially at the east coast would be good

Dynafed@Victoria HEPRC group

- most data comes from existing grid storage sites
- currently only our own Ceph storage in Victoria, ~15TB
- very small minio installations on the different clouds
    - running in Openstack VMs with a volume added
    - good for testing
        - but not much space and performance for large scale tests
- looking to expand to use other CEPH installations
    - at the order of 10s of TBs
    - not at a single location, but distributed across Canada
        - especially at the east coast would be good

Does CC operates Ceph storage and could provide buckets at different locations?

Distributed cloud-storage and cloud-compute for the GRID



Storage cloud not only **useful** for cloud jobs but **also for compute-only sites and any group that needs distributed storage with unified namespace**!

Summary

- **developed compute system that can utilize and unify different clouds and cloud types into a single infrastructure**
  - clouds hidden to the user
  - user interface is a normal batch system interface
    - HTCondor in our case
  - run successfully since many years with High Energy Physics workload
    - currently developing a new, more modern version of cloudscheduler
      - with web interface, easier multi-project use, and integration of Glint
      - in test mode right now

- **working on establishing a global distributed storage cloud based on dynafed**
  - single endpoint with fault resistant redirection to nearest storage via http(s)/dav(s)
  - need more distributed resources for testing/development and establishing such storage cloud

links:

group page : http://heprc.phys.uvic.ca dynafed: http://lcgdm.web.cern.ch/dynafed-dynamic-federation-project
group blog : https://heprc.blogspot.com
github repository: https://github.com/hep-gc
cloudscheduler: https://github.com/hep-gc/cloud-scheduler (current production version)
https://github.com/hep-gc/cloudscheduler (currently in development)
Glint : http://heprc.blogspot.com/2017/08/glint-version-2-enters-production.html
shoal : http://shoal.heprc.uvic.ca

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