Belle II and Dynafed

Ueda I.

2019.Jul.09. pre-GDB

Belle II

DIRAC

Belle II uses DIRAC

- for both workload management and for data management
- Dirac uses a **File Catalog**
  - where the replica information is stored and looked up for
  - Files existing on a SE, but without a replica entry in the File Catalog is not visible to DIRAC
    - Just federating SEs with dynafed does not make the system use files via dynafed
- Input files for Jobs
  - Dirac looks up the FC for their replicas to decide at which sites the jobs can run (**job assignment to sites**)
  - Dirac pilots look up the FC to get the replica location for **download**
- Transfers:
  - Dirac **finds the source replicas** using the FC
  - Dirac **registers the destination replicas** in the FC after successful transfers

Dynafed

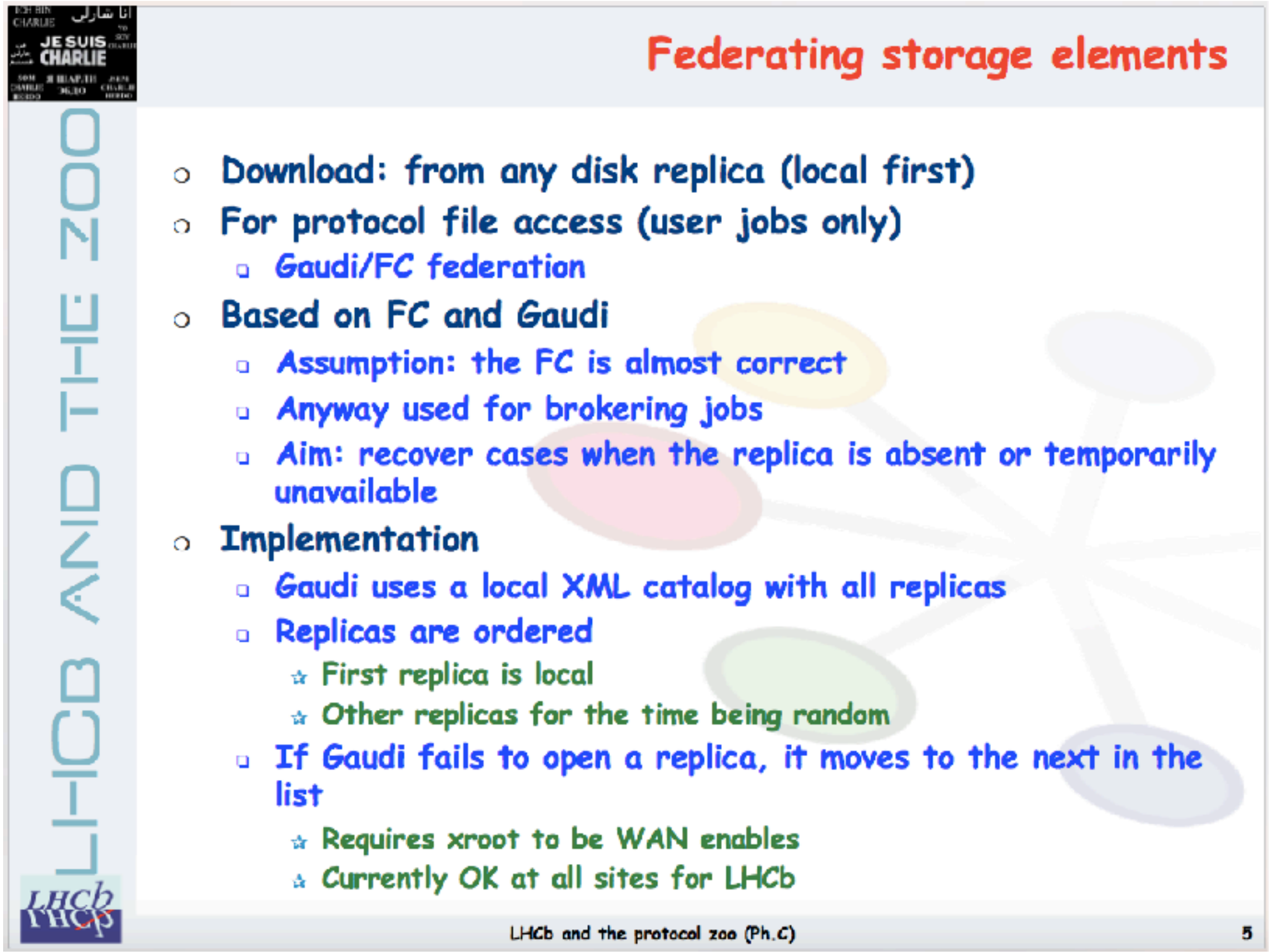
Dynafed

- is a “catalog-less” (or dynamic catalog) distributed storage by design
- does not really fit in the DIRAC workflows out of the box

Belle II

- has tried to utilize dynafed. We have instances hosted at two sites:
  - Napoli
    - federating most of the Belle II SEs
    - mostly for testing use cases
    - can be utilized for downloading files (outside of DIRAC workflows)
  - UVic
    - federating the UVic SE, the cloud storage, and some other Belle II SEs close to their cloud resources
    - now used in production with DIRAC jobs running at the “UVic” site

LHCb - Federation based on FC and Gaudi



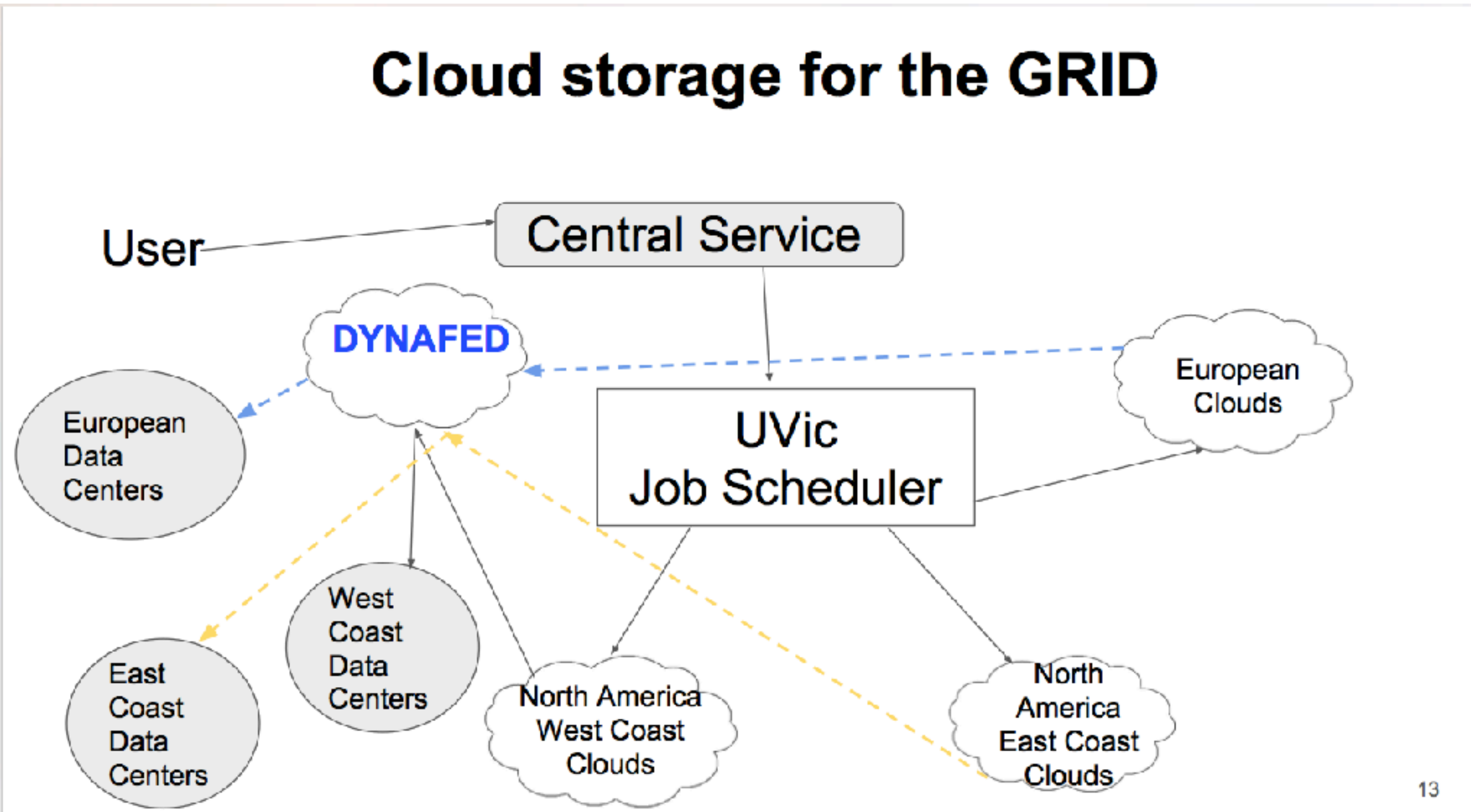
Philippe Charpentier, 13 January 2015
https://indico.cern.ch/event/319817/contributions/1697278/

Belle II and Dynafed

Belle II use cases

- Jobs to access “parent files” via the “global” dynafed at Napoli
  - Jobs go to input data (usual FC lookup and SRM access), and then access its **“parent files” hosted elsewhere via dynafed**
  - **Successfully tested a few years ago.** The workflow suspended for the first years of the experiment
- Napoli use cases
  - See Silvio’s talk
- Dynafed as a site-specific storage cloud for UVic computing clouds
  - Jobs submitted to “UVic” will run on **multiple clouds distributed over the globe**
  - Files on UVic SE are replicated by the site admin to the cloud storage behind the UVic dynafed
    - Some other SEs are also behind the dynafed, but they may not be hosting the same files
  - **Jobs access the input files via dynafed, to get the closest replica**
- “Fail-over” access
  - under consideration

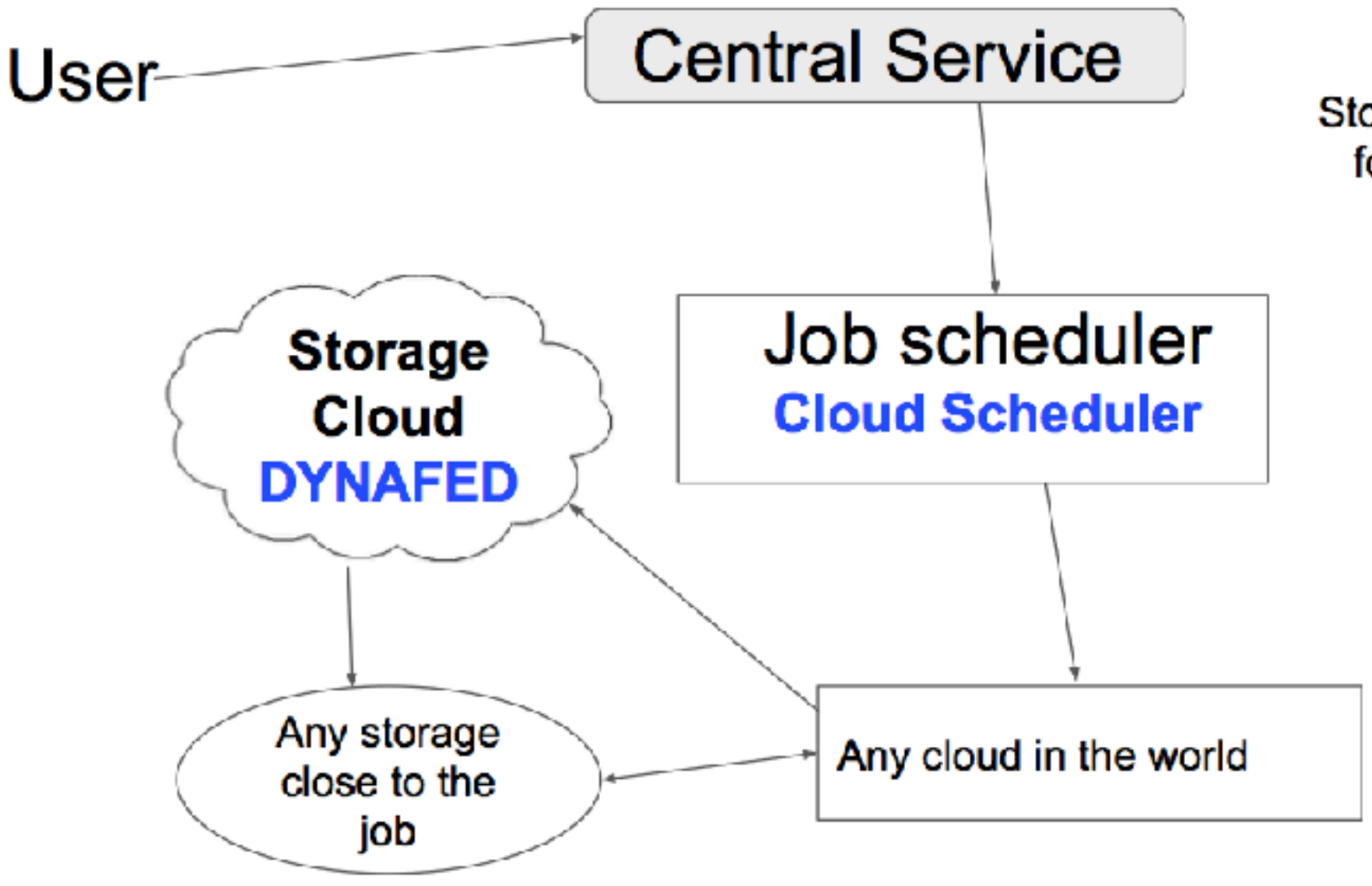
Storage cloud for UVic computing cloud



Marcus Ebert, 9 October 2017

Storage cloud for UVic computing cloud

Cloud storage for the GRID



Storage cloud not only **useful** for cloud jobs but **also for compute-only sites**!

14

Marcus Ebert, 9 October 2017

dynafed

Dynafed as a SE?

Read

- For “reading” use cases, it would work nicely

Write

- For “writing” use cases, we need to sort out **checksum issues**

Replicate / Move

- Certain files need to be hosted on multiple SEs
  - **Replication by accessing the individual SEs**, not via dynafed, at least for destination?
- Some files need to be moved from a SE to the other
  - **Choosing the dynafed as source of ‘move’ operation would be problematic** (see below “Deletion”)
- **Complication in the File Catalog registration** (next slide)

Deletion

- Deleting a file via dynafed == removing all the replicas
  - ok for getting rid of the file totally from the Grid
  - **problematic when reducing the number of replicas** (see later slides)

Dynafed and File Catalog

File Catalog Registration

- If we register each SE as replica location, the system would not see the files are available on the dynafed
- Dynafed needs to be registered in the FC as the file location, in order for the system to access files via dynafed?

Thoughts

- Files to be uploaded to SEs, but registered in FC with `location=dynafed`
  - **Problems in replication and deletion**
- Files to be uploaded to SEs, and registered in FC with `location=theSE,dynafed`
  - A special treatment needs to be implemented in the system for the dynafed replicas (because they are “dynamic” == not static replicas)
  - **It will be complicated and may cause confusions**
- Files to be uploaded to SEs, and registered in FC with `location=theSE`
  - **dynafed just another access method for the SE**
- Or, access files **not via DIRAC** (eg. directly from the Belle II software)

Dynafed and Deletion

Deleting a file via dynafed

- better be avoided in case of “global” dynafed
  - if some replicas are to be kept
  - if deletion is to be done in bulk
  - not feasible in case no replica entry for dynafed in FC
- can be done in case of “site-specific” dynafed

Deleting a file directly with SE

- requires a replica catalog entry for the SE
- In case there is a replica entry for a dynafed in addition to that for the SEs
  - **The system should not count the “dynafed entry” as a real replica**
  - **The “dynafed entry” should be removed when the last replica is removed**

summary

Currently possible usage

In Belle II

Global dynafed

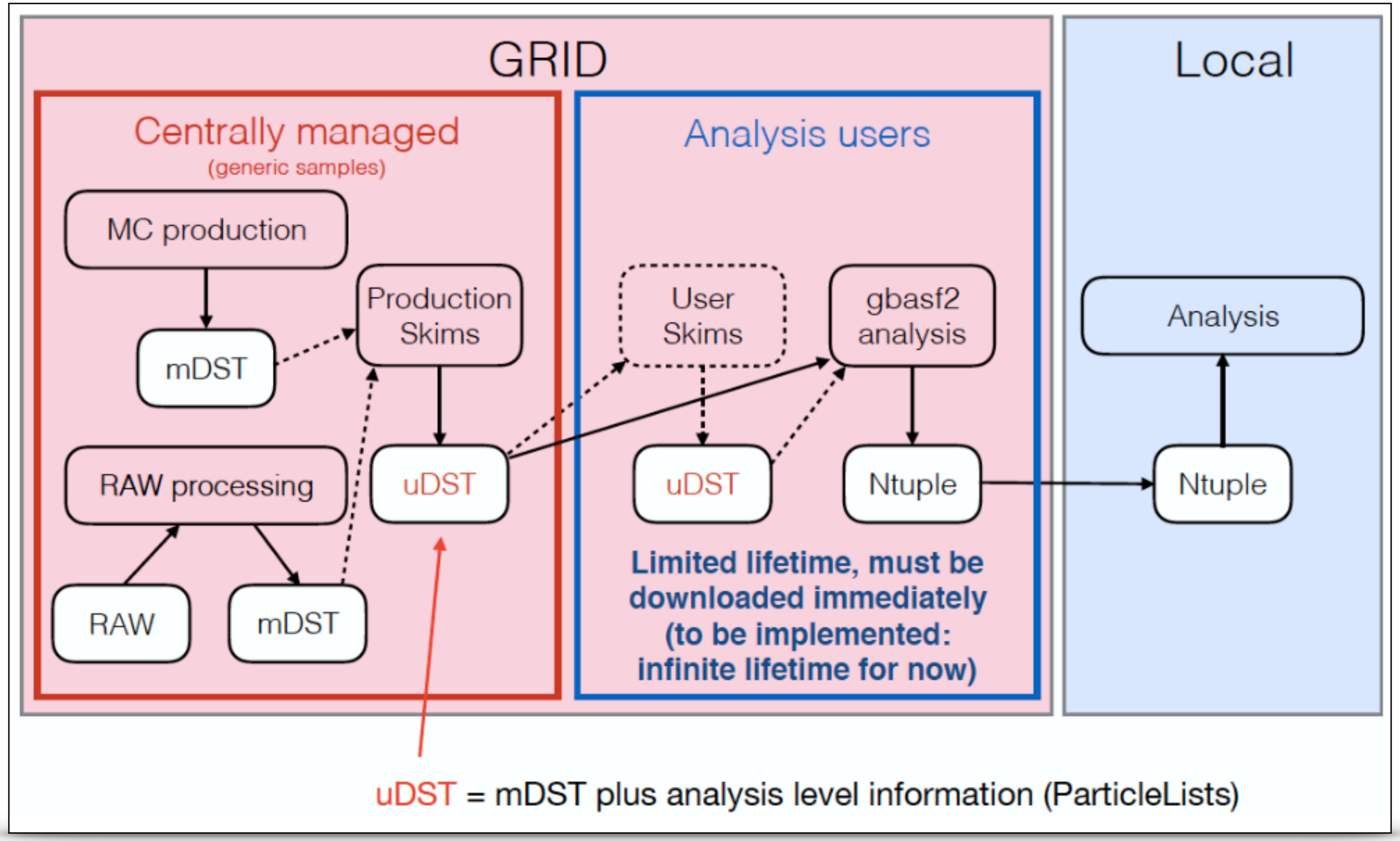
- aggregating the files on “all” the SEs (or most of them)
- **No file catalog entries** for the dynafed
- **Read-only via dynafed:** Some SEs to be defined with
  - write/delete access against the SEs
  - read access against the dynafed — to allow fail-over to the other SEs
- Jobs / end-users (**world-wide**) to access dynafed **not** via DIRAC

Site-specific dynafed

- aggregating only the replicas for a site
- should look like a site SE, with **file catalog entries only for the site**
  - read, write (w/o checksum verification), delete, …
- **Non-permanent storage** (because of the lack of checksum verification)
  - i.e. job output written onto the dynafed and moved to other permanent SE
- Jobs **for the site** to access dynafed **via DIRAC** with FC lookup

extra slides

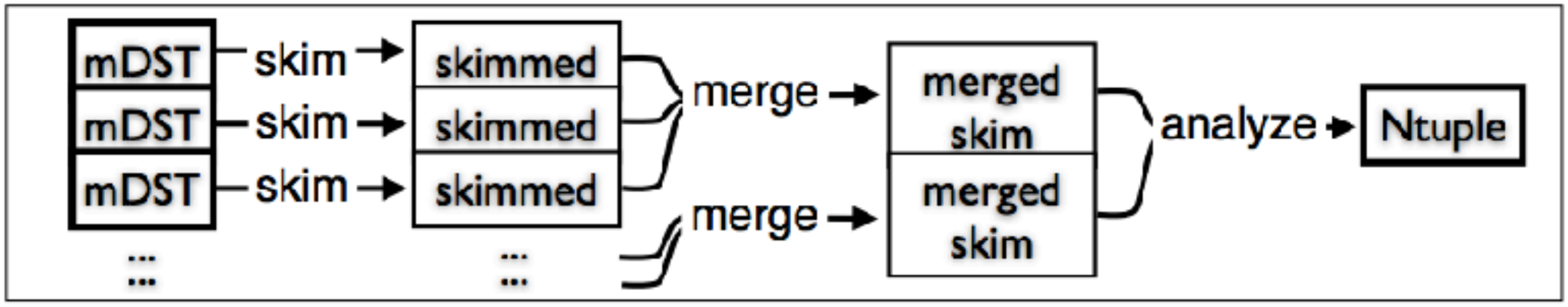
Current Belle II Analysis Model



Belle II analysis model = remote file access

Why?

- The large data volume
- The limited disk resources
- Need to minimize the data types store
  - Not to store intermediate data
- Analysis work flow
  - mDST ==> skim (event selection ~1%) ==> retrieve variables for physics (event size = 1/M) ==> Ntuple

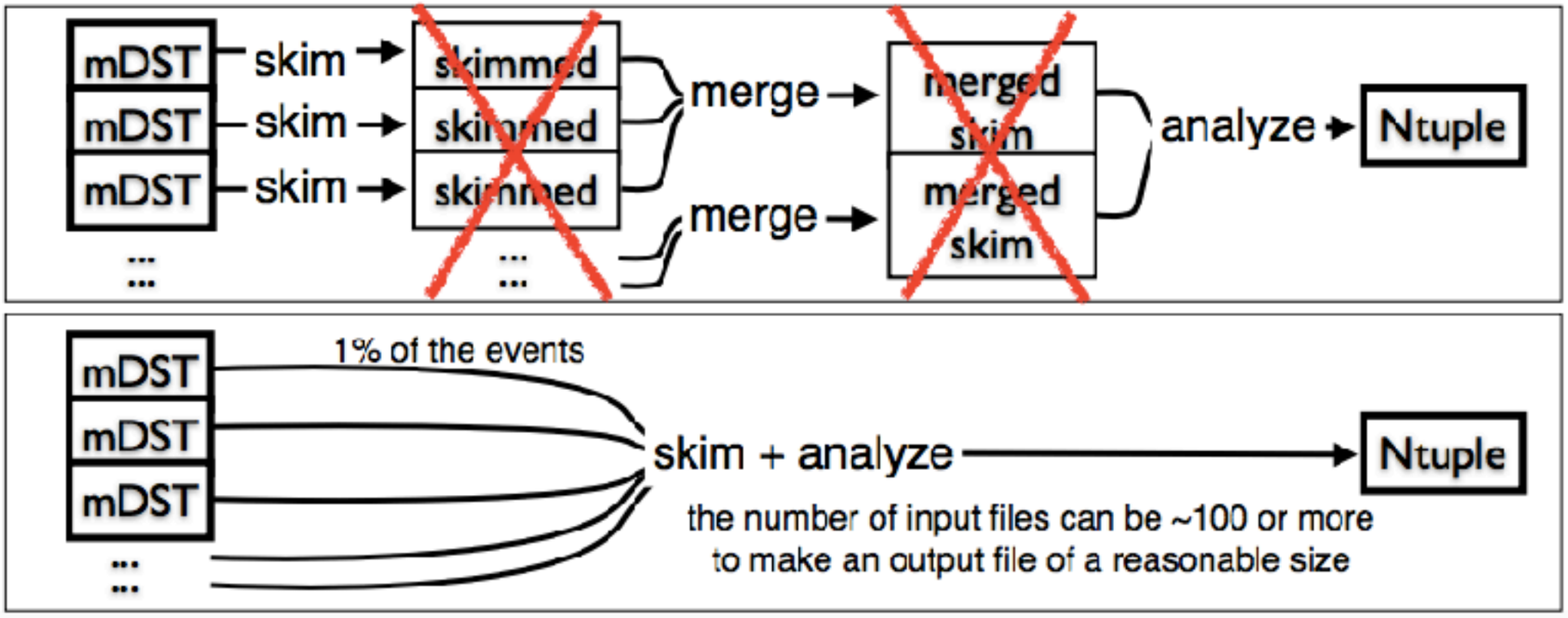


- Intermediate data not stored

Belle II analysis model = remote file access

Why?

- The large data volume
- The limited disk resources
- Need to minimize the data types store
  - Not to store intermediate data
- Analysis work flow
  - mDST ==> skim (event selection ~1%) ==> retrieve variables for physics (event size = 1/M) ==> Ntuple



Index file and remote file access

Analysis job reads events from remote files