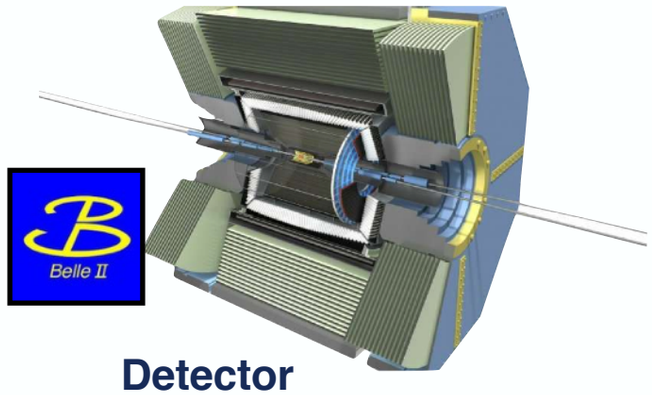
RAW data registration: Belle II

Michel H. Villanueva
The University of Mississippi

I. Ueda
KEK IPNS

9th DIRAC Users' Workshop
May 15, 2019

Online - Offline - Grid



Detector

DAQ units

Raw (SROOT)

Online disk

rsync (~ 2 GB/s)

Temporal
Grid storage

Raw (ROOT)

Permanent
Grid storage

BNL Raw
data center

TMP SE

Frontend
server

SROOT/ROOT
conversion

Permanent
Grid storage

LFC
AMGA

metadata

mDST

mDST

KEK Raw
data center

File / Metadata
catalogs

Global tag

Offline
Conditions DB

Regional data centers

Data
processing

DIRAC
THE INTERWARE

BelleDIRAC

DIRAC
THE INTERWARE

BelleRawDIRAC

THE UNIVERSITY of MISSISSIPPI

BelleRawDIRAC

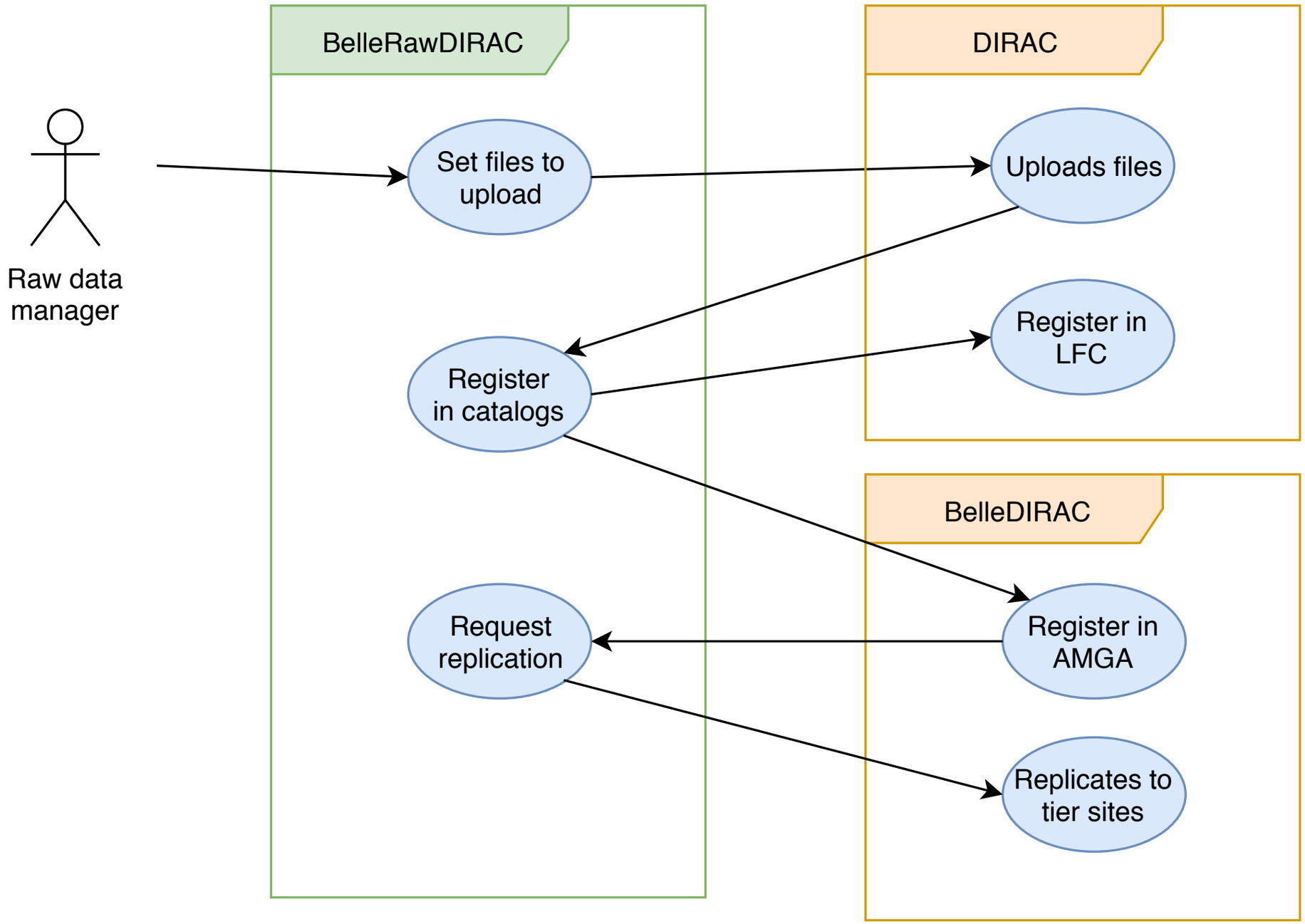
- To process raw data in the most reliable way, we are developing an extension of DIRAC which:
  - Runs on a dedicated server.
  - It is separated from the production activities (**BelleDIRAC**).
- **BelleRawDIRAC** is a DIRAC extension dedicated to the upload, registration and replication of raw data files.
- Tasks:
  - Preparation of files to be uploaded (metadata embedded, LFN, GUID, checksum).
  - Uploading of files into a Storage Element (temporal space).
  - Registration of files in file catalog and metadata catalog.
  - Replication using the Belle II Distributed Data Management system (DDM).

BelleRawDIRAC

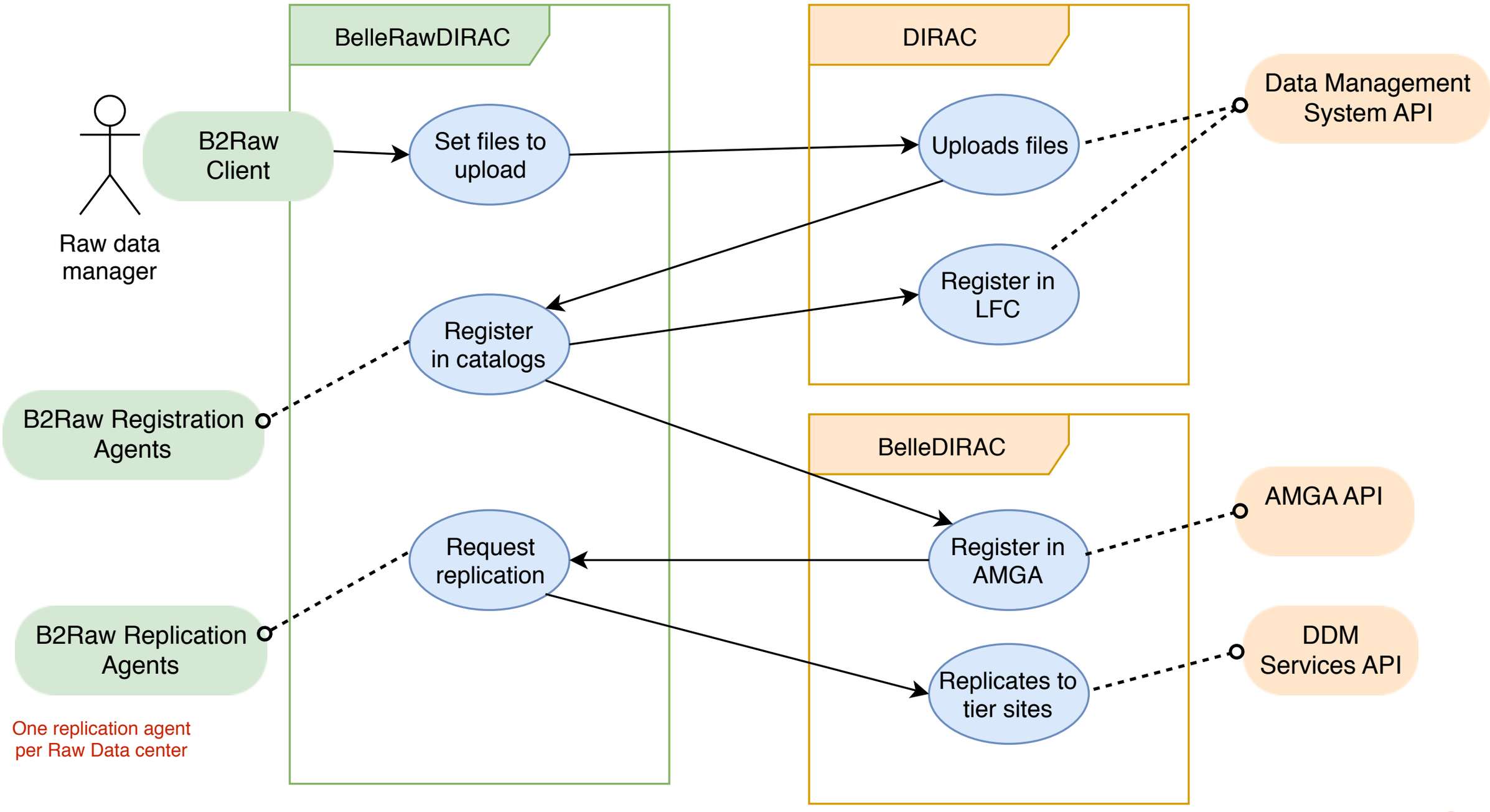
- Currently, **BelleRawDIRAC** consists of one system, `B2RawDataManagement`, containing:
  - One database: `B2RawRegistrationDB`.
    Tables: `Files, Datablocks`[1]`, Datasets, GlobalTag`.
  - One service: `B2RawDataRegistration`.
  - Several agents:
    `B2RawRegistrationAgent`,
    `B2RawReplicationAgent`,
    `B2RawReliabilityAgent`,
    etc.
  - Client and command line tools.

[1]Datablock: unit of data management (https://indico.cern.ch/event/477578/contributions/2143193/)

Use case



Use case



THE UNIVERSITY of MISSISSIPPI

Workflow

- The raw data manager (human) provides a list of files to be processed by **BelleRawDIRAC** by a command line tool.
- The client add the list of files into the `B2RawRegistrationDB`.
- Three agents work in the registration process.
  - `B2RawLocationAgent`: Reads the embedded metadata, builds the LFN, gets the global tag ID from the conditions DB, assigns a datablock and a dataset per file.
  - `B2RawUploadAgent`: Uploads the file into the temporal Grid storage.
  - `B2RawRegistrationAgent`: Register the file into the file catalog. Updates information of file, datablock and dataset into the metadata catalog.
- An instance of the module `B2ReplicationAgent` runs per permanent Grid storage (currently two), triggering the replication by the DDM system.

B2RawRegistrationDB

Table ‘Files’	
Field	Type
file_id	int unsigned
file_path	varchar(255)
lfn	varchar(255)
guid	varchar(100)
size	int
checksum	varchar(100)
checksum_type	varchar(100)
status	varchar(100)
minor_status	varchar(100)
ToTier0	varchar(32)
ToTier1	varchar(32)
dataset_id	int
datablock_id	int
metadata	text
attempt	int
error	varchar(255)
last_update	timestamp
entry_date	timestamp

Table ‘Datablocks’	
Field	Type
datablock_id	int unsigned
datablock	varchar(255)
n_files	int
dataset_id	int
status	varchar(100)
rep_doID_tier0	int
rep_doID_tier1	int
del_doID	int
attempt	int
error	varchar(255)
last_update	timestamp
entry_date	timestamp

I Monitoring

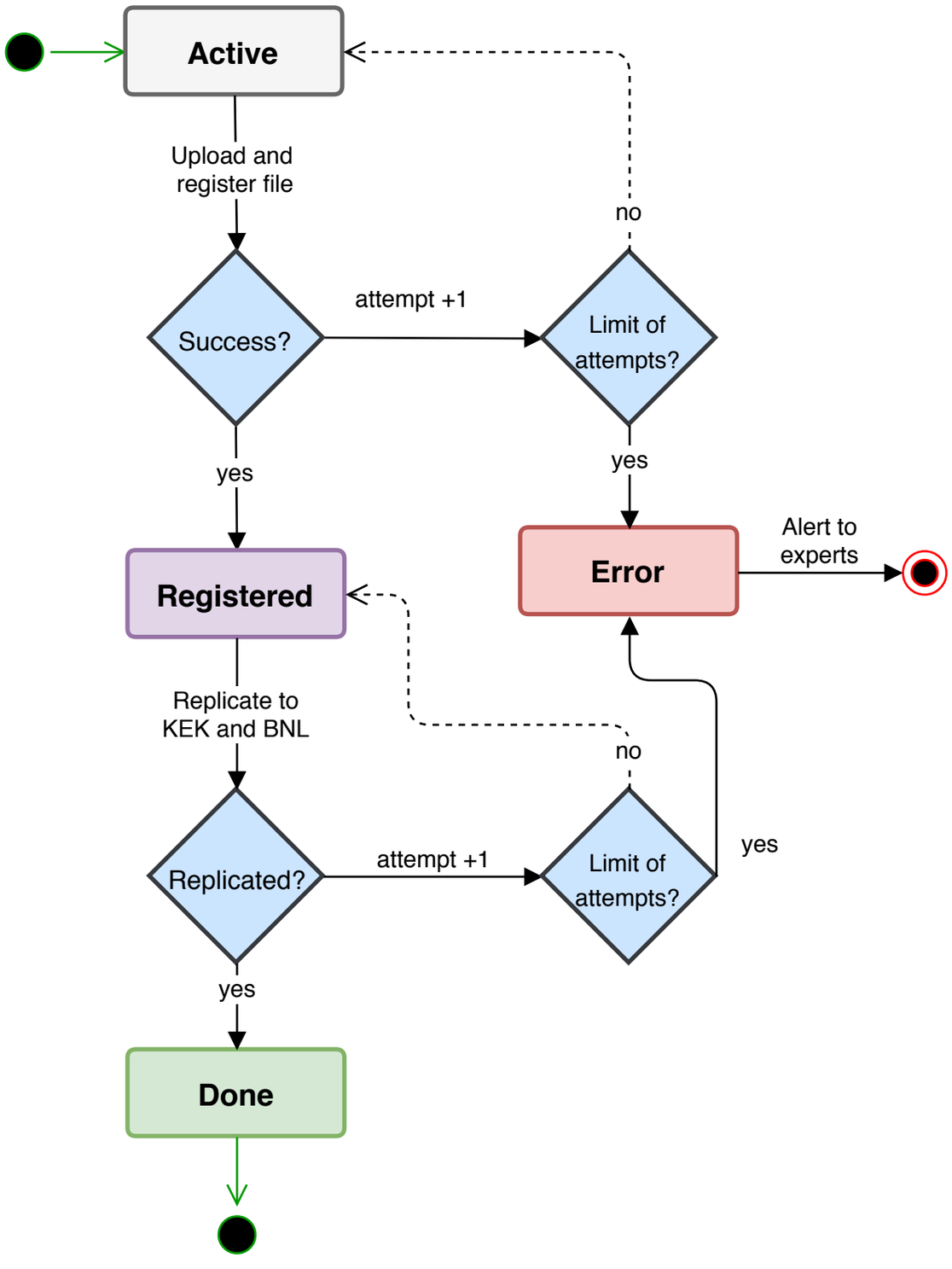
I Metadata catalog

I Track issues

I Operation ID (DDM)

THE UNIVERSITY of MISSISSIPPI

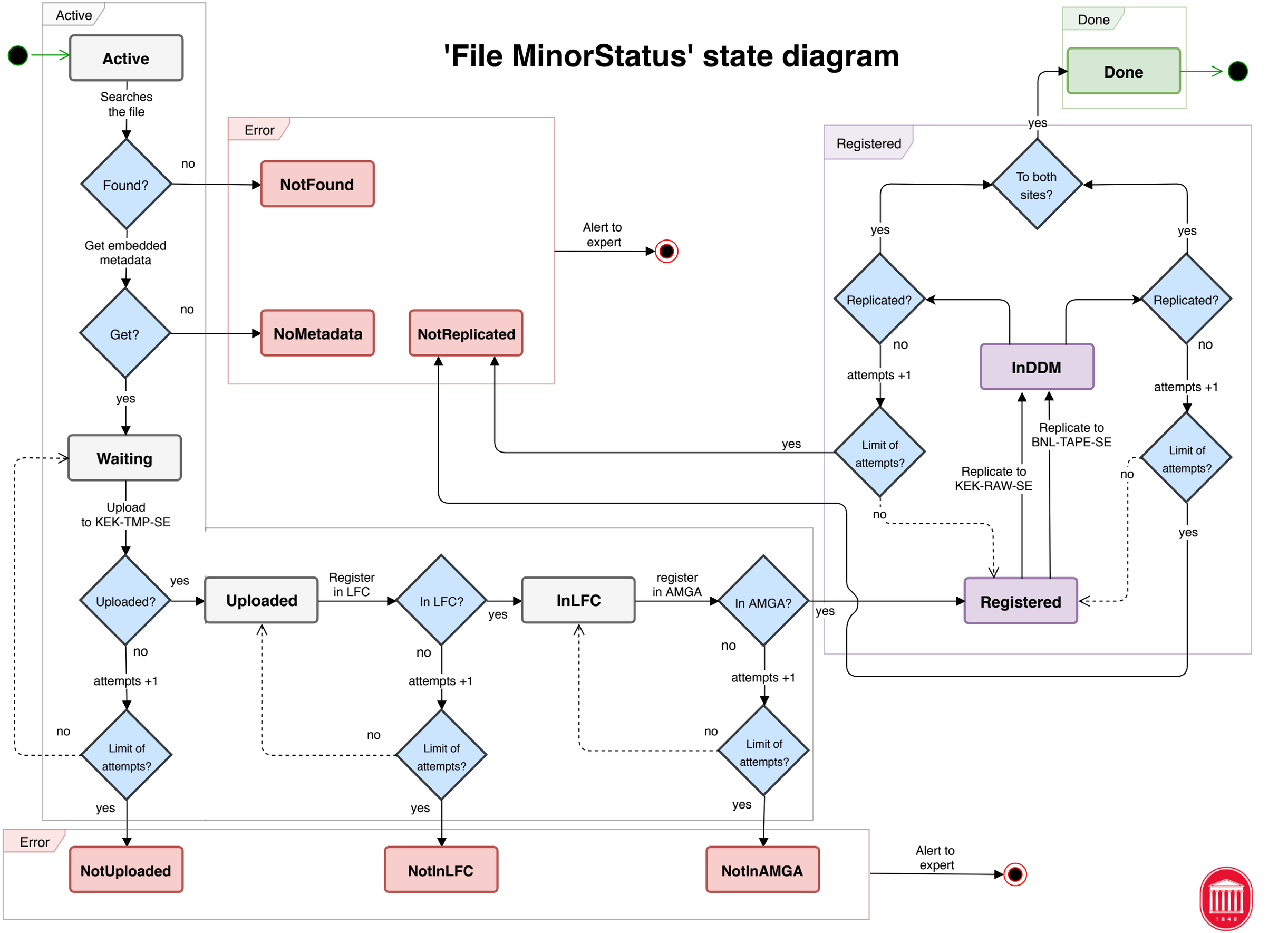
‘File status’ state diagram



- ‘Status’ gives information to the agents of which action is required in the file.
- It also gives information if some error occurs in the process.
- The number of attempts is set in DIRAC cfg.

THE UNIVERSITY of MISSISSIPPI

'File MinorStatus' state diagram



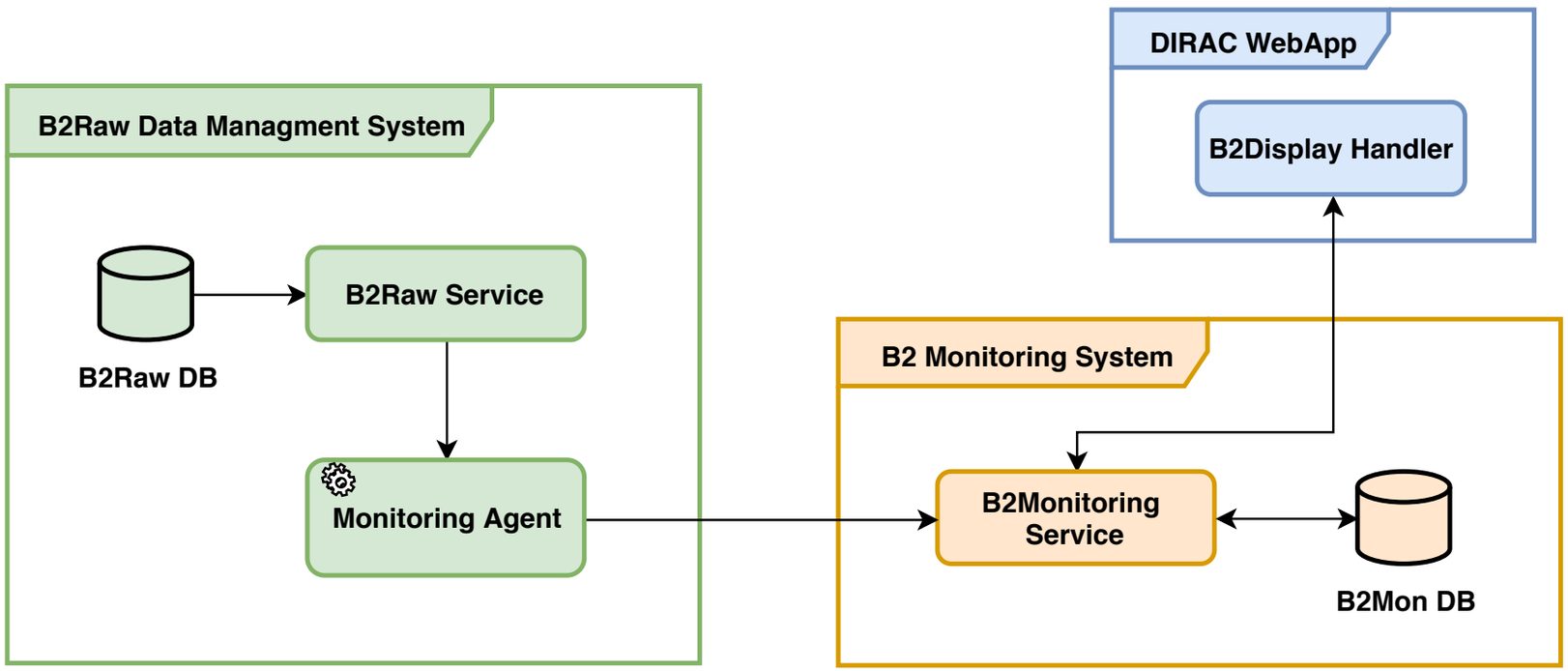
THE UNIVERSITY of MISSISSIPPI

Reliability

- Per **datablock**, we retrieve the replication ID and the status from DDM system.
- **Per file**, we look into the tasks of the DDM system. Two flags are used in the table `Files` to keep stored such status (`ToTier0` and `ToTier1`).
- An agent takes care of reliability in registration and replication.
  - `B2RawReliabilityAgent`:
    - Per file, checks that the replication to both permanent Grid storage was successful. Confirmation using checksum.
    - Confirms that all the files in a datablock were replicated.
    - Per dataset and datablock, compares the metadata in the file and metadata catalogs (as number of files, size, etc.)
    - If something is wrong, sets problematic files/datablocks to ‘Error’ status and stores an error message.

Monitoring

- An agents sends information to B2Monitoring, aiming for data production shifters taking care of the raw processing.
  - `B2StatusMonitoringAgent`: Sends information of the files currently being processed to the B2Monitoring DB.



- Command line tools provides information for computing experts looking for issues.
- Accounting, WebApp, adicional command line tools are under development.

Client tools

- Interaction between the raw data manager, experts and the system. As a few examples:

- `b2dirac-raw-setToRegister` :

```
$ b2dirac-raw-setToRegister -v --metadata-file metadata.json r03100.dat
  RawDataRegistrationClient: 3 files successfuly registered to be uploaded.
```

- `b2dirac-raw-summary` :

```
$ b2dirac-raw-summary
  ------------------------------------------------------------------------
  Exp num      Active      Registered    Done        Canceled    Error
  ------------------------------------------------------------------------
  e0002        0           0             1022        0           0
  e0003        0           0             28171       0           0
  e0005        0           40883         252         0           0
  e0006        0           112999        3984        0           148
  e0007        0           70458         561         0           0
```

- `b2dirac-raw-getErrors` :

```
  ------------------------------------------------------------------------------------------------------------------------
  File                                                    | Last Update          | Error message
  ------------------------------------------------------------------------------------------------------------------------
  /ghi/fs01/belle2/bdata/Data/Raw/e0006/r00085/sub00      |                      |
  cosmic.0006.00085.HLT1.f00053.root                      | 2019-05-02 08:43:24  | Transfer to Tier1 stalled.
  /ghi/fs01/belle2/bdata/Data/Raw/e0006/r00085/sub00      |                      |
  cosmic.0006.00085.HLT1.f00070.root                      | 2019-05-02 08:43:24  | Transfer to Tier1 stalled.
  /ghi/fs01/belle2/bdata/Data/Raw/e0006/r00085/sub00      |                      |
  cosmic.0006.00085.HLT1.f00208.root                      | 2019-05-02 08:43:24  | Transfer to Tier1 stalled.
  /ghi/fs01/belle2/bdata/Data/Raw/e0006/r00085/sub00      |                      |
```

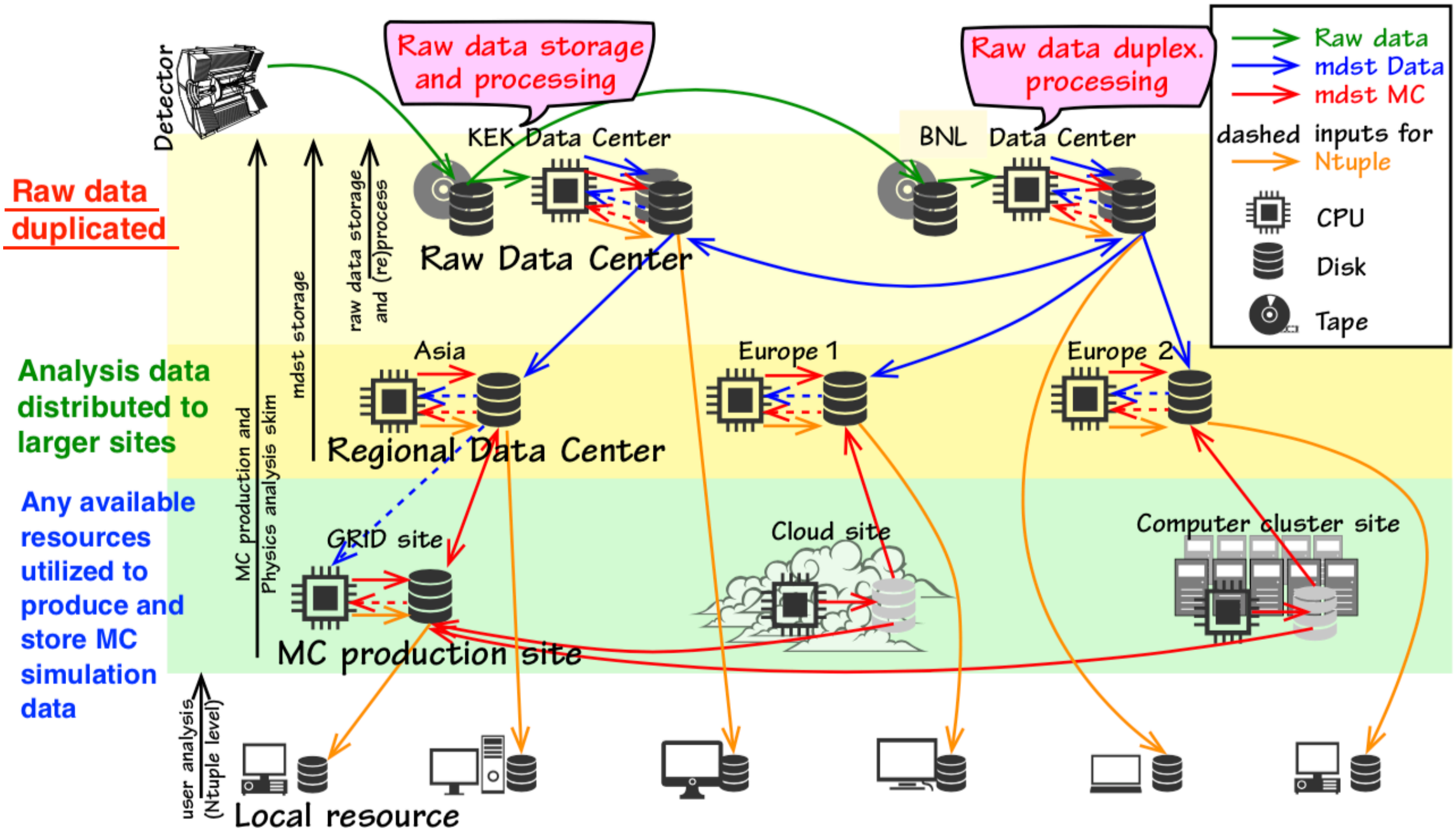
Summary

- Currently, raw data produced by the Belle II experiment is being uploaded, registered and replicated with BelleRawDIRAC.
- Client tools allows the submission of files to be treated by the raw data management system.
- Agents of the system performs all the required tasks and the reliability checks.
- Implementation of the DIRAC Accounting system will provide performance information. It is important to identify bottlenecks and points of failure.
- Monitoring tools are required. Services to access information, in order to provide monitoring plots to data production shifters via the B2Monitoring system.
- There is plenty of room for improvements.

Thank you

Backup

Belle II computing model



Belle II colleagues distributed world-wide

Definition of Datablock

Data Management Block

Datasets

- Belle II produces various types of MC data
  - Organised as “datasets” (defined as a part of LFN path)
- “Runs” can also be considered as “datasets”

Data Management Blocks

- Clustering files of the same MC type onto the same SE, to some extent, would ease some workflows
  - multiple input of the same type — merge, analysis, …
  - possible data management at directory-level
- A dataset can contains millions of files — too many as a unit of data management
- “Data block” as a unit of data management
  - max 1000 files as initial implementation, so far so good. May tune with experiences
- “Dataset” is the unit of production, but files are organised in “data blocks”
  - Some system impelementation based on “data blocks”
- A subdirectory under “dataset” path: `/belle/…dataset…name…/subNN/files`

Belle II report - Ueda I. - The 6th DIRAC User Workshop 5

Belle II DDM

Distributed Data Management System (Belle II DDM)

- does not use TransformationSystem
- manages “datablocks”
- submits file transfers to RMS
- deletes files by itself, not via RMS
  - to avoid overload on LFC and to control “priorities”

TS could have been the solution, as in LHCb
Initially tried, but the developers abandoned the idea

Currently directly calling lcg/gfal2.
Should use DIRAC APIs, with extensions where needed

Use cases

- To gather output files to “primary” SEs by “data block”
  - move == replicate by RMS + delete source by itself
- To distribute products over the grid by “data block” (yet being implemented…)

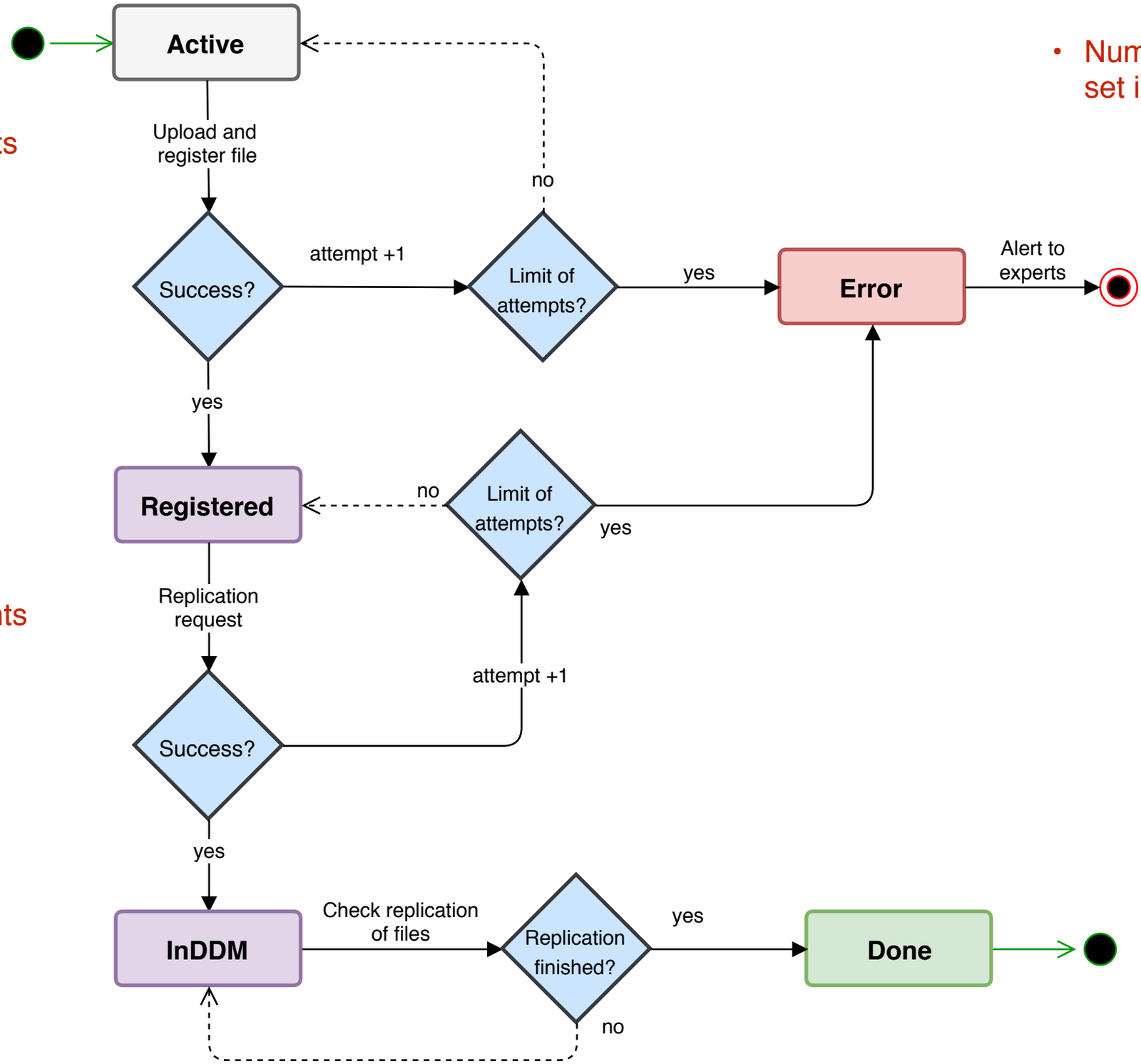
Developed by PNNL team (M. Schram, V. Bansa, et al.)

- The responsibility has moved from PNNL to BNL… (A. Undrus, S. Padolski)

More details discussed in the “Distributed Data Management” session

‘Datablock status’ state diagram

Registration agents

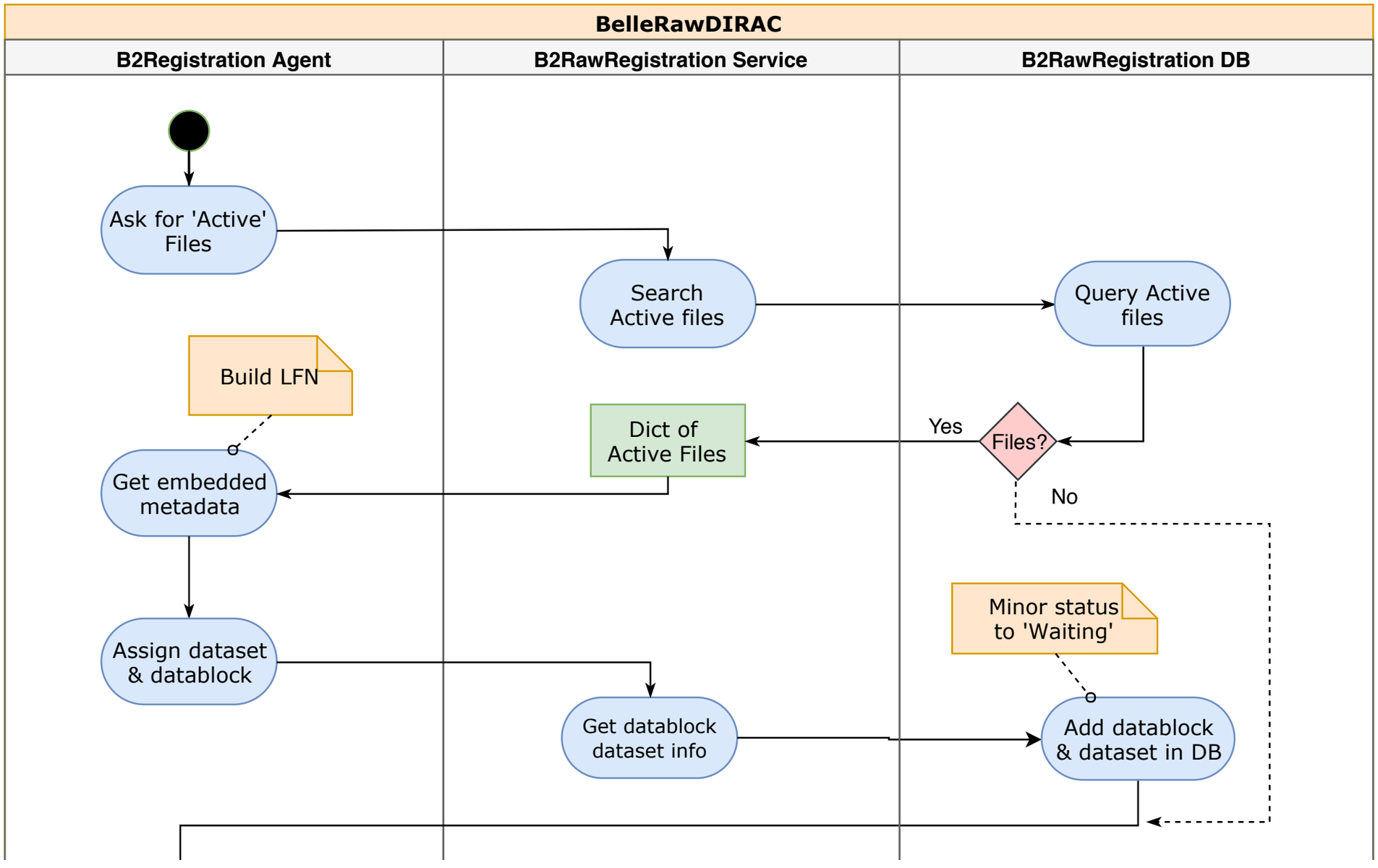


- Number of attempts set in DIRAC cfg

Replication agents

THE UNIVERSITY of MISSISSIPPI

B2RawLocation agent



'Locate' the files, which means:

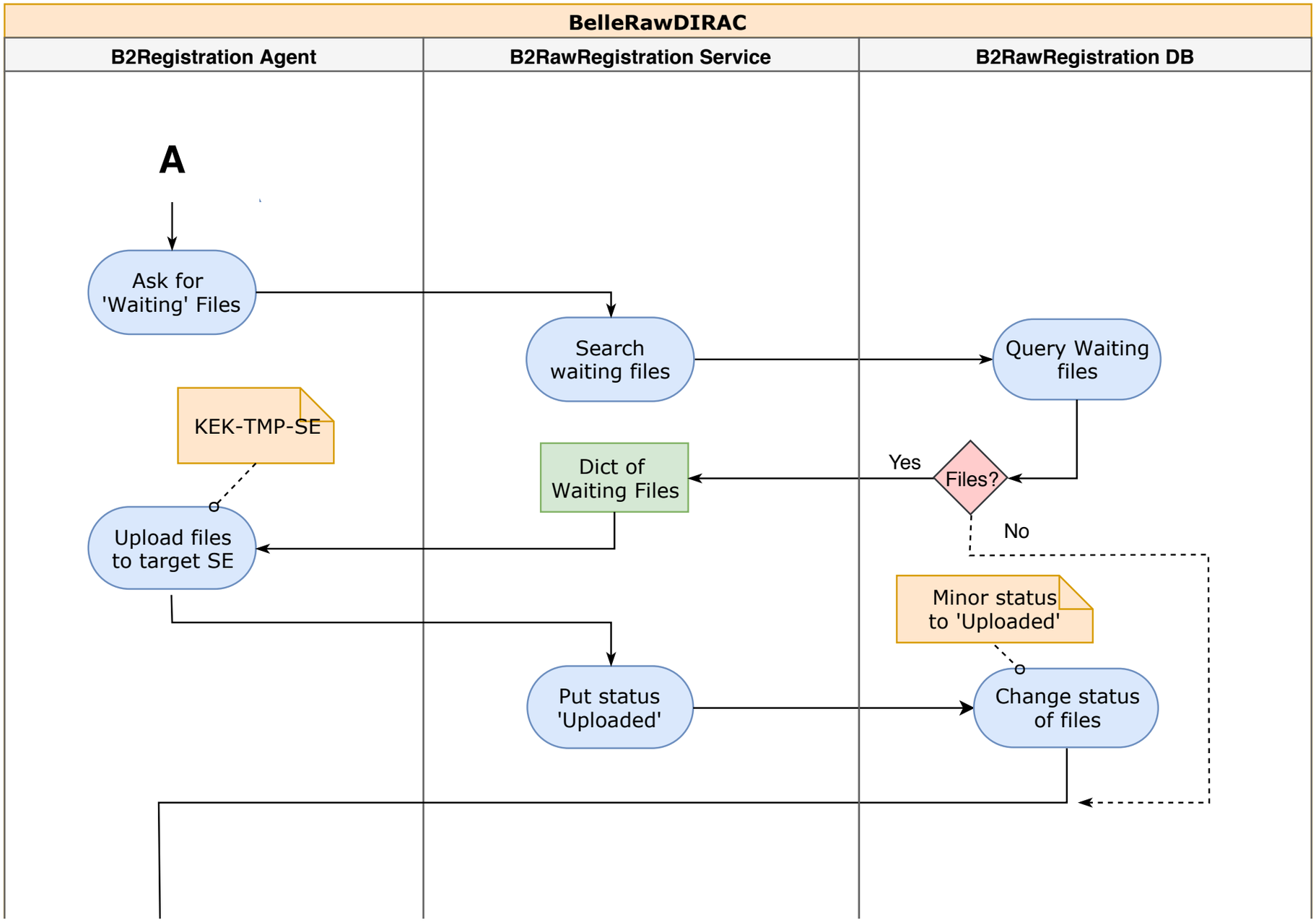
1) Get metadata.

2) Build LFN.

3) Assign dataset and datablock.

THE UNIVERSITY of MISSISSIPPI

B2RawUpload agent



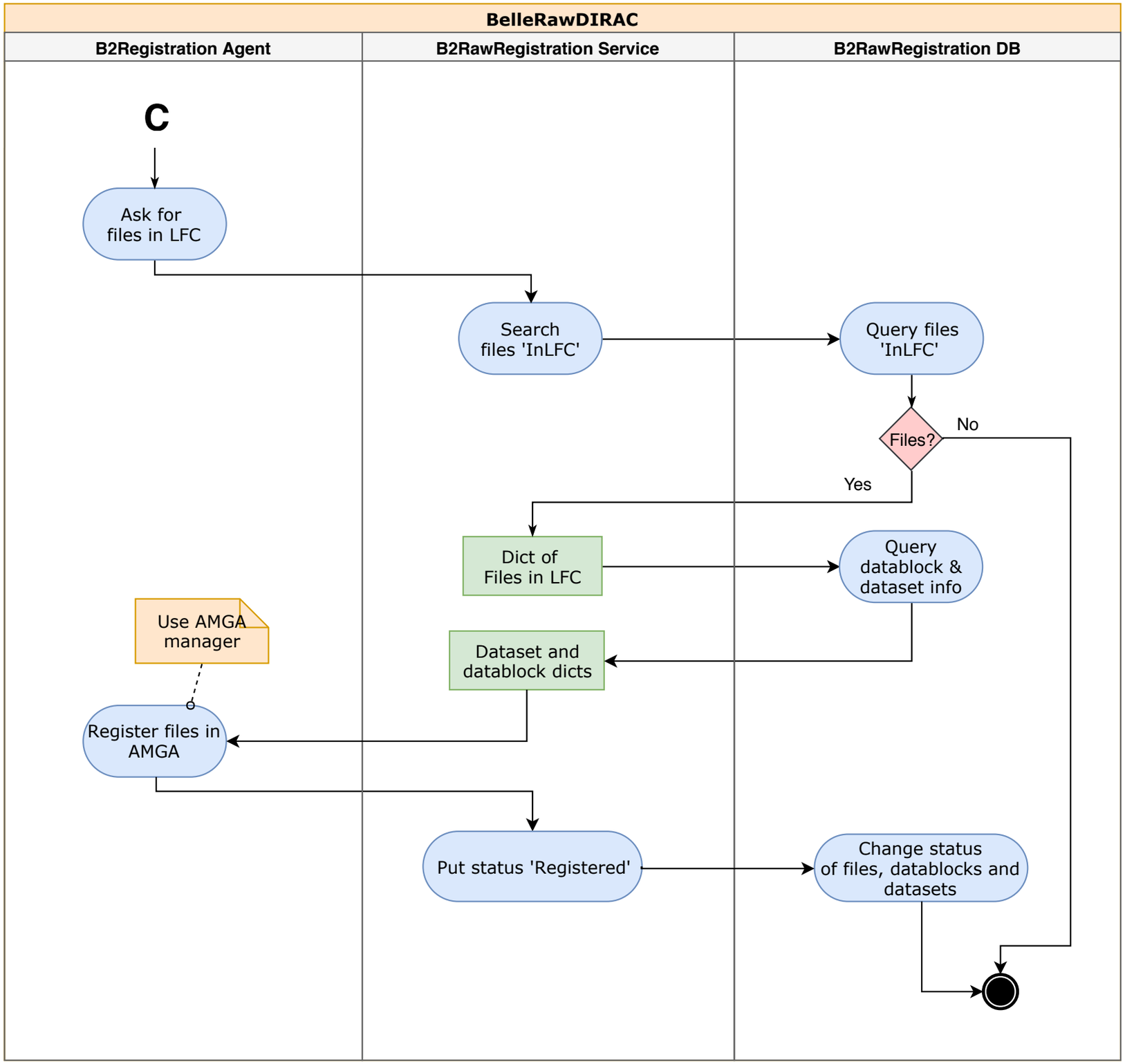
'Locate' the files, which means:

1) Get metadata.

2) Build LFN.

3) Assign dataset and datablock.

B2RawRegistration agent



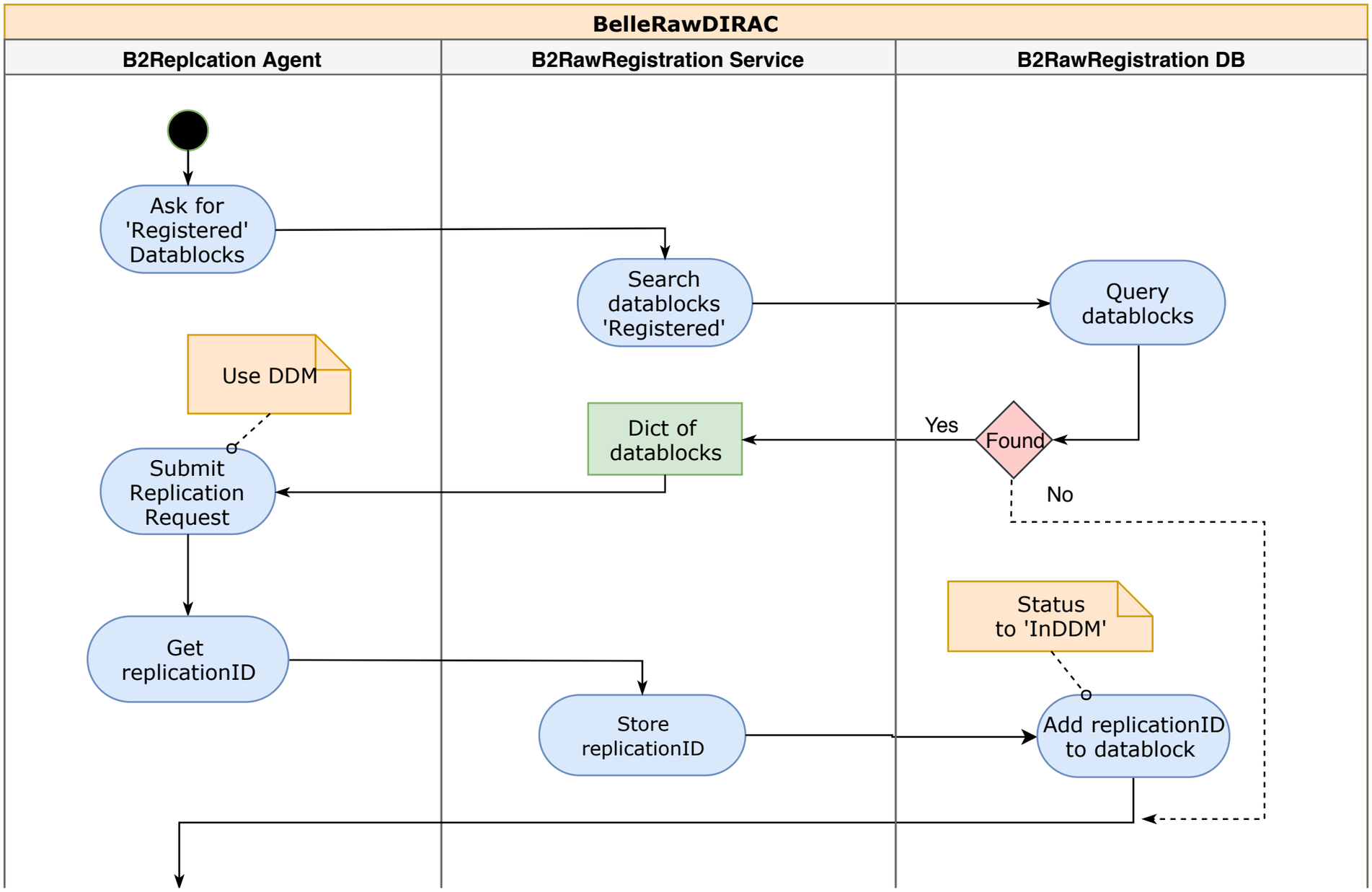
Files in LFC are registered in metadata catalog (AMGA).

It also checks if datablock and dataset entries in AMGA exist.

If yes, it updates the number of files, size, etc.

If not, it creates the entries.

Replication agents



Using the DDM, Submits replication request for datablocks with status ‘Registered’.

One instance per replica on Grid.