



THE UNIVERSITY of
MISSISSIPPI



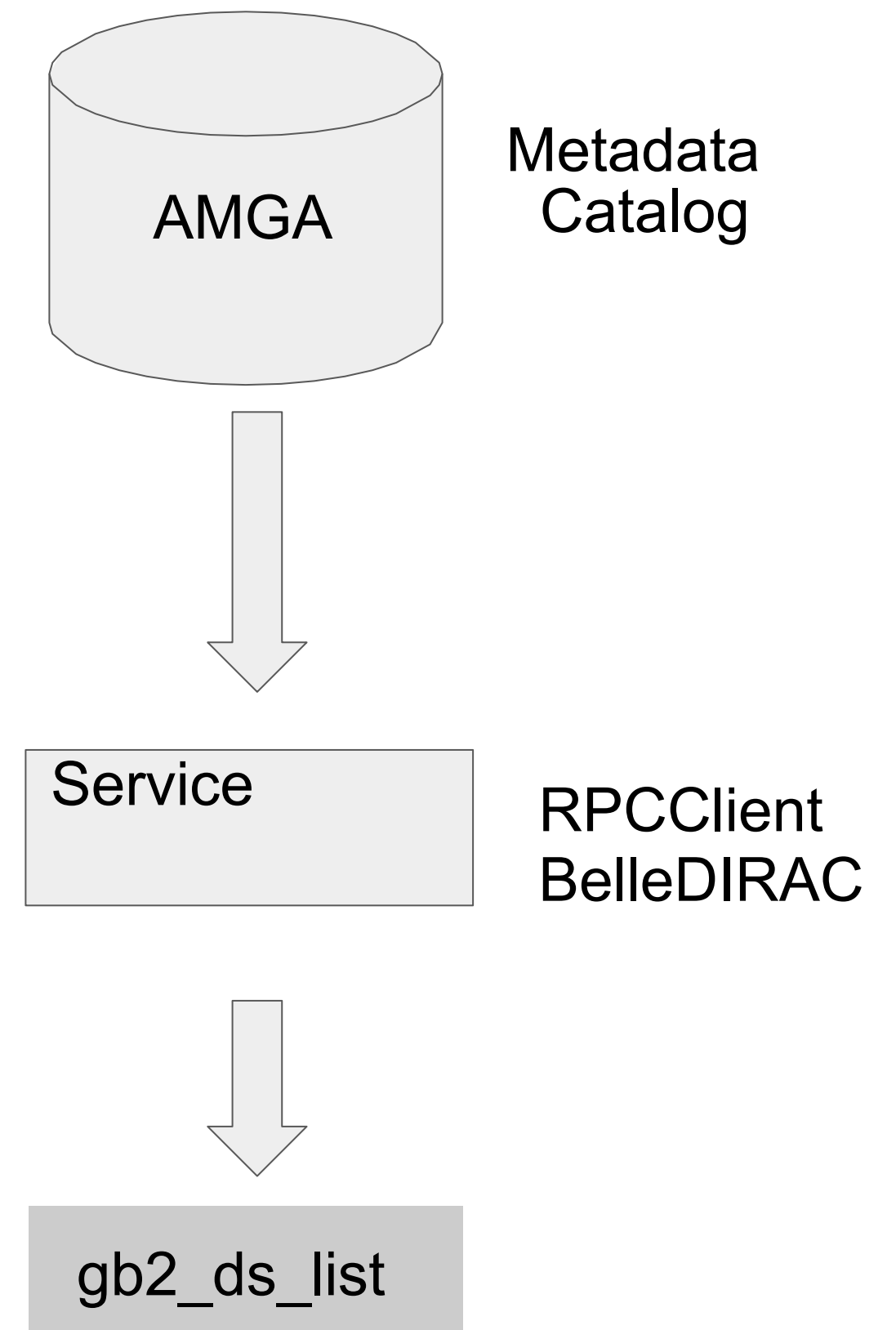
Development and prospects in gb2 tools

Anil Panta, Michel Villanueva
The University of Mississippi

Computing Workshop @ KEK
Oct 18th, 2019

gb2_ds_list (BIIDCD-902)

- Currently, **gb2_ds_list** displays all the files in a LPN without any validation in their status.
- Goal:
 - Our idea is to use a DIRAC service (RPCClient) to call AMGA .
 - Check if the file have status 'good' or not in metadata.
 - List only the files which have only good status in metadata.
 - Give the option to list all the files whatever the status.
 - And another option to list the files having specific status.



Implementation

1) If the <dataset> is LPN that contains files or <dataset> is LFN itself.

a) `gb2_ds_list <dataset>` : List the files with status 'good'

b) `gb2_ds_list <dataset> -l` : List LFN of files with site name , siteLFN and size

c) `gb2_ds_list <dataset> -l -g` : List LFN of files with the site name and size

a) `gb2_ds_list <dataset> -status <status_name>` : List the LFN which have status given by user

b) `gb2_ds_list <dataset> -a` : List all the files (doesn't check for status)(No AMGA call)

2) If the <dataset> is LPN that contains another directory then **no AMGA call**. It does the same thing as it did before this feature.

Current flags

New Flags

gb2_list_se (BIIDCD-781)

- Currently, SE access status check is done from the Configuration System (CS), but its crashing whenever the status is not specified in the CS.
- Goal:
 - To check SE Access status in RSS.
 - Also this tool list the status of file from CS but the status can be different in RSS.
 - This tools also gives the option to list the endpoint and path.
 - We want to implement only checking Access status of SE from RSS.
 - We want to take out the option of printing the endpoint and path (making a different tool for this).

Implementation

```
$ gb2_list_se
```

```
$ gb2_list_se --status Read/Write/Remove/Check
```

```
[apanta@cw08 BelleDIRAC]$ gb2_list_se
```

SE	Read	Write	Remove	Check
Adelaide-DATA-SE	Active	Active	Active	Active
Adelaide-SE	Active	Banned	Active	Active
Adelaide-TMP-SE	Active	Active	Active	Active
BNL-DATA-SE	Active	Active	Active	Active
BNL-TAPE-SE	Active	Active	Active	Active
BNL-TMP-SE	Active	Active	Active	Active
CESNET-DATA-SE	Active	Active	Active	Active
CESNET-TMP-SE	Active	Active	Active	Active
CINVESTAV-DATA-SE	Active	Banned	Active	Active

```
ReadAccess == Active
=====
Adelaide-DATA-SE
Adelaide-SE
Adelaide-TMP-SE
BNL-DATA-SE
BNL-TAPE-SE
BNL-TMP-SE
...
```

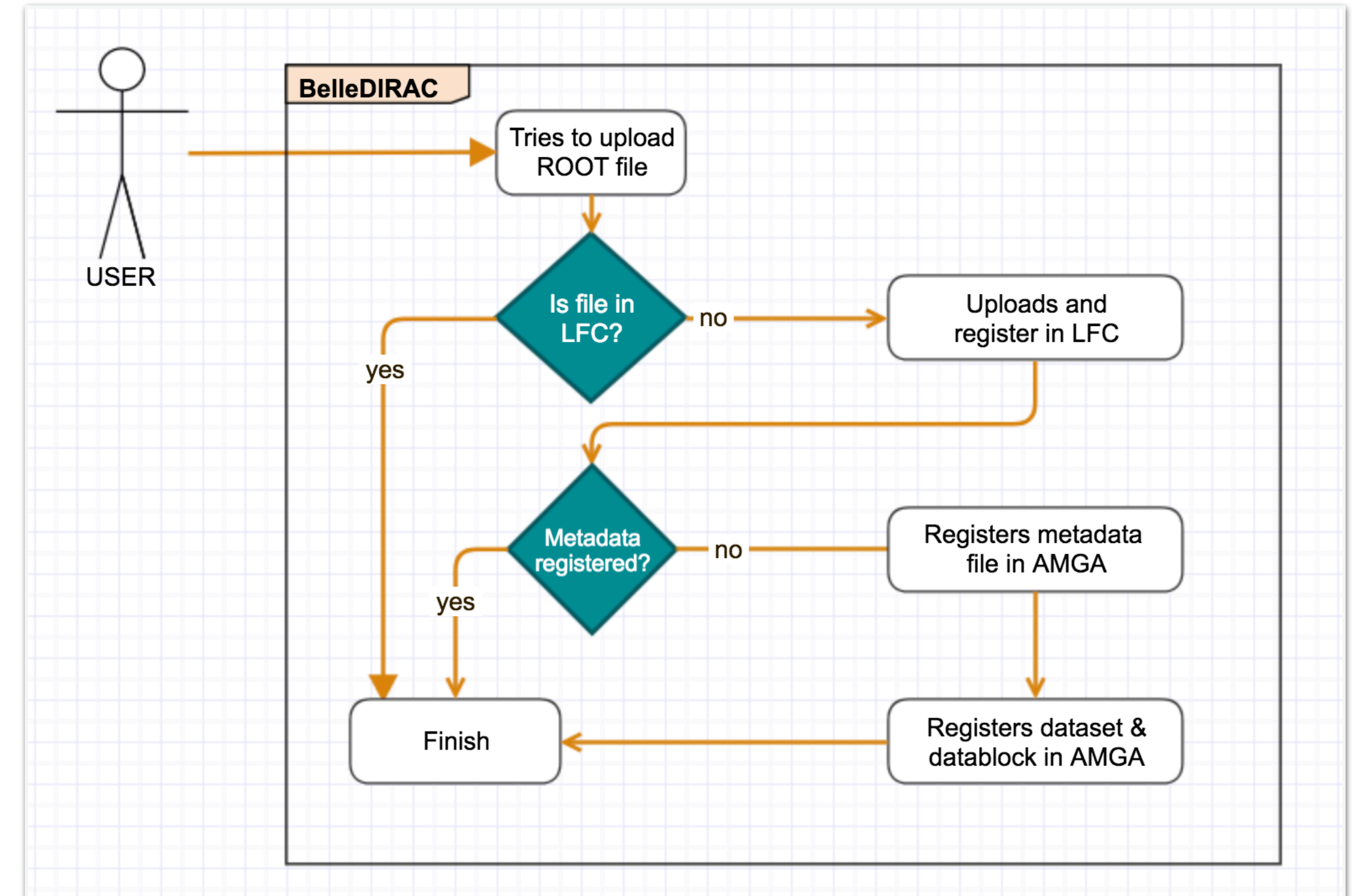
gb2_path_se

- For listing full SURL. We purpose to have a new command line tool: **gb2_path_se**
- **\$ gb2_path_se**
 - List of full SURL for all SE.
- **\$ gb2_path_se --<flag_name> <se_name>**
 - Printing full SURL of specified SE.
- We have a first implementation. Before opening a JIRA ticket for development, we would like to ask suggestion for the name.

Short/Mid term plans

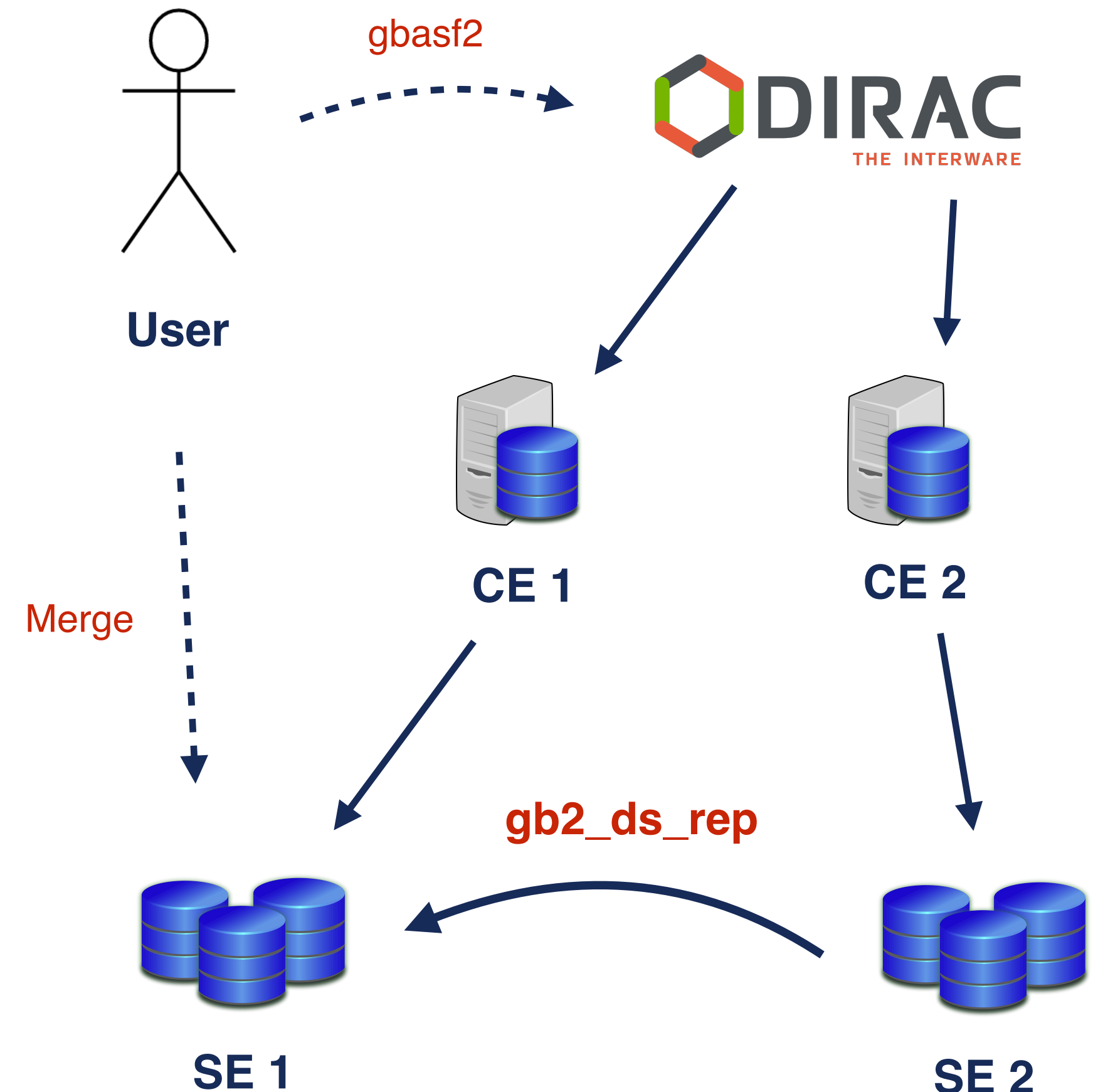
gb2_ds_put

- Minor details to be fixed, but ready in principle.
- Right now, **gb2_ds_generate** uploads and registers files, without treating the datablock structure or the metadata embedded properly.
- **gb2_ds_du** may replace it as the tool for uploading and registration of files.
- However, the is not compatible with the current structure of files produced by users with gbasf2.
- Is this oriented for users?
- How to implement it for its usage?



gb2_ds_rep

- One of the features requested from users is the merging of files before using **gb2_ds_get** (in order to reduce the time required to download a dataset).
- Merging jobs work only if the files to be merged are in the same SE.
- Since specify a destination SE is not recommended in general, the other solution is preparing tools for transferring files between SE.
- **gb2_ds_rep** in principle does this job. It uses `dataManager.replicateAndRegister`. However, the failure rate is high.
- Asynchronous operation is currently broken.
 - May we use DDM for this purpose?



gb2_ds_rep

- Several files failed during the transfer process.

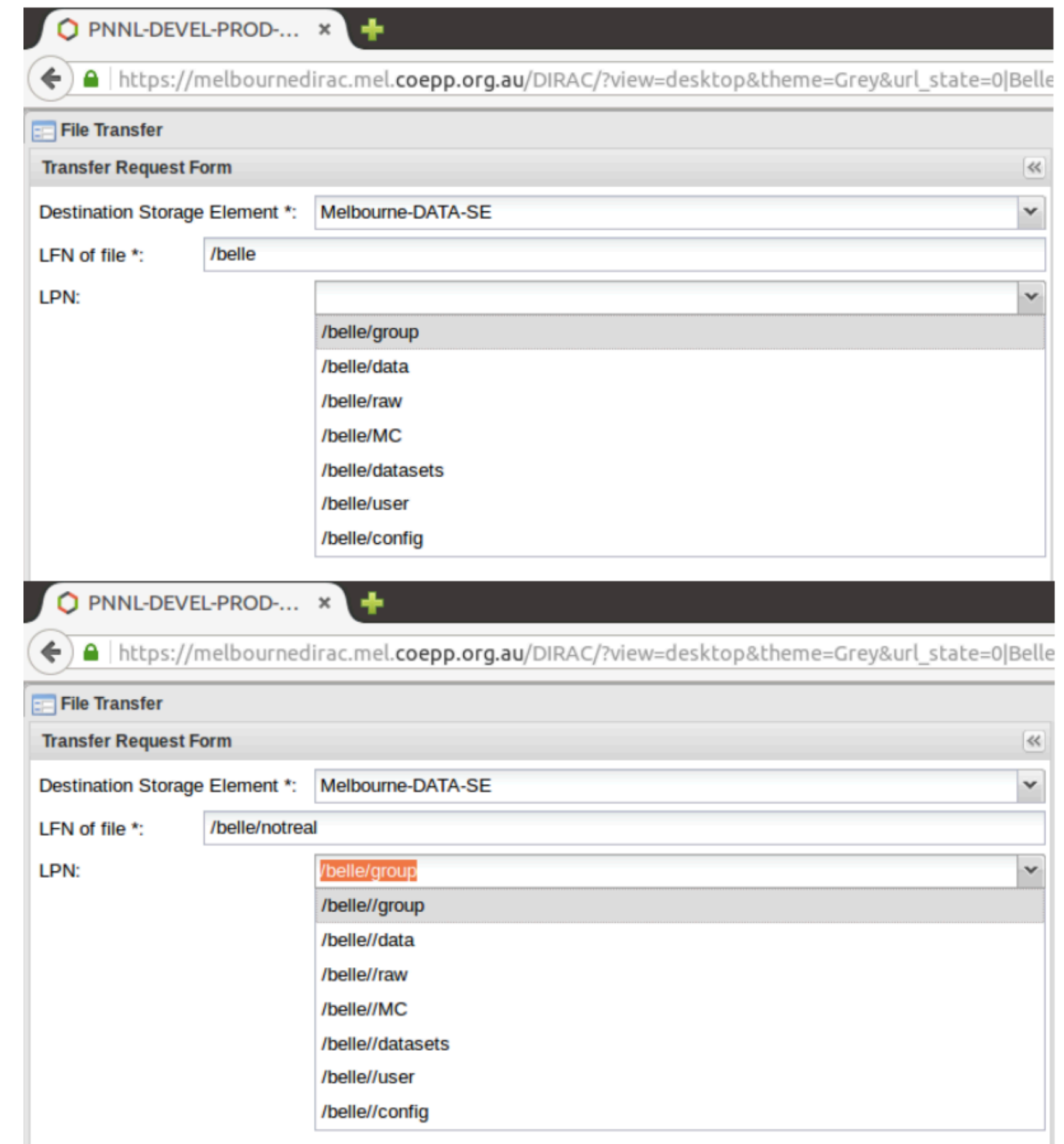
Total replication rate: 15.6 (MB)/ 123.6 (sec) = 0.1 (MB/sec)

LFN	result
=====	
Succeed	
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-101.mdst.root	OK
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-103.mdst.root	OK
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-105.mdst.root	OK
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-106.mdst.root	OK
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-108.mdst.root	OK
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-109.mdst.root	OK
Failed	
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-10.mdst.root	Completely failed to replicate file. Failed to replicate with all sources.
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-100.mdst.root	Completely failed to replicate file. Failed to replicate with all sources.
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-102.mdst.root	Completely failed to replicate file. Failed to replicate with all sources.
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-104.mdst.root	Completely failed to replicate file. Failed to replicate with all sources.
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-107.mdst.root	Completely failed to replicate file. Failed to replicate with all sources.
/belle/user/michmx/test_badFiles/sub02/ee_taeu_BGx0_1K-1000.mdst.root	Completely failed to replicate file. Failed to replicate with all sources.

- Asynchronous transfers (using data operation requests to DDM) will improve the situation.

WebApp for data management

- Work started by Anton Hawthorne (Melbourne).
- The development was almost complete, but got stuck due to the changes in the DDM.
- It submits transfer requests to DDM.
 - Selection of SE from a drop menu.
 - Competition of LFN.
- Shows information of the current transfers.
- Issues pointed by Anton:
 - No checks done on user permissions.
 - Assumes that all the files have the same size.
- We will work in finishing the tool (in parallel to the work performed in **gb2_ds_rep**).



Gbasf2 starter kit

- Python notebooks are becoming a standard in the Belle II starter kits.
- A first version of the **gbasf2 tutorial** using Python notebooks was showed at the Belle II US Summer School 2019.
- The [notebook](#) provides a guide with the input/output of real-time examples.
- For now, it is static for students.
- In general, good comments. Examples are clear, easy to follow.
- We need to define a procedure to run a notebook server for students.

4. Retrieving the output

Once the job has finished, you can list the output using `gb2_ds_list`.

The output file will be located below your user space
(`/belle/user/<username>/<project_name>`)

```
In [11]: gb2_ds_list|grep gb2Tutorial_bbarGeneration
/belle/user/michmx/gb2Tutorial_bbarGeneration
```

```
In [12]: gb2_ds_list /belle/user/michmx/gb2Tutorial_bbarGeneration
/belle/user/michmx/gb2Tutorial_bbarGeneration/B2A101-Y4SEventGeneratio
n-evtgen.root
```

To download the files, use `gb2_ds_get`.

```
In [17]: # Let's create a directory to get the files of the tutorial under home
mkdir -p ~/gbasf2Tutorial && cd ~/gbasf2Tutorial

# Now, let's download the file
gb2_ds_get /belle/user/michmx/gb2Tutorial_bbarGeneration --force
```

```
Download 1 files from SE
Trying to download srm://dcblsrm.sdcc.bnl.gov:8443/srm/managerv2?SFN=/
pnfs/sdcc.bnl.gov/data/belldiskdata/TMP/belle/user/michmx/bbar_genera
tion_example/B2A101-Y4SEventGeneration-evtgen.root to /belle2/u/michm
x/gbasf2Tutorial/bbar_generation_example/B2A101-Y4SEventGeneration-evt
gen.root
```

```
Successfully downloaded files:
/belle/user/michmx/bbar_generation_example/B2A101-Y4SEventGeneration-e
vtgen.root in /belle2/u/michmx/gbasf2Tutorial/bbar_generation_example
```

Running Jupyter Notebooks with gBasf2

- Although initially Jupyter was designed to work with Python, it is compatible with several kernels, including **Bash**.

```
$ pip install bash_kernel --trusted-host pypi.python.org  
--trusted-host pypi.org --trusted-host  
files.pythonhosted.org
```

```
$ python -m bash_kernel.install
```

- Idea: proving a gBasf2 installation with the option of having Jupyter installed.

- Forwarding a port in the ssh session

```
$ ssh user@host -L <port>:localhost:<port>
```

- Setting the gbasf2 environment

```
$ source gbasf2/BelleDIRAC/gbasf2/tools/setup  
$ gb2_proxy_init -g belle
```

- Starting the server

```
$ jupyter notebook --port <port>
```

```
michmx@cw07:~$ jupyter notebook --port 8003  
[I 09:13:02.298 NotebookApp] Serving notebooks from local directory: /gpfs/home/belle2/michmx  
[I 09:13:02.298 NotebookApp] The Jupyter Notebook is running at:  
[I 09:13:02.298 NotebookApp] http://localhost:8003/?token=be🙄🙄2b1e8ef268a40d227d6fe55a9a3f28a9644bfb00  
[I 09:13:02.298 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).  
[C 09:13:02.303 NotebookApp]
```

To access the notebook, open this file in a browser:

```
file:///gpfs/home/belle2/michmx/.local/share/jupyter/runtime/nbserver-22846-open.html
```

Or copy and paste one of these URLs:

```
http://localhost:8003/?token=be🙄🙄2b1e8ef268a40d227d6fe55a9a3f28a964
```

□

Documentation

- We will investigate the usage of **Sphinx** for gbasf2 documentation.
 - It is an standard solution for Python docs.
 - Belle II Software group and DIRAC developers use it. And it works pretty well.
- Much easier to keep updated rather than Confluence (it could be maintained inside the BelleDIRAC repository).
- If it is a feasible solution, it will require the help of the developers to keep it updated.
- <http://www.sphinx-doc.org/en/master/>

Summary

- Efforts in development of gb2 tools are on going.
 - **gb2_ds_list** is ready for checking at the status of the files in AMGA. Pull request is on going.
 - **gb2_list_se** now checks properly the status of SEs in RSS. Ready to perform a pull request.
 - **gb2_se_path** will list the full SURL. A first implementation is ready.
- Mid term plans:
 - **gb2_ds_rep** will perform replicas of files calling data operations in DDM.
 - The WebApp initially developed by Anton for data operations will be reviewed. Changes performed in DDM have to be implemented.
 - The gBASF2 tutorial using Jupyter notebooks provides the examples in cells ready to run. The students learn about how to use gbasf2 and the gb2 tools following the commands in the cells.
 - Implementation of **Sphinx** as solution for documentation will be studied. Several advantages against documentation in Confluence.

Thank you