

Rucio-DIRAC "What to do to use them together"

DIRAC User Workshop 2021

C. Serfon (BNL)





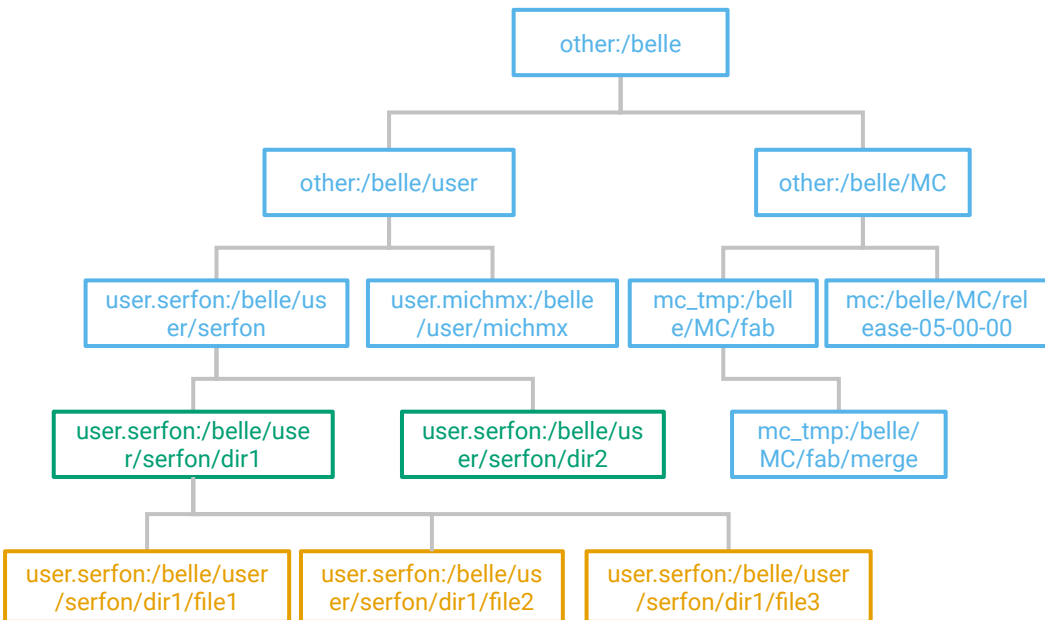
Introduction

- Belle II recently moved to Rucio as Data Management service integrated with DIRAC
 - For this integration a Rucio File Catalog Client (RFC) was developed and integrated into BelleDirac
 - This implementation originally for Belle II but has been made as experiment agnostic as possible → Proposal to move to Vanilla DIRAC
- Rucio has some constraints/features that are not available in DIRAC. It needs to be taken into account by the collaborations that want to use them together
- This talk is to present the plugin and the main differences between LFC or DFC and Rucio and gives a status of the integration of Rucio with vanilla Dirac

Constraints



Namespace organisation



- On the contrary to LFC or DFC, Rucio is not inherently a hierarchical namespace (e.g. ATLAS use it as a flat namespace)
- To mimic LFC hierarchy, following mapping is used (LFC → Rucio) :
 - File → File
 - Directory containing files → Dataset
 - Directory containing directories → Container



Container constraint

- Containers in Rucio cannot contain files therefore mixture of files and directories in the same directory cannot be handled, e.g. :

```
lfc-ls -l /grid/belle/MC/fab/
```

```
drwxrwxr-x 1 292      147          0 Oct 13 2015 merge
-rw-rw-r-- 1 292      147      45240230 Jan 10 2017 output_15634.root
drwxrwxr-x 2 353      147          0 Jun 23 2017 prerelease-00-09-00b
drwxrwxr-x 1 353      147          0 Dec 12 2017 prerelease-01-00-00b
```

← Not possible

- It was not a problem for Belle II since these are test files that have been cleaned during the transition but might be a problem for other collaborations planning to use Rucio



Scope

- Scope is a Rucio concept to :
 - Partition the namespace (between VOs, users, activity) with the possibility to introduce some specific permission at the scope level (Rucio has no fine grained ACLs)
 - Use to apply replication policies
 - Use for accounting
- The concept of scope is unknown in the LFC/DIRAC, therefore it needs :
 - Either be part of the LFN (e.g. /grid/belle/MC)
 - Or there must be any deterministic function that can be used to extract it from the LFN

→ This part has to be implemented independently for each collaboration, e.g for Belle II we chose a deterministic function that map each LFN to one specific scope. The function will be part of Rucio policy package



Rucio setup

- The plugin uses the Rucio client API that therefore needs to be installed on the Dirac server and on the grid. For Belle II :
 - Client installed using pip on the DIRAC servers (will use DIRACOS later)
 - Client installed on CVMS on the grid
- Currently, the rucio.cfg file is mandatory. It is VO dependent cannot be included into DiracOS and needs to be installed manually

```
[common]
extract_scope = belleii

[client]
rucio_host = https://xxx:443
auth_host = https://xxx:443
auth_type = x509
client_cert = /opt/dirac/etc/grid-security/hostcert.pem
client_key = /opt/dirac/etc/grid-security/hostkey.pem
account = dirac_srv
request_retries = 3

[policy]
permission = belleii
schema = belleii
||fn2pfn_algorithm_default = belleii
```

The RFC plugin in BelleDIRAC



Rucio setup

- Some environments variables need to be set in the Dirac Configuration file :



- These variables are used by the pilot to setup Rucio
- The RFC finds which account to use depending on the identity passed by DIRAC



DIRAC → Rucio synchronization

- For Belle II a RucioSynchronizerAgent is responsible of inserting the accounts and RSEs defined in Dirac configuration into Rucio :
 - Dirac users and groups → Rucio accounts
 - Dirac hosts → Rucio account dirac_srv
 - Dirac Storage Elements → RSEs
- In addition a RucioRSSAgent collects the state of the Storage Elements in DIRAC and update the states in Rucio accordingly :
 - Active, Degraded → Available in Rucio
 - Banned → Disabled in Rucio



Read methods

- Read methods can easily be mapped to Rucio. Following were implemented :
getReplicas, listDirectory, getFileMetadata, exists, getFileSize, isDirectory, isFile
 - For getFileMetadata (only returns LFC metadata : checksums, GUID, filesize)
 - Behaviour is the same as in the LCGFileCatalogClient plugin
- Most of the methods use bulk methods on the server side that allows to have fast response, but some methods suffer from "slowness", probably due to some FileCatalog implementation (under investigation)

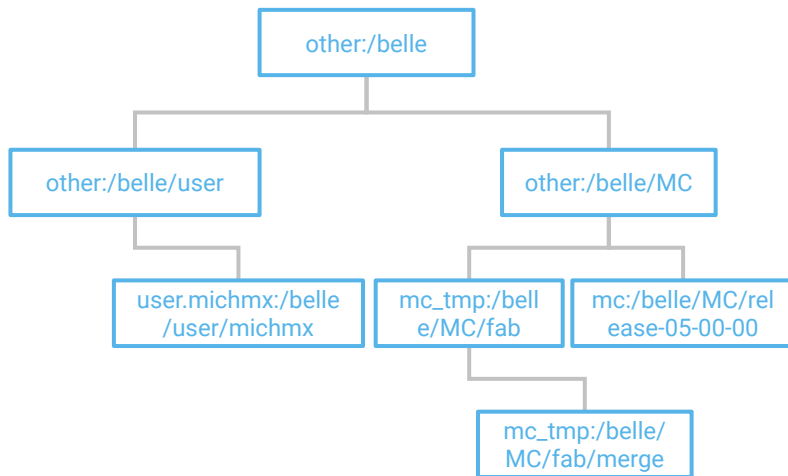


Write methods

- Methods implemented :
 - addFile :
 - Complex method since Rucio and LFC concept are quite different, that does multiple things (next slide for details)
 - New method was introduced on the Rucio side to provide atomicity + bulk
 - addReplica :
 - Add the replica + create a rule on the file



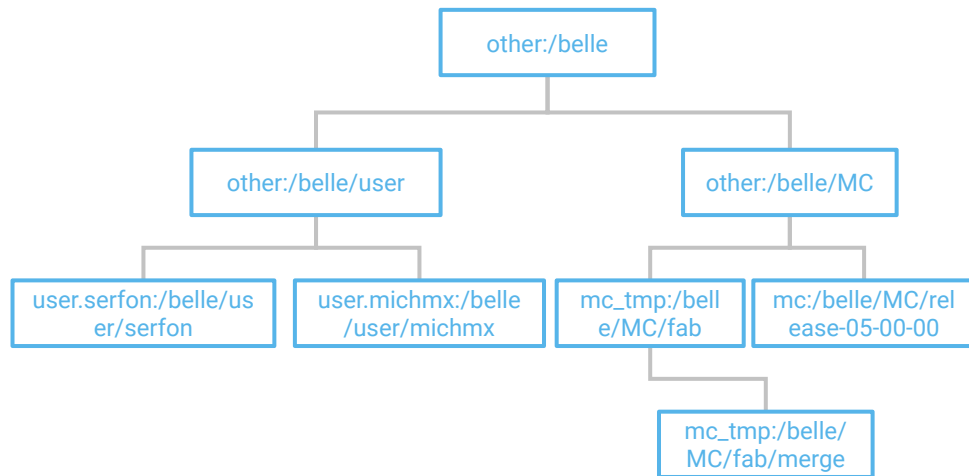
Addfile method



- Example : Creation of files /belle/user/serfon/dir1/file{1-3} on SE A
- The addfile method is a bulk and atomic method that :



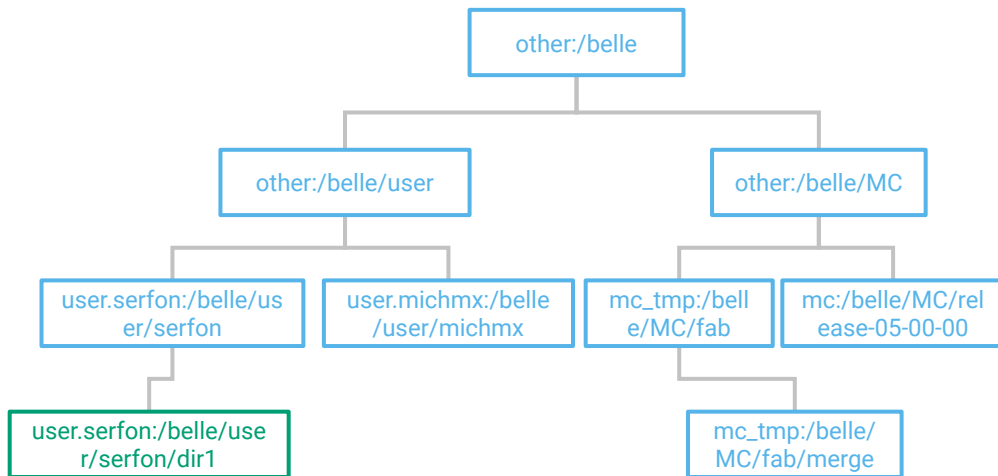
Addfile method



- Example : Creation of files /belle/user/serfon/dir1/file{1-3} on SE A
- The addfile method is a bulk and atomic method that :
 - Creates all the non-existing parent directories in the hierarchy



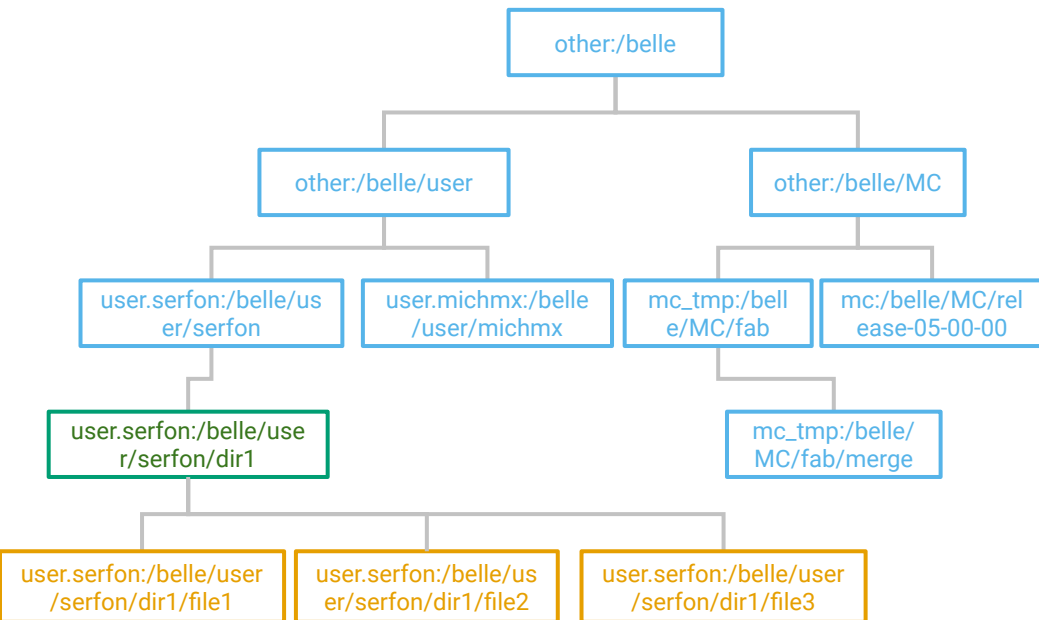
Addfile method



- Example : Creation of files /belle/user/serfon/dir1/file{1-3} on SE A
- The addfile method is a bulk and atomic method that :
 - Creates all the non-existing parent directories in the hierarchy
 - Creates the directory containing the files if it doesn't exist and create a Rucio rule (RSE expression ANY and grouping NONE)



Addfile method



- Example : Creation of files /belle/user/serfon/dir1/file{1-3} on SE A
- The addfile method is a bulk and atomic method that :
 - Creates all the non-existing parent directories in the hierarchy
 - Creates the directory containing the files if it doesn't exist and create a Rucio rule (RSE expression ANY and grouping NONE)
 - Create the files and their replicas on A



Delete methods

- Delete methods have some different behaviour because of the different concept in the LFC/Dirac and Rucio, e.g. :
 - removeReplica : In Rucio, not possible if the file belongs to a dataset and there is a rule on this dataset. If this is the case, the RFC doesn't do anything. If a rule exists on the file, remove it
 - removeFile : In LFC only removes the file if it has no replicas, otherwise trigger exception. For Rucio, no check if replicas exist : Remove file just correspond to detach a file from a dataset making all the file replicas eligible for deletion (asynchronous deletion)
 - The file will be visible until it is actually deleted by Rucio
 - The file will NOT be visible in the list of the files under the parent directory.



Belle II specificities

- We are implementing a new “find” method to list all the descendants of a particular directory
- In addition we are evaluating if Rucio can be used to store some metadata. If we decide to go that way, we might implement metadata methods in BelleDIRAC

Integration in Vanilla Dirac



Current status

- rucio was added to DIRACOS. It will need to be moved to DIRACOS2
- [RFC PR](#) was submitted to [v7r0] a few months ago :
 - A lot of feedback from DIRAC developers. Thanks a lot !
 - Most of the comments were implemented
- What is currently missing are the tests: There is currently no Rucio instance that can be used to run the tests. Two possibilities:
 - Setup an external test instance that can be used by github actions
 - Create a new Rucio instance for every tests (can be done using docker)
 - After discussion in last week BiLD meeting, it was agreed to implement tests as github actions in v7r2
- I currently do not have much time to work on this. Help welcome



Current status

- There is a consensus to get rid of the rucio config when installing the Rucio client and to read the configuration from the Dirac Configuration Service :
 - Most of the options can be passed to the Client constructor
 - BUT, some hidden access to the configuration from inside the client (in particular, but not only for the logging)
 - It will be addressed by the Rucio developers (no timescale yet)
- The current implementation will need some adaptation for the multi-VO support that might not be straight forward



Conclusion

- The RFC prototype has been tested extensively in Belle II and has been used in production for the last 4 months
- The current RFC only contains the replicas methods which were supported by the LFC. No metadata support is currently available
- Some work still needs to be done to validate the RFC PR, in particular tests