cherenkov telescope array

Rucio DIRAC integration
On-going tests for CTA

Luisa Arrabito

LUPM CNRS-IN2P3 France

DIRAC@IN2P3 F2F Meeting 14th April 2022, Lyon

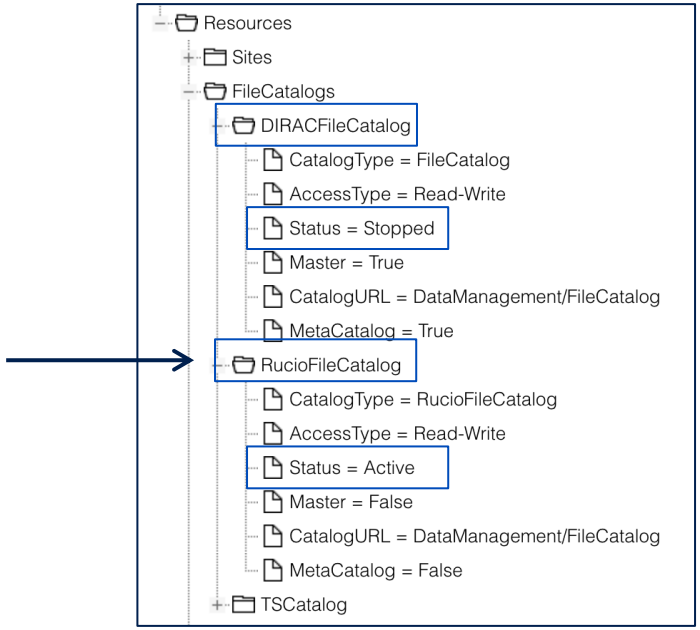
CTADIRAC-Rucio integration

- Work done in the context of Escape (F. Gillardo, PIC with the help of C. Serfon) for Use Case 2
- **Use Case 1: Long haul ingestion and replication**
  - Ingestion of CTA data from a remote site (RSE at the CTA Array Site, 'on-site') to the data lake, transfer and replication in off-site RSEs and after replication deletion of the data at origin
- **Use Case 2: Data Reprocessing**
  - Goal
    - The ability to reprocess all raw data (DL0) to higher (DL3) level
  - Workflow
    - Raw (DL0) data is identified on tape (obsid or time range) via metadata
    - Data volume is calculated
    - Data is staged from tape (COLD) storage to temporary disk (HOT)
    - Data is reprocessed using CTA pipeline software via the workload management system (WMS, based on DIRAC) using a cache area for on-the fly, transient data products
    - Final data products (DL3) are verified
    - Cache and temporary data is cleared
    - Ingest the resulting new DL3 data into the datalake
    - Update the corresponding metadata

CTADIRAC-Rucio integration

- Using the CTA-DIRAC certification instance for testing Rucio
  - Single-server installation running all DIRAC services (VM hosted at CC-IN2P3)
- Using the RucioFileCatalog plugin available in v7.3.12

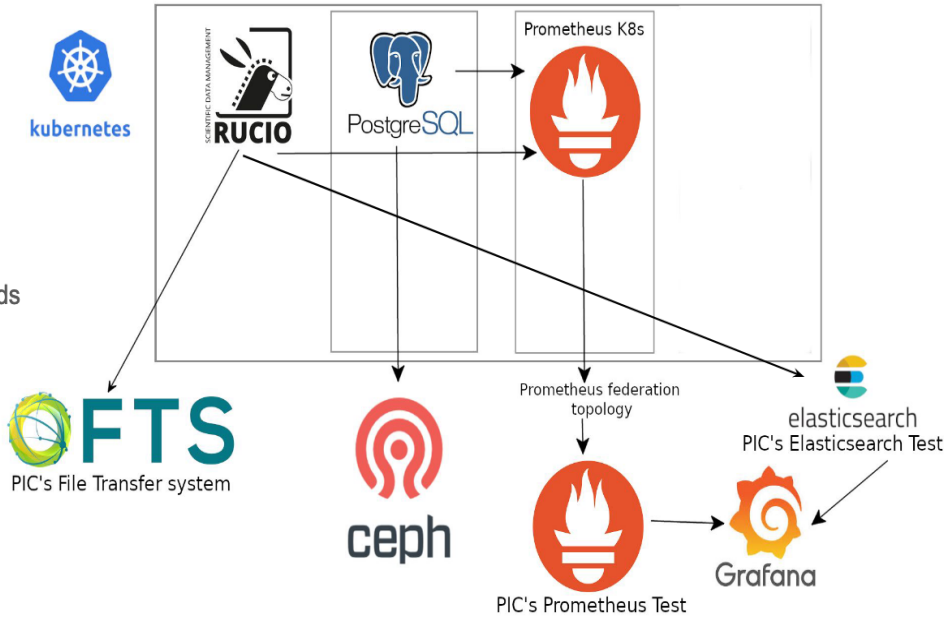
- Configured to use the RucioFileCatalog only (for simplicity)



CTADIRAC-Rucio integration

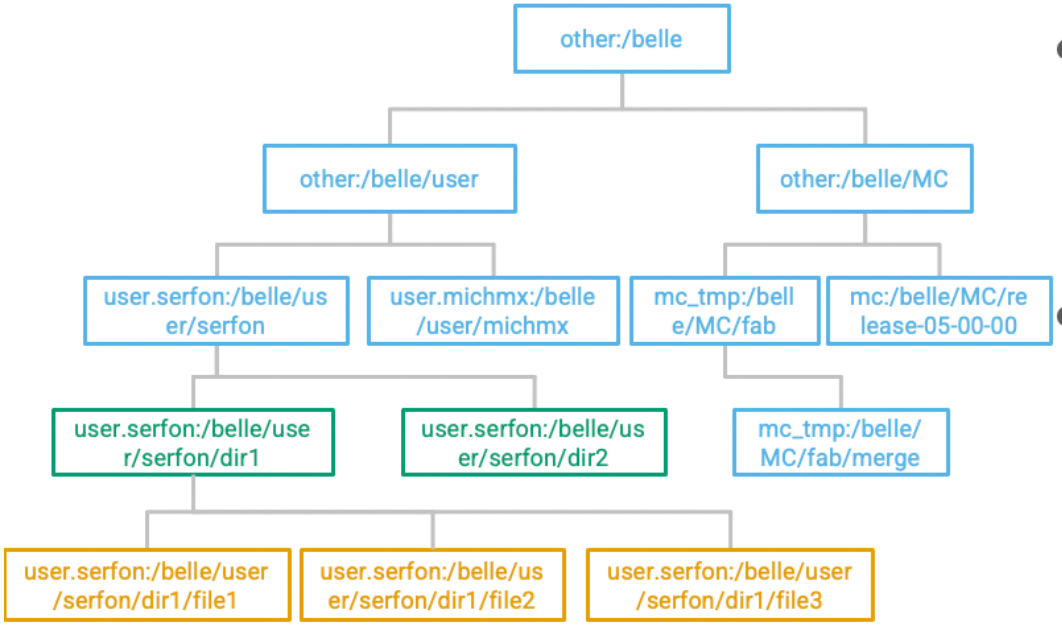
- Using a Rucio instance dedicated to CTA installed at PIC (Escape project)

- **Deployed with:**
  - The Rucio server and daemon services are fully packaged with Helm
    - V1.23.14
  - Available in : https://rucio.github.io/helm-charts/)
  - FTS
    - v3.10.1
- **Monitoring:**
  - Will be using Prometheus for all our monitoring needs
    - v2.11.1
- **Logging:**
  - Currently writing the logs using hermes2 in our own
  - Elasticsearch centralized instance
    - v7.12.1



From Jordi Delgado presentation in Escape meeting

Namespace organisation



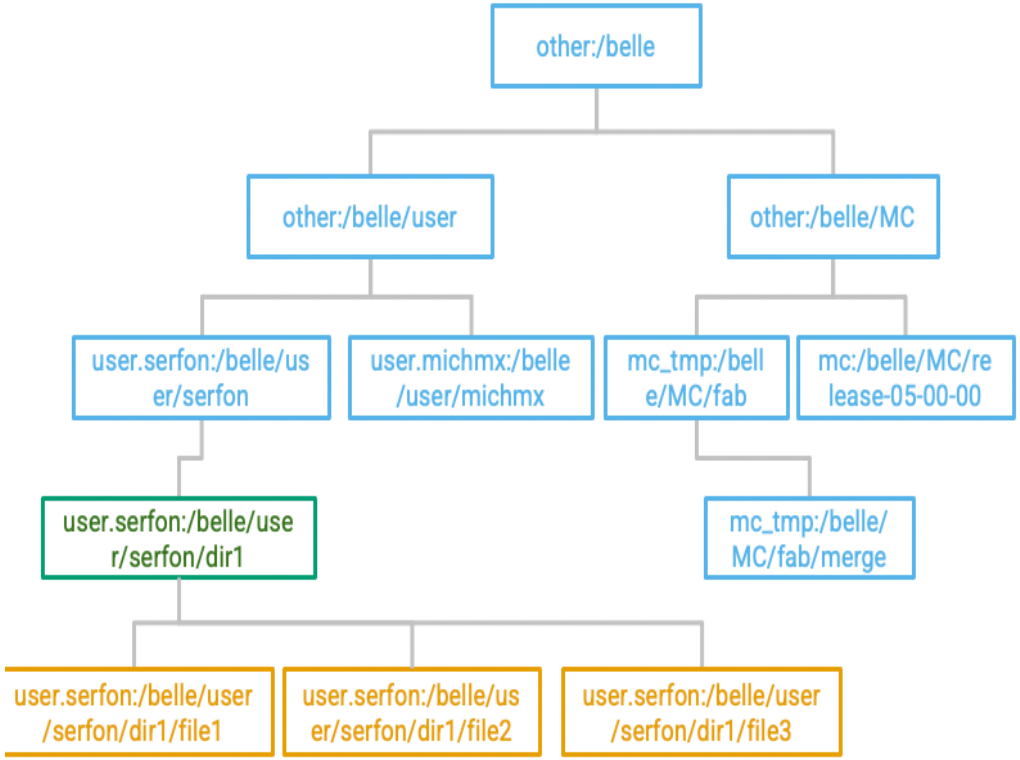
- On the contrary to LFC or DFC, Rucio is not inherently a hierarchical namespace (e.g. ATLAS use it as a flat namespace)
- To mimic LFC hierarchy, following mapping is used (LFC → Rucio) :
  - File → File
  - Directory containing files → Dataset
  - Directory containing directories → Container

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

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 exists and create a Rucio rule (RSE expression ANY and grouping NONE)
  - Create the files and their replicas on A

Tests done so far

- Put And Register file
  - We are using Belle2 Policy Package for now but we could create our own
  - Policy packages are separate Python packages that can be used to add experiment-specific customisations to Rucio. They typically customise Rucio's handling of permissions and schema as well as optionally adding their own algorithms for various purposes, such as lfn to pfn conversion and surl construction

```
from DIRAC.DataManagementSystem.Client.DataManager import DataManager

catalogs=['RucioFileCatalog']
dm = DataManager(catalogs)

fileName = "./data/testUpload.txt"
lfn = "/vo.cta.in2p3.fr/dir1/testUpload.txt"

res= dm.putAndRegister(lfn, fileName, "PIC-CTA")
```

Tests done so far

- Submit Jobs accessing data registered in Rucio
  - Need to configure pilots to access rucio
    - RUCIO_HOME=/cvmfs/cta.in2p3.fr/software/rucio
    - RUCIO_ACCOUNT=ctauser
    - RUCIO_AUTH_TYPE=userpass
  - Rucio configuration installed on CVMFS

```
[arrabito@cca001 ~]$ cat /cvmfs/cta.in2p3.fr/software/rucio/etc/rucio.cfg
[client]
rucio_host = https://pic01-rucio-server.pic.es:443
auth_host = https://pic01-rucio-server-auth.pic.es:443
ca_cert = /etc/grid-security/certificates/
#auth_type = x509_proxy
auth_type = userpass
username = ctauser
password = ********
account = ctauser
[common]
extract_scope = belleii
```