

Data management with rucio for KM3NeT

Victor Azizi, Bouwe Andela

netherlands
eScience center

1. Introduction to Rucio

- What can it do
- Rucio
 - Declarative data management
 - Architecture
 - Namespaces
- Examples
- Resources to get started

What can Rucio do for you

- **Data management:** organizing and cataloging data and metadata
- **Data replication:** managing the replication of data to various locations with user defined rules
- **Monitoring:** keep track of what data is where
- **Dynamic data placement:** plugins are available to make data dynamically available at sites when needed for a job

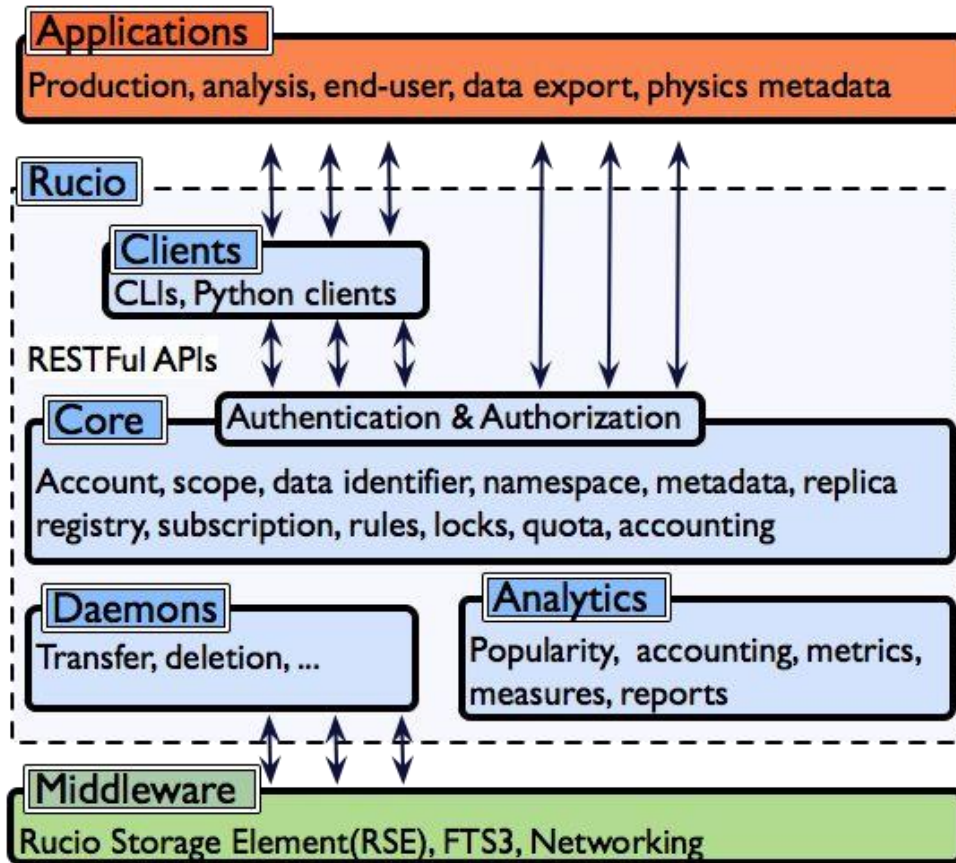


Rucio: declarative data management

- Express what you want with rules. Example:
 - **Rule**
"Three copies of this dataset, distributed evenly across three institutes on different continents, with two copies on DISK and one on TAPE"
- Rules allow a fully dynamic and automated data distribution
 - Rules can be dynamically added and removed by all accounts, some pending authorisation
 - Rucio constantly evaluates all rules and tries to satisfy them
 - Rules enforce data lifecycles with lifetimes (e.g., automatically delete temporary data after a week)
 - Rules enforce user and group quotas (e.g, 50 PB globally for a physics group, 10 extra PB at a site)
- User and group quotas can be set for individual storage endpoints



Rucio: Architecture



- **Servers**
 - HTTP REST/JSON APIs
 - Token-based authentication (x509, ssh, kerberos, ...)
 - Horizontally scalable
- **Daemons**
 - Orchestrates the collaborative work e.g., transfers, deletion, recovery, policy
 - Horizontally scalable
- **Persistence**
 - Object relational mapping
 - Oracle, PostgreSQL, MySQL/MariaDB, SQLite
- **Middleware**
 - Connects to well-established products,
 - e.g., FTS3, DynaFed, dCache, EOS, S3, ...

Credit: slides from Martin Baristis on rucio

Rucio: Namespace handling

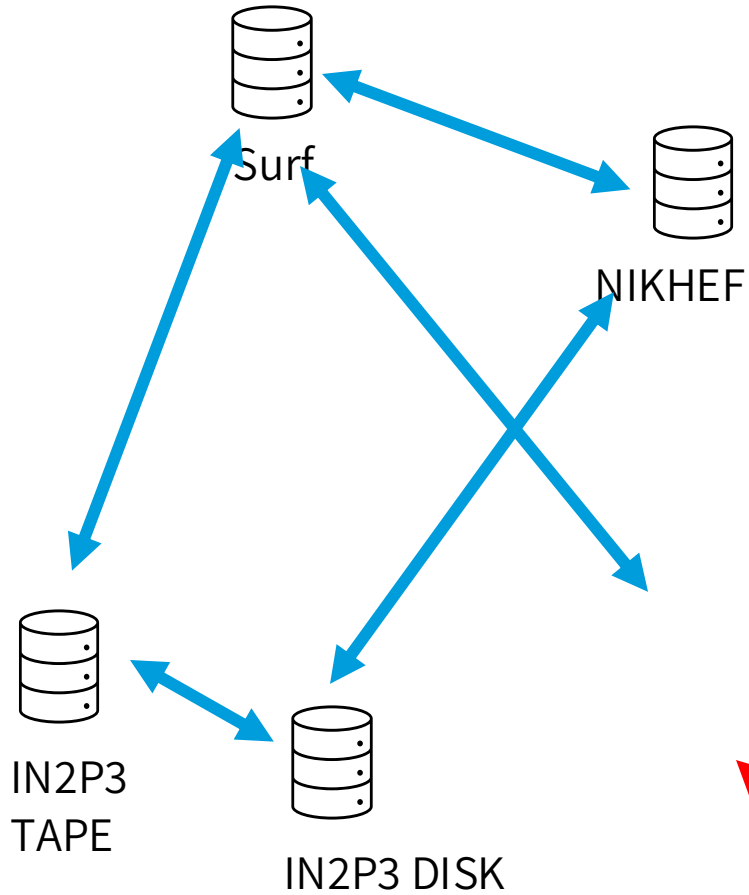
- Data is federated in a single namespace
 - Ensure transparent access across multiple locations
- Data Identifier (DID) is the primary addressable unit
 - DIDs can be either files, collections (datasets), or collections of collections (containers)
 - Datasets only hold files, containers only hold datasets and containers
- DIDs are standalone and partitioned
 - Files do not need to be in a dataset
 - Datasets do not need to be in a collection
- DIDs are globally unique Files cannot have the same name as collections, and vice versa
 - Prevent reuse of modified files for consistently repeatable science results
- Collections can be organized freely
 - Files can be in multiple datasets, datasets can be in multiple containers

Example did:

testing:test_file.txt

<scope>:<unique name>

Storage



- Define rules for data (de)duplication, and automatically ensure those rules are followed (by instructing a file transfer service)



Orchestrator
and Catalogue



User

- Interface for directly up/downloading data, adding data to the catalogue



Accessing webui

- Go to <https://webui.rucio.km3net.org>
- Login with your certificate (add it to your browser first)
- Insight in datasets, rules, etc...
- Note: no downloading, no metadata (yet).



Uploading data to rucio

- Rucio upload:

```
rucio upload --rse NIKHEF --register-after-upload --scope testing testing:hello_dataset  
hello.txt
```

- Uploads file hello.txt to the nikhef grid storage, in the scope "testing" and attaches it to the hello_dataset



Downloading data

- Rucio download

```
rucio download calibration_legacy:00000133_offline_H_1.0.0_00016415_00016563_1_v1.1.3.datx
```

- Downloads the file from any available source



Data available (currently) in rucio

Various data-sources are available in scopes:

- **Raw data:** available in the scope `data_raw_sea`
- **Processed data:** available in the scope `processed_data`
- **Calibration data:** available in the scope `calibration_legacy`



How to search for data in rucio

- Most files have metadata attached
 - `rucio get-metadata` can be used to inspect available metadata

```
$ rucio get-metadata --plugin json \  
processed_data:KM3NeT_00000146_00016415.data.jpmmuon_jppshower_dynamic.offline.dst.v9.2.root  
  
detid:      00000146  
dtype:      mc.mupage_default.sirene.jterbr  
extension:  root  
reco:       jpmmuon_jppshower_static  
run:        00016415  
typ:        dst  
version_:   v9.2
```

How to search for data in rucio

- You can query rucio with the `--filter` argument

```
vikko@niks:~/local_projects/DIKSAP$ rucio did list -d processed_data: --filter 'run=16415,version_>v9.0'
```

SCOPE:NAME	[DID TYPE]
processed_data:KM3NeT_00000146_00016415.data.jpmmuon_jppshower_dynamic.offline.dst.v9.2.root	
processed_data:KM3NeT_00000133_00016415.mc.mupage_default.sirene.jterbr.jpmmuon_aashower_static.offline.dst.v9.1.root	
processed_data:KM3NeT_00000146_00016415.mc.mupage_default.sirene.jterbr.jpmmuon_jppshower-upgoing_static.offline.dst.v9.0.root	
processed_data:KM3NeT_00000146_00016415.mc.mupage_default.sirene.jterbr.jpmmuon_jppshower_static.offline.dst.v9.2.root	
processed_data:KM3NeT_00000146_00016415.mc.pure_noise.jterbr.jpmmuon_jppshower_static.offline.dst.v9.2.root	



Resources to get started

Will be given later today. Info can be found on:

- https://wiki.km3net.de/index.php?title=Rucio_data_management
- <https://rucio.pages.km3net.de/rucio-documentation/>



e “Empowering
researchers across
all disciplines
through innovative
research software”

netherlands
eScience center

DIKSAP

A Database-integrated KM3NeT solution for automated processing

Contact Person



Bouwe Andela
Senior Research Software Engineer
b.andela@esciencecenter.nl



Victor Azizi
Research Software Engineer
v.azizi@esciencecenter.nl

Replication rules

- Let rucio replicate a file from tape to a grid site storage

```
rucio add-rule --account root testing:hello_dataset 2 "type=DISK"
```

- We can follow the progress here: [cern public fts monitor](#)
- We can follow the progress via rucio:

```
rucio list-rules testing:hello_dataset
```



Transfer 'fd40cfb0-b5f5-11ee-8b95-fa163ea3e633' STAGING

VO: km3net.org

- Delegation ID: 7c88185e6e9ec502
- Submitted time: 2024-01-18T11:37:42Z
- Job finished:
- Priority: 3
- Bring online: 43200
- Archive timeout: -1

- Received by fts-public-001.cern.ch
- Job expires: 2024-01-25T11:37:42Z
- Overwrite flag: Y
- Job type: N
- Cancel flag:
- Pin lifetime: -1
- Target QoS:

Metadata:

```
{"issuer": "rucio", "multi_sources": false, "auth_method": "certificate"}
```

Files transferred	Bytes transferred	Submission time	Start time	Running time	Avg. file throughput	Current job throughput
0 out of 1	0 bytes	2024-01-18T11:37:42Z	- (+27s)	-	-	-

Showing 1 to 1 out of 1

- SUBMITTED
- DELETE
- READY
- 1 STAGING
- ARCHIVING
- ACTIVE
- STARTED
- CANCELED
- FAILED
- FINISHED
- NOT_USED

- First
- Previous
- 1
- Next
- Last

File ID	File State	File Size	Throughput	Remaining	Start Time	Finish Time	Staging Start	Staging End	Archiving Start	Archiving End
+ 211214578	STAGING	-	MiB/s	-	-	-	-	-	-	-
<ul style="list-style-type: none"> ↑ xroot://ccxroot.in2p3.fr:1097//xrootd/in2p3.fr/tape/km3net.org/data/calibration/40K/KM3NeT_00000007/2/K40calibration_00000007_2.tar.gz ↓ xroot://km3net.dcache.nikhef.nl:1094/pnfs/nikhef.nl/data/km3net/rucio/calibration/99/80/K40calibration_00000007_2.tar.gz 										

- First
- Previous
- 1
- Next
- Last

