14th Collaboration Meeting of the
BM@N Experiment at the NICA Facility

Data Management System based on the DIRAC File Catalog for BM@N

Igor Zhironkin
JINR FLNP

Topics

- What is Data Management System
- Why DIRAC?
- Metadata service, overall concept
- What’s next

Amount of data

1st Physics BM@N Run

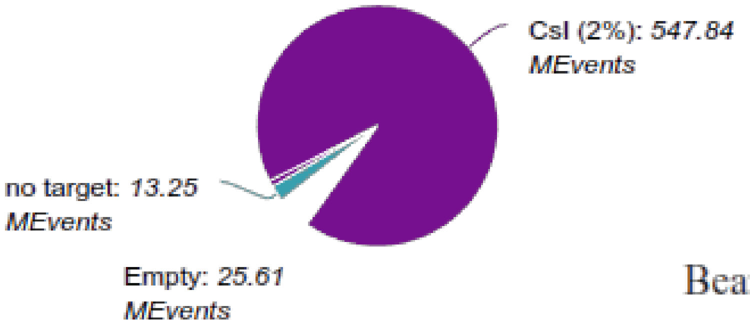
Two beam energy available for *Xe*-beam

CsI target is used as more similar to *Xe*

More than 600M events were collected

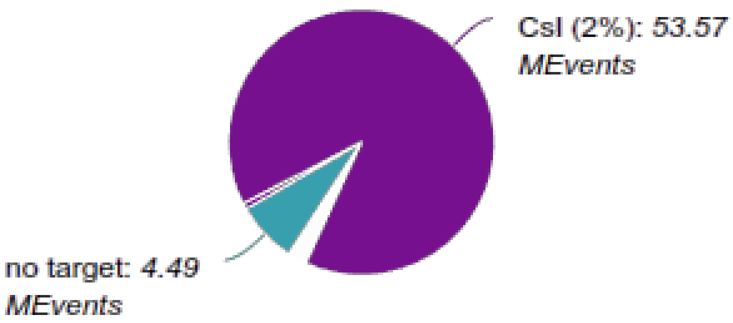
Beam Xe (E = 3.8 GeV/n)

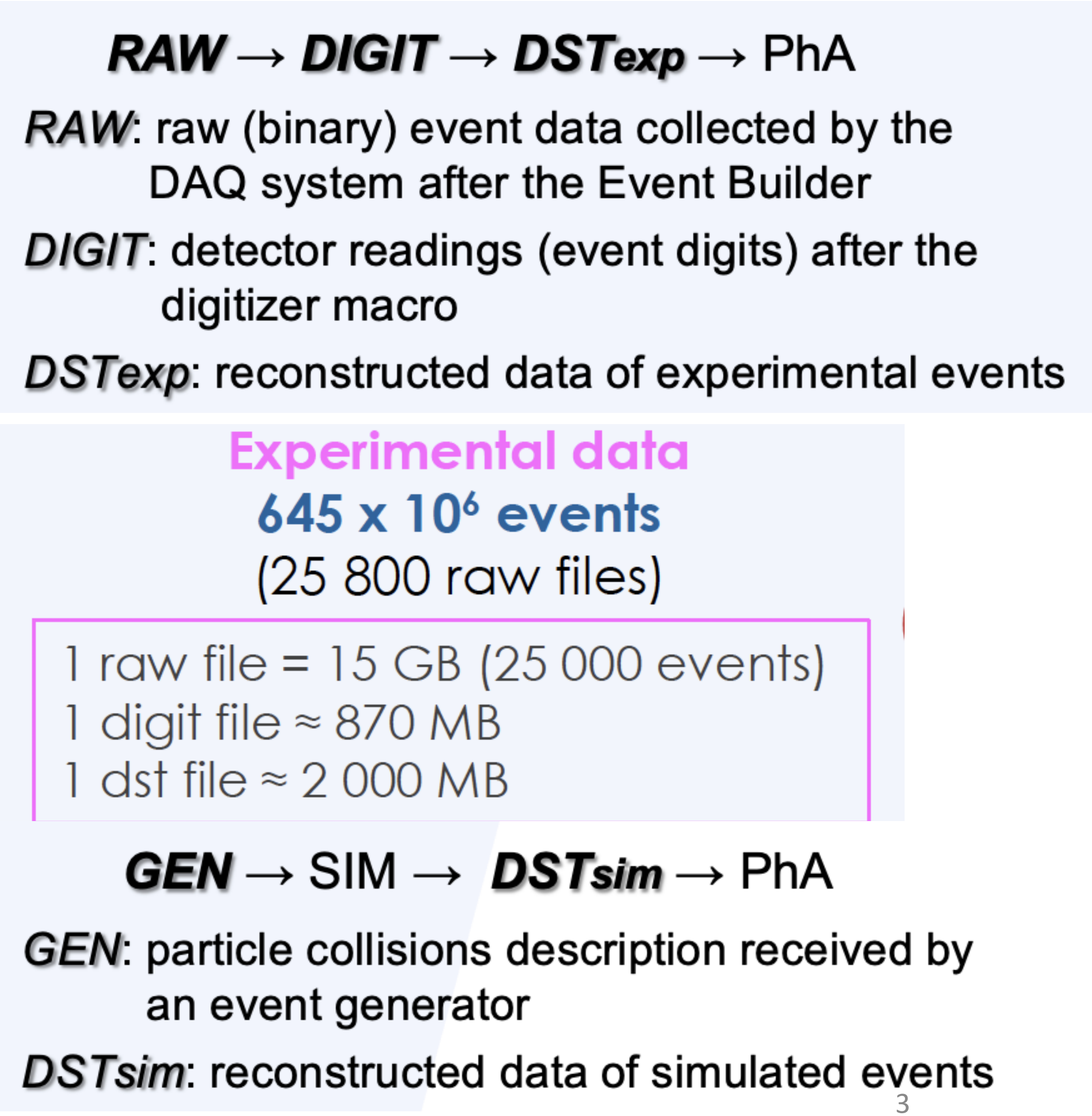
Total: 592.66 MEvents



Beam Xe (E = 3 GeV/n)

Total: 59.86 MEvents





RAW → ***DIGIT*** → ***DSTexp*** → PhA

RAW: raw (binary) event data collected by the DAQ system after the Event Builder

DIGIT: detector readings (event digits) after the digitizer macro

DSTexp: reconstructed data of experimental events

Experimental data

645 x 10^6 events

(25 800 raw files)

1 raw file = 15 GB (25 000 events)
1 digit file ≈ 870 MB
1 dst file ≈ 2 000 MB

GEN → SIM → ***DSTsim*** → PhA

GEN: particle collisions description received by an event generator

DSTsim: reconstructed data of simulated events

DMS Tasks

- Provide dataset oriented, secure access to data
- Managing data:
  - Transfer data to/from/between sites
  - Delete data from sites
- Ensure data consistency at sites
- Workflow integration

RUCIO

Rucio was originally developed to meet the requirements of the high-energy physics experiment ATLAS

Rucio now is continuously extended to support the LHC experiments and other diverse scientific communities.

- Highly scalable
- Policy driven
- Good for big amount of data
- Automated Data Rebalancing

DIRAC

DIRAC
THE INTERWARE

An open source middleware for distributed computing

- Started as an LHCb project.
- Experiment-agnostic since 2009.
- Developed by communities, for communities.
- Workload management system integrated
- Publicly documented, active assistance forum, yearly users workshops, open developers meetings and hackathons, already in JINR

DIRAC File Catalog

LFNs (Logical File Name): unique identifier of a file within DIRAC.

LFNs may have physical replicas, stored in SEs.

LFNs are registered in catalog(s). There exist multiple implementations of catalogs. Several of them can live in parallel:

- DIRAC File Catalog: full replica and metadata catalog.
- DMS integrates FTS3 to schedule and monitor efficient transfer of large amounts of data between SEs.

DFC: metadata

DFC is Replica and Metadata Catalog

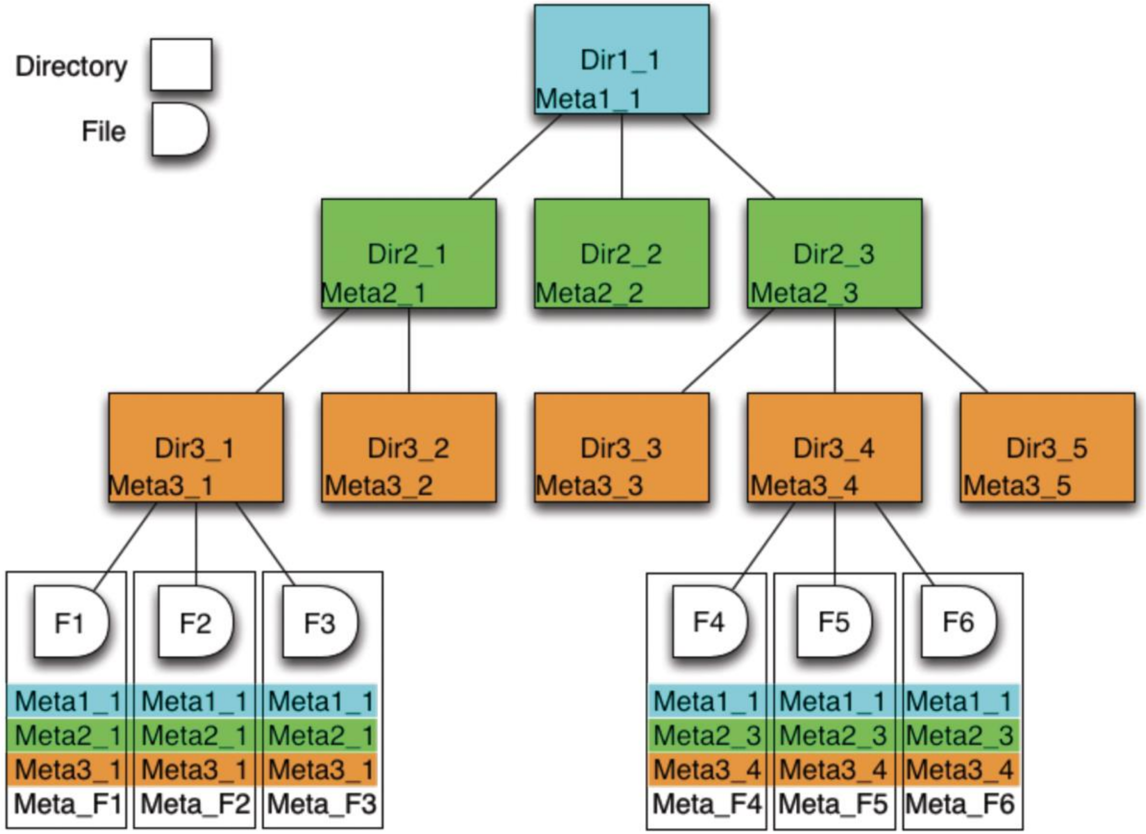
- User defined metadata
- The same hierarchy for metadata as for the logical name space
- Metadata associated with files and directories
- Allow for efficient searches

Efficient Storage Usage reports

- Suitable for user quota management

Stored ancestor/successor file relations

- Simple provenance catalog

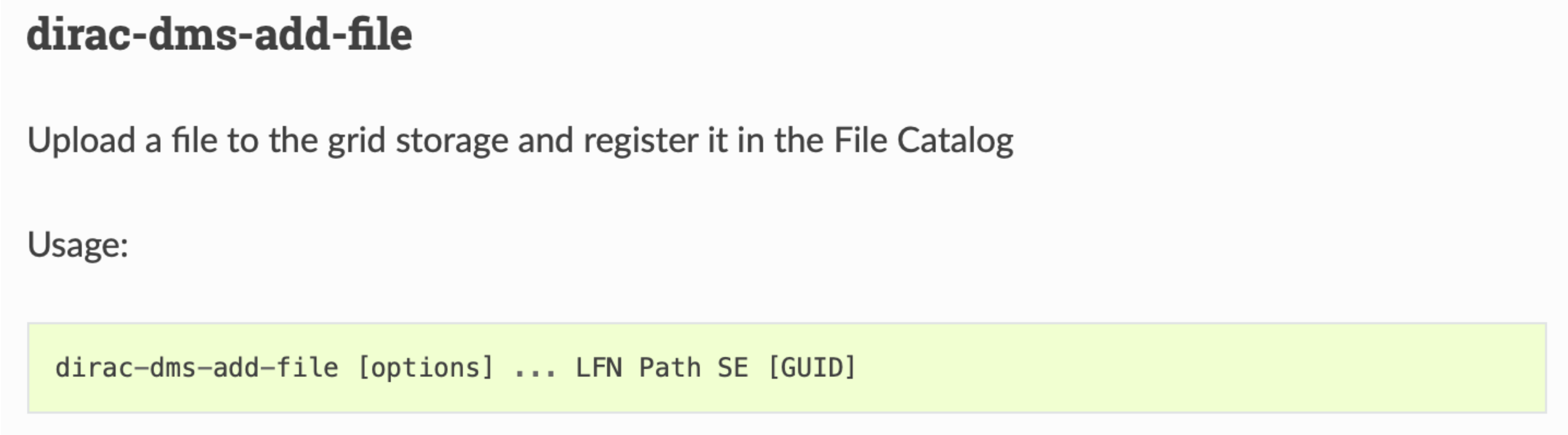


Directory
File
Dir1_1 Meta1_1
Dir2_1 Meta2_1
Dir2_2 Meta2_2
Dir2_3 Meta2_3
Dir3_1 Meta3_1
Dir3_2 Meta3_2
Dir3_3 Meta3_3
Dir3_4 Meta3_4
Dir3_5 Meta3_5
F1
Meta1_1 Meta2_1 Meta3_1 Meta_F1
F2
Meta1_1 Meta2_1 Meta3_1 Meta_F2
F3
Meta1_1 Meta2_1 Meta3_1 Meta_F3
F4
Meta1_1 Meta2_3 Meta3_4 Meta_F4
F5
Meta1_1 Meta2_3 Meta3_4 Meta_F5
F6
Meta1_1 Meta2_3 Meta3_4 Meta_F6

Current metadata

period_number	**-**	**INTEGER**
run_number	**-**	**INTEGER**
run_type	**-**	**SMALLINT**
start_datetime	**-**	**TIMESTAMP**
end_datetime	**-**	**TIMESTAMP**
beam_particle	**-**	**VARCHAR**
target_particle	**-**	**VARCHAR**
energy	**-**	**FLOAT**
field_voltage	**-**	**FLOAT**
start_event	**-**	**INTEGER**
end_event	**-**	**INTEGER**
event_count	**-**	**INTEGER**
file_size	**-**	**LONG**

DFC through: command line

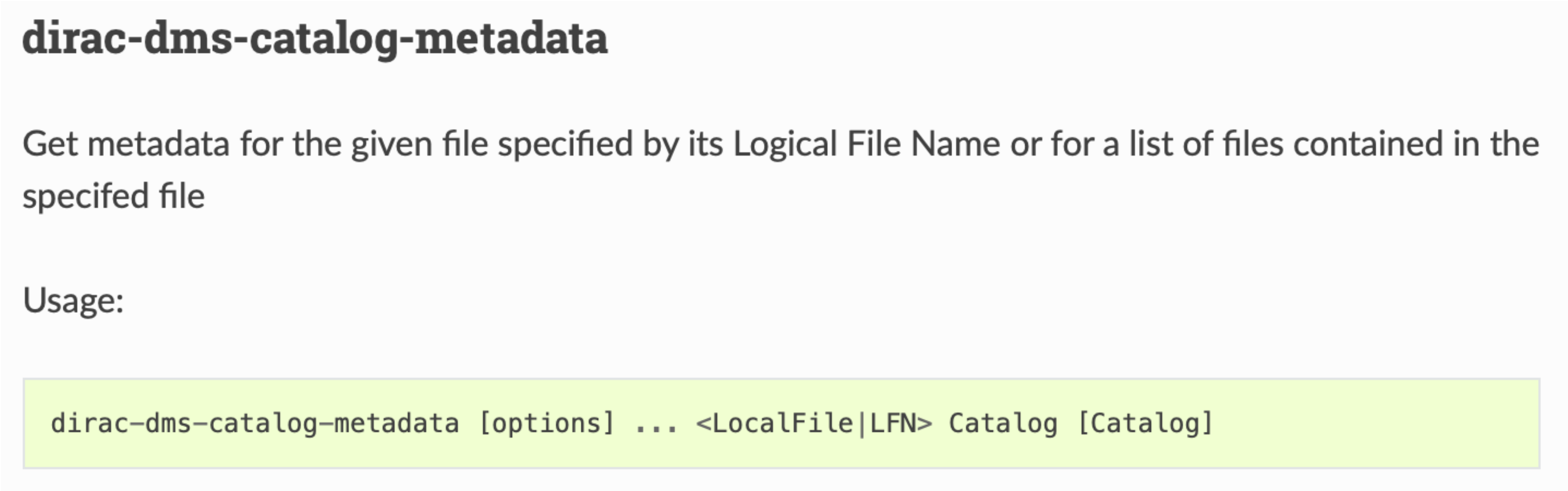


dirac-dms-add-file

Upload a file to the grid storage and register it in the File Catalog

Usage:

```
dirac-dms-add-file [options] ... LFN Path SE [GUID]
```



dirac-dms-catalog-metadata

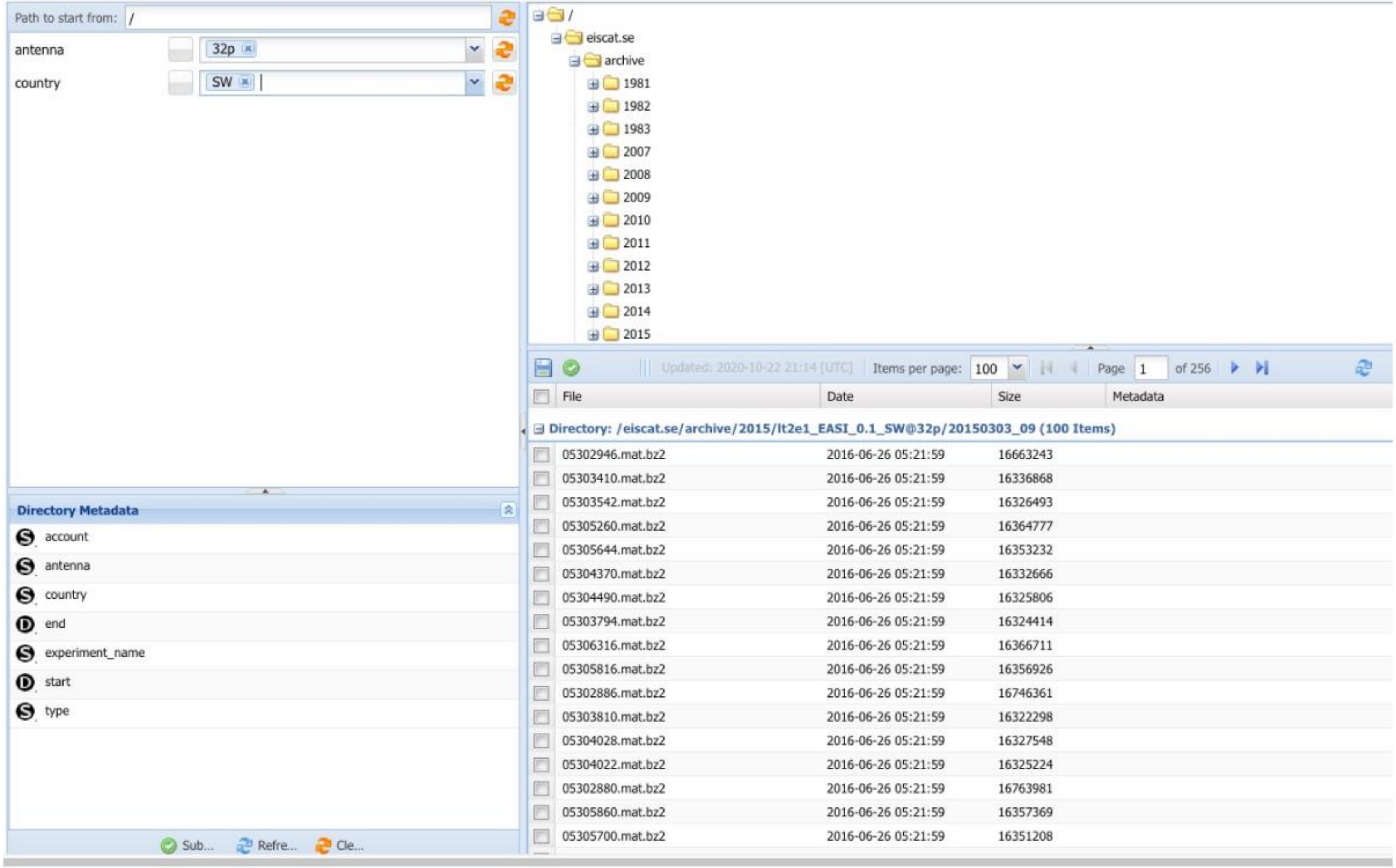
Get metadata for the given file specified by its Logical File Name or for a list of files contained in the specifed file

Usage:

```
dirac-dms-catalog-metadata [options] ... <LocalFile|LFN> Catalog [Catalog]
```

• • •

DFC through: web interface



Path to start from: /

antenna: 32p

country: SW

- /
  - eiscat.se
    - archive
      - 1981
      - 1982
      - 1983
      - 2007
      - 2008
      - 2009
      - 2010
      - 2011
      - 2012
      - 2013
      - 2014
      - 2015

Updated: 2020-10-22 21:14 [UTC]

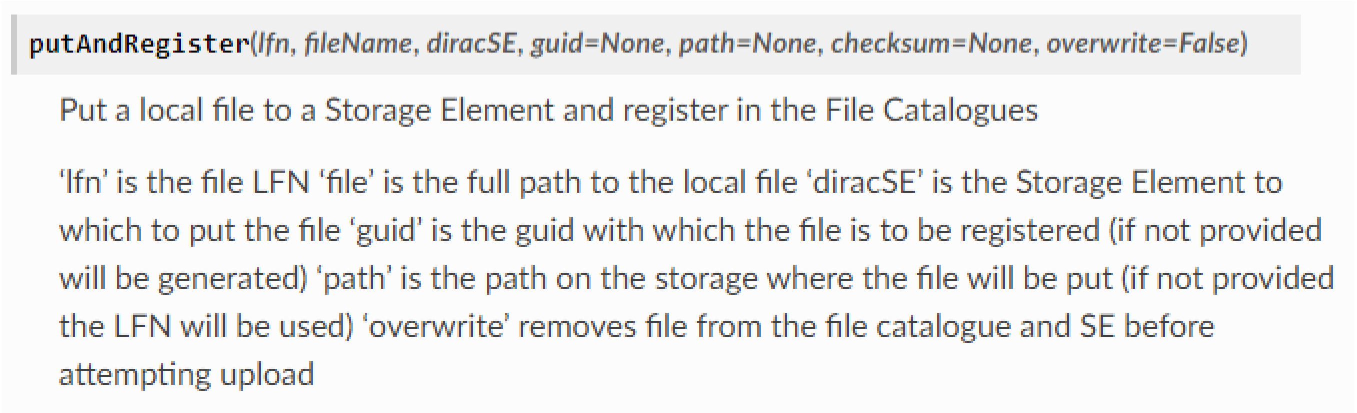
Items per page: 100 Page 1 of 256

File	Date	Size	Metadata
Directory: /eiscat.se/archive/2015/lt2e1_EASI_0.1_SW@32p/20150303_09 (100 Items)			
05302946.mat.bz2	2016-06-26 05:21:59	16663243	
05303410.mat.bz2	2016-06-26 05:21:59	16336868	
05303542.mat.bz2	2016-06-26 05:21:59	16326493	
05305260.mat.bz2	2016-06-26 05:21:59	16364777	
05305644.mat.bz2	2016-06-26 05:21:59	16353232	
05304370.mat.bz2	2016-06-26 05:21:59	16332666	
05304490.mat.bz2	2016-06-26 05:21:59	16325806	
05303794.mat.bz2	2016-06-26 05:21:59	16324414	
05306316.mat.bz2	2016-06-26 05:21:59	16366711	
05305816.mat.bz2	2016-06-26 05:21:59	16356926	
05302886.mat.bz2	2016-06-26 05:21:59	16746361	
05303810.mat.bz2	2016-06-26 05:21:59	16322298	
05304028.mat.bz2	2016-06-26 05:21:59	16327548	
05304022.mat.bz2	2016-06-26 05:21:59	16325224	
05302880.mat.bz2	2016-06-26 05:21:59	16763981	
05305860.mat.bz2	2016-06-26 05:21:59	16357369	
05305700.mat.bz2	2016-06-26 05:21:59	16351208	

Directory Metadata

S	account
S	antenna
S	country
D	end
S	experiment_name
D	start
S	type

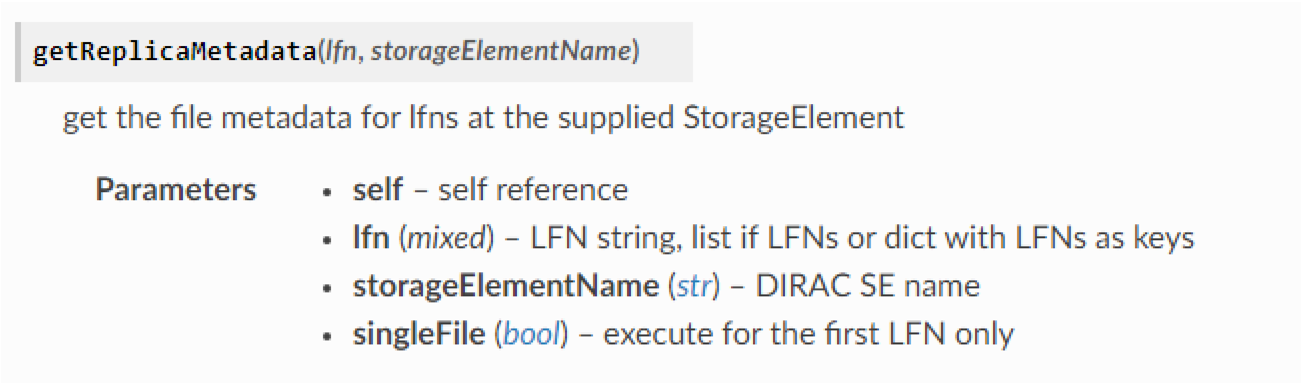
DFC through: python API



putAndRegister(*lfn, fileName, diracSE, guid=None, path=None, checksum=None, overwrite=False*)

Put a local file to a Storage Element and register in the File Catalogues

'lfn' is the file LFN 'file' is the full path to the local file 'diracSE' is the Storage Element to which to put the file 'guid' is the guid with which the file is to be registered (if not provided will be generated) 'path' is the path on the storage where the file will be put (if not provided the LFN will be used) 'overwrite' removes file from the file catalogue and SE before attempting upload

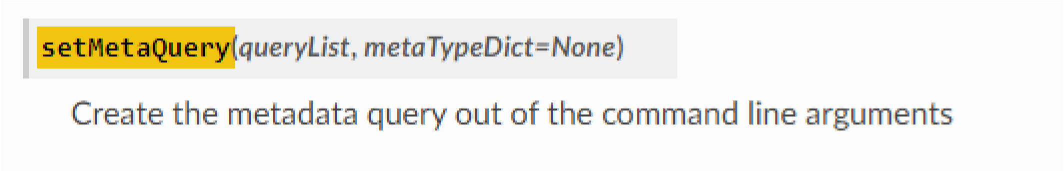


getReplicaMetadata(*lfn, storageElementName*)

get the file metadata for lfns at the supplied StorageElement

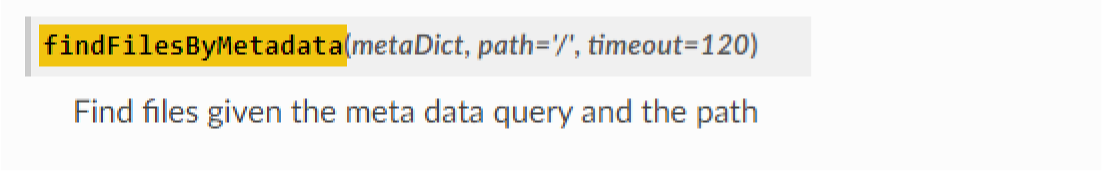
Parameters

- **self** – self reference
- **lfn** (*mixed*) – LFN string, list if LFNs or dict with LFNs as keys
- **storageElementName** (*str*) – DIRAC SE name
- **singleFile** (*bool*) – execute for the first LFN only



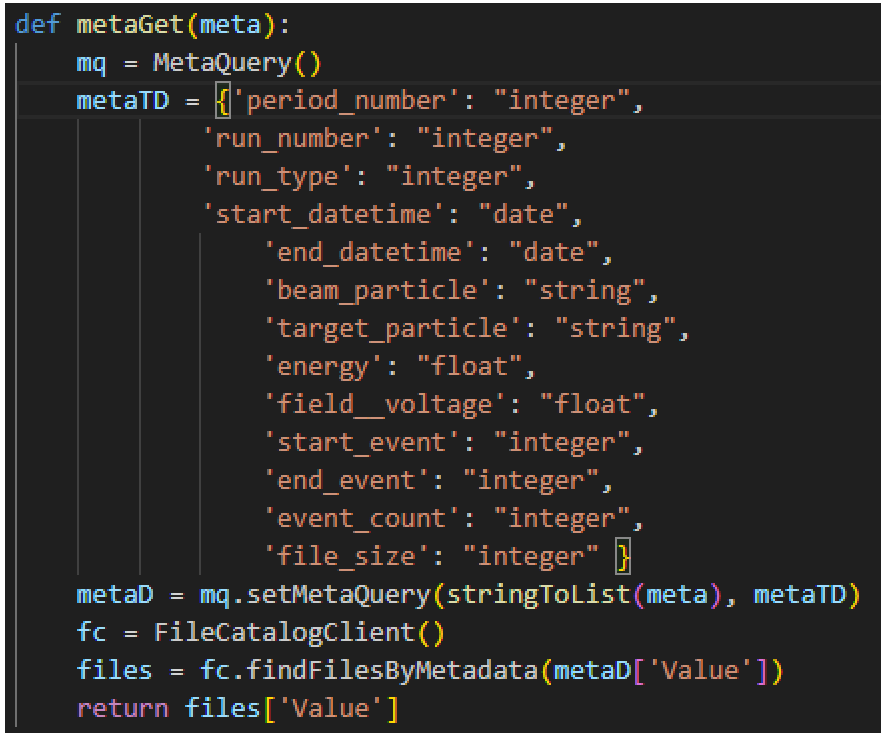
setMetaQuery(*queryList, metaTypeDict=None*)

Create the metadata query out of the command line arguments



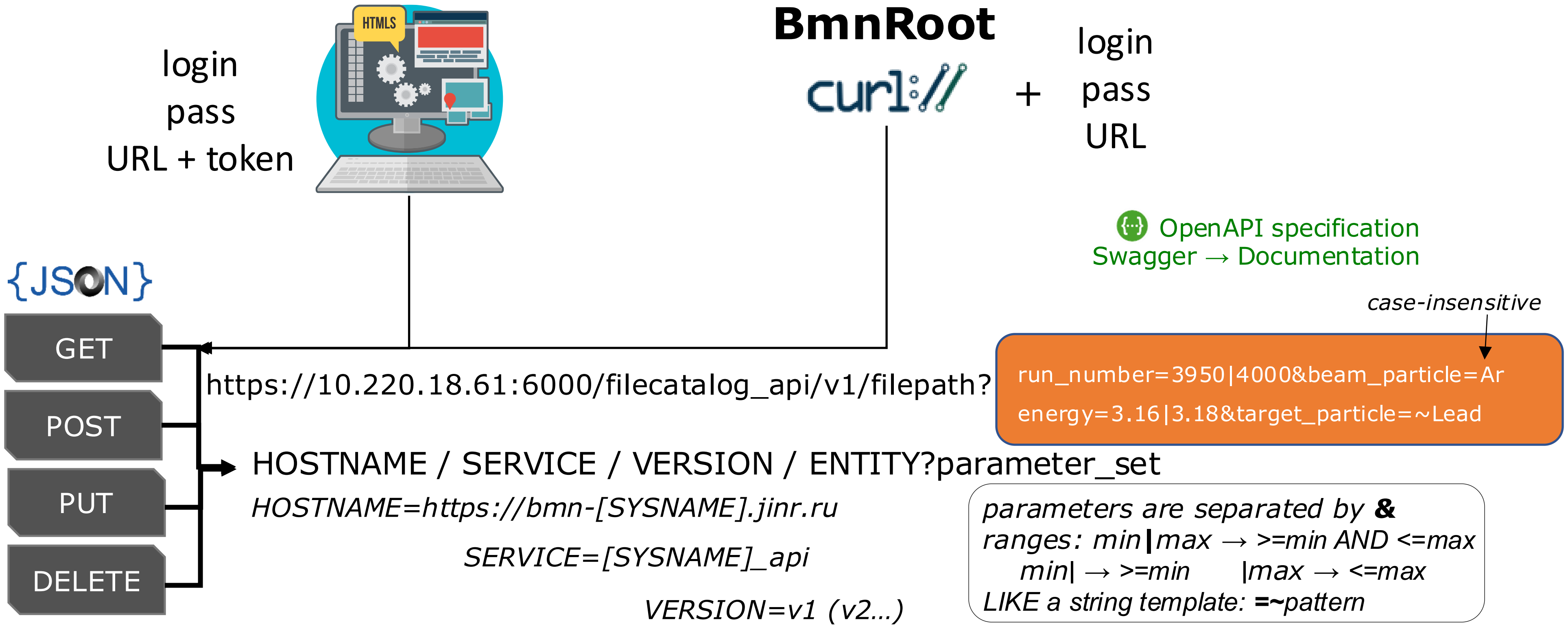
findFilesByMetadata(*metaDict, path='/', timeout=120*)

Find files given the meta data query and the path



```
def metaGet(meta):
    mq = MetaQuery()
    metaTD = {'period_number': "integer",
            'run_number': "integer",
            'run_type': "integer",
            'start_datetime': "date",
                'end_datetime': "date",
                'beam_particle': "string",
                'target_particle': "string",
                'energy': "float",
                'field__voltage': "float",
                'start_event': "integer",
                'end_event': "integer",
                'event_count': "integer",
                'file_size': "integer" }
    metaD = mq.setMetaQuery(stringToList(meta), metaTD)
    fc = FileCatalogClient()
    files = fc.findFilesByMetadata(metaD['Value'])
    return files['Value']
```

REST API



login
pass
URL + token

BmnRoot

+ login
pass
URL

OpenAPI specification
Swagger → Documentation

https://10.220.18.61:6000/filecatalog_api/v1/filepath?

case-insensitive

run_number=3950|4000&beam_particle=Ar
energy=3.16|3.18&target_particle=~Lead

HOSTNAME / SERVICE / VERSION / ENTITY?parameter_set

HOSTNAME=https://bmn-[SYSNAME].jinr.ru

SERVICE=[SYSNAME]_api

VERSION=v1 (v2...)

parameters are separated by **&**
*ranges: min***|***max → >=min AND <=max*
min| → >=min |max → <=max
LIKE a string template: **=~***pattern*

Unified Condition Database, SYSNAME = **uniconda**

Event Metadata System, SYSNAME = **event**

https://bmn-event.jinr.ru/event_api/v1/event?...
/eventFile?...
/eventFileRef?...

BM@N File Catalog, SYSNAME = **filecatalog** (prototype)

Keycloak

Open Source Identity and Access Management

Authorization using BM@N credentials

VM at 10.220.18.61

Credentials via CURL:

```
curl -u user:password -X GET
"10.220.18.61:6000/filecatalog_api/v1/meta?filepath=/bmn.nica.jinr/vo/test/someFile.data"
```

Request types

POST - add new metadata field

GET - get file list matching specific metadata

- get specific file metadata
- get all metadata fields
- is file exist
- get file catalog stats. (Number of files/directories/replicas etc.)
- get status of the last consistency check
- get result of the last consistency check

PUT - update file metadata

- run consistency check. Return check ID, status can be verified through get

DELETE - delete specific file metadata

- remove specific metadata field

Consistency check

- Basically goes through storages and compare existing files with registered ones in the File Catalog.
- The output is two counters and two vectors of paths to files that doesn’t match

```
Missing from EOS 991567:
/bmn.nica.jinr/exp/dst/run8/24.12.0/mpd_run_Top_7327_ev1_p7.root
/bmn.nica.jinr/exp/dst/run8/24.12.0/mpd_run_Top_7327_ev1_p8.root
/bmn.nica.jinr/exp/dst/run8/24.12.0/mpd_run_Top_7327_ev1_p9.root
/bmn.nica.jinr/exp/dst/run8/24.12.0/mpd_run_Top_7328_ev0_p0.root
. . .
Missing from FC 4:
/eos/nica/bmn/exp/digi/run8/25.04.0/mpd_run_Top_7797_ev1_p2.root
/eos/nica/bmn/exp/digi/run8/25.04.0/mpd_run_Top_7797_ev0_p5.root
/eos/nica/bmn/exp/digi/run8/25.04.0/mpd_run_Top_8106_ev0_p70.root
/eos/nica/bmn/exp/digi/run8/25.04.0/mpd_run_Top_7444_ev1_p14.root
```

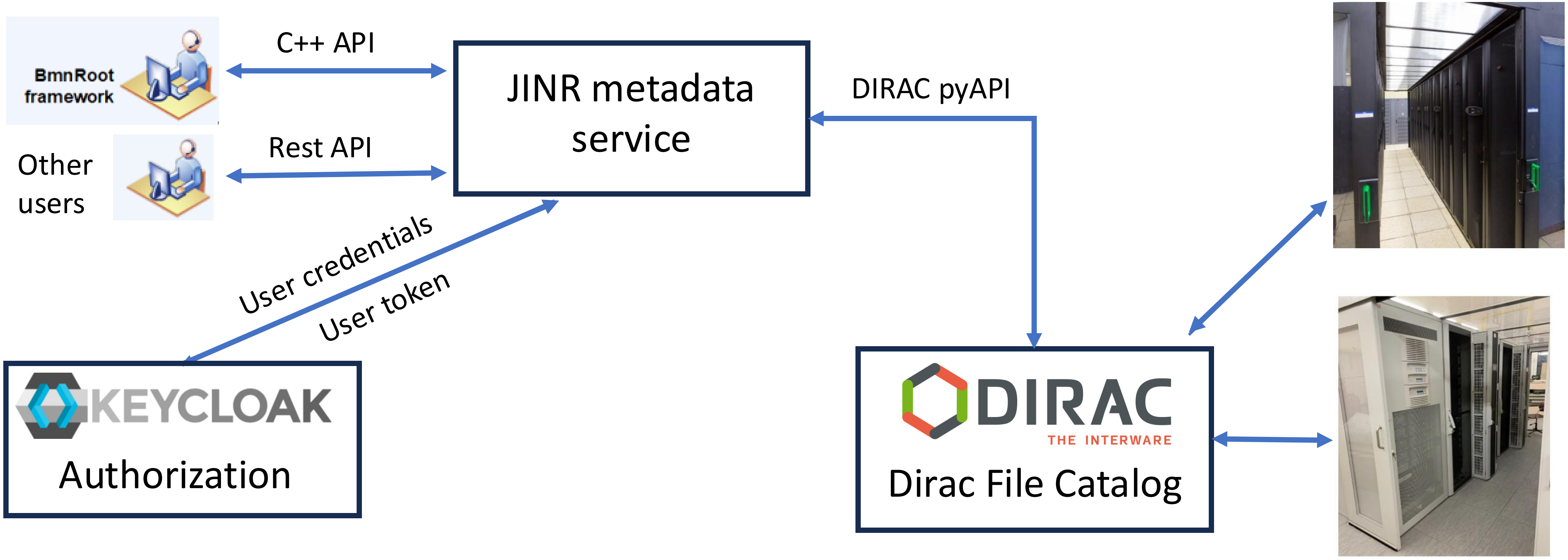
C++ API for BmnRoot

- Choose one of the **FileCatalog** class functions that best suits your needs. Like: *…GetFileList(…), GetFileInfo(…), UpdateFileInfo(…), DeleteMetadataField(…)*

Depending on use case:

- Prepare **FileInfo** object containing metadata info (*int RunNumber, string BeamParticle* etc.), or declare one if you need it as a result
- Define **FileCondition** (*less, greater, null, greaterOrEqual* etc.) to specify a metadata range for the selection request

DMS. Current state of development



BmnRoot framework

C++ API

Other users

Rest API

JINR metadata service

DIRAC pyAPI

User credentials

User token

KEYCLOAK

Authorization

DIRAC THE INTERWARE

Dirac File Catalog

What’s next

- Basic and crucial DMS operations with files itself (add, delete, get) with similar API.
- Some monitoring/logging services
- Web interface for viewing statistics and performing REST API requests

Спасибо!