

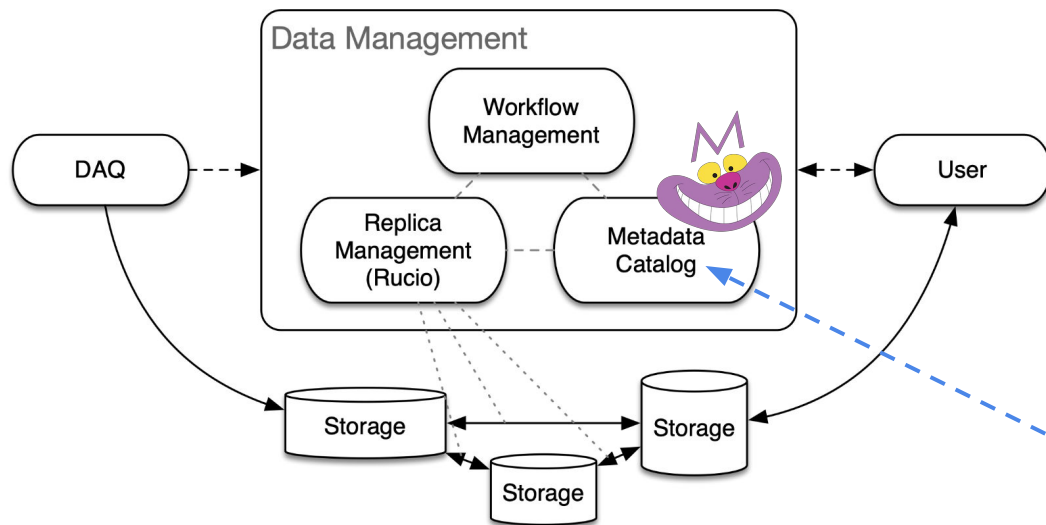


# MetaCat - metadata catalog for Rucio-based data management systems

Igor Mandrichenko

Rucio Workshop, 11/10/2022

# What is MetaCat ?



MetaCat = **Meta**data **Cat**alog  
for data management systems  
where Rucio can be used as  
the Replica Manager

# MetaCat Target Users

- Primary: DUNE
- Other FNAL experiments migrating from SAM
  - SAM is DM system used at FNAL, combining all 3 functions
- HEP experiments
- Rucio users

# Functions

- Store and make available metadata associated with objects (files) and object collections (datasets)
- Provide efficient query mechanism to select objects (files) matching the user criteria
- Provide flexible, efficient, integrated access to external metadata sources

# Conceptual Compatibility with Rucio

## Rucio:

- Scope
- Scope:Name (DID)
- Container
- Dataset

## MetaCat:

- Namespace
- Namespace:Name
- Dataset
  - In MetaCat, there is no distinction, a dataset can contain files and/or other datasets

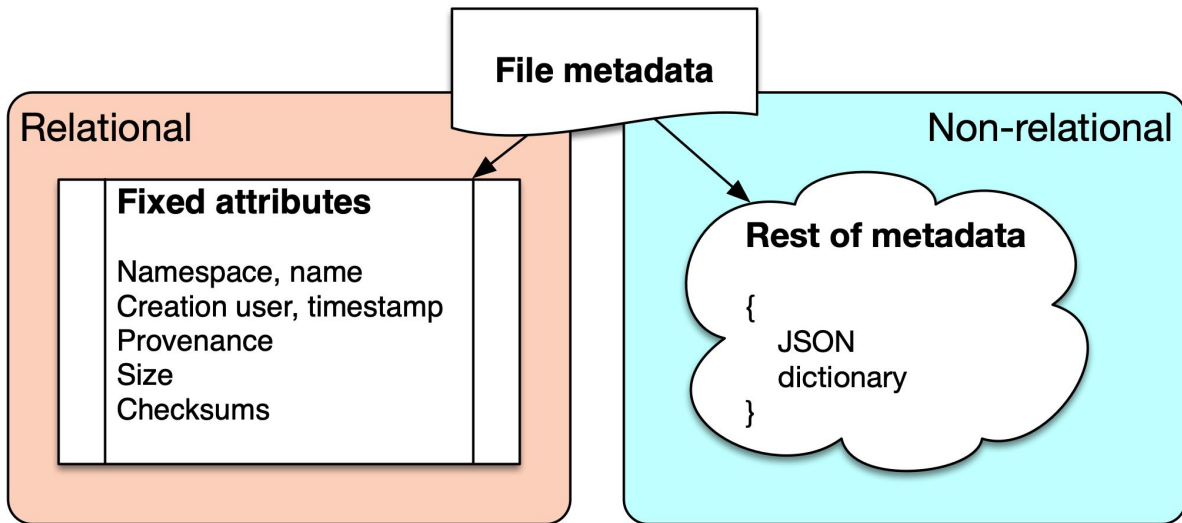
Being conceptually compatible, MetaCat does not depend on Rucio, nor does it communicate with Rucio directly, therefore can be used with other replica management systems

# Files or Objects

MetaCat unit of operation: file or *object* - abstract entity with the following properties

- Fixed attributes - every file has them
  - Unique text ID (assigned by user or auto generated)
    - Immutable
  - Unique name within a namespace (Rucio: scope, name)
    - Can be renamed
    - Name can be auto-generated
  - Creator username, timestamp
  - File Provenance
    - Parents, children (file A was created from files B, C, D)
- Rest of metadata - arbitrary JSON dictionary

# File Metadata



Rest of metadata:

- Application-defined
- Non-relational, fluid
- JSON dictionary - arbitrarily complex
- Restrictions can be defined via categories and/or dataset restrictions
- Implementation: Postgres jsonb type, GIN-indexed

Very few attributes are in relational schema

- Common attributes
- Attributes used by MetaCat itself
- Better indexing, fast lookup, table joins (datasets, namespaces, provenance, ...)

# Datasets

- Dataset has a name within a namespace
- Contains files
  - Many-to-many (a file can belong to many datasets)
  - Explicitly added/removed
- Combines Rucio dataset and container functionality
  - Datasets may contain other datasets, recursively (Rucio: container)
- Standard attributes:
  - Creator username, timestamp
  - Dataset flags
    - Frozen - files can not be added or removed
    - Monotonic - files can only be added
- Rest of metadata - arbitrary JSON dictionary



# Queries

- Querying is the fundamental function of MetaCat
  - Find all files matching a set of criteria expressed in terms of their metadata, provenance, external metadata
- Written in Metadata Query Language (MQL)
- A query can be named, saved and reused inside another query or as is
  - Conceptually similar to a relational database view

# Datasets vs Queries

Dataset - explicit collection of files

- Recorded in the database
- Files added/removed explicitly
- Has a name within a namespace

*Relational DB: table*

Query - *instructions* how to select files from a dataset or datasets

- Recalculated every time it runs
- Results can change at any time
- Can be saved under a name within a namespace and reused by name

*Relational DB: SQL “select”*

Bridge:

Query results can be saved as a new dataset or added to an existing dataset

# Metadata Query Language (MQL)

```
files from dune:raw
  where DUNE.reco_version = "v1.2"
  limit 1000
```

```
files from dune:raw_2019 where
  DUNE.reco_version in ("v1.2", "v1.3")
  and core.file_type = "root"
  or DUNE.reco_version = "v1.0"
```

```
union (
  files from dune:raw_2019
    where DUNE.reco_version = "v1.2"
  ,
  files from dune:raw_2020
    where DUNE.detector = "near" and
      DUNE.reco_version >= "v1.3"
) where data.format in ("root", "hdf5")
```

- Keyword - files query
- Dataset to select files from (DID)
- Metadata filtering
- Limit results to first 1000 files
- Parameter category
- Boolean algebra
- Queries can be combined using "union", "join", "-" (subtraction)
- Metadata filters can be applied again to the combined query

# MQL Compiler

- MQL query is compiled into SQL query and executed by the database engine
  - Exception: external metadata access - executed by the MetaCat application server
- Resulting SQL query complexity in terms of number of table joins *does not* depend on the complexity of the original MQL query
- MQL metadata expressions are compiled into JSON/JSON Path expressions interpreted by Postgres

# Parameter Categories

A mechanism to restrict the fluidity of the metadata schema in application-specific way

MetaCat parameter name:

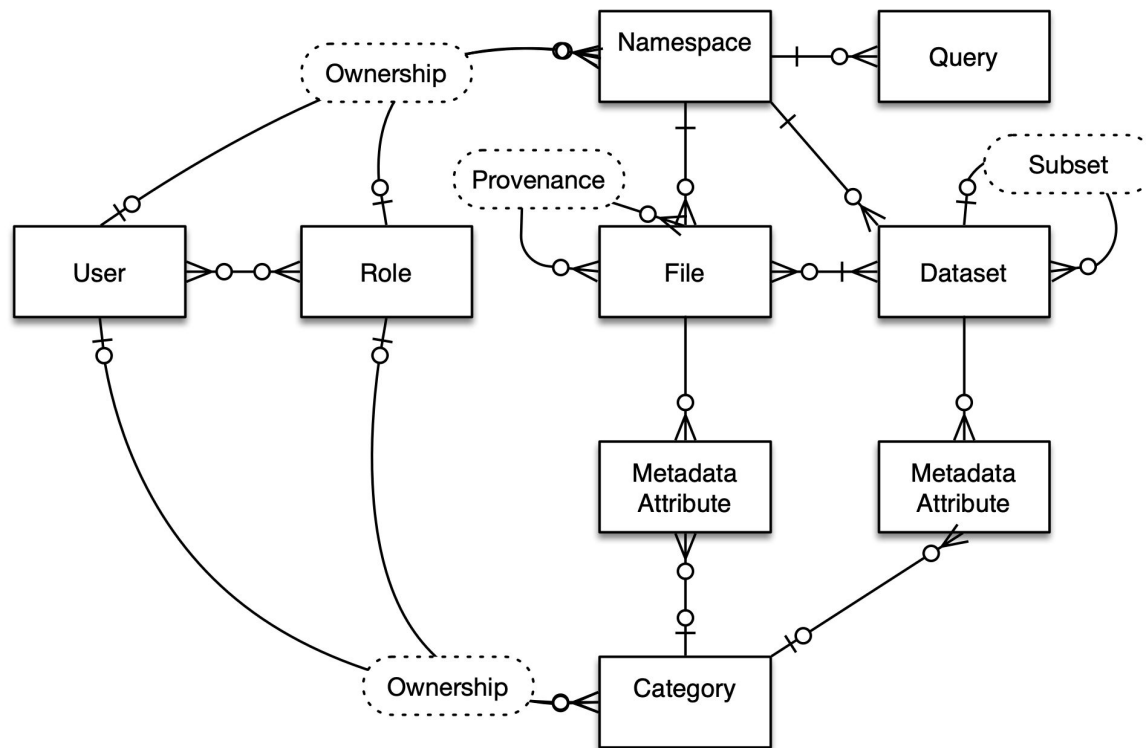
`<category>.<parameter name>`

Category owner can restrict areas of the metadata namespace

- Parameter types
  - Int, float, string, boolean, list of ints, floats, ... dict, ...
- Accepted values
  - Range, enumeration, pattern
- Restricted category: only known parameter names are allowed

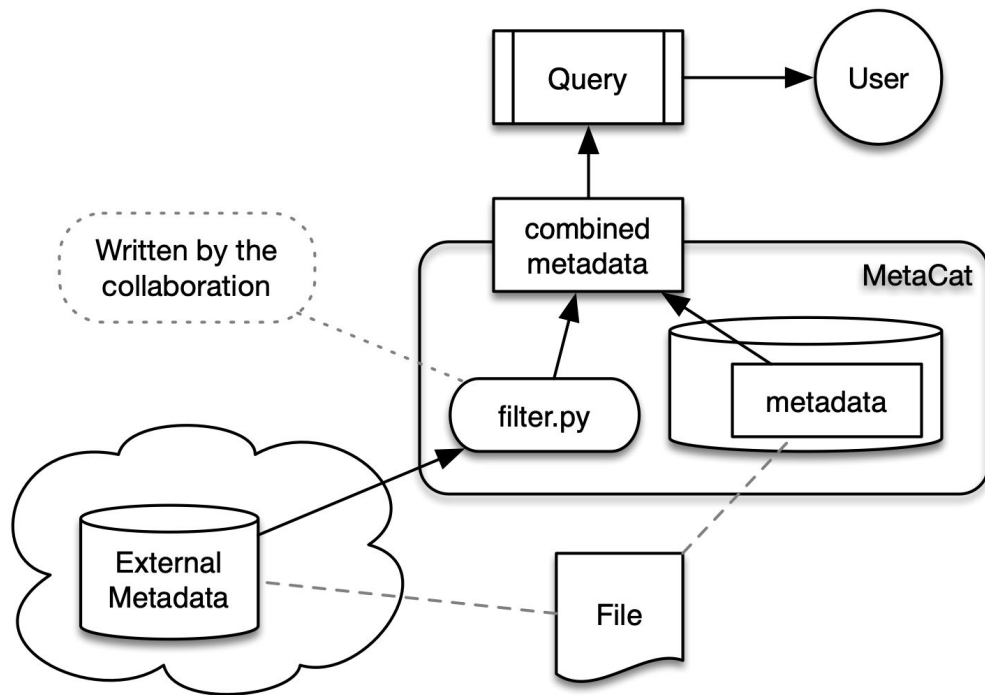
Enforced at the time of the file declaration

# Data Model



MetaCat Data Model

# External Metadata Sources



Use case:

- Run conditions are stored in the Runs database by run number
- Files need to be selected based on some run conditions values
- We do not want to duplicate run conditions data in MetaCat as file metadata
- Implemented for ProtoDUNE

# External Filter in MQL

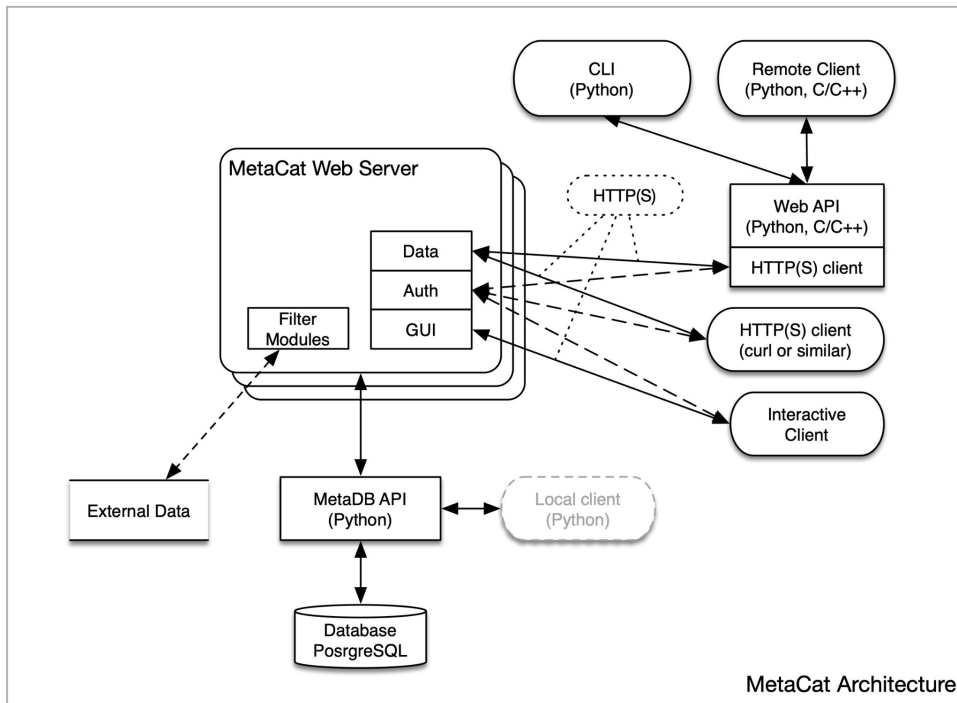
```
# real life DUNE example

filter rucio_replicas() (
    files from dc4:dc4
    limit 100
) where "DUNE_CERN_EOS" in rucio.rses
```

- **Filter name** - collaboration defined
- Filter is applied to the results of the **Intermediate query**
- This filter contacts Rucio and gets replica information for the given files
- Injects the replica information as **new metadata** making it available for querying and as the query output
- Makes the replica location information appear as if it is stored in MetaCat, but it is not



# MetaCat Architecture



## Software Stack

- PostgreSQL v12 - the database
- Python3 (both client and server)
- psycopg2 - Python/PostgreSQL


# Client API (Python)

- Uses HTTP/HTTPS, **requests** Python library
- Datasets - create, get, list, add files, remove files, update metadata, ...
- Namespaces - create, get, list, ...
- Files - declare, get, update metadata, provenance, ...
- Query - run, run asynchronously, save results as dataset, add results to dataset
- Parameter categories, validation
- Authentication
  - JWT tokens

# Command Line Interface Functions

- Client authentication
  - Log in (obtain JWT token, save it in local FS, similar to Rucio)
- Datasets
  - Create, list, show, update
- Namespaces
  - Create, list, show
- Files
  - Declare, show, add, update
- Metadata validation
- Parameter categories
- Queries

# MetaCat GUI



## DUNE/ProtoDUNE MetaCat

[users](#) [roles](#) [namespaces](#) [datasets](#) [find file](#) [categories](#) [groups](#)

### Datasets

[create](#)

| Dataset  | Creator                  | Created      |
|--|--------------------------|--------------|
| <a href="#">amcnab_test:amcnab-test02</a>                              | <a href="#">amcnab</a>   | 2022-08-13 1 |
| <a href="#">calcuttj_test:calcuttj_test</a>                            | <a href="#">calcuttj</a> | 2022-09-08 1 |
| <a href="#">dc4:dc4</a>  | <a href="#">dunepro</a>  | 2022-06-10 1 |
| <a href="#">dc4-hd-protodune:output-test-01</a>                        | <a href="#">dunepro</a>  | 2022-09-05 0 |
| <a href="#">dc4-interactive-tests:dd-interactive-test-data</a>         | <a href="#">calcuttj</a> | 2022-09-21 1 |
| <a href="#">dc4-interactive-tests-dunepro:dd-interactive-test-data</a> | <a href="#">calcuttj</a> | 2022-09-21 1 |
| <a href="#">dc4-test:amcnab-01</a>                                     | <a href="#">amcnab</a>   | 2022-08-16 0 |
| <a href="#">dc4-vd-coldbox-bottom:output-test-01</a>                   | <a href="#">dunepro</a>  | 2022-09-05 0 |
| <a href="#">dc4-vd-coldbox-bottom:wfs-25pc-01-data</a>                 | <a href="#">dunepro</a>  | 2022-09-15 0 |
| <a href="#">dc4-vd-coldbox-bottom:wfs-25pc-01-logs</a>                 | <a href="#">dunepro</a>  | 2022-09-15 0 |
| <a href="#">dc4-vd-coldbox-bottom:wfs-5pc-01-data</a>                  | <a href="#">dunepro</a>  | 2022-09-05 1 |

### Metadata Query

Load: no saved queries

Default namespace:

Query: 

```
files from dc4:dc4
where
  core.run_type = "dc4-vd-coldbox-bottom"
and 3221 in core.events
```

Save query as: Namespace:  Name:

Include metadata: ☒ view metadata as:

Re-run and save results as dataset: ☐ Namespace:  Name:

### File Information

File ID: 0047810b9725451d9deb0a402a6c9714

Namespace: [dc4-vd-coldbox-bottom](#)

Name: dc4\_np02bde\_306091554\_np02\_bde\_coldbox\_run012352\_0020\_20211215T223238.hdf5

DID: dc4-vd-coldbox-bottom:dc4\_np02bde\_306091554\_np02\_bde\_coldbox\_run012352\_0020\_20211215T223238.hdf5

Size: 4284852088 (4086.353 MB)

Created at: 2022-06-13 12:43:47.718178-05:00

Created by:

Checksums: Adler32: ae8597a9

Parents:

Children:

### Metadata

| Parameter        | Value  |
|------------------|--|
| DUNE.campaign    | dc4  |
| core.data_stream | Test   |
| core.data_tier   | raw  |
| core.event_count | 161  |
| core.events      | [3221, 3222, 3223, 3224, 3225, 3226, 3227, 3228, 3229, 3230, 3231, 3232, 3233, 3234, 3235, 3236, 3237, 3238, 3239, 3240, 3241, 3242, 3243, 3244, 3245, 3246, 3247, 3248, 3249, 3250, 3251, 3252, 3253, 3254, 3255, 3256, 3257, 3258, 3259, 3260, 3261, 3262, 3263, 3264, 3265, 3266, 3267, 3268, 3269, 3270, 3271, 3272, 3273, 3274, 3275, 3276, 3277, 3278, 3279, 3280, 3281, 3282, 3283, 3284, 3285, 3286, 3287, 3288, 3289, 3290, 3291, 3292, 3293, 3294, 3295, 3296, 3297, 3298, 3299, 3300, 3301, 3302, 3303, 3304, 3305, 3306, 3307, 3308, 3309, 3310, 3311, 3312, 3313, 3314, 3315, 3316, 3317, 3318, 3319, 3320, 3321, 3322, 3323, 3324, 3325, 3326, 3327, 3328, 3329, 3330, 3331, 3332, 3333, 3334, 3335, 3336, 3337, 3338, 3339, 3340, 3341, 3342, 3343, 3344, 3345, 3346, 3347, 3348, 3349, 3350, 3351, 3352, 3353, 3354, 3355, 3356, 3357, 3358, 3359, 3360, 3361, 3362, 3363, 3364, 3365, 3366, 3367, 3368, 3369, 3370, 3371, 3372, 3373, 3374, 3375, 3376, 3377, 3378, 3379, 3380, 3381, 3382, 3383, 3384, 3385, 3386, 3387, 3388, 3389, 3390, 3391, 3392, 3393, 3394, 3395, 3396, 3397, 3398, 3399, 3400, 3401, 3402, 3403, 3404, 3405, 3406, 3407, 3408, 3409, 3410, 3411, 3412, 3413, 3414, 3415, 3416, 3417, 3418, 3419, 3420, 3421, 3422, 3423, 3424, 3425, 3426, 3427, 3428, 3429, 3430, 3431, 3432, 3433, 3434, 3435, 3436, 3437, 3438, 3439, 3440, 3441, 3442, 3443, 3444, 3445, 3446, 3447, 3448, 3449, 3450, 3451, 3452, 3453, 3454, 3455, 3456, 3457, 3458, 3459, 3460, 3461, 3462, 3463, 3464, 3465, 3466, 3467, 3468, 3469, 3470, 3471, 3472, 3473, 3474, 3475, 3476, 3477, 3478, 3479, 3480, 3481, 3482, 3483, 3484, 3485, 3486, 3487, 3488, 3489, 3490, 3491, 3492, 3493, 3494, 3495, 3496, 3497, 3498, 3499, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511, 3512, 3513, 3514, 3515, 3516, 3517, 3518, 3519, 3520, 3521, 3522, 3523, 3524, 3525, 3526, 3527, 3528, 3529, 3530, 3531, 3532, 3533, 3534, 3535, 3536, 3537, 3538, 3539, 3540, 3541, 3542, 3543, 3544, 3545, 3546, 3547, 3548, 3549, 3550, 3551, 3552, 3553, 3554, 3555, 3556, 3557, 3558, 3559, 3560, 3561, 3562, 3563, 3564, 3565, 3566, 3567, 3568, 3569, 3570, 3571, 3572, 3573, 3574, 3575, 3576, 3577, 3578, 3579, 3580, 3581, 3582, 3583, 3584, 3585, 3586, 3587, 3588, 3589, 3590, 3591, 3592, 3593, 3594, 3595, 3596, 3597, 3598, 3599, 3600, 3601, 3602, 3603, 3604, 3605, 3606, 3607, 3608, 3609, 3610, 3611, 3612, 3613, 3614, 3615, 3616, 3617, 3618, 3619, 3620, 3621, 3622, 3623, 3624, 3625, 3626, 3627, 3628, 3629, 3630, 3631, 3632, 3633, 3634, 3635, 3636, 3637, 3638, 3639, 3640, 3641, 3642, 3643, 3644, 3645, 3646, 3647, 3648, 3649, 3650, 3651, 3652, 3653, 3654, 3655, 3656, 3657, 3658, 3659, 3660, 3661, 3662, 3663, 3664, 3665, 3666, 3667, 3668, 3669, 3670, 3671, 3672, 3673, 3674, 3675, 3676, 3677, 3678, 3679, 3680, 3681, 3682, 3683, 3684, 3685, 3686, 3687, 3688, 3689, 3690, 3691, 3692, 3693, 3694, 3695, 3696, 3697, 3698, 3699, 3700, 3701, 3702, 3703, 3704, 3705, 3706, 3707, 3708, 3709, 3710, 3711, 3712, 3713, 3714, 3715, 3716, 3717, 3718, 3719, 3720, 3721, 3722, 3723, 3724, 3725, 3726, 3727, 3728, 3729, 3730, 3731, 3732, 3733, 3734, 3735, 3736, 3737, 3738, 3739, 3740, 3741, 3742, 3743, 3744, 3745, 3746, 3747, 3748, 3749, 3750, 3751, 3752, 3753, 3754, 3755, 3756, 3757, 3758, 3759, 3760, 3761, 3762, 3763, 3764, 3765, 3766, 3767, 3768, 3769, 3770, 3771, 3772, 3773, 3774, 3775, 3776, 3777, 3778, 3779, 3780, 3781, 3782, 3783, 3784, 3785, 3786, 3787, 3788, 3789, 3790, 3791, 3792, 3793, 3794, 3795, 3796, 3797, 3798, 3799, 3800, 3801, 3802, 3803, 3804, 3805, 3806, 3807, 3808, 3809, 3810, 3811, 3812, 3813, 3814, 3815, 3816, 3817, 3818, 3819, 3820, 3821, 3822, 3823, 3824, 3825, 3826, 3827, 3828, 3829, 3830, 3831, 3832, 3833, 3834, 3835, 3836, 3837, 3838, 3839, 3840, 3841, 3842, 3843, 3844, 3845, 3846, 3847, 3848, 3849, 3850, 3851, 3852, 3853, 3854, 3855, 3856, 3857, 3858, 3859, 3860, 3861, 3862, 3863, 3864, 3865, 3866, 3867, 3868, 3869, 3870, 3871, 3872, 3873, 3874, 3875, 3876, 3877, 3878, 3879, 3880, 3881, 3882, 3883, 3884, 3885, 3886, 3887, 3888, 3889, 3890, 3891, 3892, 3893, 3894, 3895, 3896, 3897, 3898, 3899, 3900, 3901, 3902, 3903, 3904, 3905, 3906, 3907, 3908, 3909, 3910, 3911, 3912, 3913, 3914, 3915, 3916, 3917, 3918, 3919, 3920, 3921, 3922, 3923, 3924, 3925, 3926, 3927, 3928, 3929, 3930, 3931, 3932, 3933, 3934, 3935, 3936, 3937, 3938, 3939, 3940, 3941, 3942, 3943, 3944, 3945, 3946, 3947, 3948, 3949, 3950, 3951, 3952, 3953, 3954, 3955, 3956, 3957, 3958, 3959, 3960, 3961, 3962, 3963, 3964, 3965, 3966, 3967, 3968, 3969, 3970, 3971, 3972, 3973, 3974, 3975, 3976, 3977, 3978, 3979, 3980, 3981, 3982, 3983, 3984, 3985, 3986, 3987, 3988, 3989, 3990, 3991, 3992, 3993, 3994, 3995, 3996, 3997, 3998, 3999, 4000] |

### Query results

Runtime: 0.658 seconds  
Files: 657 (results in GUI are always limited to 1000 results)  
JSON: [with](#) [without](#) metadata

+ Metadata values distribution

### Files

dc4-vd-coldbox-bottom:dc4\_np02bde\_306091554\_np02\_bde\_coldbox\_run012352\_0020\_20211215T223238.hdf5 (0047810b9725451d9deb0a402a6c9714)

|                  |  |
|------------------|--|
| DUNE.campaign    | "dc4"  |
| core.data_stream | "Test"   |
| core.data_tier   | "raw"  |
| core.event_count | 161  |
| core.events      | [3221, 3222, 3223, 3224, 3225, 3226, 3227, 3228, 3229, 3230, 3231, 3232, 3233, 3234, 3235, 3236, 3237, 3238, 3239, |

[https://metacat.fnal.gov:9443/dune\\_meta\\_prod/app/gui/index](https://metacat.fnal.gov:9443/dune_meta_prod/app/gui/index)

# Existing Databases

- DUNE/ProtoDUNE
  - 16.7 M files
  - ~480 M name-value metadata pairs
  - 21 GB “files” table + 6 GB metadata index over non-relational JSON data
  - Total database size: ~40 GB
    - ~ 2.5 KB/file
  - Passed the data challenge, plan is to make it official production soon
- NOvA - *not used, but imported to test scalability*
  - 191.2 M files
  - ~5.3 B name-value metadata pairs
  - 221 GB “files” table + 35 GB metadata index over non-relational JSON data
  - Total database size: ~326 GB
    - ~ 1.7 KB/file

# References

Documentation: <https://metacat.readthedocs.io>

vCHEP 2021 paper:

<https://cdcv.s.fnal.gov/redmine/attachments/download/64700/MetaCat%20CHEP%202021%20paper%20v5.pdf>

DUNE MetaCat GUI:

[https://metacat.fnal.gov:9443/dune\\_meta\\_prod/app/gui/index](https://metacat.fnal.gov:9443/dune_meta_prod/app/gui/index)

# Backup

# MQL syntax: file queries

```
<file query>: files [from [dataset|datasets] <dataset selector list> [,...]]  
    | <file query> [where <metadata expression>]  
    | <file query> [skip <integer>]  
    | <file query> [limit <integer>]  
    | query <saved query namespace>:<saved query name>  
    | filter <filter name>( <parameter> [,...] ) ( <file query> [,...] )  
    | union ( <file query> [,...] )  
    | join ( <file query> [,...] )  
    | <file query> - <file query>  
    | children ( <file query> )                # file provenance  
    | parents ( <file query> )  
    | ( <file query> )
```



# MQL syntax: metadata expressions

```
<metadata expression>: <scalar> <cmp op> <constant>
    | <attribute> [[not] present]                # file has this attr
    | <constant> [not] in <attribute>            # in list or dict
    | <scalar> [not] in <constant> : <constant>  # range
    | <scalar> [not] in ( <constant> [,...] )    # enumeration
    | ( <metadata expression > )
    | ! <metadata expression>
    | <metadata expression> and <metadata expression>
    | <metadata expression> or <metadata expression>

<cmp op>: = == != < <= > >= ~ ~* !~ !~*
```

# Named Queries

```
query DUNE:supernova_production_latest_version
  where len(core.events) > 10
  limit 100
```

```
join (
  query DUNE:supernova_production_latest_version,
  query joe:my_favorite_files,
  files from dune:all
  where
    run.quality > 10
    and core.runs[any] in 7375:7380
)
```

A query could be created,  
debugged and saved to be  
reused by name

# Datasets and Subsets in MQL

```
files
  from dune:raw_2019
    with children
    recursively
  where
    created_timestamp > '2019-05-01'
    and reco_version = "v1.2"
```

Include files from the top dataset

- and its subsets
- recursively

# Other ways to use filters

```
filter random_mix(0.4, 0.6)
(
    files from dune:raw_2019                # file set 1
        where reco_version = "v1.2",
    files from dune:raw_2020                # file set 2
        where detector = "near" and
            reco_version >= "v1.3"
)
```

A filter *does not* actually have to access any external data.

This “random\_mix” filter mixes two file sets into one according to target ratios

# File Provenance

File provenance supported by MetaCat

- A file can have zero or more parent files and zero or more child files
- Which files were used to create which files

# File Provenance in MQL

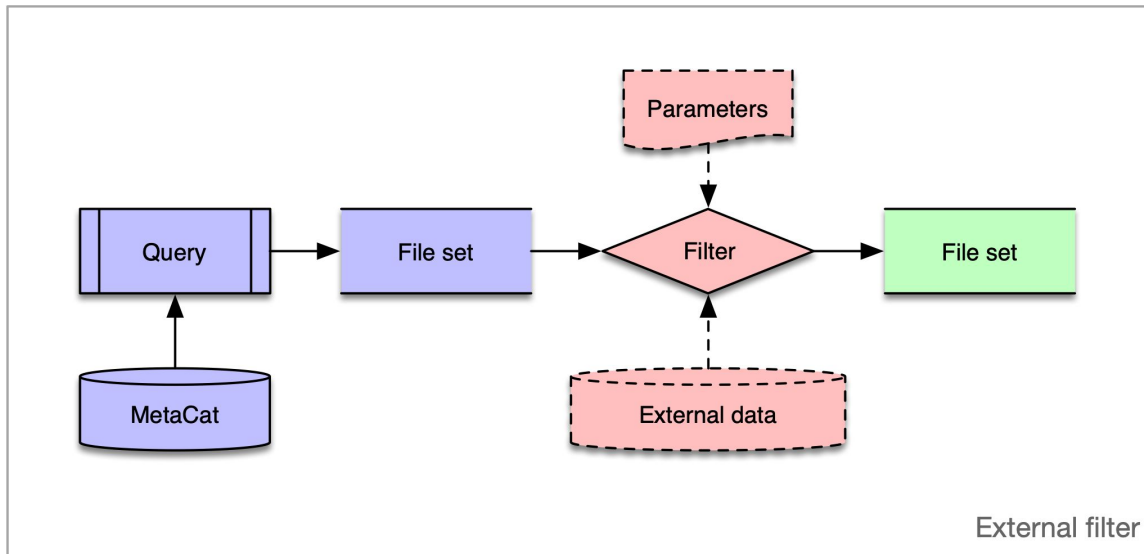
```
children ( file scope:file.data )           # children of a single file

parents (                                     # parents of all files
    files from dune:raw_2019                 # in the file set
    where reco_version = "v1.2"
)

files from dune:raw - parents(               # unprocessed files
    files from dune:processed
)

files from dune:raw -                       # files without any children
    parents(
        children(files from dune:raw)
    )
```

# External Filter



- A **filter** takes a file set (one or more) - results of a **query** (or queries)
- Produces **new file set**
  - *Optionally*: accesses the external data
  - removing or even adding files
  - modifying metadata
  - injecting new metadata

Filter is a python class provided by the collaboration and plugged into MetaCat server instance.

Not every user can add a filter - security

# Arrays and Dictionaries

```
bit_mask[2] = 1
```

Array element by index

```
config["version"] > "2.3"
```

Dictionary access by key

```
runs[any] = 1234
```

Any array element

```
1234 in runs
```

Array contains element

```
runs[all] < 1234
```

All elements

```
config[any] != "raw"
```

Any dictionary element

```
len(core.events) > 10
```

Array length



# Ranges and Enumerations

```
x in 1234:1332
```

Range of values

```
x in (1234,1235,2345)
```

Enumerated set of values

```
run_type in ("calibration","test")
```

- Can be strings too

```
file_type not in ("mc","test")
```

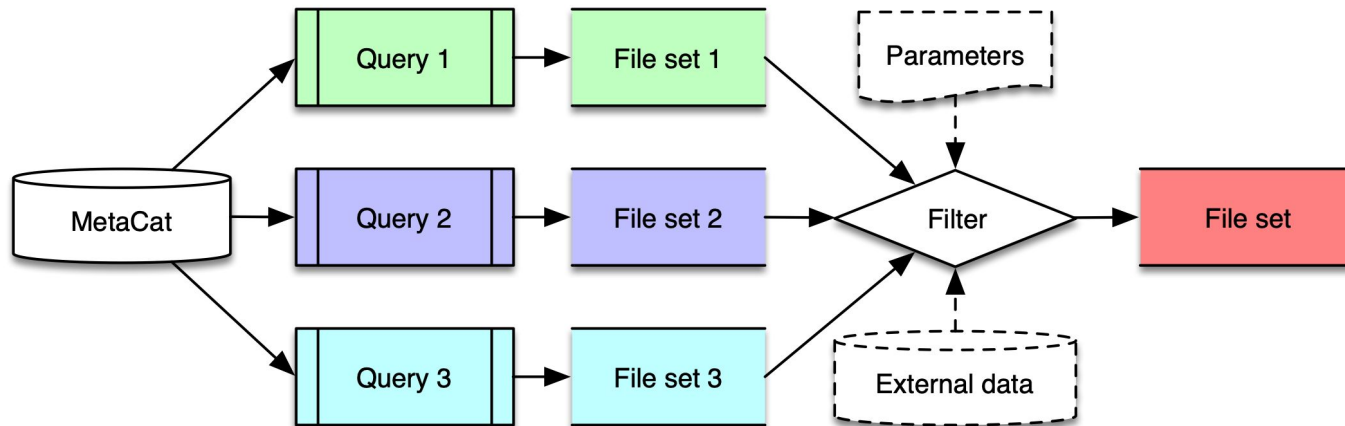
# Dataset Restrictions on File Metadata

Dataset can have restrictions on metadata for its files

- Parameter types, allowed values
  - Similar to categories
- Required parameters

When declaring a file or adding a file to a dataset, both dataset and parameter category restrictions apply

# External Filter with Multiple Inputs



External filter with multiple inputs

A filter can take **multiple file sets** as input  
and combine them into a **single output file set**