

Use of Rucio by DUNE

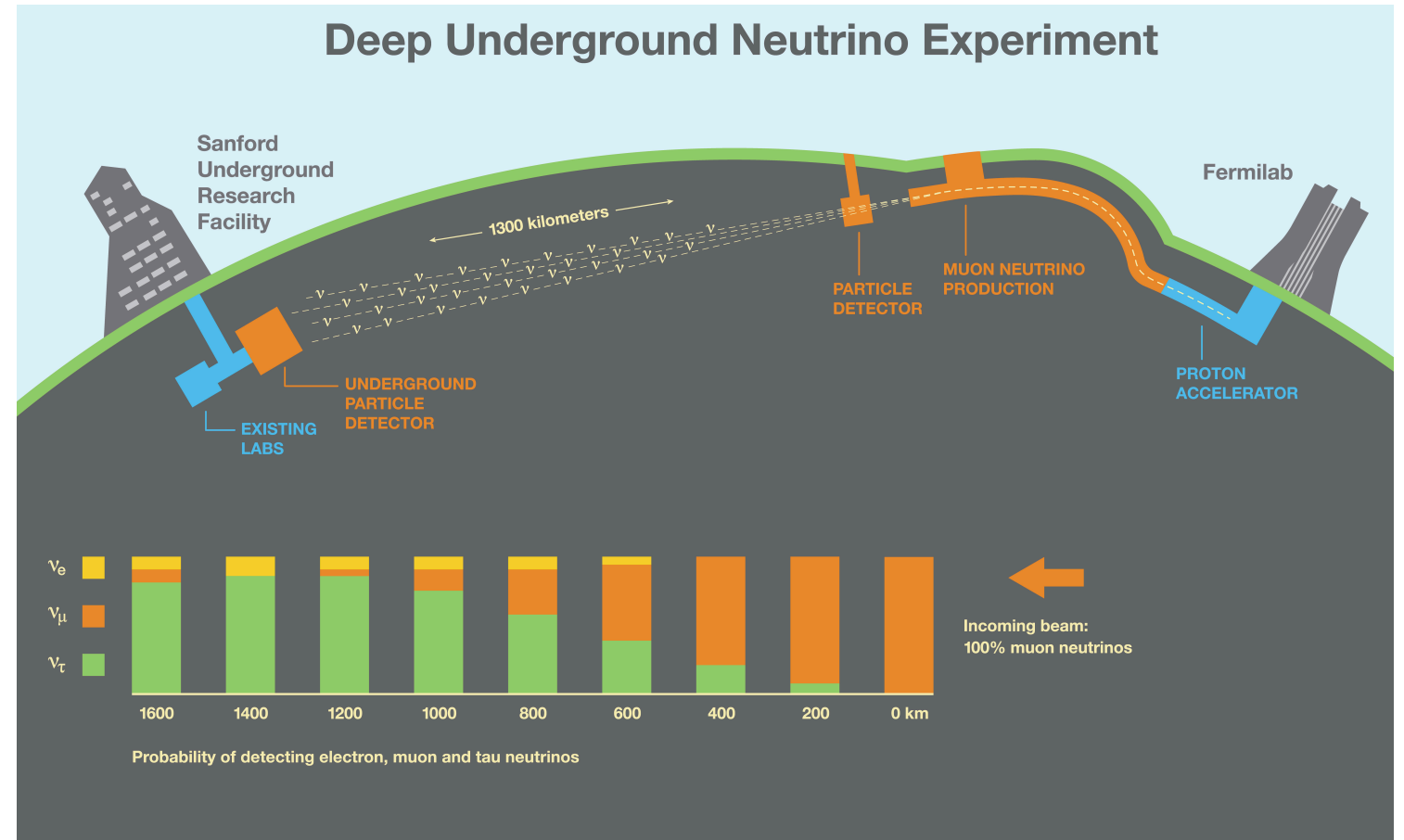
Wenlong Yuan
For DUNE Data Management Team

GDB Meeting, 14th Oct. 2020



Outline

- Description of DUNE Data Model
- DUNE Experience With Rucio
- Full Rucio-centric Data Management System



What is Rucio

SCIENTIFIC DATA MANAGEMENT



- Data Management software developed originally by ATLAS experiment
- Other large scientific collaborations (CMS, DUNE among others) have picked it up.
- Rule-based declarative system--tell it where you want the data and Rucio will get it there
- Uses FTS3 for file transport
 - Can also use Globus Online

DUNE Data Management Currently

- Two ProtoDUNE detectors at CERN, each 5% of full detector size
- ~10 PB of data accumulated thus far
 - Roughly half raw data, half reconstruction output products, small amount of MC.
Event size ~60MB
- 36 compute sites around the world
- 13 disk only sites, 4 disk+tape sites
- Data streamed via xrootd from the closest location.
- Very similar to other experiments that use the grid.

DUNE Data Terms

SUBDETECTOR	SD ₁	SD ₂	SD ₃	SD ₄	SD ₅	SD ₆
Trig Record	1	1	1	1	1	1
Trig Record	2	2	2	2	2	2
Trig Record	3	3	3	3	3	3
...

ProtoDUNE 6 subdetectors
60 MB compressed

Trigger record: Output of the DAQ for a single trigger

Data Unit: 1 Subdetector for 1 trigger record

Data Object: Collection of data units

SUBDETECTOR	SD ₁	SD ₂	...	SD ₁₄₉	SD ₁₅₀
Trig Record	1	1		1	1
Trig Record	2	2		2	2
Trig Record	3	3		3	3
...

Full DUNE: 150 subdetectors
6GB/readout (5 ms)

Trigger records may be split by subdetector across many different data objects.

The MANIFEST tells us what is where.

Unique DUNE data challenges

- Subdetectors from the same *trigger record* may end up in multiple files
- Time slices from the same *trigger record* may end up in multiple files
- In case of supernova burst readout we will have to split both by subdetector and by time slices, and get it out fast – 180TB/readout (100s)
- Exploring the HDF5 format to store the data
- Exploring non-file-based object stores in general.
- DUNE far detector output 30PB per year, plus more from near detector
- Working with signed JSON Web Tokens (JWT) for authentication.

Rucio Care and Feeding

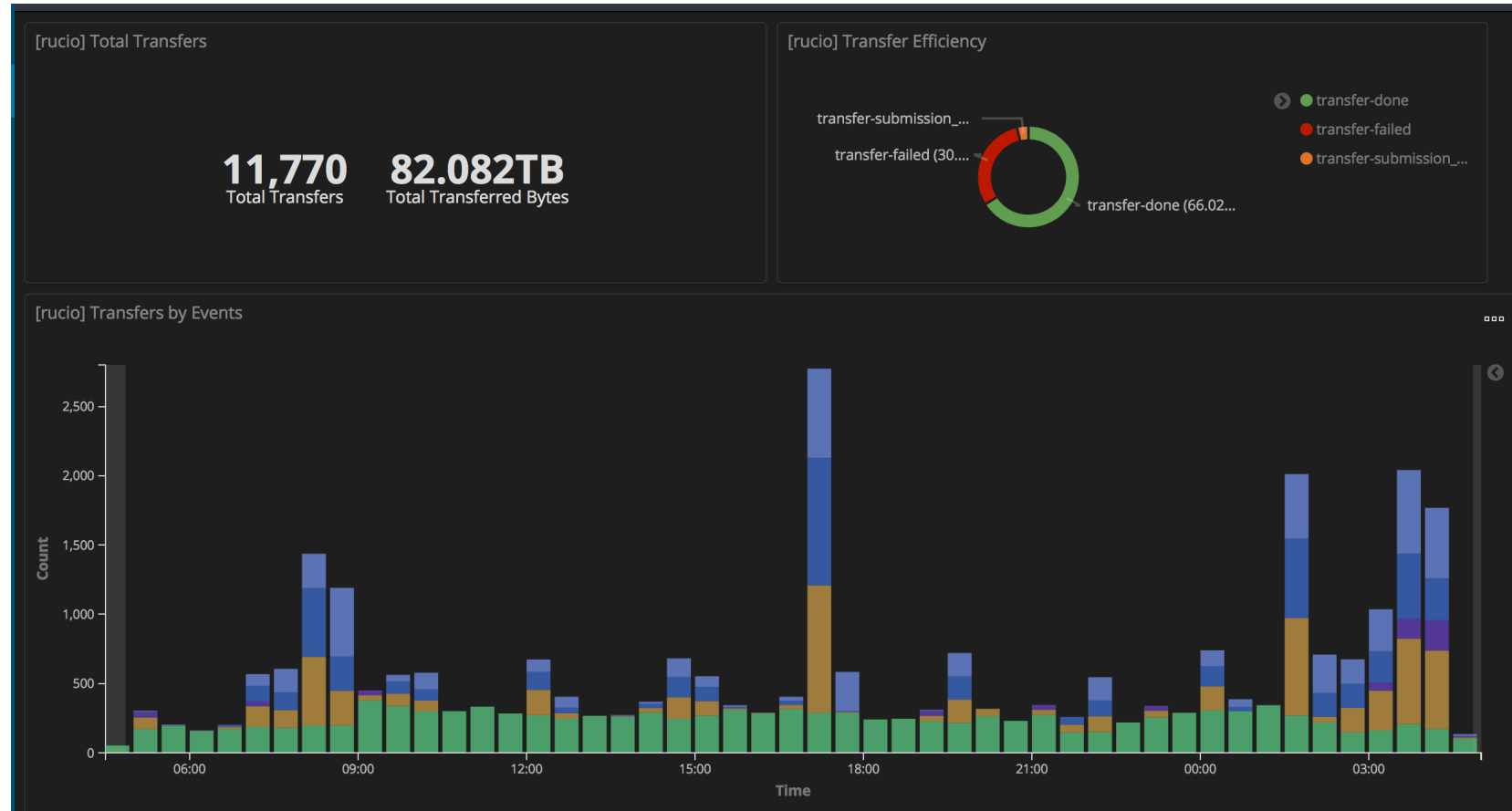
- DUNE has been running Rucio since Fall of 2018.
- Fermilab Scientific Data Storage Dept.
 - Rucio server deployed as a containerized application
 - changing to deployment on Fermilab's OKD cluster
 - Use shared Helm configuration to configure the various containers
 - Using Postgres DB on back end
 - Manages all schema and software upgrades.
- DUNE Data Management Team
 - Rucio clients to move data from point A to point B. (asking for help if things get stuck)
 - Creation and declaration of new Rucio replicas
 - Onboarding new remote Rucio storage elements
 - Interaction with remote sites for transfer

Current Rucio status

- Ingest of ProtoDUNE raw data still done with legacy system.
- Legacy system gets data from CERN EOS to CERN Castor and FNAL dCache/Enstore Tape
- Declared to Rucio once it gets to Fermilab
 - One dataset per run, large containers of related data sets
 - Different detector types are different scopes
Rucio is used to send it everywhere else.
- Rucio is also used to manage limited disk space on CERN EOS
- We use legacy system to tell us what it is, and where it is (But not for much longer)
- We have 17 commissioned Rucio Storage Elements (RSE)
- 13 PB under Rucio management, 1,398,000 DIDs, 3,112,187 replicas as of 10 Oct. 2020.

Monitoring: Transfers

- [Rucio Kibana Monitoring](#). Shows **queued**, **failed**, **submitted**, **done**.



Rucio Monitoring

[rucio] Mock _Total dids

1,398,000 DIDs
5.9PB Total bytes

[rucio] total replicas

3,112,187 Total replicas
13PB Total bytes

Total DIDs/replicas/sites monitoring

[rucio] DIDs per scope

- protodune-sp
- protodune-dp
- np04_pdspprod2_mc
- np04_reco_keepup
- np04_pdspprod2_reco

[rucio] DIDs per account

- root
- dunepro
- bjwhite
- ivm
- jperry

[rucio] DIDs per did type

- F
- D
- C

[rucio] DIDs per availability

- A
- L

Country/scope/container monitoring

[rucio] Replicas per site

RSE	Replicas	Total bytes
FNAL_DCACHE	1,377,955	5.9PB
CERN_PDUNE_CASTOR	1,034,222	4.4PB
CERN_PDUNE_EOS	260,421	816.1TB
RAL_ECHO	179,633	844.9TB
MANCHESTER	123,689	463.9TB
PRAGUE	39,558	272.7TB
DUNE_US_BNL_SDCC	36,599	71.8TB
IMPERIAL	33,894	243TB
LANCASTER	24,045	122.3TB

Export: [Raw](#) [Formatted](#)

[rucio] Replicas pie

- FNAL_DCACHE
- CERN_PDUNE_CAST...
- CERN_PDUNE_EOS
- RAL_ECHO
- MANCHESTER
- PRAGUE
- DUNE_US_BNL_SDCC
- IMPERIAL
- LANCASTER

Rucio Usage by Country

- US
- CH
- GB
- CZ
- FR

Rucio Usage by Scopes

- protodune-sp
- protodune-dp
- np04_reco_keepup
- np04_pdspprod2_mc
- np04_pdspprod2_reco

Rucio Usage by Containers

- np04_raw_runs_4000...
- np04_raw_all_run_nu...
- np04_full-reconstruct...
- np04_raw_runs_1-3999
- np02_commissioning...

Rucio Country by Containers

- GB
- US
- CH
- CZ
- np04_raw_all_run_nu...
- np04_raw_runs_1-3999
- np04_raw_runs_4000...
- PDSPProd2_MC_1Ge...
- np04_full-reconstruct...
- PDSPProd2_MC_6Ge...
- np02_commissioning...

Rucio Containers by Scope

- np04_raw_runs_4000...
- np04_raw_all_run_nu...
- np04_full-reconstruct...
- np04_raw_runs_1-3999
- np02_commissioning...
- protodune-sp
- np04_reco_keepup
- protodune-dp

Rucio Scopes by Country

- protodune-sp
- np04_reco_keepup
- np04_pdspprod2_mc
- protodune-dp
- GB
- CH
- US
- CZ



Use Case: Vacating a Storage Element

- Given notice we had to get ~500TB off of a storage element
- Had 1 month notice initially
- All of data in question was data that existed in at least one or two other places
- Used rucio list-dataset-replicas to
 - Identify which datasets were there now
 - Identify at least one other site where that dataset is *not*
- Then make one rule per dataset (886 datasets in all)
- And sit back and wait
- Transfers were kicked off on 23 Jan, finished on 1 Feb.
- And then we learned we really had 6 months to get out

Rucio features we know we need

- Quality of Service
 - Better way to know when a file is online or on tape.
 - Detection of condition when Fermilab dCache has the file in online storage and prefer that as a source.
- Deterministic vs. non-deterministic
 - “Deterministic” is for disk sites—files stored in a hashed path
 - “Non-deterministic” used on tape sites—human-readable path constructed from metadata fields
 - Pending a new feature to serve the path to Rucio.
- Lightweight client for REST API
 - Stock client has lots of dependencies

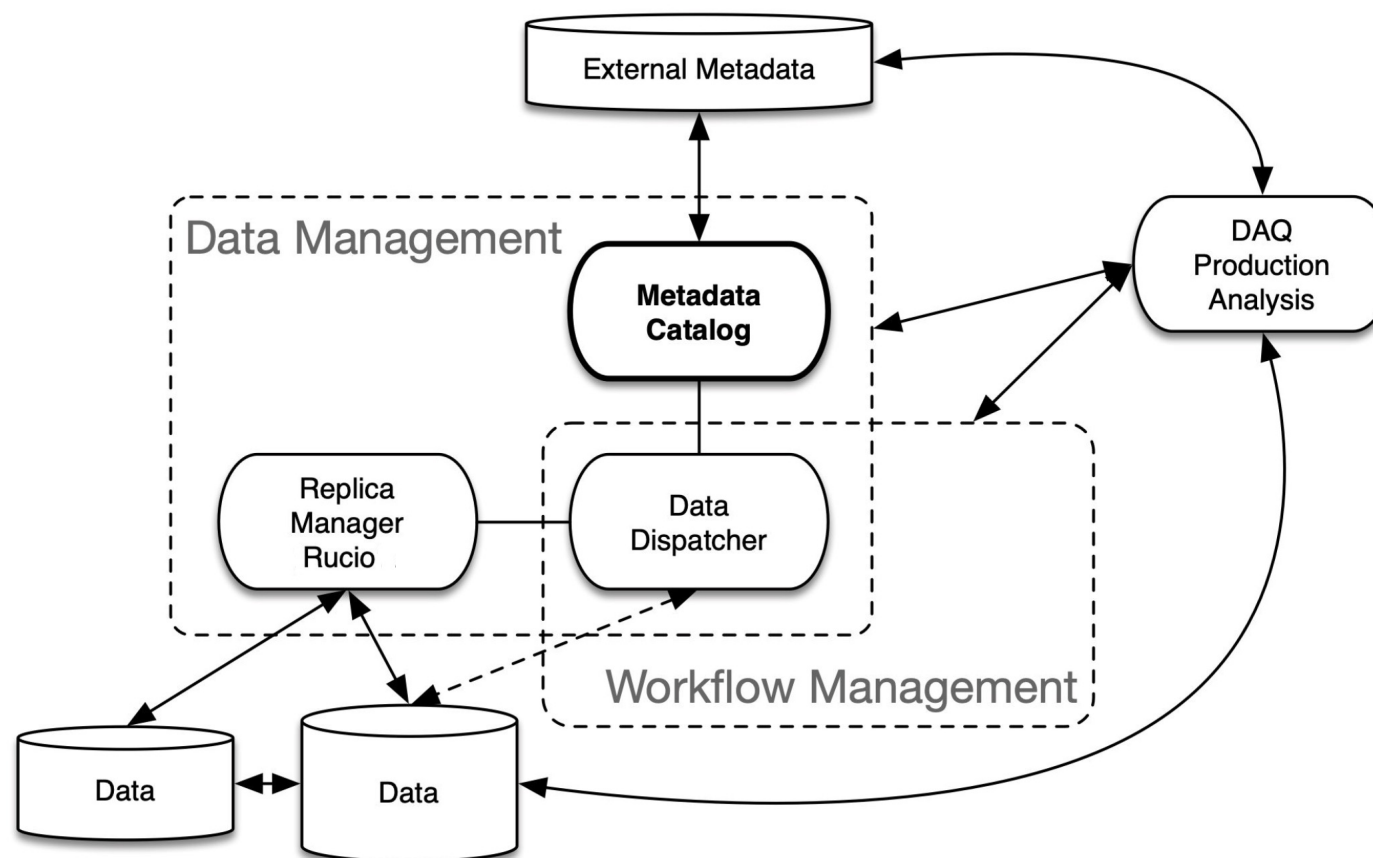
Changes needed to use Rucio in reco/analysis workflow

- We are using Rucio for data movement
 - Need to change our data delivery system so we use the Rucio file location info—and use Rucio to deliver the file location.
- Need to replace 3 main functions of monolithic legacy system
 - **Replica manager** (Where is the file) -> **Rucio**
 - **File Provenance** (Metadata) -> Metadata Catalog (**MetaCat**)
 - **Data Delivery / project tracking** -> **Data Dispatcher**
- Three projects needed to get there:
 - **New Data Ingest** service
 - **New Metadata service** (MetaCat)
 - **New Data Delivery service** (Data Dispatcher)

=



DUNE Data Management Architecture

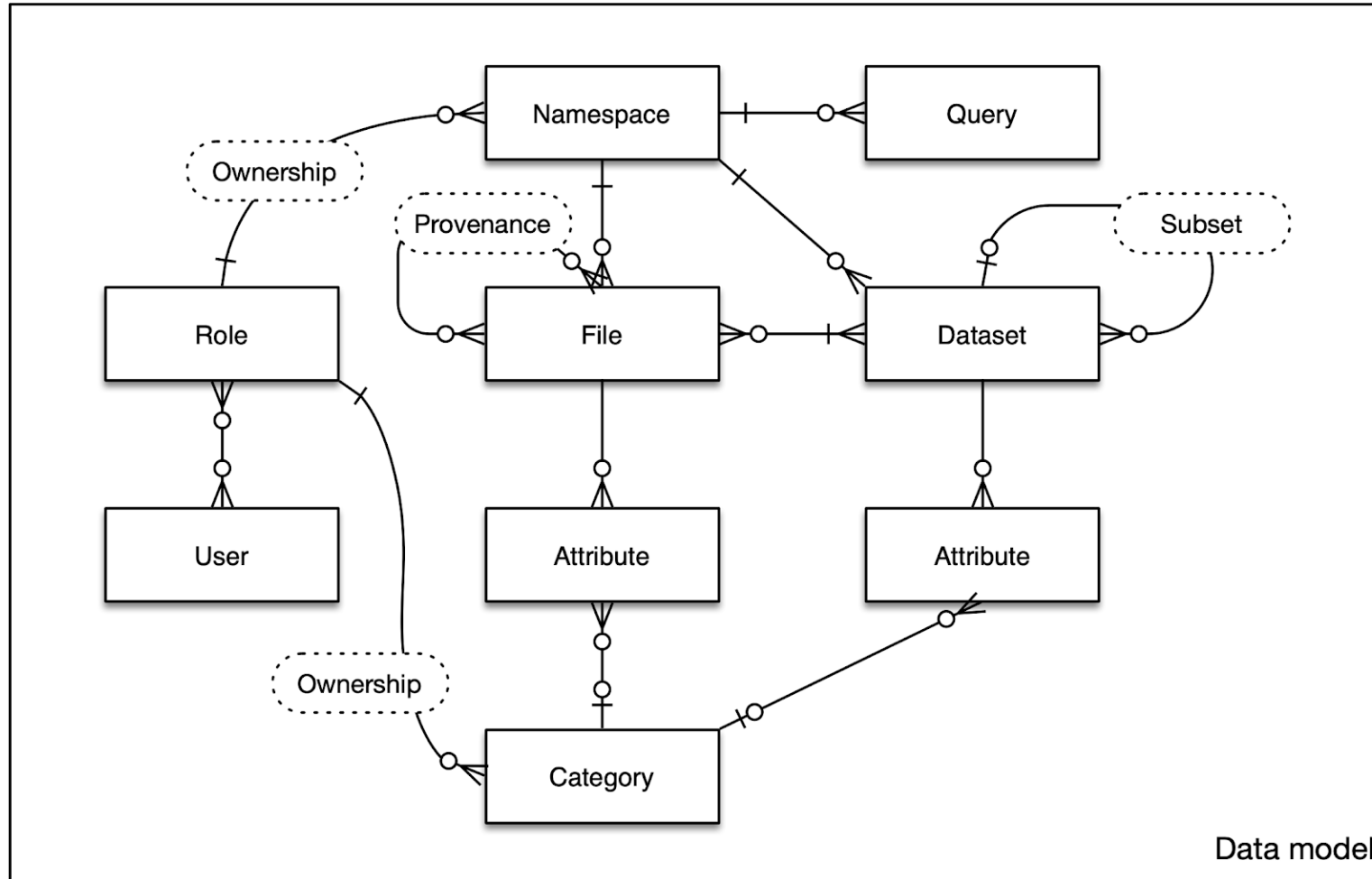


Data Ingest

- Still in planning phase
- Replace legacy system with:
 - Process that scans the newly created files on the DAQ buffer disks
 - @ ProtoDUNE, Far Detector(s), Near Detector
- Declare file location to **Rucio**, metadata to **Metacat**
- Use FTS3 to do 3rd party transfer to FNAL and from FNAL to elsewhere, under Rucio control

Metadata service: MetaCat

- Metadata sets very similar to Rucio model
- Requirements are complete
- Reference implementation in testing



Data Delivery service: Data Dispatcher

- Users are used to :
 - Defining a dynamic data set
 - Running a project across the whole data set:
 - Each job says “give me the next file”
 - Each job notifies the project manager when it’s processed successfully.
 - Recovery jobs can be generated.
- Rucio has fixed data sets.
 - Some important ones we will declare “immutable”
 - Others we will declare “monotonic” to which you can add but not delete
- Metadata server API will allow queries of the metadata plus user callouts to the conditions database.
- “Data Dispatcher” in this case is very low level—much simpler than Dirac or WMAgent for instance. DUNE will eventually choose a higher level workflow manager which interfaces to this.

Conclusions

- Rucio is working well for DUNE to date
- No known issues keeping us from scaling up to full DUNE
- Goal to have Rucio-centric data management in place by next run of ProtoDUNE (ProtoDUNE II) early 2022
- Thanks to Rucio core developers, site admins all over the world, and DUNE Data Management team