

April 27, 2026

Fermi Data Platform

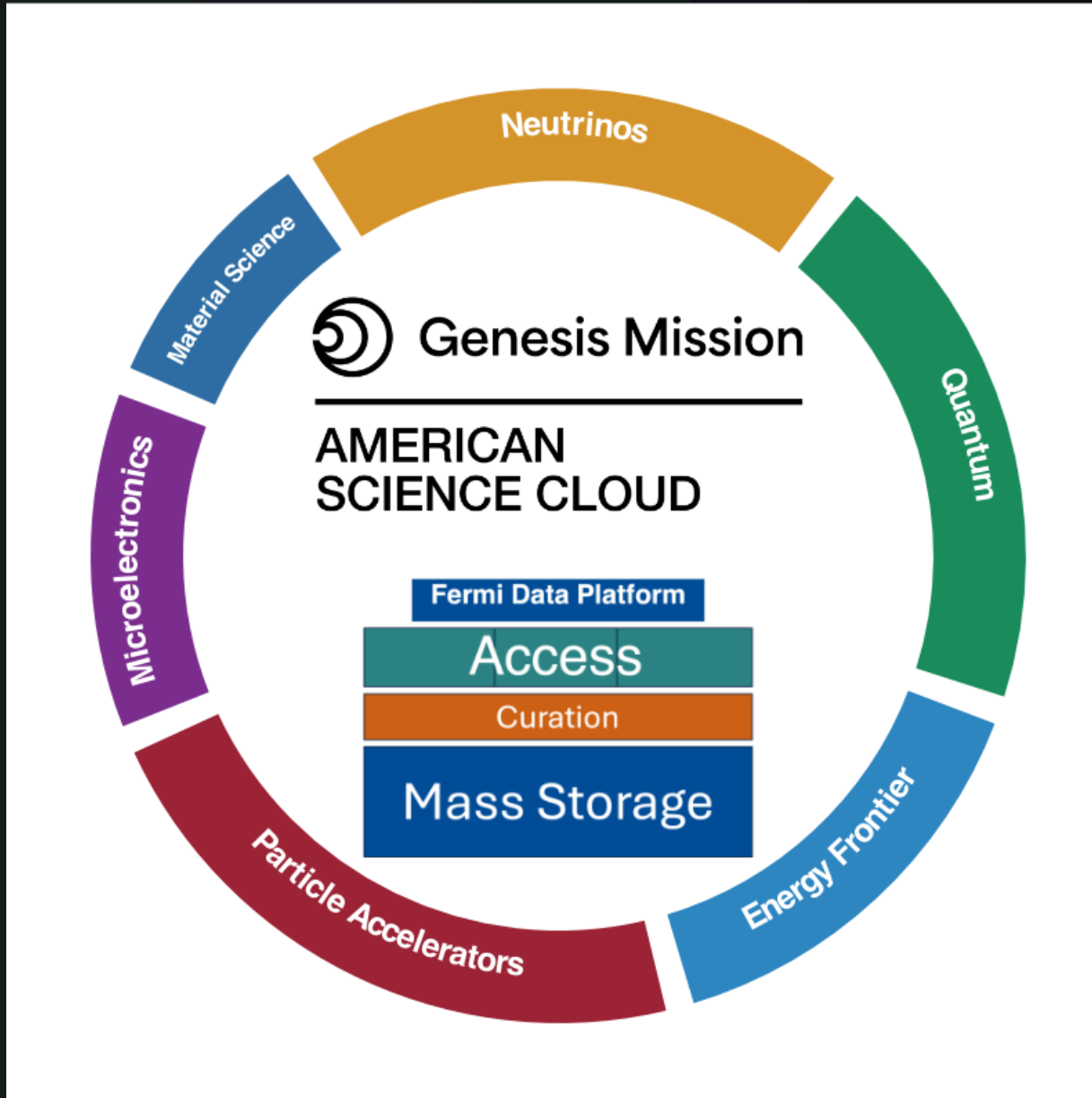
Scarlet Norberg on Behalf of the FDP team



U.S. DEPARTMENT
of **ENERGY**

Fermi National Accelerator Laboratory is managed by
FermiForward for the U.S. Department of Energy Office of Science

Genesis Mission



Fermi Data Platform

- Core mission is to host and serve curated data sets for the Genesis mission
- Initially we will host HEP activities (Treasure, SUF), accelerator and nuclear physics data activities, (including LQCD, Cosmiq, Axess)
- We are striving to move towards supporting a broader constituency within the DOE complex
- FDP:
 - Storage Architecture used is:
 - dCache running on distributed storage resources backed by RAID6
 - Accessible through Fermilab's Multi-Terabit ESnet connection
 - Currently deployed transfer and streaming [protocols](#)
 - WebDav
 - XRootD
 - Globus

Fermi Data Platform Access

- First get a Fermilab account
- Right now, you need to be related to the overall Genesis mission
- Request access to the AmSC Fermilab VO [here](#)
- Request a Role within that VO [here](#)
- Request to: fdp_data_management@fnal.gov
- Some code has been used to stream FDP data on NERSC
- Data has been transferred in and out of FDP

- To access the data, you need to have a Fermilab account or be added to the globus endpoint
 - From globus online:
 - <https://www.globus.org/>
 - Log in with Fermilab credentials, We can add external people
 - Search for the “Fermi Data Platform” collection
 - The data is in the /amsc/public directory
 - From the command line:
 - `globus transfer [source-endpoint-ID]:[/path/to/source] [destination-ID]:[/path/to/destination]`
 - The Fermi Data Platform UUID collection (used to access and transfer files) is b35955d3-14d1-4aab-a1c9-189989f7d8d0 and the path is /amsc/public//path/to/FileName



Globus Endpoint

- Each activity will have a globus collection
 - The treasure collection is dd31b09b-a21e-4c12-b054-b4da220b5297

Collection

Path

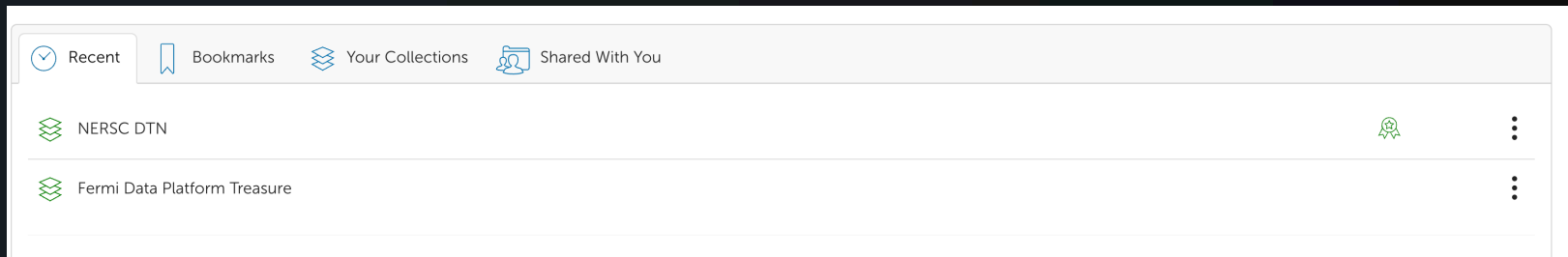
select all up one folder refresh list filter


NAME ▾

- aoj
- aoj_tokenized_Feb2026
- collide-1m
- dwd
- test

Globus Endpoint Finding UUID

- When you search for Fermi a few end points show up if you click the three dots on the right information on the end point will appear, including the collection UUID



Authentication Timeout	11 days
Multi-factor authentication required	No
UUID	dd31b09b-a21e-4c12-b054-b4da220b5297 

Webdav

- From browser:
 - <https://amsc.fnal.gov:2880/amsc/public/>
- From command line:
 - `curl -L https://amsc.fnal.gov:2880/\[source_path\] -o [destination_path]`
 - The `source_path` is the path to the data on amsc
 - Using `gfal`, which requires the `gfal2-*` packages:
 - `gfal-ls https://amsc.fnal.gov:2880/amsc/public/`
 - `gfal-copy https://amsc.fnal.gov:2880/amsc/\[source_path\] [destination_path]`
 - Token is needed for none public data

- Requires the *xrootd-client* package
- From the command line:
 - Token will be needed for none public data
 - Copy a file:
 - `xrdcp root://amsc.fnal.gov/[source_path] [destination_path]`
 - List a directory:
 - `xrdfs root://amsc.fnal.gov ls -l /amsc`
 - Using *gfal*, which requires the *gfal2-** packages:
 - `gfal-ls root://amsc.fnal.gov/amsc/public`
 - `gfal-copy root://amsc.fnal.gov/amsc/[source_path] [destination_path]`
- `xrdcp -f root://amsc.fnal.gov/amsc/public/treasure/aoj/data/RunH_batch38.h5 /dev/null`
 - `[2.515GB/2.515GB][100%][=====][107.3MB/s]`



Tokens

- FDP sorts the public and private data on the file system dCache in this way:
 - /amsc/public/<activity>
 - /amsc/<activity>
- For reading the public data
 - This can be done with xrootd and webdav
 - For Globus by being added to or having a Fermilab account
- For reading the private data
 - You need a Fermilab account, to be add to the internal AmSC Fermilab VO and be added to a role that is associated with an activity
- For writing to the private or public area
 - Currently you need a Fermilab account, you need to be added to the internal AmSC Fermilab VO and be added to a role that is associated with an activity
- To be added to the VO you can go here with [service now](#), the role you can get from [here](#)




Meta Data Catalog

- Fermi Data Platform is using Metacat as its [metadata catalog](#)
- It is very flexible
- Namespaces
 - Currently Treasure is the name space used
- Datasets
 - The datasets are used for grouping but not for distinguishing files apart
 - A few of the current datasets are: aoj, collide
- Files
 - All files under a namespace need to have unique file names
 - A workaround for the same filename in different directories in the same namespace is to use the directory structure in the filename field to make it unique
- To access the Metacat [website](#) of FDP, you need to have a Fermilab account and AmSC VO association at Fermilab



Meta Data Catalog

metacat.fnal.gov



AmSC MetaCat

[users](#) [roles](#) [namespaces](#) [datasets](#) [find file](#) [categories](#) [query](#) [named queries](#) [filters](#) [docs](#) logged in as Alison Peisker [apeisker](#) [log out](#)

File Information

File ID [6WeJWk4BSMSf7evu](#)

Namespace [treasure](#)

Name [RunG_batch0.h5](#)

DID [treasure:RunG_batch0.h5](#)

Size 2379411451 (2269.184 MB)

Created 2026-02-20 10:22:19.693243-06:00 by [apeisker](#)

Updated 2026-04-08 10:59:56.478321-05:00 by [mengel](#)

Checksums sha256: c2058846a224b1a915d65e89cec6eb9586db780eab6e83b1822ff5c22f63e143

Parents

Children

Datasets

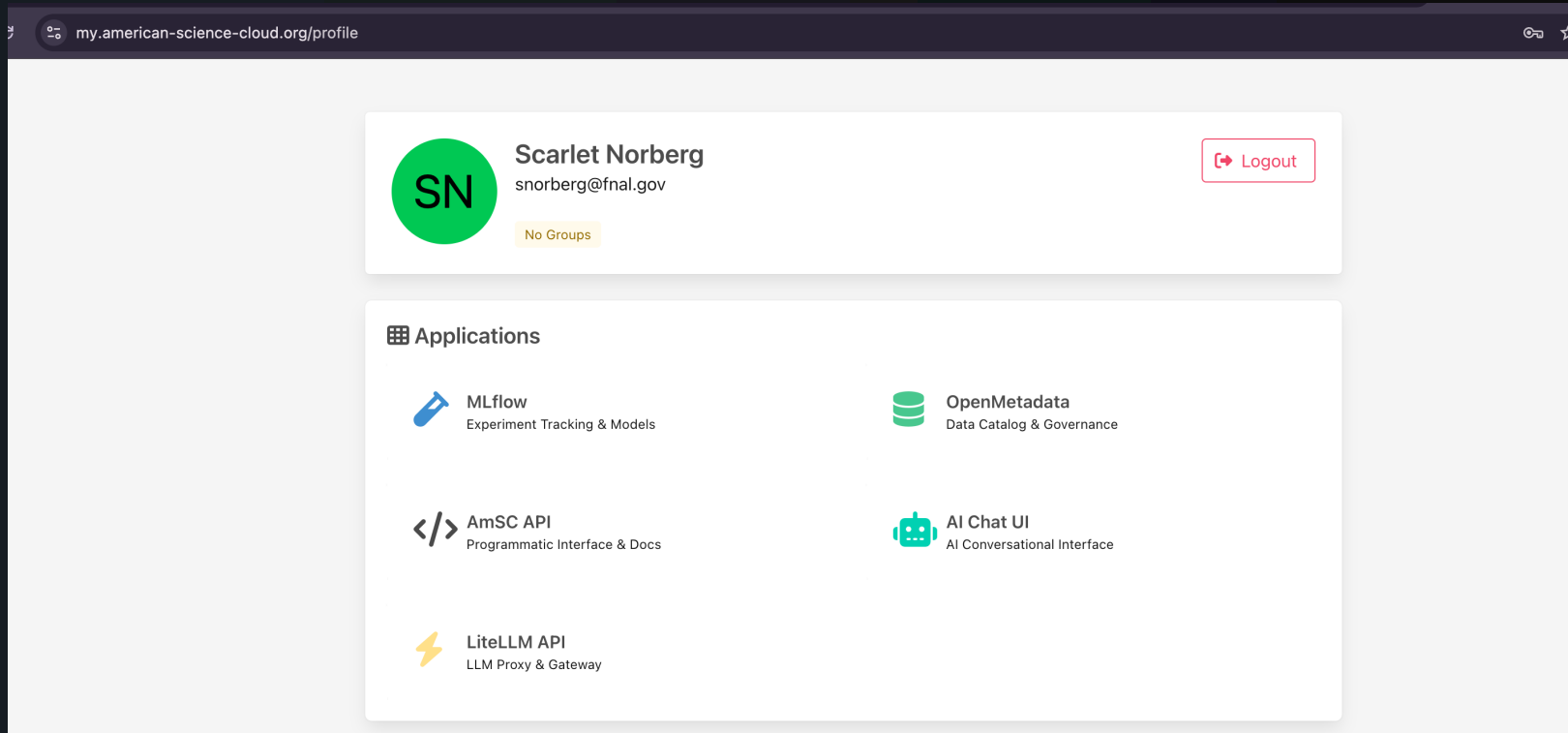
[treasure:aoj](#)

Metadata

Parameter	Value
AmSC.common.description	This is a file from the aoj dataset
AmSC.common.fnqn	fnal-amsc-storage.fnal-amsc-data-catalog.aoj."RunG_batch0.h5"
AmSC.common.location	https://amsc.fnal.gov:2880/amsc/public/treasure/aoj/data/RunG_batch0.h5
AmSC.common.type	artifact
fn.locations	['https://amsc.fnal.gov:2880/amsc/public/treasure/aoj/data/RunG_batch0.h5', 'root://amsc.fnal.gov/amsc/public/treasure/aoj/data/RunG_batch0.h5']
fn.path	/amsc/public/treasure/aoj/data/RunG_batch0.h5

Central American Science cloud Account

- Go to my.american-science-cloud.org
- You need a globus account, click your globus affiliation login



The screenshot shows a web browser window with the address bar displaying `my.american-science-cloud.org/profile`. The main content area features a user profile card for Scarlet Norberg, with a green circular avatar containing the letters 'SN', the email address `snorberg@fnal.gov`, and a 'Logout' button. Below the profile card is an 'Applications' section with a grid of five items: MLflow (Experiment Tracking & Models), OpenMetadata (Data Catalog & Governance), AmSC API (Programmatic Interface & Docs), AI Chat UI (AI Conversational Interface), and LiteLLM API (LLM Proxy & Gateway).



Data Pushed to Central American Science Cloud

The screenshot shows a web browser window at `openmetadata.american-science-cloud.org`. The page displays the details for the "AOJ (Aspen Open Jets Dataset for TREASURE)" dataset. The interface includes a search bar, navigation menu, and various tabs for exploring the dataset's metadata.

Dataset Details:

- Name:** AOJ (Aspen Open Jets Dataset for TREASURE)
- Identifier:** aoj
- Followers:** 0
- Version:** 0.1
- View in CustomStorage:** [Link]

Metadata Tabs: Domains, Owners, Tier, Certification

Children: 80

Description: Four vectors of jets from CMS used by the TREASURE Genesis team in the Office of Science. This dataset contains approximately 180 M boosted jets, derived from open data collected by the CMS experiment at the Large Hadron Collider (LHC) in 2016 — specifically the JetHT datastream — and presented in a format suitable for Machine Learning (ML)...

Name	Description
RunG_batch0.h5	This is a file from the aoj dataset
RunG_batch10.h5	This is a file from the aoj dataset
RunG_batch13.h5	This is a file from the aoj dataset
RunG_batch14.h5	This is a file from the aoj dataset
RunG_batch15.h5	This is a file from the aoj dataset
RunG_batch18.h5	This is a file from the aoj dataset
RunG_batch19.h5	This is a file from the aoj dataset
RunG_batch20.h5	This is a file from the aoj dataset

Custom Properties:

- entityType:** scientificWork
- mimeType:** No data



Streaming

- Streaming data is the continuous flow of real-time data from various sources
- FDP provides proof-of-concept code to stream data from FDP to NERSC
- In addition, providing code executed at NERSC using awkward array streaming parquet files from FDP
- Prototype code to access h5 files using h5py streaming exists, but has not yet been tested by Treasure
- This information is being added to the git area for fdpdocs linked [here](#),
 - We are working on finishing editing and cleaning up documenting the code prototypes and will also create github pages for better readability

FDP Areas Currently Being Worked on

- We are working on making syncing between the local Metacat metadata catalog and the central AmSC metadata Openmetadata catalog automatic
- We are working on finalizing the directory structures on the dCache file system as well as the mapping to activity tokens
- Websites for information is continually being edited and updated
- Streaming has been done successful with parquet and awkward array code
- Currently we require a Fermilab account to access the FDP, but when American Science Cloud provides a central AmSC token, we will work to accept that token to read and write data



Conclusion

- We are implementing and deploying FDP and developing pieces of workflows accessing data on FDP
- We have a lot of experience supporting teams with their storage and data access needs
- We are here to help you to run your workflows in the easiest way possible



Fermilab

Fermi *FORWARD*



U.S. DEPARTMENT
of ENERGY



Streaming

H5py

```
>>> import h5py
>>> import fsspec
>>> with fsspec.open('root://amsc.fnal.gov/amsc/cms/aoj/data/RunG_batch39.h5') as f:
...     with h5py.File(f) as h:
...         for name in h:
...             print(name)
```



Streaming

Awkard array with Parquet

```
import awkward
import os, sys

def iter_files(filename, columns=None):
    metadata = awkward.metadata_from_parquet(filename)
    row_groups = metadata['num_row_groups']
    for rg in range(row_groups):
        data = awkward.from_parquet(filename, columns=columns, row_groups=set((rg,)))
        for d in data:
            yield d

def test(fname):
    print("\n Attempting to read FPD file %s" % fname)
    data = iter_files(fname)
    print("\n First element: ")
    print(next(data))
    print("\n Second element: ")
    print(next(data))
    print("\n Success!")

    return

if __name__ == "__main__":

    test(sys.argv[1])
```