DUNE Rucio development and monitoring

Wenlong Yuan
On behalf of DUNE Collaboration

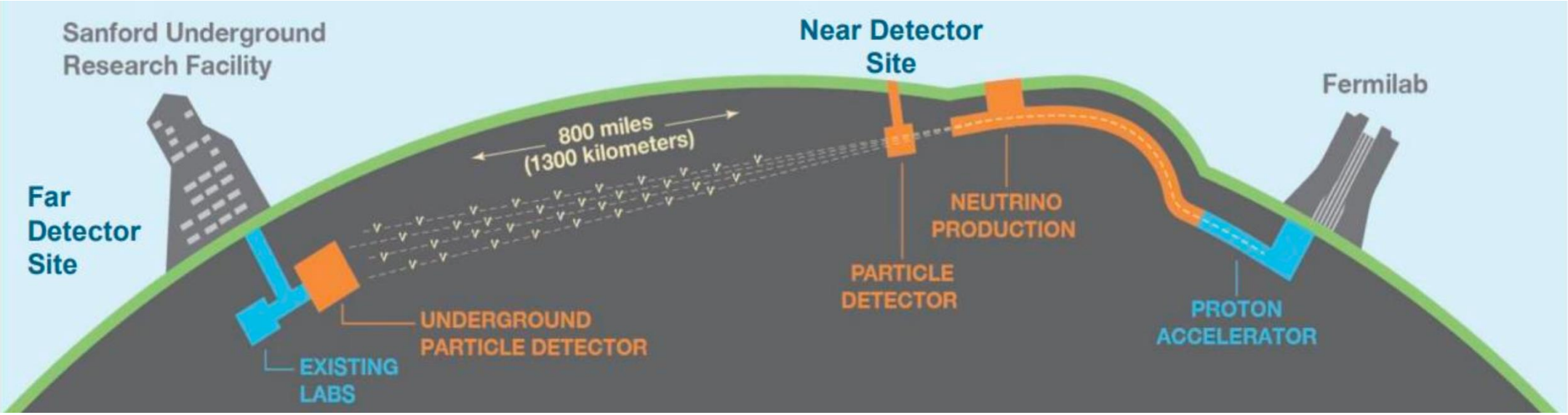
CHEP 2024

23rd October 2024

THE UNIVERSITY *of* EDINBURGH

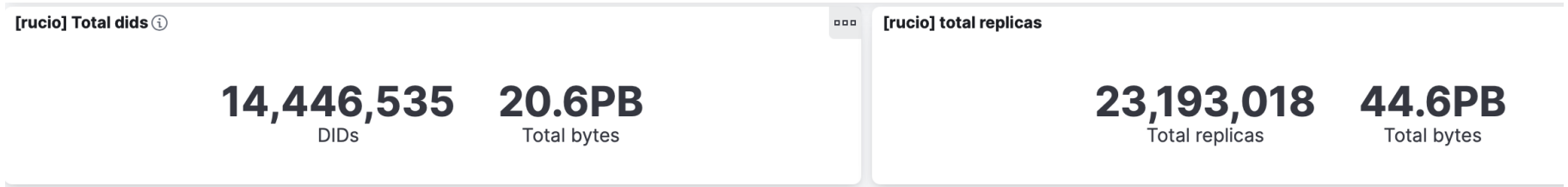
DEEP UNDERGROUND NEUTRINO EXPERIMENT

DUNE



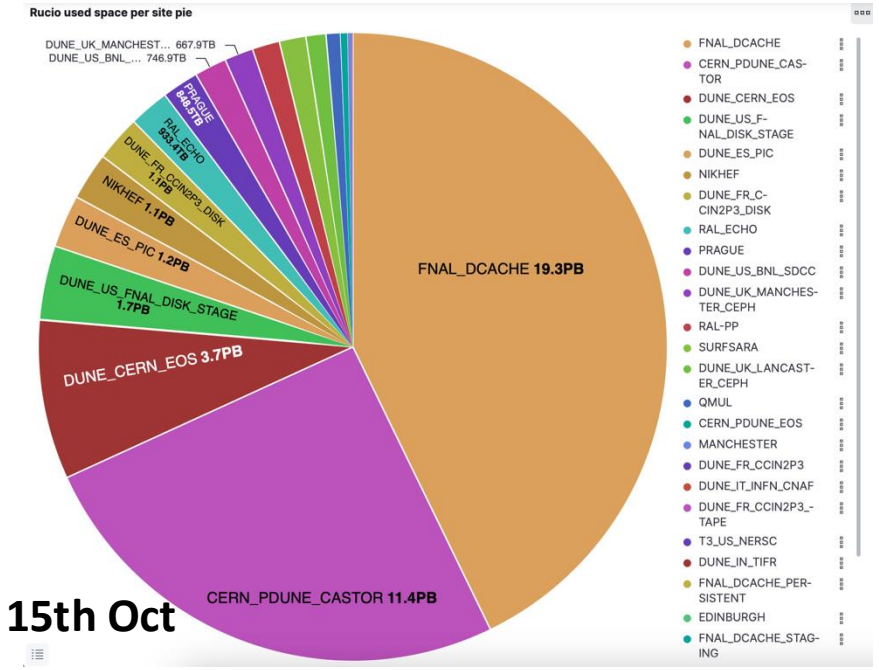
- The Deep Underground Neutrino Experiment (DUNE) is a long-baseline experiment which aims to study neutrino oscillations and Astroparticle physics amongst other things
- DUNE will consist of the near and the far detectors placed on the path of the most intense accelerator neutrino beam in the world
- DUNE is scheduled to start running in 2029, expected to record 30 PB/year of raw data

Rucio



(DID refers to an individual raw file; not all of DUNE's raw files have been declared in Rucio yet)

- The next-generation Data Replica service originally designed by the ATLAS collaboration
- DUNE has adopted and deployed Rucio, to handle large-scale data, like other experiments
- Rucio is an essential component of DUNE Distributed Data Management system, along with MetaCat (Metadata Catalog), to replace legacy SAM system



Outline

- Rucio/DUNE Rucio development – James Perry
  - Objectstore Support
  - Lightweight Rucio Client
  - Policy Packages
  - DUNE-specific continuous integration tests
  - Custom replica sorter algorithm
- DUNE Rucio monitoring – Brandon White, Wenlong Yuan
  - Rucio internal monitoring
  - Transfer monitoring
  - Rucio database dumping

Rucio Development - Objectstore Support

- Full support for S3, Swift and Google Cloud Storage now in Rucio core code, legacy code removed
  - Clients no longer need to have objectstore credentials
  - Used for all upload, download and deletion operations on objectstores
  - Third party transfers can either use objectstore protocol directly in FTS, or use Dynafed as a frontend
- Scalability testing successfully performed
- Minimal objectstore (based on min.io) added to Rucio development Docker image for easy development and testing

Rucio Development - Lightweight Rucio Client

- Make Rucio client package easier to install
- Could use containers, but inefficient to spin up container to run (e.g.) single Rucio upload command at the end of a workflow,
- Removed several dependencies and made others optional
  - Fewer conflicts with other packages
- Many config values can now be specified in environment
  - Don’t always need config file
- This work is beneficial to all communities
- Possible further work: allow client to choose suitable data access library instead of server dictating this

Rucio Development - Policy Packages

- Allows experiments to “plug in” custom code to Rucio using a standard interface
  - Permissions, schema, logical to physical filename conversion, etc.
  - Includes support for multi-VO installations with separate policy package per VO (Virtual Organization)
- Created a policy package for DUNE
  - Includes physical filename generation - queries MetaCat (DUNE’s metadata catalogue) to find path components
  - Blocks upload of files not registered in MetaCat via custom permission check
- Other experiments are moving to use their specific policy package

Rucio Development - DUNE-specific continuous integration tests

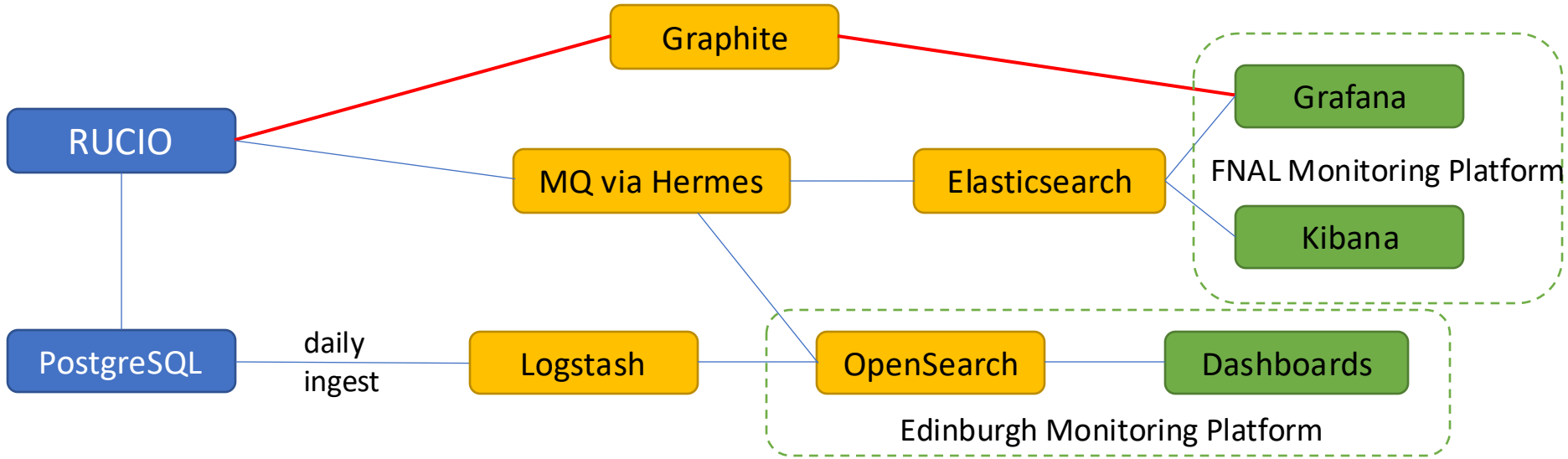
- These would run via GitHub Actions for each git commit, allowing us to more easily catch any Rucio regressions affecting DUNE
- Some experiments already have these (ATLAS, Belle II)
- DUNE's environment is more challenging:
  - Need local MetaCat instance
  - Need multiple Docker containers
  - Need DUNE policy package deployed
- Test suite has been written, waiting for changes to Rucio test framework to allow this to be done in a more generic way

Rucio Development - Custom replica sorter algorithm

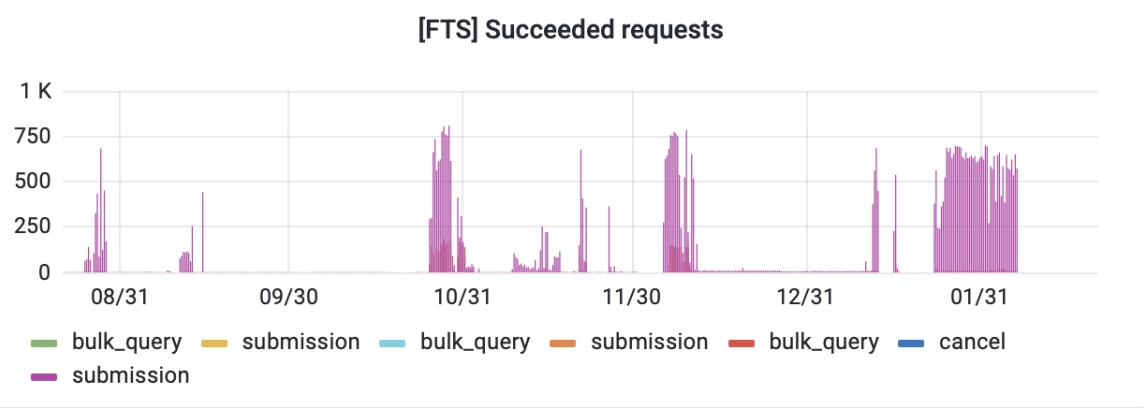
- Rucio allowed replicas to be sorted by preference according to (for example) geographical location
- However DUNE has unique network topology
  - DUNE distributed computing model let jobs to run on both storage sites and storageless sites
  - Requires custom sorting method
- Added new custom sorting algorithm
  - Fetches network information from simple CSV file
- While implementing this, found some bugs in the existing replica sorter code
  - Currently fixing these and adding unit tests

Rucio Monitoring - Internal Monitoring

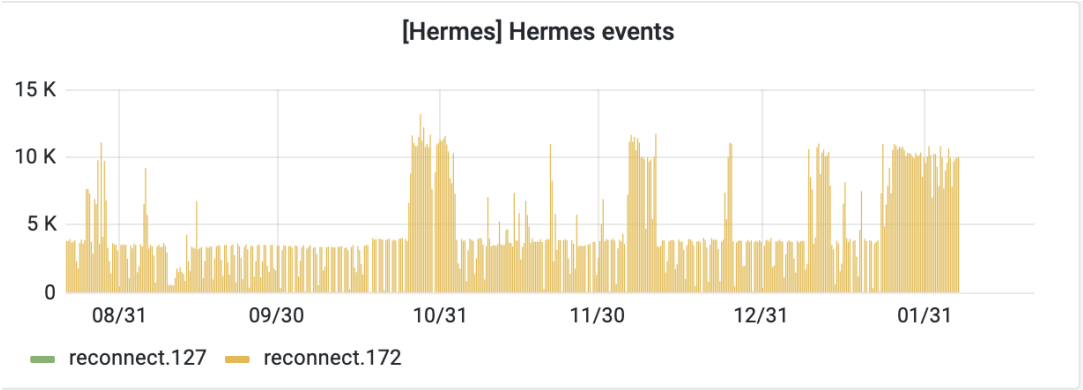
- Using Graphite/Grafana monitoring toolset to monitor the internals of Rucio servers and daemons, e.g., submission rate of the conveyor daemon (manage file transfers), state of conveyor queues, reaper (deletion daemon) rate, server response times, etc.
- DUNE has deployed this new internal monitoring based on Fermilab (FNAL) monitoring Platform



Rucio Monitoring - Internal Monitoring



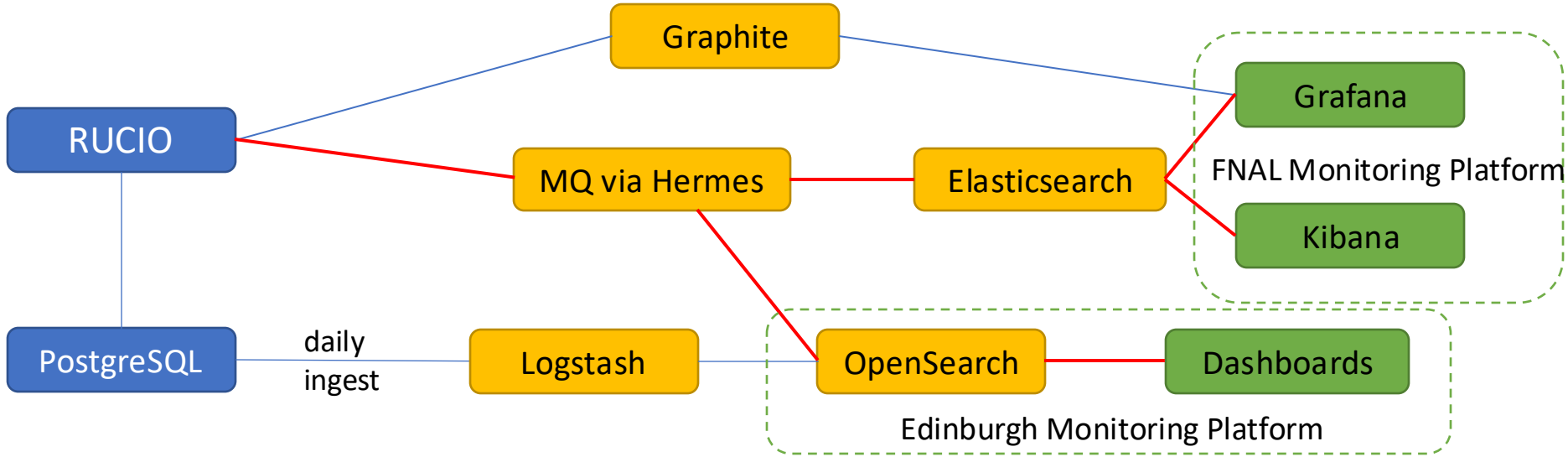
Rucio FTS requests



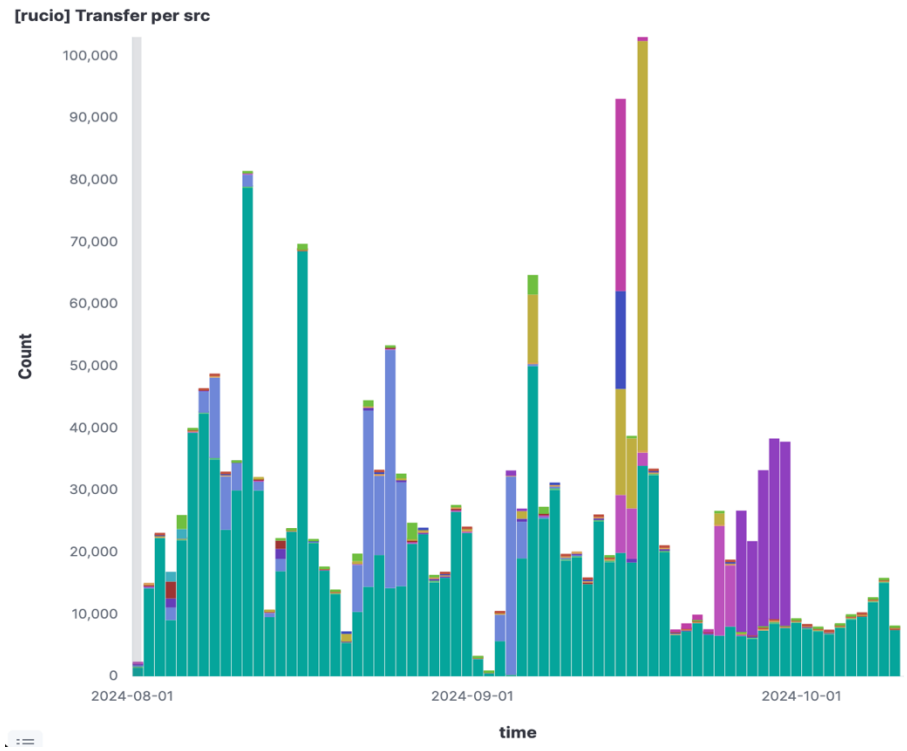
Rucio Hermes daemon events

Rucio Monitoring - Transfer/deletion monitoring

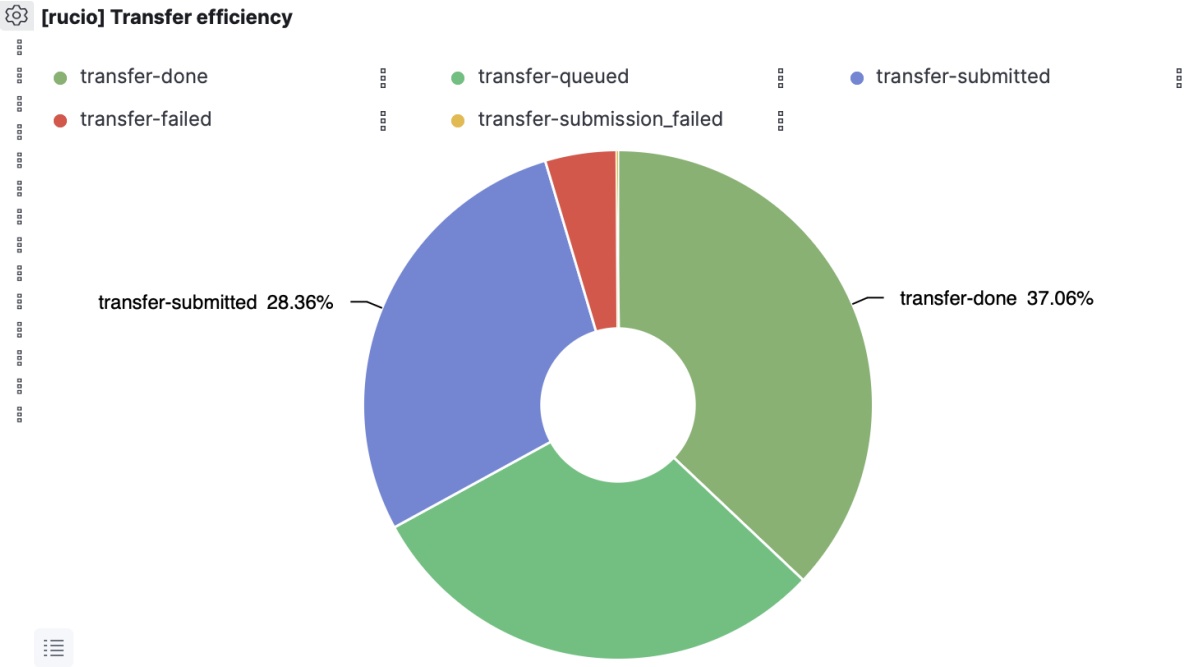
- When a Rucio transfer/deletion been submitted, the status messages (submitted, queued, waiting, done, failed) are sent to a Message Queue (MQ) via Hermes daemon
- DUNE reads these status messages from MQ, aggregates them and then writes the aggregated data into OpenSearch at Edinburgh and ElasticSearch at FNAL, where they can be visualized using OpenSearch Dashboards, Kibana, or Grafana.



Rucio Monitoring - Transfer/deletion monitoring



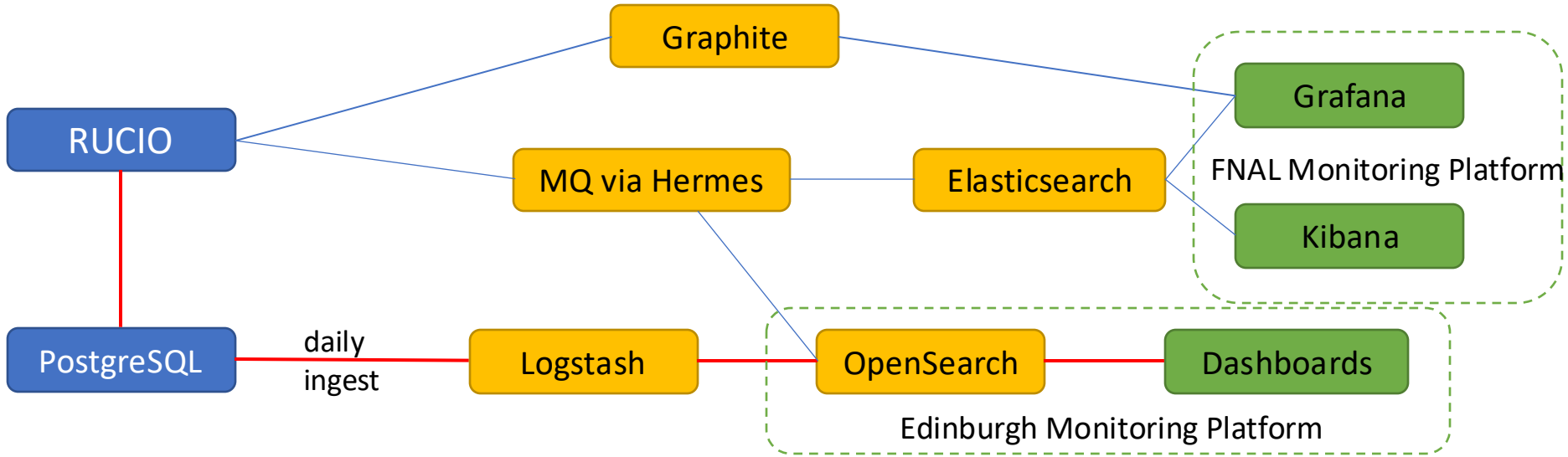
Rucio transfers per source RSE



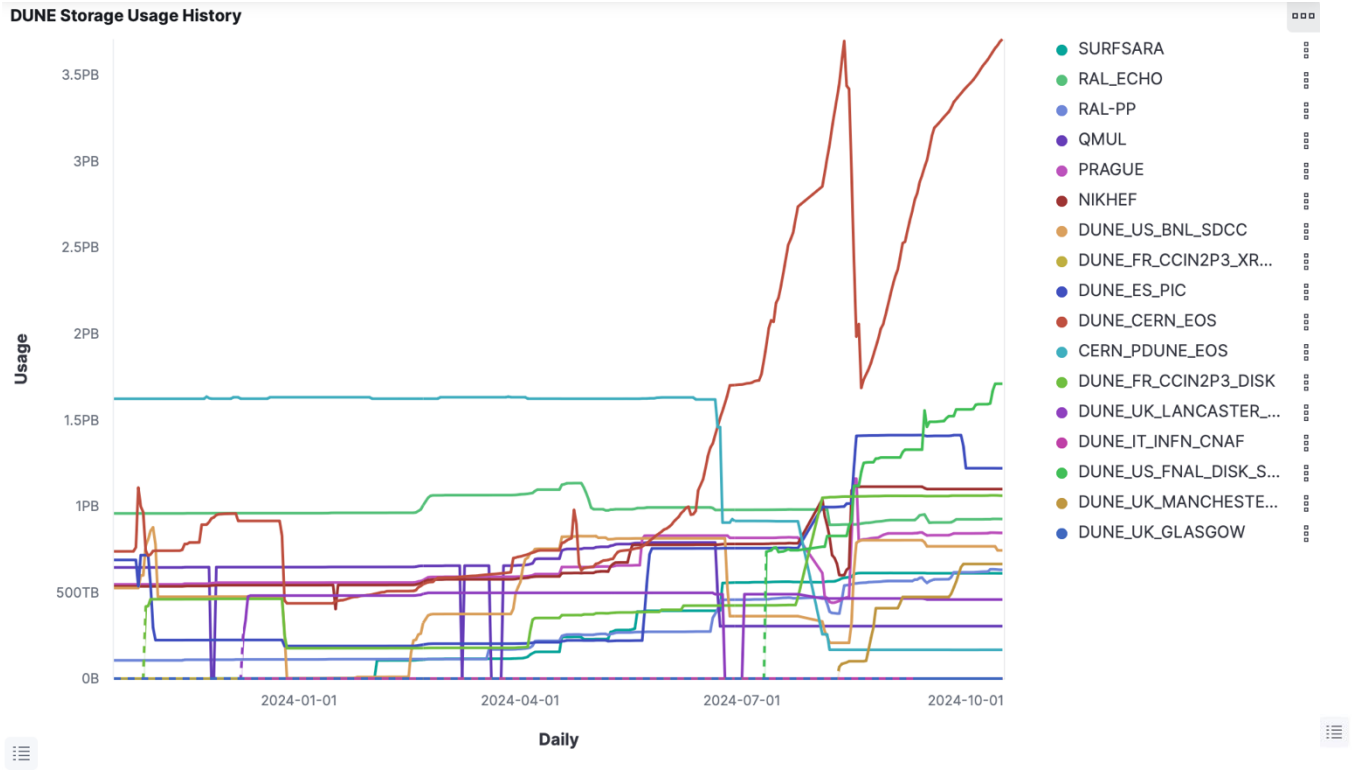
Rucio transfers status

Rucio Monitoring - database dumping

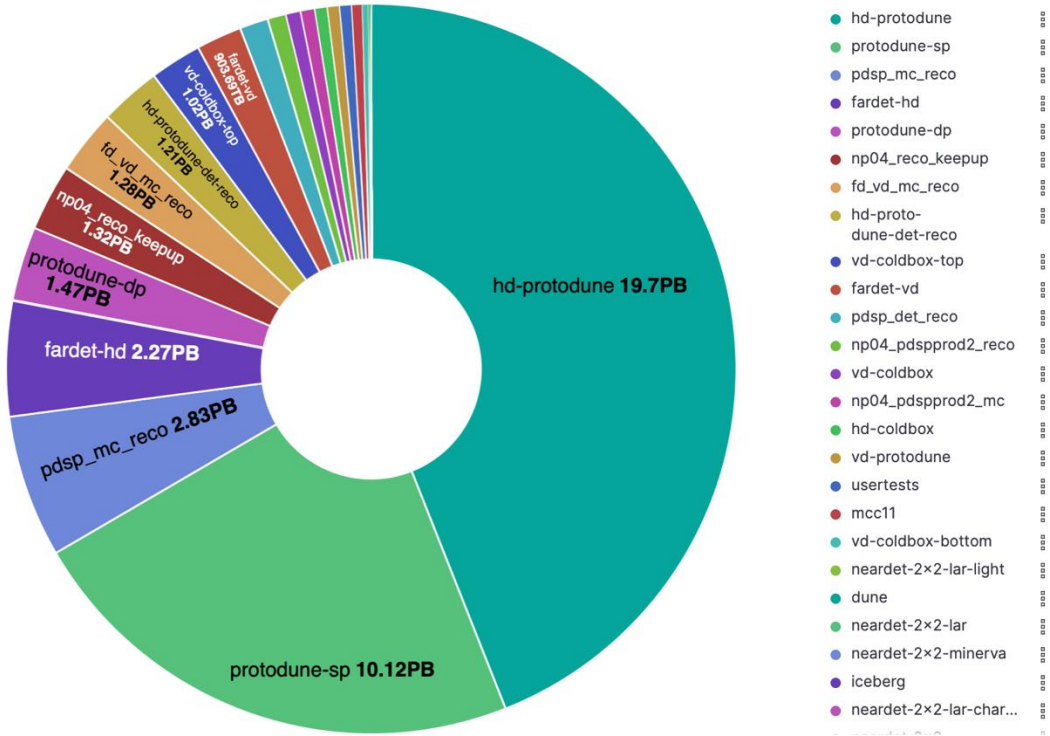
- DUNE daily dumped Rucio internal database directly to OpenSearch at Edinburgh
- Information like data location, accounting, RSE summary and history could be visualised using OpenSearch Dashboards/Grafana
- Storage summary is very helpful for various purpose, e.g. computing resource request, computing/physics coordinate, funding purpose



Rucio Monitoring - database dumping



RSE Usage in the past year



Rucio scope pie

Summary

- DUNE has adopted and deployed Rucio, as an essential component of its Distributed Data Management system to handle large-scale data
- DUNE Rucio development has involved developing various features to the Rucio code base to meet DUNE's specific needs
- Implementation of general functionalities that are crucial for DUNE and potentially beneficial for other experiments/communities
- DUNE Rucio monitoring has deployed various visualization components at Fermilab and Edinburgh, essential for DUNE data transfer and management development.