Rucio: Open-source Scientific Data Curation

FERMILAB-POSTER-22-187-STUDENT : Jason Wyatt, OMNI Fermilab Intern

Rucio Daily Transfer Test

- This transfer test is a containerized application, written in the Python programming language, that generates files and tracks their transfer success rate to Rucio
- This is accomplished by using the RabbitMQ message-broker in combination with STOMP (simple text-oriented message protocol)
- Rucio tackles the curation of data across a variety of distributed storage systems for the scientific community, which perform experiments that generate large volumes of data at high velocity. The veracity of this data must be maintained, while being made available and accessible for value extraction by authenticated and authorized client requests world-wide

Namespace: Data Identifiers (DID)

- DIDs are a naming scheme that consists of a unique combination of scope and name, each stored as a string

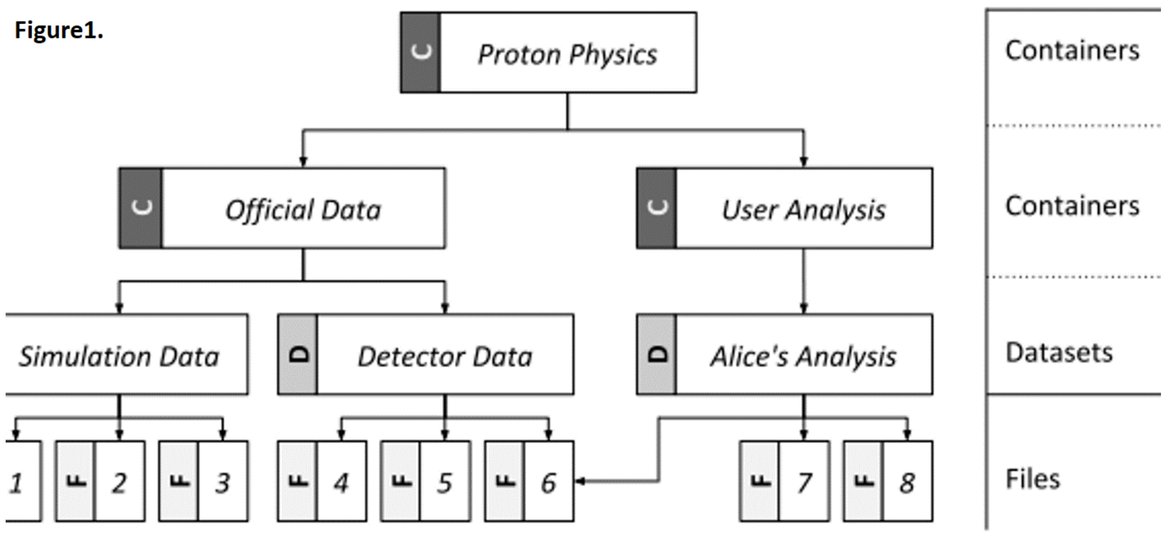
(“scope” : “name”)

- These fields can contain metadata that reflect the experiment, as well as file format and other identifiers
- In order to prevent accidental modification or data exchange, DIDs may never be reused to point to any other data
- Therefore, if data is purposefully modified, a new DID must be assigned

DID Granularity: Files, Datasets, and Containers

- ***Files*** are Rucio’s smallest units of operation
- ***Datasets*** organize files by grouping potentially distributed sets of files into one logical unit for bulk deletions and transfers
- Files and Datasets may be grouped into ***Containers*** which are used to keep a logical organization of experiment data

Figure1.



Storage: Rucio Storage Elements (RSE)

- A ***replica*** is a physical instance of a file defined by a DID logical abstraction
- RSEs (Rucio Storage Elements) are a logical abstraction of a storage system. The main RSE attributes are hostname, port, protocol, and local file system path, which are required to access storage space
- After DIDs are created, replication rules are defined which control their life-cycle by protecting them from final replica deletion while the rule remains extant

Authorizing and Authenticating Accounts

- X509 is a cryptographic authentication standard that spells out the format of public key certificates. This certificate binds an identity to a public key with a digital signature
- The Rucio server relies on trusted certification chains for secure connection assurance. The Certification Authorities present in these chains verify the ownership of a public key by the named subject of the certificate
- In the Daily Transfer Test, a client’s certification and key pair are used to establish a broker connection to a specified broker host ipaddress and port
- In the case of containerization, this private data should be added to a Docker image as a mounted volume
- Adding the certifications as mounted volumes keeps the sensitive information out of images pushed to Github and Dockerhub, and from being exposed in CLI calls to

docker inspect [image name]

- A short-lived authentication token is generated by Rucio upon successful authentication. This X-Rucio-Auth-Token can be reused for any number of operations until expiration

Internship Experience

- The mentorship introduced open-source software technologies that form a framework for increasing productivity, workflow, version control, data security, and application independence
- Simultaneous multiple terminal session access in a single window with Tmux
- Mouse free text editing with Vim
- Message brokering with RabbitMQ and STOMP plug-in
- Distributed Version Control with Github
- Developed an understanding of X509 authentication in the Open Science Grid environment
- Application containerization with Docker

**Acknowledgements : ***This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics ** Figure1 data model image provided by CERN ***Thanks to My Mentor: Brandon J. White