Scientific data management with Rucio

Dr. Martin Barisits (CERN)

CERN Storage day
14 March 2024

CERN

Rucio in a nutshell

Rucio provides a mature and modular scientific **data management federation**

- **Seamless integration** of **scientific and commercial** storage and their network systems
- Data is stored in **global single namespace** and can contain **any potential payload**
- Facilities can be **distributed at multiple locations** belonging to **different administrative domains**
- Designed with **more than a decade of operational experience** in very large-scale data management

Rucio is location-aware and manages data in a heterogeneous distributed environment

- Creation, location, transfer, deletion, annotation, and access
- **Orchestration of dataflows** with both low-level and high-level policies

Principally developed by and for the ATLAS Experiment, now with many more communities

Rucio is free and open-source software licenced under *Apache v2.0*

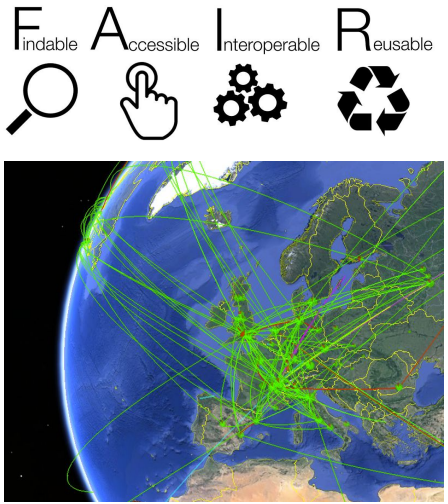
Open community-driven development process

Rucio main functionalities

Provides many features that can be enabled selectively

More advanced features

- **Horizontally scalable catalog** for files, collections, and metadata
- Transfers between facilities including **disk, tapes, clouds, HPCs**
- **Authentication and authorisation** for users and groups
- **Many interfaces** available, including CLI, web, FUSE, and REST API
- **Extensive monitoring** for all dataflows
- Expressive **policy engine** with rules, subscriptions, and quotas
- Automated **corruption identification and recovery**
- Transparent support for **multihop, caches, and CDN dataflows**
- **Data-analytics based flow control**



Rucio is not a distributed file system, it connects existing storage infrastructure over the network

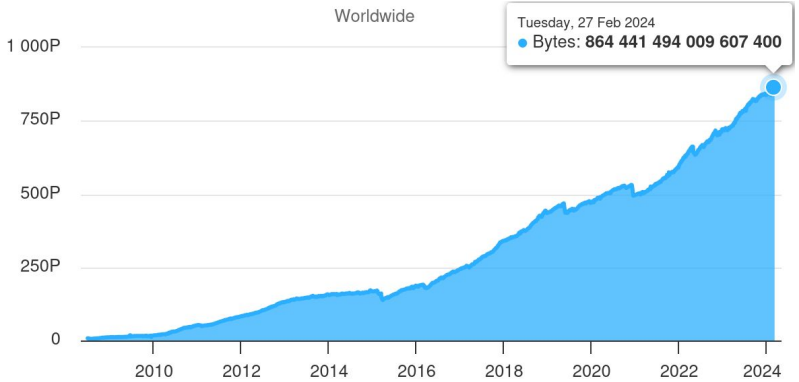
No Rucio software needs to run at the data centres

Data centres are free to choose which storage system suits them best - avoids vendor lock-in

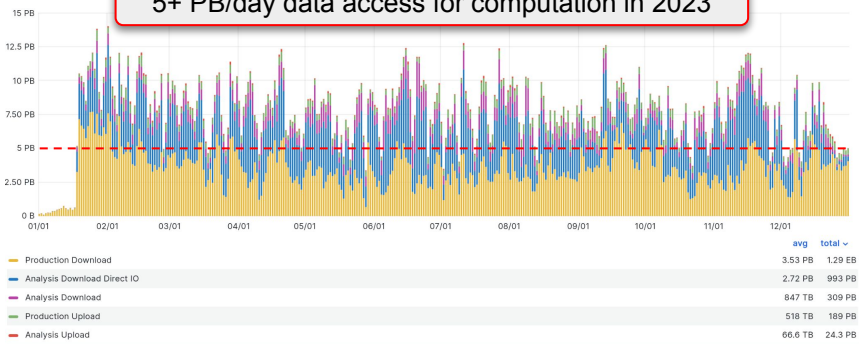
Data transfer rates

A few numbers showing the ATLAS scale

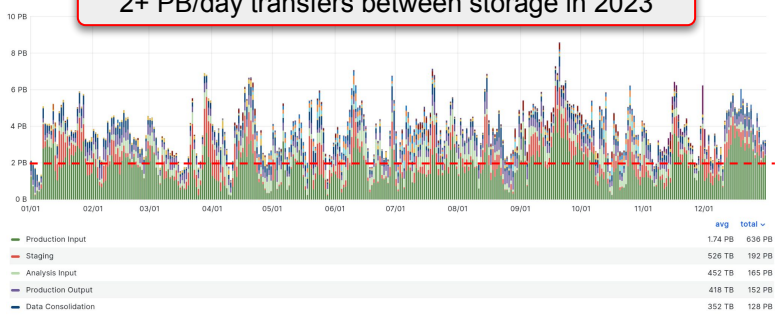
- 1B+ files, 700+ PB of data, 400+ Hz interaction
- 120 data centres, 5 HPCs, 3 clouds, 1000+ users
- 1.5 Exabytes/year transferred
- 3 Exabytes/year uploaded & downloaded



5+ PB/day data access for computation in 2023



2+ PB/day transfers between storage in 2023



Declarative data management

Express what you want, not how you want it

e.g., *"Three copies of this dataset, distributed across MULTIPLE CONTINENTS, with at least one copy on TAPE"*

e.g., *"One copy of this file ANYWHERE, as long as it is a very fast DISK"*

Replication rules

Rules can be **dynamically added and removed** by all users, some pending **authorisation**

Evaluation **engine resolves all rules** and tries to satisfy them by requesting transfers and deletions

Lock data against deletion in particular places for a given lifetime

Cached replicas are **dynamically created replicas** based on traced usage over time

Workflow system can drive rules automatically, e.g., **job to data flows** or vice-versa

Subscriptions

Automatically generate rules for newly registered data matching a **set of filters or metadata**

e.g., *"All derived products from this physics channel must have a copy on TAPE"*

Rucio concepts - Metadata

Rucio supports storage and querying of metadata

- Generic metadata that can be set by the users
- Up to the community to define the schema
- Searchable via name and metadata, aggregation based on metadata searches

Metadata interfaces

- Per default, generic metadata stored “within” Rucio (json data types)
- Metadata interfaces enable communities to connect other metadata backends (mongodb, science specific metadata stores, …)
- Metadata queries against Rucio are internally relayed to the matching backend and aggregated

Generic metadata can be restricted

- Enforcement possible by types and schemas
- Naming convention enforcement and automatic metadata extraction

Operations model

Objective was to minimise the amount of human intervention necessary

Large-scale and repetitive operational tasks can be automated

- Bulk migrating/deleting/rebalancing data across facilities at multiple institutions
- Popularity driven replication and deletion based on data access patterns
- Management of disk spaces and data lifetime
- Identification of lost data and automatic consistency recovery

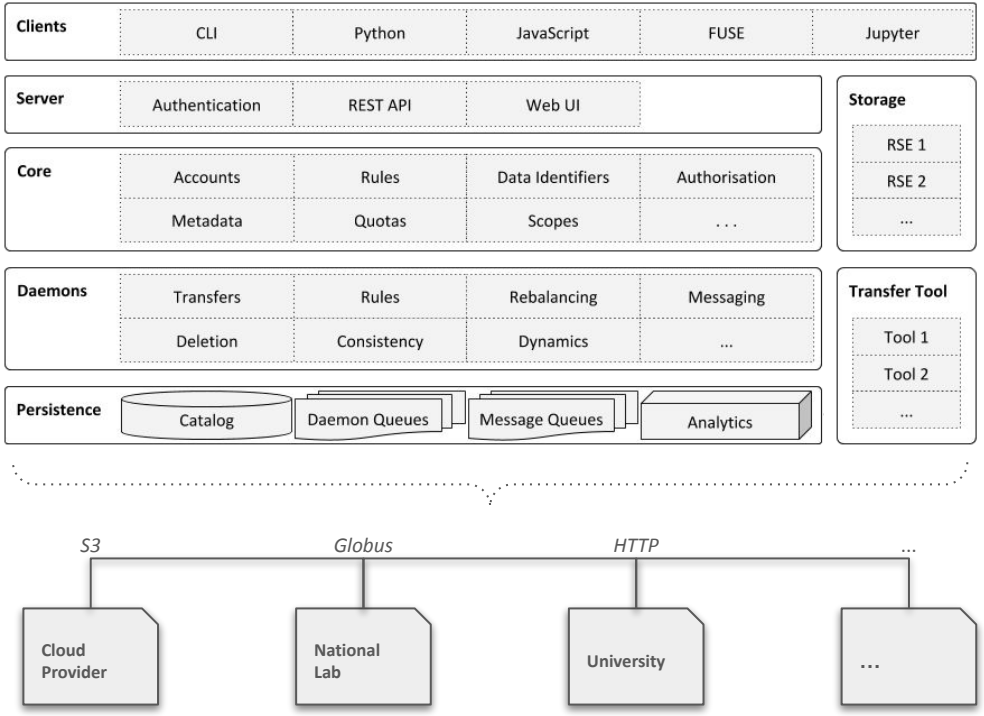
Administrators at the sites are not operating any Rucio service

- Sites only operate their storage exposed via protocols (POSIX, ROOT, HTTP, WebDAV, S3, gsiftp, …)
- Users have transparent access to data in a federated way

Easy to deploy

- PIP packages, Docker containers, helm-charts, Kubernetes

High-Level Architecture



Horizontally scalable component-based architecture

Servers interact with users
- HTTP API using REST/JSON
- Strong security (X.509, SSH, GSS, OAuth2, …)
- Many client interfaces available

Daemons orchestrate the collaborative work
- Transfers, deletion, recovery, policy, …
- Self-adapting based on workload

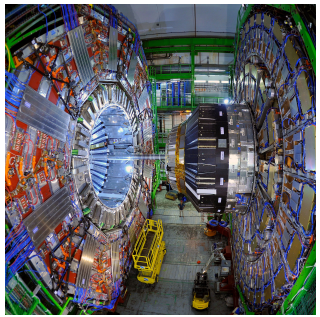
Messaging support for easy integration
- STOMP / ActiveMQ-compatible protocol

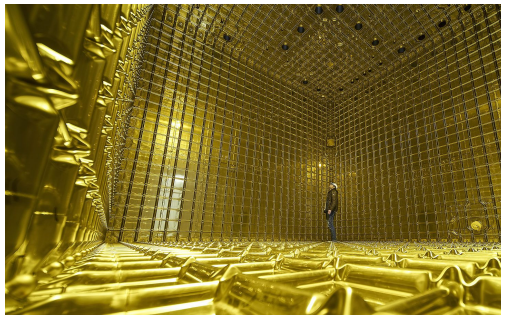
Persistence layer
- Oracle, PostgreSQL, MySQL/MariaDB, SQLite
- Analytics with Hadoop and Spark

Middleware
- Connects to well-established products, e.g., FTS3, XRootD, dCache, EOS, Globus, …
- Connects commercial clouds (S3, GCS, AWS)

Community experiences

- Rucio has become the de-facto standard for open scientific data management
  - Used by CERN-based experiments — AMS, ATLAS, CMS
  - And non-CERN experiments — Belle II, CTAO, LBNF/DUNE, SBN/ICARUS, KIS Solar, LIGO/VIRGO/KAGRA, Vera Rubin Observatory, XENON, …
  - Under evaluation by many others — EIC/ePIC, KM3NeT, SKA, …
  - Long tail of science — Fermilab, RAL, CERN run Rucio as a service for small experiments
  - Used by several EU projects — ESCAPE, InterTwin, DaFab



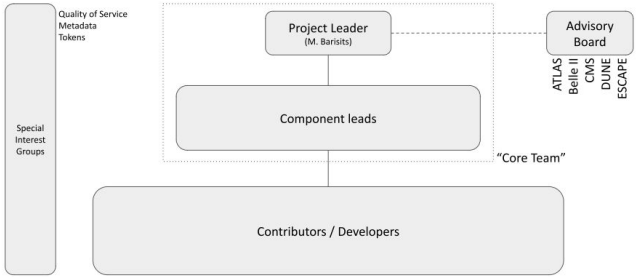

Community-driven development

Behind Rucio stands a strong open-source community helping each other across sciences

- Weekly meetings on development and operational issues
- Requirements, features, issues, releases are **publicly discussed**
- **Component leads** (core team) in charge of maintenance and coordination of contributions
- Focus on **scalability** and **long-term sustainability** of the software
- Automation and containerisation of development **lowers barrier of entry** for newcomers

Project is steered by the communities scientific needs

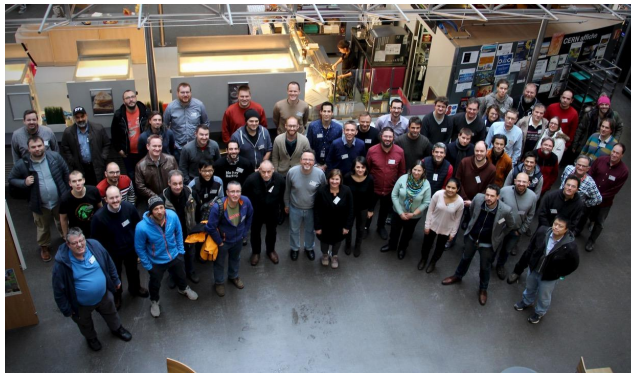
- Yearly **community workshops** to assess scientific requirements
- **Advisory board** to advise on long-term direction of the project
- **Special interest groups** to coordinate hot topics in the project



Regular events

Community Workshops

- CERN, Switzerland [2018]
- University of Oslo, Norway [2019]
- Fermilab, USA [2020]
- Remote [2021]
- Lancaster University, UK [2022]
- KEK, Japan [2023]
- SDSC, USA [2024]



Summary

Rucio is an open, reliable, and efficient data management system

- Supporting the world's largest scientific experiments, but also a good match for smaller sciences
- Extended continuously for the growing needs and requirements of the sciences

Strong cooperation between physics and multiple other fields

- Diverse communities have joined, incl. astronomy, atmospheric, environmental, …
- Community-driven innovations to enlarge functionality and address common needs

Benefit from advances in both scientific computing and industry

- Lower the barriers-to-entry by keeping control of data in scientist hands
- Seamless integrations with scientific infrastructures and commercial entities
- Detailed monitoring capabilities and easy deployment have proven crucial

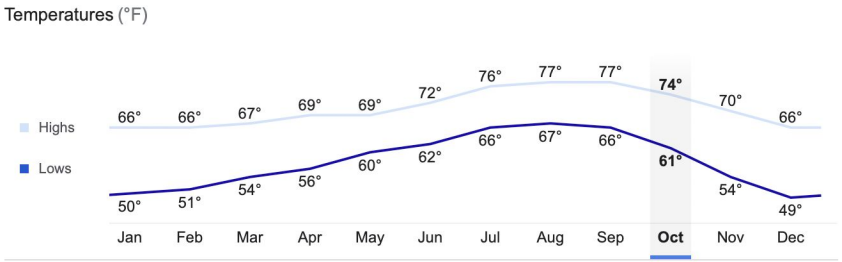
Additional information

Website	http://rucio.cern.ch
Documentation	https://rucio.cern.ch/documentation
Repository	https://github.com/rucio/
Images	https://hub.docker.com/r/rucio/
Online support	http://rucio.cern.ch/doc../join_rucio_mattermost/
Contact	rucio-contact@cern.ch
Journal article	https://doi.org/10.1007/s41781-019-0026-3
Twitter	https://twitter.com/RucioData

Rucio Community Workshop 2024

Sep 30 - Oct 4, 2024
San Diego Supercomputer Center (SDSC)
San Diego, California, US

https://indico.cern.ch/event/1343110/




CERN

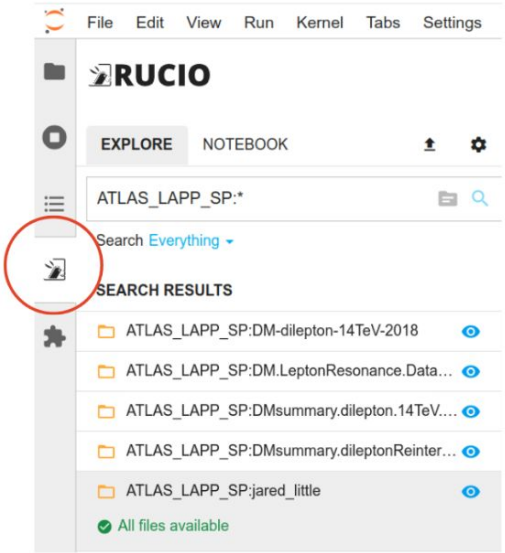
Rucio Jupyterlab Extension

Jupyterlab notebooks tool of choice for data analysis for today's scientists

Rucio Jupyterlab Extension enables users to

- Discover Rucio data from within the notebooks
- Select data and replicate it into the notebook environment
- Import data
- Run analysis on the data

One step further: Virtual Research Environments



The Virtual Research Environment, E. Gazzarrini, CHEP 2023

Monitoring & analytics

RucioUI

- Provides several views for different types of users
- Normal users: Data discovery and details, transfer requests and monitoring
- Site admins: Quota management and transfer approvals
- Central administration: Account / Identity / Site management

Monitoring

- Internal system health monitoring with Graphite / Grafana
- Transfer / Deletion / … monitoring built on HDFS, ElasticSearch, and Spark
- Messaging with STOMP

Analytics and accounting

- e.g., Show which the data is used, where and how space is used, …
- Data reports for long-term views
- Built on Hadoop and Spark

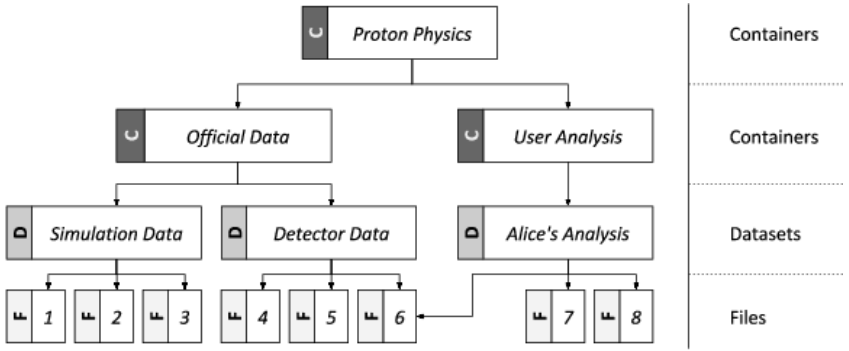
Rucio concepts - Namespace

All data stored in Rucio is identified by a Data IDentifier (DID)

There are different types of DIDs

Files	
Datasets	Collection of files
Container	Collection of dataset and/or container

Each DID is uniquely identified and composed of a scope and name, e.g.:



`detector_raw.run34:observation_123.root`

scope name