Open-source scientific data management with Rucio

Martin Barisits (CERN), Thomas Beermann (DESY), Mario Lassnig (CERN)

PUNCHLunch
30 June 2022

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*

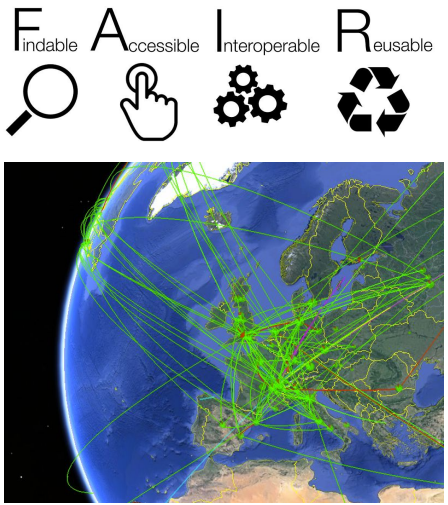
Open community-driven development process

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

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 based on data access patterns
- Popularity driven deletion
- Management of disk spaces and data lifetime
- Identification of lost data and automatic consistency recovery

Administrators at the sites are not operating any local Rucio service

- Sites only operate their storage
- Users have transparent access to all data in a federated way

Easy to deploy

- PIP packages, Docker containers, Kubernetes

Rucio concepts - Namespace with DIDs

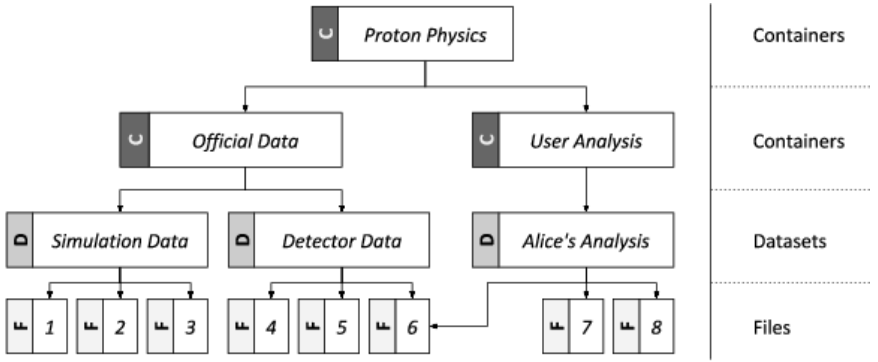
All data stored in Rucio is identified by a Data IDentifier (DID)

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

- Scope
- Name
- Example: `user.martin:test.file.001`



Rucio concepts - Metadata

Rucio supports different kinds of metadata backends via a plugin approach

- Column-based stores, generic json-based metadata, mongodb, …
- Plugin approach is adjustable to proprietary (experiment internal) metadata stores

Metadata are custom attributes on data identifiers

- Enforcement possible by type, e.g., enum
- Naming convention enforcement and automatic metadata extraction

Provides additional namespace to organise the data

- Searchable via name and metadata
- Aggregation based on metadata searches

Rucio concepts - RSEs

Rucio Storage Elements (RSEs) are logical entities of space

- No software needed to run at the site!
- RSE names are arbitrary (e.g., "CERN-PROD_DATADISK", "AWS_REGION_USEAST", …)
- Usually one RSE per site and storage data class

RSEs collect all necessary metadata for a storage system

- protocols, hostnames, ports, prefixes, paths, implementations, …
- data access priorities can be set (e.g. to prefer a protocol for LAN access)

RSEs can be assigned meta data

- Key/Value pairs (e.g., *country=UK*, *type=TAPE*, *support=brian@unl.edu*)
- You can use RSE expressions to describe a list of RSEs (e.g. `country=UK&type=TAPE`)

Rucio concepts - 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

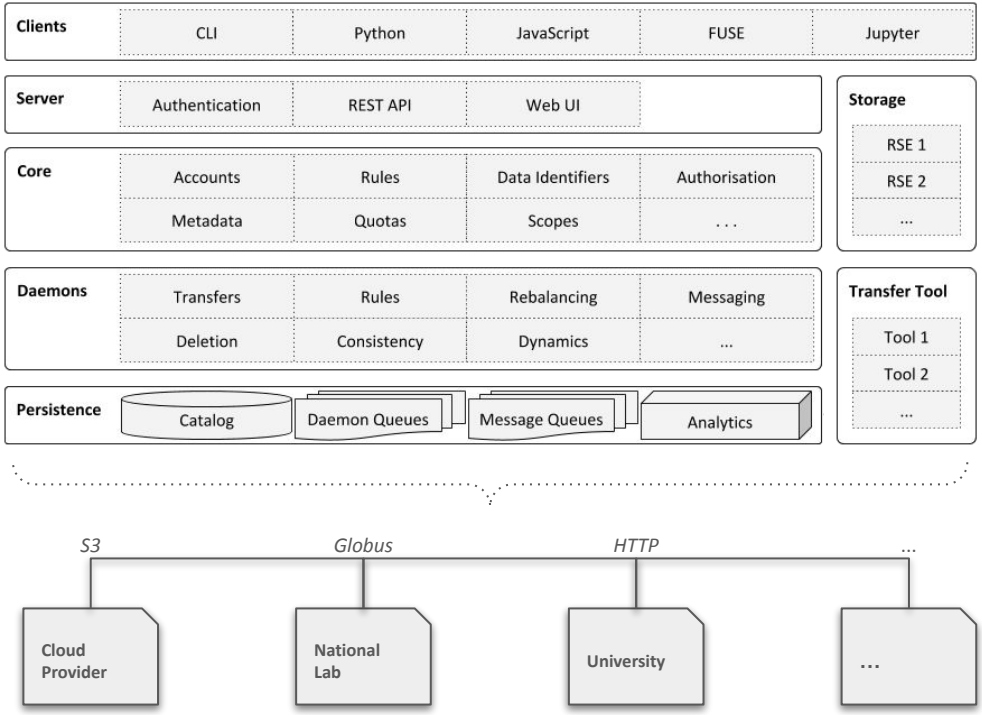
Workflow system can drive rules automatically, e.g., **job to data flows** or vice-versa

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"*

High-Level Architecture



Horizontally scalable component-based architecture

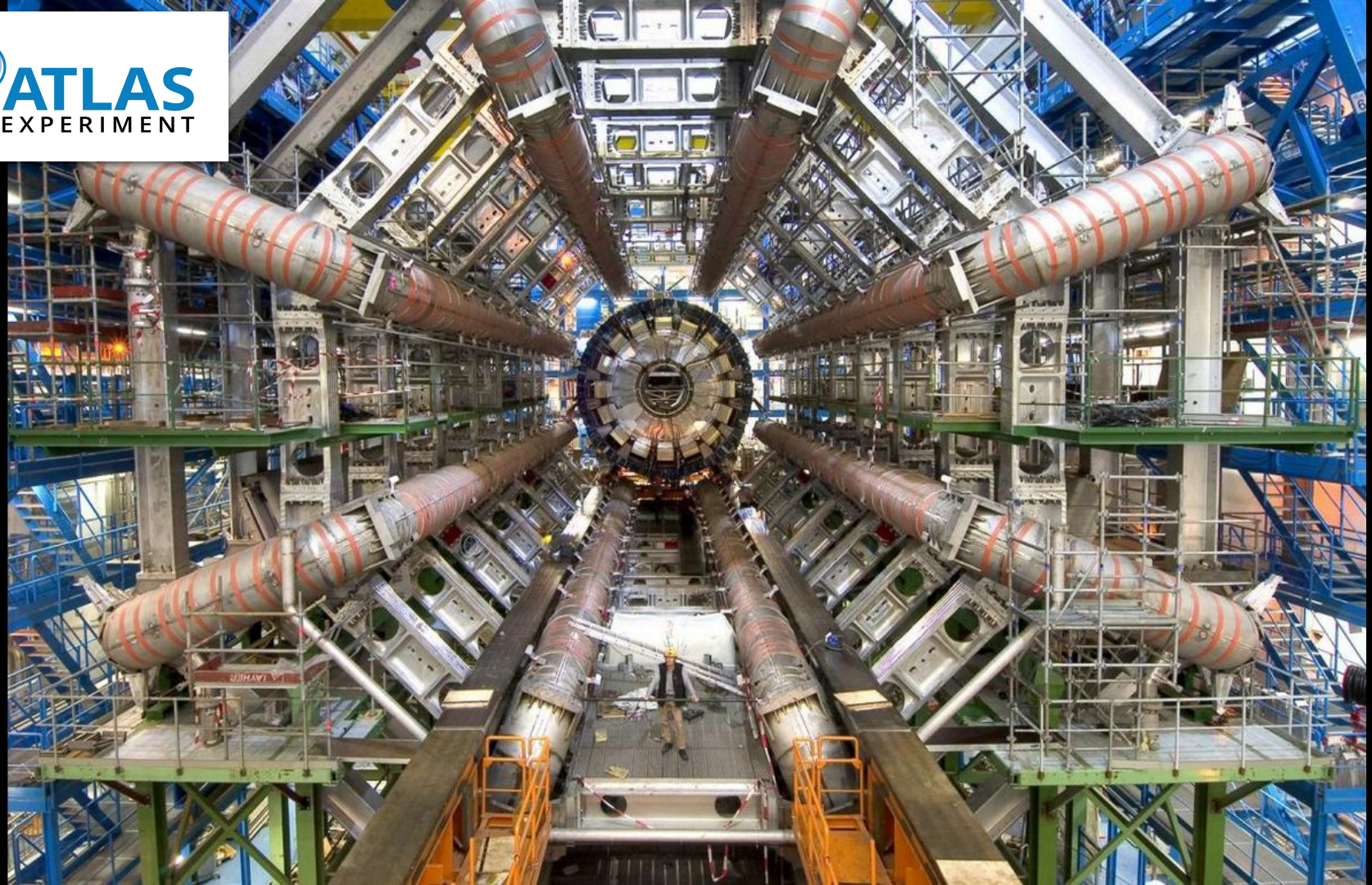
Servers interact with users
- HTTP API using REST/JSON
- Strong security (X.509, SSH, GSS, OAuth2, …)
- Many client interfaces available

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)

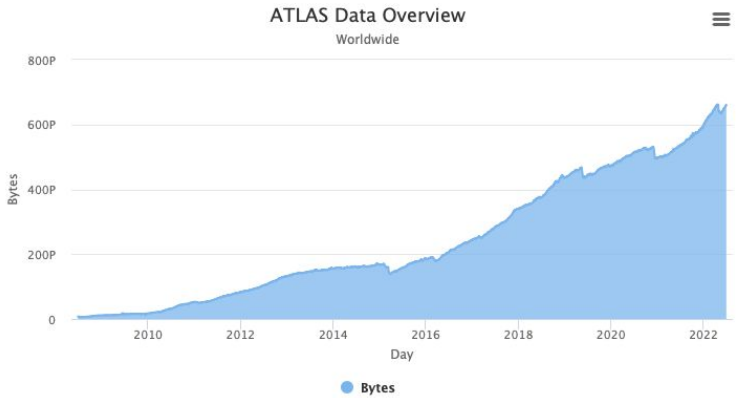

ATLAS
EXPERIMENT

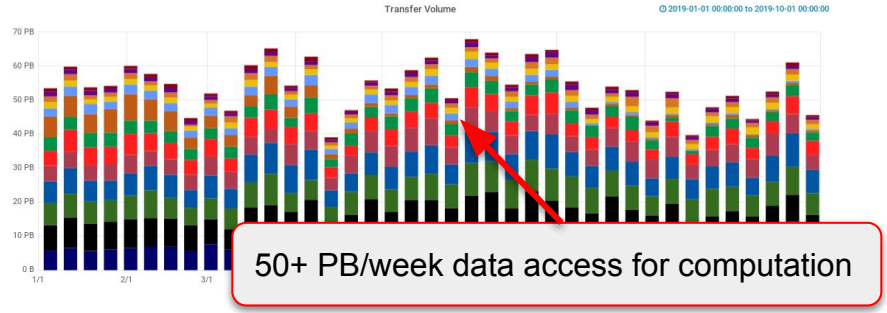
Scale

A few numbers showing the ATLAS scale

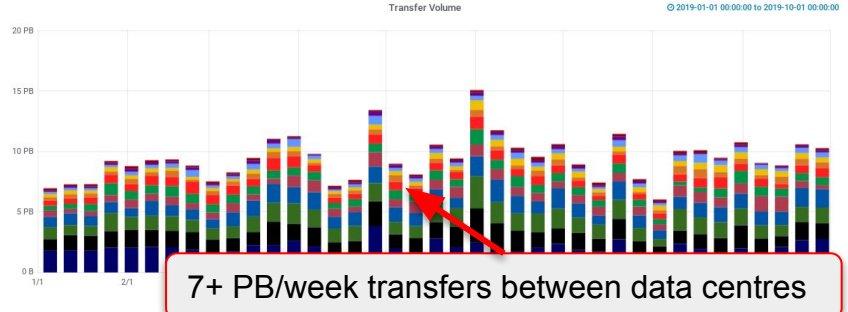
- 1B+ files, 500+ PB of data, 400+ Hz interaction
- 120 data centres, 5 HPCs, 2 clouds, 1000+ users
- 500 Petabytes/year transferred & deleted
- 2.5 Exabytes/year uploaded & downloaded

Increase 1+ order of magnitude for HL-LHC

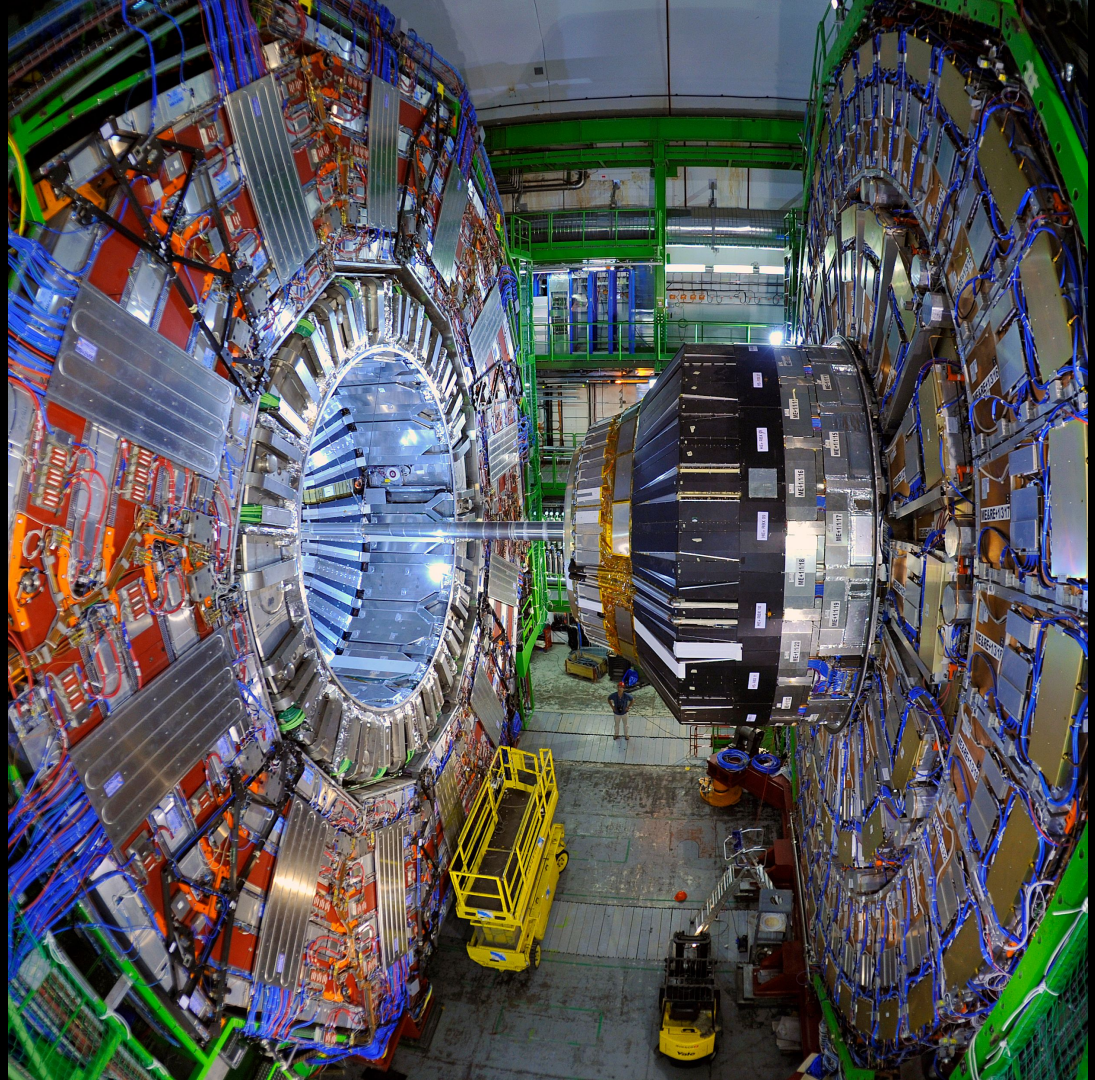




50+ PB/week data access for computation



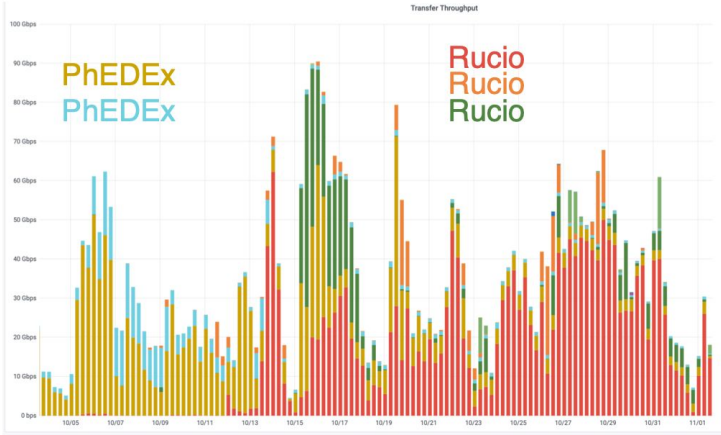
7+ PB/week transfers between data centres



Transition to Rucio

- June 2020 we transitioned our smallest NanoAOD to Rucio; almost no one noticed
  - This required a subset of what we need for a full Rucio transition and taught us valuable experience
  - Gave operators much needed experience in operating Rucio
- Full transition in November 2020
- Done with zero downtime of the production system.
  - Obviously non-trivial development to make this happen in other CMS software

FTS bandwidth at right shows a rapid transition with no let-up

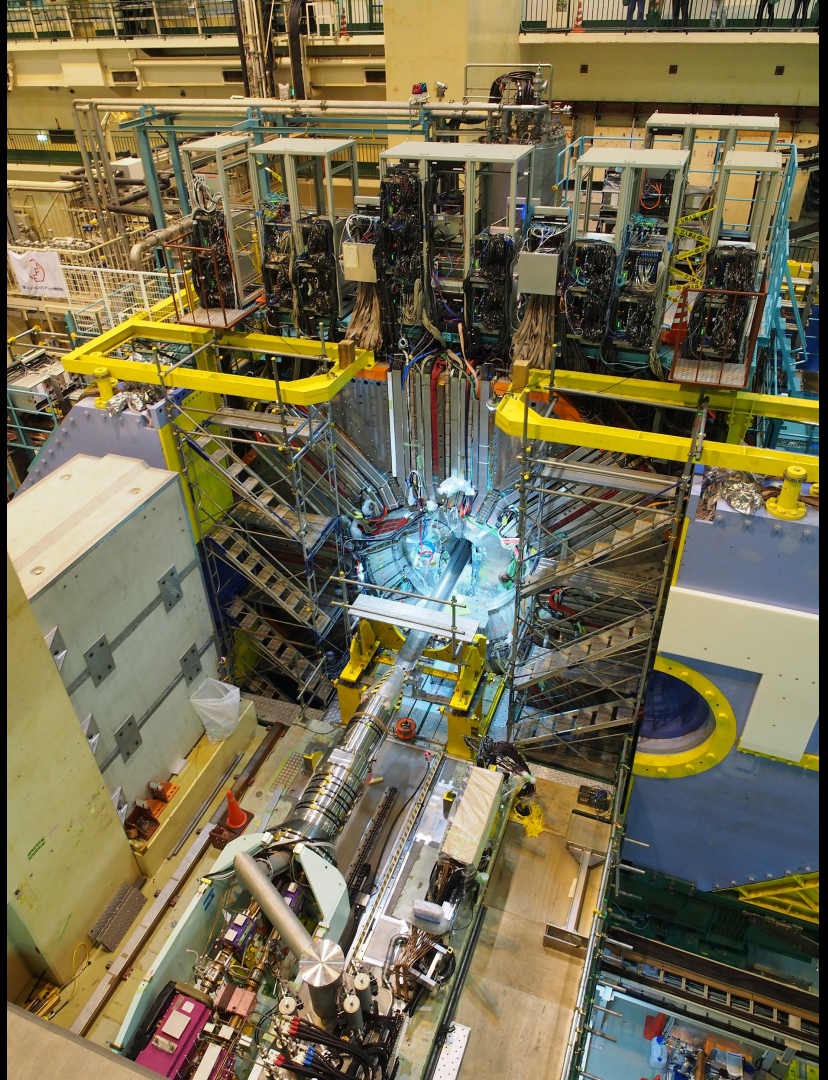


Eric Vaandering for CMS Rucio Team | Rucio Workshop 2021 | 3

CMS contributions to Rucio

- Our team is small, but we have still made some contributions to making Rucio a community project
  - CMS has been running on kubernetes since sometime in our evaluation process
    - ★ *Everything* except central services (database, message brokers, monitoring) is running in kubernetes
    - ★ Includes CMS specific components and necessary 3rd party components, *e.g.* Prometheus
    - ★ Hosted on CERN open stack
    - ★ So we've contributed quite a few extensions and updates as Rucio changes
    - ★ We are extremely pleased with this mode of operations
  - Monitoring probes
    - ★ Were ATLAS specific and centric. We've been systematically evaluating them, moving the relevant ones to common code base, and adding "missing" probes
  - Archived to tape verification
    - ★ Requested by our review, added to FTS, functionality added to Rucio to implement
  - Consistency checking
    - ★ Still a work in progress. CMS uses a different model so we refactored the entire process and are developing a toolkit which others can use to implement a consistency checks

Eric Vaandering for CMS Rucio Team Rucio Workshop 2021 6



Belle II

Belle II Rucio instance

- Rucio instance hosted at BNL using PostGreSQL
  - ~100M replicas/18PB registered (AFAIK, biggest PSQL instance running Rucio

- Interaction rate 80 to 100 Hz
- Hosts :
  - 2 servers
  - 2 daemons node
  - 1 WebUI node
  - 2 tracers (introduced recently)

Brookhaven National Laboratory

4

Belle II

Interfacing Rucio and DIRAC

- Integration of Rucio into BelleDIRAC :
  - Rucio File Catalog plugin + Synchronization agents
  - Many challenges due to the differences between Rucio and DIRAC (e.g. DIRAC user/group/host vs Rucio accounts)
  - The Catalog does a bit more than a catalog since it also creates replication rules
  - Rucio and BelleDIRAC work nicely together
- The code developed for BelleDIRAC was merged into Vanilla DIRAC
  - Further changes are being done by J. Martyniak (Imperial College London) to support multiVO and rucio.cfg less setup
  - It is important that what is implemented in Vanilla DIRAC can be used by Belle II

Brookhaven National Laboratory

8

Community


Advanced European Network of E-infrastructures for Astronomy with the SKA
LCLS
Belle II
Science & Technology Facilities Council
COMPASS
CERN
ARC
ATLAS EXPERIMENT
ESCAPE
European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures
Alpha Magnetic Spectrometer
AMS-02
Europe • Asia
North America
Xe
XENON
Dark Matter Project
CMS
Compact Muon Solenoid
DUNE
DEEP UNDERGROUND NEUTRINO EXPERIMENT
3D
EISCAT
egi
VIRGO
cta
cherenkov telescope array
LSST
Large Synoptic Survey Telescope
IceCube
SOUTH POLE NEUTRINO OBSERVATORY
FTS
Fermilab
PAWSEY
supercomputing centre
SKA
SQUARE KILOMETRE ARRAY
LIGO
FOLDING @HOME
eXtreme DataCloud
BROOKHAVEN
NATIONAL LABORATORY
Open Science Grid
EUXDAT

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

Open-source community project

Very active Slack channel

Weekly DevOps meetings, yearly community workshops, Hackathons

Software by and for the community

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 common 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	https://rucio.slack.com/messages/#support/
Developer contact	rucio-dev@cern.ch
Journal article	https://doi.org/10.1007/s41781-019-0026-3
Twitter	https://twitter.com/RucioData


CERN