Scientific Data Management with Rucio

Anton Schwarz (CERN)

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

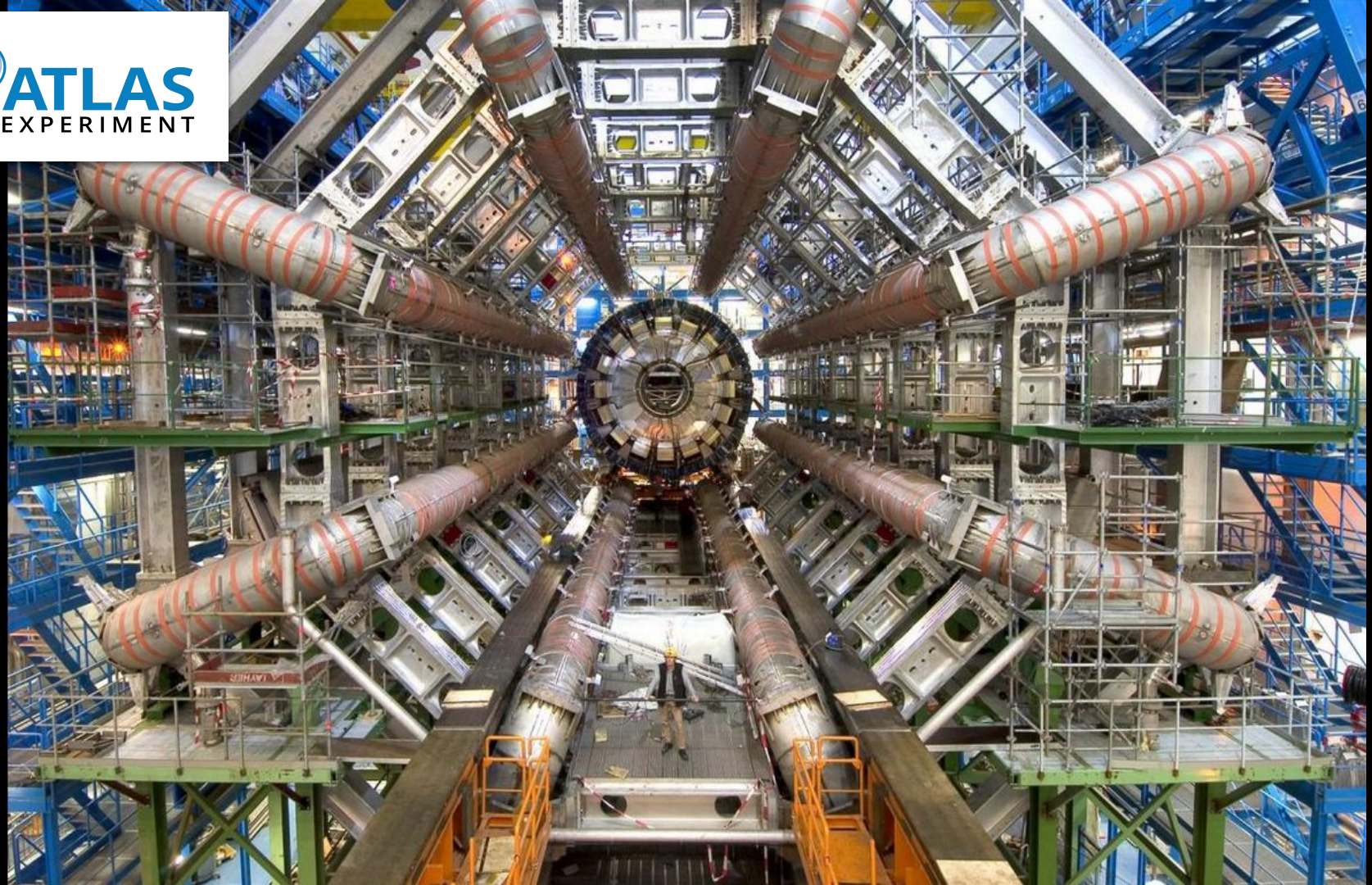
Rucio is location-aware and manages data in a heterogeneous distributed environment

- 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

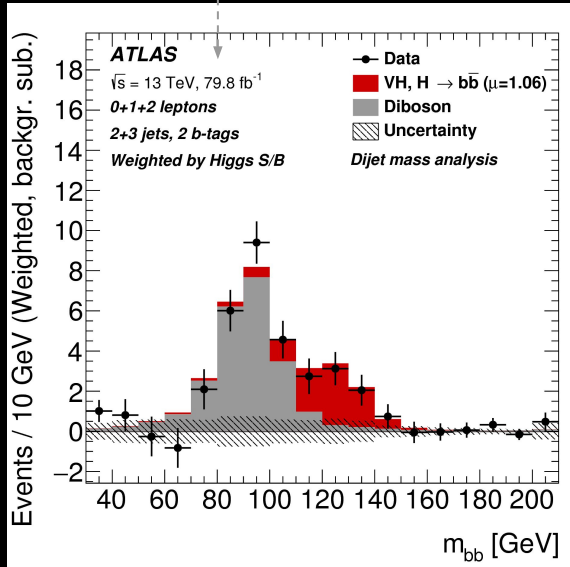

ATLAS
EXPERIMENT

ATLAS EXPERIMENT

Candidate Event:
$pp \rightarrow H(\rightarrow b\bar{b}) + W(\rightarrow \mu\nu)$
Run: 338712 Event: 335908183
2017-10-19 23:31:18 CEST

13 TeV detector data
8 quadrillion collision candidates
92 petabytes
130 million files

13 TeV simulation data
166 petabytes
544 million files



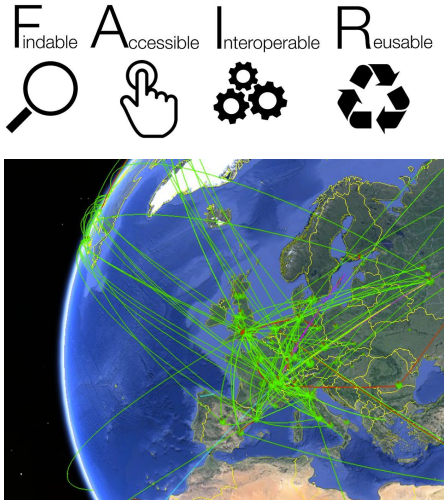
A candidate event display for the production of a Higgs boson decaying to two b-quarks (blue cones), in association with a W boson decaying to a muon (red) and a neutrino. The neutrino leaves the detector unseen, and is reconstructed through the missing transverse energy (dashed line). (Image: ATLAS Collaboration/CERN)

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

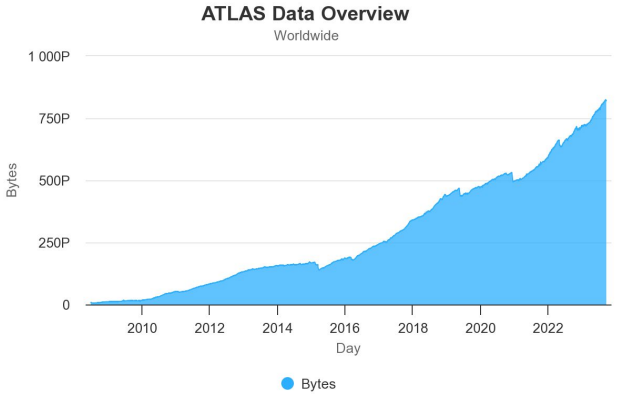
- No Rucio software needs to run at the data centres
- Data centres are free to choose which storage system suits them best

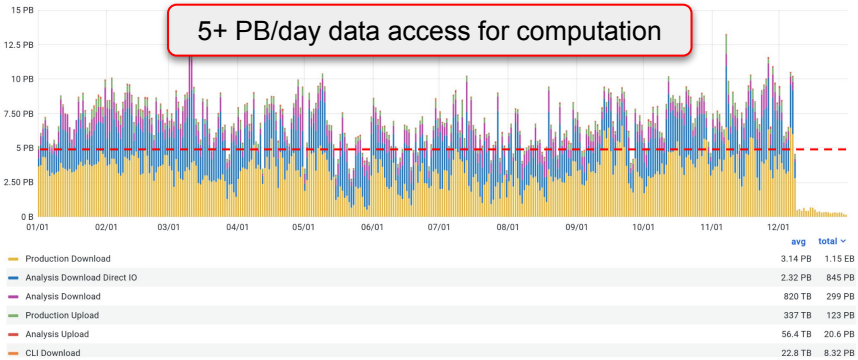
Data transfer rates

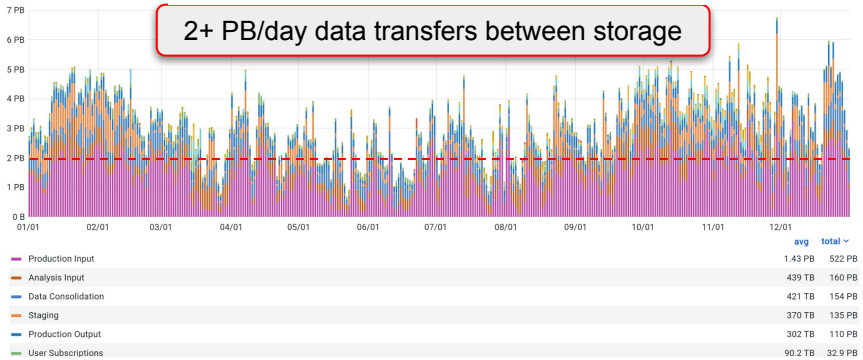
A few numbers showing the ATLAS scale

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

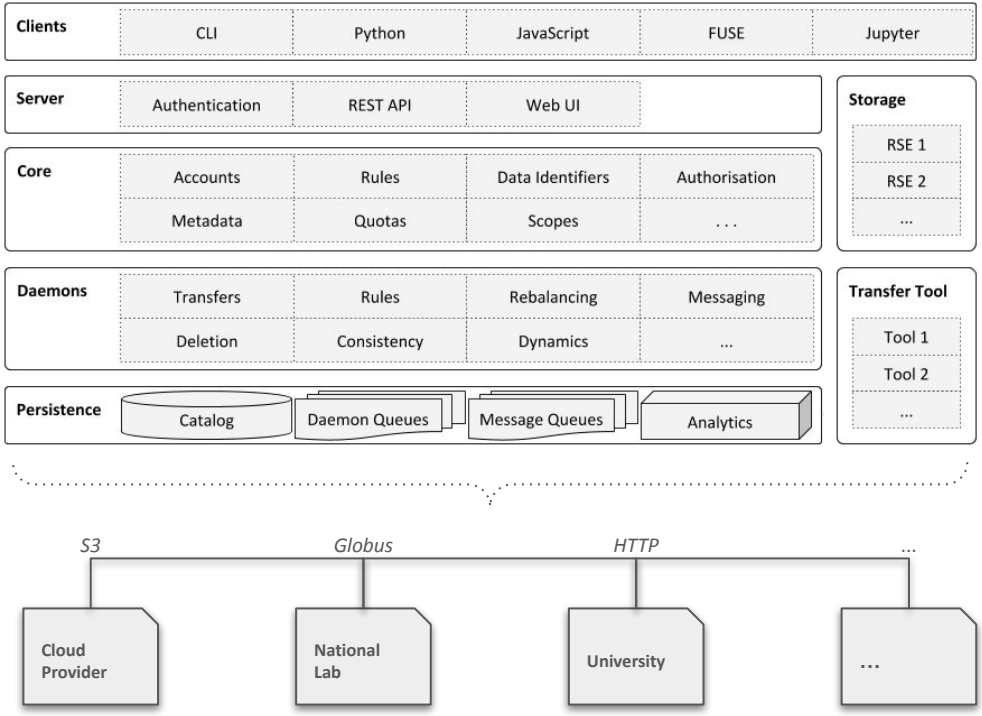
Increase 1+ order of magnitude for HL-LHC







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)

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"*

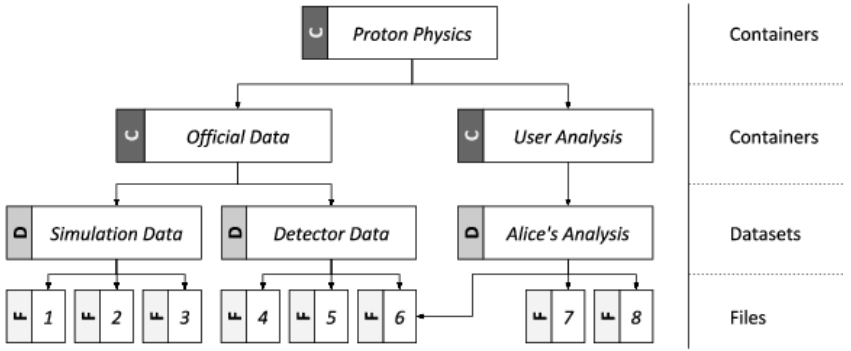
Rucio concepts - Namespace

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

Rucio concepts - Metadata

- Rucio supports different kinds of metadata
  - File internal metadata, e.g., *size, checksum, creation time, status*
  - Fixed physics metadata, e.g., *number of events, lumiblock, cross section, …*
  - Generic metadata that can be set by the users
  - Searchable via name and metadata, aggregation based on metadata searches
- Metadata interfaces
  - Allow Rucio to be connected to different metadata backends (json-column, mongodb, external, …)
  - 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 all data in a federated way
- Easy to deploy
  - PIP packages, Docker containers, Kubernetes

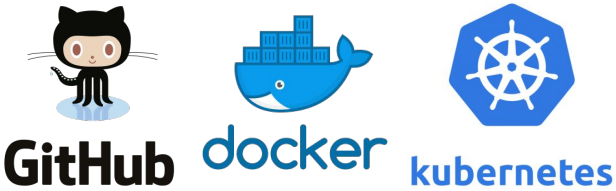
Monitoring & analytics

- 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

Community-driven development

- We have successfully moved to **community-driven development**
  - Requirements, features, issues, release are **publicly discussed** (e.g., weekly meetings, GitHub, MM) **Component leads** (core team) coordinate contributions from the community **design / guidance / review**
  - Usually 1-2 persons from a **community take responsibility** for a contribution to **develop** the software extension and also its **continued maintenance**

- Communities are helping each other **across experiments**
  - Effective across time zones due to US involvement
  - Automation, containerisation and documentation of development **lowers barrier of entry** for newcomers

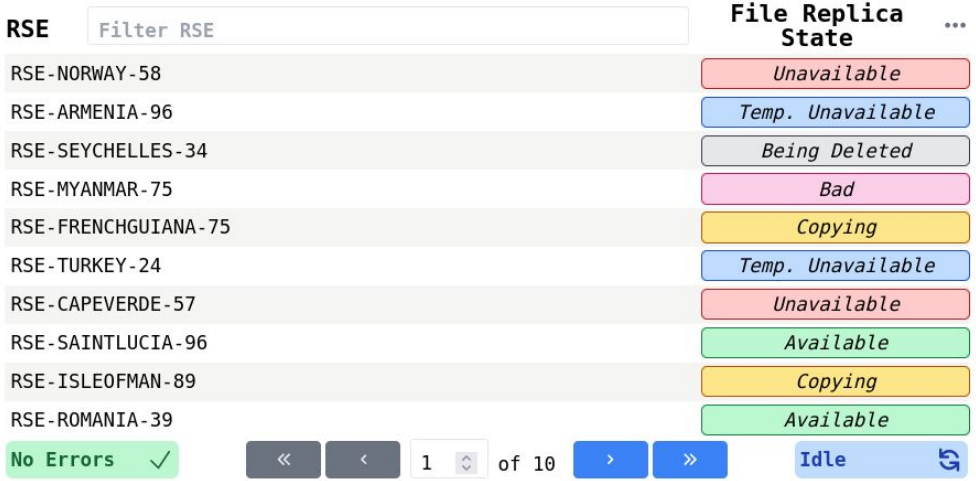


New WebUI

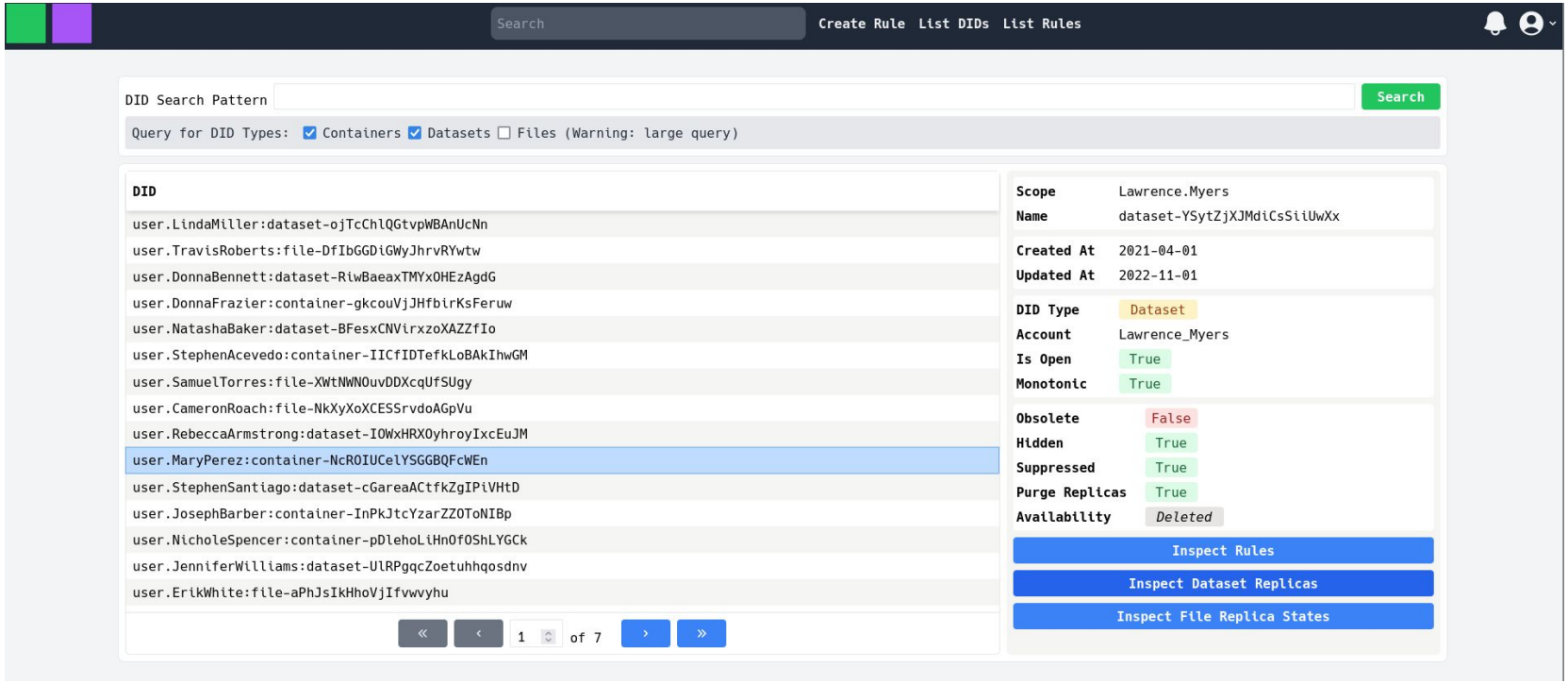
- Development of a new WebUI frontend
- Using NextJS backend and React frontend, fully typed using TypeScript
- Frontend requirements
  - Common design theme throughout
  - Dark/Light mode
  - Responsive design (goal: minimum size of 375x667px)
  - Accessible (vision-impaired, screen-reader accessible, etc.)
- Backend requirements
  - Framework-independent
  - Fully testable
  - UI-independent
  - “Independent” of Rucio-backend: not a direct mapping of Rucio endpoints

New WebUI Example: StreamedTable

- Requirement to show streamed data
  - Not implemented in old WebUI
  - Show data “as it is incoming”
- Data streams from Rucio already implemented
- Possible to garnish stream elements with data from additional queries
- “Streaming object” is passed from backend to frontend
- “StreamedTable” framework implements common table design and functionality



New WebUI Example: Full Page



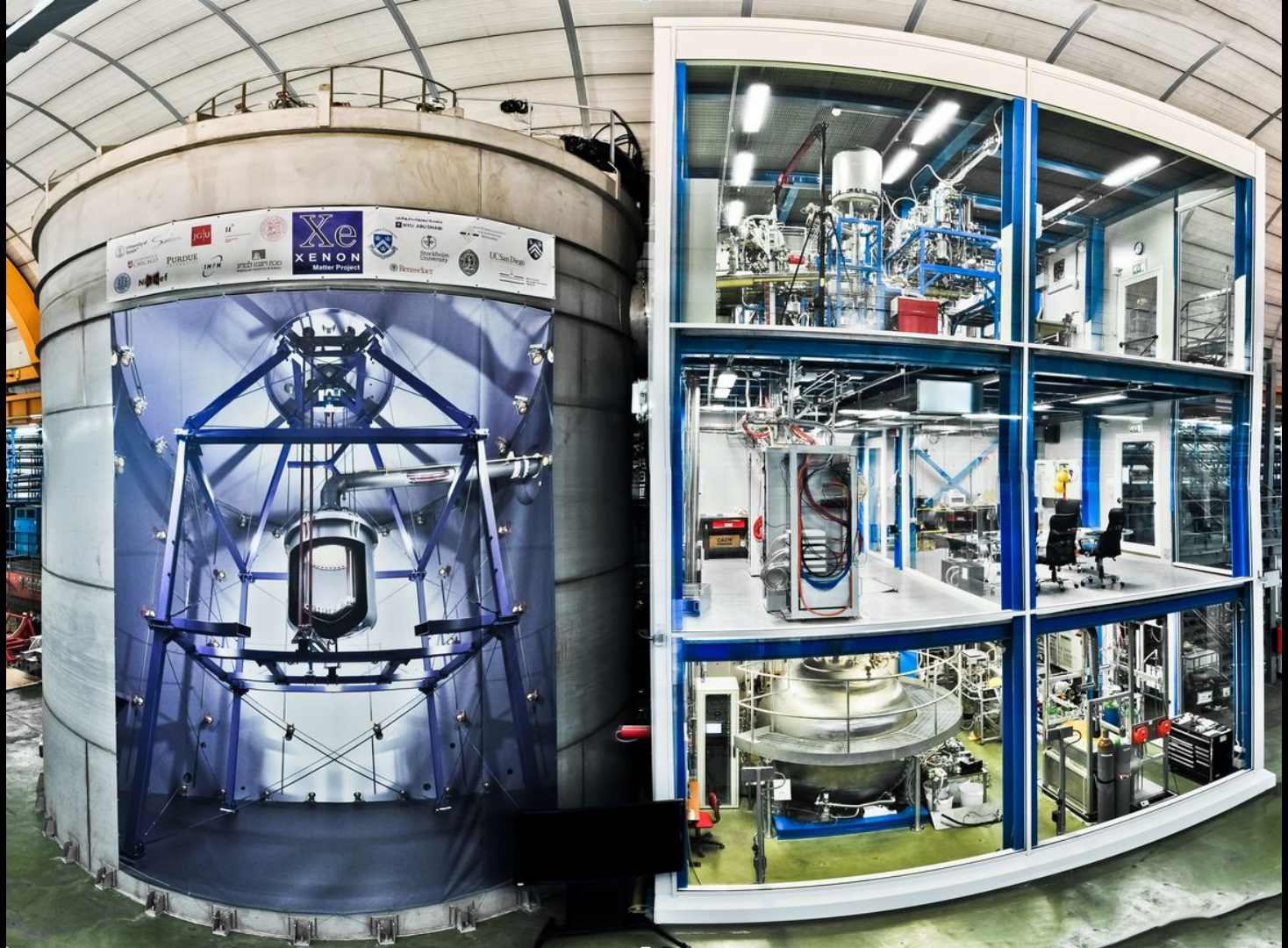
Concepts

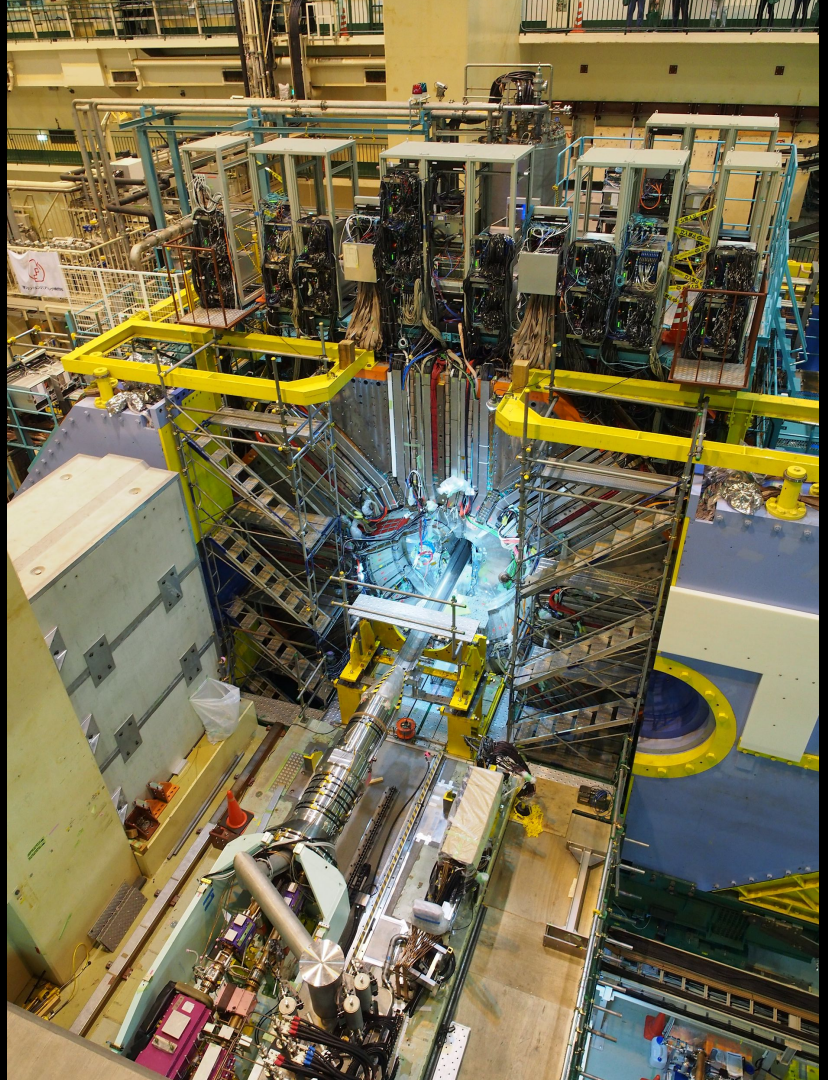
Rucio community experiences

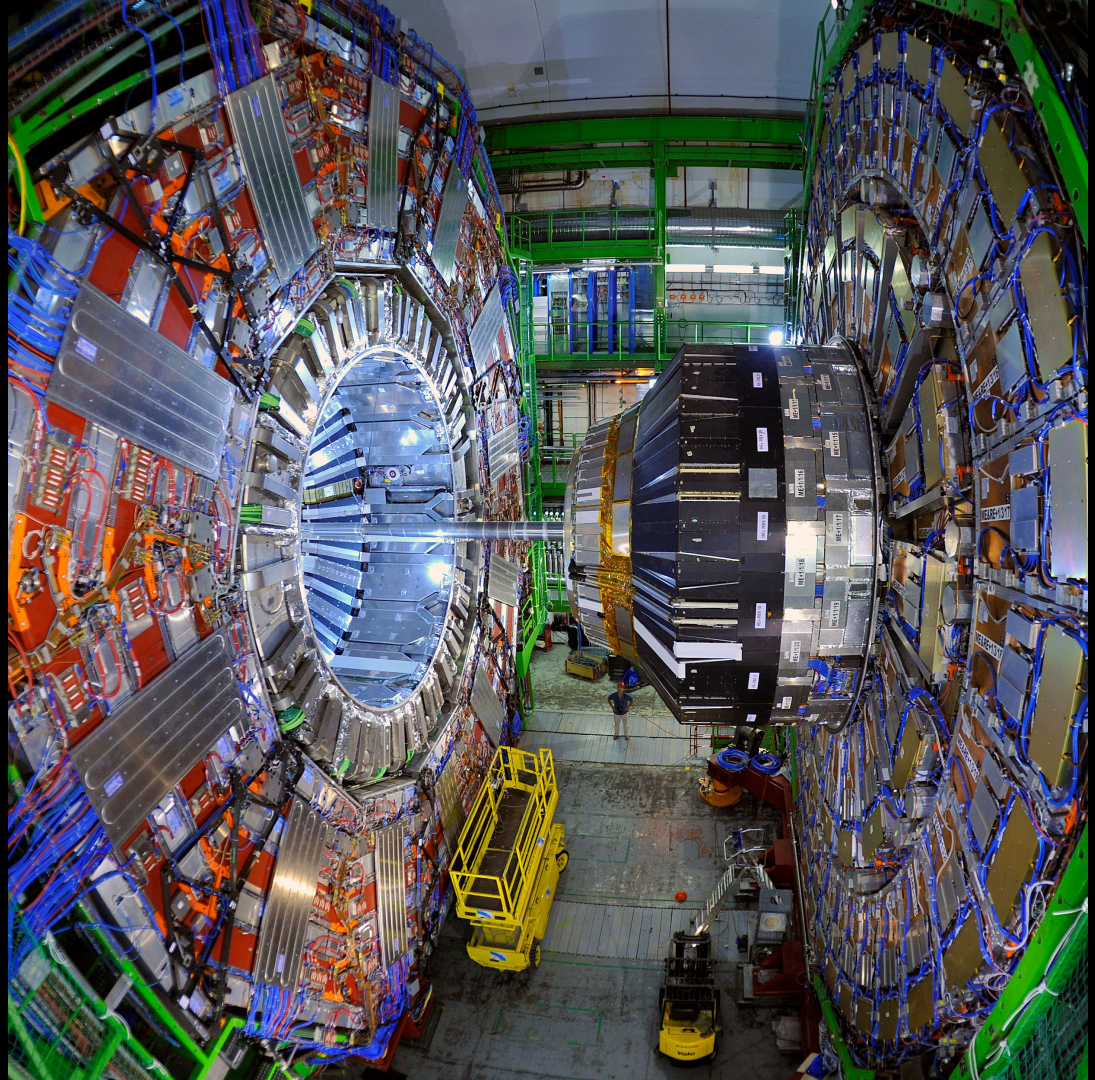
Summary

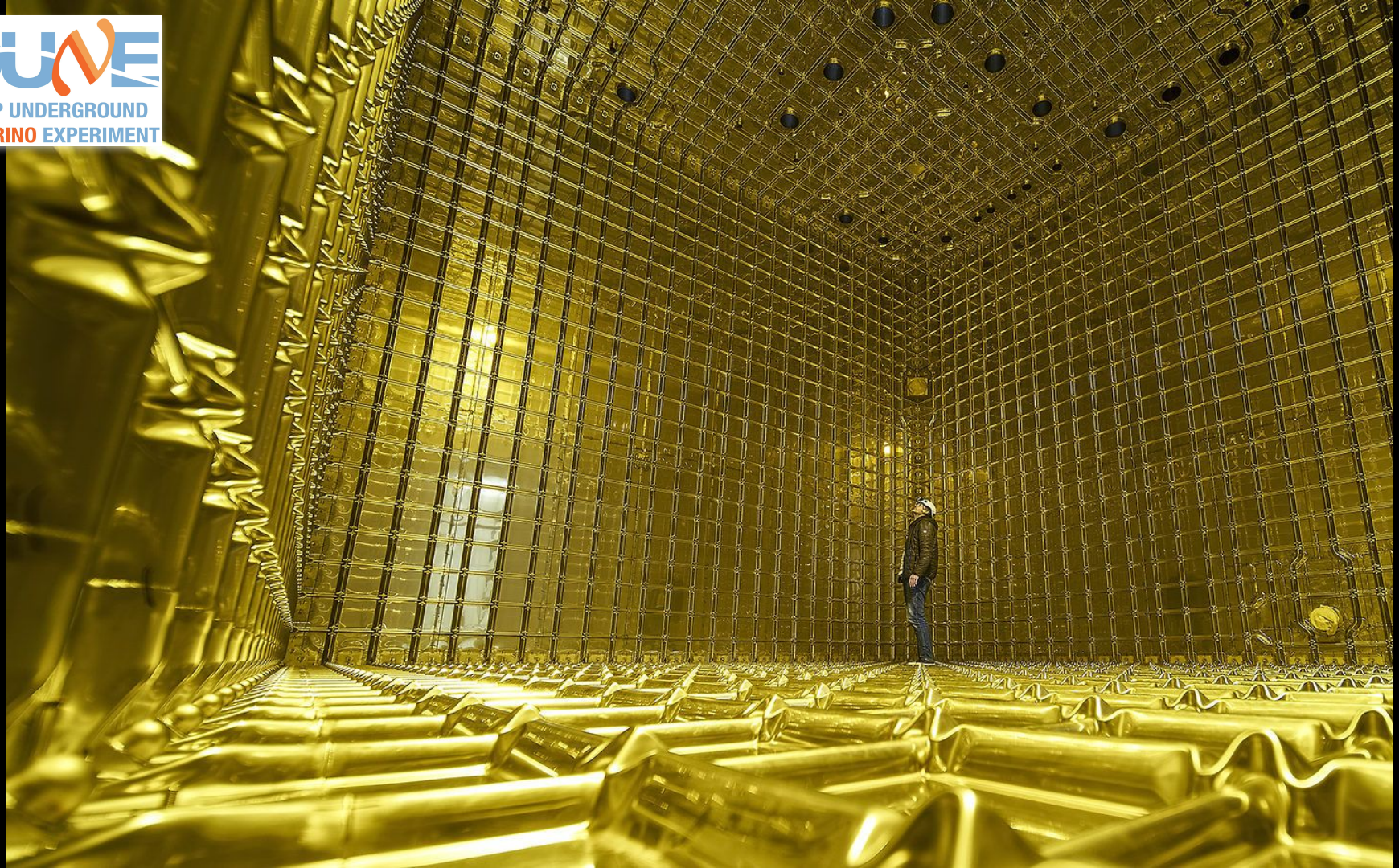
TWGrid ASGC

DiCOS
Distributed Cloud Operating System

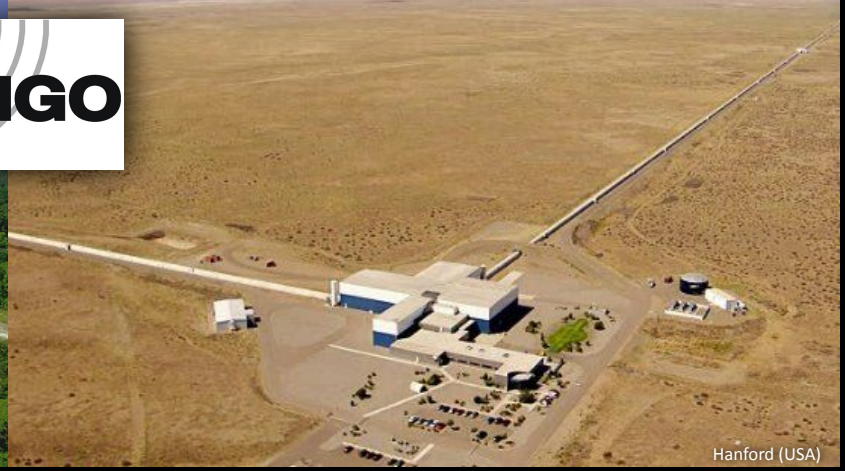

Xe
XENON
Matter Project
PURDUE
UC San Diego
Stockholm University
Princeton

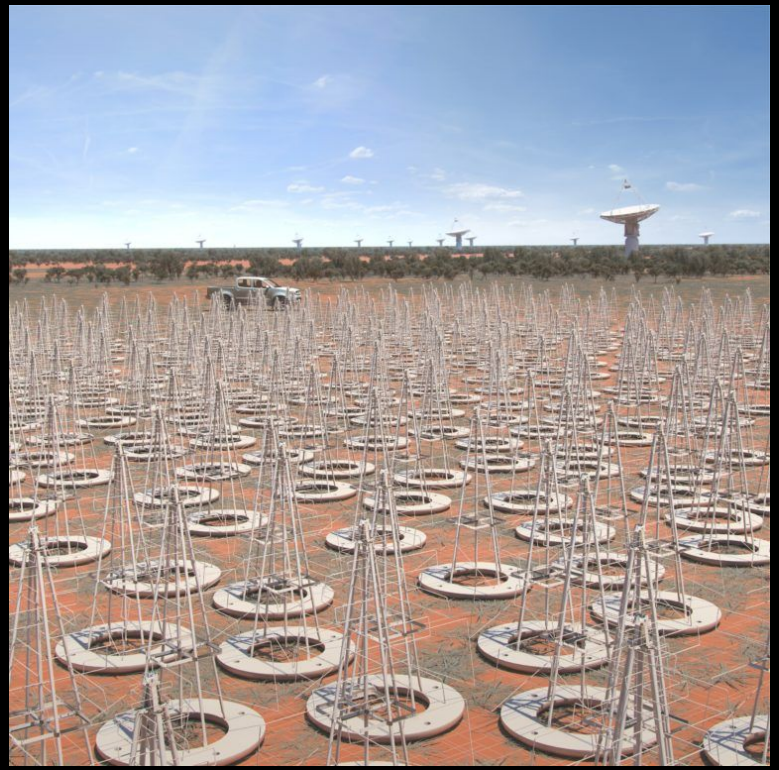





DUNE
DEEP UNDERGROUND
NEUTRINO EXPERIMENT

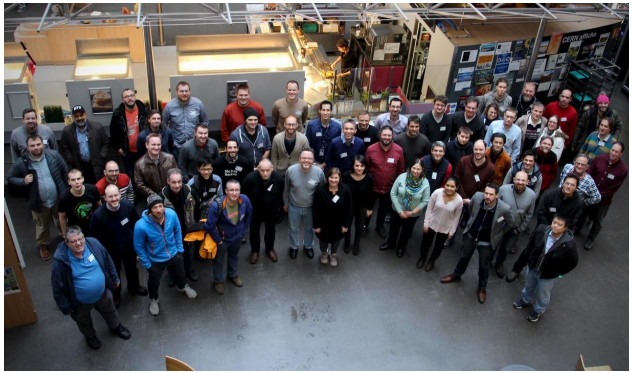
LIGO





Regular events

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



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

Community


eneas
Advanced European Network of E-infrastructures
for Astronomy with the SKA


LCLS


Belle II

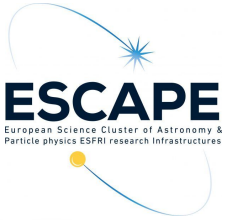

Science & Technology
Facilities Council


COMPASS


CERN

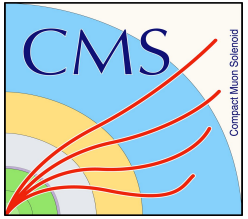

ARC


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

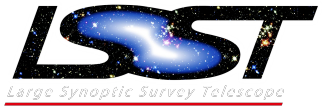

DUNE
DEEP UNDERGROUND
NEUTRINO EXPERIMENT


egi


3D
EISCAT


VIRGO


cta
cherenkov telescope array


LSST
Large Synoptic Survey Telescope


IceCube
SOUTH POLE NEUTRINO OBSERVATORY


FTS


eXtreme DataCloud


Fermilab


BROOKHAVEN
NATIONAL LABORATORY


PAWSEY
supercomputing centre


SKA
SQUARE KILOMETRE ARRAY


LIGO

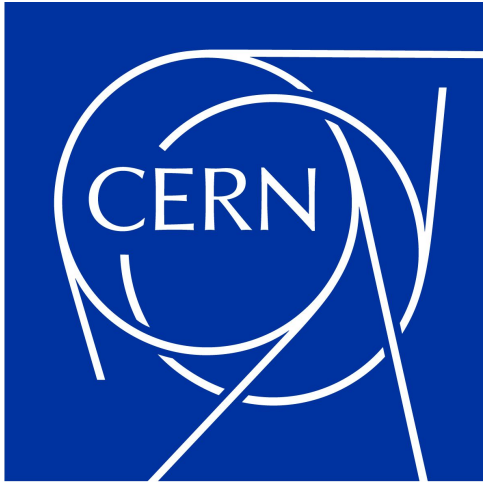

FOLDING
@HOME


Open Science Grid


EUXDAT

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


CERN