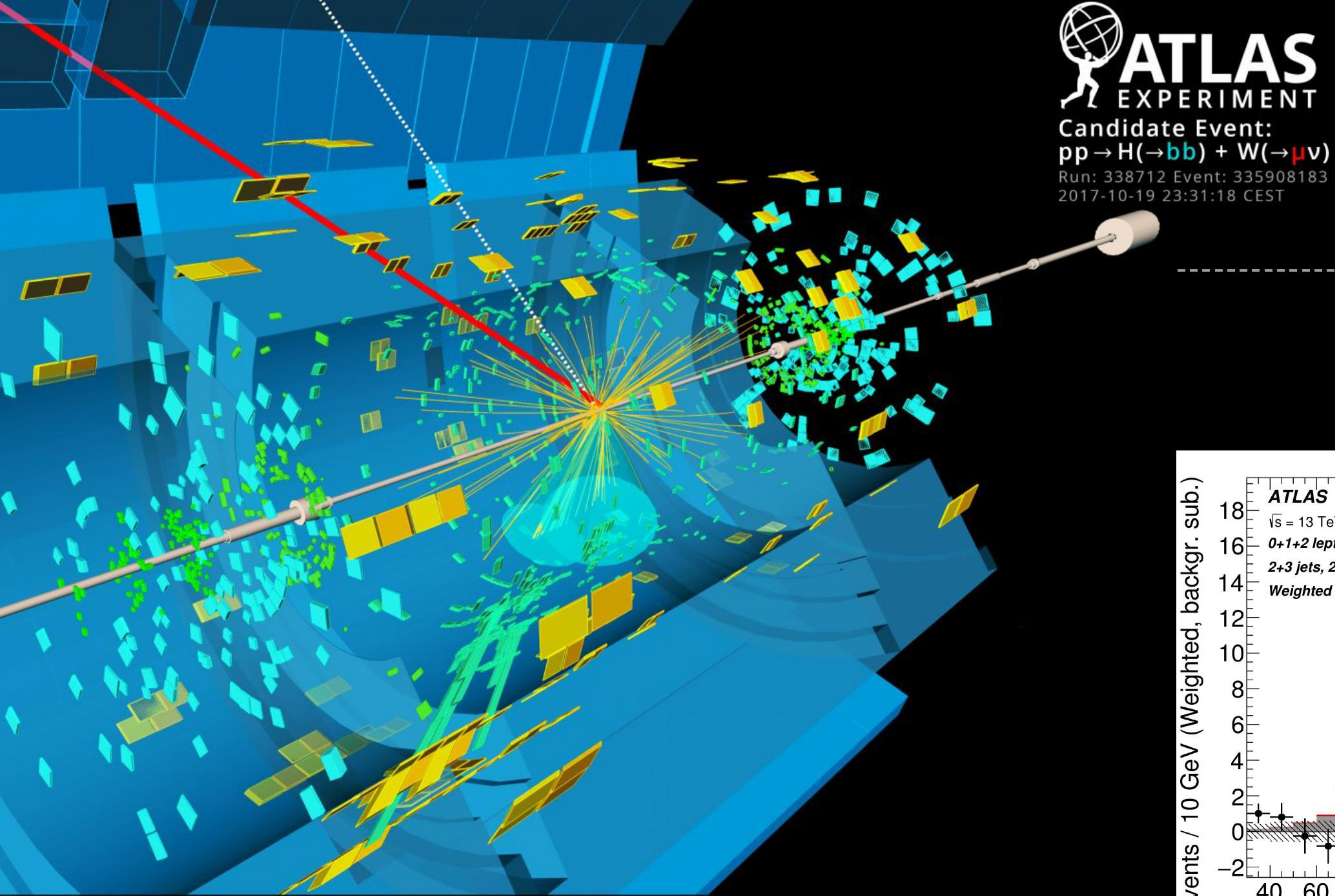
Scientific data management for exascale experiments

Mario.Lassnig@cern.ch
on behalf of the Rucio team

EIROforum Workshop
26 Oct 2020 - 29 Oct 2020
https://indico.cern.ch/event/881752

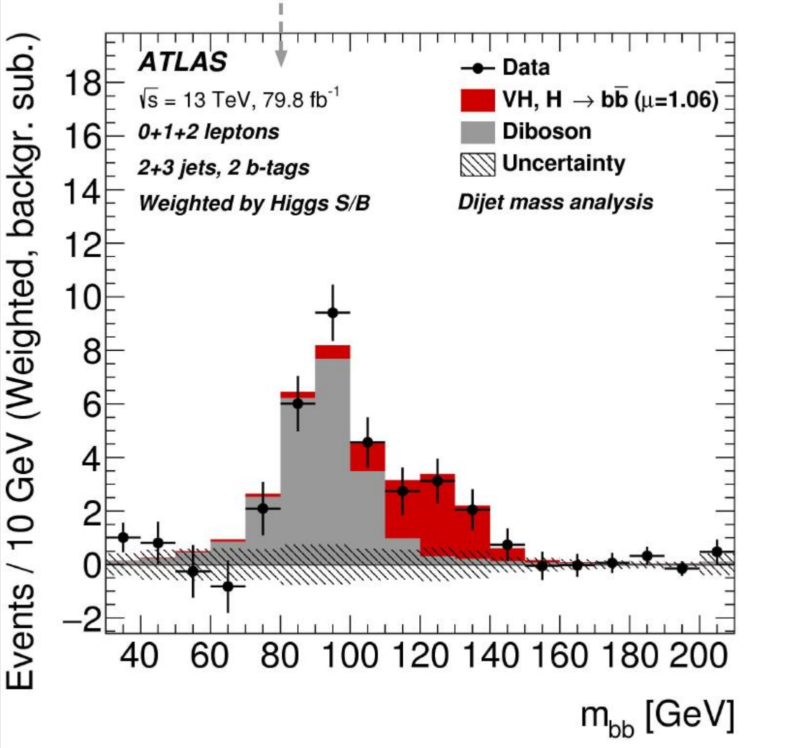


13 TeV detector data

- 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)

System overview

Rucio community experiences

Looking forward

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 source initiative Approved License

Open community-driven development process

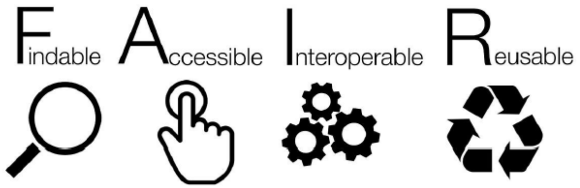
docker

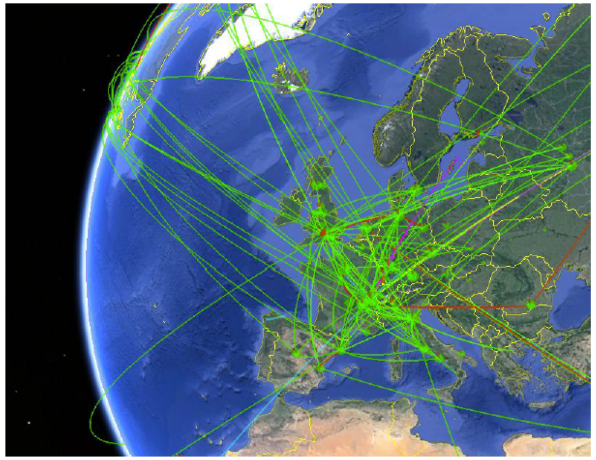
Rucio main functionalities

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

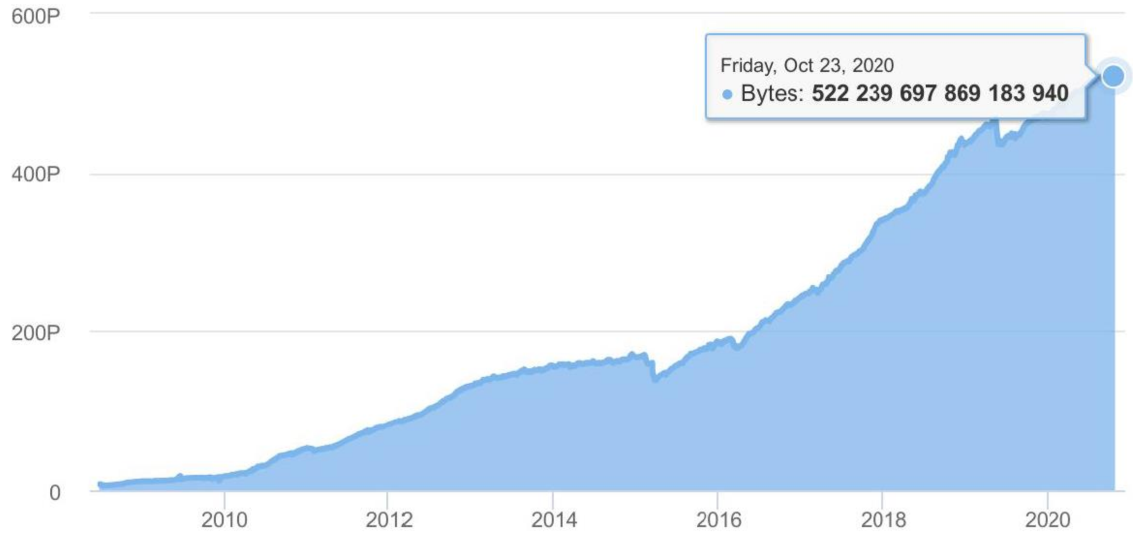
- No Rucio software needs to run at the data centres
- Data centres are free to choose which storage system suits them best

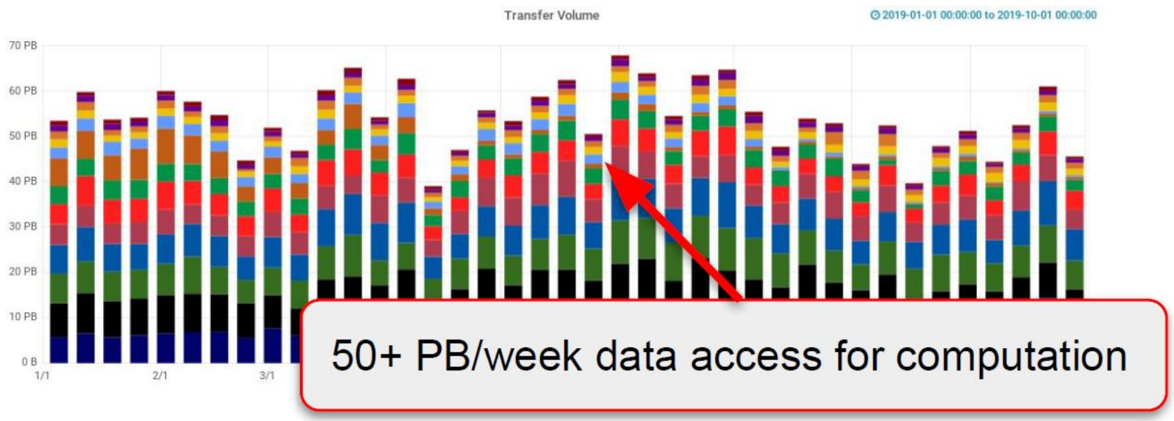
Data management for ATLAS

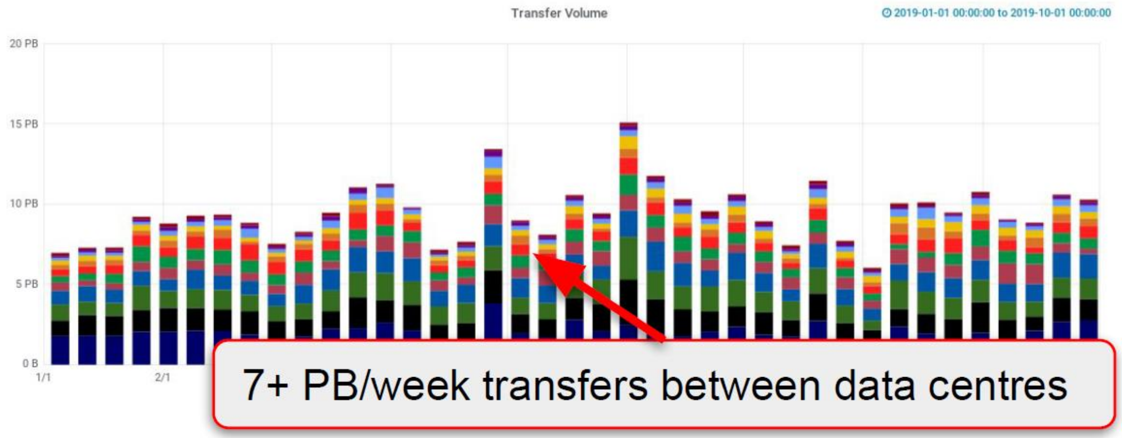
A few numbers to set the scale

- 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

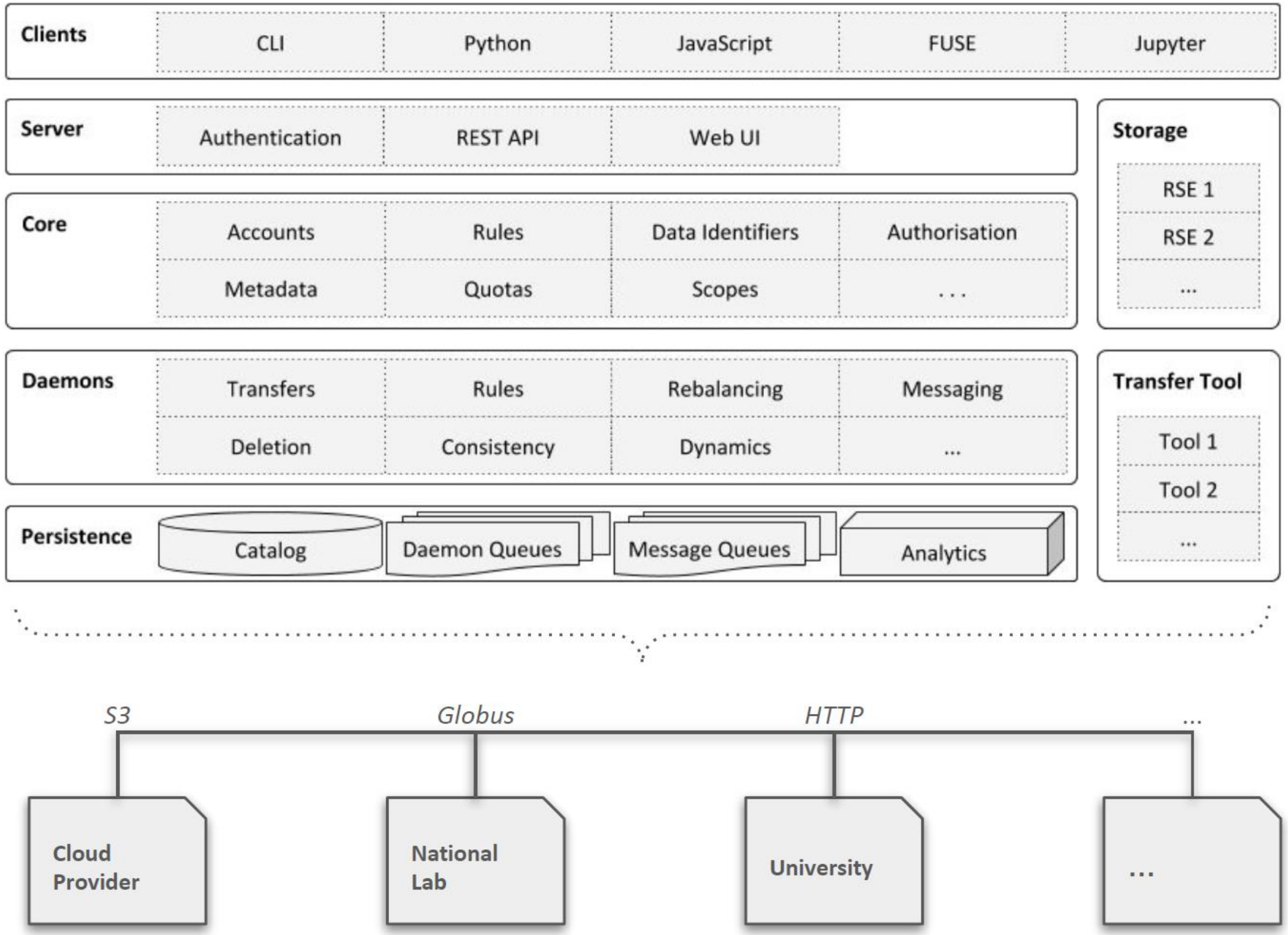
Increase 1+ order of magnitude for HL-LHC







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"*

System overview

Rucio community experiences

Looking forward

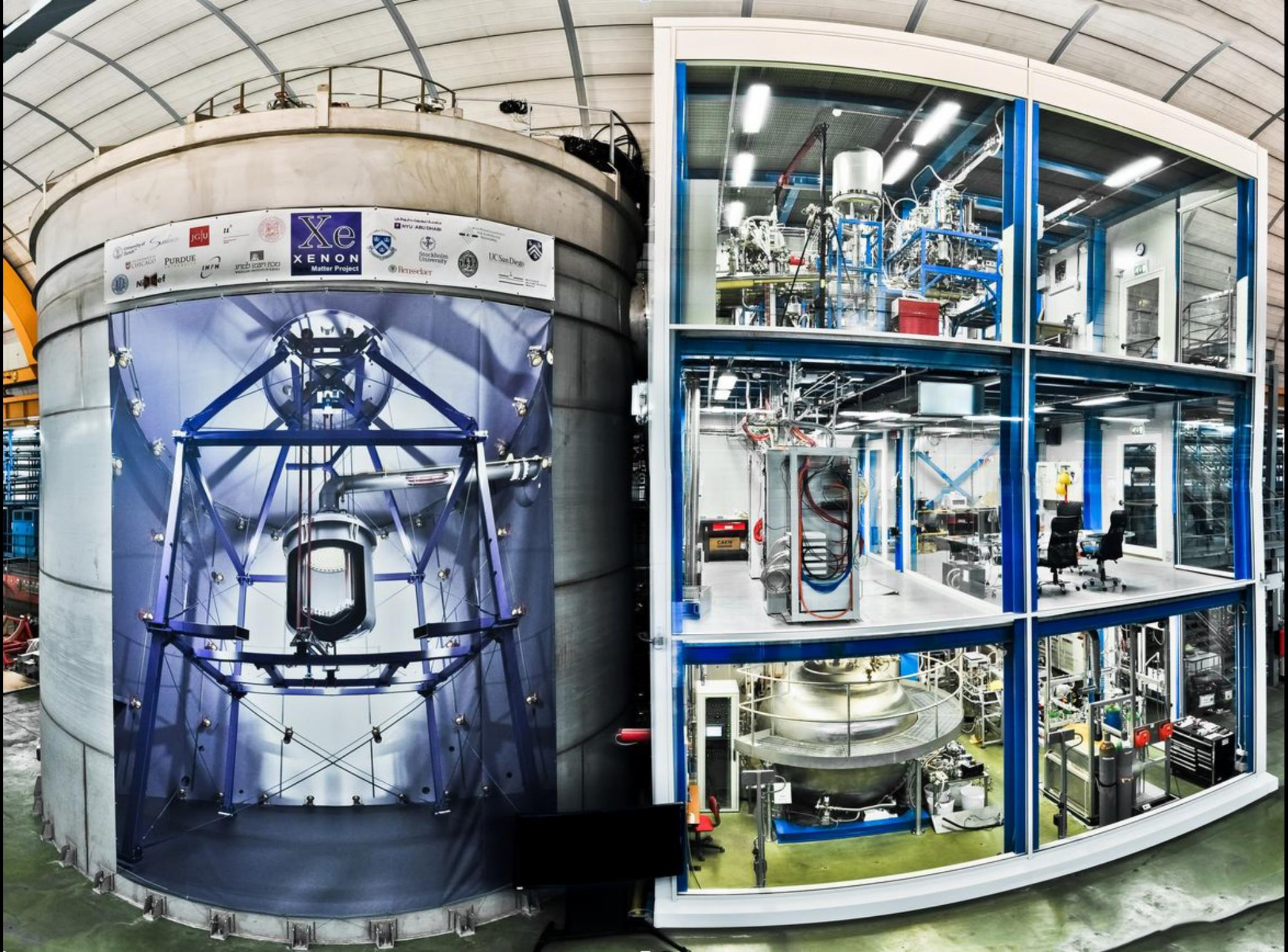
AMS-02

ASGC

DiCOS

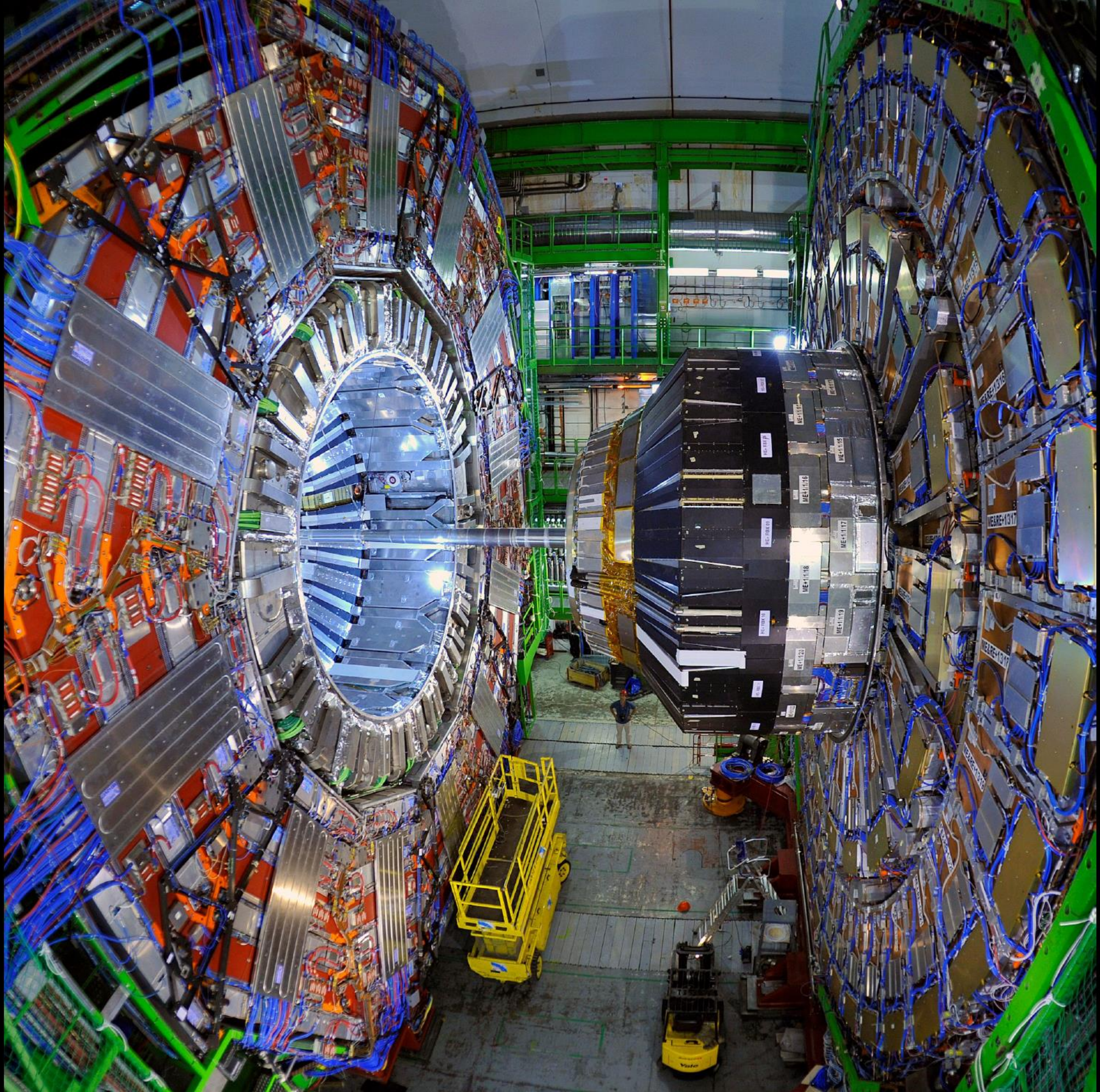
DISTRIBUTED CLOUD OPERATING SYSTEM

Xe
XENON
Dark Matter Project

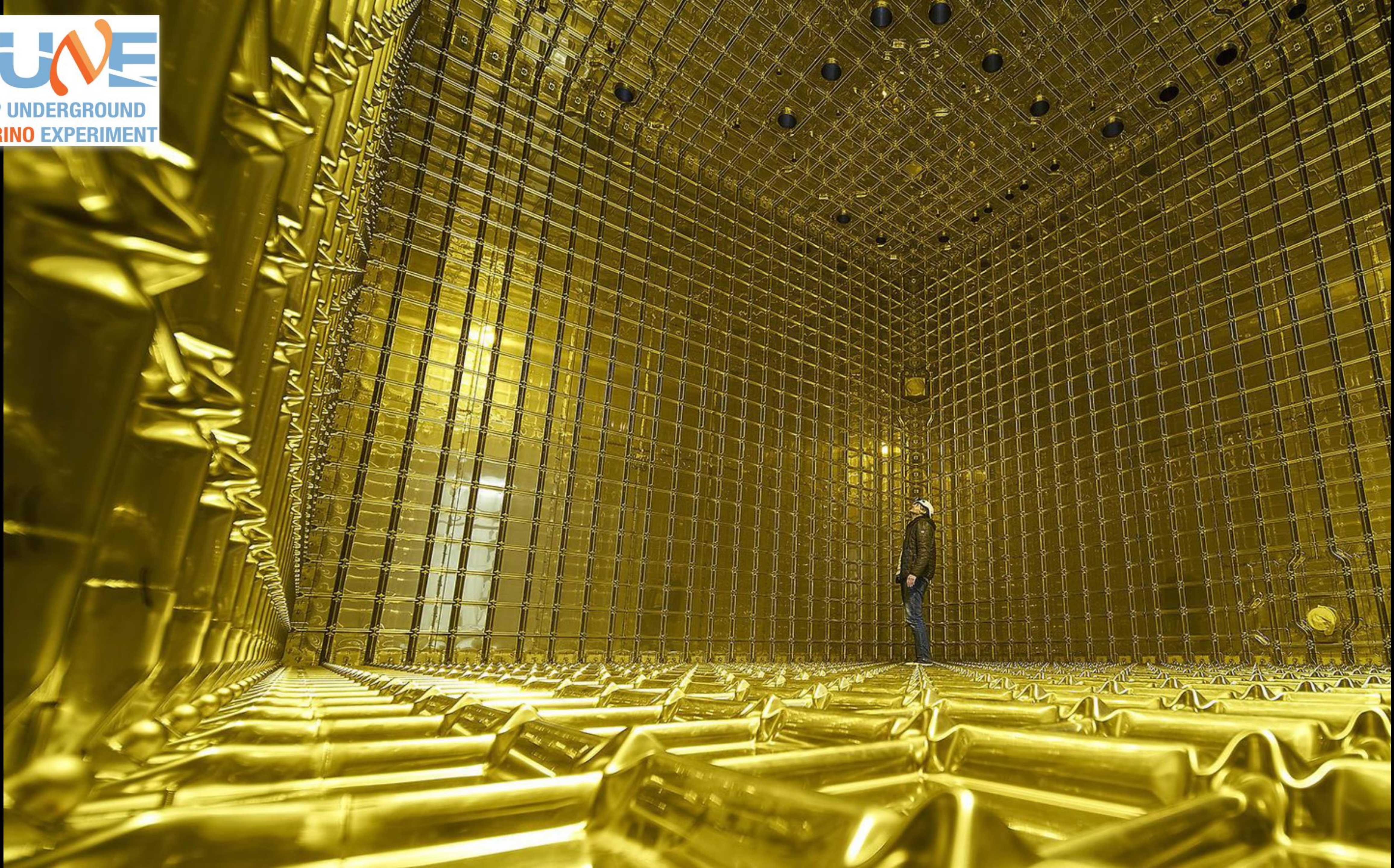


CMS

Compact Muon Solenoid



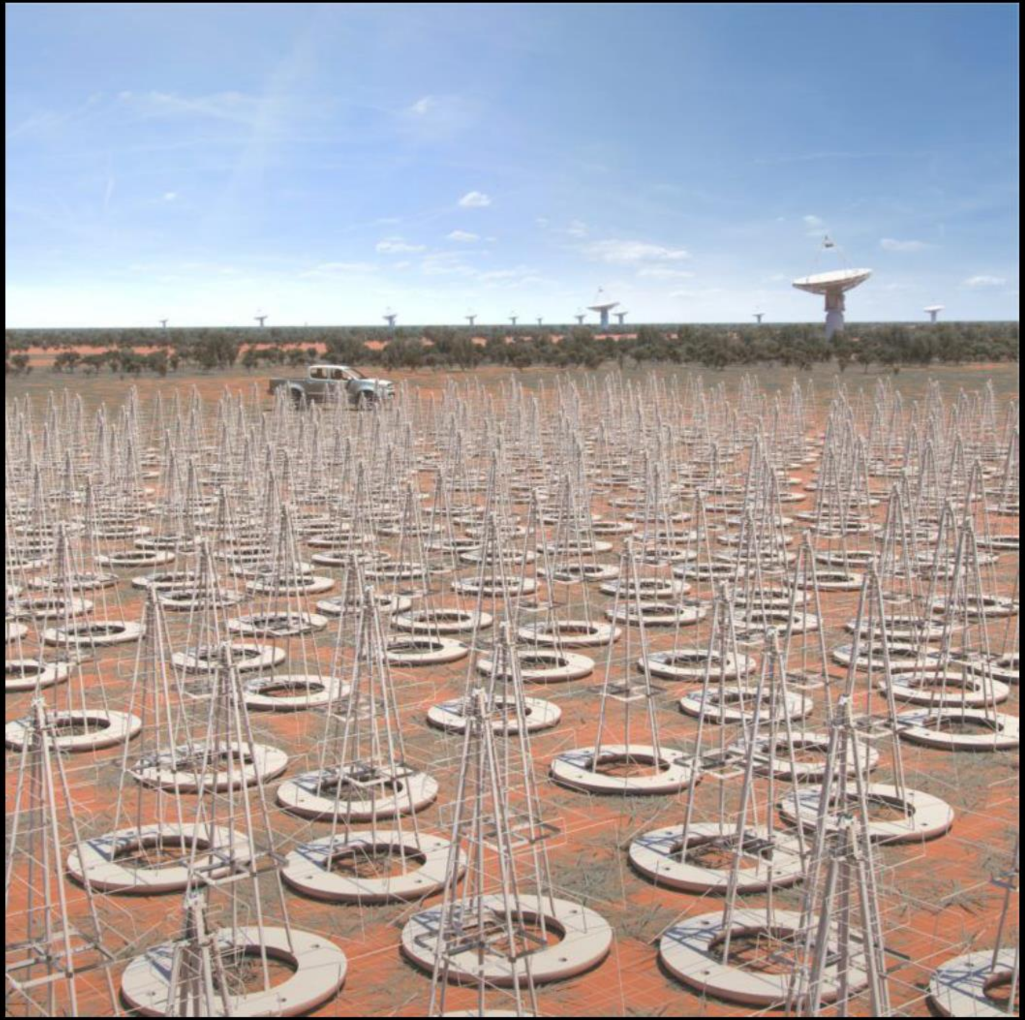
DUNE
DEEP UNDERGROUND
NEUTRINO EXPERIMENT




LIGO
Livingston (USA)


Hanford (USA)


VIRGO
Cascina (IT)



Regular events

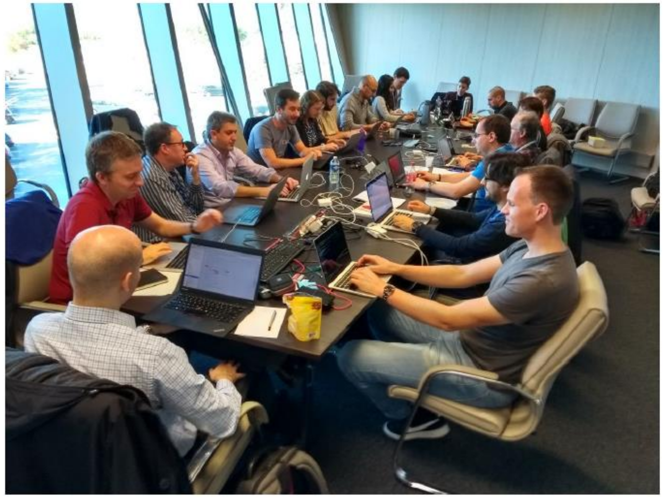
Community Workshops

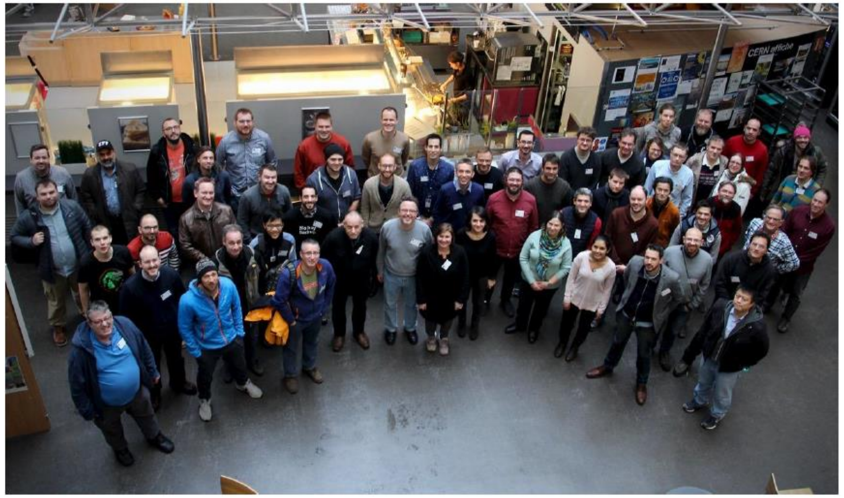
CERN, Switzerland [2018]

University of Oslo, Norway [2019]

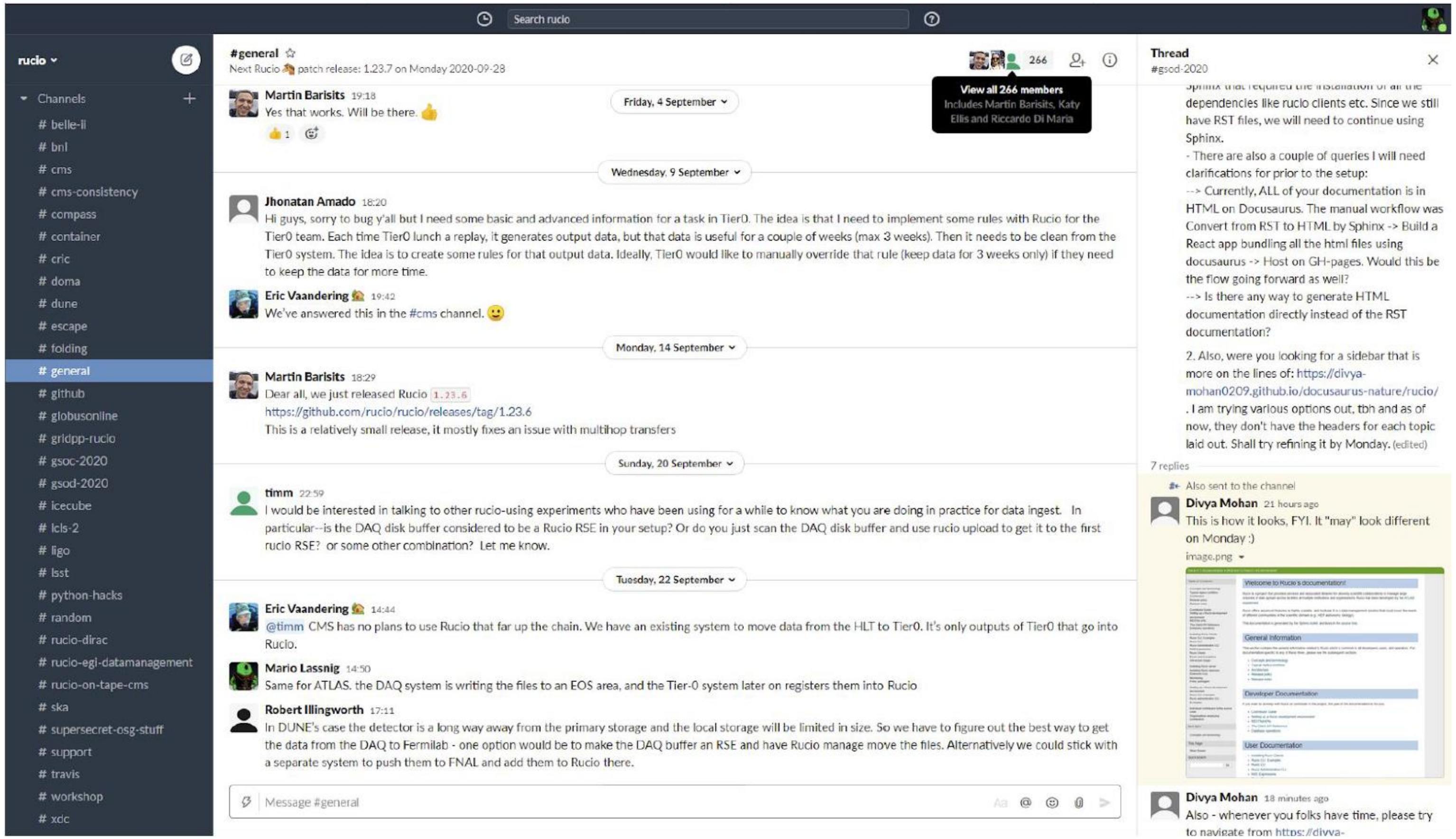
Fermilab, USA [2020]

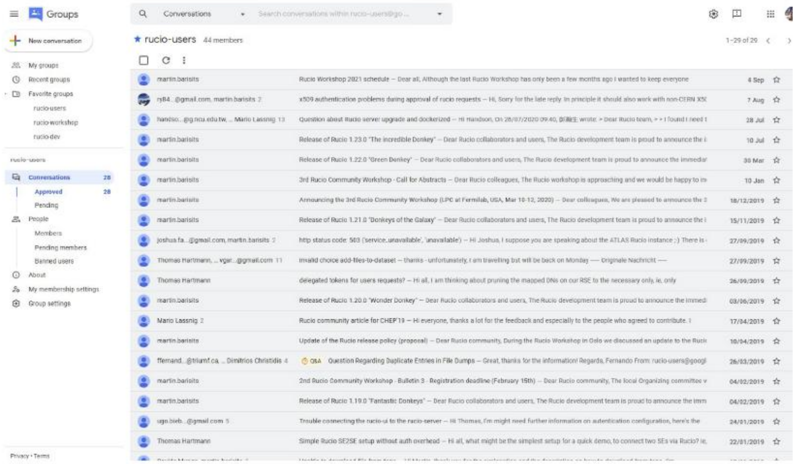
Coding Camps [2018] [2019] [2020]

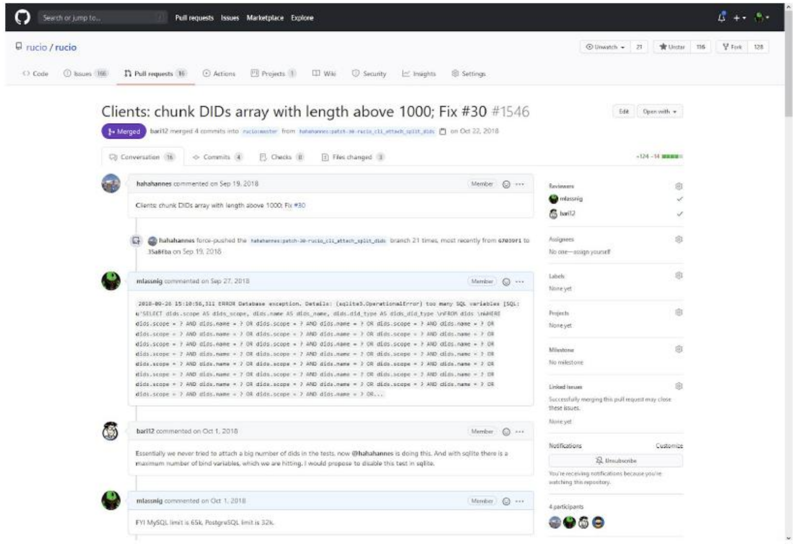




Communication channels







Community-driven development

We have successfully moved to **community-driven development**

- Requirements, features, issues, release are **publicly discussed** (e.g., weekly meetings, GitHub, Slack)
- The core team is usually only **providing guidance** for architecture/design/tests
- Usually 1-2 persons from a **community then take responsibility**
- to **develop** the software extension and also its **continued maintenance**

Communities are helping each other **across experiments**

- Effective across time zones due to **multi-continent** involvement
- Automation and containerisation of development **lowers barrier of entry** for newcomers
- Core team then only takes care about the management and packaging of the releases

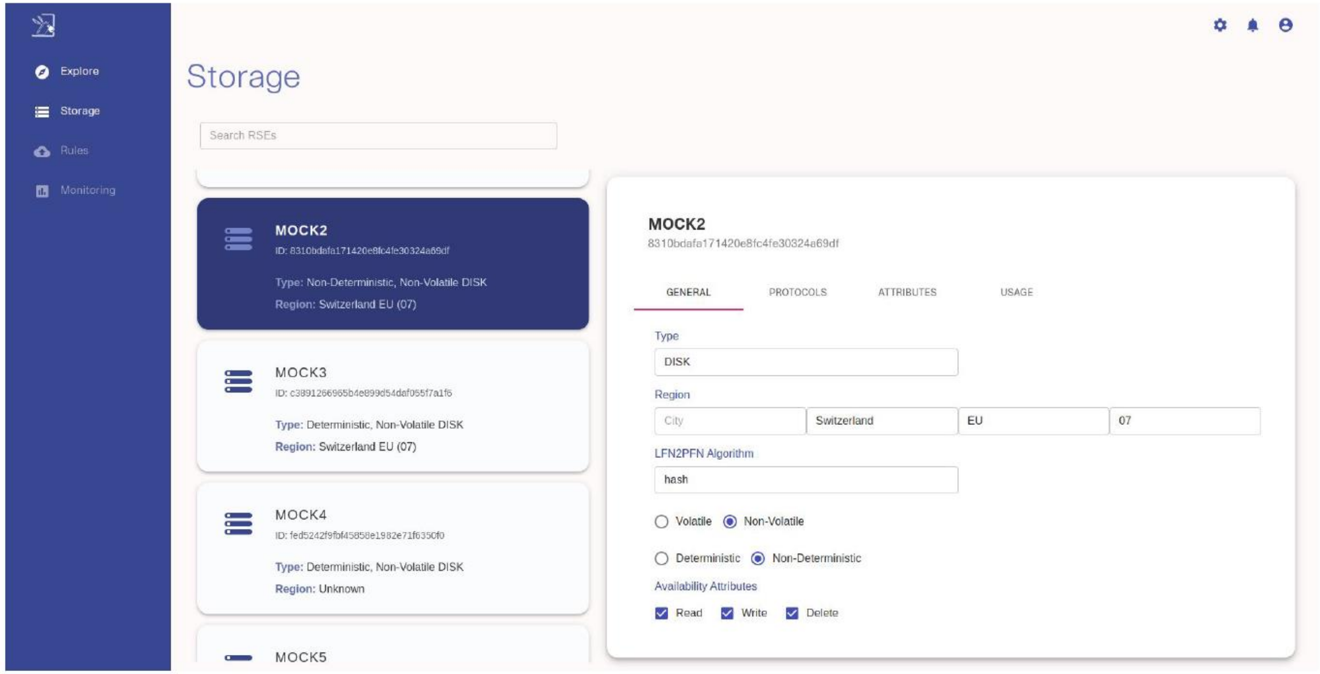
Variety of different topics addressed by **focus groups**

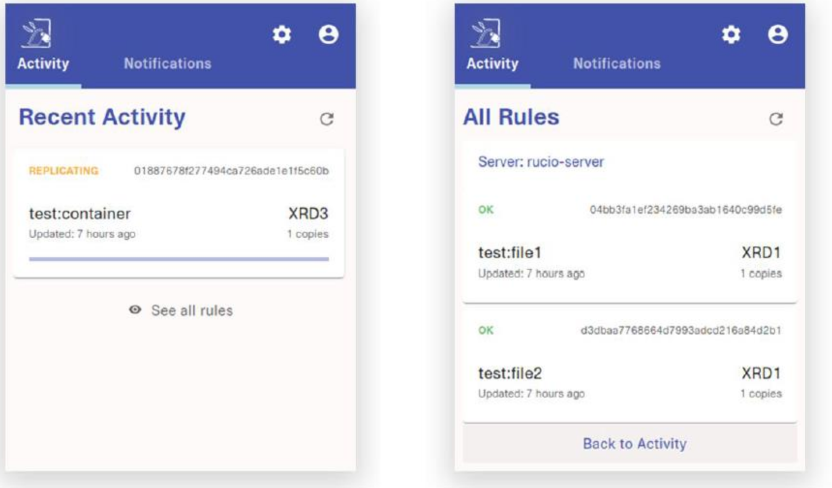
- Third-party-copy, Access and IO, Storage Quality of Service, Token-based Authn/z, SDNs, Cloud integration, ...

Google Summer of Code Sneak Peek

Rucio Desktop

Rucio Notifier





A growing community

System overview

Rucio community experiences

Looking forward

Data management for HL-LHC

HL-LHC will bring an order of magnitude increase in requirements

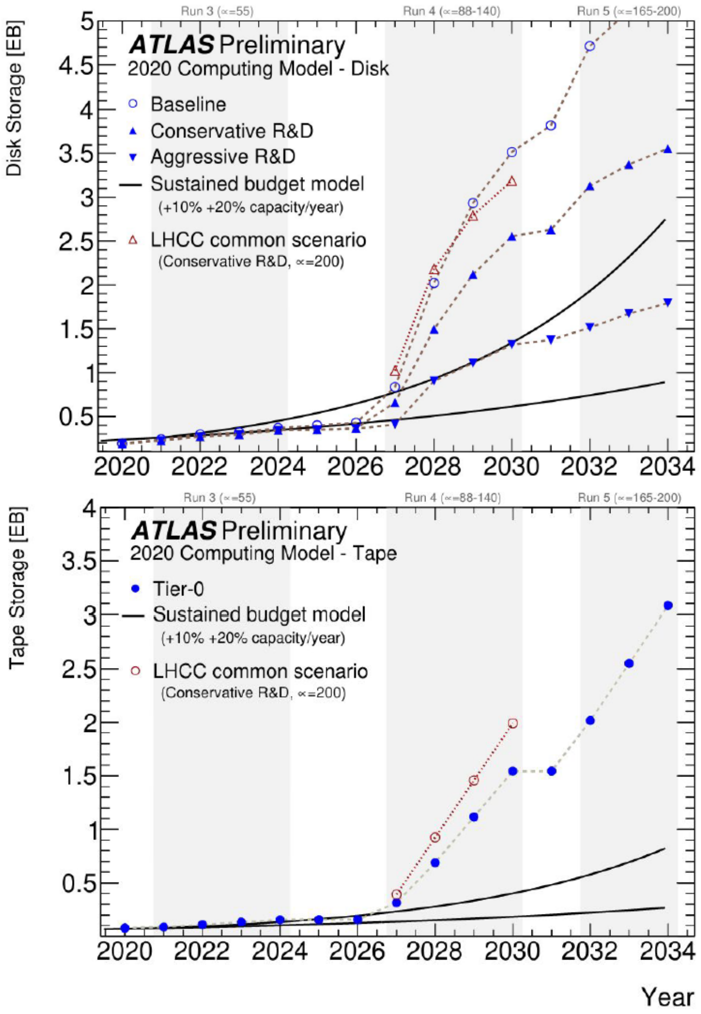
- Resource envelope critical from 2027
- R&D programmes: WLCG/DOMA, H2020 ESCAPE, IRIS-HEP, IRIS,
  and many more national and international initiatives

Long-term data management R&D strategy

- Distributed data centres ("data lakes") with wide-area cache control
- Fine-grained processing of data for accelerated compute and HPCs
- Dynamic storage quality adaptation (QoS for Storage)
- SDNs across multiple NRENs with flow control

Rucio is at centre of these R&D efforts

- Drives the R&D from the experiment's perspective
- Connects the developments from the different working groups
- Implements and evaluates new dataflows, and support software integration



Towards a common data management solution

Shared use of the global research infrastructures will become the norm, especially with sciences at the scale of HL-LHC, DUNE, and SKA

- Competing requests on a **limited set of storage and network**
- Data centres are already supporting **multiple experiments**
- **Compute** seems well-covered — Good scheduling systems, interfaces, and specifications exist
- **Data** was always missing **a common open-source solution** to tackle our **shared challenges**

Ensure more efficient use of available data resources

- **Allocate storage and network based on science needs**, not based on administrative domains
- **Orchestrate dataflow policies across experiments**
- Dynamically support compute workflows with **adaptive data allocations**
- **Unify monitoring**, reporting and analytics to data centres and administration
- Potential for **shared operations across experiments**

Summary

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

- Supporting the world's largest scientific experiments
- 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

Fresh off the press - IEEE Data Engineering article:
http://sites.computer.org/debull/A20mar/A20MAR-CD.pdf

Thank you!

Website	http://rucio.cern.ch
Documentation	https://rucio.readthedocs.io
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

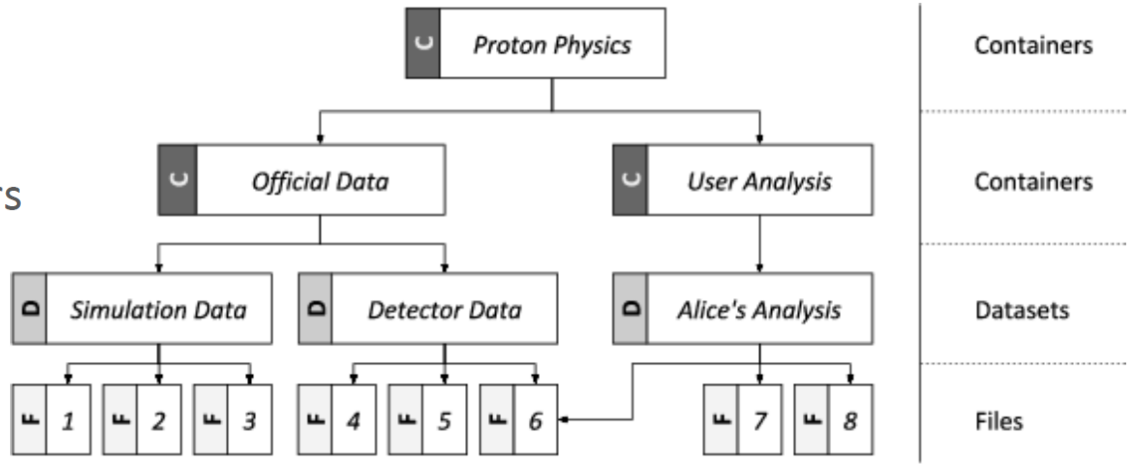
Backup

Namespace

All data stored in Rucio is identified by a **D**ata **ID**entifier (DID)

There are different types of DIDs

Files	
Datasets	Collection of files
Container	Collection of datasets and/or containers



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

`detector_raw.run34:observation_123.root`

scope name

RSEs

Rucio Storage Elements (RSEs) are logical entities of space

- No software needed to run at the facility except the storage system, e.g., EOS/dCache/S3, …
- RSE names are arbitrary, e.g., "CERN-PROD_DATADISK", "AWS_REGION_USEAST", …
- Common approach is one RSE per storage class at the site

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 different protocol for LAN-only access

RSEs can be assigned metadata as well

- Key/Value pairs, e.g., *country=UK, type=TAPE, is_cached=False, …*
- You can use RSE expressions to describe a list of RSEs, e.g. `country=FR&type=DISK`, for the rules

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, …*
- Internal metadata necessary for the organisation of data, e.g., *replication factor, job-id,*
- Generic metadata that can be set by the users

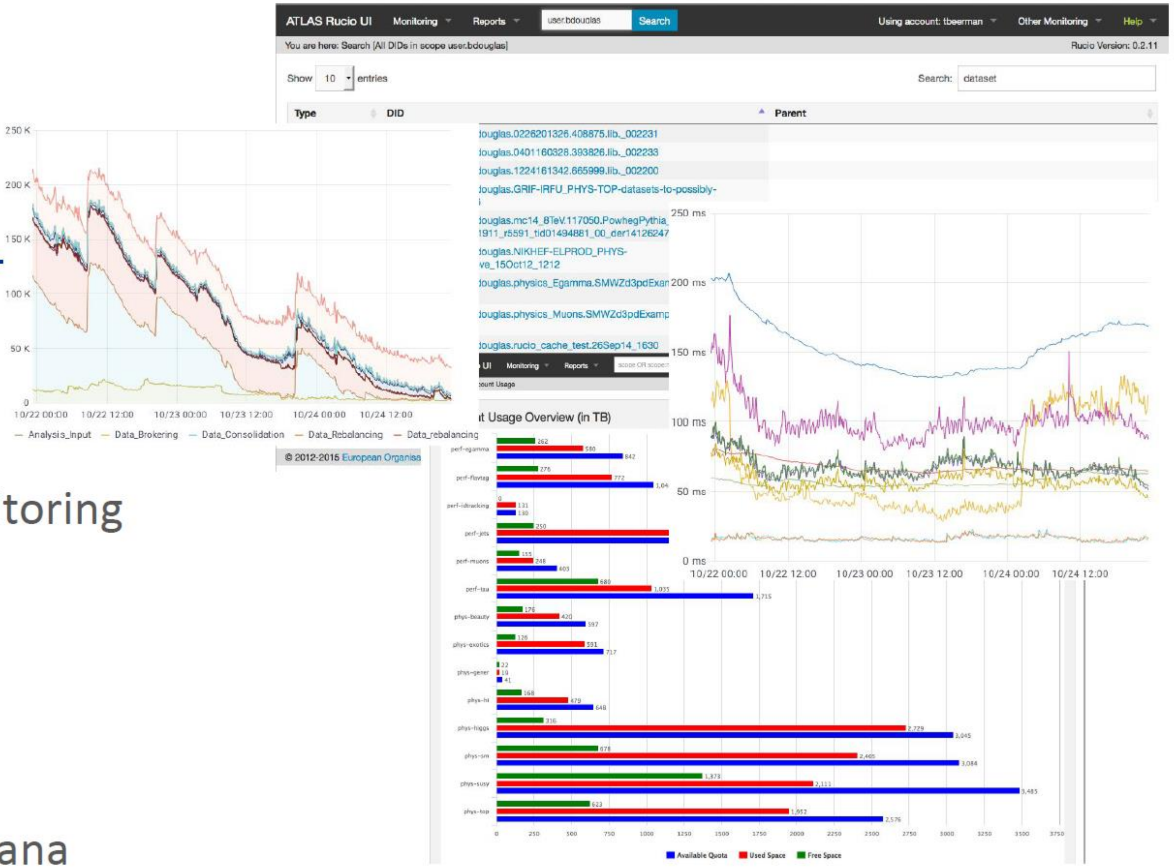
Generic metadata can be restricted

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

Provides additional namespace to organise the data

- Searchable via name and metadata
- Aggregation based on metadata searches
- Can also be used for long-term reporting, e.g., evolution of particular metadata selection over time

Monitoring & analytics



Rucio Web-UI

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

Detailed monitoring

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

Analytics and accounting

- Data aggregation for long-term reporting and decision-making
- Built on Hadoop and Spark