



Scientific data management for exascale experiments

Mario.Lassnig@cern.ch

on behalf of the Rucio team

EIROforum Workshop

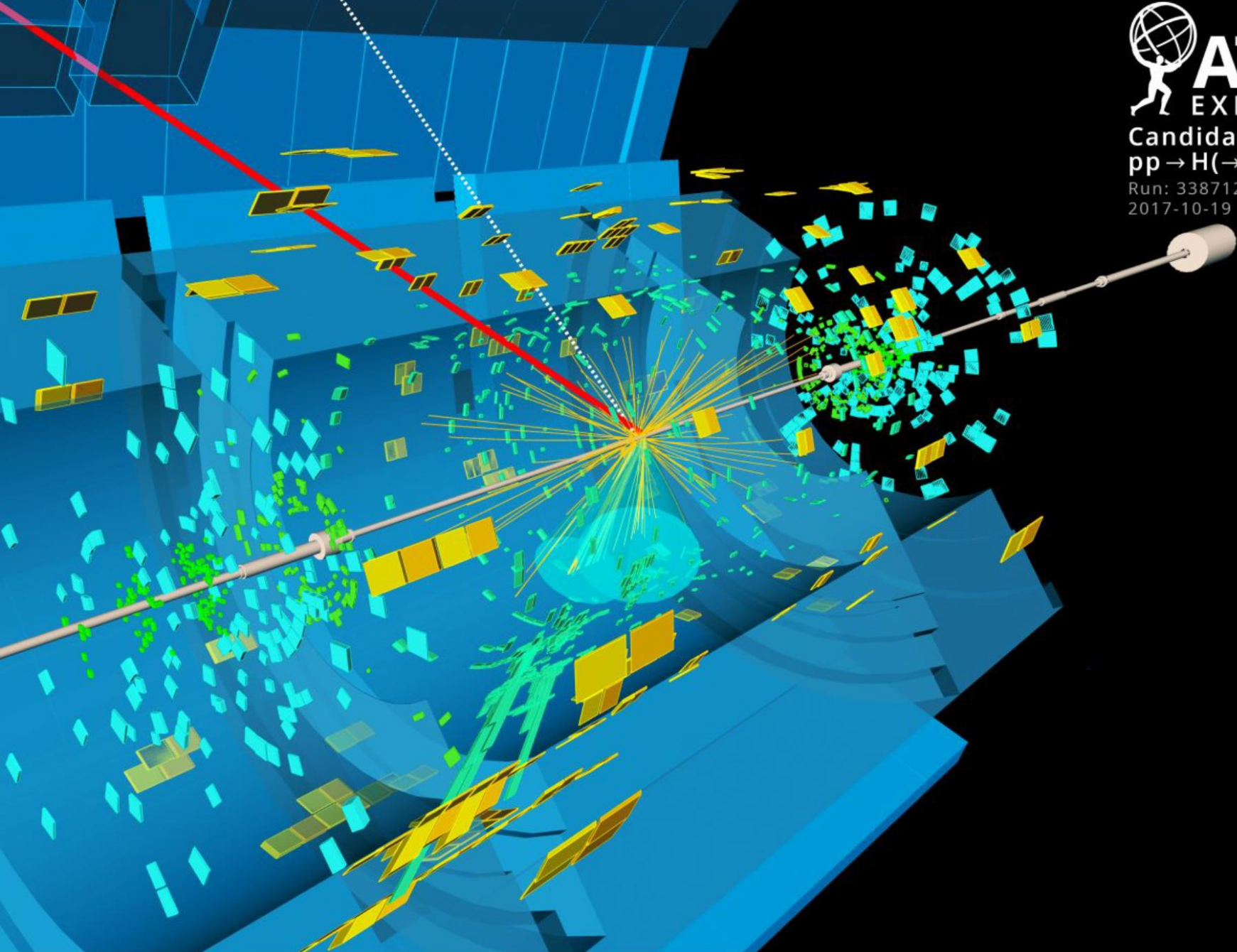
26 Oct 2020 - 29 Oct 2020

<https://indico.cern.ch/event/881752>



Candidate Event:
 $pp \rightarrow H(\rightarrow bb) + W(\rightarrow \mu\nu)$

Run: 338712 Event: 335908183
 2017-10-19 23:31:18 CEST

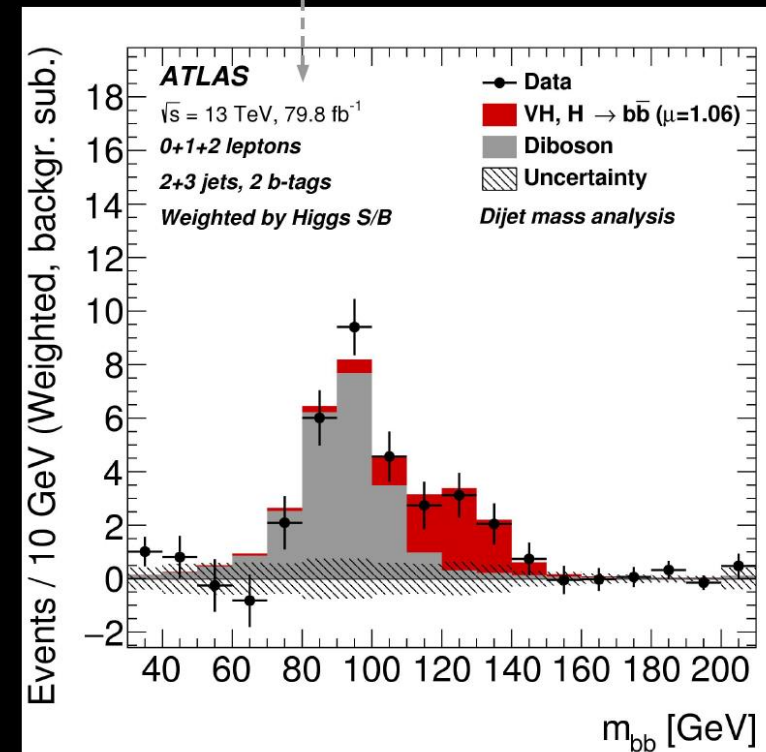


13 TeV detector data

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)

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 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

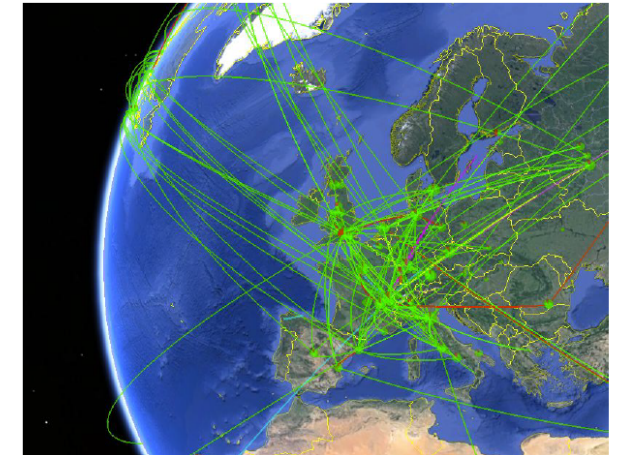
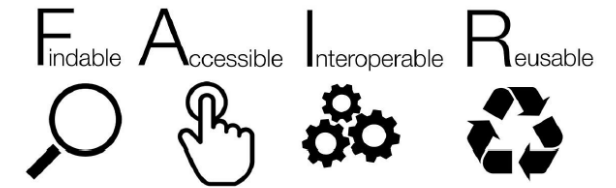
Extensive monitoring for all dataflows

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

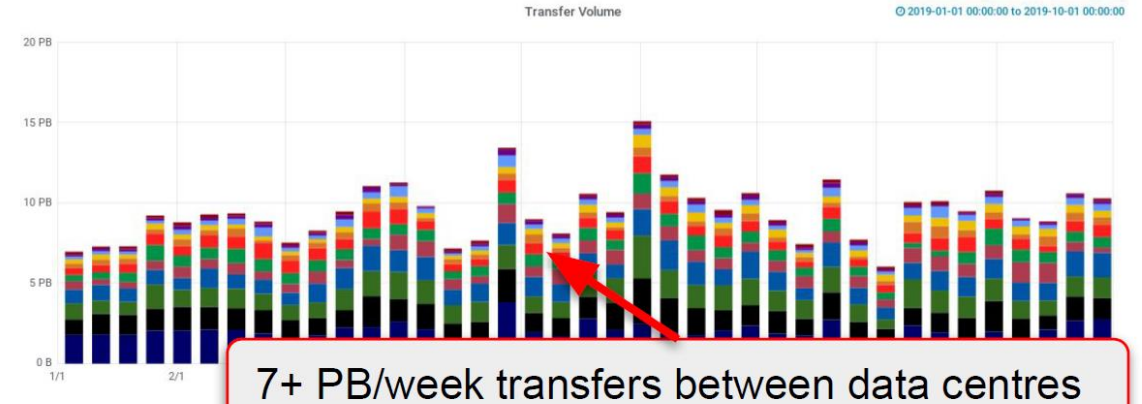
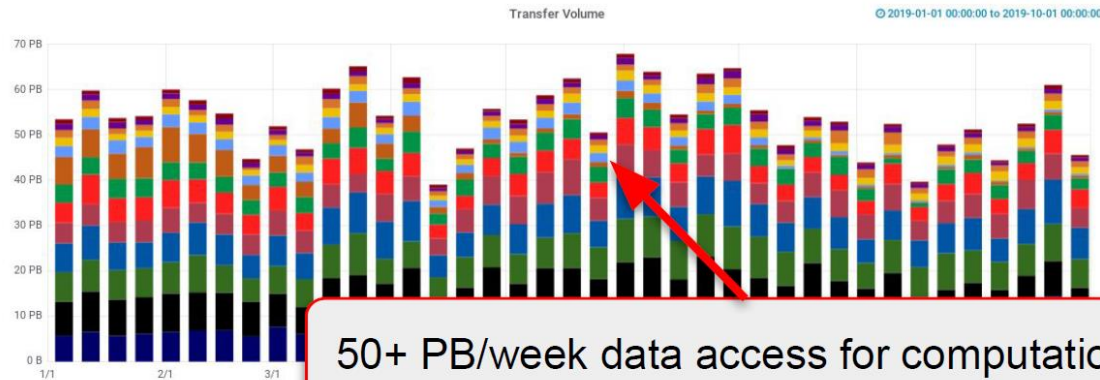
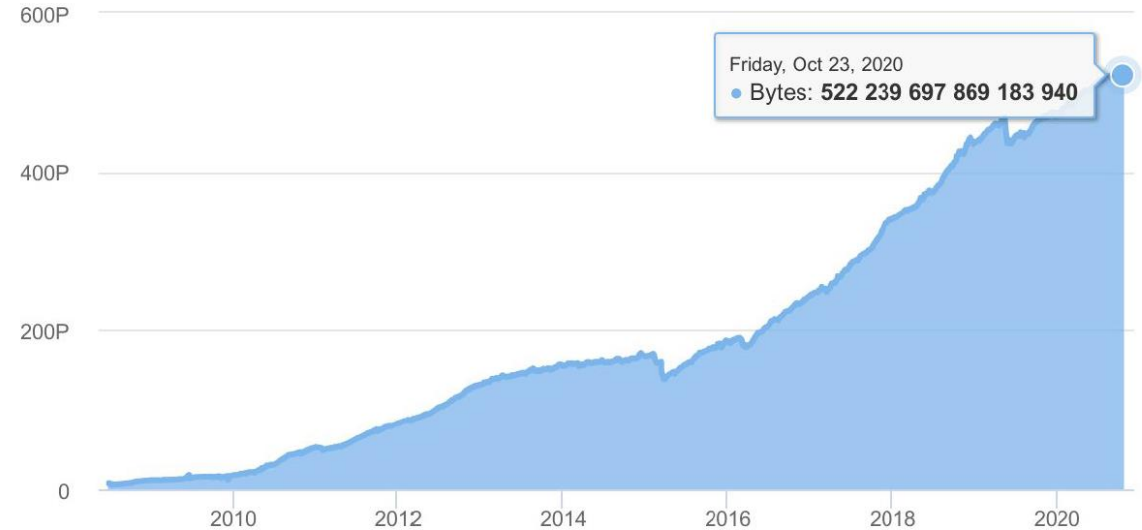
Data management for ATLAS



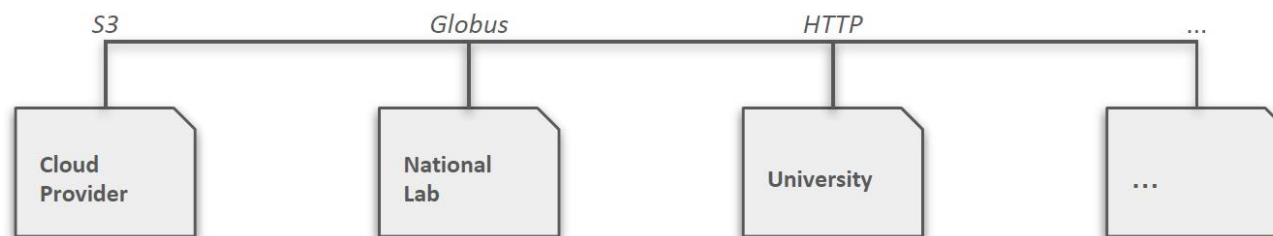
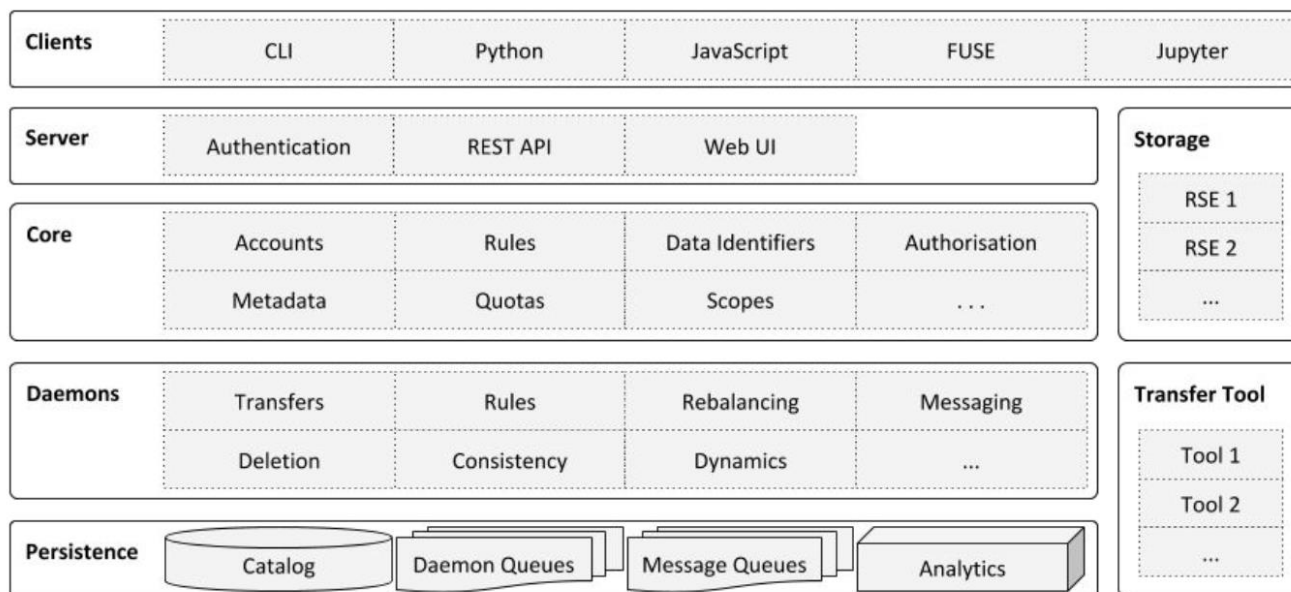
A few numbers to set the scale

- 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**

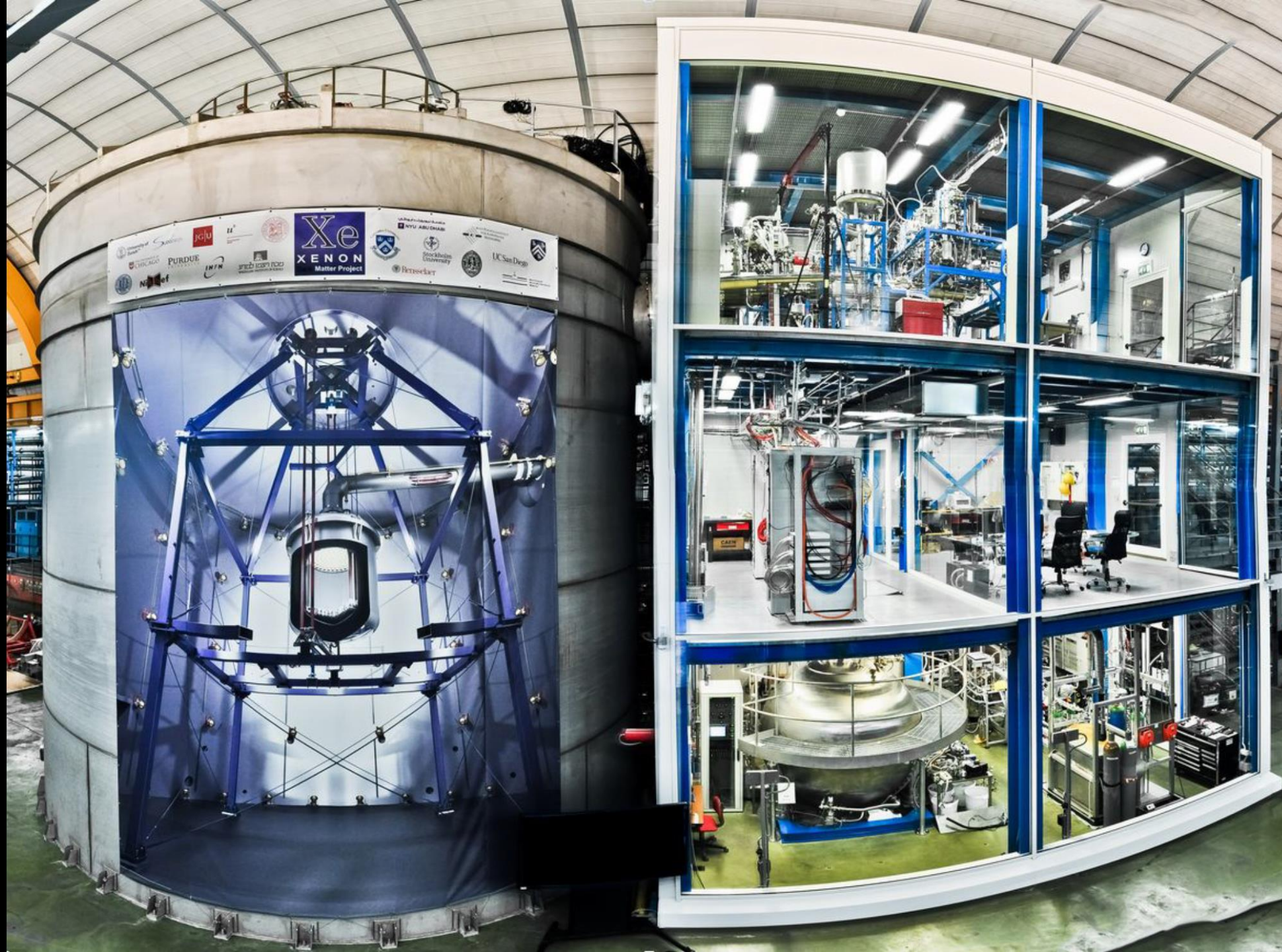
e.g., *"All derived products from this physics channel must have a copy on TAPE"*

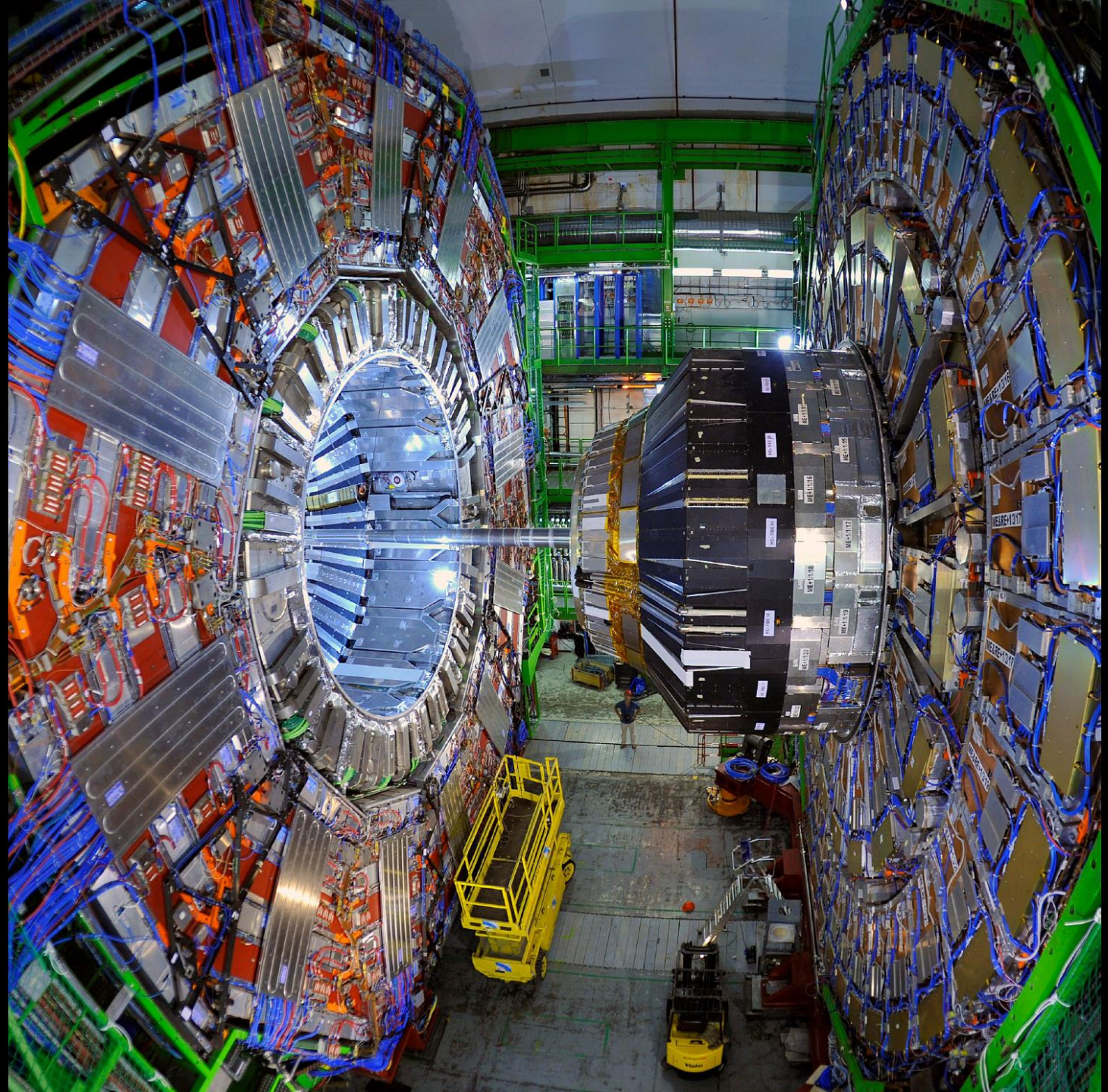
System overview

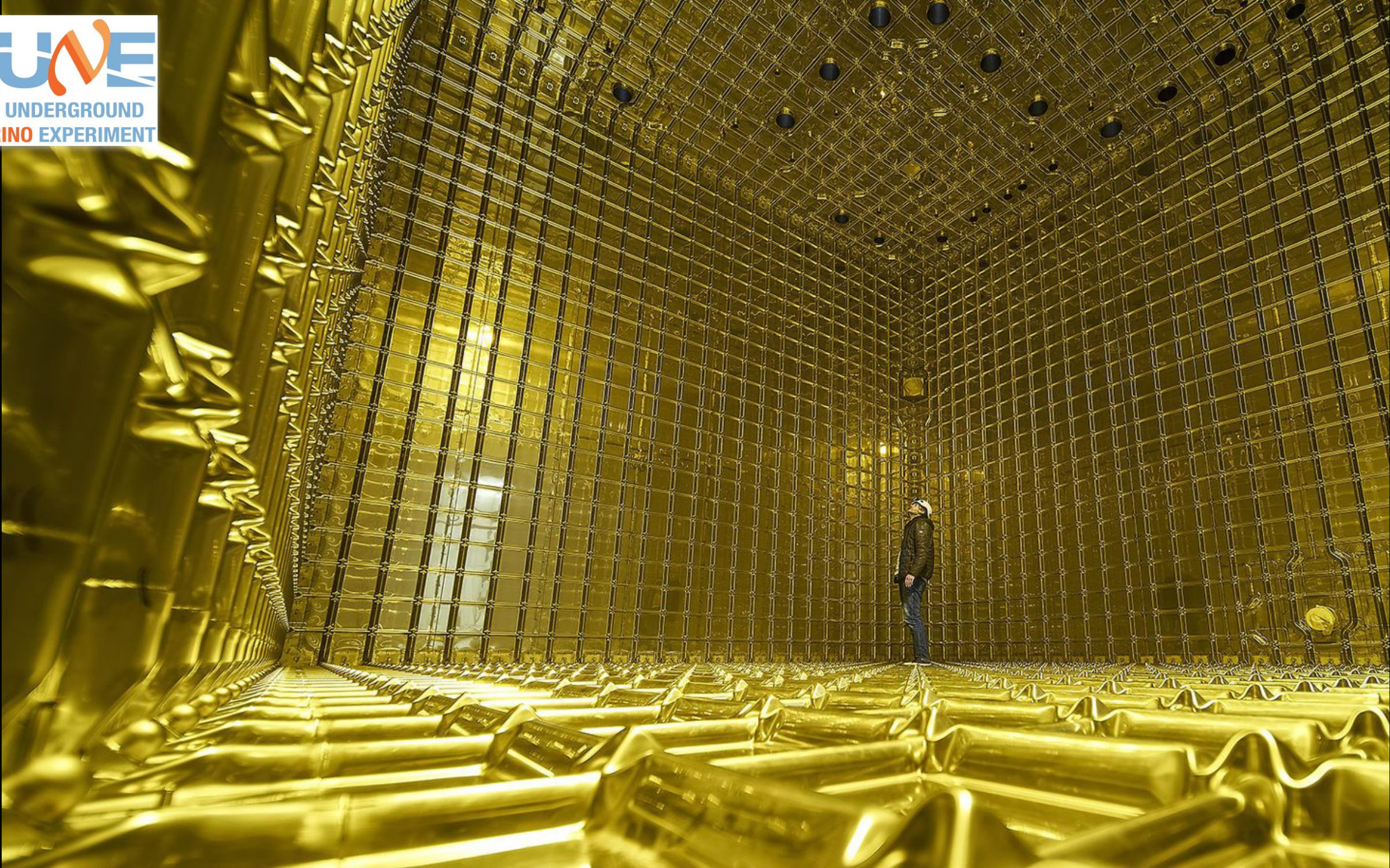
Rucio community experiences

Looking forward











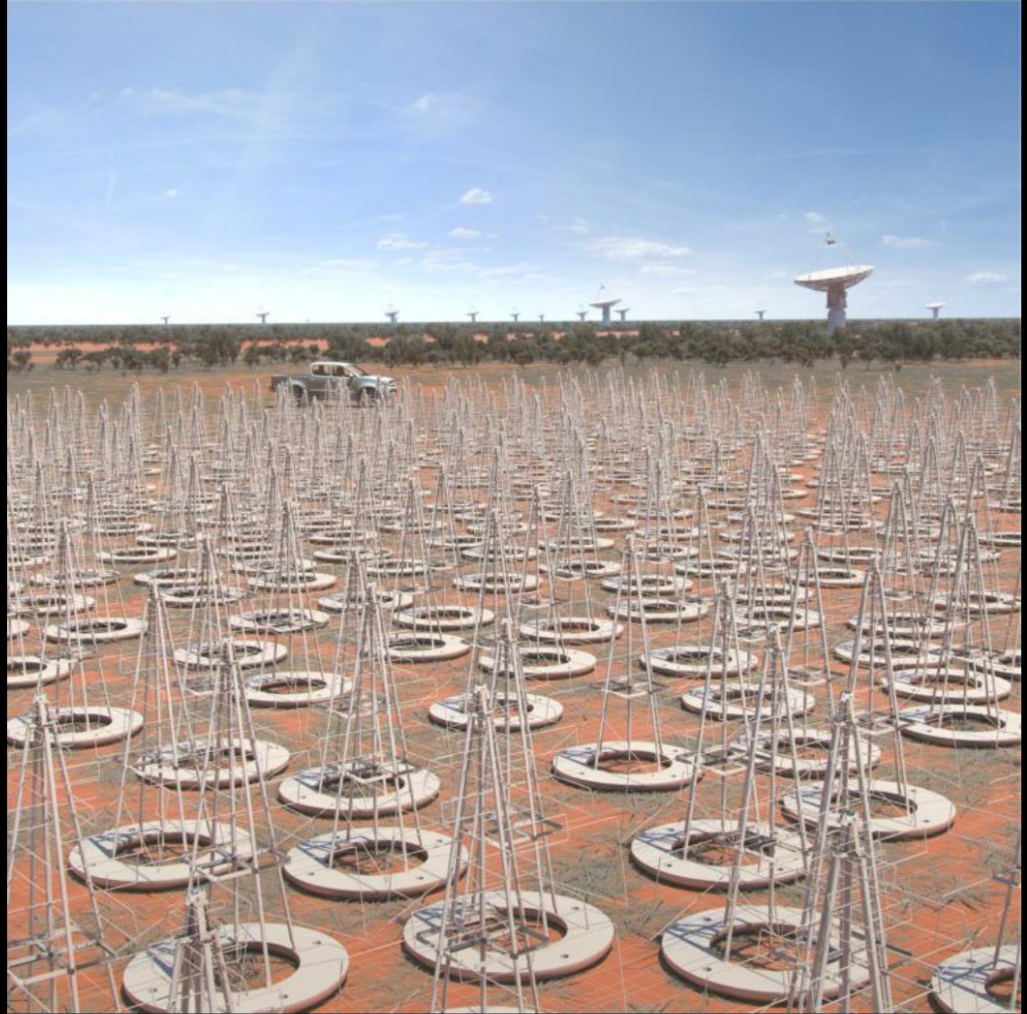
Livingston (USA)



Hanford (USA)



Cascina (IT)



Regular events



Community Workshops

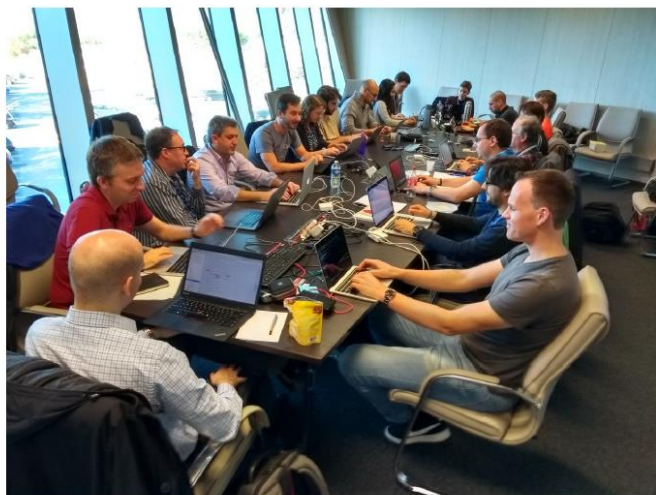
CERN, Switzerland [\[2018\]](#)

University of Oslo, Norway [\[2019\]](#)

Fermilab, USA [\[2020\]](#)



Coding Camps [\[2018\]](#) [\[2019\]](#) [\[2020\]](#)



Communication channels



#general 264 members

Next Rucio patch release: 1.23.7 on Monday 2020-09-28

Martin Barisits 19:18
Yes that works. Will be there. 🙌

Jhonatan Amado 18:20
Hi guys, sorry to bug y'all but I need some basic and advanced information for a task in Tier0. The idea is that I need to implement some rules with Rucio for the Tier0 team. Each time Tier0 lunch a replay, it generates output data, but that data is useful for a couple of weeks (max 3 weeks). Then it needs to be clean from the Tier0 system. The idea is to create some rules for that output data. Ideally, Tier0 would like to manually override that rule (keep data for 3 weeks only) if they need to keep the data for more time.

Eric Vaandering 19:42
We've answered this in the #cms channel. 😊

Martin Barisits 18:29
Dear all, we just released Rucio 1.23.6
<https://github.com/rucio/rucio/releases/tag/1.23.6>
This is a relatively small release, it mostly fixes an issue with multihop transfers

tim 22:59
I would be interested in talking to other rucio-using experiments who have been using for a while to know what you are doing in practice for data ingest. In particular—is the DAQ disk buffer considered to be a Rucio RSE in your setup? Or do you just scan the DAQ disk buffer and use rucio upload to get it to the first rucio RSE? or some other combination? Let me know.

Eric Vaandering 14:44
@tim CMS has no plans to use Rucio that far up the chain. We have an existing system to move data from the HLT to Tier0. It's only outputs of Tier0 that go into Rucio.

Mario Lassnig 14:50
Same for ATLAS, the DAQ system is writing the files to an EOS area, and the Tier-0 system later on registers them into Rucio

Robert Illingworth 17:11
In DUNE's case the detector is a long way away from the primary storage and the local storage will be limited in size. So we have to figure out the best way to get the data from the DAQ to Fermilab - one option would be to make the DAQ buffer an RSE and have Rucio manage move the files. Alternatively we could stick with a separate system to push them to FNAL and add them to Rucio there.

Issues

| Issue # | Title | Assignee | Status |
|---------|---|------------|--------|
| 489 | Rucio Workshop 2021 schedule | Open at | Open |
| 490 | x509 authentication problems during approval of rucio requests | H. Jansons | Open |
| 491 | Question about Rucio server upgrade and disklayout | H. Jansons | Open |
| 492 | Release of Rucio 1.23.0 'The Invisible Donkey' | Open | Open |
| 493 | Release of Rucio 1.22.0 'Green Donkey' | Open | Open |
| 494 | 3rd Rucio Community Workshop - Call for Abstracts | Open | Open |
| 495 | Announcing the 3rd Rucio Community Workshop (LPC at Fermilab, USA, Mar 10-13, 2020) | Open | Open |
| 496 | Release of Rucio 1.21.0 'Donkeys of the Galaxy' | Open | Open |
| 497 | http status code 503 (Service unavailable, 'unavailable') | H. Jansons | Open |
| 498 | email choice add files-to-dataset | Open | Open |
| 499 | delegated tokens for users requests | Open | Open |
| 500 | Release of Rucio 1.20.0 'Wonder Donkey' | Open | Open |
| 501 | Rucio community article for CHEP13 | Open | Open |
| 502 | Update of the Rucio release policy (proposal) | Open | Open |
| 503 | Question Regarding Duplicate Entries in File Dumps | Open | Open |
| 504 | 2nd Rucio Community Workshop - Bulletin 5 - Registration deadline (February 19th) | Open | Open |
| 505 | Release of Rucio 1.19.0 'Tantalus Donkey' | Open | Open |
| 506 | trouble connecting the rucio to the rucio-server | Open | Open |
| 507 | Simple Rucio S3FS setup without auth overhead | Open | Open |

pull request #1546

Clients: chunk DIDs array with length above 1000; Fix #30 #1546

hahubian commented on Sep 19, 2018

hahubian pushed the hahubian:patch-in-rucio-1.23.6 branch 21 times, most recently from exstors to hahubian on Sep 19, 2018

relating commented on Sep 27, 2018

hahubian commented on Oct 1, 2018

relating commented on Oct 1, 2018

hahubian commented on Oct 1, 2018

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**



GitHub



docker



kubernetes

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



The screenshot shows the 'Storage' page in Rucio Desktop. On the left is a navigation sidebar with 'Explore', 'Storage', 'Rules', and 'Monitoring'. The main content area has a search bar for RSEs and a list of storage configurations. The 'MOCK2' configuration is selected, showing its details in a modal window.

Storage

Search RSEs

- MOCK2**
ID: 8310bdafa171420e8fc4fe30324e69df
Type: Non-Deterministic, Non-Volatile DISK
Region: Switzerland EU (07)
- MOCK3**
ID: c389128695b4e899e54def9557a1f6
Type: Deterministic, Non-Volatile DISK
Region: Switzerland EU (07)
- MOCK4**
ID: fed52429f643959e1992e71f6350f0
Type: Deterministic, Non-Volatile DISK
Region: Unknown
- MOCK5**

MOCK2
8310bdafa171420e8fc4fe30324e69df

GENERAL | PROTOCOLS | ATTRIBUTES | USAGE

Type:

Region:

LFN2PFN Algorithm:

Volatile Non-Volatile

Deterministic Non-Deterministic

Availability Attributes: Read Write Delete

The screenshot shows the 'Recent Activity' page in Rucio Notifier. It displays a table of activity with columns for status, ID, and name. A 'test:container' activity is shown as 'REPLICATING' with 'XRD3' and '1 copies'. A 'See all rules' link is at the bottom.

Recent Activity

| Status | ID | Name | Details |
|-------------|----------------------------------|----------------|------------------|
| REPLICATING | 01887678f277494ca726ade1e1f5c60b | test:container | XRD3 1 copies |

[See all rules](#)

The screenshot shows the 'All Rules' page in Rucio Notifier. It lists rules for the 'rucio-server' with columns for status, ID, name, and details. Two rules, 'test:file1' and 'test:file2', are shown as 'OK' with 'XRD1' and '1 copies'. A 'Back to Activity' link is at the bottom.

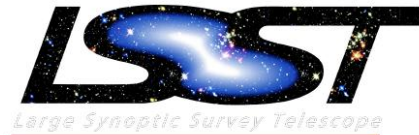
All Rules

Server: rucio-server

| Status | ID | Name | Details |
|--------|----------------------------------|------------|------------------|
| OK | 04bb3fa1ef234269b3ab1640c99d5fe | test:file1 | XRD1 1 copies |
| OK | d3bba07768664d7993adcd216a84d2b1 | test:file2 | XRD1 1 copies |

[Back to Activity](#)

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)

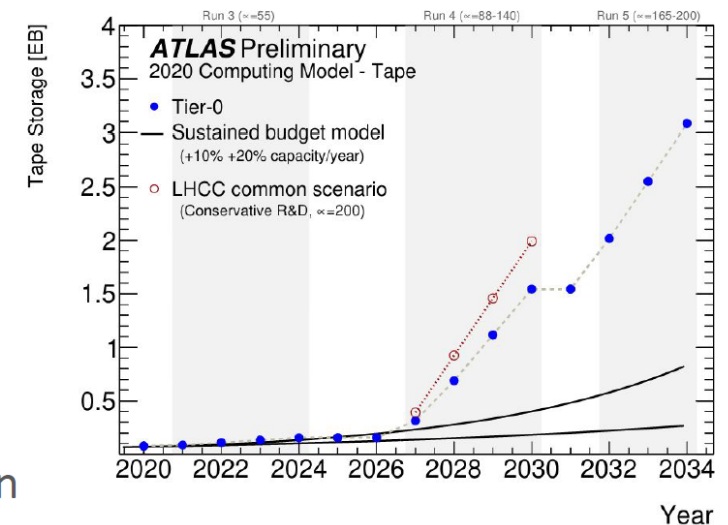
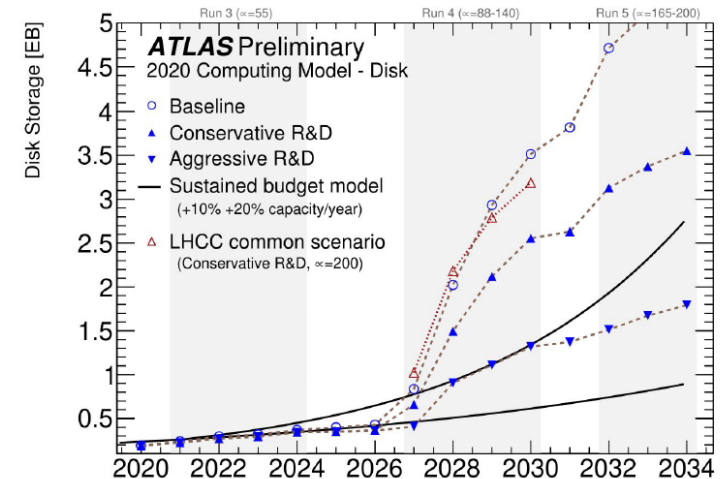
SDNs across multiple NRENs with flow control

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 **Data Identifier (DID)**

There are different types of DIDs

Files

Datasets Collection of files

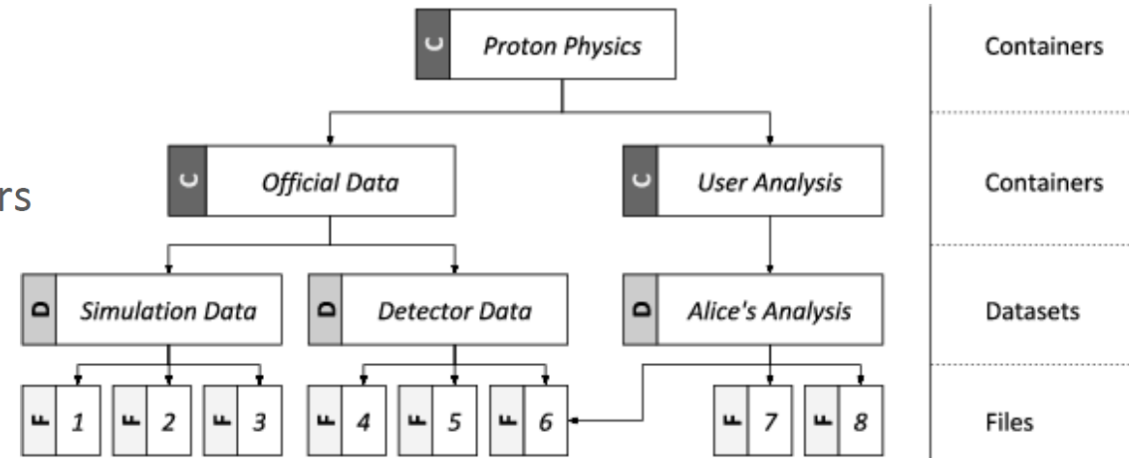
Container Collection of datasets and/or containers

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

`detector_raw.run34:observation_123.root`

scope

name



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

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

