Rucio in the DOMA ecosystem

Martin Barisits & Mario Lassnig
on behalf of the Rucio team
Rucio in a nutshell

- Developed by the High-Energy Physics experiment [ATLAS](https://arxiv.org)
- Rucio provides a complete and generic scientific data management service
  - Data can be scientific observations, measurements, objects, events, images saved in files
  - Facilities can be distributed at multiple locations belonging to different administrative domains
  - Designed with more than 10 years of operational experience in large-scale data management!
- Rucio manages multi-location data in a distributed environment
  - Creation, location, transfer, and deletion of replicas of data
  - Orchestration according to both low-level and high-level driven data management policies (usage policies, access control, and data lifetime)
- Rucio ([arXiv](https://arxiv.org)) is open source and available under Apache 2.0 license
- Make use of established open source tools
Data management at ATLAS

- ATLAS instance in a few numbers
  - More than 1B files, 450 PB of data
  - Up to 4M files/2.5 PB transferred per day
  - 10PB access via Rucio mover; >1000 active users

- Expect to increase one order of magnitude for Run4
Community

- **1st Rucio Community Workshop** was held at CERN on March 1st-2nd 2018 to present Rucio to scientific communities
- **1st Rucio Coding Camp** in November 2018
- **2nd Rucio Community Workshop** was held in Oslo on Feb 28th - Mar 1st
Third party copy

- **Rucio DOMA instance running full mesh test**
  - Functional and scale tests for xrootd/http
  - See TPC session

- **Protocol (compatibility) mapping**
  - Currently hardcoded (srm→gsiftp, https→davs, ...)
  - Plan to make this configurable → Allow protocol translation

- **Rucio support for Multihop**
  - FTS supports multi-hop
  - Possibility for Rucio to request Multihop transfers for RSEs without connection (=without distances)
    - Selecting hop-RSE based on SPF of distances

- **Support for transferring content of archives**
  - Rucio already supports downloading of archive contents via xrootd → Extend to TPC as well?
Third party copy

- **Non-X509 authentication (AAI) → JWT**
  - Not only limited to TPC, we want to focus on JWT for all Rucio AuthN/AuthZ
  - Token based delegation to FTS
  - Integration of Identity and Access Management service (IAM)?

- **HPCs**
  - Current data flows are limited due to HPC site policies
    - Workaround with Harvester (Edge services)
  - Rucio should be centrally involved in orchestrating HPC data workflows
    - gsiftp + client side load balancing OR using transfer mechanisms the site supports

- **FTS**
  - Rucio strongly connected to FTS
  - Intelligent grouping of transfers (→ session re-use), Scalability
Data access

- Rucio already provides several ways to access data
  - Rucio mover
  - Volatile RSEs populated by storage events (not transparent caches, but ~ cache-like)
  - Redirector with metalink (+root)
  - Proxy support for list_replicas (Xcache support)

- Actively following the work of DOMA access group
  - Will support/develop features needed to make access model work
  - Slight concern in terms of dataflow control loss
    - Especially if model foresees to have uncontrolled WAN transfers between caches
    - Extensive work has been done to be able to control dataflows (TPC, Rucio mover, etc.)
      - Nowadays possibility to control vast majority of data access in ATLAS (~12PB / day)
      - A move in the opposite direction raises concerns for data management
Reminder: Replica management in Rucio is based on replication rules
- Put 1 copy of file.001 on an RSE in country=uk&type=disk
- country and type are RSE specific attributes
- Rule engine matches the rule against existing replicas (rules); Only requests transfers if necessary
- Replication rules serve two purposes
  - Creation of transfers/replicas
  - Protecting replicas from deletion (as long as a rule covers them)

QoS capabilities need to be integrated in the replication rule (=replica management/storage management) concept
- Each rule will also express the QoS requirement of the rule
- Possibly the largest development in Rucio since its creation
Quality of Service

Objectives

1. Find the matching RSE based on the requested QoS of the rule
   - E.g. put 1 copy of file.001 on an RSE in country=uk with qos archive

2. Manage QoS over time
   - 1 copy of file.001 on an RSE in country=uk with qos latency<50 for 30 d
     after that change qos to latency<1000 for ∞ d
     ● Existing rule concept very well suited to express data placement over time
Quality of Service

● How to express QoS requirements in Rucio: Classes vs. attributes
  ○ Classes
    ■ “fast”, “custodial”, “cold”, “online”, ...
    ■ RSEs get tagged with classes (can be multiple, if multiple “zones” are supported)
    ■ Adv: Only a few classes to remember; Dis: might be inflexible
  ○ Attributes
    ■ Fixed amount of attributes: latency, throughput, resilience, cost, ...
    ■ RSEs get values assigned for each attribute
    ■ Adv: Flexible; Dis: Complicated
  ○ We can just support both, as classes could simply be mapped to sets of attributes

● QoS part of RSE expression or a separate field?
  ○ country=uk&fast or country=uk&latency<50 vs exp country=uk qos fast
  ○ Mostly cosmetic question as it won’t change underlying functionality
Quality of Service

- Rucio will support RSEs with multiple storage classes/zones
  - Do not want a half-solution where different RSEs represent storage zones in one storage
  - Conceptually this will be one RSE with multiple QoS zones
    - Rucio should be aware in which zone a replica resides!

- Need to agree on standard how to communicate QoS needs to storage
  - CDMI (Some work in the XDC CDMI reference implementation)?

- Can storage independently move a file into different QoS zone?
  - Different modes possible: no / ask / notify / mixed
  - Notify: Might be difficult since it could invalidate rules
  - Mixed: Notify for move to “higher” area, ask for move to “lower”
    - This implies a de-facto ranking of QoS classes/attributes
Quality of Service

Do we want this defacto-ranking? → Implications for data placement

- Example: **Rule A**: country=uk qos fast AND **rule B**: rule country=uk qos archive
- Without: 2 replicas: 1 on fast storage; 1 on archive storage
  - Restricts Rucio in optimizing data placement
- With ranking: Only 1 replica on fast storage; When **rule A** gets removed transition data to archive storage
  - Better fits the “Rucionic” way of optimizing placement
- **Ranking not obvious, as classes do not “outrank” each other in all attributes**
  - E.g. fast has lower latency compared to archive, but archive might have better resilience…

Resilience

- Rucio will not keep track of how many replicas a storage creates internally
- The increased data “safety” of the storage is reflected in the resilience qos attribute of the RSE
Summary

- Rucio evaluated/used by multiple experiments
  - Naturally lots of interest about our plans in TPC / Access / QoS
- Rucio is participating in all three DOMA workgroups
- TPC
  - Multiple enhancements in Rucio planned
  - Should get more involved in orchestrating data flows to HPCs
- Access
  - Several ways to access data already supported (mover, volatile RSE, redirector, proxy)
  - Some concerns for control loss
- QoS
  - Replication rule concept needs to be extended to QoS
  - Classes vs Attributes, Independent QoS transitions, ranking of QoS classes
### More information

<table>
<thead>
<tr>
<th>Category</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Website</td>
<td><a href="http://rucio.cern.ch">http://rucio.cern.ch</a></td>
</tr>
<tr>
<td>Documentation</td>
<td><a href="https://rucio.readthedocs.io">https://rucio.readthedocs.io</a></td>
</tr>
<tr>
<td>Repository</td>
<td><a href="https://github.com/rucio/">https://github.com/rucio/</a></td>
</tr>
<tr>
<td>Continuous Integration</td>
<td><a href="https://travis-ci.org/rucio/">https://travis-ci.org/rucio/</a></td>
</tr>
<tr>
<td>Images</td>
<td><a href="https://hub.docker.com/r/rucio/">https://hub.docker.com/r/rucio/</a></td>
</tr>
<tr>
<td>Online support</td>
<td><a href="https://rucio.slack.com/messages/#support/">https://rucio.slack.com/messages/#support/</a></td>
</tr>
<tr>
<td>Developer contact</td>
<td><a href="mailto:rucio-dev@cern.ch">rucio-dev@cern.ch</a></td>
</tr>
</tbody>
</table>