Some approaches for metadata in HEP

A. Redelbach

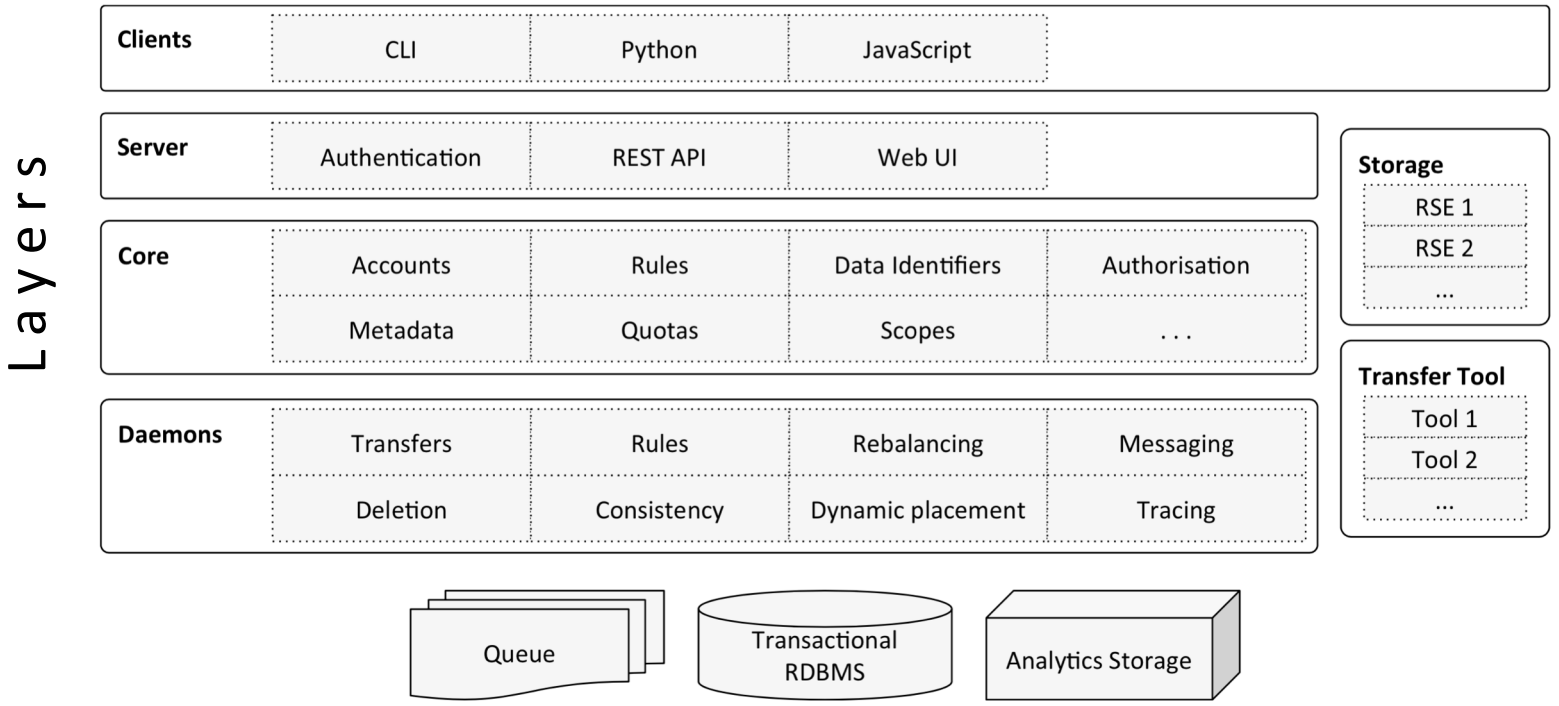
Outline

- Some features of Rucio
- Non-Rucio solution in ALICE (for Calibration and Conditions Database)

→ Sketch some relevant (documented) numbers, no comprehensive overview of systems

Rucio: Some features and experiments

Installation for ATLAS Experiment (http://rucio.cern.ch/publications.html):
More than 450 Petabytes of data, stored in a billion files, distributed over 120 data centres globally, orchestrating an Exabyte of data access and transfer per year

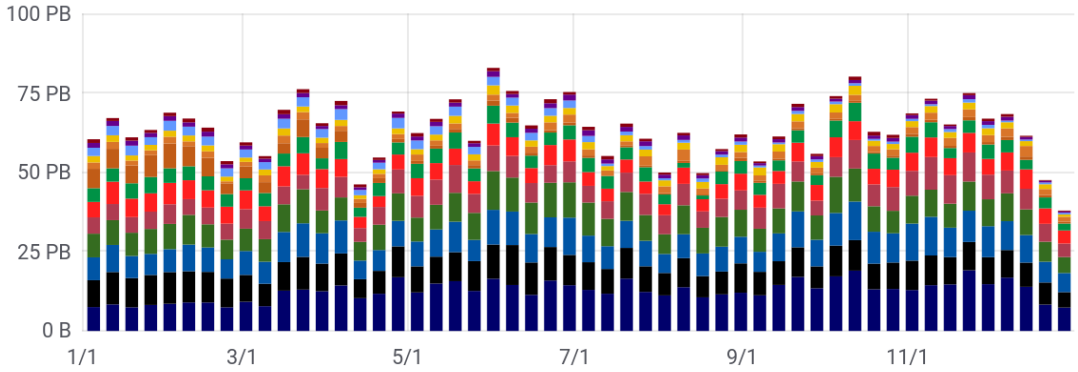


Queueing systems, transactional relational database management systems (RDBMS), non-relational analytics storage

Rucio Storage Element (RSE):

- minimal unit of globally addressable storage
- holds description of all attributes to access storage space, (hostname, port, protocol, local file system path)
- RSEs can be **extended** with arbitrary key-value pairs

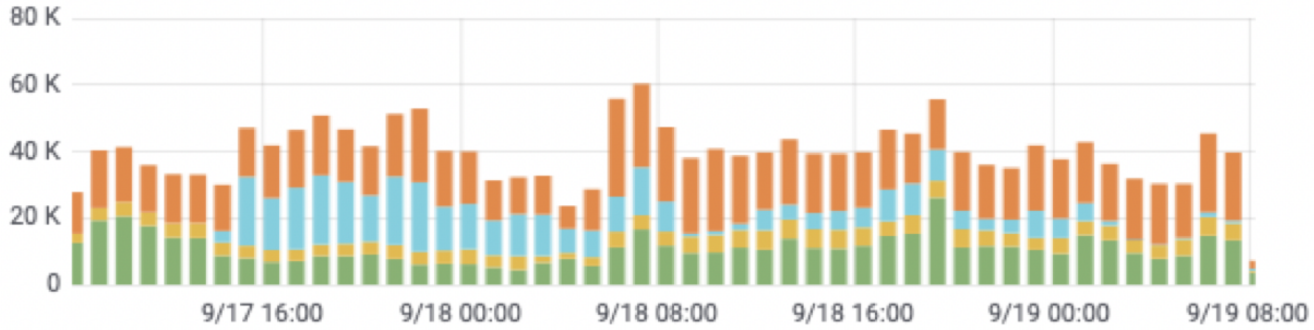
Rucio: Some features and experiments



ATLAS data transfers and downloads are regularly above 50 Petabytes per week throughout 2019.

Computing and Software for Big Science (2019) 3:11
https://doi.org/10.1007/s41781-019-0026-3

CMS Rucio team worked towards incorporating Rucio into CMS software system needed for Run 3 operations. **Rucio** has met these challenges and will be deployed as **CMS data management solution in 2020**.
https://doi.org/10.1051/epjconf/202024504033



Number of files transferred with PhEDEx (orange and yellow), Rucio (blue), and by users (green) during the "Million File Test."

"Million File Test" aims to distribute multiple copies of data with Rucio in a way that mimics the real data distribution of CMS.
On that metric (i.e. moving large numbers of files), the test was a success

ALICE: Towards Calibration and Conditions Database

Run 1&2: Conditions data

- Run number granularity
- Part of the Grid file catalogue
- Can be accessed by any job or user, same semantics as any data file
- Special `*find*` queries to assemble the entire set of such files needed by a job

Run-3: Calibration and Conditions data

New constraints from real time reconstruction

- Timeframe O(10ms) resolution for objects

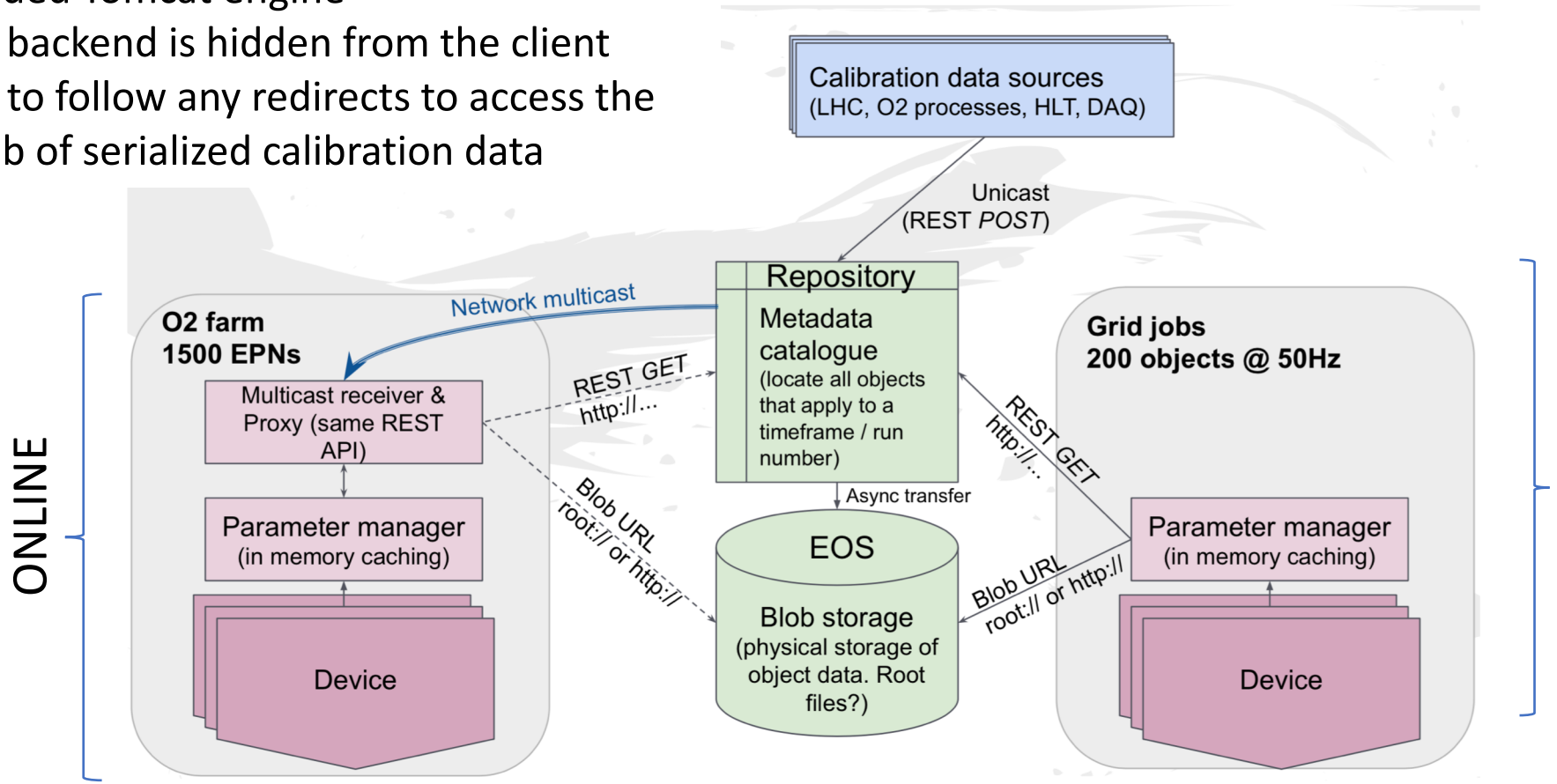
→ challenge of generating ms-precise time-stamp for raw data

- Real-time distribution (feedback loop) in online cluster
- O(100M) objects to store in one series
- O(kHz) access rate from Grid jobs

ALICE: Calibration and Conditions Database

CCDB repository:

- Implemented as a HTTP REST web server using an embedded Tomcat engine
- Metadata backend is hidden from the client
- Client has to follow any redirects to access the binary blob of serialized calibration data



binary large object (**BLOB**): collection of binary data stored as a single entity

ALICE: REST queries and PostgreSQL performance

CCDB repository: implemented as a HTTP REST web server

Query format: …/Task/Detector/[more/]/*tStart*[/tEnd][/UUID][/key=value]

Folder structure is between 1 and 10 levels deep
For most requests the reference time is mandatory

POST - upload new objects
PUT - update existing objects
GET - retrieve the content (query by path / time)
HEAD - only the headers (metadata)
DELETE - obvious

/browse
GET - listing of all matching objects and subfolders, with regexp matching for the namespace
/latest
GET - like the above but only the most recent object in each folder is returned == current calibration data

PostgreSQL backend performance:
INSERT or SELECT methods allow **rates slightly above 10 kHz**
Flat performance (as function of number of rows)

Towards concepts for metadata in TA5

Related to **deliverable**: D-TA5-WP2-1 (31 May 2022): Curation & metadata schemes for dynamic filtering.

Collection of relevant requirements and open points:
Internal document: https://gitlab-p4n.aip.de/-/ide/project/punch/intra-docs-content/tree/master/-/docs/TA5/TA5-metadata.md

Complementary approaches for online/offline workflows in HEP
Different modi of operation in astrophysics (time domain, imaging, spectroscopy, polarisation)
Existing paper from MPIfR

Rucio experiences: Used by additional collaborations in production, (ASGC/AMS, Xenon1T, CMS, DUNE), evaluated by other collaborations also outside of HEP, such as LIGO and SKA

Preparation of PUNCHLunch:
- Developer of Rucio
- SKA experiences (workflow, issues)

Work on actual note to **summarize requirements and concepts**:
→ Contributions welcome (and needed!)