

Rucio - S3-compatible access interface



IRIS-HEP Fellow Final Presentation

Kyrylo Meliushko

Mentors: Lukas Heinrich (TUM), Matthew Feickert (UWM), Mario Lassnig (CERN), Martin Barisits (CERN)

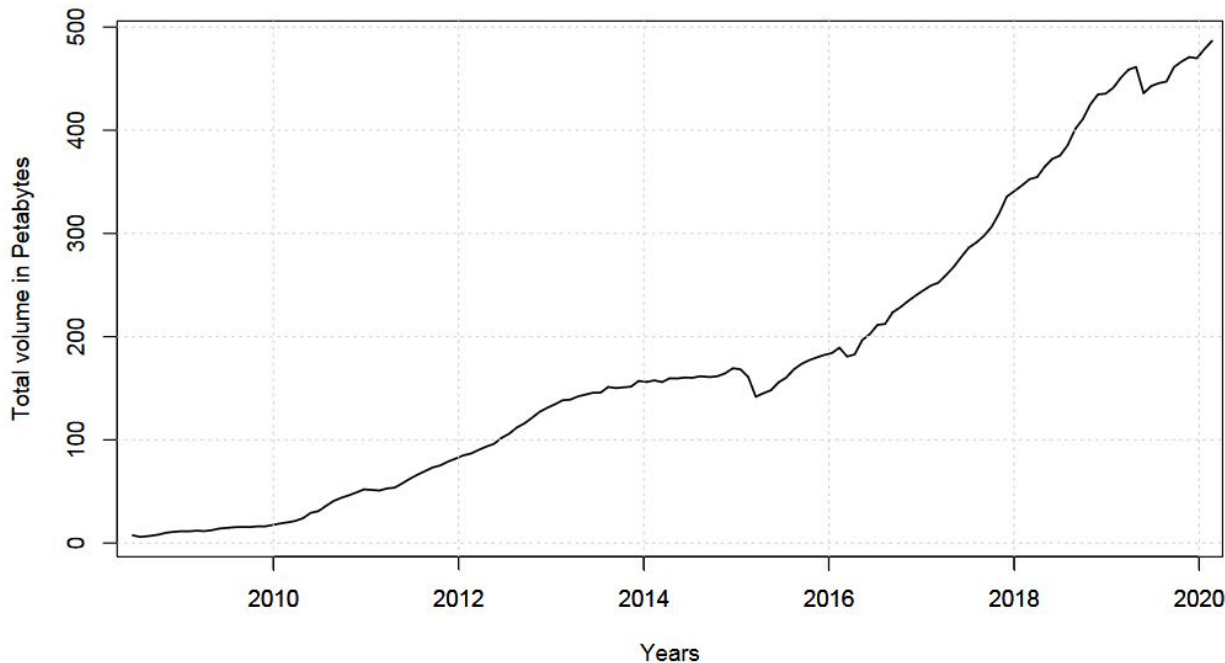
October 11, 2023

Agenda



- 1) Understanding Rucio
- 2) Project overview and goals
- 3) Required knowledge and my work
- 4) Personal takeaways

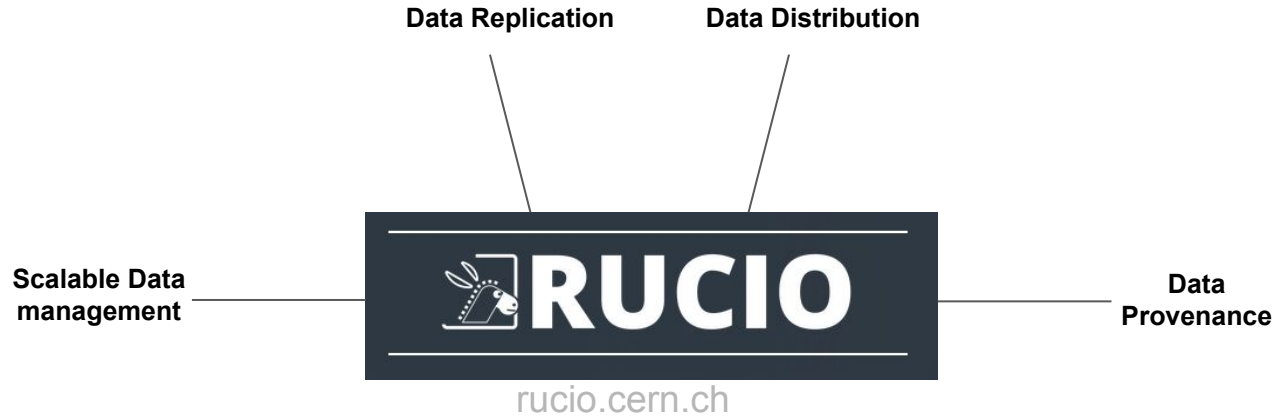
Understanding Rucio



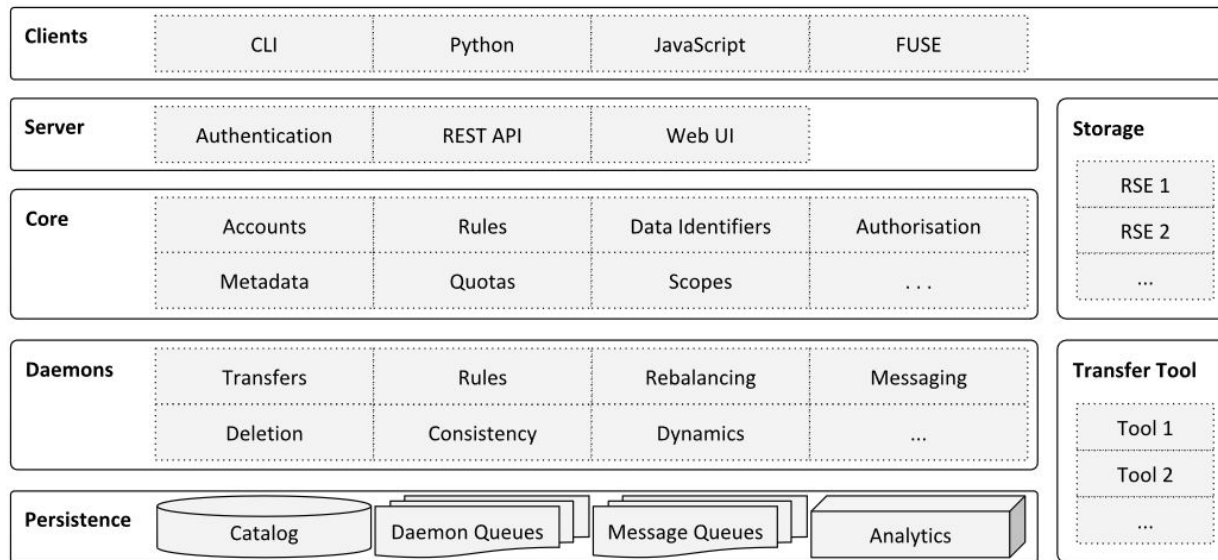
Barisits, M., Beermann, T., Berghaus, F. *et al.* Rucio: Scientific Data Management. *Comput Softw Big Sci* **3**, 11 (2019). <https://doi.org/10.1007/s41781-019-0026-3>

Figure 1: The cumulative ATLAS data volume approaches 500 Petabytes in early 2020. Growth has been linear with respect to the scale of the experiment, with considerable data deletion before longer observation periods.

Understanding Rucio



Understanding Rucio



Barisits, M., Beermann, T., Berghaus, F. *et al.* Rucio: Scientific Data Management. *Comput Softw Big Sci* **3**, 11 (2019). <https://doi.org/10.1007/s41781-019-0026-3>

Project overview and goals (S3)

S3 - Simple
Storage
Service



↓

Portable way to share less-important experiment data/notes
between analysts



S3



Project overview and goals

Rewind: Interface to interact with S3 directly from Rucio.

Weeks 1-3: Explore Rucio, MinIO's "mc" tool, S3 API. Prepare development environment.

Weeks 4-5: Implement basic data transfer capabilities between Rucio and S3.

Weeks 5-7: Integration testing, credentials extraction.

Weeks 7-10: Additional features: access sharing, quotas, ...

Weeks 10-12: close GitHub Issues, refine docs, prepare final presentation.

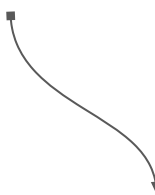
A large, thick, grey arrow pointing downwards, indicating the duration of the project.

July 3 - Sept 22

Required knowledge & my work



- S3 -> python boto3 library
- General Rucio architecture

- 
- Working with Python script as CLI
 - Pytest framework -> remote testing with Docker
 - General S3 usage, exceptions when calling API's
 - Triggering other Rucio clients (DIDClient, DownloadClient), as well as exploring LFN2PFN for reproducible scope:path naming
 - Loading and caching of S3 Credentials
 - Understanding of communication between code of Rucio components

Required knowledge & my work (Changes)



```
rucio s3 make-bucket user.lheinric:someuniqueusername
```

```
└──┬──  
    └── rucio bucket create user.lheinric:/someuniqueusername/
```

```
rucio download user.lheinric:someuniqueusername/file.root
```

```
└──┬──  
    └── rucio download user.lheinric:/someuniqueusername/file.root
```

```
rucio s3 credentials > ~/.mc/config.json
```

```
└──┬──  
    └── removed since raises a lot of security issues
```

Required knowledge & my work (Structure)



General S3 implementation ->

DIDClient
UploadClient
S3Client
CredentialClient
DownloadClient
...

```
class S3Client:
    """S3 client class
    ...
    """
    def __init__(self, _client=None, logger=None, config: dict = None):
        """
        Initialises the basic settings for an S3Client object
        ...
        """

    def bucket_create(self, bucket_path):
        """Create an S3 bucket.
        param bucket_path: Bucket bucket_path, e.g. user.dquijote:/mybucket/
        :return: True if bucket created, else False
        ...
        """

    def bucket_upload(self, from_path, to_path):
        """Upload a file/folder to an S3 bucket.
        :param from_path: Path to the file/folder to upload
        :param to_path: Bucket path, e.g. user.dquijote:/mybucket/file.ext
        :return: True if file/folder uploaded, else False
        ...
        """

    def _register_bucket_did(self, scope, name):
        """Register scope:name as DatasetDID to track with Rucio
        :param scope: Scope, e.g. user.dquijote
        :param name: Folder path to register, e.g. /data/exp22/
        ...
        """

    def bucket_download(self, from_path, to_path):
        """Download a file/folder from an S3 bucket.
        :param from_path: Bucket path, e.g. user.dquijote:/mybucket/file.ext
        :param to_path: Path to the file/folder to download
        :return: 0 if data written successfully, else 1
        ...
        """
```

Required knowledge & my work (Structure)



```
def get_s3_credentials(path_to_credentials_file: Optional[Union[str, os.PathLike]] = None):
    """ Returns credentials for S3. """

    path = ''
    if path_to_credentials_file:
        path = path_to_credentials_file
    else: # Use file defined in th RSEMgr
        for confdir in get_config_dirs():
            p = os.path.join(confdir, 's3client.cfg')
            if os.path.exists(p):
                path = p

    try:
        with open(path) as cred_file:
            credentials = json.load(cred_file)
    except Exception as error:
        raise exception.ErrorLoadingCredentials(error)
    return credentials
```

Configuration loading

```
def test_create_bucket(s3_client):
    """S3CLIENT: Create a bucket"""
    # TODO: add more scopes for validation
    scope = "user.dquijote:/folder/"
    status = s3_client.bucket_create(scope)
    assert status == 0

def test_upload_download_bucket(s3_client, file_factory):
    """S3CLIENT: Upload a bucket"""
    scope = "user.dquijote:/folder/"
    local_file = str(file_factory.file_generator())
    fn = str(os.path.basename(local_file))
    did_name = scope + fn
    base_name = generate_uuid()
    s3_client.bucket_upload(from_path=local_file, to_path=scope)

    with TemporaryDirectory() as tmp_dir:
        result = download_client.download_dids([{'did': '%s:%s.*' % (scope, base_name), 'base_dir': tmp_dir}])
        # triggers s3_client.bucket_download(from_path=scope + fn, to_path=tmp_dir)
        _check_download_result(
            actual_result=result,
            expected_result=[
                {
                    'did': did_name,
                    'clientState': 'DONE',
                }
            ],
        )
```

S3Client tests

Personal takeaways



- Pytest: integration and unit tests
- Test driven development (TDD)
- Optimal architecture for big projects
- S3 and it's identity and access management
- Improved python knowledge, remote debugging skills
- General Rucio understanding

...

**Thank you
for your
attention!**

