SPD data management

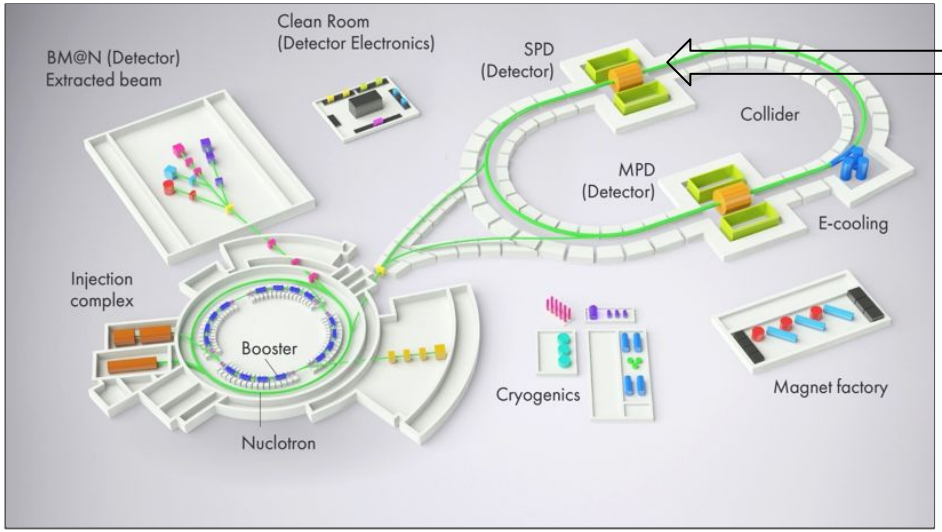
Alexey Konak
JINR MLIT
konak@jinr.ru

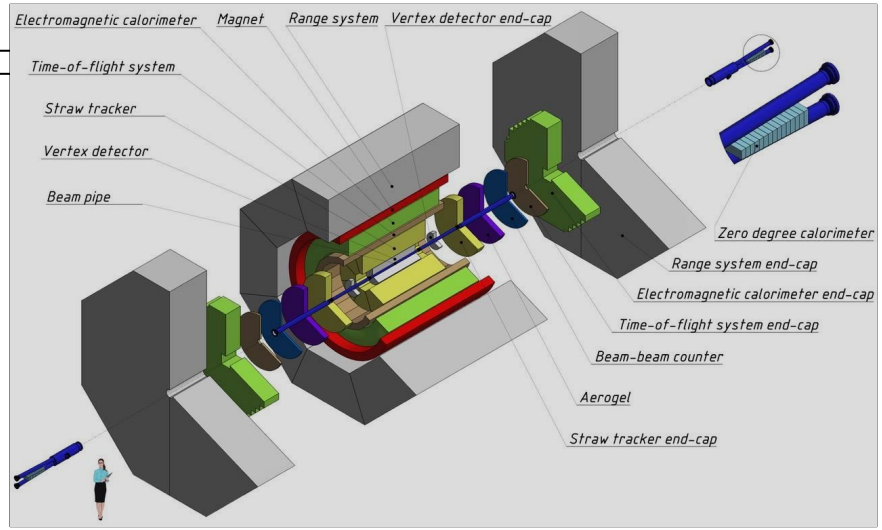
11th International Conference GRID’2025
Dubna, JINR, MLIT
08.07.2025

Spin Physics Detector (SPD)

The spin structure of the **nucleon** is one of the fundamental properties of matter. The spin of a nucleon is distributed between its components — **quarks** and **gluons**, and their mutual movement.

The EMC, HERMES, and COMPASS experiments have made it possible to study in detail the contribution of **quarks** to spin. However, the role of **gluons** remains poorly understood and requires further research.



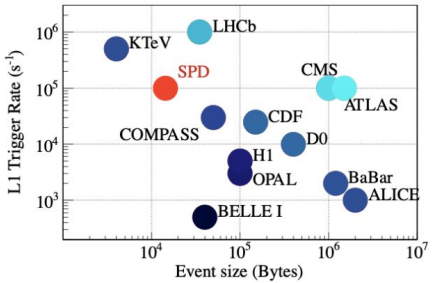


The SPD facility is being created for a more accurate study of the contribution of **gluons** to the spin of the **nucleon**.

SPD as data source

The expected event rate of the SPD experiment is about 3 MHz (pp collisions at $\sqrt{s}$ = 27 GeV and 10^{32} cm^{-2}s^{-1} design luminosity). This is equivalent to a raw data rate of 20 GB/s or 200 PB/year, assuming a detector duty cycle is 0.3, while the signal-to-background ratio is expected to be on the order of 10^{-5}. Taking into account the bunch-crossing rate of 12.5 MHz, one may conclude that pile-up probability cannot be neglected.

- SPD TDR



The goal of the online filter is at least to decrease the data rate by a factor of 20, so that the annual growth of data, including the simulated samples, stays within 10 PB. Then, data are transferred to the Tier-1 facility, where a full reconstruction takes place and the data is stored permanently. The data analysis and Monte-Carlo simulation will likely run at the remote computing centres (Tier-2s). Given the large data volume, a thorough optimization of the event model and performance of the reconstruction and simulation algorithms are necessary.

- Data from the detector – 20 GB/s (or 200 PB/year "raw" data, ~3*10^13 events/year)
- Simulation results – ??? (the exact volume is unknown, but there will be no less of them than the data from the detector.)
- Data of various intermediate formats along the way from “raw” to ready for analysis by physical groups – ??? (there will be a lot of them...)

About Rucio

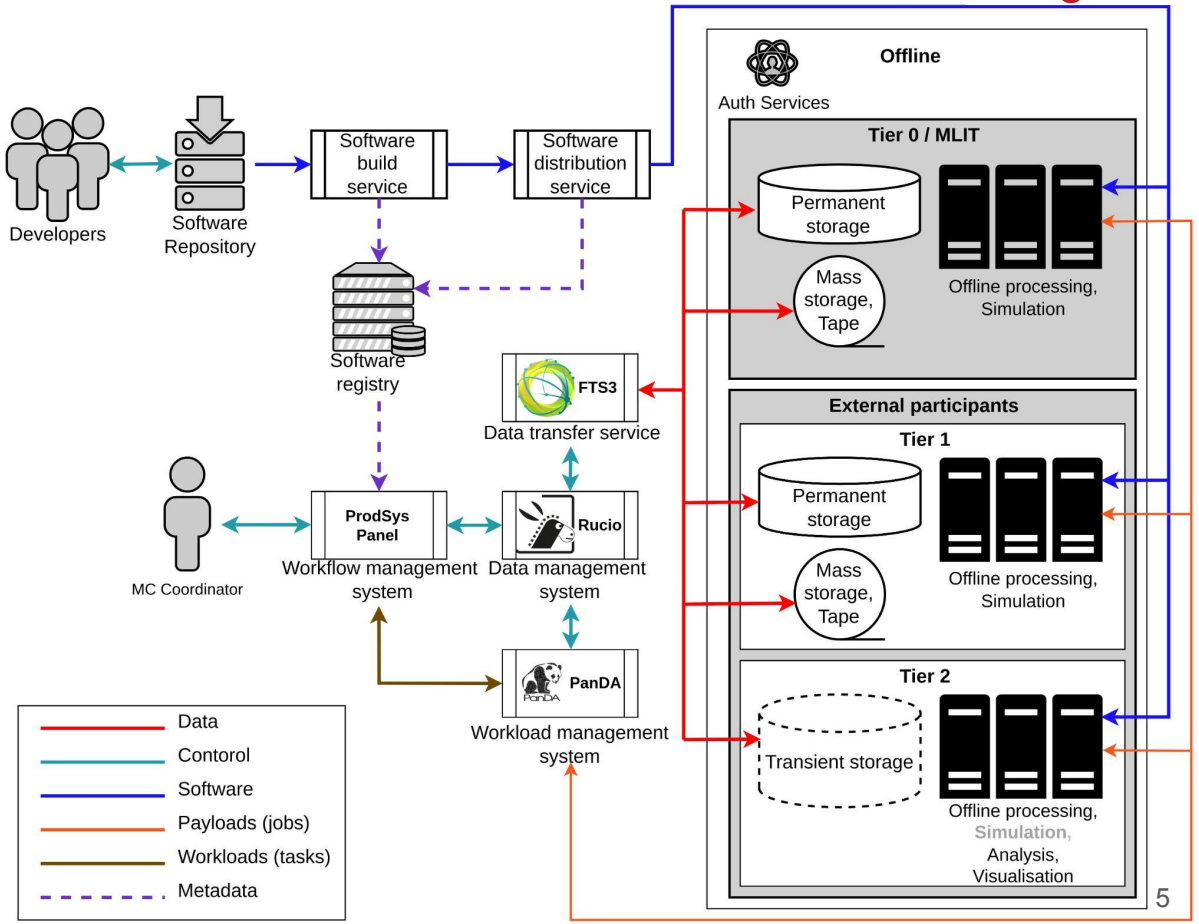
Rucio is an open-source software framework that provides functionality for data management and access in a distributed storage environment.

Currently, the Rucio system can be used to:

- data catalogue;
- organize data in a hierarchical structure for easy navigation and management;
- storage of any types of experimental data;
- storage and management of metadata;
- data lifecycle management;
- unified interaction of a heterogeneous network and storage infrastructure;
- adaptive data replication and recovery;
- automated data transfer between storages;
- providing monitoring metrics: data usage, system performance, services health.

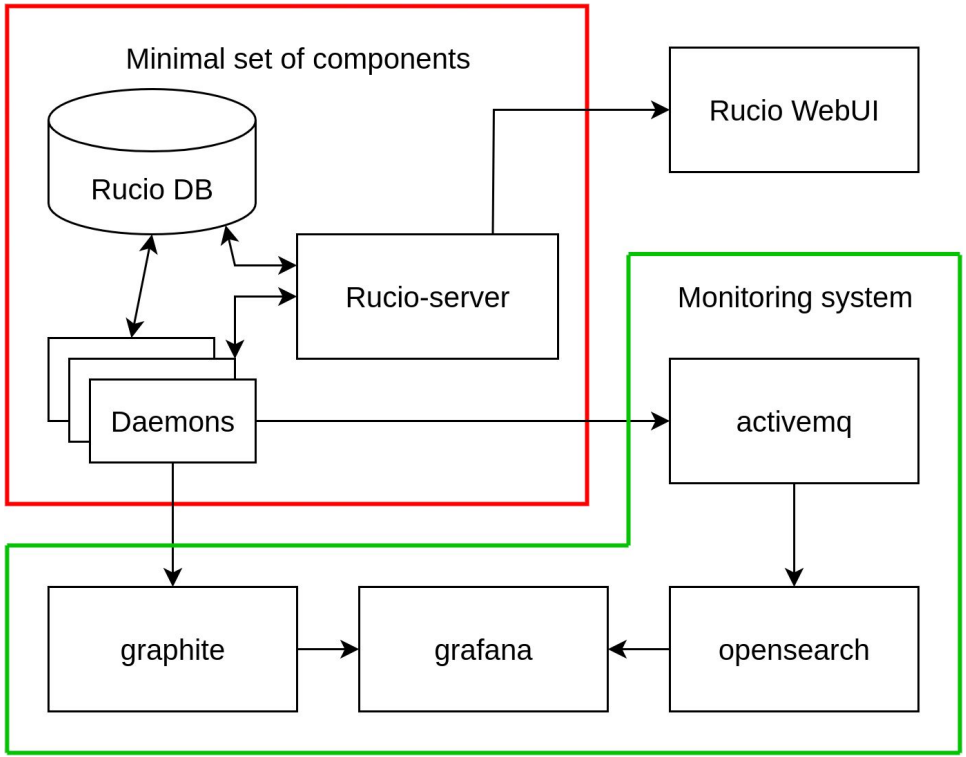
Starting point

- deploying all necessary components of Rucio
- figure out how it should work ;-)
- general configuration
- integration into the SPD distributed data processing system



Deploying

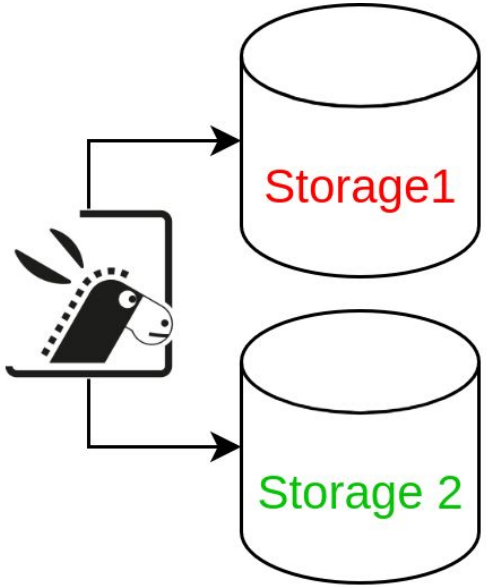
At the moment, the required set of system components of three Rucio-servers are deployed in Docker containers based on JINR cloud computing infrastructure



- **Production Infrastructure** with main Rucio-server which work stable and used for the needs of the SPD collaboration.
- **Development environment.** This infrastructure is used for development, testing and debugging.
- **Integration environment.** All updates and innovations are checked and tested on this installation before being put on the Prod infrastructure.

How does it work

Rucio catalogue	
File1 -> Storage1	PFN (f1 on s1)
File2 -> Storage2	PFN (f2 on s2)
File3 -> Storage1	PFN (f3 on s1)
File3 -> Storage2	PFN (f3 on s2)



Storage1 catalogue	
File1	PFN (f1)
File3	PFN (f3)

Storage2 catalogue	
File2	PFN (f2)
File3	PFN (f3)

1) Rucio knows protocols to get/put files on storages and PFNs of files that registered in Rucio
2) If File2 need on Storage1 – we make rule for Rucio, it make transfer and register file on storage (in Rucio catalogue).
3) If **File3** is lost on Storage2 – Rucio figure it out, get **File3** from Storage1 and put it on Storage2

Configuration

Main configuration:

- DB connection
- Add host certificate
- Update CA trust list
- Automate of updating CRLs
- Turn on x509_proxy module

Daemons configuration:

- connection with external services
- configure special information

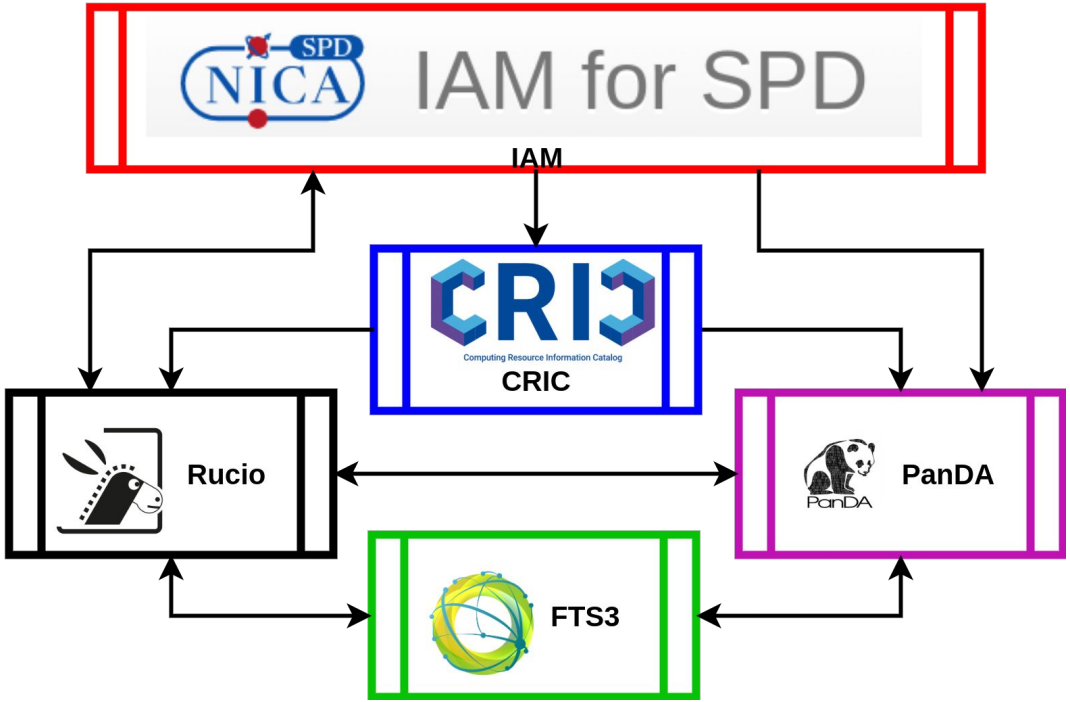
Configuration of Storage Elements:

- Add RSEs
- Definition of protocols on each RSE
- Add attributes to each RSE
- Definition of “distances” between RSEs

Account configuration:

- Add accounts
- Add authentication information for each account
- Add scope for each account
- Add quota for each account

Integration into offline data processing system

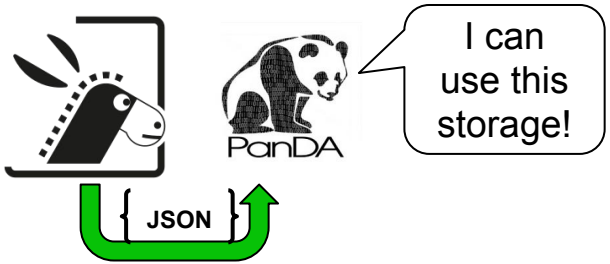


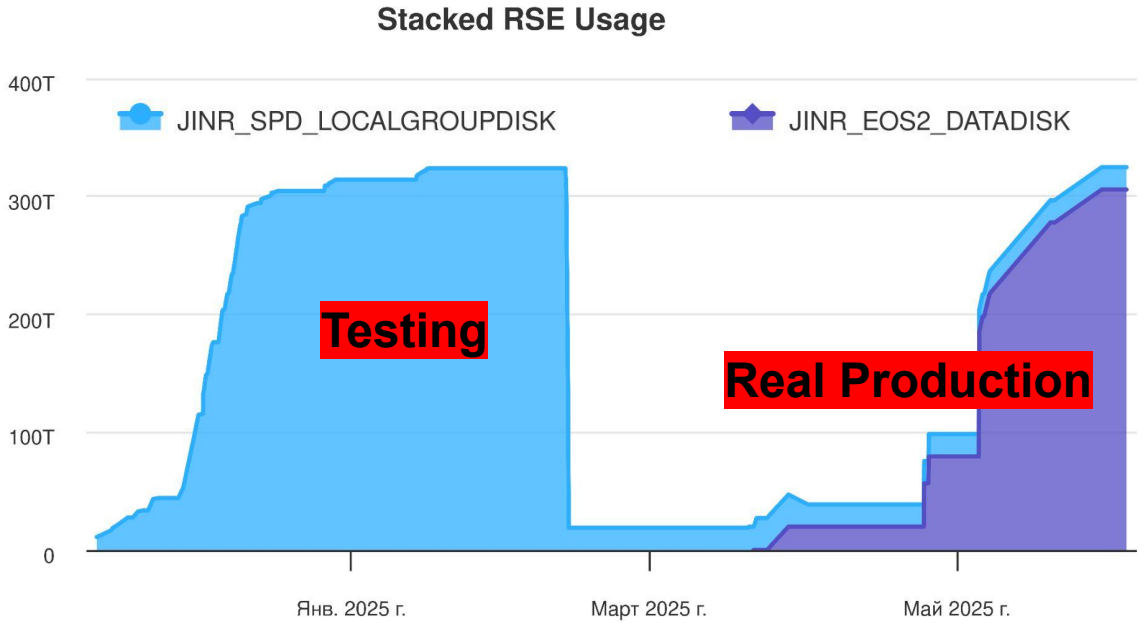
- Identity and Access Management (IAM) – manage authorization policies on distributed resources.
- Computing Resource Information Catalog (CRIC) used to manage and provide information about Storage and Network resources.
- Production and Distributed Analysis (PanDA) – a workload management system for distributed computing environments such as GRID.
- File Transfer Service 3 (FTS3) – a service for copying of data between storages.

PanDA integration

- Add special account for PanDA (plus quota, identity information, scope etc…)
- Make mechanism to inform PanDA about storage usage

Special utility was developed to inform PanDA

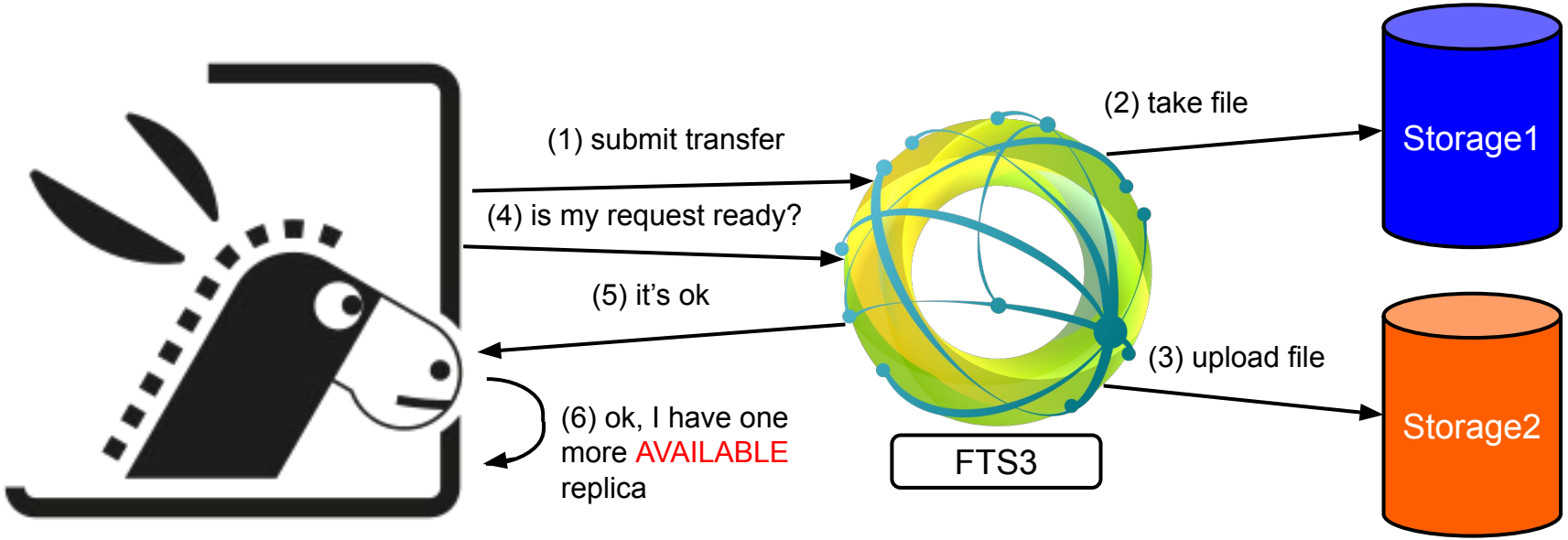




Currently, Rucio is used for mass production of SPD. During production, we tested interaction of PanDA and Rucio in various forms as well as different data organisation.

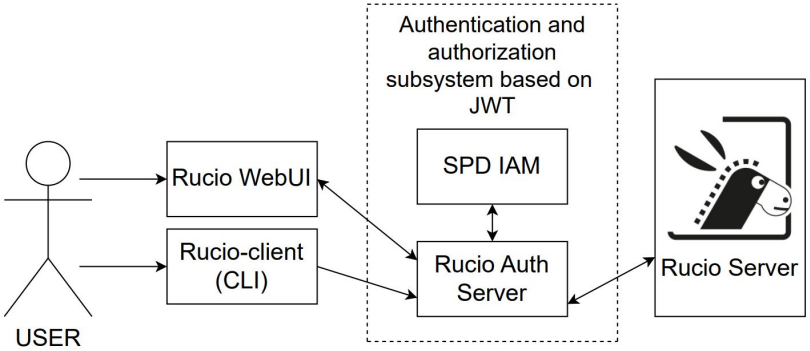
FTS3 integration

- Add FTS host:port as attribute for each RSE
- Add FTS host:port in configuration for special Daemons
- Automate fts delegation and getting proxy cert for Daemons



IAM integration

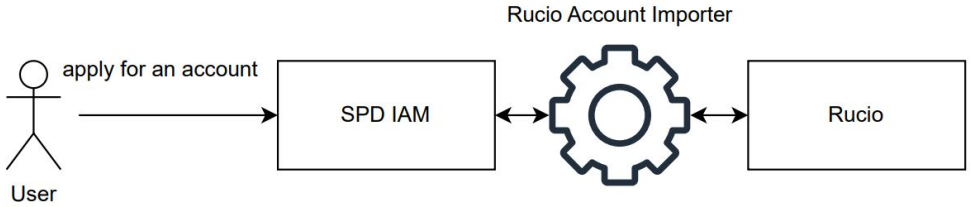
- Register authentication client for Rucio in SPD IAM
- Configure info about Identity Provider on Rucio-server side (SPD IAM in this case)
- Configure Daemon that interacts with tokens obtained in SPD IAM
- Automate of user registration



SPD IAM provides authentication to all services and systems (including Rucio) with an access token and an ID token obtained during authorization in SPD IAM.

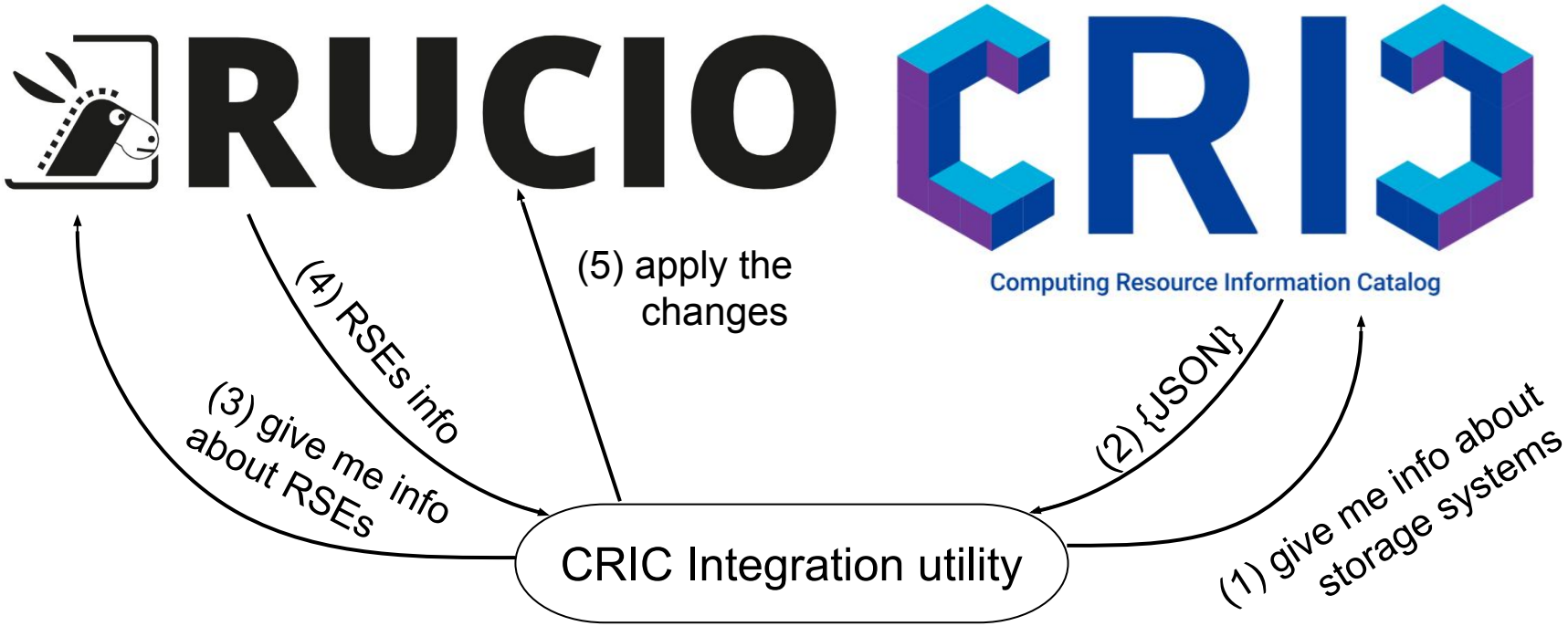
Special utility that imports **user** and **group** accounts from SPD IAM to Rucio was developed. This solution simplifies the access control process.

The Rucio Account Importer is configured to run in cron once per day. The utility adds new accounts and updates the identification information of existing Rucio accounts.



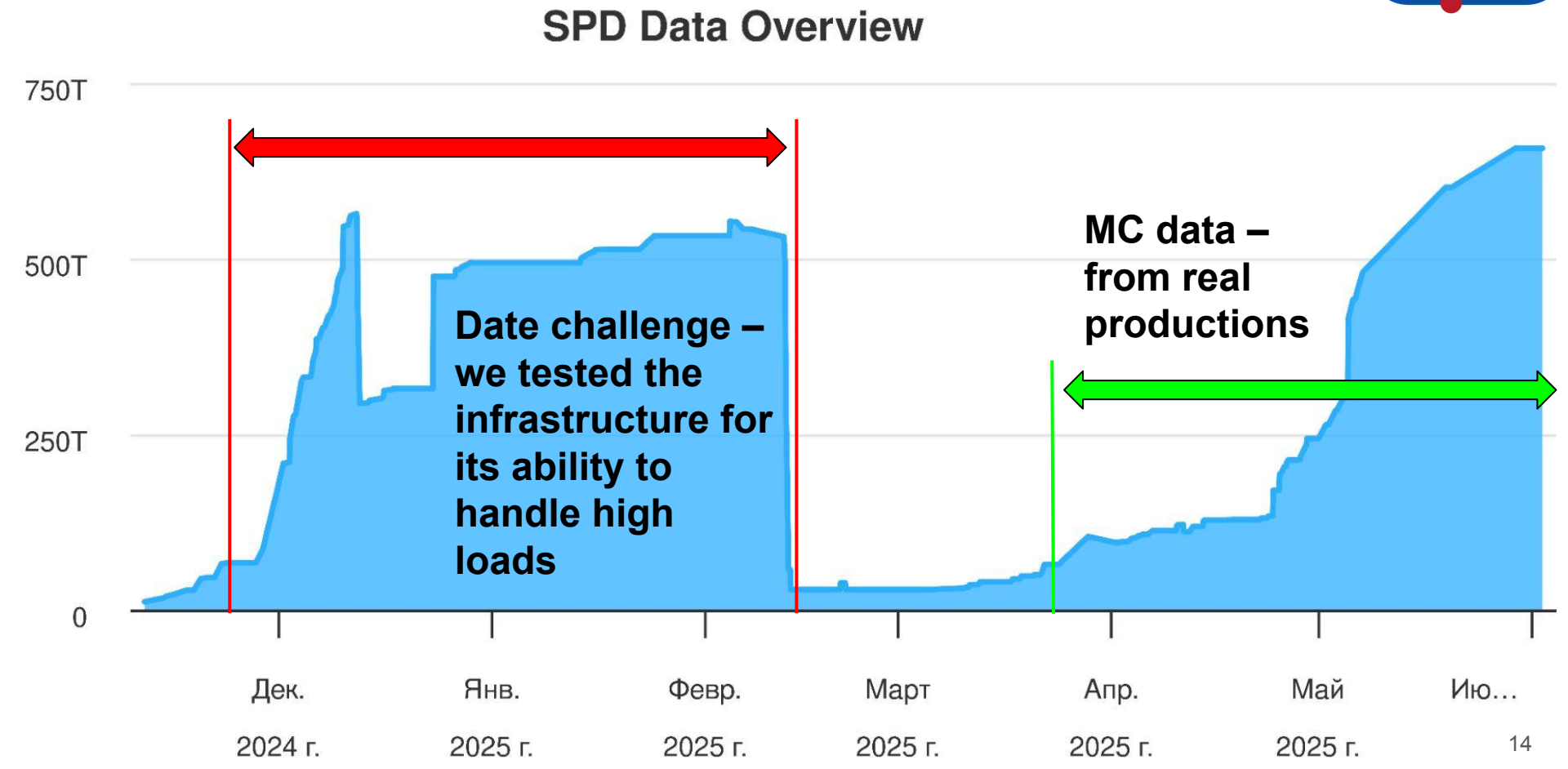
CRIC integration

- Make mechanism to manage storage system configuration in Rucio from CRIC



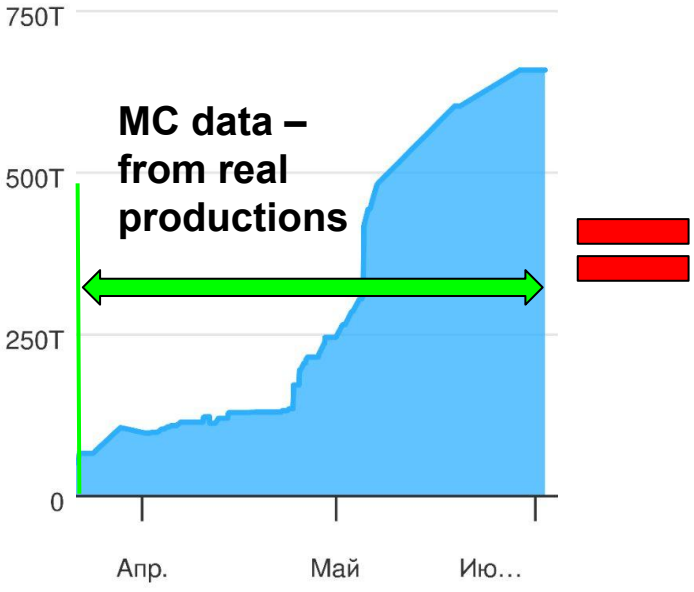
A module has been developed for importing configuration information about storage systems from CRIC to Rucio. This utility is configured to run in cron once per hour.

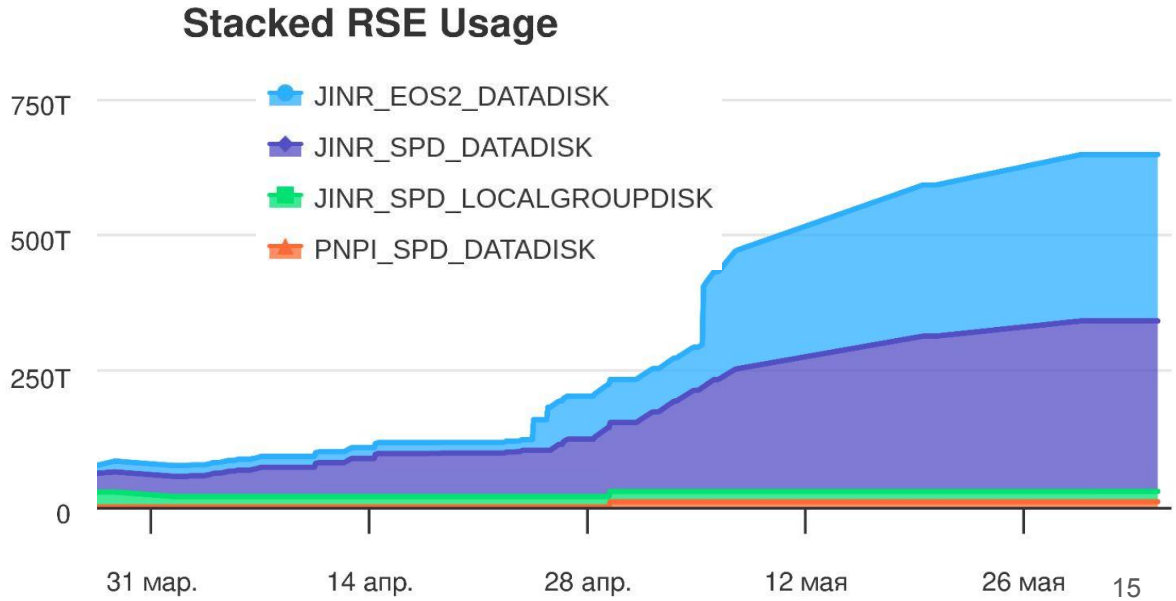
Operation


SPD Data Overview
750T
500T
250T
0
Date challenge – we tested the infrastructure for its ability to handle high loads
MC data – from real productions
Дек. 2024 г.
Янв. 2025 г.
Февр. 2025 г.
Март 2025 г.
Апр. 2025 г.
Май 2025 г.
Ию…

Data Management

We have several storage facilities:
- **SPD EOS**: Since this year, SPD has own EOS in JINR and we are now transferring data from the old to the new one.
- JINR EOS: **DATADISK** – for production data generation, **LOCALGROUPDISK** – is used for results of test data productions.
- **PNPI EOS**: It currently stores a bit of data. It will be used to store replicas.





Future plans

- Monitoring system – monitoring system to monitor the state of the system and its performance, as well as user activity and storages status.
- Development of a lifetime model and user policy.

Thank you for your attention!

Backup slides

Quick terminology recap

File – the smallest operational unit of data in Rucio.

Dataset – a named set of files.

Container – a named set of datasets or, recursively, containers

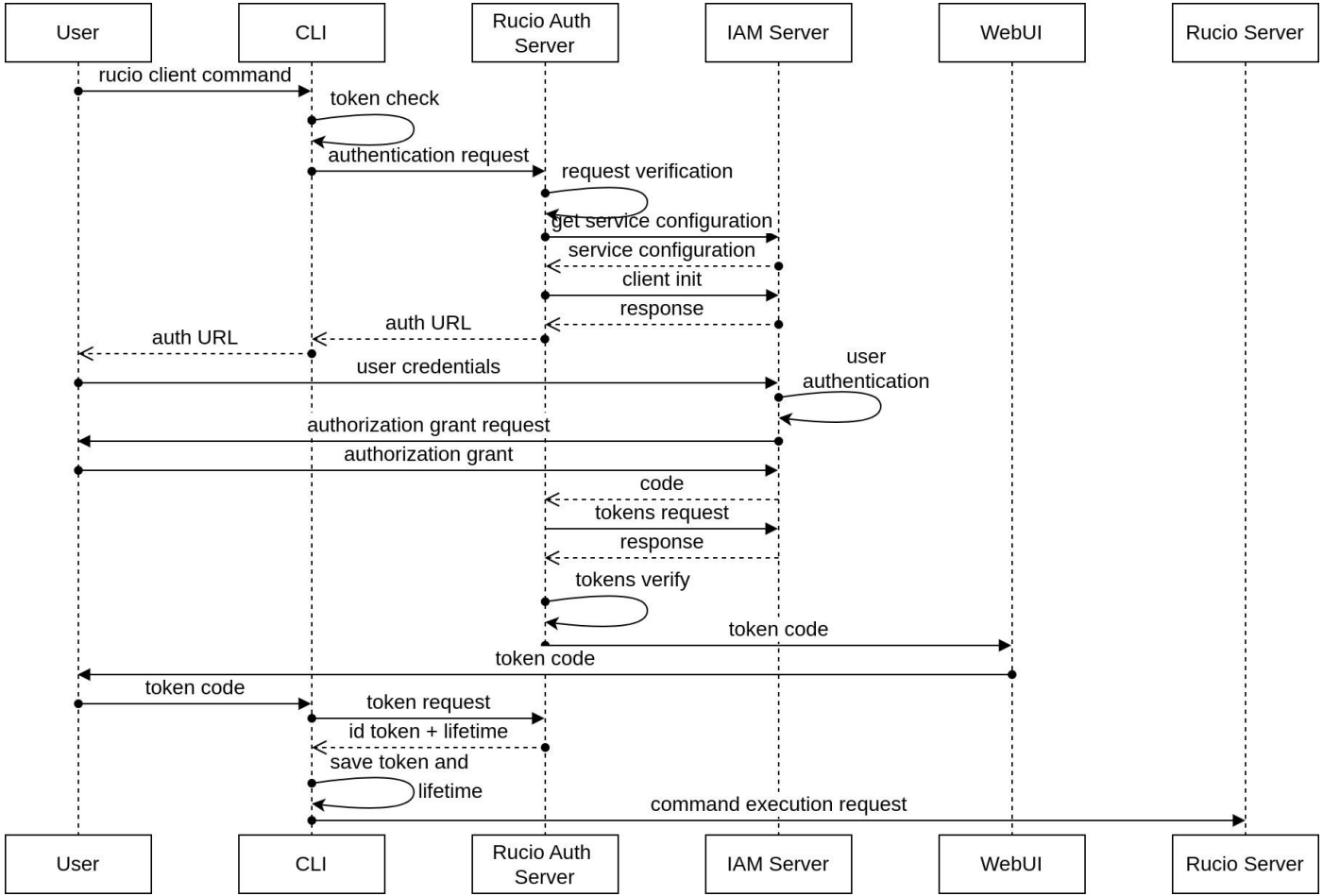
DID – rucio LFN for data (file/dataset/container) as combination of a scope and a name.

Scope – a scope partitions the namespace into several sub namespaces.

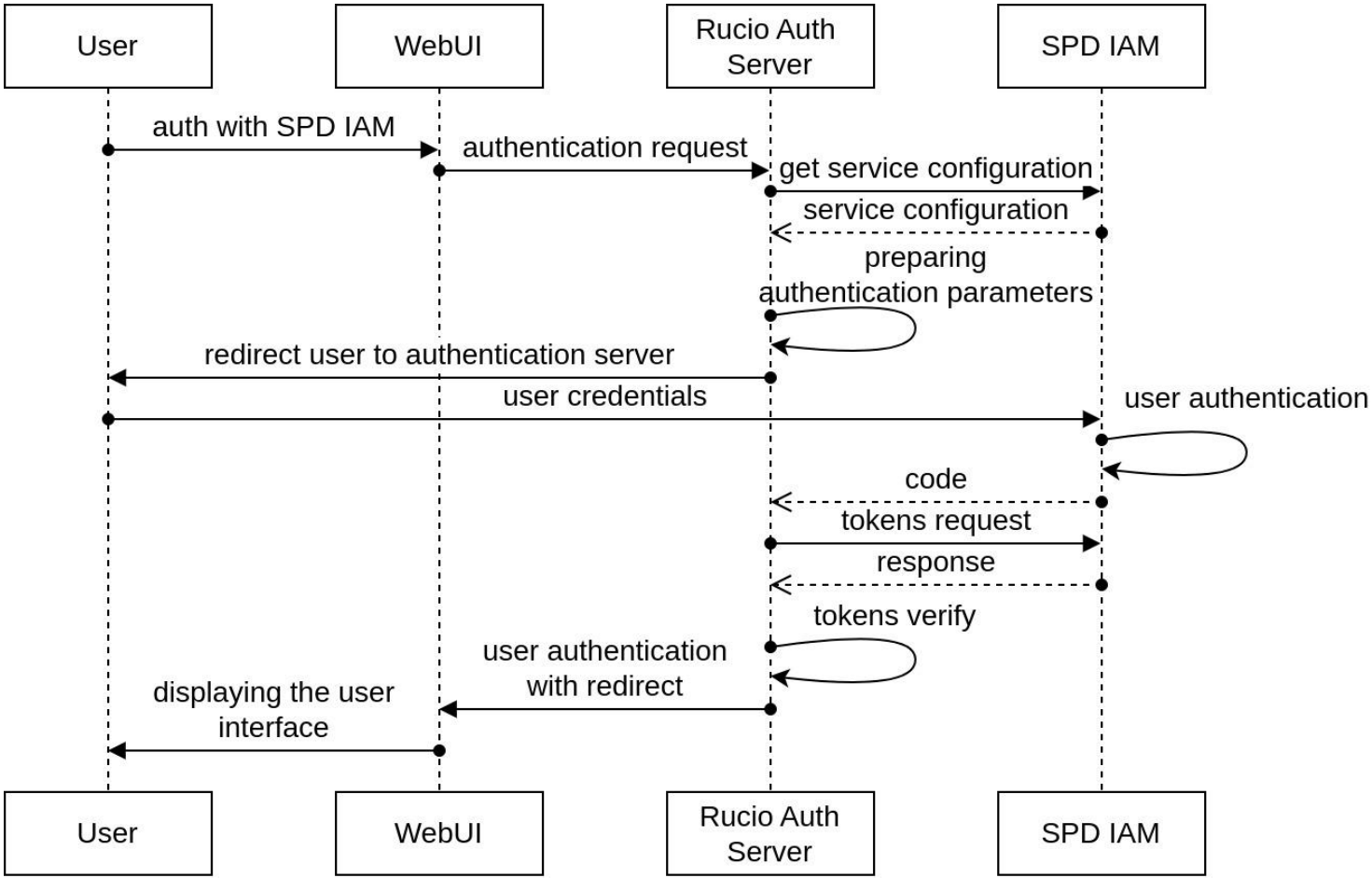
Replica – a managed copy of a file.

RSE – the logical abstraction of a storage system for physical files. It has a unique identifier and a set of meta attributes describing properties.

Authentication flow with IAM (CLI)



Authentication flow with IAM (Rucio WebUI)



Rucio Account Importer [1/3]

Rucio Account Importer is designed to import accounts from SPD IAM to Rucio and also to adding identities to rucio accounts.

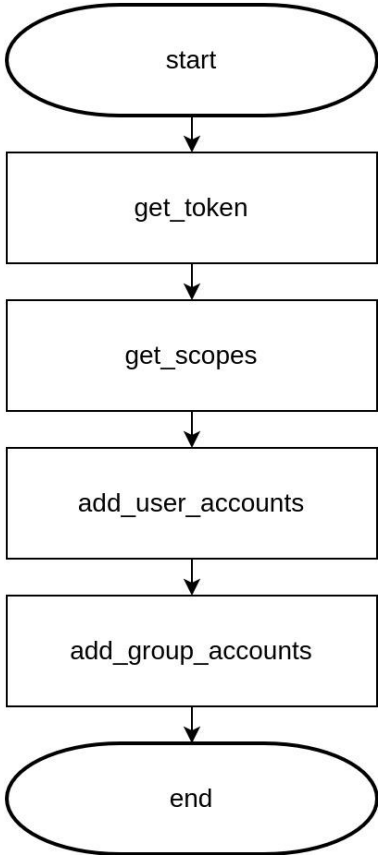
Implemented the addition of the OIDC identities for authentication in Rucio via SPD IAM and addition of the user certificates subject DN for authentication in rucio with usercert and proxy certificates. There is a functionality for adding a subject DN string in legacy format. Also, for each user will be added their standard user scope and a global limit.

Rucio Account Importer [2/3]

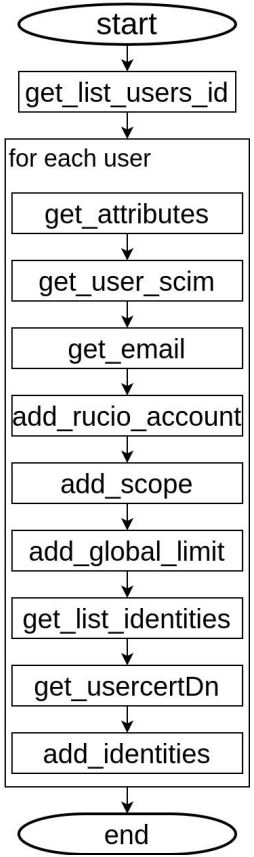
The developed utility interacts with SPD IAM using an access, issued to the client who performs actions to import accounts and identification information. This client is registered in SPD IAM with the scope iam.admin:read, scim:read.

The access token contains only these two scopes, which allow the client (in this case, the developed utility) to obtain information about users, their identification information, groups, etc. from the SPD IAM. The lifetime of the token is five minutes. During this time, all operations are performed to obtain information about SPD IAM users and add new information to Rucio. The token grant flow is client_credentials (client_id + client_secret).

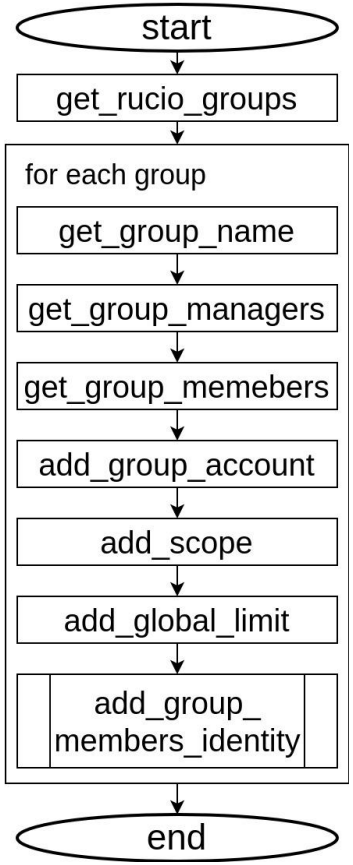
Rucio Account Importer [3/3]



(1) General algorithm

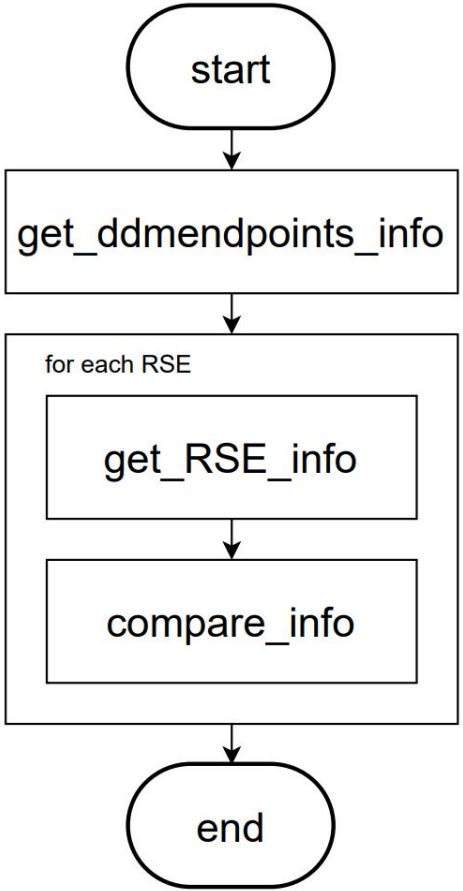


(2) add_user_accounts algorithm



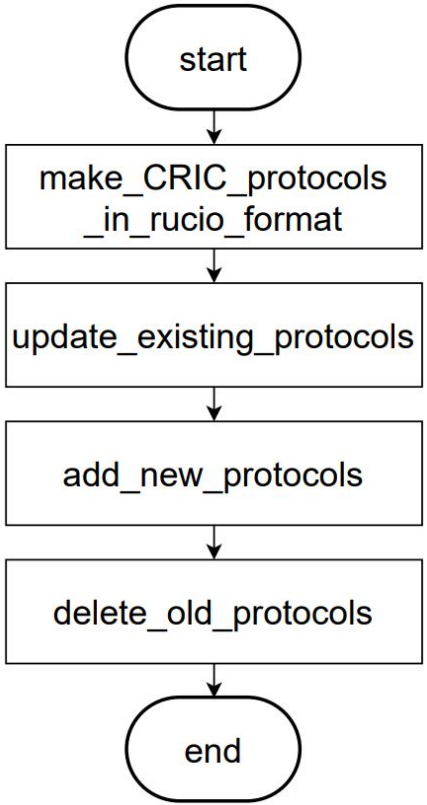
(3) add_group_accounts algorithm

CRIC integration [1/2]



1) Takes information from CRIC about all storage systems registered in it.
2) Requests protocols and attributes of RSE from Rucio.
3) Compares info (changes it if necessary):
   - checks attributes;
   - checks FTS;
   - checks protocols.

CRIC integration [2/2]



Rucio protocol description

```
{
 "domains": {
   "lan": {
     "delete": 0,
     "read": 0,
     "write": 0
   },
   "wan": {
     "delete": 0,
     "read": 0,
     "third_party_copy_read": 1,
     "third_party_copy_write": 1,
     "write": 0
   }
 },
 "extended_attributes": null,
 "hostname": "somehostname.jinr.ru",
 "impl": "rucio.rse.protocols.webdav.Default",
 "port": 8000,
 "prefix": "/eos/rucio/spd",
 "scheme": "https"
}
```

Hostname, **port** and **scheme** are key attributes of protocol in Rucio.

If any other attribute has been changed -> update_existing_protocol

If any key attribute has been changed -> add_new_protocol and delete_old_protocol

To be continued...