

# SPD data management

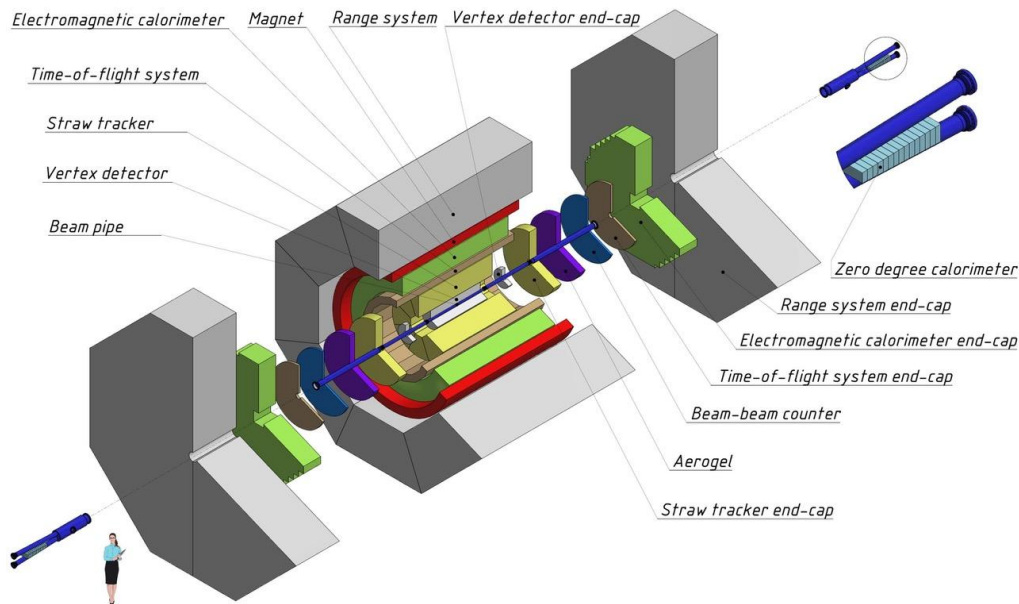
Alexey Konak  
JINR MLIT  
konak@jinr.ru

JINR AYSS Conference "Alushta-2025"  
11.06.2025

# 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.



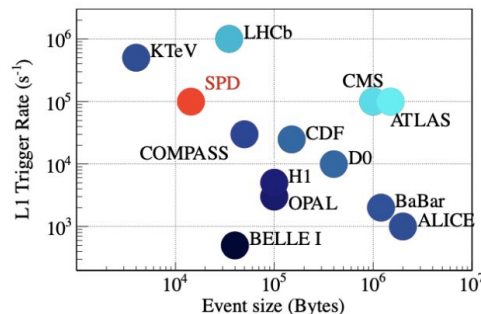
The SPD facility is being created for a more accurate study of the contribution of **gluons** to the spin of the **nucleon**.

# SPD as data source

- Data from the detector – 20 GB/s (or **200 PB/year** "raw" data,  $\sim 3 \cdot 10^{13}$  events/year)
- Simulation results – ??? (the exact volume is unknown, but it is comparable to the data volume 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...)

The expected event rate of the SPD experiment is about 3 MHz (pp collisions at  $\sqrt{s} = 27$  GeV and  $10^{32} \text{ cm}^{-2}\text{s}^{-1}$  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.

## About Rucio

Rucio is an open-source software framework that provides functionality for data management and access in a distributed storage environment. Rucio organizes a single namespace, realizing the possibility of data management, interacting with the entire storage space as one. Rucio also provides protection against data loss and speed up access to data through a controlled number of replicas.



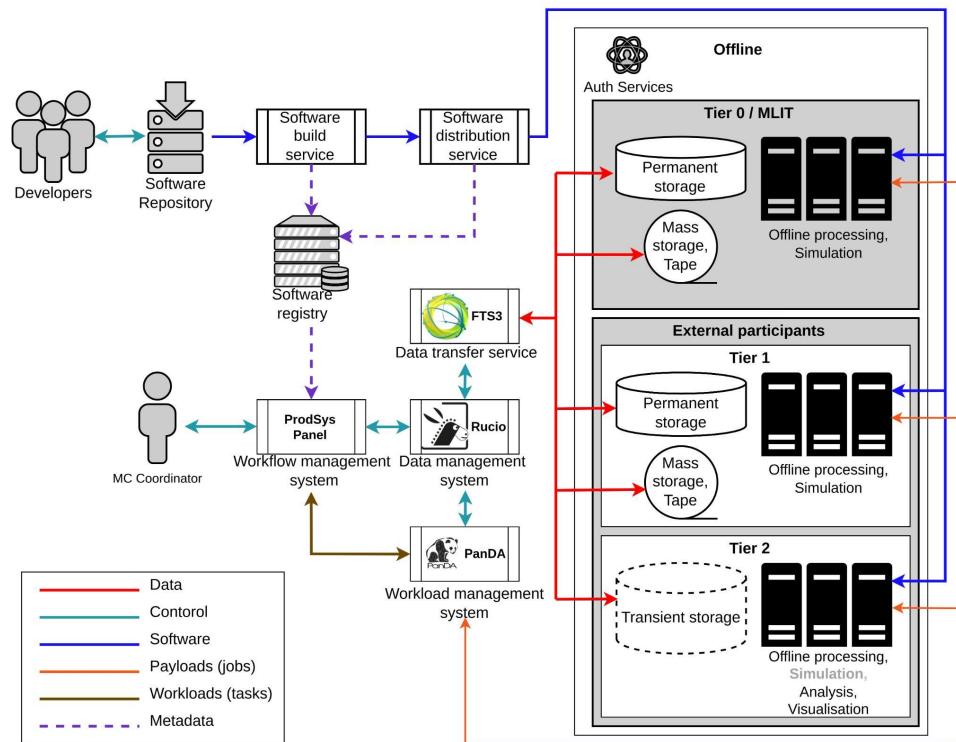
# Rucio functionality

1. data catalogue;
  - 1.1. organize data in a hierarchical structure for easy navigation and management;
  - 1.2. storage of any types of experimental data;
  - 1.3. storage and management of metadata;
2. data lifecycle management;
3. unified interaction of a heterogeneous network and storage infrastructure (a resource catalog and data transfer service are required);
  - 3.1. adaptive data replication and recovery;
  - 3.2. automated data transfer between storages;
4. Exhaustive monitoring metrics: data usage, system performance, services health.



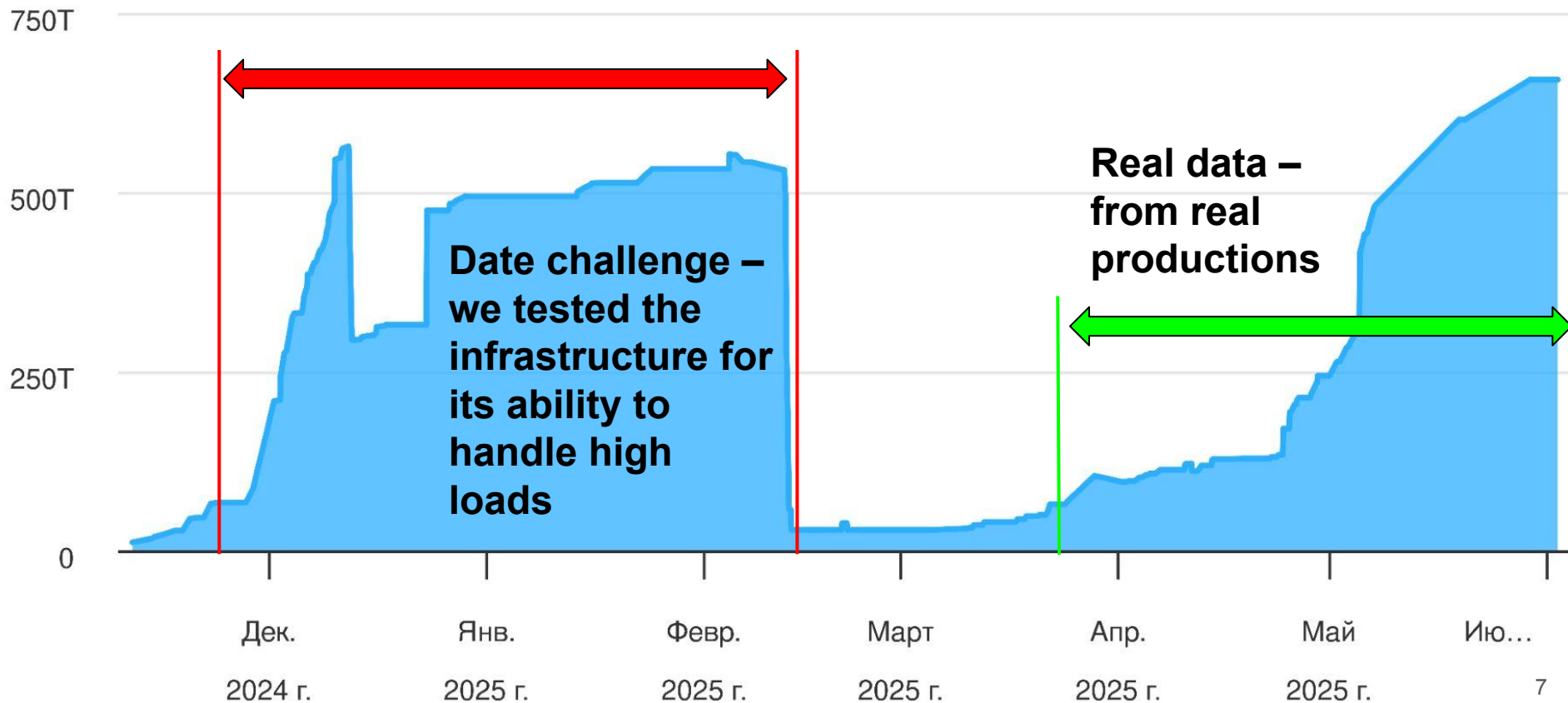
# Starting point

- deploying all necessary components of Rucio
- figure out how it should work ;-)
- general configuration
- integration and testing interaction with PanDA
- integration and testing interaction with FTS
- integration with IAM
- integration with CRIC
- initial service monitoring



# Data Management [1/2]

## SPD Data Overview

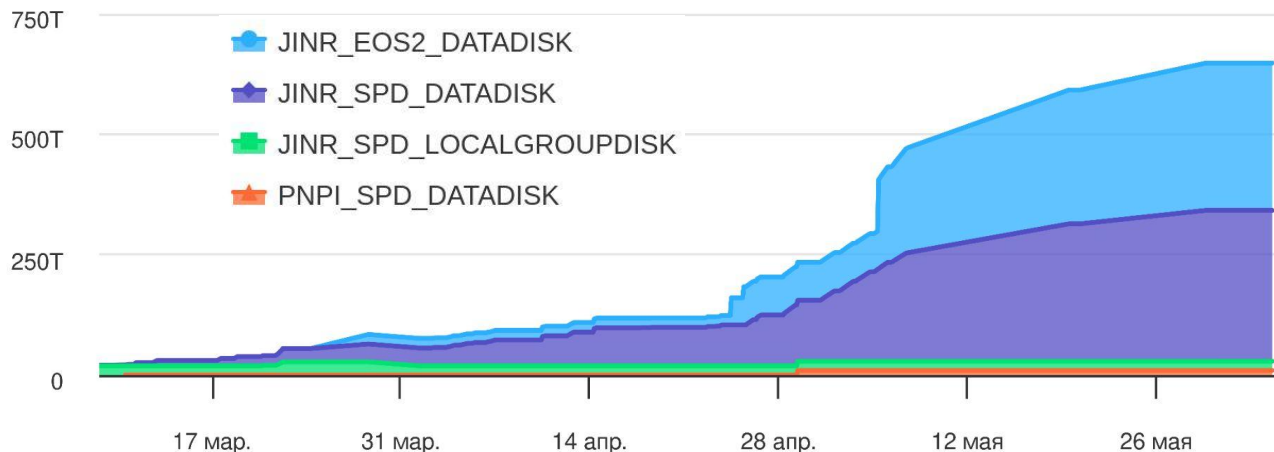
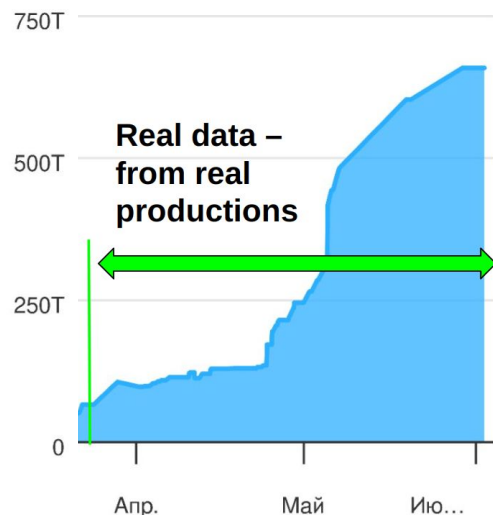


# Data Management [2/2]

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.

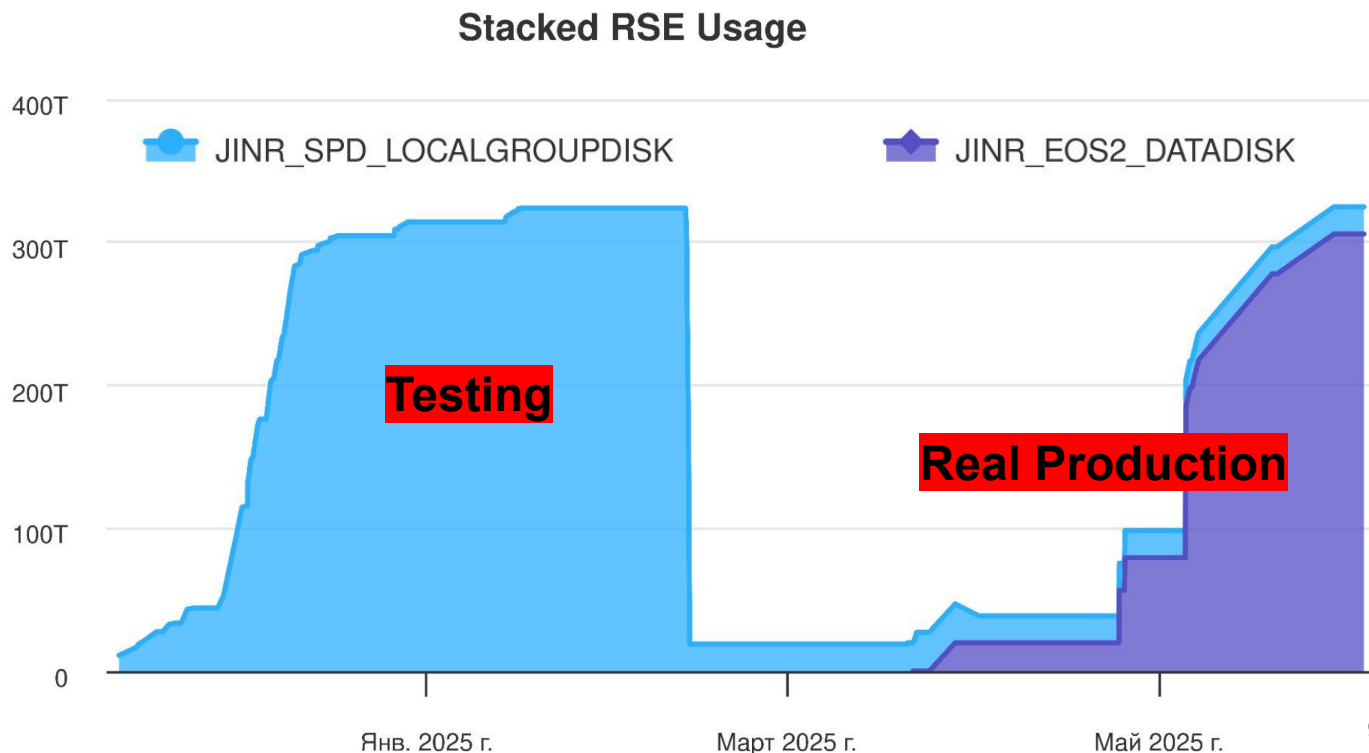
**Stacked RSE Usage**





# PanDA integration[1/2]

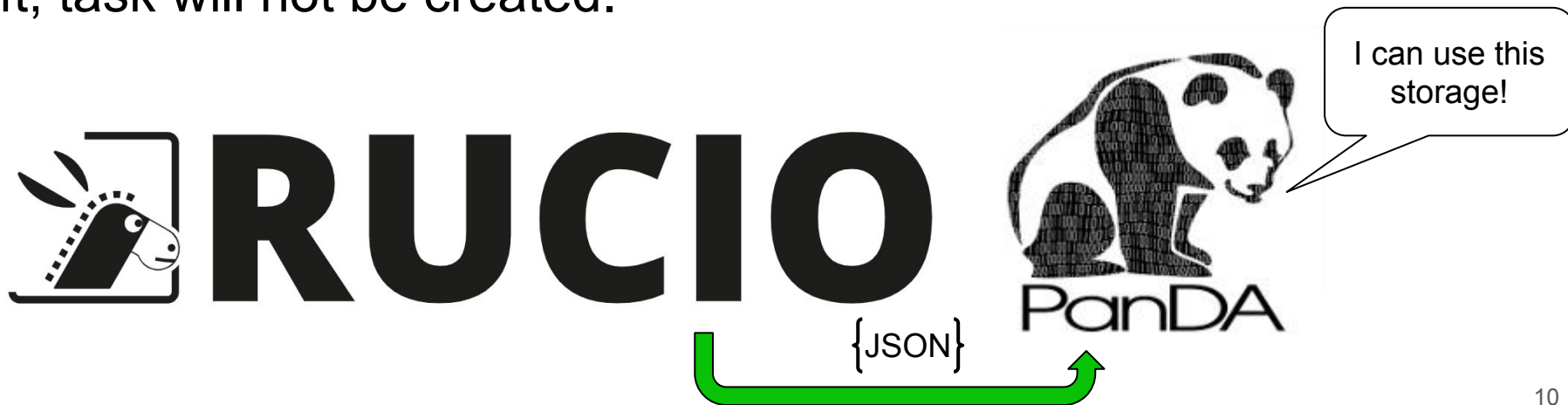
Currently, Rucio is used for mass production of SPD. During production, we tested interaction of PanDA and Rucio in various forms as well as different data organisation.



## PanDA integration[2/2]

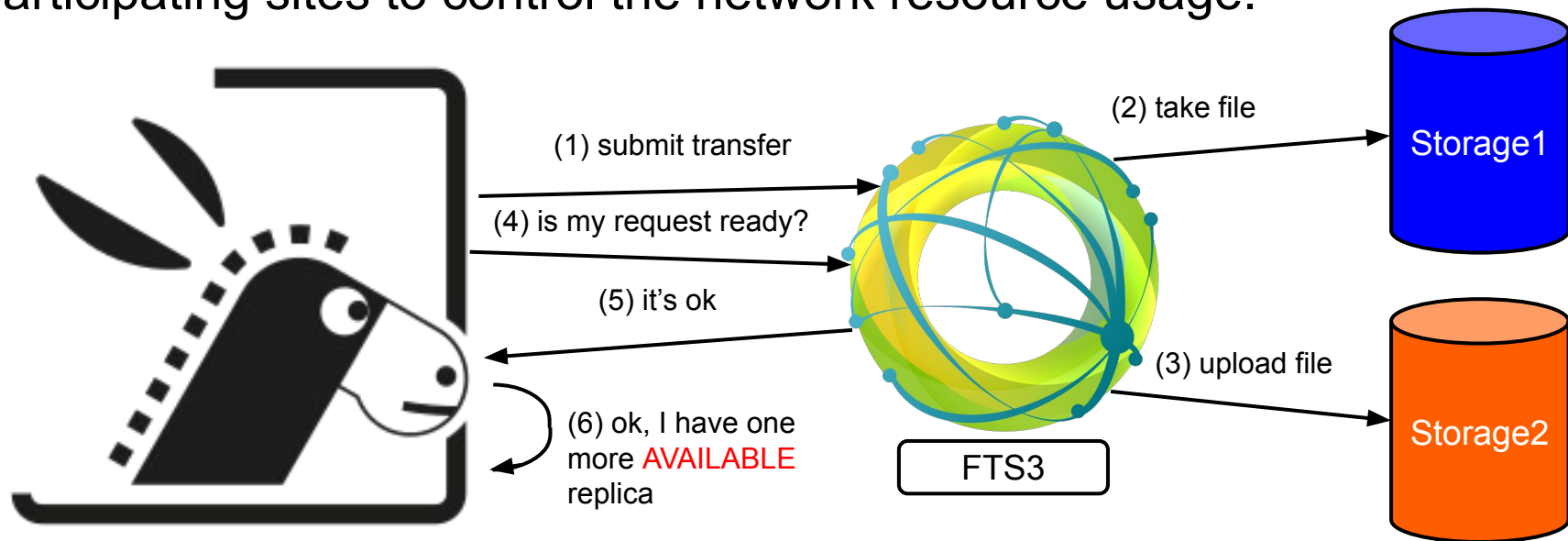
Special utility which informs PanDA about storages space usage was developed.

PanDA finds out about the storages space usage from Rucio. If the estimated task output size is larger than the storage space left, task will not be created.



# FTS3 integration

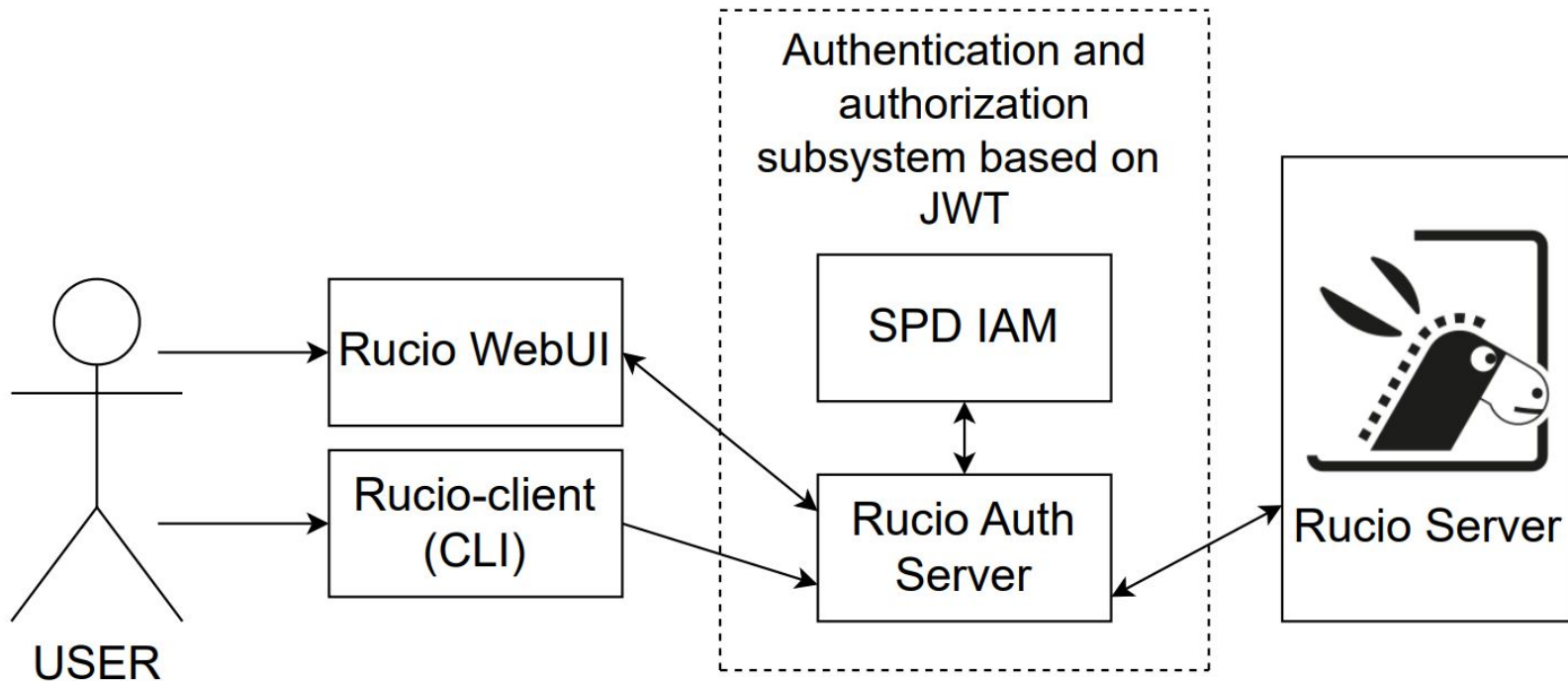
File Transfer Service 3 (FTS3) is the service responsible for globally distributing the majority of the LHC data across the WLCG infrastructure. Is a low level data movement service, responsible for reliable bulk transfer of files from one site to another while allowing participating sites to control the network resource usage.



# Integration with the SPD IAM [1/2]

Identity and Access Management (IAM) allows you to log in to a third-party application or system using an access token and an ID token obtained during authorization in IAM.

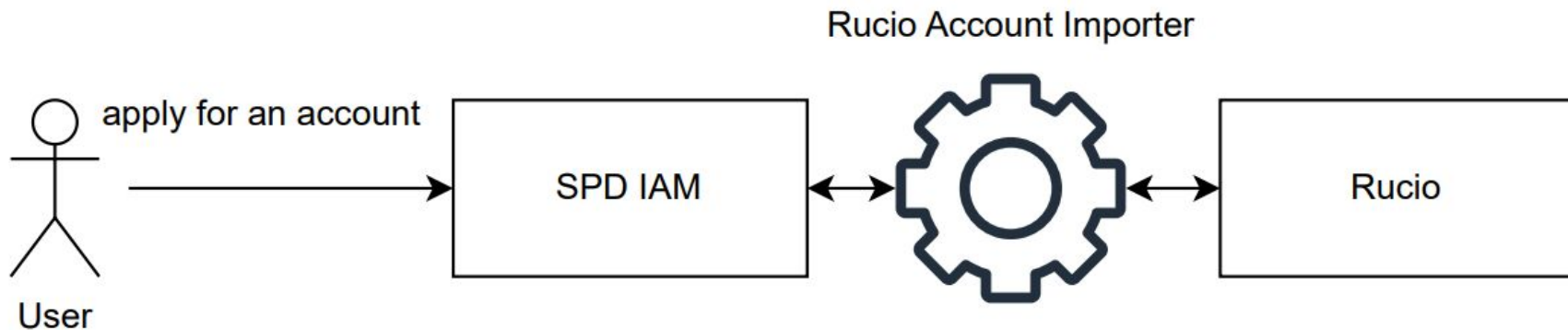
SPD IAM provides authentication to all services and systems (including Rucio). In the foreseeable future, each member of the collaboration will be registered there and will use this system to log in to other systems and applications (this will be used as a single entry point to the entire infrastructure).



# Integration with the SPD IAM [1/2]

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 [1/2]

The Computing Resource Information Catalog (CRIC) is a system designed to manage and provide information about computing resources used in distributed computing infrastructures.

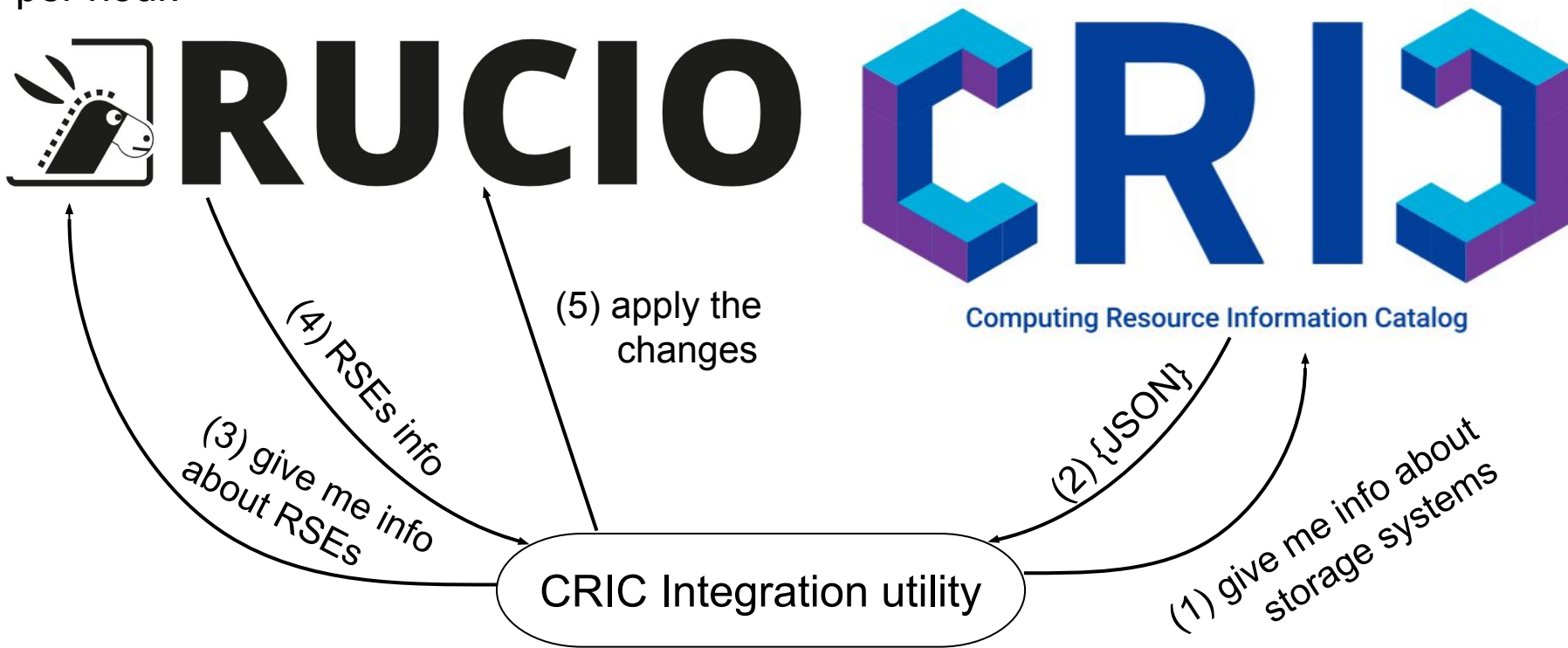
CRIC is integrated with Rucio to manage storage system configuration from a single location, by an administrator.



Computing Resource Information Catalog

## CRIC integration [2/2]

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.



## 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.

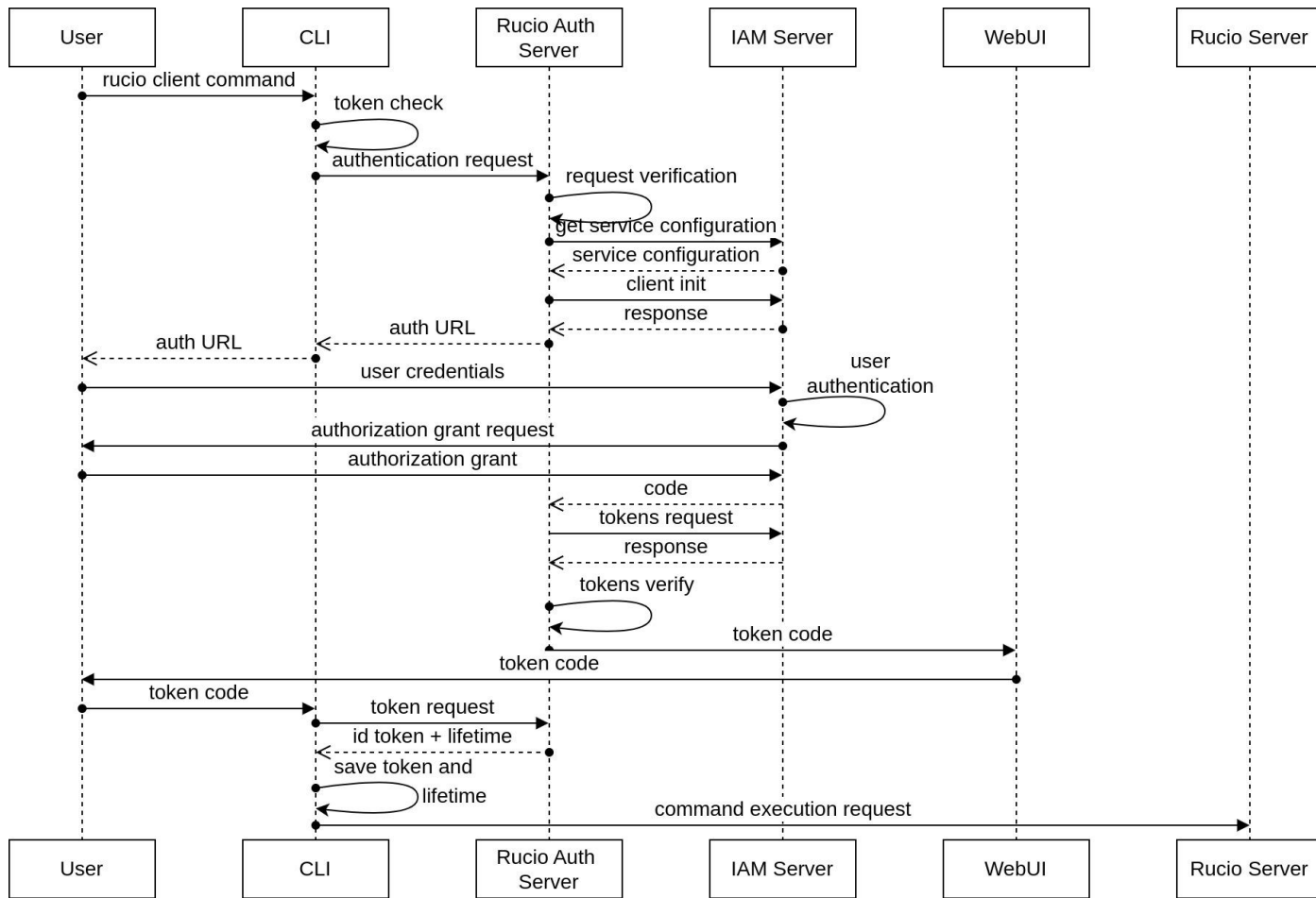
**RSE** – the logical abstraction of a storage system for physical files. It has a unique identifier and a set of meta attributes describing properties.

# Current status

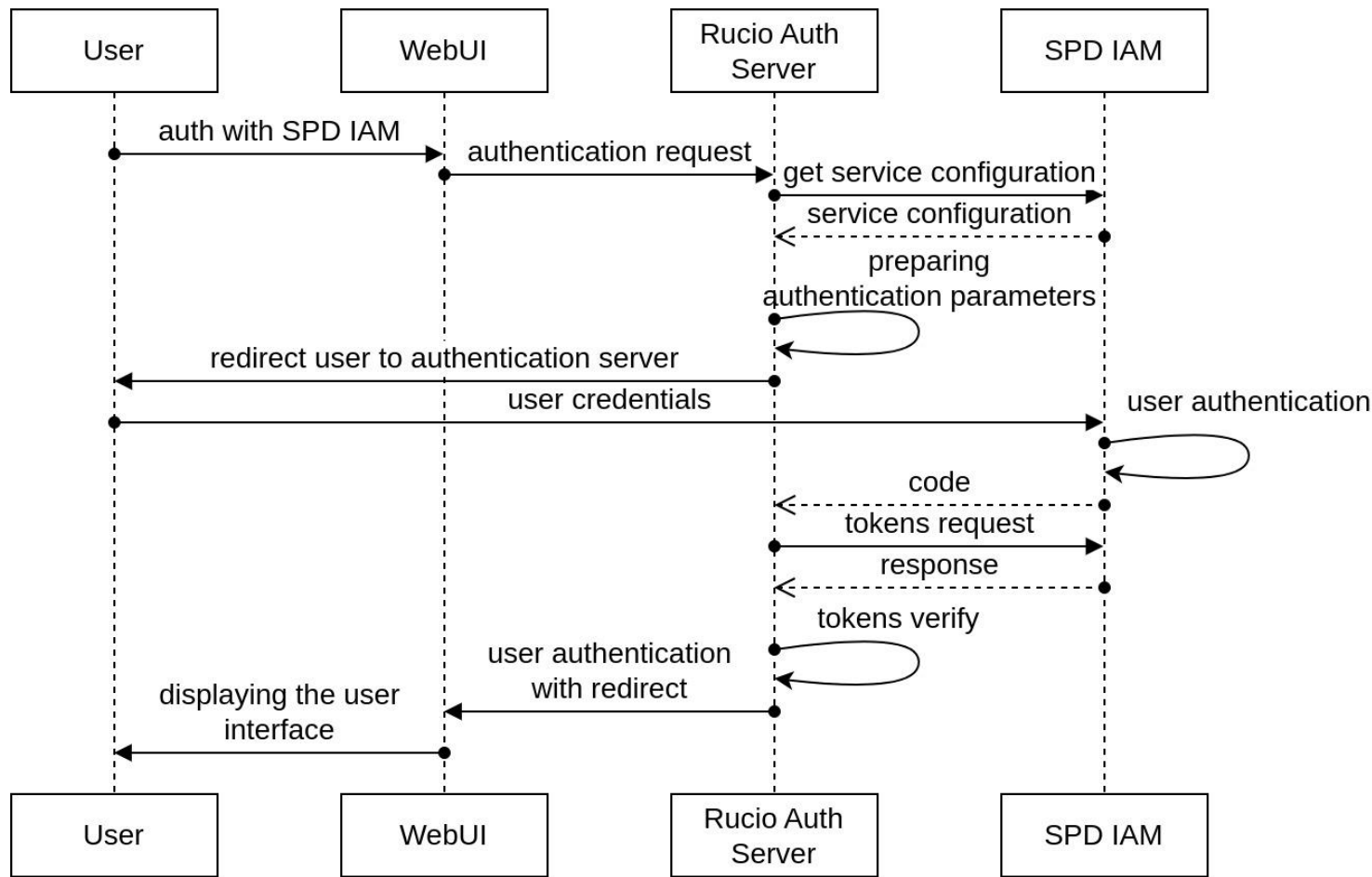
At the moment, the required set of system components of three Rucio-servers are deployed in Docker containers based on JINR cloud computing infrastructure:

- **Prod** Rucio-server. The main Rucio-server which work stable and used for the needs of the SPD collaboration.
- **Dev** Rucio-server. This server is used for development, testing and debugging.
- **Int** Rucio-server. All updates and innovations are checked and tested on this installation before being put on the Prod Rucio-server.

# 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**.

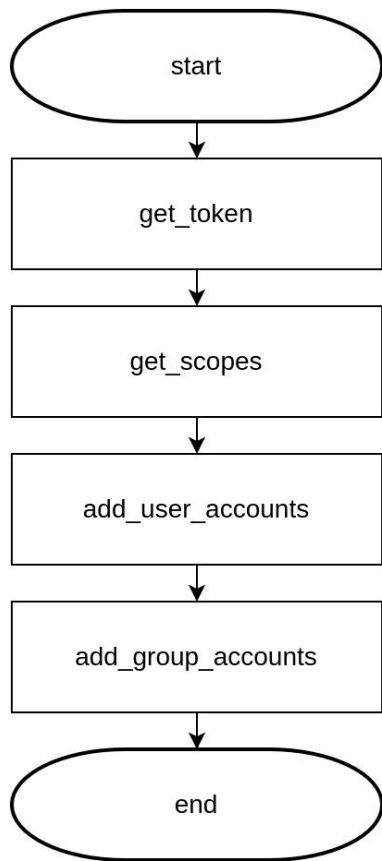
## Rucio Account Importer [2/3]

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**.

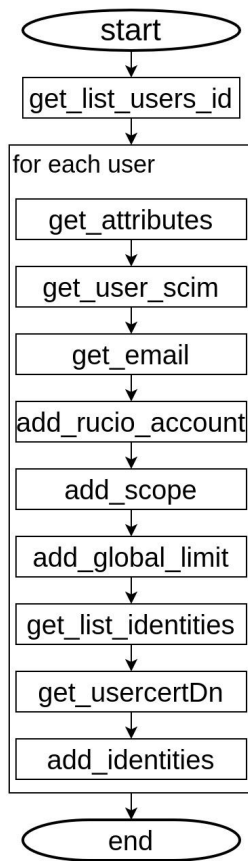
The access token contains only these two scopes, which allow the client (in this case, the developed utility) to obtain information about users, their identification information, groups, etc. from the SPD IAM. The **lifetime** of the token is **five minutes**. During this time, all operations are performed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



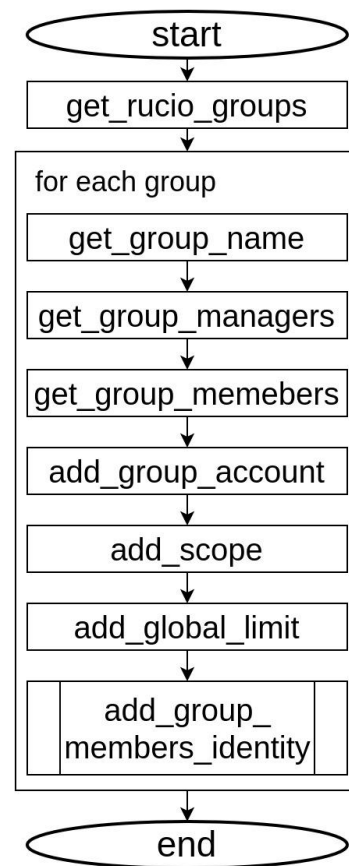
# Rucio Account Importer [3/3]



(1) General algorithm

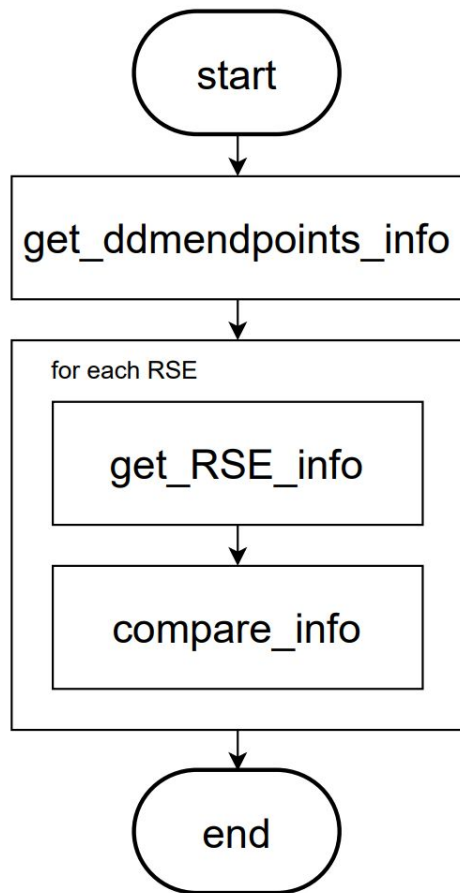


(2) add\_user\_accounts algorithm



(3) add\_group\_accounts algorithm 25

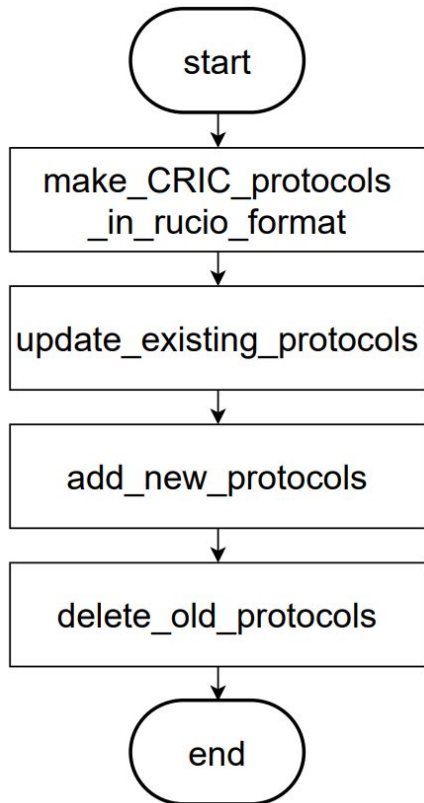
# CRIC integration [1/2]



- 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...

