Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

Storage federation in DataCloud

Integration roadmap between Spoke0 and Spoke8

A. Troja on behalf of DataCloud-WP6

25/06/2024

Meeting WP3

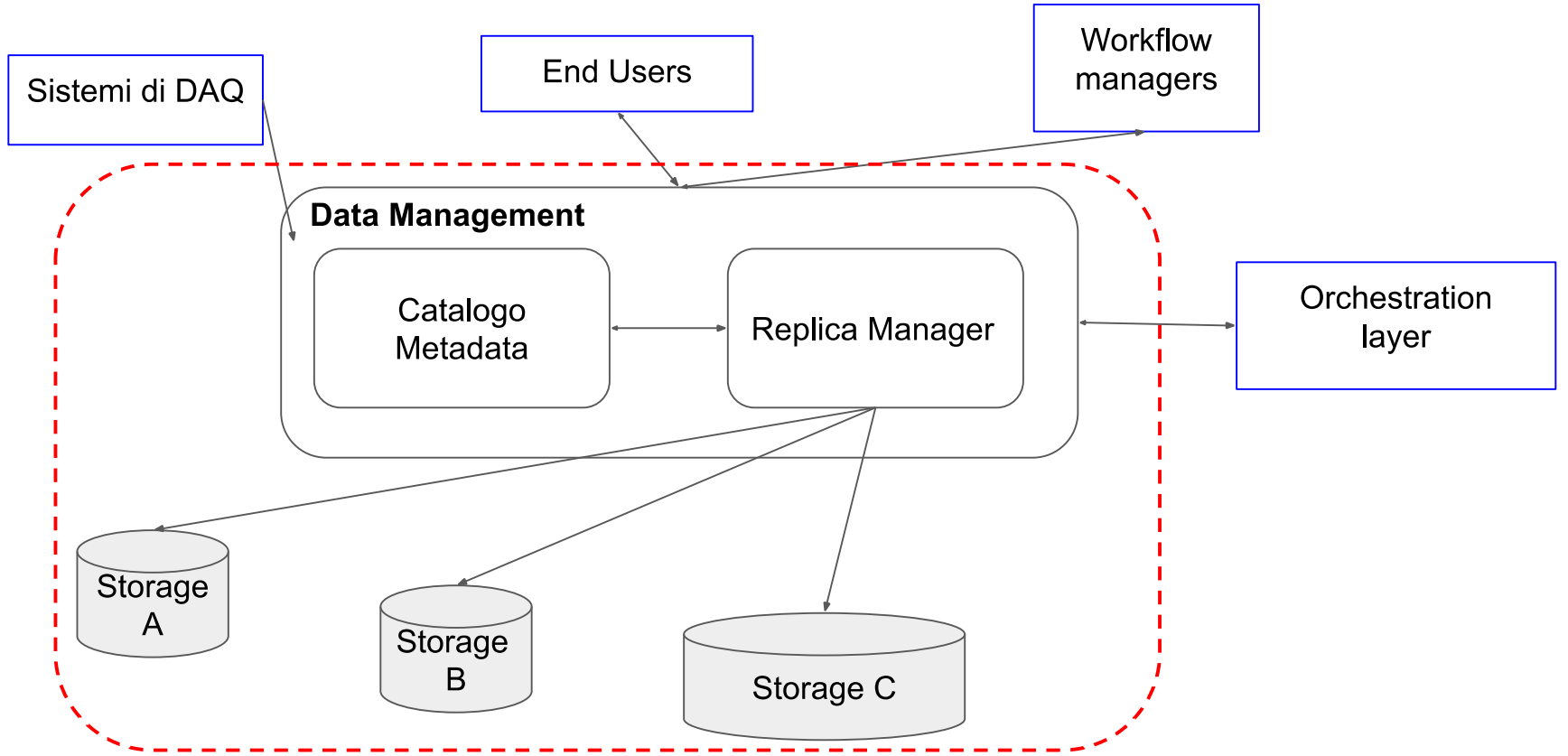
Outline

- The datalake: definition
- What we did: Data Management testbed
- Data Management and security
- Data Management Logging
- Early adopter example: Cygno experiment
- Final Remarks

What is our “datalake”

A datalake is a deployment where physical storages are federated no matter their implementation, and data are distributed among them.

The datalake allows users to access and manage data spread among different storage, by simply interacting with the Data Management (DM) layer.



Storage & Data Magement, M. Sgaravatto e D. Spiga, 2022

The replica manager

In a distributed storage system, the replica manager is the core component of the DM.

In fact:

- It decouples the logic level from the physical one, making the users interact with the former;
- It creates abstraction of the storage protocols, allowing the federation of different backends.

The replica manager:

- Manages file replicas on different storage systems;
- Tracks the file replicas physical location;
- Coordinates the file transfers among the storage systems;
- Carry out the policies defined on data.

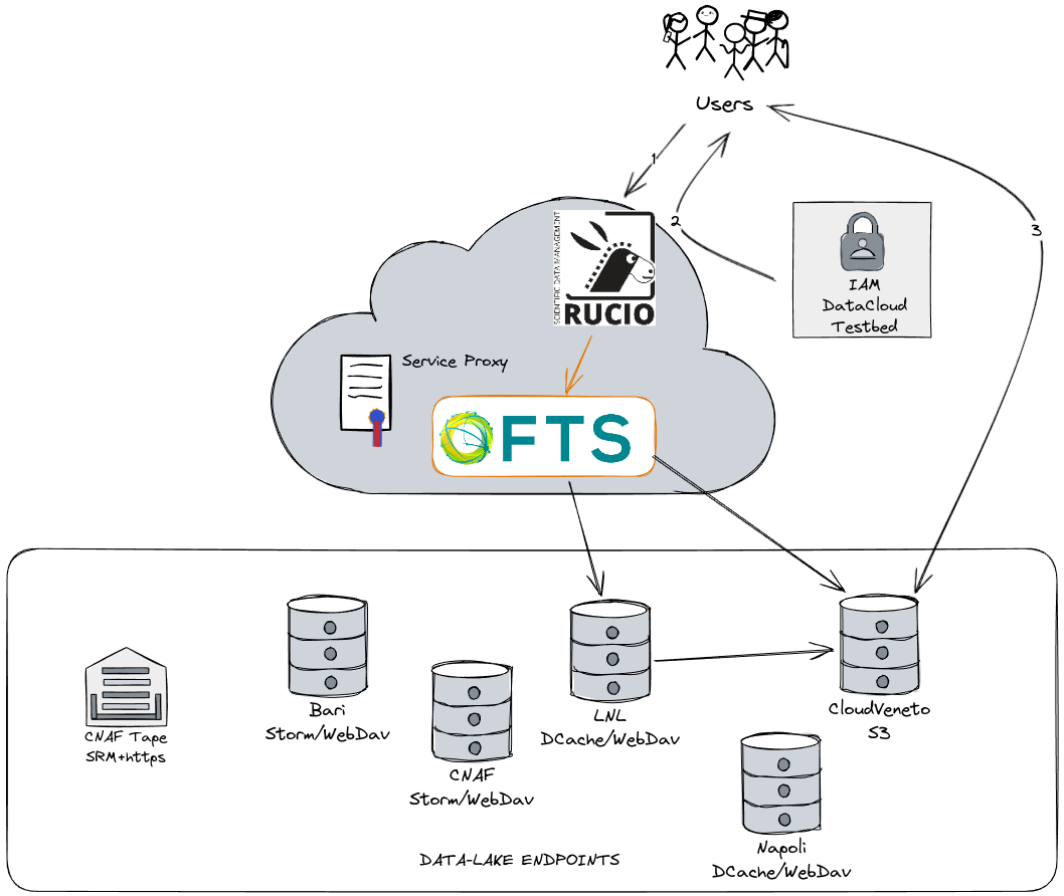
DM testbed in DataCloud WP6

We integrated tools *de facto* standard in scientific environments close to us (e.g., LHC, where the the size of data exceeds petabytes):

- **Rucio+FTS**: replica manager;
- **Indico IAM**: Handles authentication and authorization;
- **Metadata Catalog**: Embedded in RUCIO.

Federated 6 heterogeneous storage systems of INFN with different:

- **Technology** (disk, tape);
- **Implementation** (dCache, Storm, etc…);
- **Protocols** (webDav, SRM, S3).



Federare lo storage distribuito nazionale, D. Ciangottini, 2023

What does these tools offer

Infrastructure perspective:

- Different **storage systems** are federated regardless of their geographical location, implementation, or quality of service (QoS, whether disk or tape).

User perspective:

- Users interact with data **declaratively**. For example, they can specify how many replicas a certain file needs, on which storage systems, and for how long they should exist;
- **Data** can be organized hierarchically (using datasets and containers);
- **Metadata** management enables query functions;
- **Transparency**: This level of abstraction shields users from heterogeneous and implementation-specific details.

Authentication and Authorization

The user interacts with RUCIO via token, provided by IAM:

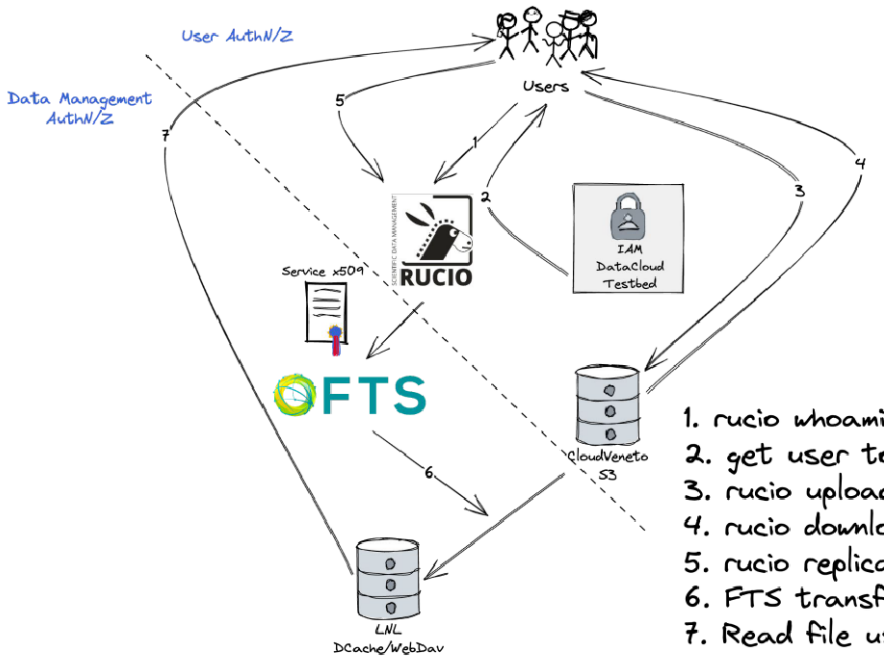
- 2FA can be implemented within the Identity Provider used to log into IAM
- Support needed in IAM to have 2FA also for non-IDP users

The token contains all the information about the user needed for authentication.

Once the user is authenticated, Rucio will check its authorization prior to any operation on data.

If authorized, the user can create replicas of data among storages via Third-Party Copy (TPC) with FTS, managed by Rucio.

- Rucio (not user) is authorized with a VOMS service proxy.

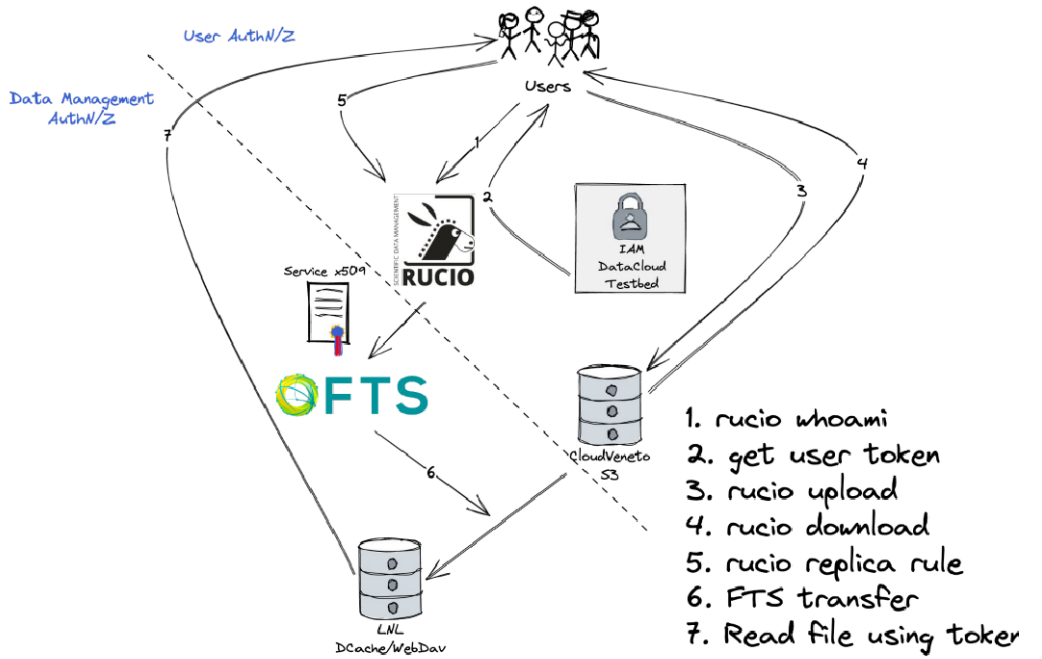


Federare lo storage distribuito nazionale, D. Ciangottini, 2023

Authentication and Authorization

We tested various authorization models on the data:

- **Group-based policies**: group A users won't be allowed to interact data from group B, while it can perform any action on group A data (even deletion);
- **Scope-based policies**: scopes are defined within the IAM, and passed to Rucio through the token. Scopes give user permission to certain action on certain area in the storages. As an example, we implemented *user isolation* (e.g., each user can read everything but only write in his/her own designated area).



Federare lo storage distribuito nazionale, D. Ciangottini, 2023

Rucio allows the implementation of highly customizable policies. That is especially useful when dealing with confidential data (for example, medical data).

Logging

Each Rucio component provides a log with information on what it does.

The Rucio policy framework can be customized to properly log each operation on data according to the specific needs.

For example:

- user X asked Rucio to replicate file Y from storage A to storage B;
- User XX deleted file YY from storage A or from the datalake;
- Etc etc...

Also FTS logs each transfer, but in this case the owner of the transfer is the service account used by RUCIO.

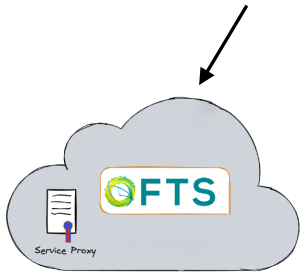
When an application interacts directly with the storage system it is up to the storage system admin to properly configure the logging on the storage.

DM for user communities

The scenario we imagined for DataCloud is:

- Rucio deployed specifically for a user community;
- IAM specific for a user community,
  it manages users and policies;
- Both Rucio and IAM managed by the community;
- Multi-tenant FTS (centrally managed);
- Federation of storage *pledged* to the community,
  no matter the size.

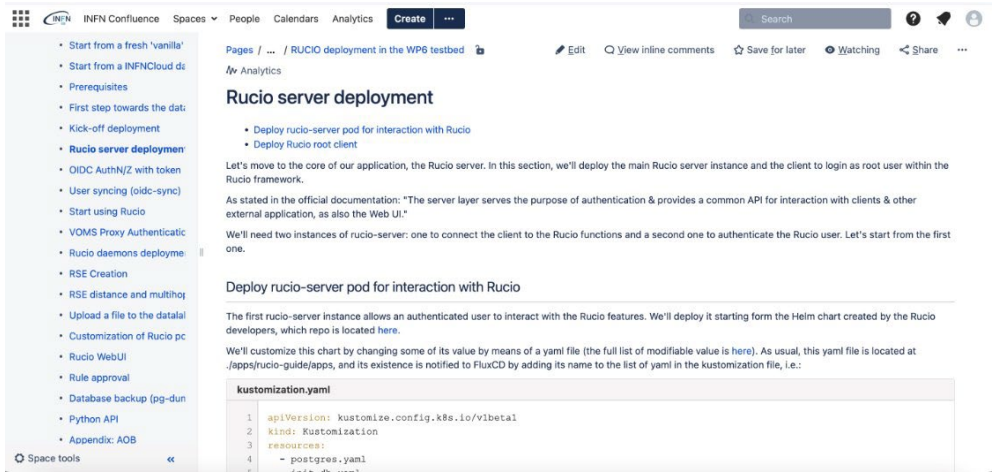
This is not the only possible scenario. For example, it makes sense having a **dedicated FTS** when dealing with confidential data.

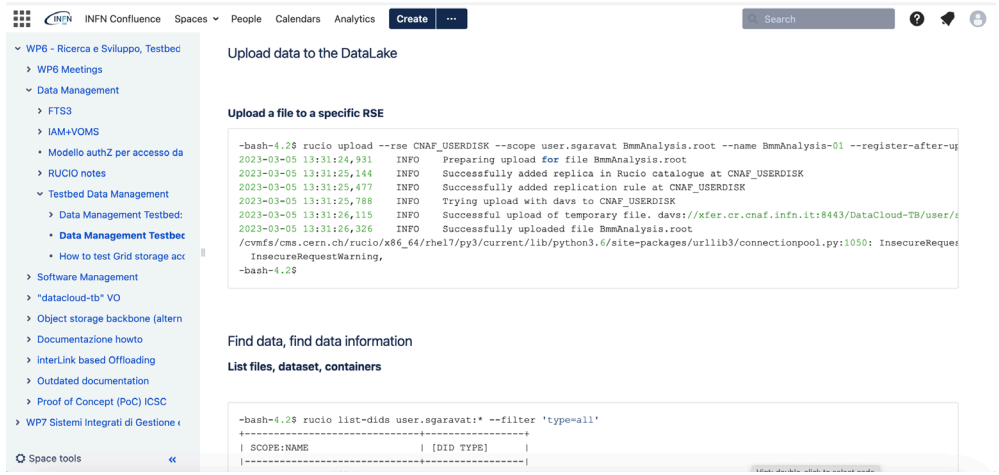


Documentation

Since official documentation is rather limited, we have produced some documentation for both users (here) and operators (here).

This documentation allows any user to replicate the DM system anytime is required, without major effort.

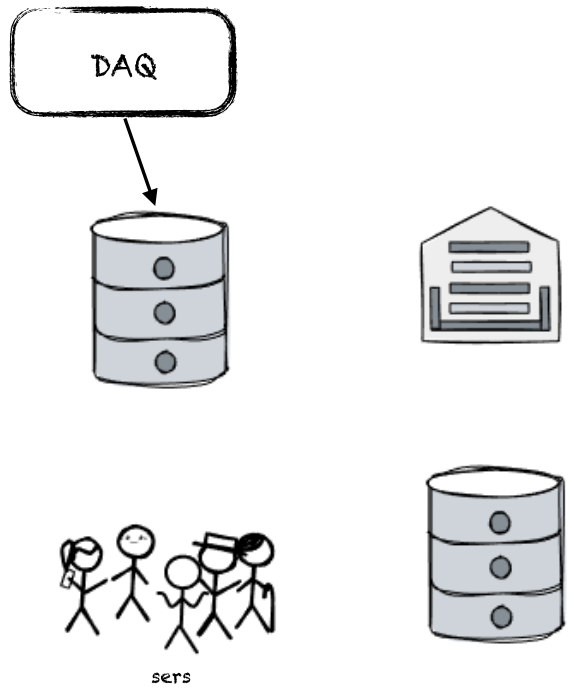




Early adopter example: Cygno experiment

Cygno is an INFN experiment on dark matter.

- Still in its preliminary stage, but it already produces data.



The data is initially loaded onto an S3 disk. From there, it is manually copied onto tape storage. A third S3 disk storage is available.

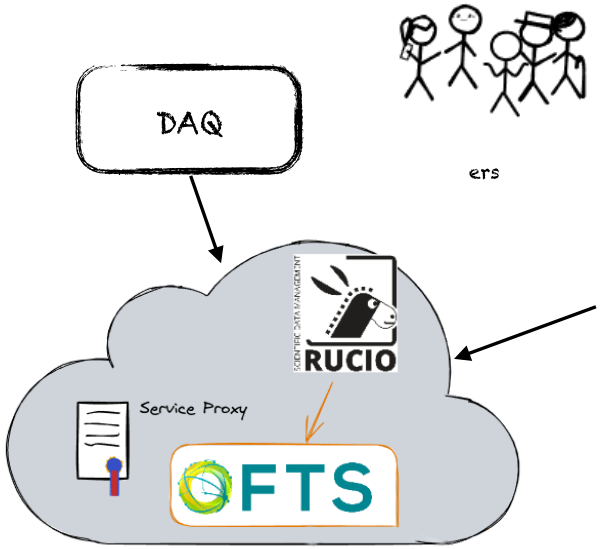
After processing, the data will be deleted from any disk storage. Thus, data will be stored only on tape after analysis.

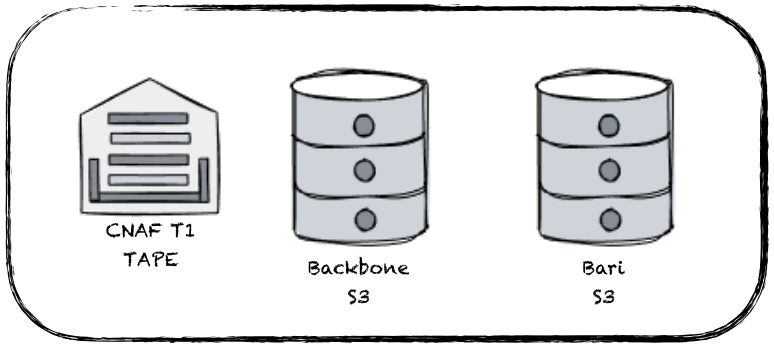
This continous process of copy and delete data among different storages provides the perfect case study for the data management system (DM).

Cygno datalake

We have implemented a datalake that:

- Federates the two disk storage systems and the tape storage;
- Implements ad hoc policies (**more stringent** than those of our testbed), particularly by introducing an authorization layer for specific rules (transfers to/from tape).
    - Through a WebUI, appropriately authorized personnel can approve or reject replication requests.





Final remarks

- Spoke0 and DataCloud provided a system to manage **large amounts of data in a distributed system**, aiming to the development of a Datalake.
- Everything we showed today is **production ready.**
- We're already supporting adopters communities (that acted also as beta tester).
- In-depth documentation helps new adopters to start easily, with a fast pace.

This should enable the deployment of the **DM tools on the EPIC platform**.

Grazie per l'attenzione

antonino.troja@pd.infn.it