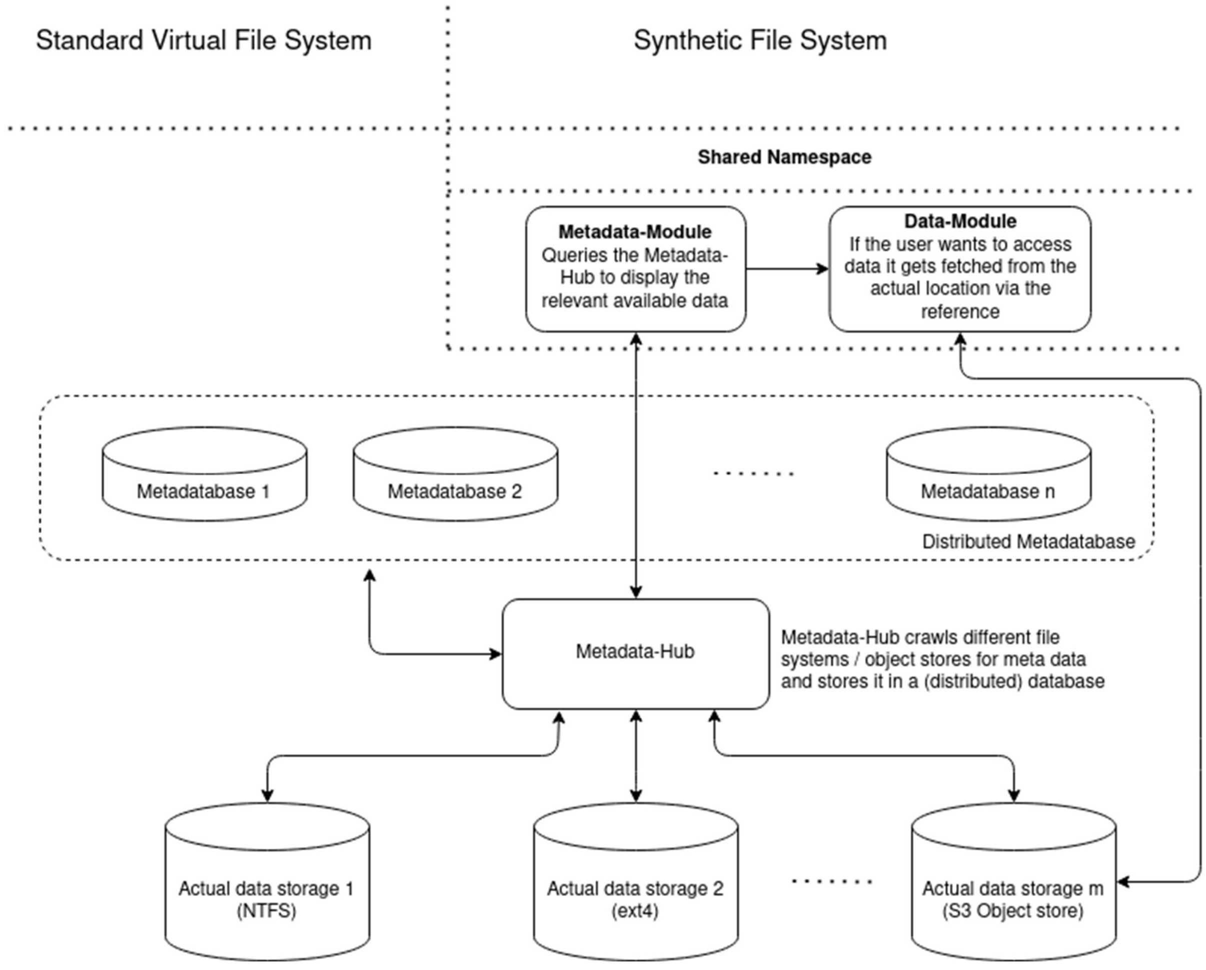
Software Architecture Description

Run-time Components


Standard Virtual File System
Synthetic File System
Shared Namespace
Metadata-Module
Queries the Metadata-Hub to display the relevant available data
Data-Module
If the user wants to access data it gets fetched from the actual location via the reference
Metadatabase 1
Metadatabase 2
....
Metadatabase n
Distributed Metadatabase
Metadata-Hub
Metadata-Hub crawls different file systems / object stores for meta data and stores it in a (distributed) database
Actual data storage 1 (NTFS)
Actual data storage 2 (ext4)
....
Actual data storage m (S3 Object store)

The Synthetic File System can be browsed like a normal file system. For example a command like “ls” only involves metadata so the Metadata-Module queries the relevant data from the Metadata-Hub and displays it in regular files and folder. If the user wants to access such data for example with “open” the Data-Module fetches the data from the data storage.
Even though the data is stored in different file systems and object stores, they get exposed to the user in a unified way. Which means that the Synthetic File System displays the different data formats under a shared / common namespace.

Code components

At this stage, the software is not yet developed to the point where the code components can be determined. This will be submitted as soon as possible to give an overview of the code components.

<u>Technology stack</u>

To be independent of a specific platform and to make sure that every team member has the same run-time environment, we decided to use docker.

The choice of programming language is not fixed, but since the team members have different backgrounds and the focus from the industry partners is prototyping, we decided on python.

PostgreSQL and GraphQL are chosen to be in synch with the Metadata-Hub, which is an integral part the service.

It was clear that we not only wanted to integrate automated tests, but also to use these to guide the development (TDD).
While pytest is more popular than unittest, we use unittest, because most of the time it is the first testing framework someone learns when learning python.

Development (internal)	
Version Control	Git
Source Code Hosting	GitHub
Testing Framework	Unittest (Python)
Deployment	Docker
Product / Prototype	
Programming language	Python
Databases	PostgreSQL
DB-Query language	GraphQL
Metadata generation	Metadata-Hub