Synthetic File System
Technical Documentation
Technology Stack

To be independent of a specific platform and to ensure that every team member has the same runtime environment, we decided to use docker.

The choice of programming language is python, because it has simplified syntax and is easy to learn and use. Furthermore, the focus of the project from the industry partner is to prototype the synthetic file system. Therefore, python is the most suitable language to achieve this.

PostgreSQL and GraphQL are integral parts of Metadata-Hub's service. Because the synthetic file system must be synchronized with the Metadata-Hub; PostgreSQL and GraphQL are chosen for storing and retrieving metadata.

It was clear that we not only want to integrate automated tests, but also use them to guide development (Test Driven Development). Even though pytest is more popular than unittest, we chose unittest, because it is usually the first testing framework someone learns when he or she starts learning python.

A summary of the mentioned technologies used to develop the synthetic file system can be found in the table.

List of relevant technologies

Development (internal):
Version Control: Git
Source Code Hosting: GitHub
Testing Framework: Unittest (Python)

Deployment: Docker

Product/Prototype:
Programming language: Python
Databases: PostgreSQL, GraphQL
Metadata generation: Metadata-Hub

Software architecture

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 folders. 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.

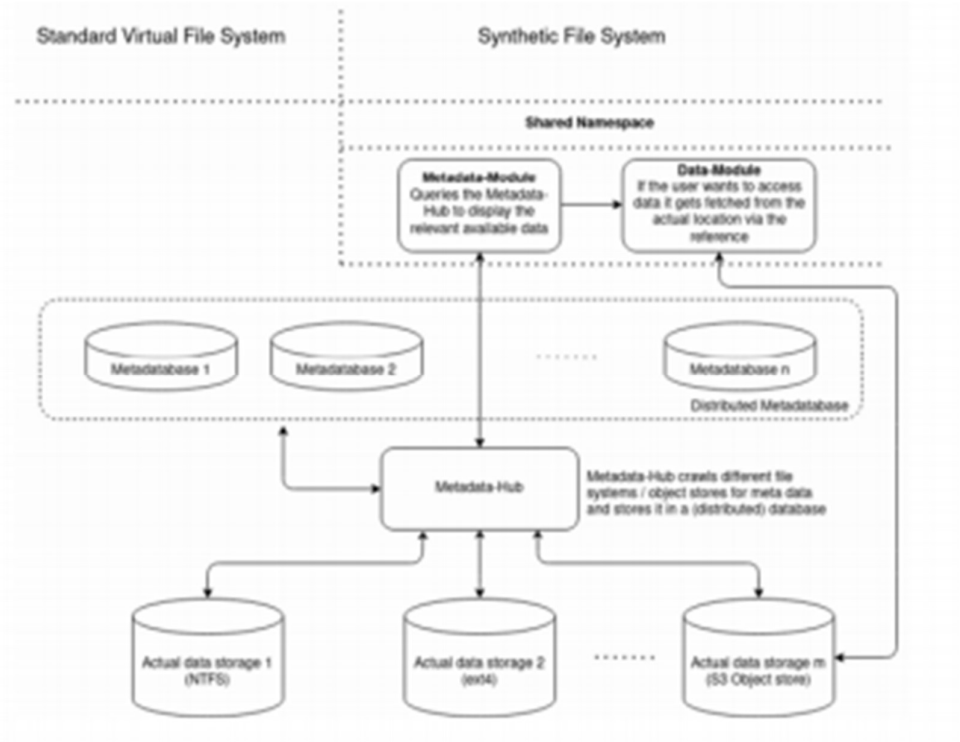

Figure 1: An overview of runtime-components