

This documentation explains the architecture of the OpenSearch-Metadata-Hub.

## **Runtime architecture**

[IMAGE: Runtime\_Architecture drawio]

### **MetaDataHub**

This is a platform build by the company GrauData to search and find data by using specific MetaData.

### **MDH-Database**

The MetaDataHub uses a PostgreSQL database to store documents with various Meta-data-tags. This data can be requested.

### **OpenSearch-MDH**

#### **OpenSearch Node**

This Docker container contains one so called "OpenSearch Node". This node allows to systematically storage data by indexing it. Each data (JSON-document) gets an unique ID and belongs to a predefined Index.

#### **OpenSearch Dashboard**

Next to the "OpenSearch Node" an additional dashboard as a administrative tool runs in a Docker container. This Dashboard has an interface to the "OpenSearch Node" Docker container and can be accessed via a Web-Browser.

#### **Application**

The main application runs in a third Docker container called "Application". This container consists of two components.

#### **Data-Import**

The "Data-Import" component functions as an import script to automatically import data from the MetaDataHub into the "OpenSearch Node". Via a Python SDK and a GraphQL query various data can be requested from the MetaDataHub and then be uploaded into the OpenSearch Node that runs in the other Docker container. This is done by using a Python library called "opensearch-py".

#### **Web-Server**

The "Web-Server" is the second component of the Application Docker container and builds the front-end interface between the user and the data stored in the "OpenSearch Node". The server runs via the Python framework "Flask". Additionally some JavaScript code is used to make the front-end dynamic. Via the Python library "opensearch-py" and "Flask" information about the data can be transferred to the Web-Server and then be displayed on a Web-Browser