EOSC Technical Specification

Common Services

Data Discovery and Access

Version:	1
Status:	EOSC-hub Proposal
Dissemination Level:	Public
Document Link:	

Abstract
The EOSC Data Discovery and Access service comprises the ability for end-users to search for data resources and access the referenced data.

OSC-HUB RECEIVES FUNDING FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME UNDER RANT AGREEMENT NO. 777536.

COPYRIGHT NOTICE

DELIVERY SLIP

Date	***Name***	***Partner/Activity***
From:	Heinrich Widmann	DKRZ
Reviewed by:	Bartosz Kryza Sara Ramezani	CYFRONET SURFSARA

DOCUMENT LOG

Issue	***Date***	***Comment***	***Author***
v.1	24/01/2020	First release ready for public consultation	Heinrich Widmann (DKRZ)

TERMINOLOGY

https://wiki.eosc-hub.eu/display/EOSC/EOSC-hub+Glossary

Terminology/Acronym	***Definition***

Contents

1 Introduction

Metadata Cataloguing and Indexing comprises the entire metadata ingestion workflow, i.e.

1. Metadata harvesting from community repositories
2. Metadata mapping on common schema including curation and validation and
3. Uploading and indexing of metadata records in the metadata catalogue, to enable Data Discovery and Access , see related macro feature 'Metadata Cataloguing and Indexing'

2 High-level Service Architecture

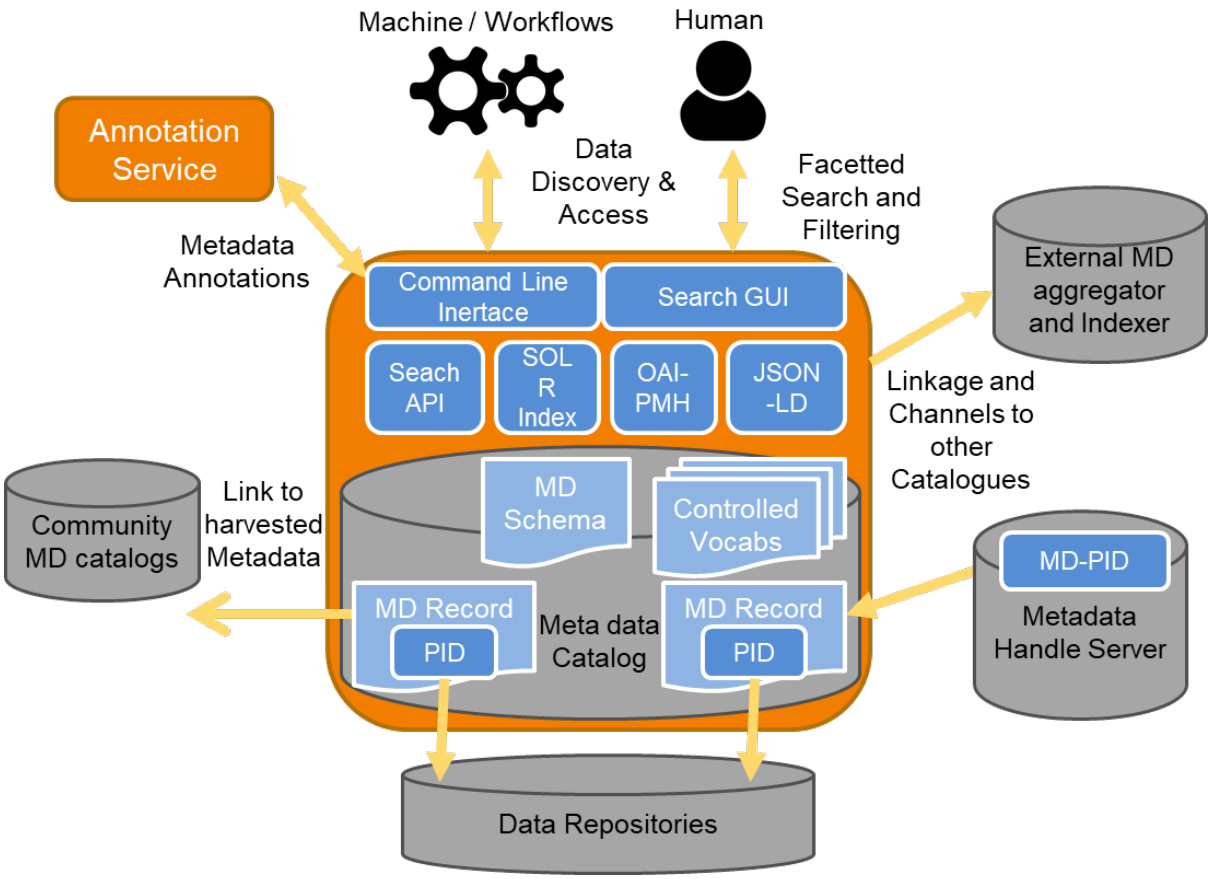

Figure 1: High-level Architecture of the EOSC Data Discovery and Access Service

The technical implementation of a data discovery and access service enabling searching for and identifying digital data should comprise the following components:

1. A discovery portal with an intuitive Graphical User Interface with faceted search and filtering options.
2. Command Line Interface allowing embedding discovery in a data processing workflow and machine readability.

3. A RESTful Search API with functionalities to identify referenced data collections by persistent identifiers
4. A search indexer and search index of a comprehensive metadata catalogue (see macro feature 'MD cataloguing')

3 Adopted standards

Standard	Short description	References
DataCite Metadata Schema 4.1.	Common and widely-used Metadata Schema, on which as well the B2FIND metadata scema and faceted search is based on	https://schema.datacite.org/meta/kernel-4.1/ http://b2find.eudat.eu/guidelines/mapping.html#b2fmdschema
ISO 639-1 codes	ISO 639 is a standardized nomenclature used to classify the search facet 'Language'	https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
B2FIND classification for Disciplines (not yet standardized)	Taxonomy for the central B2FIND facet *Discipline*, which specifies the research discipline the data belongs to. This allows filtering and selecting of datasets according to a multi-level discipline hierarchy	https://cryptpad.fr/pad/#/1/edit/KDecbjauKCtZclOmZAbbWg/L4aEiGrzJlSbRSXrFutOb0Cd/
ElasticSearch	Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.	https://www.elastic.co/products/elasticsearch
CKAN-API	CKAN's Action API is a powerful, RPC-style API that exposes all of CKAN's core features to API clients.	https://docs.ckan.org/en/ckan-2.7.3/api/
SOLR	SOLR is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. SOLR powers the search and navigation features of many of the world's largest internet sites.	https://lucene.apache.org/solr/

4 Interoperability guidelines

General, how researchers can search for data via the GUI or the CLI is explained in a detailed search guide of the discovery services, e.g. search guides of DataCite (https://support.datacite.org/docs/datacite-search-user-documentation) or of EUDAT-B2FIND (see http://b2find.eudat.eu/help/searchguide.html). Often the CLI of discovery services are used to perform the first step of complex processing workflows, usually starting with identifying datasets, which serve as input for following data transfer, processing and storing tasks, executed by other services. Interoperability guidelines should show how the discovery workflow step can be integrated in such a processing chains. E.g. the CLI for B2FIND is implemented as python script (retrievable at https://github.com/EUDAT-B2FIND/md-ingestion/blob/master/searchB2FIND.py).

In particular OpenAire and B2FIND follows – as all EOSC services – the FAIR principles to promote interoperability in research data management.

5 Examples of solutions implementing this specification

While there are countless domain-specific search portals and also many interdisciplinary discovery services, we will mention just two examples of cross domain services here:

- Google Dataset Search (https://toolbox.google.com/datasetsearch) allows users to find records stored on the Web using a simple keyword search. The tool can find information about records hosted in thousands of repositories across the Web. This makes these records generally accessible and usable. On the other hand, google datasearch offers only a limited amount of metadata and does not follow the FAIR principles in means of free and open access to the data.
- EUDAT-B2FIND (http://b2find.eudat.eu/) is a cross-domain discovery service based on metadata steadily harvested from research data collections from EUDAT data centres and other repositories covering all possible scientific fields . The service offers faceted browsing and it also allows, in particular, to filter via the facet ‘Discipline’ discovering data that is stored through the B2SAFE and B2SHARE services. The B2FIND service includes rich and validated metadata that is harvested from many different community and domain specific repositories. Within EOSC-hub EUDTA-B2FIND is intended to get the central search index for research date within and beyond EOSC-hub. Into this context fall the activities ‘Integration of B2FIND with B2SAFE and EGI DataHub’ already mentioned in ‘Metadata Cataloging and Indexing’

5.1 Procedure to integrate a service with the EOSC Hub Metadata Cataloguing and Indexing

The usual method to integrate discovery and access of data within other services is to add this function in a processing chain, which use other services. For example, using B2FIND, to search and identify datasets, which serve as input for services further down in the chain. The first workflow

step ‘Discovery of input data’ can be implemented as a call of the python script searchB2IND.py with specified search criteria. A list of PIDs is then returned, which can be used to identify and retrieve data collections needed for further processing steps.

On the other hand, integration of other services in this context can mean, that the data of the associated provider are indexed and made searchable by the Discovery Service.