

Resource Management Overview



EOX IT Services <https://eox.at>



EOfarm P.C. <http://eofarm.com>

Φ-Week 2021-10-15

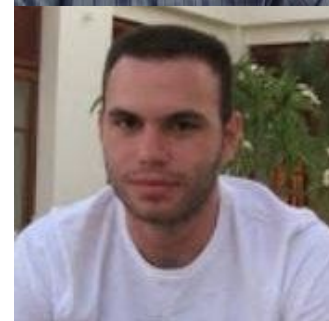
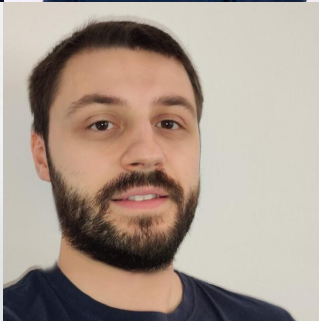


This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Overview

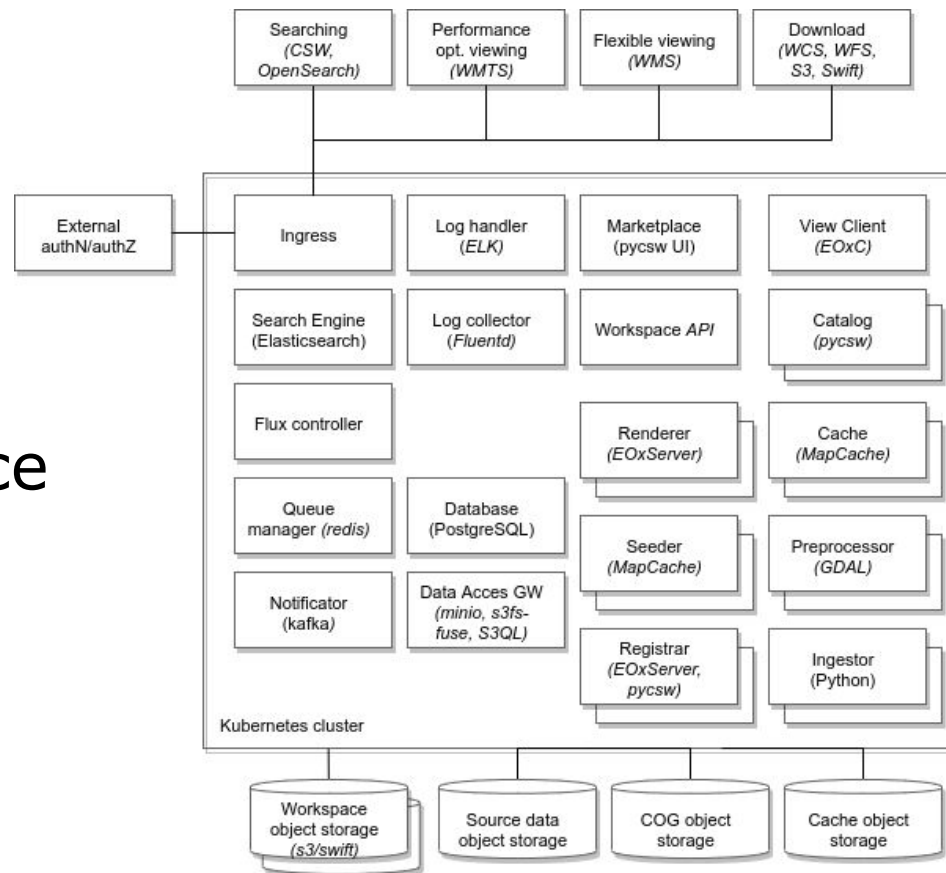
- Team
- Resource Management Architecture
- Resource Catalogue
- Data Access Service & Ingestion
- User Workspace
- Summary

Team



Resource Management Architecture

- Heavily based on [Python stack/ecosystem](#)
- Kubernetes mainly using helm charts
- Automation via flux
- System and per Workspace deployments



Resource Catalog Overview

- Discover/Search, Harvest, Distributed/Federated Search (user and system catalog)
- Metadata
- Standards
 - ISO 19115/19139
 - OGC CSW 2.0.2 / 3.0.0
 - Core Profile
 - ISO Application Profile
 - OGC OpenSearch Geo/Time, EO Profiles
 - OGC API - Records (Part1:Core)
 - STAC API (1.0.0-beta2), STAC Item (1.0.0)

Resource Catalog Technologies

- [pycsw](#)
 - Python OGC Reference Implementation Server
 - Multiple metadata formats (ISO, DC, etc.)
 - Flexible backends (RDBMS)
- [OWSLib](#)
 - Swiss army knife Python client library for OGC Web Services
- [pygeometa](#)
 - Metadata composition
- [pygeoapi](#)
 - Python OGC Reference Implementation Server
 - evolving OGC API standards

Resource Catalogue Technologies

- PostGIS
 - PostgreSQL Spatial extension
 - OGC SFSQL
 - Metadata Store for pycsw, EOxServer
- Elasticsearch
 - Full Text Search Engine
 - Highly Distributed

Resource Catalogue Examples

← → ↻ 🏠 🔒 https://resource-catalogue.demo.eoepca.org ... 📄 ☆ 🔍 Search



pycsw

pycsw Geospatial Catalogue

[Home](#) [JSON | Contact](#)

pycsw is an OGC CSW server implementation written in Python

[Collections](#)

OpenAPI

[Swagger](#)

[JSON](#)

[Conformance](#)

[CSW 3.0.0](#)

[CSW 2.0.2](#)

[OpenSearch](#)

[OAI-PMH](#)

[SRU](#)

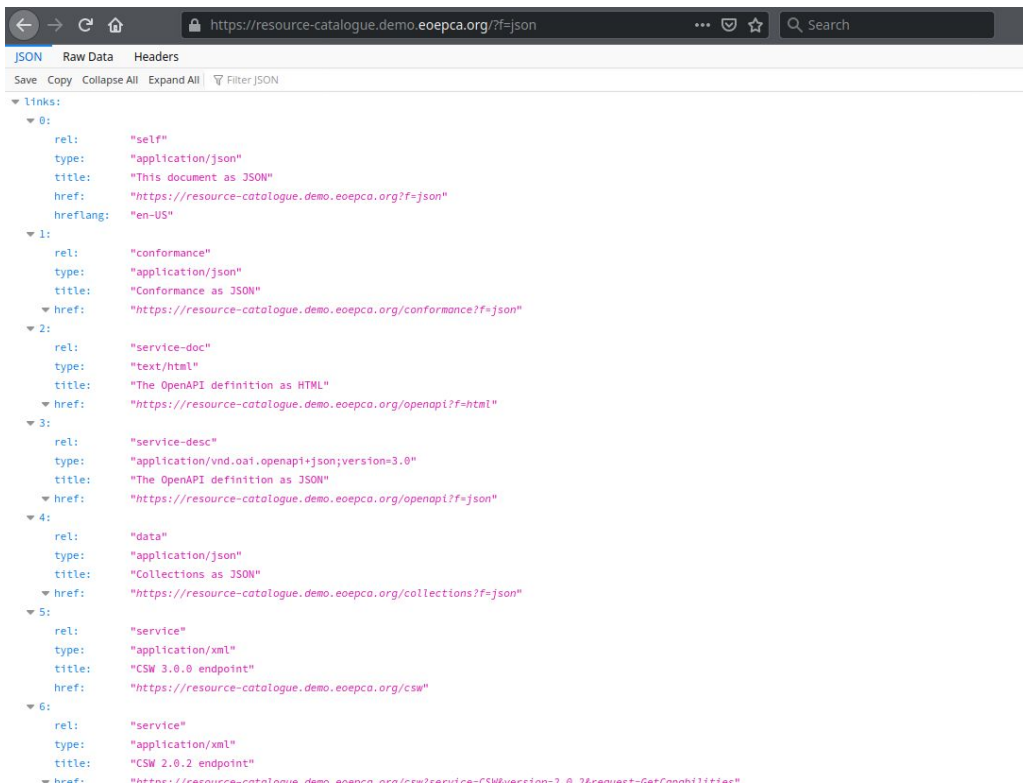
Powered by  pycsw 3.0.dev0

← → ↻ 🏠 🔒 https://resource-catalogue.demo.eoepca.org/csw ... 📄 ☆ 🔍 Search

This XML file does not appear to have any style information associated with it. The document tree is shown below.

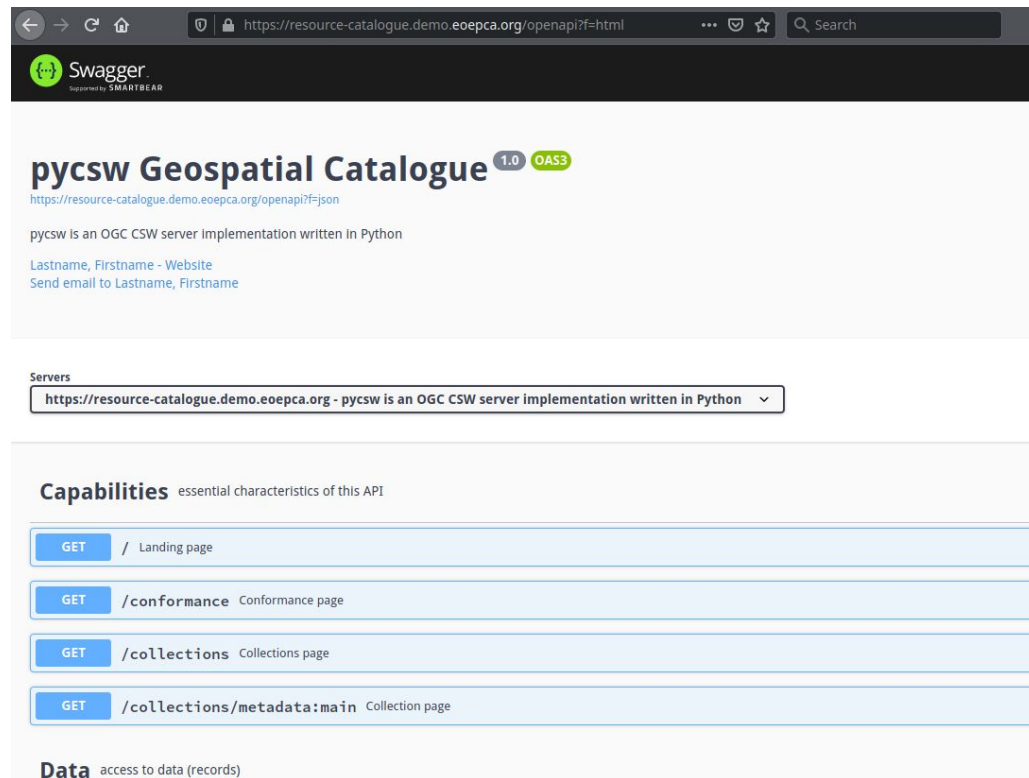
```
<!-- pycsw 3.0.dev0 -->
<csw30:Capabilities version="3.0.0" updateSequence="1631886261" xsi:schemaLocation="http://www.opengis.net/cat/csw/3.0 http://schemas.opengis.net/csw/3.0/cswGetCapabilities.xsd">
  <ows20:ServiceIdentification>
    <ows20:Title>pycsw Geospatial Catalogue</ows20:Title>
    <ows20:Abstract>
      pycsw is an OGC CSW server implementation written in Python
    </ows20:Abstract>
    <ows20:Keywords>
      <ows20:Keyword>catalogue</ows20:Keyword>
      <ows20:Keyword>discovery</ows20:Keyword>
      <ows20:Keyword>metadata</ows20:Keyword>
      <ows20:Type codeSpace="ISOTC211/19115">theme</ows20:Type>
    </ows20:Keywords>
    <ows20:ServiceType codeSpace="OGC">CSW</ows20:ServiceType>
    <ows20:ServiceTypeVersion>2.0.2</ows20:ServiceTypeVersion>
    <ows20:ServiceTypeVersion>3.0.0</ows20:ServiceTypeVersion>
    <ows20:Profile>http://www.opengis.net/cat/csw/apiso/1.0</ows20:Profile>
    <ows20:Fees>None</ows20:Fees>
    <ows20:AccessConstraints>None</ows20:AccessConstraints>
  </ows20:ServiceIdentification>
  <ows20:ServiceProvider>
    <ows20:ProviderName>Organization Name</ows20:ProviderName>
    <ows20:ProviderSite xlink:type="simple" xlink:href="https://pycsw.org/">
    <ows20:ServiceContact>
      <ows20:IndividualName>Lastname, Firstname</ows20:IndividualName>
      <ows20:PositionName>Position Title</ows20:PositionName>
    </ows20:ServiceContact>
    <ows20:ContactInfo>
      <ows20:Phone>
        <ows20:Voice>+xx-xxx-xxx-xxxx</ows20:Voice>
        <ows20:Facsimile>+xx-xxx-xxx-xxxx</ows20:Facsimile>
      </ows20:Phone>
      <ows20:Address>
        <ows20:DeliveryPoint>Mailing Address</ows20:DeliveryPoint>
        <ows20:City>City</ows20:City>
        <ows20:AdministrativeArea>Administrative Area</ows20:AdministrativeArea>
        <ows20:PostalCode>Zip or Postal Code</ows20:PostalCode>
      </ows20:Address>
    </ows20:ContactInfo>
  </ows20:ServiceProvider>
</csw30:Capabilities>
```


Resource Catalogue Examples



The screenshot shows a web browser with the address bar displaying `https://resource-catalogue.demo.eoepca.org/?f=json`. The page content is a JSON document with a root key `links` containing an array of link objects. The links are categorized by their `rel` values: `self`, `conformance`, `service-doc`, `service-desc`, `data`, and `service`. Each link object includes `rel`, `type`, `title`, `href`, and `hreflang` properties.

```
{
  "links": [
    {
      "rel": "self",
      "type": "application/json",
      "title": "This document as JSON",
      "href": "https://resource-catalogue.demo.eoepca.org/?f=json",
      "hreflang": "en-US"
    },
    {
      "rel": "conformance",
      "type": "application/json",
      "title": "Conformance as JSON",
      "href": "https://resource-catalogue.demo.eoepca.org/conformance?f=json"
    },
    {
      "rel": "service-doc",
      "type": "text/html",
      "title": "The OpenAPI definition as HTML",
      "href": "https://resource-catalogue.demo.eoepca.org/openapi?f=html"
    },
    {
      "rel": "service-desc",
      "type": "application/vnd.oai.openapi+json;version=3.0",
      "title": "The OpenAPI definition as JSON",
      "href": "https://resource-catalogue.demo.eoepca.org/openapi?f=json"
    },
    {
      "rel": "data",
      "type": "application/json",
      "title": "Collections as JSON",
      "href": "https://resource-catalogue.demo.eoepca.org/collections?f=json"
    },
    {
      "rel": "service",
      "type": "application/xml",
      "title": "CSW 3.0.0 endpoint",
      "href": "https://resource-catalogue.demo.eoepca.org/csw"
    },
    {
      "rel": "service",
      "type": "application/xml",
      "title": "CSW 2.0.2 endpoint",
      "href": "https://resource-catalogue.demo.eoepca.org/csw?version=2.0.2&request=GetCapabilities"
    }
  ]
}
```



The screenshot shows the Swagger UI for the **pycsw Geospatial Catalogue** (version 1.0, OAS3). The page title is **pycsw Geospatial Catalogue** and the URL is `https://resource-catalogue.demo.eoepca.org/openapi?f=json`. The description states: "pycsw is an OGC CSW server implementation written in Python". Below the description, there are links for "Lastname, Firstname - Website" and "Send email to Lastname, Firstname".

Servers

`https://resource-catalogue.demo.eoepca.org - pycsw is an OGC CSW server implementation written in Python`

Capabilities essential characteristics of this API

Method	Endpoint	Description
GET	/	Landing page
GET	/conformance	Conformance page
GET	/collections	Collections page
GET	/collections/metadata:main	Collection page

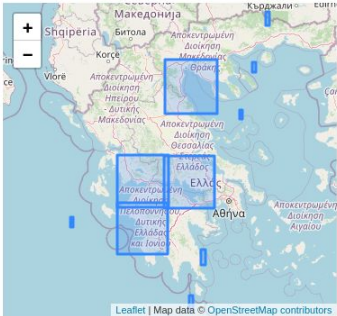
Data access to data (records)

Resource Catalogue Examples

← → ↻ 🏠 🔒 https://resource-catalogue.demo.eoepca.org/collections/metadata/ ... 🔍 Search


pycsw
pycsw Geospatial Catalogue

[Home](#) / [Collections](#) / [pycsw Geospatial Catalogue](#) / [Items](#) [JSON](#) | [Contact](#)



Leaflet | Map data © OpenStreetMap contributors

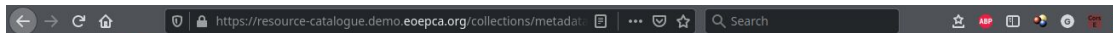
[Prev](#) [Next](#)

Title

S2A_MSIL1C_20200930T092031_N0209_R093_T34SDG_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T34TFK_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T35TLF_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T34SFF_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T34TGG_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T34SEH_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T34SEG_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T35SKD_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T34SFH_20200930T105731.SAFE	dataset
S2A_MSIL1C_20200930T092031_N0209_R093_T34SFF_20200930T105731.SAFE	dataset

[Prev](#) [Next](#)

Resource Catalogue Examples



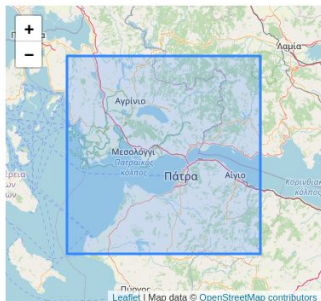
pycsw Geospatial Catalogue

[Home](#) / [Collections](#) / [pycsw Geospatial Catalogue](#) / [Items](#) /

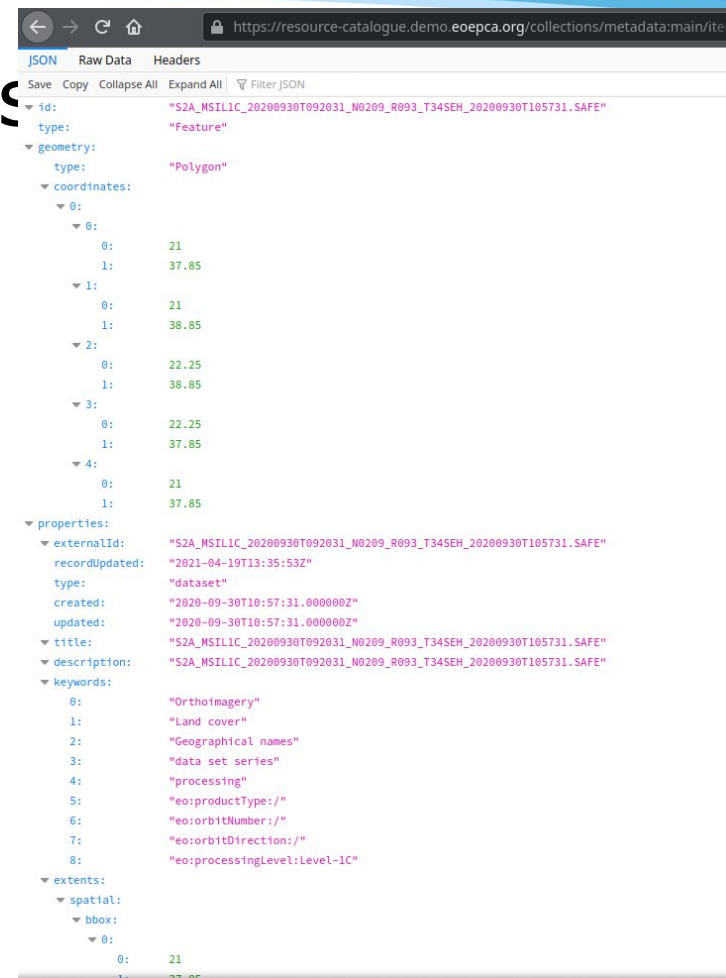
[JSON](#) | [Contact](#)

[S2A_MSIL1C_20200930T092031_N0209_R093_T34SEH_20200930T105731.SAFE](#)

S2A_MSIL1C_20200930T092031_N0209_R093_T34SEH_20200930T105731.SAFE



Property	Value
externalId	S2A_MSIL1C_20200930T092031_N0209_R093_T34SEH_20200930T105731.SAFE
recordUpdated	2021-04-19T13:35:53Z
type	dataset
created	2020-09-30T10:57:31.000000Z
updated	2020-09-30T10:57:31.000000Z
title	S2A_MSIL1C_20200930T092031_N0209_R093_T34SEH_20200930T105731.SAFE
description	S2A_MSIL1C_20200930T092031_N0209_R093_T34SEH_20200930T105731.SAFE
keywords	['Orthoimagery', 'Land cover', 'Geographical names', 'data set series', 'processing', 'eo:productType:/', 'eo:orbitNumber:/', 'eo:orbitDirection:/', 'eo:processingLevel:Level-1C']
extents	{ 'spatial': { 'bbox': [[21.0, 37.85, 22.25, 38.85]], 'crs': 'http://www.opengis.net/def/crs/OGC/1.3/CRS84' } }



Resource Catalogue Examples

```

Launcher
x 04 Resource Catalogue.ipyn
+ ✂ 📄 ▶ ■ ↺ ▶▶ Markdown v

[11]: #bbox_query = BBox([37.8, 23.4, 38.8, 24.5])
      bbox_query = BBox([37, 13.9, 37.9, 15.1])

[12]: begin = PropertyIsGreaterThanOrEqualTo(propertyname='apiso:TempExtent_begin', literal='2021-04-02 00:00')

[13]: end = PropertyIsLessThanOrEqualTo(propertyname='apiso:TempExtent_end', literal='2021-04-03 00:00')

[14]: cloud = PropertyIsLessThanOrEqualTo(propertyname='apiso:CloudCover', literal='20')

[15]: filter_list = [
      And(
        [
          bbox_query, # bounding box
          begin, end, # start and end date
          cloud # cloud
        ]
      )
    ]

```

The filter is then applied to the GetRecords request and results are shown:

```

[17]: csw.getrecords2(constraints=filter_list, outputschema='http://www.isotc211.org/2005/gmd')
      csw.results

[17]: {'matches': 1, 'returned': 1, 'nextrecord': 0}

[18]: selected_record = list(csw.records)[0]

[19]: for rec in csw.records:
      print(f'identifier: {csw.records[rec].identifier}\ntype: {csw.records[rec].identification.identtype}\n')

identifier: S2B_MSIL1C_20210402T095029_N0300_R079_T33SVB_20210402T121737.SAFE
type: dataset
title: S2B_MSIL1C_20210402T095029_N0300_R079_T33SVB_20210402T121737.SAFE

```

```

Launcher
x 04 Resource Catalogue.ipyn
+ ✂ 📄 ▶ ■ ↺ ▶▶ Markdown v Python 3

[ ]: for ref in rec.references:
      url = ref['url']
      if 'WCS' in url:
          print(msg(geolink=sniff_link(url), **ref))
          break

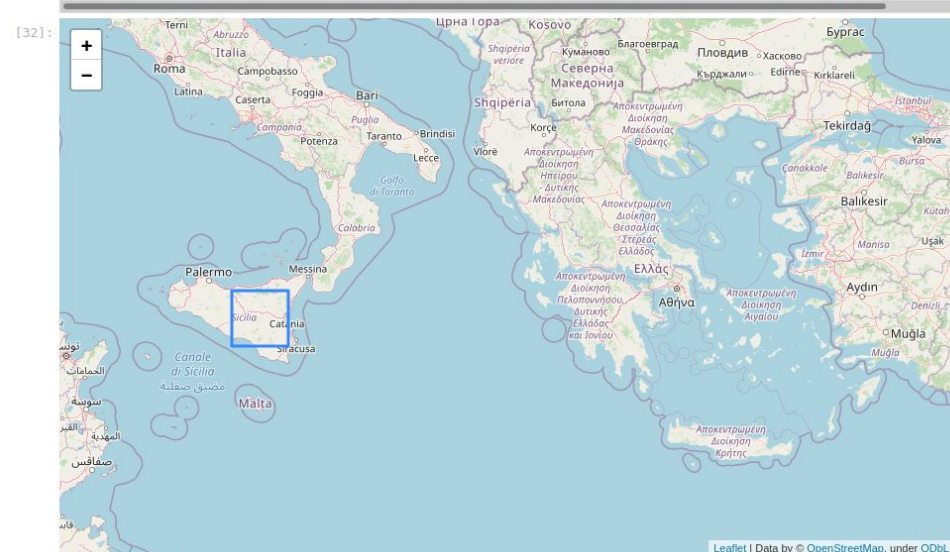
```

Finally we demonstrate how to show the record footprint on a map, using the **Folium** Python library:

```

[32]: m = folium.Map(location=[38, 20], zoom_start=6, tiles='OpenStreetMap')
      folium.Rectangle(bounds=[[float(rec.bbox.miny), float(rec.bbox.minx)], [float(rec.bbox.maxy), float(rec.bbox.maxx)]]).
      m

```



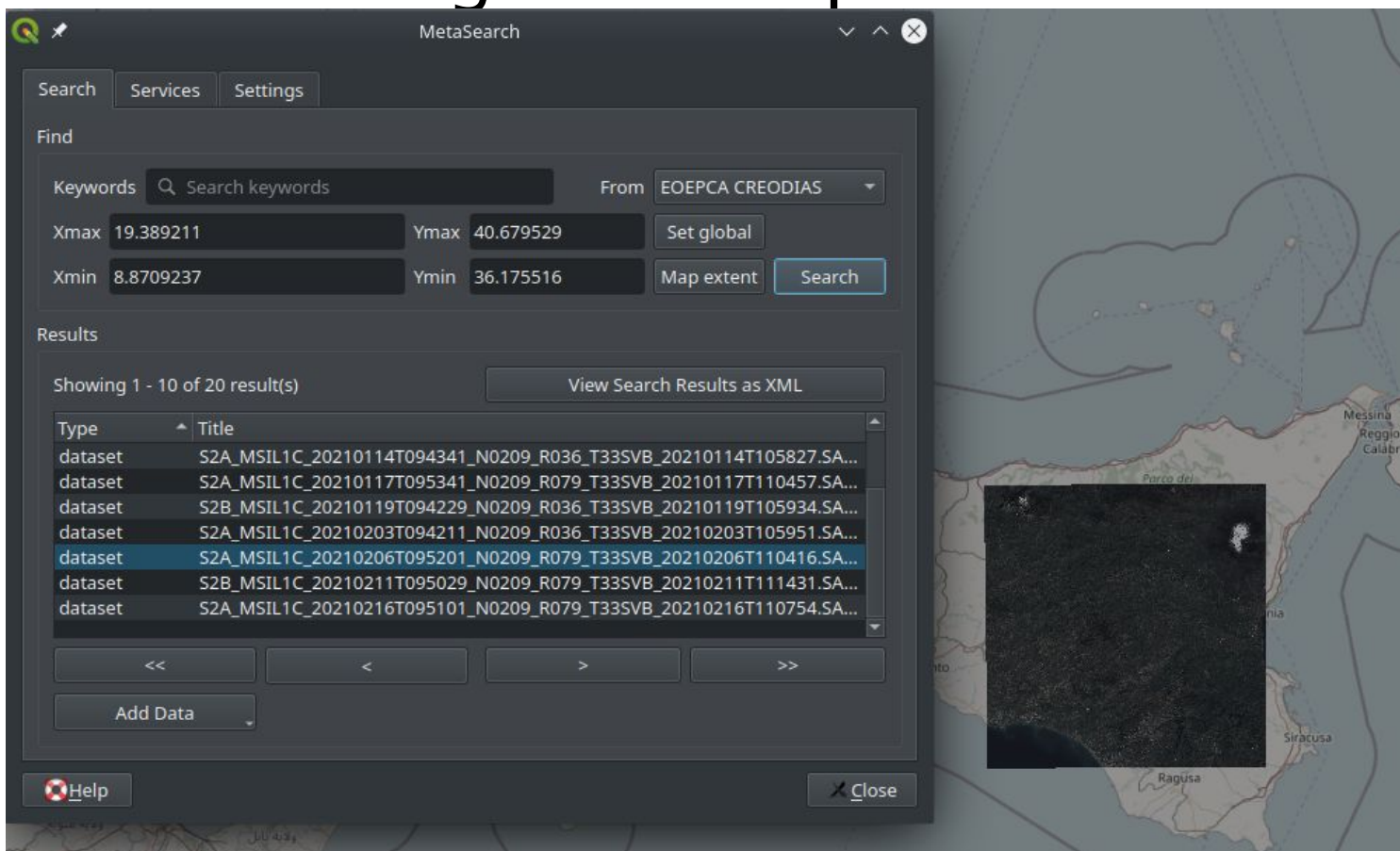
Resource Catalogue Examples

The left screenshot displays the 'Edit Catalog Service' dialog box. It has tabs for 'Search', 'Services', and 'Settings'. The 'Name' field is set to 'EOECA CREODIAS'. The 'URL' field is 'http://resource-catalogue.185.52.193.87.nip.io/'. The 'Authentication' section includes fields for 'User name' and 'Password'. The 'OK' button is highlighted.

The right screenshot displays the 'MetaSearch' window. It has tabs for 'Search', 'Services', and 'Settings'. The 'Find' section shows 'Keywords' as 'Search keywords' and 'From' as 'EOECA CREODIAS'. The 'Xmax' field is '19.389211', 'Ymax' is '40.679529', 'Xmin' is '8.8709237', and 'Ymin' is '36.175516'. The 'Search' button is highlighted. The 'Results' section shows 'Showing 1 - 10 of 20 result(s)' and a 'View Search Results as XML' button. The results table lists datasets with their titles.

Type	Title
dataset	S2A_MSIL1C_20210114T094341_N0209_R036_T33SVB_20210114T105827.SA...
dataset	S2A_MSIL1C_20210117T095341_N0209_R079_T33SVB_20210117T110457.SA...
dataset	S2B_MSIL1C_20210119T094229_N0209_R036_T33SVB_20210119T105934.SA...
dataset	S2A_MSIL1C_20210203T094211_N0209_R036_T33SVB_20210203T105951.SA...
dataset	S2A_MSIL1C_20210206T095201_N0209_R079_T33SVB_20210206T110416.SA...
dataset	S2B_MSIL1C_20210211T095029_N0209_R079_T33SVB_20210211T111431.SA...
dataset	S2A_MSIL1C_20210216T095101_N0209_R079_T33SVB_20210216T110754.SA...

Resource Catalogue Examples



The screenshot displays the MetaSearch application window, which is used for searching and managing datasets. The window is divided into several sections:

- Search Section:** Contains tabs for "Search", "Services", and "Settings". The "Find" section includes a "Keywords" search bar, a "From" dropdown menu set to "EOEPCA CREODIAS", and coordinate input fields for Xmax (19.389211), Ymax (40.679529), Xmin (8.8709237), and Ymin (36.175516). There are buttons for "Set global", "Map extent", and "Search".
- Results Section:** Shows "Showing 1 - 10 of 20 result(s)" and a button to "View Search Results as XML". Below this is a table of search results.
- Table of Results:**

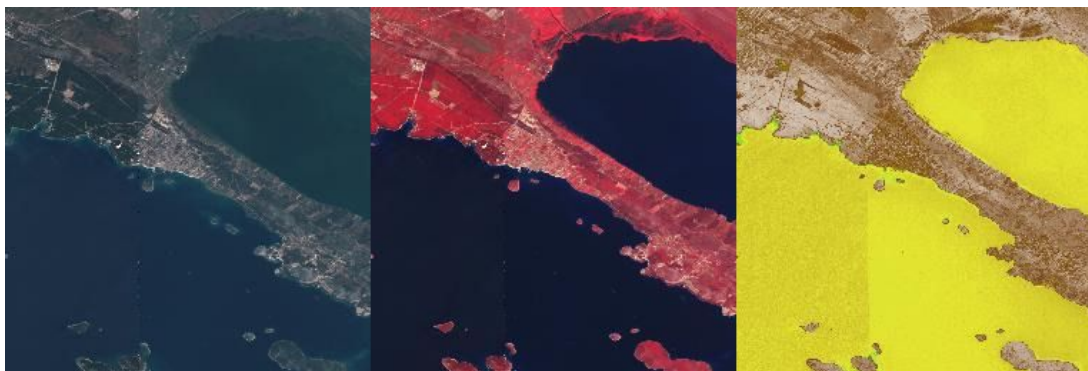
Type	Title
dataset	S2A_MSIL1C_20210114T094341_N0209_R036_T33SVB_20210114T105827.SA...
dataset	S2A_MSIL1C_20210117T095341_N0209_R079_T33SVB_20210117T110457.SA...
dataset	S2B_MSIL1C_20210119T094229_N0209_R036_T33SVB_20210119T105934.SA...
dataset	S2A_MSIL1C_20210203T094211_N0209_R036_T33SVB_20210203T105951.SA...
dataset	S2A_MSIL1C_20210206T095201_N0209_R079_T33SVB_20210206T110416.SA...
dataset	S2B_MSIL1C_20210211T095029_N0209_R079_T33SVB_20210211T111431.SA...
dataset	S2A_MSIL1C_20210216T095101_N0209_R079_T33SVB_20210216T110754.SA...

Below the table are navigation buttons: "<<", "<", ">", ">>". At the bottom left is an "Add Data" button, and at the bottom right are "Help" and "Close" buttons.

The background of the application shows a map of Sicily, Italy, with a dark rectangular area indicating the search results location. Labels on the map include "Messina", "Reggio di Calabria", "Siracusa", and "Ragusa".

Data Access Service Overview

- View Server (EOxServer as core)
- Flexible WMS viewing interface
 - raster expressions, outlines, masking, property filtering
- Performance optimized WM(T)S cached viewing



Data Access Service Overview

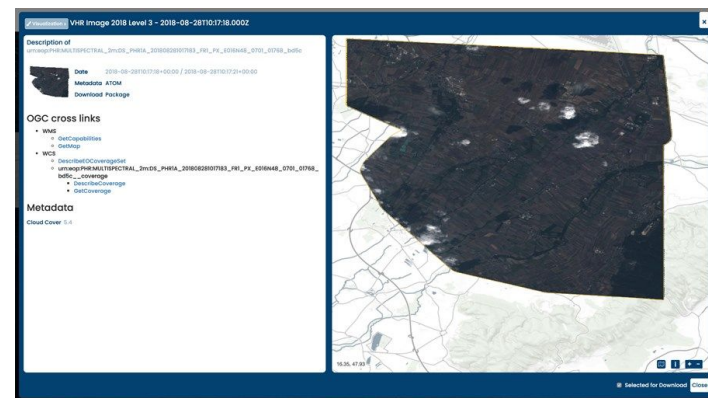
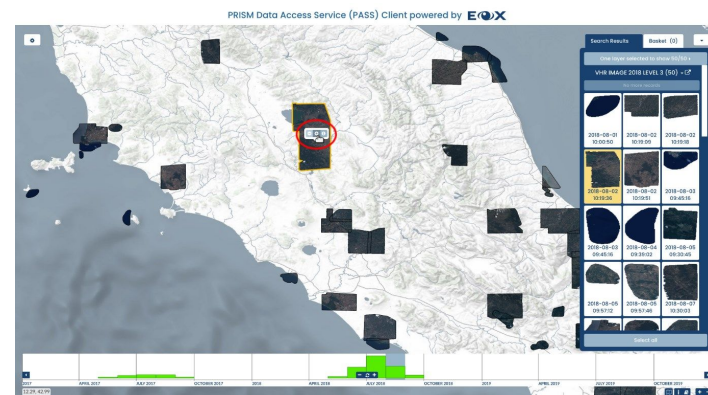
- Download interfaces
 - WCS + EO Application Profile
 - DSEO
- OGC OpenSearch catalog interface
 - Geo Extension
 - Time Extension
 - EO Extension
 - CQL filtering

Data Access Ingestion

- Harvesting from System catalog
- Metadata enrichment
- Export to STAC Item
- Registration into Data Access and Resource Catalog

Data Access Service & Ingestion Components

- Database - [PostgreSQL](#)
- Renderer - [EOxServer](#)
- Cache - [Mapcache](#)
- Registrar - [Python](#) + [EOxServer](#)
- Harvester - [Python](#)
- [Redis](#) queues
- Webclient based on [EOxC](#)
- [Elasticsearch](#)/[Fluentd](#)/[Kibana](#) logging stack



User Workspace

- EOEPKA Model
 - Object storage bucket
 - Data Access Service
 - Resource Catalog with federated system catalog
 - Deployment scenario(s) up to operator
- [Workspace API](#) based on FastAPI
 - create, delete, etc.
 - using “Bucket” claims via K8s CRD & K8s Secret
 - register data & processes
- [Bucket operator](#) for OpenStack

User Workspace

FastAPI 0.1.0 OAS3

/openapi.json

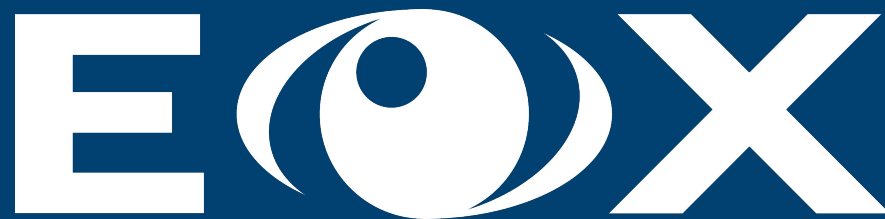
default



GET	/probe	Probe	▼
POST	/workspaces	Create Workspace	▼
GET	/workspaces/{workspace_name}	Get Workspace	▼
DELETE	/workspaces/{workspace_name}	Delete Workspace	▼
PATCH	/workspaces/{workspace_name}	Patch Workspace	▼
POST	/workspaces/{workspace_name}/redeploy	Redeploy Workspace	▼
POST	/workspaces/{workspace_name}/register	Register	▼
POST	/workspaces/{workspace_name}/deregister	Deregister	▼

Summary

- Open Standards (OGC, ISO)
- Free and Open Source (FOSS)
 - Core components
 - Reference implementation
 - Geospatial (FOSS4G)
- Best of breed Python geospatial components
- Download, configure, and run
 - <https://github.com/EOEPCA/eoepca>
- Open communities
- Open to contributions



Thanks for your attention!



EOX IT Services <https://eox.at>



EOfarm P.C. <http://eofarm.com>

Φ-Week 2021-10-15



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).