Science and Technology Facilities Council

Natural Environment Research Council

Developing Community standards-based Search Tools for Earth System Model Data using STAC

Richard Smith[1], Philip Kershaw[1], Ag Stephens[2], Rhys Evans[2], Aparna Radhakrishnan[3], V. Balaji[3], Ryan Abernathey[4]

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

JASMIN

Scientific Computing
Science & Technology Facilities Council

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

RAL Space

Overview

1 What are we looking to solve?

A look at our shared problem

2 STAC Overview

What is STAC?

3 Proposed Solution

System including indexing framework, server and clients

Science and Technology Facilities Council

Natural Environment Research Council

Background

The CEDA Archive

- Holds > 13 PB of atmospheric and earth observation data
- Data from many different sources
- 7300 datasets in the CEDA catalogue
- 344+ million files
- +200,000/day
- Our data is stored on POSIX disk, Tape archive and Object Store

Background

ESGF

International collaboration that develops, deploys and maintains software infrastructure for the management, dissemination, and analysis of model output and observational data.
Data catalog comprises several observation and modelled datasets.

Pangeo

Community promoting open, reproduceable, scalable science. Aims to coordinate scientists, software and computing to further research.
Looking for a cataloguing standard (familiar with Intake)
Converting popular data to Cloud Optimised Formats (Zarr) with Google Cloud and Amazon

What do we want?

Develop a search tool which allows users to perform faceted search and find the relevant data for their use-case, taking into account the heterogeneity of the data.

It needs to

- Allow low level search of all items (granules)
- Work with different domains/vocabularies
- Provide faceted search
- Handle heterogeneous datasets
- Be scalable
- Also comprise an indexing framework to generate content

Science and Technology Facilities Council
Natural Environment Research Council

What have we tried?

Intake Catalogs (PANGEO)

- Static
- Scales to a certain extent
- Familiarity in the community

ESGF Search

- Proprietary API format
- Complex publication process
- Provides rich faceted search

File-based Search (CEDA)

- Leverages Elasticsearch
- Quick and scalable search
- Powers light-touch directory browser

Benefits of Shared Approach

- Community only needs to learn one API
- Shared development effort
- Expanding scope and use cases allows solutions to be found for common issues

Science and Technology Facilities Council
Natural Environment Research Council

Shared Problem

We think this problem is common among data providers with heterogeneous data.

We are actively looking for collaborators to work with us on this

Science and Technology Facilities Council

Natural Environment Research Council

Science and Technology Facilities Council

Natural Environment Research Council

STAC Overview

STAC SpatioTemporal Asset Catalog

Science and Technology Facilities Council

Natural Environment Research Council

What is STAC?

“The SpatioTemporal Asset Catalog (STAC) specification provides a common language to describe a range of geospatial information, so it can more easily be indexed and discovered. A 'spatiotemporal asset' is any file that represents information about the earth captured in a certain space and time.

The goal is for all providers of spatiotemporal assets (Imagery, SAR, Point Clouds, Data Cubes, Full Motion Video, etc) to expose their data as SpatioTemporal Asset Catalogs (STAC), so that new code doesn't need to be written whenever a new data set or API is released.”

https://stacspec.org/

Science and Technology Facilities Council
Natural Environment Research Council

STAC
SpatioTemporal Asset Catalog

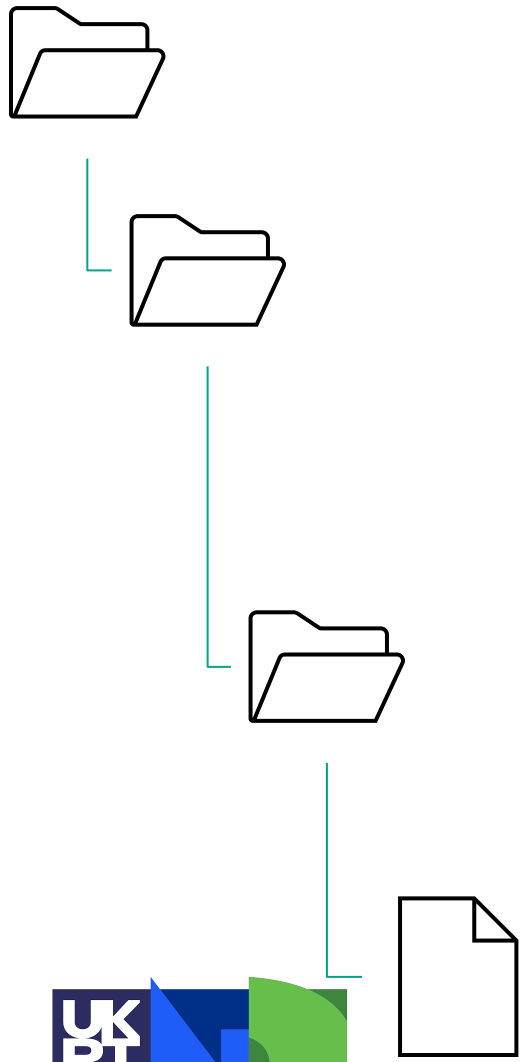
What is STAC?

- Based on OGC API Features
- JSON formatted response and data standard
- Extension is encouraged
- Requires temporal and spatial attributes
- Available as either a static or dynamic API
- Strong community engagement

Science and Technology Facilities Council
Natural Environment Research Council

STAC
SpatioTemporal
Asset Catalog

What is STAC?



“Catalog - *a simple, flexible JSON file of links that provides a structure to organize and browse STAC Items.”*

“Collection - *an extension of the STAC Catalog with additional information such as the extents, license, keywords, providers, etc. that describe STAC Items that fall within the Collection.”*

“Item - *the core atomic unit, representing a single spatiotemporal asset as a GeoJSON feature plus datetime and links.”*

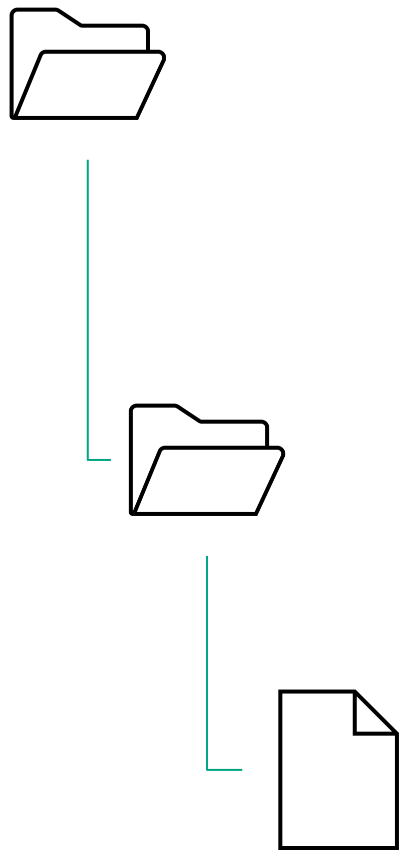
“Assets - *an object that contains a URI to data associated with the Item that can be downloaded or streamed.”*

Science and Technology Facilities Council

Natural Environment Research Council

https://stacspec.org/

What is STAC? - For us



Collection - Set of items with a common vocabulary/DRS (e.g. CMIP6)

Item - Meaningful group of 1+ files (e.g. ESGF *atomic* Dataset)

Asset - Single data file (e.g. one NetCDF file)

STAC – Static or Dynamic?

Static Catalogs – A set of interlinked JSON files which can be navigated hierarchically (navigation is set)

Dynamic Catalogs – Use the STAC API to provide enhanced navigation and search capability (can perform item search, faceted search, etc.)

Science and Technology Facilities Council
Natural Environment Research Council

Public listing of catalogs: https://stacindex.org/catalogs#/

STAC Ecosystem

Active community

- Public register of tools (https://stacindex.org/ecosystem)
- Public register of catalogs (https://stacindex.org/catalogs#/)
- Community Extensions (https://github.com/stac-extensions/)

Documentation

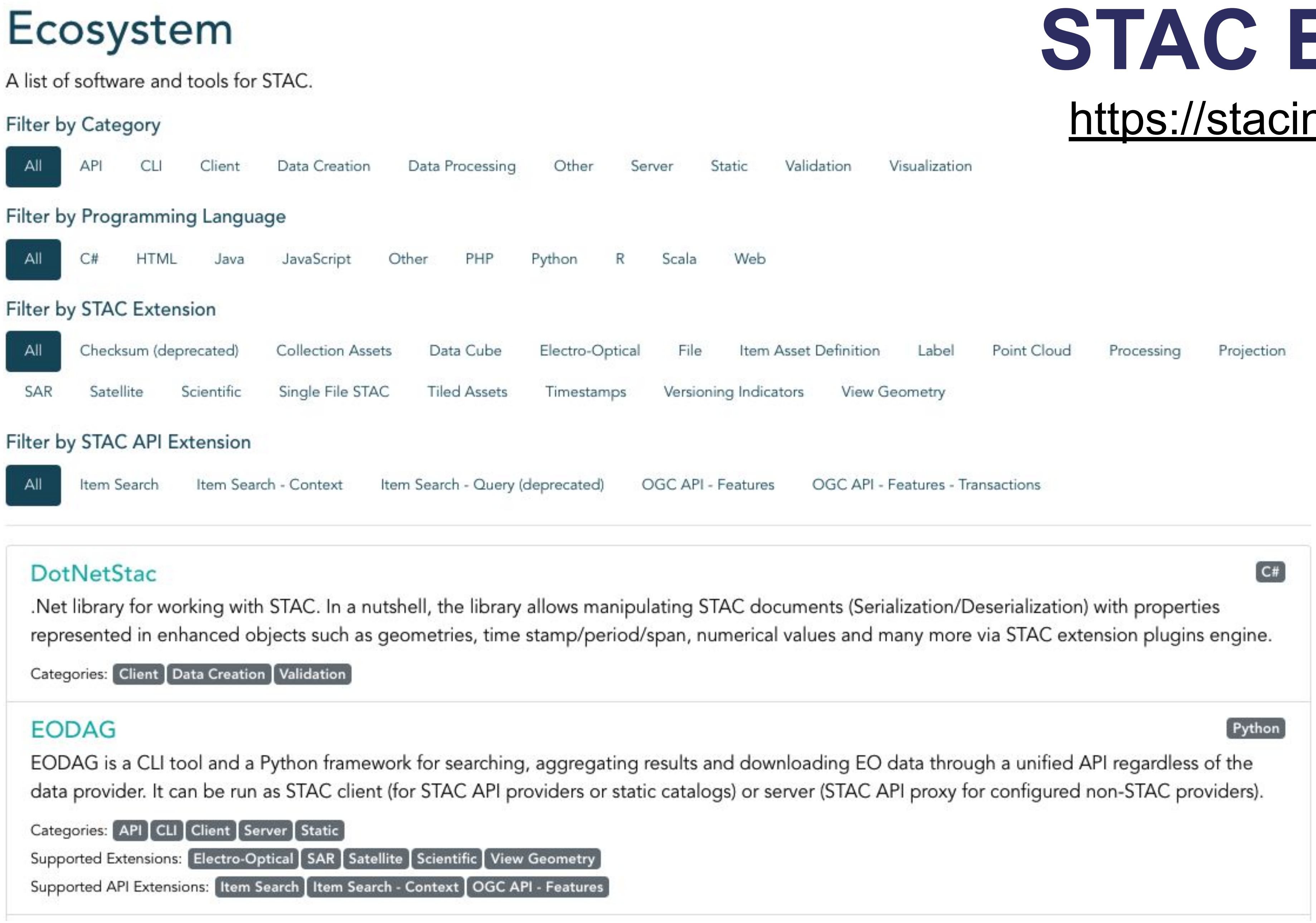
- STAC Specification (https://github.com/radiantearth/stac-spec)
- STAC API Specification (https://github.com/radiantearth/stac-api-spec)

Science and Technology Facilities Council Natural Environment Research Council

https://stacindex.org/ecosystem

STAC Ecosystem

https://stacindex.org/ecosystem



Ecosystem

A list of software and tools for STAC.

Filter by Category

All API CLI Client Data Creation Data Processing Other Server Static Validation Visualization

Filter by Programming Language

All C# HTML Java JavaScript Other PHP Python R Scala Web

Filter by STAC Extension

All Checksum (deprecated) Collection Assets Data Cube Electro-Optical File Item Asset Definition Label Point Cloud Processing Projection SAR Satellite Scientific Single File STAC Tiled Assets Timestamps Versioning Indicators View Geometry

Filter by STAC API Extension

All Item Search Item Search - Context Item Search - Query (deprecated) OGC API - Features OGC API - Features - Transactions

DotNetStac C#

.Net library for working with STAC. In a nutshell, the library allows manipulating STAC documents (Serialization/Deserialization) with properties represented in enhanced objects such as geometries, time stamp/period/span, numerical values and many more via STAC extension plugins engine.

Categories: Client Data Creation Validation

EODAG Python

EODAG is a CLI tool and a Python framework for searching, aggregating results and downloading EO data through a unified API regardless of the data provider. It can be run as STAC client (for STAC API providers or static catalogs) or server (STAC API proxy for configured non-STAC providers).

Categories: API CLI Client Server Static

Supported Extensions: Electro-Optical SAR Satellite Scientific View Geometry

Supported API Extensions: Item Search Item Search - Context OGC API - Features

Some issues

- "ST" (Spatial and Temporal) in STAC is mandatory
  - CEDA holds Martian datasets
- Dynamic facet reduction (based on search context) not yet in the spec

Initially gained traction with Earth Observation community. Challenges with model data include:

- Rotated grids
- Non-standard dates (paleo-climatology, non-standard calendars)

Science and Technology Facilities Council
Natural Environment Research Council

Given the flexibility of STAC and community engagement, we feel STAC fulfils the majority of our requests and can be adapted to fill others.

Science and Technology Facilities Council

Natural Environment Research Council

Science and
Technology
Facilities Council

Natural
Environment
Research Council

Progress So Far

Overview

Indexing

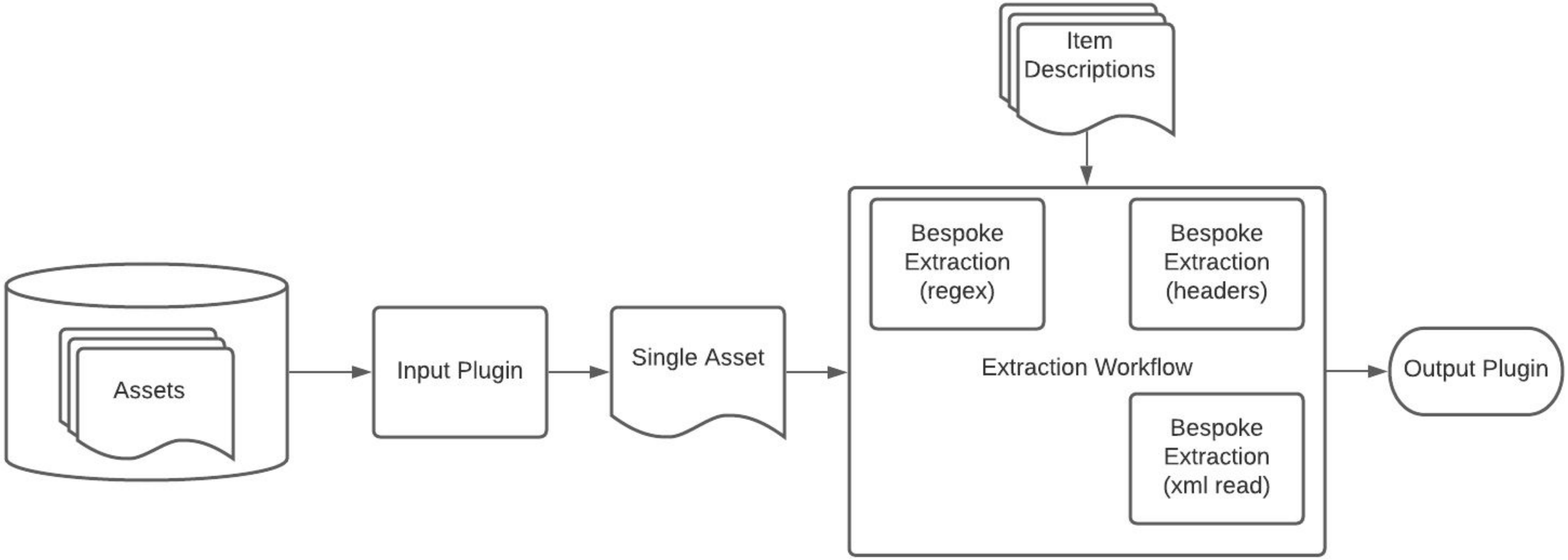
Server

Clients

Science and Technology Facilities Council

Natural Environment Research Council

Indexing Framework – Asset Scanner

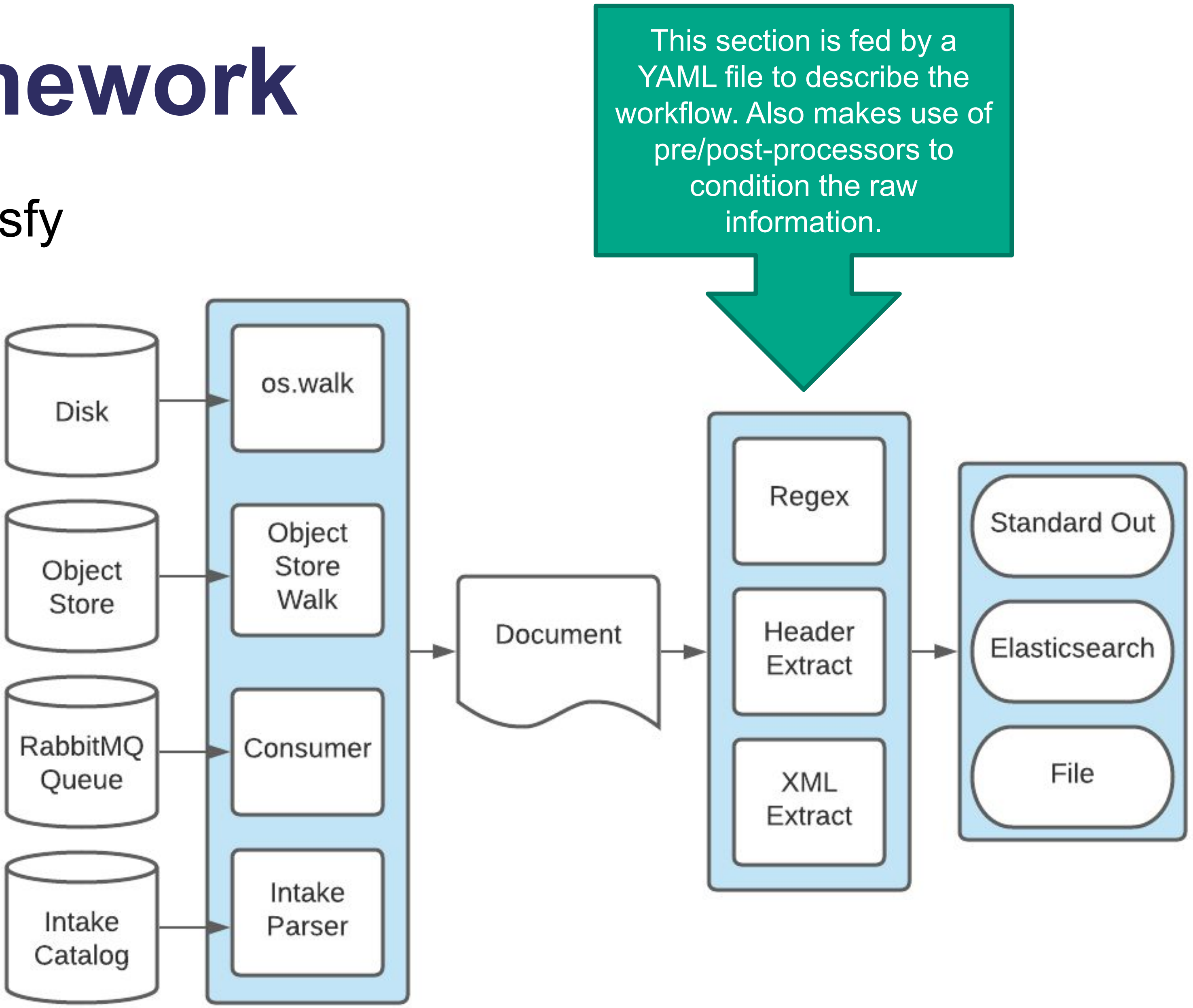

Item Descriptions
Assets
Input Plugin
Single Asset
Bespoke Extraction (regex)
Bespoke Extraction (headers)
Extraction Workflow
Bespoke Extraction (xml read)
Output Plugin

https://github.com/cedadev/asset-scanner

Science and Technology Facilities Council
Natural Environment Research Council

Indexing Framework

Plugin architecture can satisfy different use cases:



STAC Server

- Based on community server STAC FastAPI
  https://github.com/stac-utils/stac-fastapi

- Elasticsearch backend
  https://github.com/cedadev/stac-fastapi-elasticsearch

Science and Technology Facilities Council
Natural Environment Research Council

STAC Extensions

Allow you to write content and feature specifications to extend the basic STAC specification.

Some things we want:

- Free-text search
- Dynamic facet discovery

STAC Extensions

Two categories:

Content

- Processing – *(processing level and provenance)*
- Datacube – *(variables & dimensions)*

API

- Filter
- Context
- Sort
- Transaction

https://github.com/orgs/stac-extensions/repositories

https://github.com/radiantearth/stac-api-spec/blob/master/extensions.md

STAC Extensions

So far we haven't looked at content extensions. The flexible properties attribute can hold rich metadata.

On the feature side:

- **Free-text search**
  https://github.com/cedadev/stac-freetext-search
- **Context collections**
  https://github.com/cedadev/stac-context-collections

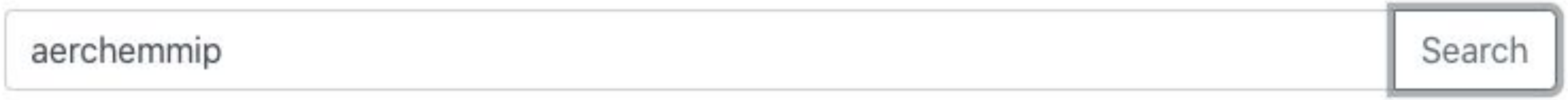
Science and Technology Facilities Council

Natural Environment Research Council

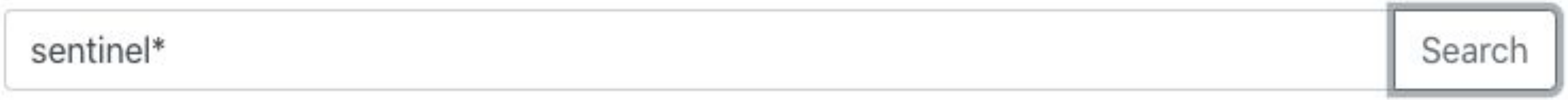
Free-text Search

Powered by Elasticsearch query string

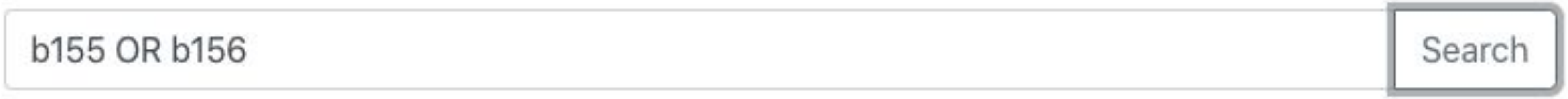
Free-text Search any “property”


aerchemmip

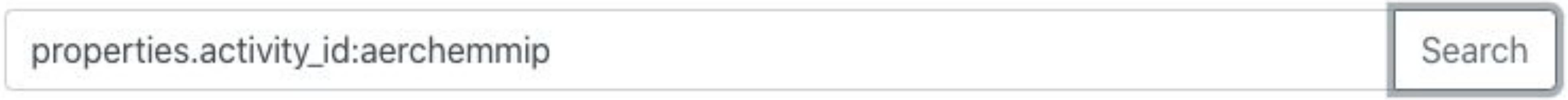
Free-text Search with wildcard


sentinel*

Free-text Search with logical operators


b155 OR b156

Free-text Search for specific fields


properties.activity_id:aerchemmip

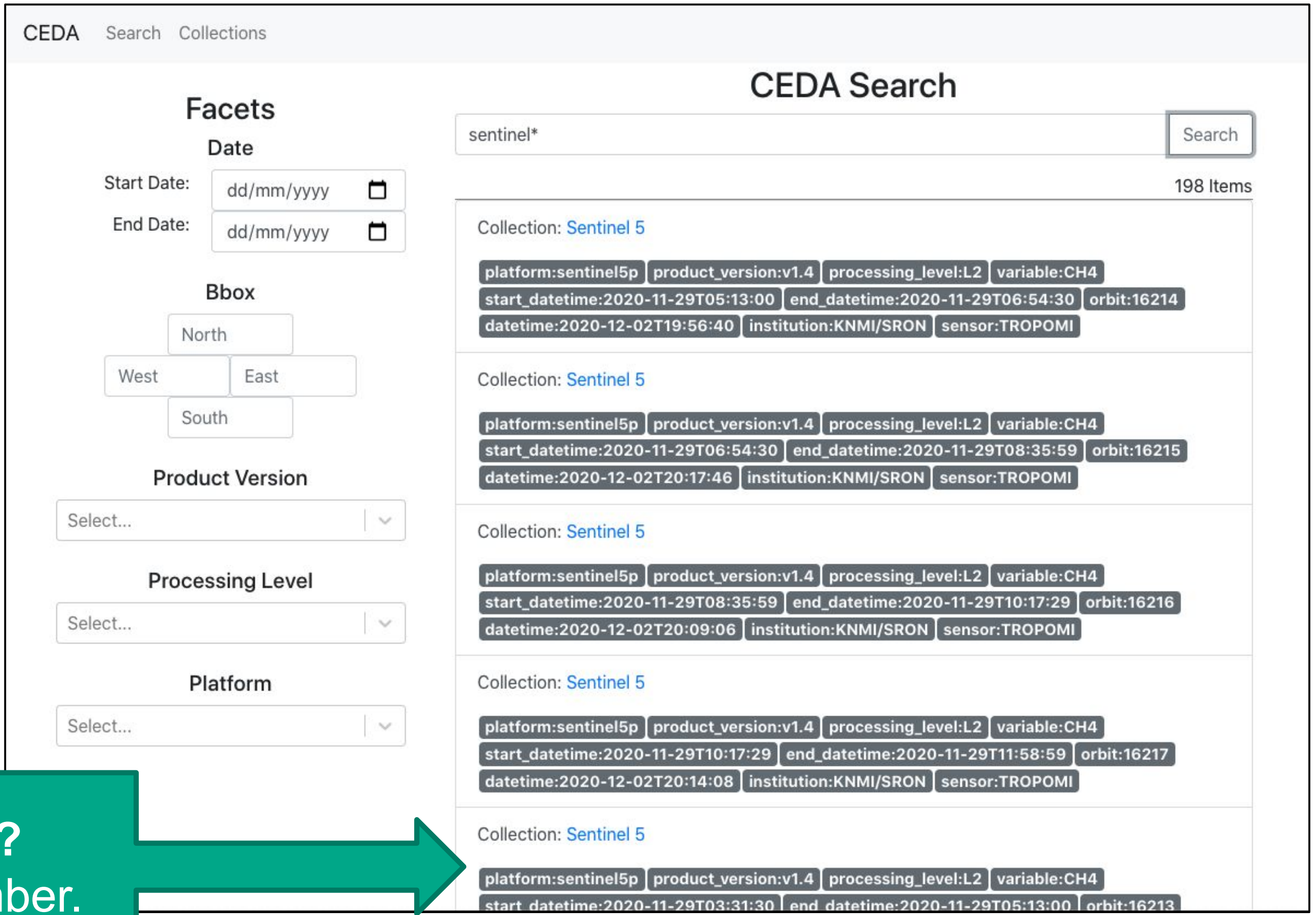
Science and Technology Facilities Council
Natural Environment Research Council

https://github.com/cedadev/stac-freetext-search

Clients - Web

Why make our own?

- Existing tools are focussed on static catalogs
- API supporting UIs are very map-centric (not relevant for global model datasets)
- Want to be able to move quickly and try out new features (free-text search, faceted-search)
- Loosely based on the STAC Browser



CEDA Search Collections

Facets

Date

Start Date: dd/mm/yyyy

End Date: dd/mm/yyyy

Bbox

North

West East

South

Product Version

Select...

Processing Level

Select...

Platform

Select...

CEDA Search

sentinel* Search

198 Items

Collection: Sentinel 5

platform:sentinel5p | product_version:v1.4 | processing_level:L2 | variable:CH4
start_datetime:2020-11-29T05:13:00 | end_datetime:2020-11-29T06:54:30 | orbit:16214
datetime:2020-12-02T19:56:40 | institution:KNMI/SRON | sensor:TROPOMI

Collection: Sentinel 5

platform:sentinel5p | product_version:v1.4 | processing_level:L2 | variable:CH4
start_datetime:2020-11-29T06:54:30 | end_datetime:2020-11-29T08:35:59 | orbit:16215
datetime:2020-12-02T20:17:46 | institution:KNMI/SRON | sensor:TROPOMI

Collection: Sentinel 5

platform:sentinel5p | product_version:v1.4 | processing_level:L2 | variable:CH4
start_datetime:2020-11-29T08:35:59 | end_datetime:2020-11-29T10:17:29 | orbit:16216
datetime:2020-12-02T20:09:06 | institution:KNMI/SRON | sensor:TROPOMI

Collection: Sentinel 5

platform:sentinel5p | product_version:v1.4 | processing_level:L2 | variable:CH4
start_datetime:2020-11-29T10:17:29 | end_datetime:2020-11-29T11:58:59 | orbit:16217
datetime:2020-12-02T20:14:08 | institution:KNMI/SRON | sensor:TROPOMI

Collection: Sentinel 5

platform:sentinel5p | product_version:v1.4 | processing_level:L2 | variable:CH4
start_datetime:2020-11-29T03:31:30 | end_datetime:2020-11-29T05:13:00 | orbit:16213

What's different?
Start, end, orbit number.

https://stac.ceda.ac.uk/search

Science and Technology Facilities Council Natural Environment Research Council

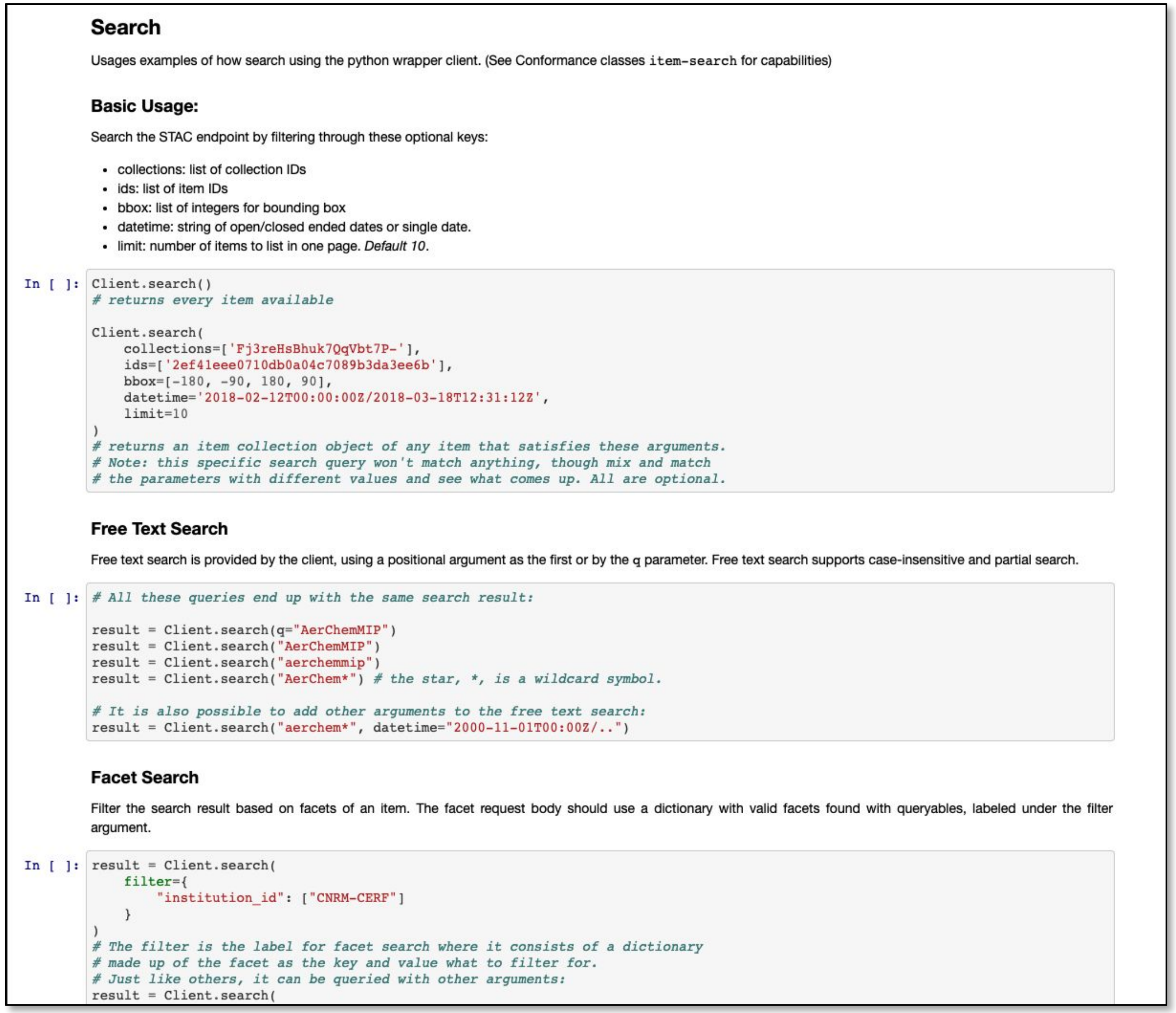
Clients - Python

Based upon stac.py.

Fully exposes all the capabilities provided by the STAC API server.

Retrieve collection and items as their own models with respective functions for their model.

The response models are no different in structure than the JSON counterpart!



Search

Usages examples of how search using the python wrapper client. (See Conformance classes `item-search` for capabilities)

Basic Usage:

Search the STAC endpoint by filtering through these optional keys:

- collections: list of collection IDs
- ids: list of item IDs
- bbox: list of integers for bounding box
- datetime: string of open/closed ended dates or single date.
- limit: number of items to list in one page. *Default 10*.

```
In [ ]: Client.search()
        # returns every item available

        Client.search(
            collections=['Fj3reHsBhuk7QqVbt7P-'],
            ids=['2ef41eee0710db0a04c7089b3da3ee6b'],
            bbox=[-180, -90, 180, 90],
            datetime='2018-02-12T00:00:00Z/2018-03-18T12:31:12Z',
            limit=10
        )
        # returns an item collection object of any item that satisfies these arguments.
        # Note: this specific search query won't match anything, though mix and match
        # the parameters with different values and see what comes up. All are optional.
```

Free Text Search

Free text search is provided by the client, using a positional argument as the first or by the q parameter. Free text search supports case-insensitive and partial search.

```
In [ ]: # All these queries end up with the same search result:

        result = Client.search(q="AerChemMIP")
        result = Client.search("AerChemMIP")
        result = Client.search("aerchemmip")
        result = Client.search("AerChem*") # the star, *, is a wildcard symbol.

        # It is also possible to add other arguments to the free text search:
        result = Client.search("aerchem*", datetime="2000-11-01T00:00Z/..")
```

Facet Search

Filter the search result based on facets of an item. The facet request body should use a dictionary with valid facets found with queryables, labeled under the filter argument.

```
In [ ]: result = Client.search(
            filter={
                "institution_id": ["CNRM-CERF"]
            }
        )
        # The filter is the label for facet search where it consists of a dictionary
        # made up of the facet as the key and value what to filter for.
        # Just like others, it can be queried with other arguments:
        result = Client.search(
```

Science and Technology Facilities Council
Natural Environment Research Council

Binder Example Notebook

Where next?

- Improving faceted search
  https://github.com/radiantearth/stac-api-spec/issues/182
- Improve indexing coverage of the CEDA Archive
- Build out python client, specifically with ESGF community in mind
- Work with others to improve our approach

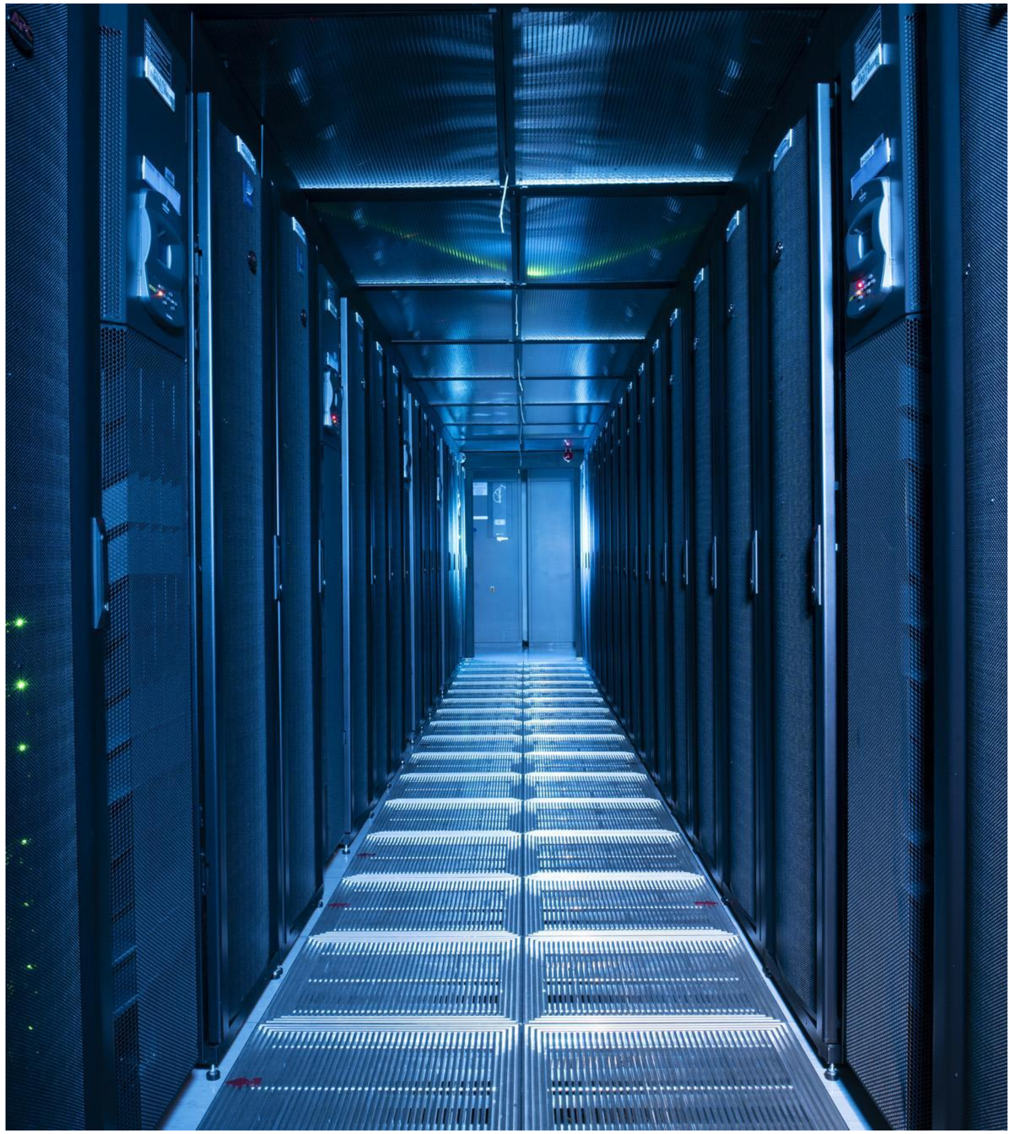
Science and Technology Facilities Council
Natural Environment Research Council

Science and Technology Facilities Council
Natural Environment Research Council

Thank you!

JASMIN: support@jasmin.ac.uk
CEDA: support@ceda.ac.uk
Twitter - @cedanews
Website - www.ceda.ac.uk

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°824084



Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL
JASMIN
Scientific Computing
Science & Technology Facilities Council
National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL
National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL
RAL Space