

STAC for CEDA - Developing a scalable, standards-based search system

Rhys Evans, Sam Pepler, Ag Stephens,
Mahir Rahman, Richard Smith

CEDA (Centre for Environmental Data Analysis)

Who are CEDA?

- Part of UKRI
- Facilitate research in atmospheric and Earth observation
- The CEDA Archive
 - Long term data store
- JASMIN
 - Data intensive supercomputer



JASMIN



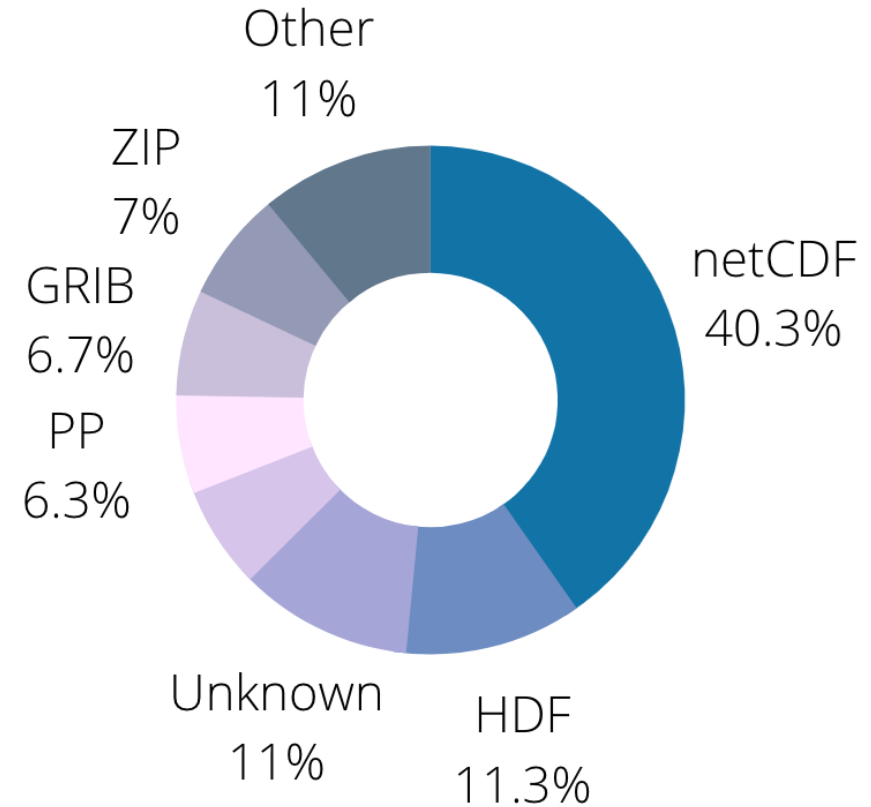
RAL Space



Motivation

The CEDA Archive

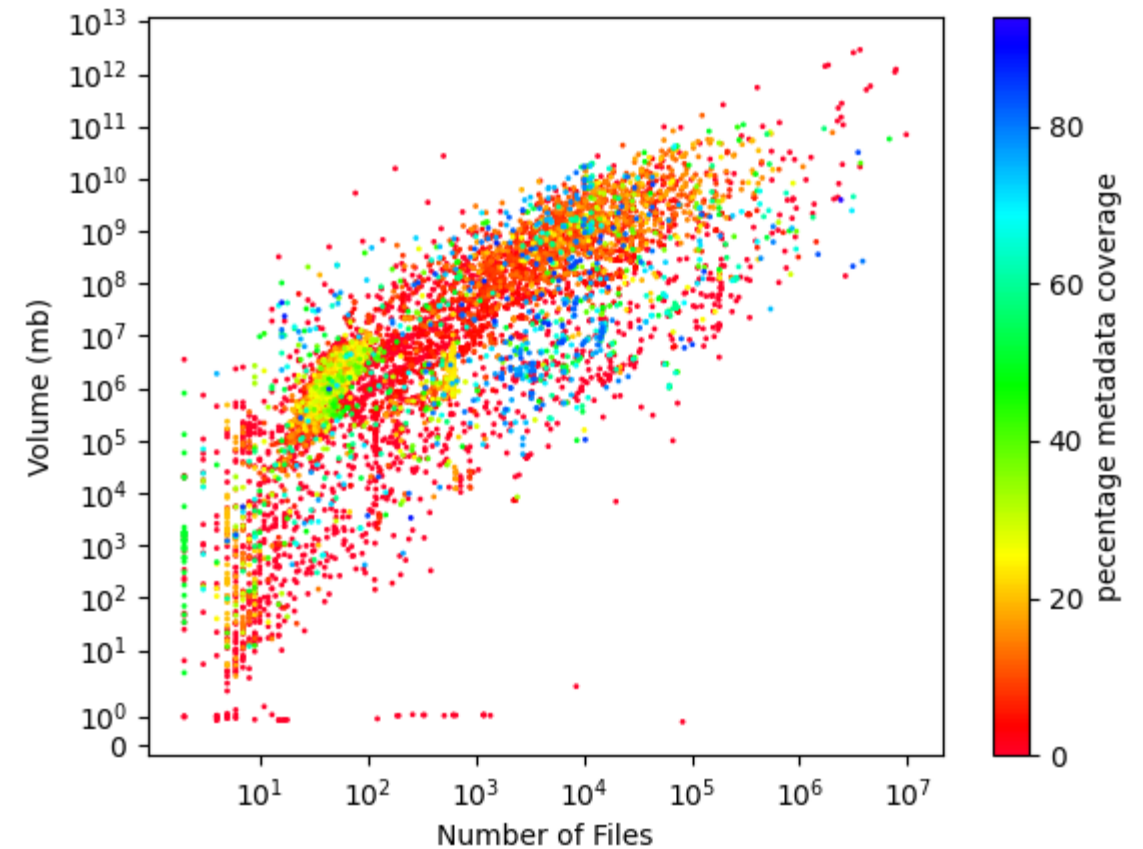
- Large scale
 - > 20 PB of data
 - > 350 million files
 - > 200,000 files a day
- Highly Heterogeneous
 - Different sources
 - Different storage types
 - Different formats



PreSTAC search

General search

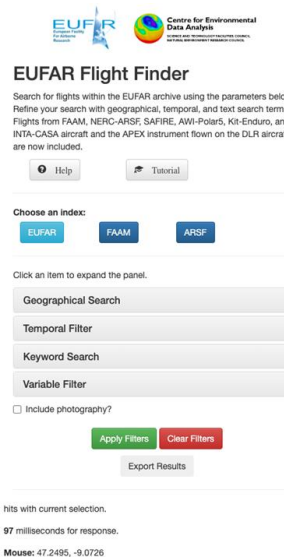
- Catalogue
- File Based Index
 - Basic file level information
 - Metadata from the most common formats
 - Only 48% coverage



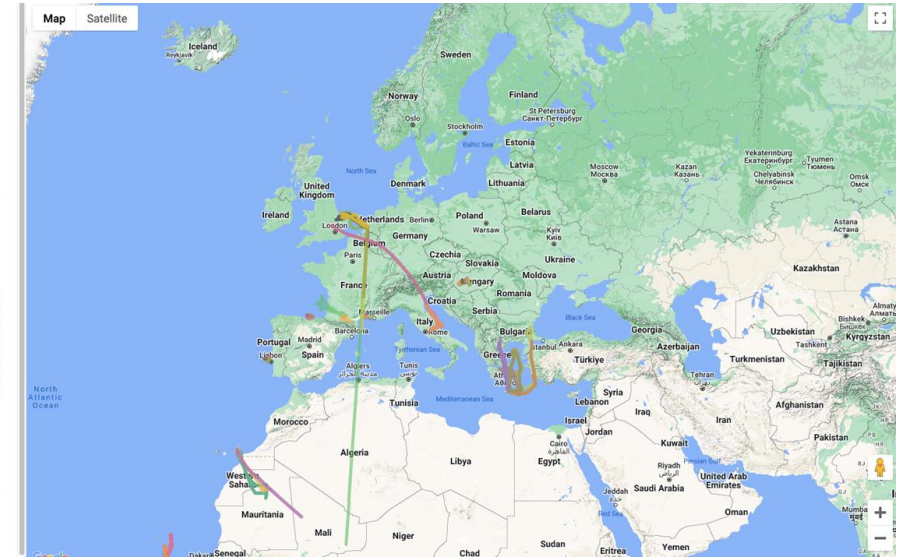
PreSTAC search

Specific search

- EUFAR Flight Finder
- Focus on a specific area of archive
 - More homogeneous
 - Rich metadata
- Only for a section of the archive



The screenshot shows the EUFAR Flight Finder web interface. At the top, there are logos for EUFAR and the Centre for Environmental Data Analysis. Below the title, there is a search instruction: "Search for flights within the EUFAR archive using the parameters below. Refine your search with geographical, temporal, and text search terms. Flights from FAAM, NERC-ARSF, SAFIRE, AWI-Polars, K8-Enduro, and INTA-CASA aircraft and the APEX instrument flown on the DLR aircraft are now included." There are links for "Help" and "Tutorial". A section titled "Choose an index:" contains three buttons: "EUFAR" (selected), "FAAM", and "ARSF". Below this, a message says "Click an item to expand the panel." There are four expandable filter sections: "Geographical Search", "Temporal Filter", "Keyword Search", and "Variable Filter". A checkbox for "Include photography?" is present. At the bottom of the filter section are "Apply Filters" (green), "Clear Filters" (red), and "Export Results" (grey) buttons. At the very bottom, it displays "hits with current selection.", "97 milliseconds for response.", and "Mouse: 47.2495, -9.0726".



What do we want?

Develop a search tool which allows users to perform faceted and free-text search to 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)
- Provide faceted and free-text search
- Handle heterogeneous data
- Work at scale
- Work with different domains/vocabularies

Shared Problem

We think this problem is common among data providers with heterogeneous data. Wanted to use an existing standard to enable more collaboration.

SpatioTemporal Asset Catalog:

- Developed with the Earth Observation community
- Community project
- Reusable solution
- Existing tools and extensions

STAC

Designed to be minimalistic with the core specification requiring only space and time on a latitude, longitude grid in standard Julian calendar.

Features:

- Extension of GeoJSON
- Minimum specification for Geospatial data
- Extensible
 - Ecosystem of tools and extensions
 - Can write your own extensions
- STAC community

STAC for CEDA

STAC:

- Asset
 - *file representing information about the earth in a certain space and time*
- Item
 - *an atomic collection of inseparable data and metadata*
- Collection
 - *a structure to organise and browse Items*



CEDA:

- Asset
 - *a file within the archive*
- Item
 - *group of files that are meaning to a user (depends on the collection)*
- Collection
 - *group of items with a shared vocabulary*

Extracting Metadata

We needed a way to extract the necessary metadata to create the Assets, Items, and Collections required for the STAC catalog.

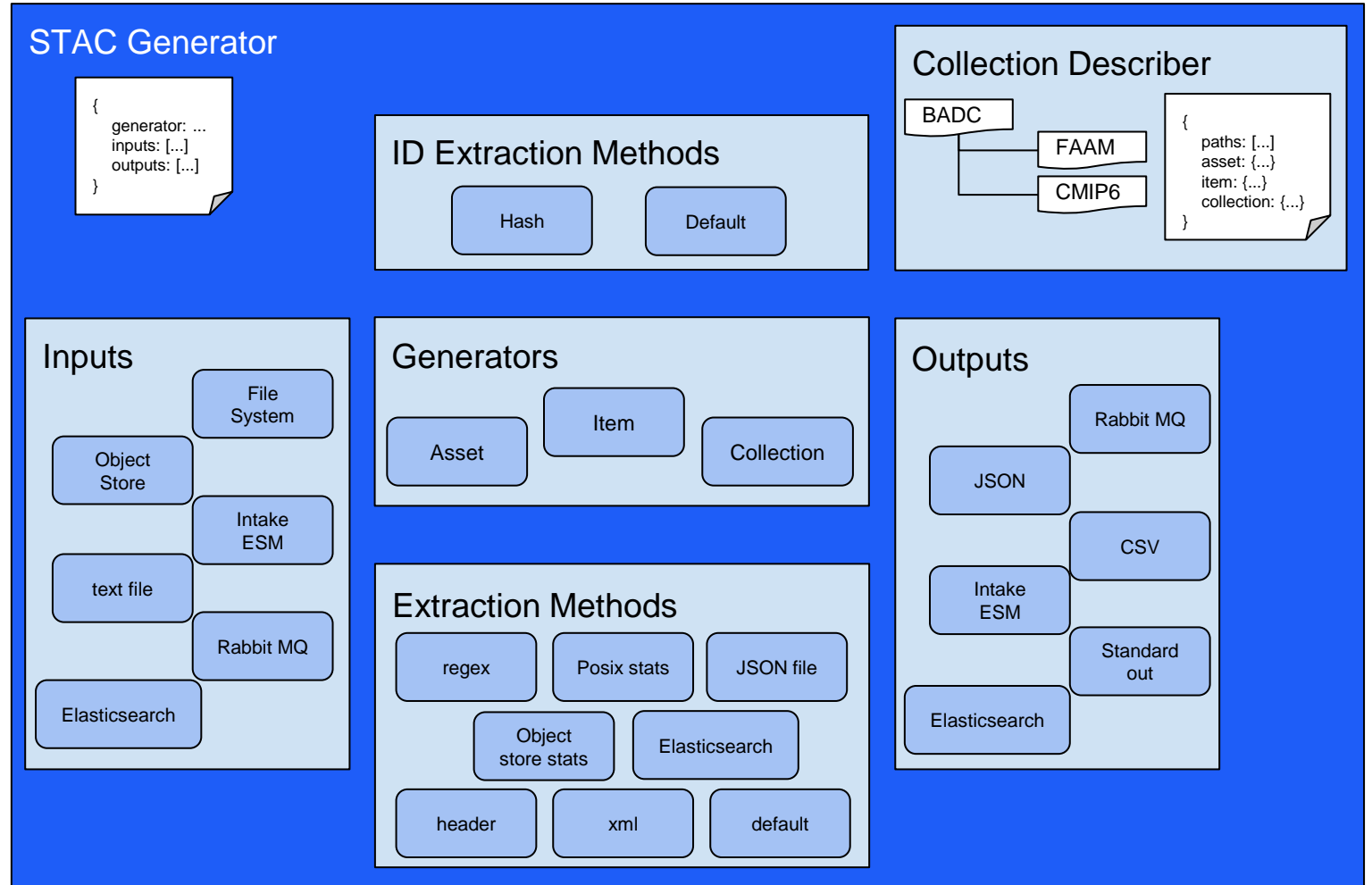
Requirements:

- *Fast*
 - Match data ingestion
- *Flexible*
 - Cope with heterogeneous data
 - Extract collection specific facets
- *Scaleable*
 - Work across the archive

STAC Generator

Plugin Architecture

- Generator
- Inputs
- Outputs
- Extraction methods
 - ID
 - Facets



Generator Configuration

```
{  
  generator: asset  
  
  inputs:  
  - name: file_system  
    path: /badc/cmip6/data  
  
  outputs:  
  - name: elasticsearch  
    connection_kwargs:  
      hosts: ["es9.ceda.ac.uk:9200"]  
      use_ssl: false  
      verify_certs: false  
      ssl_show_warn: false  
  index:  
    name: stac-assets  
}
```

STAC Generator

```
{  
  generator: ...  
  inputs: [...]  
  outputs: [...]  
}
```

ID Extraction Methods

Hash Default

Collection Descriptor

BADC FAAM CMIP6

```
{  
  paths: [...]  
  asset: {...}  
  item: {...}  
  collection: {...}  
}
```

Inputs

Object Store File System
text file Intake ESM
Elasticsearch Rabbit MQ

Generators

Asset Item Collection

Extraction Methods

regex Posix stats JSON file
Object store stats Elasticsearch
header xml default

Outputs

JSON Rabbit MQ
Intake ESM CSV
Elasticsearch Standard out

Collection Descriptions

STAC Generator

```
{
  generator: ...
  inputs: [...]
  outputs: [...]
}
```

ID Extraction Methods

Hash

Default

Collection Descriptor

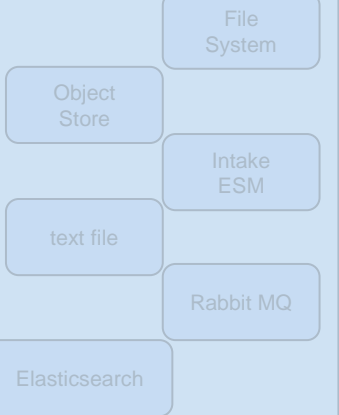
BADC

FAAM

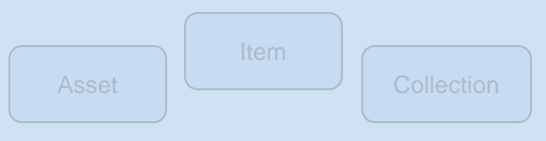
CMIP6

```
{
  paths: [...]
  asset: {...}
  item: {...}
  collection: {...}
}
```

Inputs



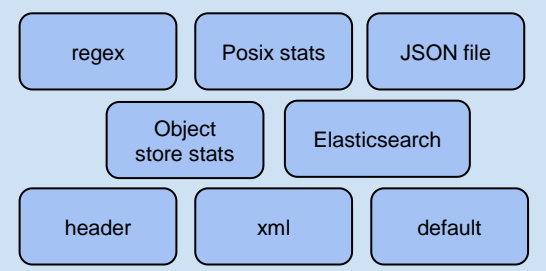
Generators



Outputs



Extraction Methods



```
{
  paths:
    - /badc/cmip6/data/

  asset:
    id:
      method: hash
      terms:
        - uri
      extraction_methods:
        - method: default
      inputs:
        defaults:
          general_data_type: climate models
          permitted_use:
            - academic
            - educational
        - method: regex
      inputs:
        regex: '^(?<mip_era>[^_]+)(?<table_id>[^_]+)(?<source_id>[^_]+)(?<experiment_id>[^_]+)_x(?<activity_id>\d*)i(?<institution_id>\d*)'
        - method: os_stats

  item:
    id:
      method: hash
      terms:
        - mip_era
        - activity_id
        - institution_id
        - table_id
      extraction_methods:
        - method: elasticsearch
      inputs:
        list:
          - mip_era
          - activity_id
          - institution_id
          - table_id

  collection:
    id:
      method: default
      value: cmip6
      extraction_methods:
        - method: elasticsearch
      inputs:
        list:
          - mip_era
          - activity_id
          - institution_id
          - table_id
}
```

Scanning the archive

Reason

- Scale of the archive
- Breadth of the archive

Method

- “All The Other Data” Collection description
- Used batch compute
- Run in series
 - Assets → Items → Collections

```
{
  paths:
  - /badc

  asset:
    id:
      method: hash
      terms:
        - uri
    extraction_methods:
      - method: default
      inputs:
        defaults:
          inspire_theme: Meteorological geographical features
          gemet_topic:
            - climatology
            - meteorology
            - atmosphere
      - method: path_parts
      - method: os_stats

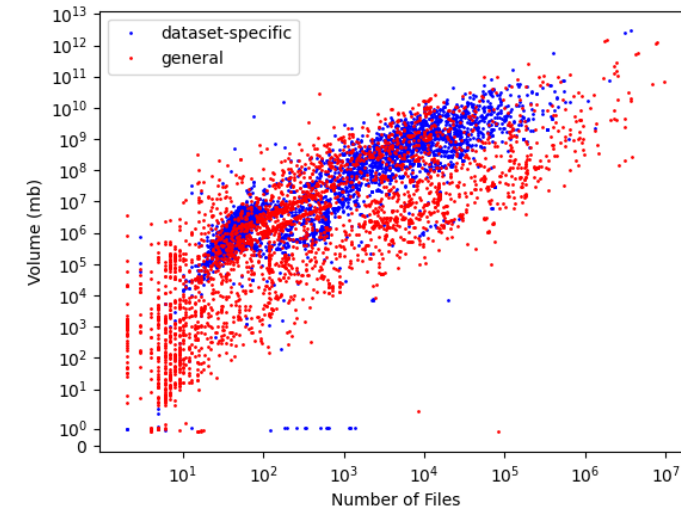
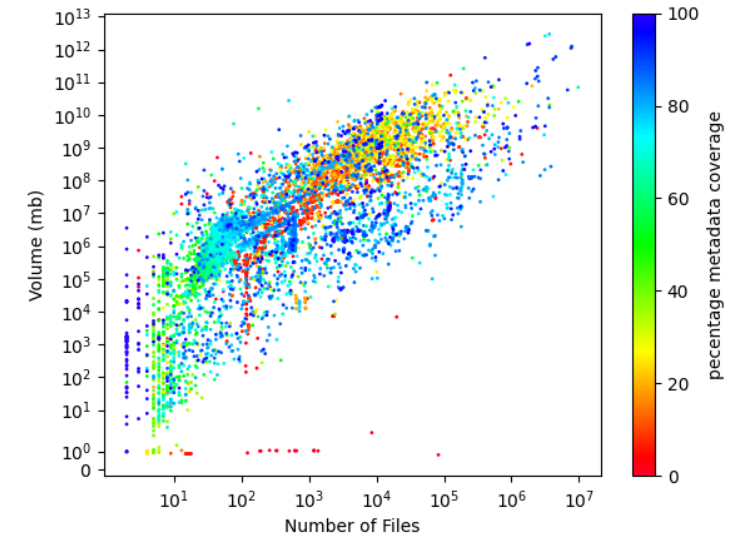
  item:
    id:
      method: hash
      terms:
        - _dir1
        - _dir2
    extraction_methods:
      - method: elasticsearch
      inputs:
        list:
          - general_data_type
          - inspire_theme
          - gemet_topic
          - permitted_use
          - _dir[1-9][0-9]

  collection:
    id:
      method: default
      value: badc
    extraction_methods:
      - method: elasticsearch
      inputs:
        list:
          - general_data_type
          - inspire_theme
          - gemet_topic
}
```

Scanning the archive

Result

- Took
 - 50 nodes
 - ~ 2 weeks
- Generated
 - ~ 325 million assets
 - ~ 4 million items
 - 10 collections
- Properties
 - 100% have basic
 - 12% have dataset specific



Scanning the archive

Next steps

- Iterative approach
- Expand ATOD
 - Add Extraction Methods
- Chop up ATOD
 - Add Collection Descriptions
 - Reduce coverage of ATOD
 - Allows specific extraction

```
https://api.stac.ceda.ac.uk/asset/search?limit=1

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "stac_version": "1.0.0",
      "stac_extensions": [],
      "asset_id": "hb43f6a3d9c1c298b2b48d388315a026",
      "roles": [],
      "item": "00ee2018d58a8d25ce7c28eefa3a348b",
      "bbox": null,
      "properties": {
        "_dir7": "IPF_v2",
        "_dir6": "m",
        "_dir9": "05",
        "extension": ".zip",
        "_dir8": "2018",
        "_dir10": "26",
        "inspire_theme": "orthoimagery",
        "_dir1": "neodc",
        "_dir3": "data",
        "uri": "/neodc/sentinellb/data/EW/L1_GRD/m/IPF_v2/2018/05/26/S1B_EW_GRDM_1SDH_20180526T022500_20180526T022600_011086_014527_5CF2.zip",
        "_dir2": "sentinellb",
        "_dir5": "L1_GRD",
        "_dir4": "EW",
        "collection_id": "neodc",
        "filename": "S1B_EW_GRDM_1SDH_20180526T022500_20180526T022600_011086_014527_5CF2.zip",
        "gemet_topic": "environment",
        "modified_time": "2018-06-26T10:21:09",
        "size": 244845430,
        "categories": [
          "data"
        ],
        "magic_number": "application/zip"
      },
      "links": [...],
      "context": {
        "returned": 1,
        "limit": 1,
        "matched": 323492562
      }
    }
  ]
}
```


Continuous integration

Asset

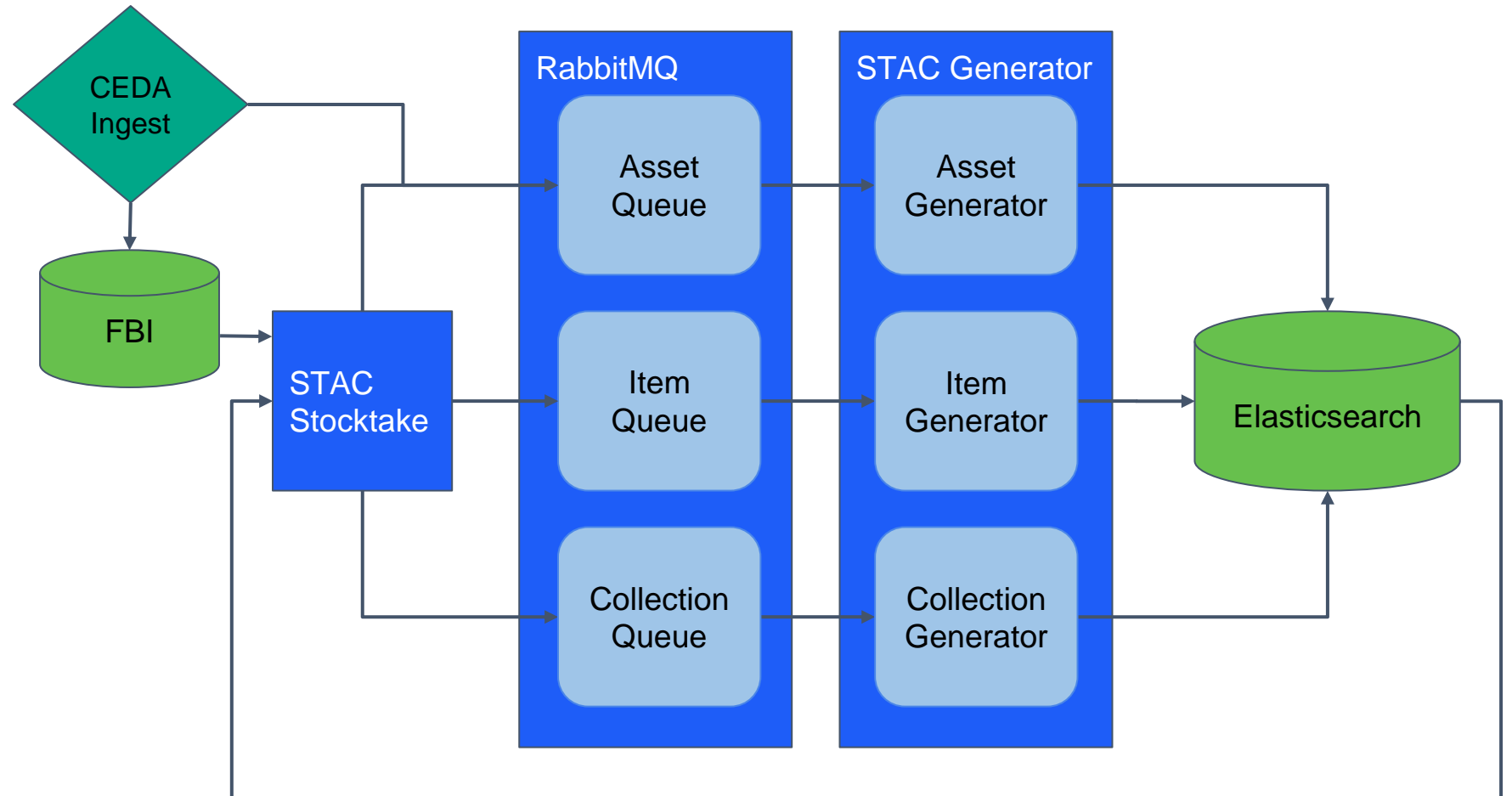
1. Ingest
- ~~2. FBI~~
- ~~3. STAC stocktake~~
4. Rabbit
5. Elasticsearch

Item

1. STAC stocktake
2. Rabbit
3. Elasticsearch

Collection

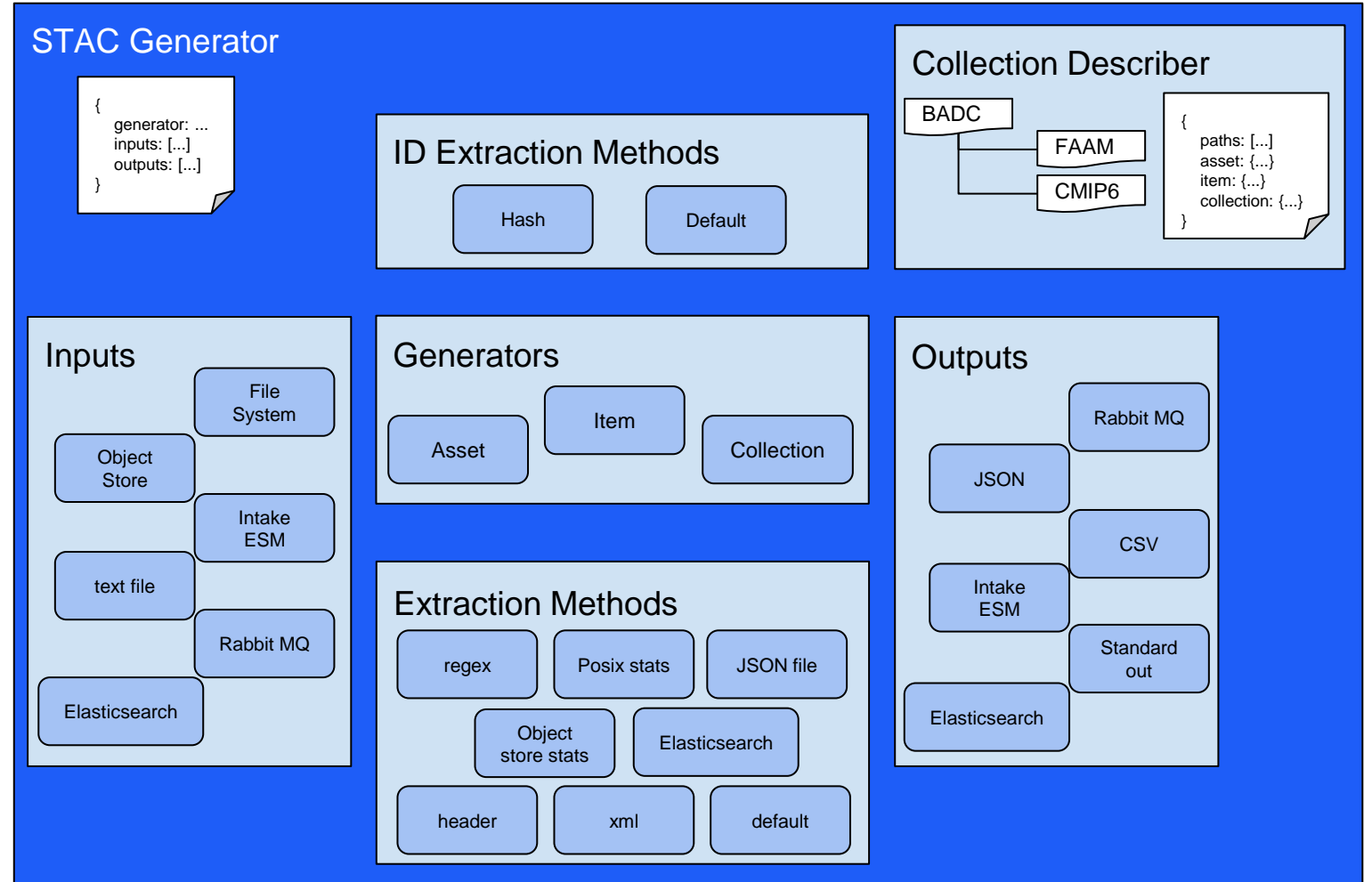
1. STAC stocktake
2. Rabbit
3. Elasticsearch



STAC Generator

What we've found

- Highly configurable
 - Different workflow
 - Collection specific
- Work at scale
 - Fast enough



STAC API

Follows the same principles of the STAC standards, with a basic set of core functions and then a set of extensions so it can adapt to different domains.

Requirements:

- Based on OGC API
- Core
 - '/' - Landing page
 - '/search' - Search items
- Extensions
 - Sort
 - Transactions
 - Filter
 - More ...

STAC API

Elasticsearch backend:

- Based on STAC FastAPI

<https://github.com/cedadev/stac-fastapi-elasticsearch>

CEDA extensions:

- Free-text search

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

- context-collection

<https://github.com/cedadev/stac-context-collections>

- Multi-level search

<https://github.com/cedadev/stac-asset-search>

```
https://api.stac.ceda.ac.uk

{
  "type": "Catalog",
  "id": "stac-fastapi",
  "title": "CEDA STAC API",
  "description": "This is an experimental STAC API server.\n
                The content is subject to change the and there is no guarantee surrounding its uptime.\n",
  "stac_version": "1.0.0",
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/core",
    "https://api.stacspec.org/v1.0.0-beta.4/core",
    "https://api.stacspec.org/v1.0.0-beta.4/item-search",
    "https://api.stacspec.org/v1.0.0-beta.2/item-search#free-text-search",
    ...
  ],
  "links": [
    {
      "rel": "self",
      "type": "application/json",
      "href": "https://api.stac.ceda.ac.uk/"
    },
    {
      "rel": "root",
      "type": "application/json",
      "href": "https://api.stac.ceda.ac.uk/"
    },
    {
      "rel": "data",
      "type": "application/json",
      "href": "https://api.stac.ceda.ac.uk/collections"
    },
    {
      "rel": "conformance",
      "type": "application/json",
      "title": "STAC/WFS3 conformance classes implemented by this server",
      "href": "https://api.stac.ceda.ac.uk/conformance"
    },
    {
      "rel": "search",
      "type": "application/geo+json",
      "title": "STAC search",
      "href": "https://api.stac.ceda.ac.uk/search",
      "method": "GET"
    },
    ...
  ]
}
```

User interface

Web:

- No existing tools

<https://github.com/cedadev/stac-ui>

<https://stac.ceda.ac.uk>

The screenshot displays the CEDA Search web interface. At the top, there are navigation links for 'CEDA', 'Search', and 'Collections'. The main heading is 'CEDA Search'. On the left, there are 'Facets' for filtering results, including 'Date' (with 'Start Date' and 'End Date' fields), 'Bbox' (with 'North', 'West', 'East', and 'South' buttons), and 'Product Version' (with a 'Select...' dropdown). The search bar contains the text 'sentinel*' and a 'Search' button. Below the search bar, it indicates '198 Items'. The search results are displayed as a list of collections, each labeled 'Collection: Sentinel 5'. Each collection entry shows a set of metadata tags: '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', and 'sensor:TROPOMI'.

Clients

Python:

- Extension of pystac

<https://github.com/cedadev/pystac-client>

- Wrapper for ESGF

<https://github.com/cedadev/esgf-stac-client>

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=['Fj3reHsBhuk7QgVbt7P-'],
    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.
```

Conclusion

Question	Answer
Can STAC be used as a general model for environmental data search?	✓
Can we generate the metadata required for STAC?	✓
Are we able to <i>quickly</i> search on the generated metadata?	✓
Does STAC have all the functionality we need?	✗
Can we extend STAC to meet our requirements?	✓
Do we need layers in our framework?	?
Could Temporal and Spatial information be optional?	?
Can we use STAC extensions for specific parts of our archive?	?
Do we need Asset Search?	?



Science and
Technology
Facilities Council

Natural
Environment
Research Council

Thank you!

JASMIN: support@jasmin.ac.uk

CEDA: support@ceda.ac.uk

Twitter - [@cedanews](https://twitter.com/cedanews)

Website - www.ceda.ac.uk

