SpatioTemporal Asset Catalogs (STAC)

Status of this Document

This Request for Comment (RFC) provides information to the NASA Earth Science community. This RFC describes the SpatioTemporal Asset Catalog specification. Distribution of this document is unlimited.

Change Explanation

This document is not a revision to an earlier version.

Copyright Notice

Suggested Citation

D. J. Newman and ESDIS Standards Coordination Office (ESCO). 2023. SpatioTemporal Asset Catalogs (STAC). NASA Earth Science Data and Information System Standards Coordination Office. https://doi.org/10.5067/DOC/ESCO/ESDS-RFC-044v1.

Abstract

This document designates the SpatioTemporal Asset Catalog (STAC) specification as a discovery and catalog standard for NASA Earth science data systems.

Table of Contents

1 Introduction

The SpatioTemporal Asset Catalogs (STAC) specification [1] is a common language to describe geospatial information, so spatiotemporal assets can more easily be leveraged, indexed, and discovered. It is a standardized way to expose spatiotemporal data and provides a common structure for describing and cataloging spatiotemporal assets. With wide adoption of this common standard, different data repositories can be compared and aggregated, tooling and libraries can be applied in the same manner against multiple repositories and user interfaces can be reused. The STAC Specification consists of 4 semi-independent specifications: item, collection, catalog and Application Programming Interface (API).

STAC defines a spatiotemporal asset as any file (science data, browse image, ancillary data, etc.) that represents information about the earth captured in a certain space and time.

The goal is for all providers of spatiotemporal assets (Imagery, Point Clouds, Data Cubes, Full Motion Video, etc) to expose their data within STAC catalogs, so that software packages can continue to support STAC without modification whenever a new data set or API is released.

1.1 Motivation

STAC is a lightweight, JSON-based language that is asset-oriented, self-navigable and tailored towards the spatial and temporal domain. All of these qualities make it ideal for a cloud-based catalog solution. STAC has rapidly become the de facto standard for cloud-based catalog representation, both in terms of static catalog portrayals and catalog APIs. Whilst there is nothing within the specification that deals with the challenges of data discovery and access in the cloud per se, STAC solves a number of problems with online asset catalogs in general and arose at the same time that earth science communities started thinking about how best to discover and access data on the cloud. The reality of the situation is that NASA has aligned with STAC as a key part of their cloud migration strategy. NASA's Earth Observing System Data and Information System (EOSDIS), for example, defines 'cloud enabled data' as being discoverable via STAC (ESDIS memo: 423-MEMO-004).

The uptake of STAC as a catalog standard is demonstrated by the amount of tooling and library support that has evolved since its first release in May of 2021. That support is detailed in later sections of this document.

NASA EOSDIS provides a STAC API covering its entire data holdings at https://cmr.earthdata.nasa.gov/stac as well as a cloud-based catalog API at https://cmr.earthdata.nasa.gov/cloudstac. NASA EOSDIS also leverages STAC catalogs for describing transient collections of assets like those produced by the Harmony API that creates on-demand products, asynchronously and portrays them to the end user as STAC catalogs that can be consumed by a variety of libraries and tooling (see section 1.2.3).

1.2 Evidence of Implementation

The following examples demonstrate implementations of STAC across a diverse set of stakeholders.

1.2.1 STAC Catalogs

STAC catalogs are a succinct and standardized way of conveying the location of results of processing operations to a user. Harmony, NASA EOSDIS' Open Geospatial Consortium Coverages API, leverages static STAC catalogs to convey the location of cloud-based, processing request assets to users [2]. STAC catalogs are also used to portray the contents of static data archives. A list of publicly available STAC catalogs can be found at https://stacindex.org/catalogs?type=static.

1.2.2 STAC APIs

NASA EOSDIS' Common Metadata Repository (CMR) has a STAC API to enable discovery of its collections, granules and assets [3]

All USGS Landsat data stored in the AWS cloud platform are exposed via a STAC API [4]

Microsoft's Planetary Computer [5] provides a STAC API with access to imagery archives of Landsat, MODIS, Sentinel, and more [6]

A list of publicly available STAC APIs can be found at https://stacindex.org/catalogs?type=api

1.2.3 STAC Software

STACBrowser [7] is a web application that can be used to discover and visualize catalogs, collections, items and assets portrayed as either a static STAC catalog or a dynamic STAC catalog via a STAC API.

Intake [8] is a lightweight python package for loading and sharing data in data science projects. Intake supports a number of data sources such as xarray. In the context of STAC, Intake has been extended to intakeSTAC to allow users to reuse and share ephemeral STAC catalog files as convenient and portable collections of bespoke data.

IntakeSTAC [9] is an Intake data source, utilizing STAC catalogs, that provides an opinionated way for users to load assets from STAC catalogs into the scientific Python ecosystem. IntakeSTAC builds on the traditional intake-xarray plugin for multi-dimensional array data and supports several file formats including GeoTIFF, netCDF, GRIB, and OpenDAP.

stac-server [10] is an implementation of the STAC API specification for searching and serving metadata for geospatial data written in NodeJS.

QGIS [11] plugins are available for using STAC catalogs and APIs as data sources for QGIS.

2 STAC Structure

The STAC Specification consists of 4 semi-independent specifications. Each can be used alone, but they work best in concert with one another.

STAC Item is the core atomic unit, representing a single spatiotemporal asset as a GeoJSON feature plus datetime and links.

```
{
    "stac_version": "1.0.0",
    "type": "Feature",
    "id": "20201211_223832_CS2",
    "bbox": [],
    "geometry": {},
    "properties": {},
    "collection": "simple-collection",
    "links": [],
    "assets": {}
}
```

STAC Catalog is a simple, flexible JSON file of links that provides a structure to organize and browse STAC Items. A series of best practices helps make recommendations for creating real world STAC Catalogs.

```
{
    "stac_version": "1.0.0",
    "type": "Catalog",
    "id": "20201211_223832_CS2",
    "description": "A simple catalog example",
    "links": []
}
```

STAC Collection is 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.

```
{
    "stac_version": "1.0.0",
    "type": "Collection",
    "license": "ISC",
    "id": "20201211_223832_CS2",
    "description": "A simple collection example",
    "links": [],
    "extent": {},
    "summaries": {}
}
```

A simple STAC structure might look like this:

- catalog (root)
    - catalog
        - catalog
            - item
                - asset
            - item
                - asset
            - item
                - asset
                - asset
    - catalog
        - item
            - asset

STAC API provides a RESTful endpoint that enables search of STAC Items, specified in OpenAPI [12], following OGC's WFS 3 [13].

The core of the STAC API specification is a single endpoint:

`/stac/search` (for example: https://cmr.earthdata.nasa.gov/stac/PODAAC/search)

It takes a JSON object that can filter on date and time:

```
{
  "bbox": [5.5, 46, 8, 47.4],
  "time": "2018-02-12T00:00:00Z/2018-03-18T12:31:12Z"
}
```

This request instructs the server to return all the catalog items it has that are from the second half of March, 2018 and that intersect with this area:

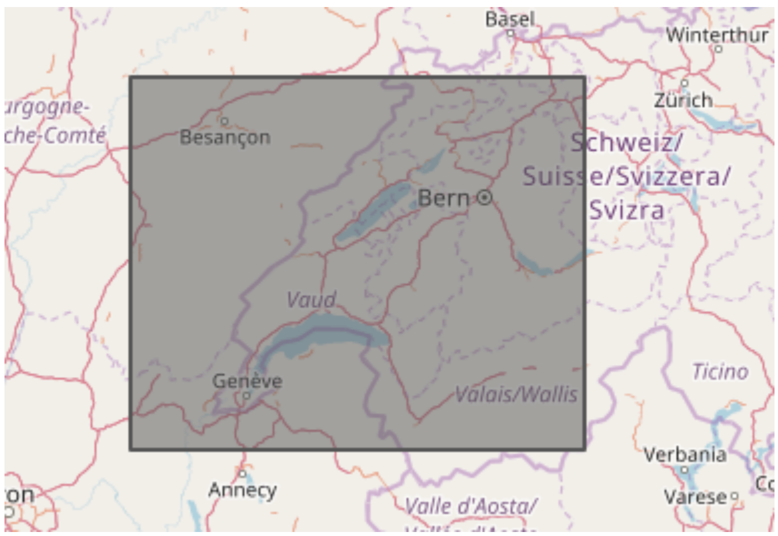

Figure 1: Spatial Extent of STAC API query

The server could respond with the following in JSON format,

```
{
  "type": "FeatureCollection",
  "stac_version": "1.0.0",
  "numberMatched": 1,
  "numberReturned": 1,
  "features": [
    {
      "type": "Feature",
      "id": "sg178_Oct2019_data_L2",
      "stac_version": "1.0.0",
      "stac_extensions": [],
      "collection": "SEAGLIDER_GUAM_2019.vV1",
      "geometry": {
        "type": "Polygon",
        "coordinates": [[ [-180,-85.5],[180,-85.5], [180,89.5],[-180,89.5],[-180,-85.5]]]
      },
      "bbox": [-180, -85.5,180,89.5],
      "links": [
```

```
    ...,
    {
      "rel": "via",
      "href": "https://cmr.earthdata.nasa.gov/search/concepts/G2188230253-POCLOUD.json"
    },
    {
      "rel": "via",
      "href": "https://cmr.earthdata.nasa.gov/search/concepts/G2188230253-POCLOUD.umm_json"
    }
  ],
  "properties": {
    "datetime": "1900-01-01T00:00:00.000Z",
    "start_datetime": "1900-01-01T00:00:00.000Z",
    "end_datetime": "2018-12-31T00:00:00.000Z"
  },
  "assets": {
    "data": {
      "title": "Download sg178_Oct2019_data_L2.nc",
      "href": "https://archive.podaac.earthdata.nasa.gov/podaac-ops-cumulus-
protected/SEAGLIDER_GUAM_2019/sg178_Oct2019_data_L2.nc"
    },
    "metadata": {
      "href": "https://cmr.earthdata.nasa.gov/search/concepts/G2188230253-POCLOUD.xml",
      "type": "application/xml"
    }
  }
 }
}
```

2.1 Catalog

A STAC catalog is a representation of the whole or a subset of a spatiotemporally defined inventory. A catalog can contain catalogs and/or collections and/or STAC items. STAC catalogs can represent spatial areas of interest, temporal areas of interest, disciplines, campaigns and tangible entities such as providers.

Catalogs can be static or dynamic in their nature. Either can be synthesized using the STAC Catalog, Item and Collection specifications.

A static catalog is one that is implemented as a set of flat files on a web server or an object store like S3 [14] or Google Cloud Storage [15]. Static catalogs could represent the results of on-demand processing or event-based assets.

A dynamic catalog is one that generates its responses dynamically, generally backed by some sort of server and can be used to portray the results of STAC API queries. An example of this is NASA's EOSDIS STAC API which is part of the Common Metadata Repository at https://cmr.earthdata.nasa.gov/stac

2.2 API

A STAC API is a dynamic version of a SpatioTemporal Asset Catalog. It can be used to retrieve a STAC Catalog, Collection, Item, or STAC API ItemCollection objects from various endpoints.

The API can be implemented in compliance with the OGC API - Features standard (OAFeat is a shorthand) [13]. In this case, STAC API can be thought of as a specialized Features API to search STAC catalogs, where the features returned are STAC Item objects, that have common properties, links to their assets and geometries that represent the footprints of the geospatial assets.

2.3 Relevant Extensions

STAC is an extensible specification with a minimal core specification. This is intentional in order to attract a large usage base, allowing that base to implement their own specializations in extensions. Some extensions that are relevant to the EOSDIS domain are as follows,

2.3.1 Raster

An item can describe assets that are rasters of one or multiple bands with some information common to them all (raster size, projection) and also specific to each of them (data type, unit, number of bits used).

For more information: https://github.com/stac-extensions/raster

2.3.2 Grid

The purpose of the Grid Extension is to provide fields related to gridded data products. This can enable use cases such as colocation, where items may be geolocated using different coordinate reference schemes.

For more information: https://github.com/stac-extensions/grid

2.3.3 Label

This extension supports using labeled areas of interest with Machine Learning models, particularly training data sets, but can be used in any application where labeled AOIs are needed.

For more information: https://github.com/stac-extensions/label

2.3.4 Scientific

This extension adds the ability to indicate from which publication data originates and how the data itself should be cited or referenced. Overall, it helps to increase reproducibility and citability.

For more information: https://github.com/stac-extensions/scientific

2.3.5 Storage

STAC has become the de facto cataloging standard for cloud-based assets. This extension adds fields to STAC Item and Asset objects, allowing for details related to cloud storage access and costs to be associated with a STAC Item.

For more information: https://github.com/stac-extensions/storage

3 Interoperability and Applicability Considerations

STAC is an asset-driven capability designed with spatial and temporal notions of data at the forefront. Any catalog that is not centered around the above characteristics is not a good fit for a STAC implementation. There are categories of data within EOSDIS which should not be surfaced in STAC catalogs for this reason but, in general, EOSDIS data is a very good fit for STAC.

Whilst the STAC specification has gained significant traction in cloud-based inventories there is nothing in the core specification geared towards cloud-hosted assets. The evolution of assets as first-class citizens has propelled STAC into the position of the catalog implementation of choice on the cloud. As such, STAC should not be considered as solely a cloud-based asset solution. For example, CMR supports both cloud and cloud plus on-premises STAC APIs at https://cmr.earthdata.nasa.gov/cloudstac and https://cmr.earthdata.nasa.gov/stac respectively.

The adoption of HATEOAS [17] and linked data by STAC, coupled with the concept of recursive catalogs lends itself to federated discovery in much the same way as OpenSearch [18] did. Any cloud-based federated system may want to adopt STAC as a means of linking loosely-coupled archives using the two-step discovery process [19].

CMR STAC demonstrates that STAC catalogs, collections and items map well to the EOSDIS concepts of providers, and the Universal Metadata Model (UMM) concepts of UMM-C (collections) and UMM-G (granules).

4 Future versions of the specification

STAC enhancements are primarily driven by extensions but some additions to the core specification have been mooted. For example, collection level discovery akin to the OGC Features API have been discussed.

5 References

5.1 Normative references

[1] STAC Specification https://stacspec.org

5.2 Informative references

[2] EOSDIS Harmony https://harmony.earthdata.nasa.gov/

[3] Common Metadata Repository https://cmr.earthdata.nasa.gov/search

[4] Landsat STAC API https://landsatlook.usgs.gov/stac-server

[5] Microsoft's Planetary Computer https://planetarycomputer.microsoft.com/

[6] Planetary Computer catalogs https://planetarycomputer.microsoft.com/catalog

[7] STACBrowser https://radiantearth.github.io/stac-browser/#/

[8] Intake https://intake.readthedocs.io/en/latest/

[9] IntakeSTAC https://github.com/intake/intake-stac

[10] stac-server https://github.com/stac-utils/stac-server

[11] QGIS STAC Plugin https://github.com/stac-utils/qgis-stac-plugin

[12] OpenAPI https://www.openapis.org/

[13] OGC Web Feature Service Version 3
https://www.ogc.org/projects/initiatives/wfs3hackathon

[14] AWS Simple Storage Service https://aws.amazon.com/pm/serv-s3

[15] Google Cloud Storage https://cloud.google.com/storage

[16] OGC Features API https://ogcapi.ogc.org/features/

[17] Hypermedia as the Engine of Application State https://en.wikipedia.org/wiki/HATEOAS

[18] CEOS OpenSearch RFC
https://www.earthdata.nasa.gov/esdis/eso/standards-and-references/ceos-opensearch

[19] Two step discovery process
https://ceos.org/document_management/Working_Groups/WGISS/Interest_Groups/OpenSearch/CEOS-OPENSEARCH-BP-V1.2.pdf

6 Authors' Addresses

Douglas J. Newman, NASA GSFC

Email: douglas.j.newman@nasa.gov

ESDIS Standards Coordination Office (ESCO) , NASA GSFC

Email: esco-staff@lists.nasa.gov

https://www.earthdata.nasa.gov/esdis/esco

Appendix A

Acronym	Description
AOI:	Area of interest. Either spatial or temporal or both
API:	Application Programming Interface
EOSDIS:	Earth Observing System Data and Information System
ESDS:	Earth Science Data Systems
GeoJSON:	Geospatial extension for JavaScript Object Notation
GeoTIFF:	Geospatial Tagged Image File Format
GRIB:	GRIdded Binary data format

HATEOAS:	Hypermedia as the Engine of Application State
JSON:	JavaScript Object Notation
MODIS:	Moderate Resolution Imaging Spectroradiometer
netCDF:	Network Common Data Form data format
OAFeat:	OGC Features API
OGC:	Open Geospatial Consortium
OpenAPI:	Standard Machine-readable API description
OpenDAP:	Open-source Project for a Network Data Access Protocol
REST:	Representational state transfer
RFC:	Request for comment
S3:	Amazon Web Services' Simple Storage System
SAR:	Synthetic Aperture Radar
STAC:	Spatio-Temporal Asset Catalog
USGS:	United States Geological Survey
WFS:	OGC Web Feature Service