



developmentSEED

On-the-fly raster visualizations leveraging STAC metadata standards

Sadiq Mohiuddin, Hanbyul Jo,
Aimee Barciauskas, Brian Freitag

1 Where it started: Project VEDA

2 STAC + Render Extension

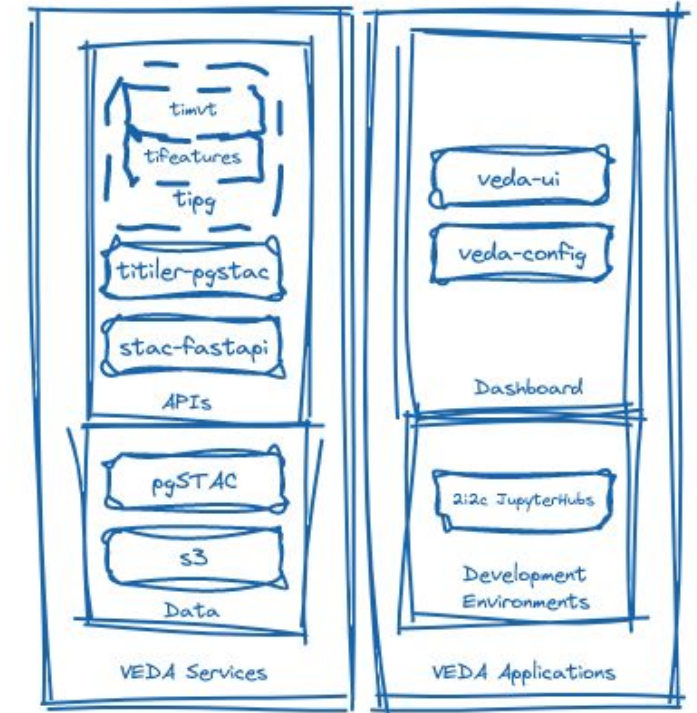
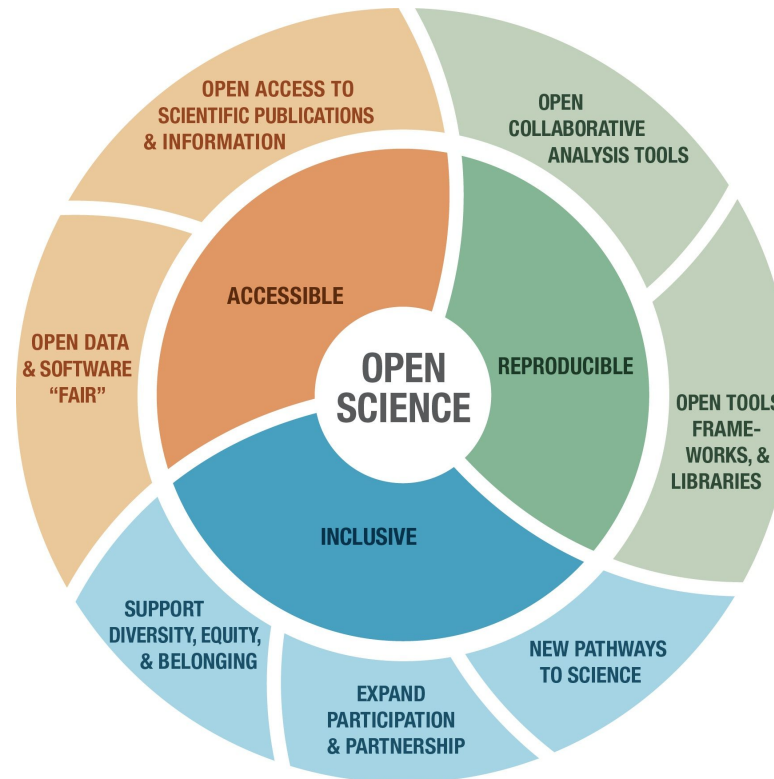
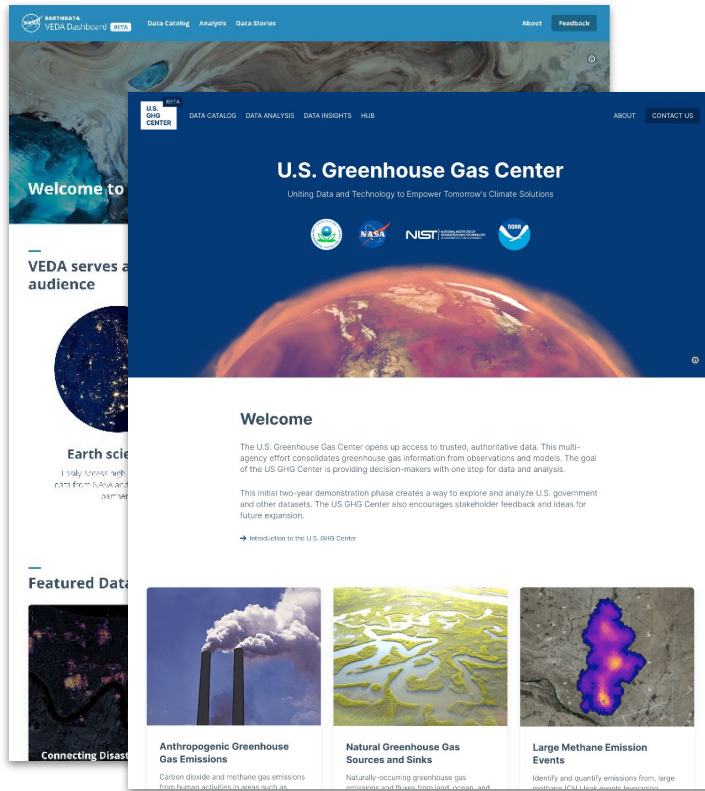
3 Render Extension Use-cases

4 Resources

Project VEDA: Visualization, Exploration & Data Analysis

Where the need for the STAC Render Extension started

VEDA is for people who are...

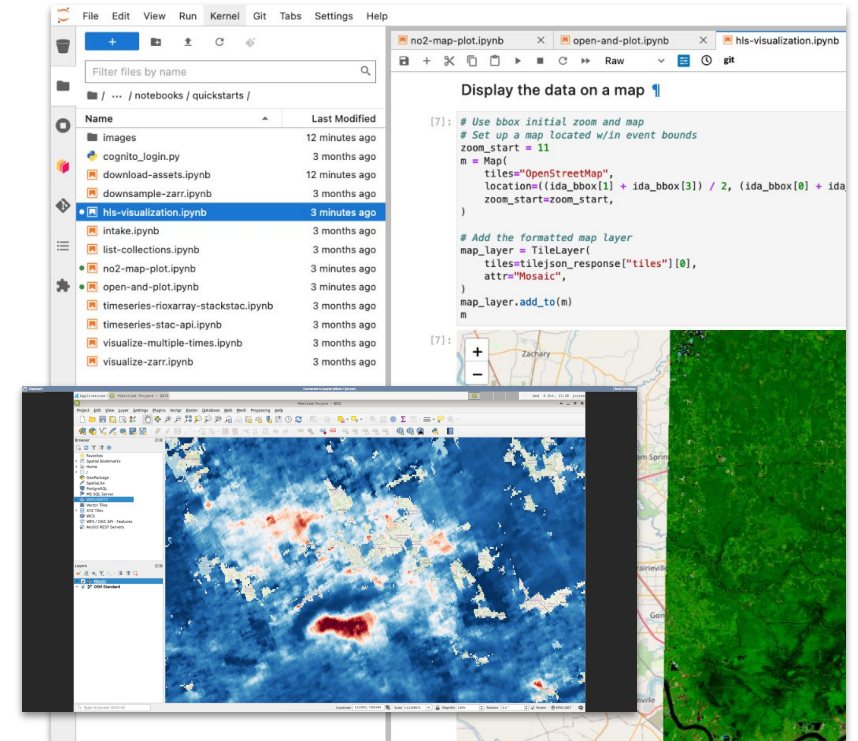
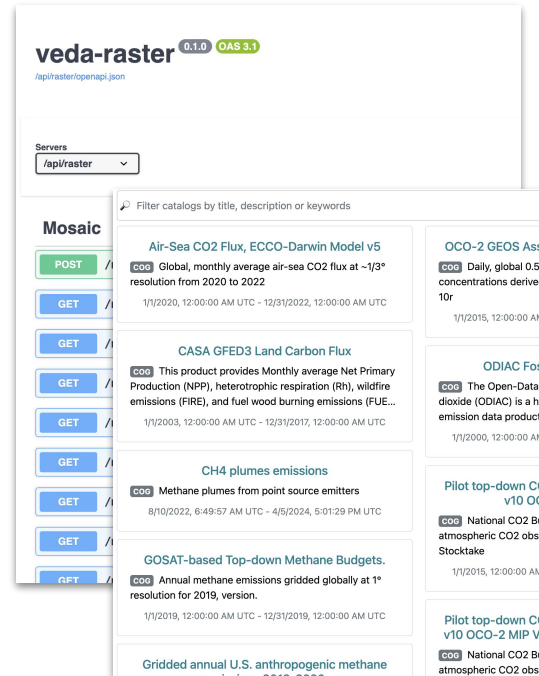
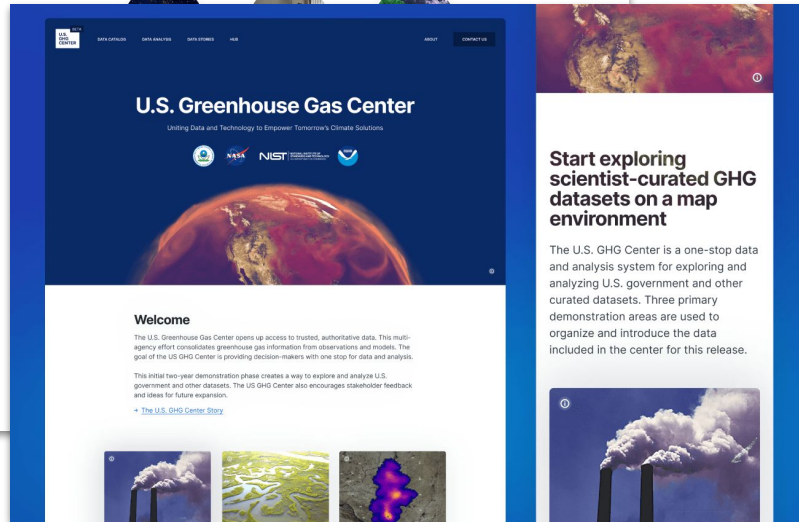
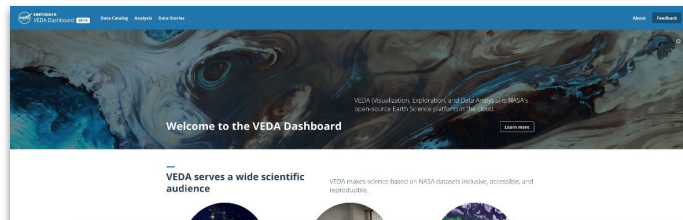


Doing science with **Earth data**

Wanting to widen **participation** in your science

Looking for examples and **open-source software** to get started

VEDA integrates existing community standards for tool-agnostic, human- and machine-friendly Earth data science

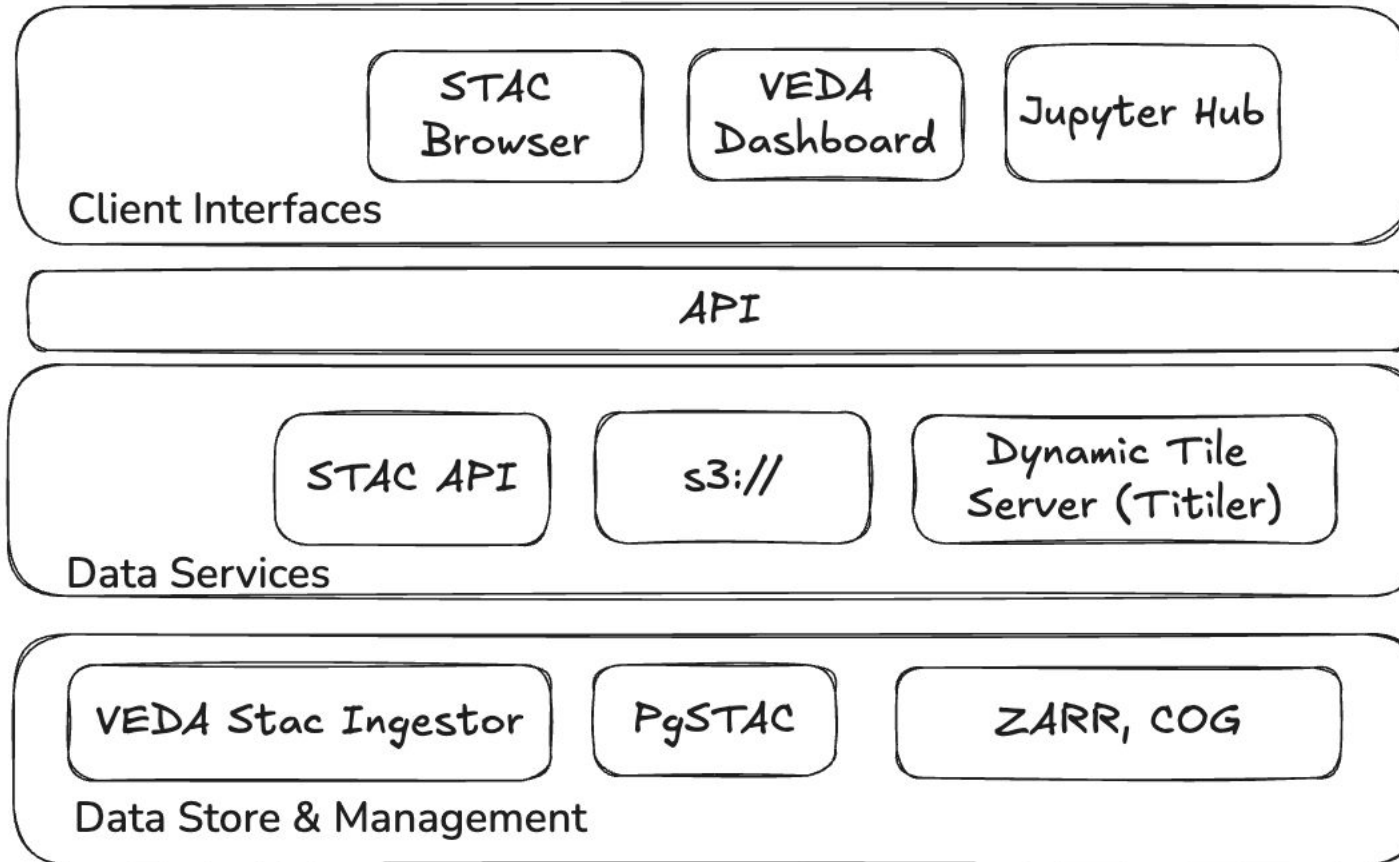


Science stories, catalog, data browsing

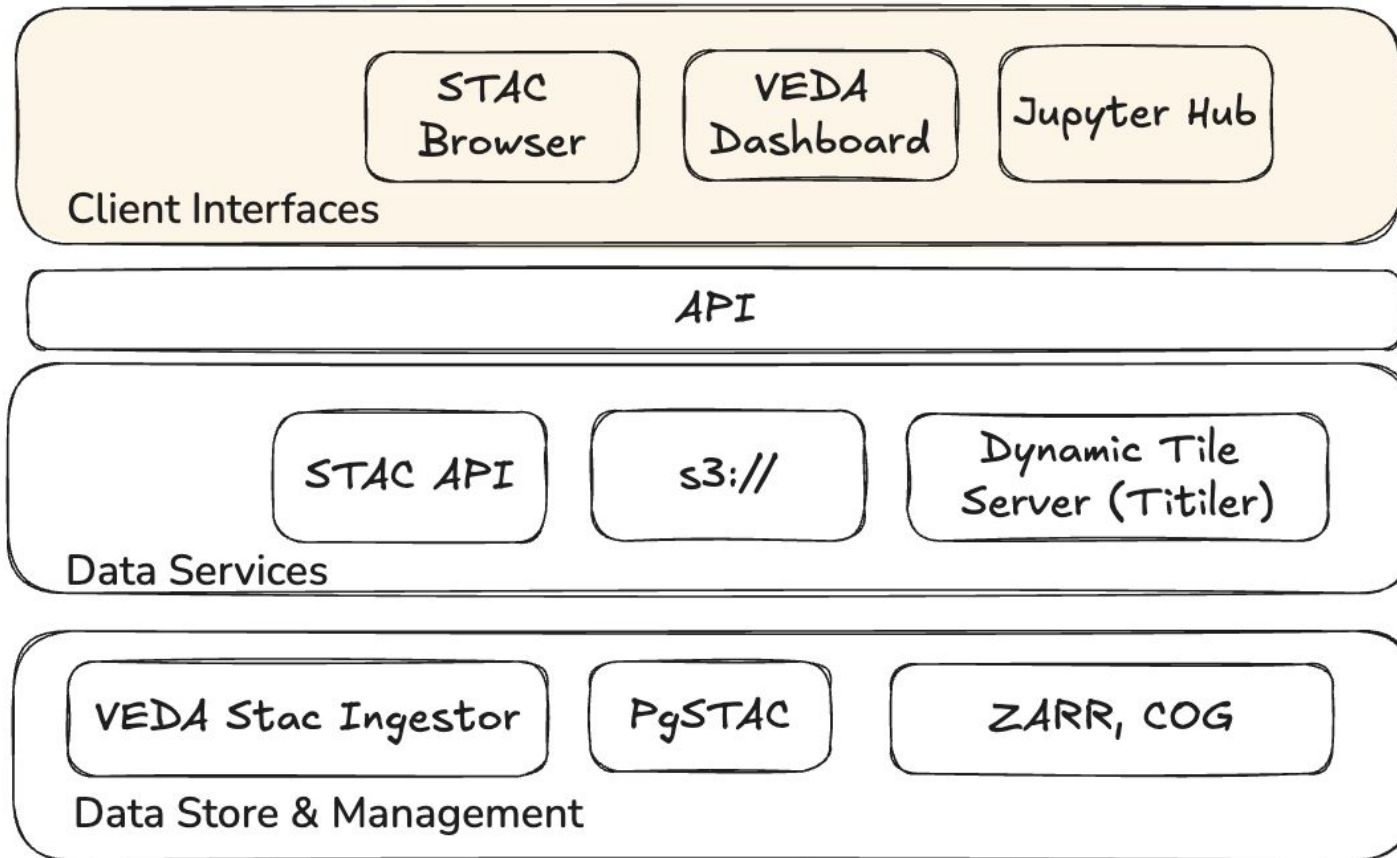
Cloud storage (S3), STAC and OGC APIs

JupyterHub and cloud computing resources

VEDA Systems Architecture



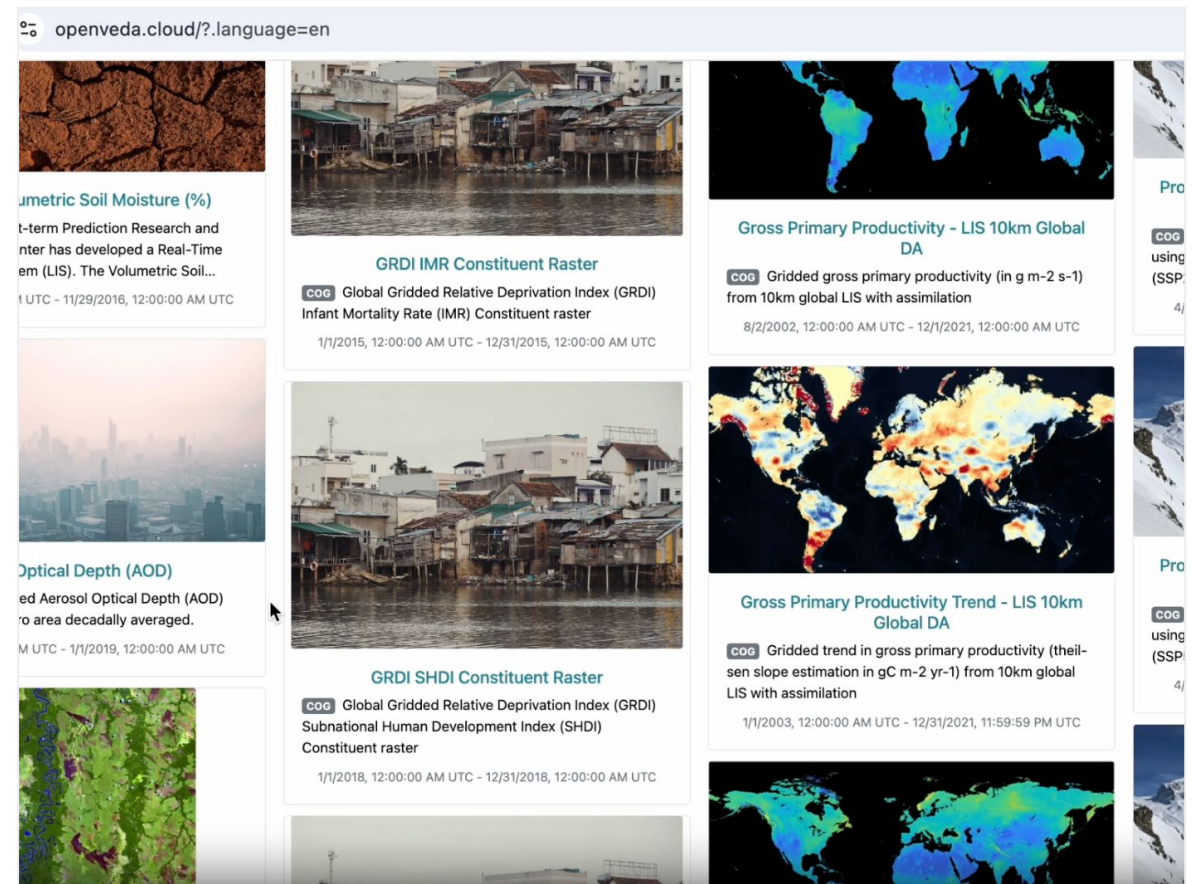
VEDA Systems Architecture



VEDA Client interface – STAC Browser

<https://openveda.cloud/>, Original work from Radiant Earth: <https://github.com/radiantearth/stac-browser>

- Web application for browsing static catalogs
- Attempts to surface all included datasets in the catalog in a user-centric way
- Great for data browsing and discovery



VEDA Client Interface – Data needs context

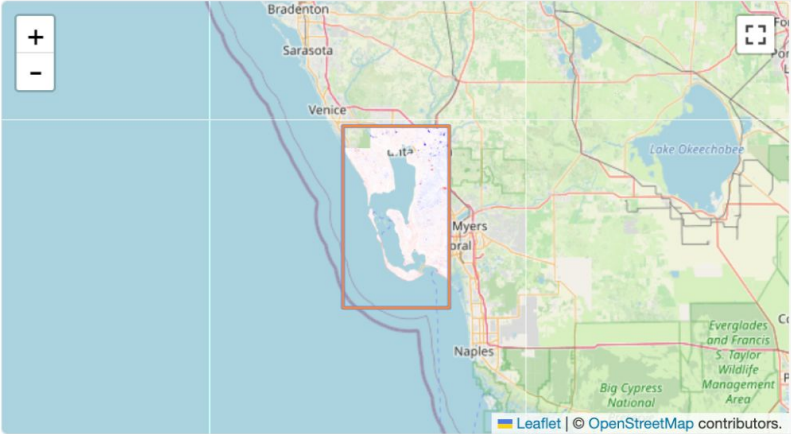
HLS-derived entropy difference for Assessing impacts from Hurricane Ian SS

in [openveda.cloud](#) ↑ Up 📖 Browse 🔍 Search

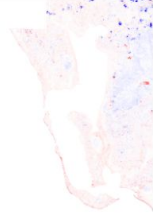
Description
Entropy difference calculated from Harmonized Landsat and Sentinel-2 Data for September 5th and September 30th, 2022

License CC0-1.0

Temporal Extent 9/30/2022, 12:00:00 AM UTC - 9/30/2022, 11:59:59 PM UTC

Map
+
-

Map
Thumbnails

Item
🔍 Show Filters


entropy_difference_2022-09-30
GeoTiff
9/30/2022, 12:00:00 AM UTC - 9/30/2022, 11:59:59 PM UTC

Asset in Items
▼ Default COG Layer

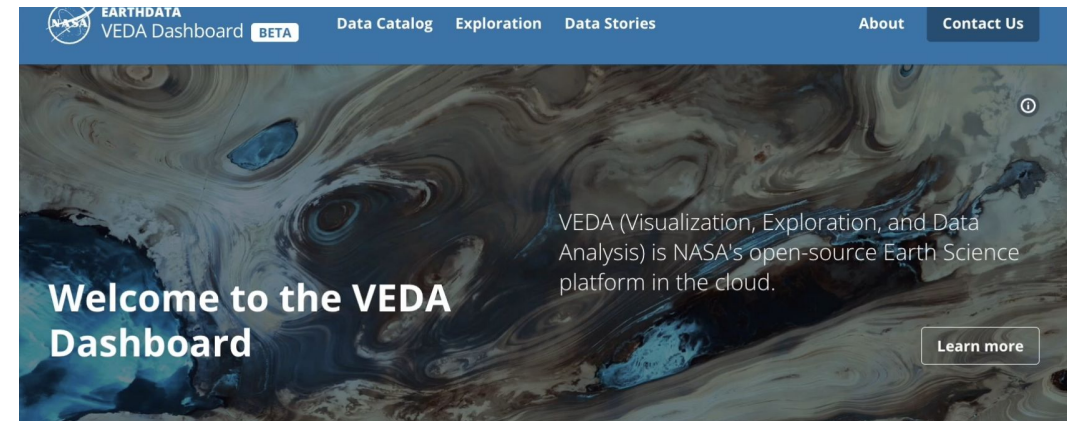
Asset

Ex. HLS-Derived NDVI difference dataset for post hurricane Ian

VEDA Client Interface – VEDA Dashboard

VEDA Dashboard is a web platform with curated datasets for scientists that

- Accurately represents their work and the context of it to the audience including scientists, policy makers, and general audience.
- Becomes a trusted source of information and data
- Is easy to set up, customize, and maintain curation without paid subscription



VEDA serves a wide scientific audience

VEDA makes science based on NASA datasets inclusive, accessible, and reproducible.

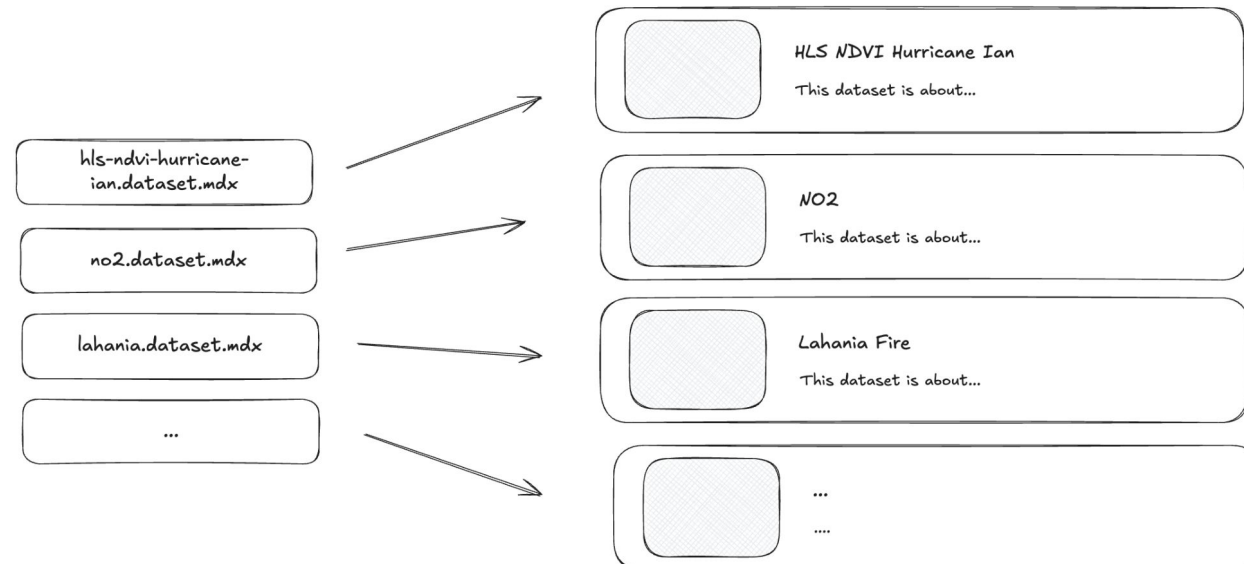


<https://earthdata.nasa.gov/dashboard/>

<https://github.com/nasa-impact/veda-config>

VEDA Dashboard curation – File-based ‘CMS’

- One dataset has one configuration file.
- The configuration file is consist of metadata of the dataset and the contents (overview) of the dataset
- Each ‘dataset’ has a group of the layers (collections) to visualize



VEDA Dashboard example dataset

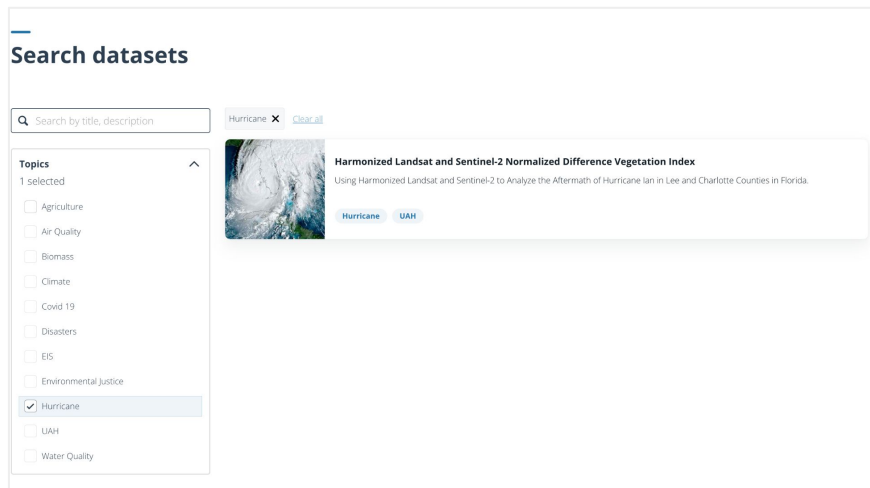
The screenshot shows the GitHub interface for the 'veda-config' repository. At the top, the repository name 'veda-config' is displayed as 'Public'. Action buttons include 'Edit Pins', 'Unwatch 25', 'Fork 36', and 'Star 13'. Below this, the current branch is 'develop'. A 'Go to file' button and a 'Code' button are visible. The main content area features a merge pull request by 'maxrjones' from 'NASA-IMP...' with a commit hash '1250da7' from '3 weeks ago'. Below the pull request is a table of files and folders with their respective commit messages and dates.

.devcontainer	Add codespaces setup	last year
.github	Update Node version to 20	3 months ago
.veda	Update veda-ui to v5.5.0	last month
admin	Remove thematic area related docs	last year
datasets	Update datasets/global-reanalysis-d...	last month
docs	Move documentation to veda-ui	last year
overrides	Add additional github repos in about ...	last year
static/graphics	Include VEDA logo and update readme	last year
stories	Change pubDate for stories (#422)	2 months ago

About
Configuration template for the Dashboard Evolution Project
visex.netlify.app/
Readme
View license
Activity
Custom properties
13 stars
25 watching
36 forks
Report repository

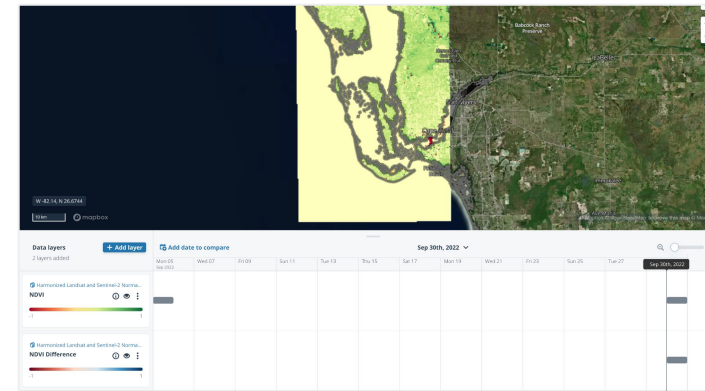
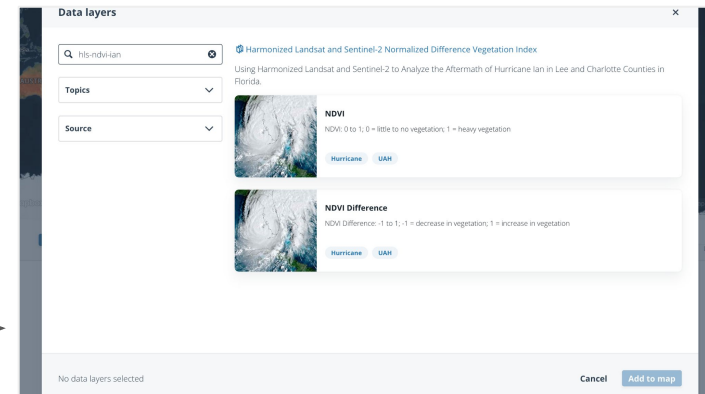
Releases 19
v0.15.2 **Latest**
on May 22

HLS NDVI Post-Ian Hurricane Dataset on Dashboard

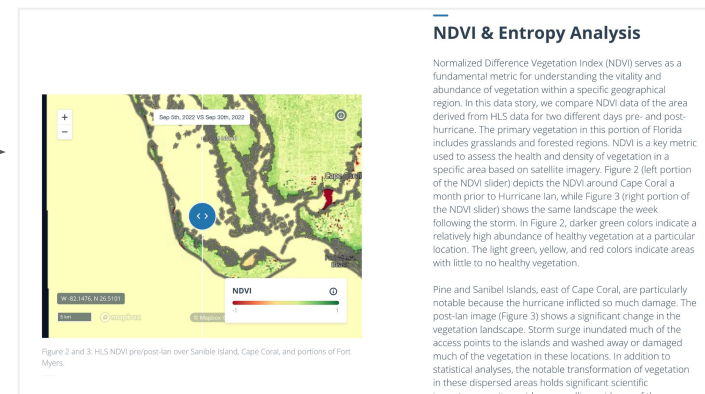


Catalog page

HLS NDVI dataset is consist of two collections (NDVI, NDVI Difference)



Explore page



Story page

How one dataset file for VEDA Dashboard looks like

- YAML for metadata
 - The metadata needed to populate the UI components (title, description)
 - The metadata needed to fetch stac collection and to visualize collection (stac collection id, colormap, rescale, band index...)
- MDX (Markdown + JSX) for contents:
Contents for overview page

```
Example.dataset.mdx

layers: Layers[]
layers1:
  title: Example Collection
  description: A great example collection
  ...
  stacCol: example-collection-id
  sourceParams:
    colormap_name: 'viridis'
  ...

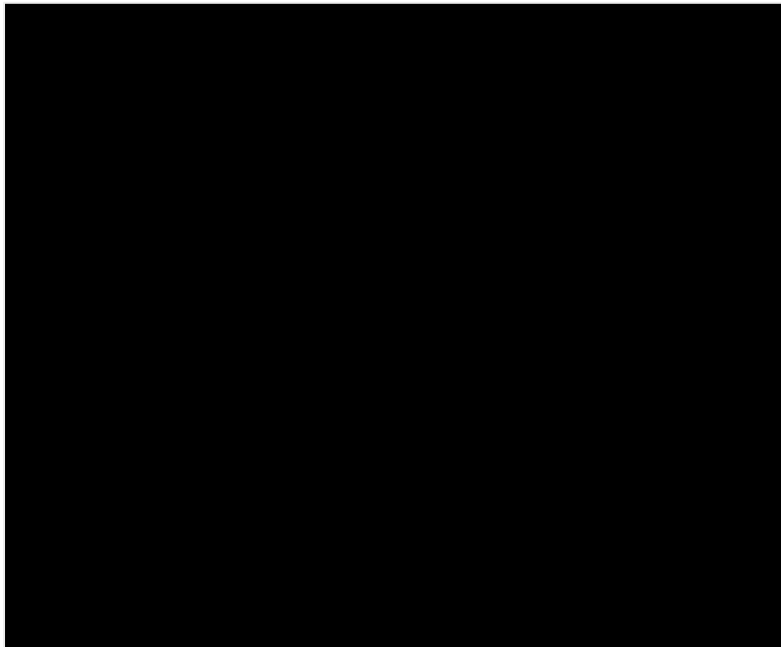
---

### Example Collection

Example Collection is a great dataset
```

Metadata (query parameters) needed for visualization

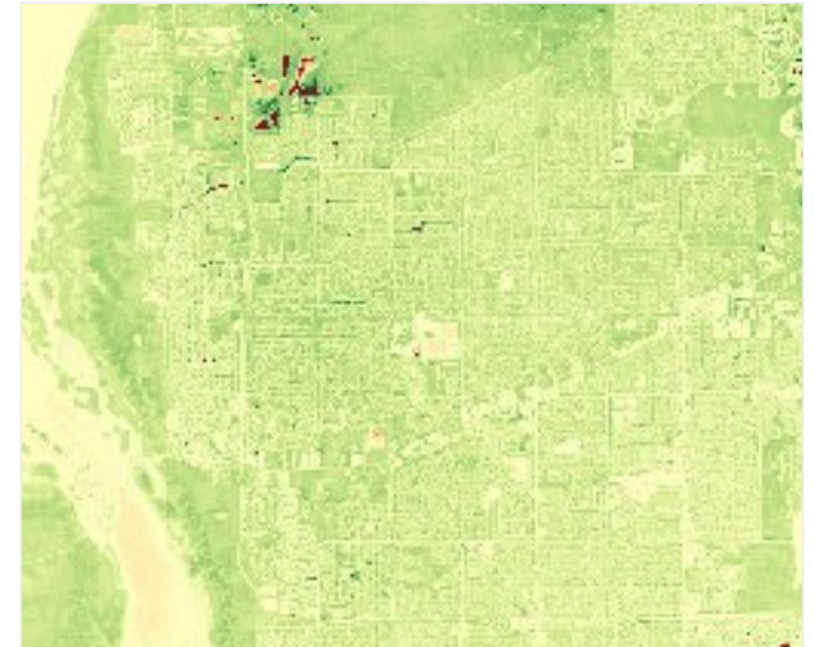
HLS NDVI Post Hurricane Ian



?assets=cog_default



?assets=cog_default&rescale=-1%2C1



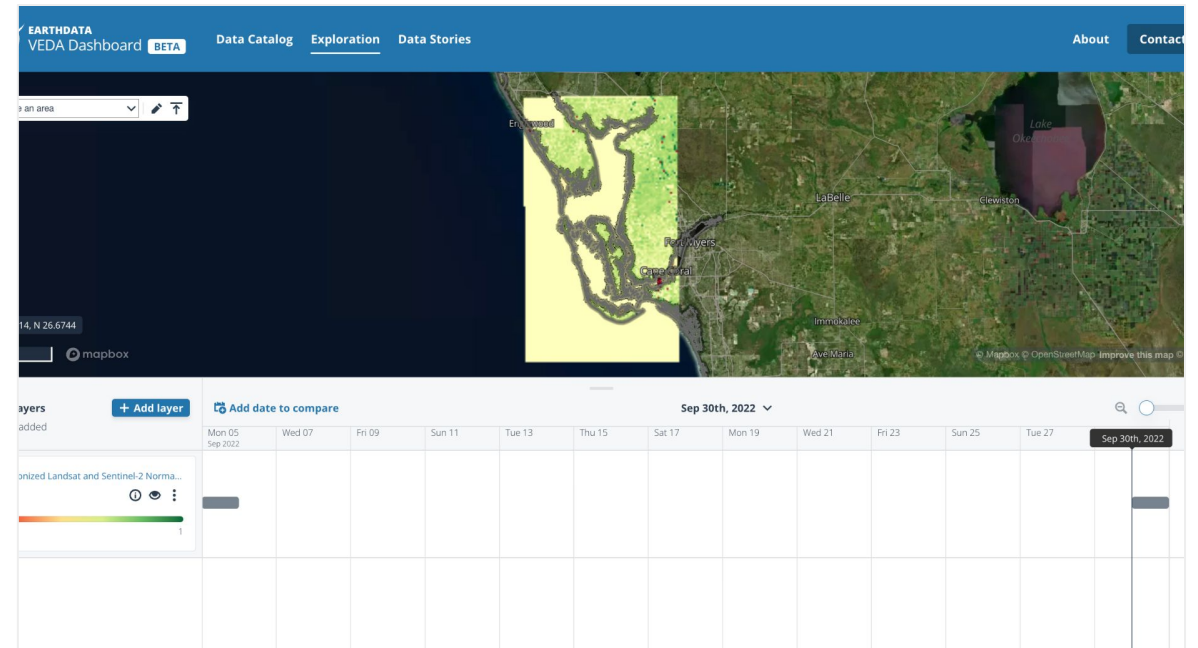
?assets=cog_default&rescale=-1%2C1&colormap_name=rdylgn&

Visualization of the dataset

HLS NDVI Post Hurricane Ian

sourceParams	
colormap_name	rescale
rdylgn	-11

Configuration from the file



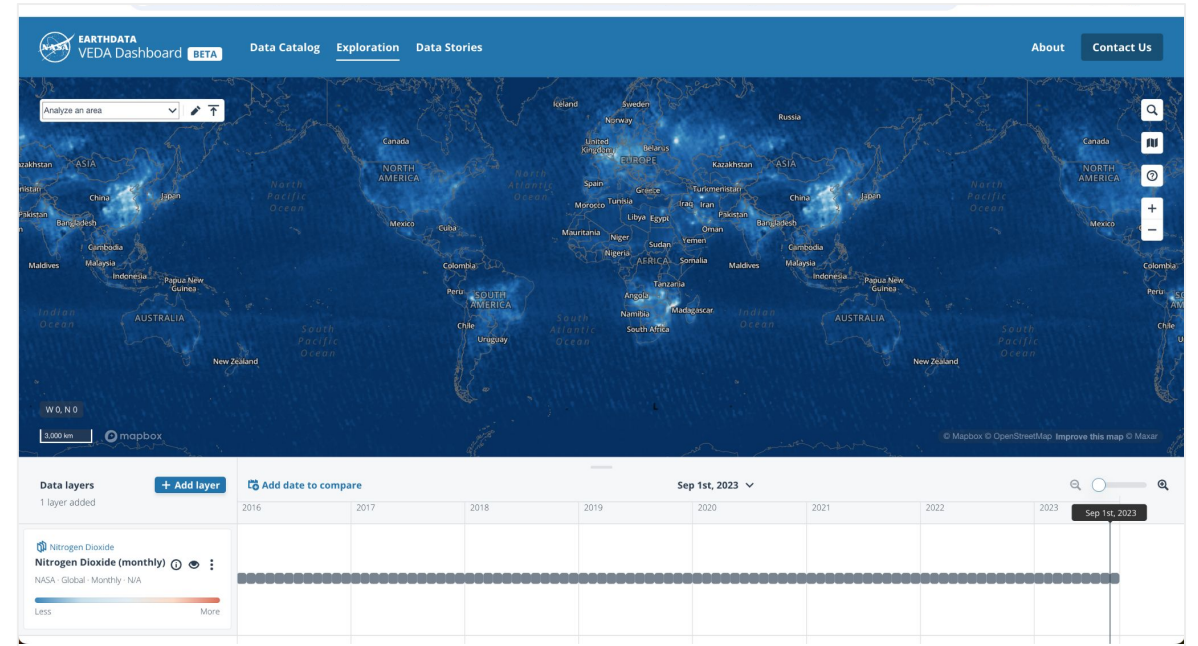
Visualization on VEDA Dashboard (with titiler)

Visualization of the dataset

No2 Monthly data

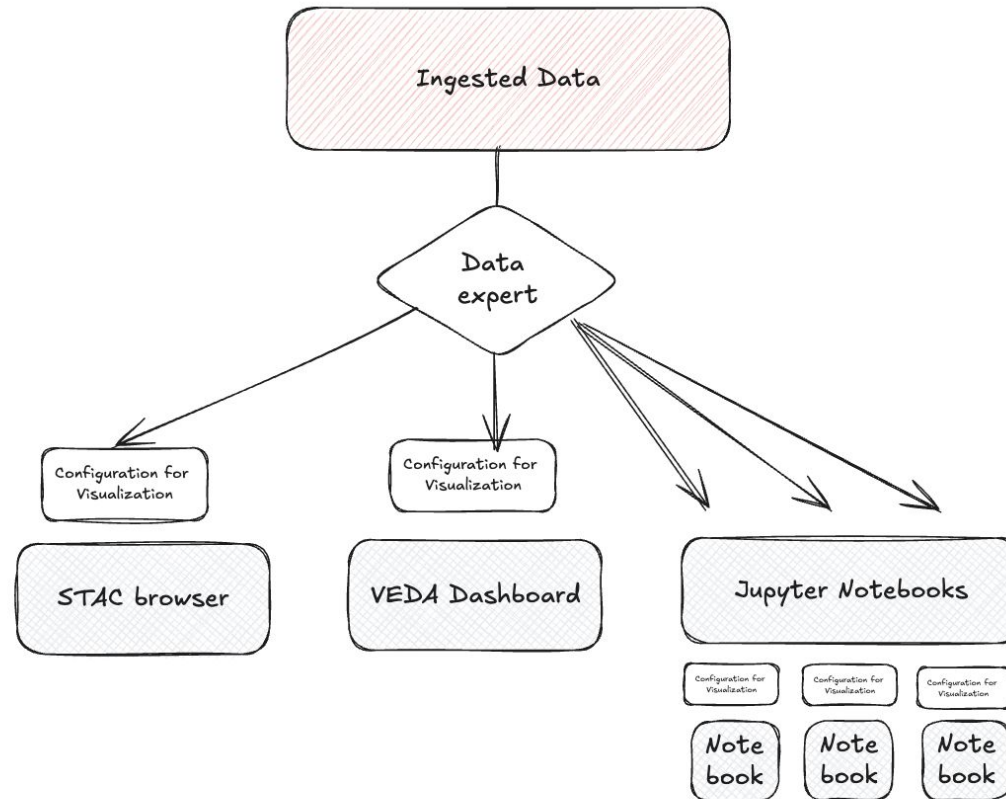
sourceParams						
resampling	bidx	color_formula	colormap_name	rescale		
bilinear	1	gamma r 1.05	rdbu_r	<table border="1"> <tr> <td>0</td> <td>1.5e16</td> </tr> </table>	0	1.5e16
0	1.5e16					

Configuration from the file



Visualization on VEDA Dashboard (with titiler)

VEDA Dashboard is not the only client



Problems of manual configuration

- Disconnection between data store and its visual representation
- Configuration redundancies across clients
- Challenges to spot the errors in visualization without domain expertise
- Reproducibility of data visualization

→ **Need for a scalable solution that works with our current stack**

STAC & Render Extension

A solution for reproducible, scalable visualization

STAC & Extensions

- STAC: SpatioTemporal Asset Catalog
- The STAC specification is a **common language to describe geospatial information, so it can more easily be worked with, indexed, and discovered** (source: <https://stacspec.org/en/>)
- JSON format (programming language agnostic)
- **Designed for extension to adapt to any domain**
- **Concepts:** Catalog, Collection, Item, Asset

The STAC Specification

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

Item

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

Collection

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

Catalog

Extending STAC

- Core STAC specification defines minimal core, **designed for extension**
- Expect real-world usage to **use extensions to fully describe data**
- Can extend Catalog, Collection, Item, or even other extensions
- An extension is **defined via JSON Schema**

Render Extension

- Include asset styling, reprojection or image encoding as metadata at the collection or item level
- Works best with dynamic tilers: render parameters can be directly used as query parameters for tilers
- Multiple namespaces for different usages (ex. Different visualization parameters for different assets)

Example Render extension schema

N02 Monthly collection metadata

```
"renders": {  
  "dashboard": {  
    "bidx": [  
      1  
    ],  
    "title": "VEDA Dashboard Render Parameters",  
    "assets": [  
      "cog_default"  
    ],  
    "rescale": [  
      [  
        0,  
        15000000000000000  
      ]  
    ],  
    "resampling": "bilinear",  
    "color_formula": "gamma r 1.05",  
    "colormap_name": "rdbu_r"  
  }  
},
```

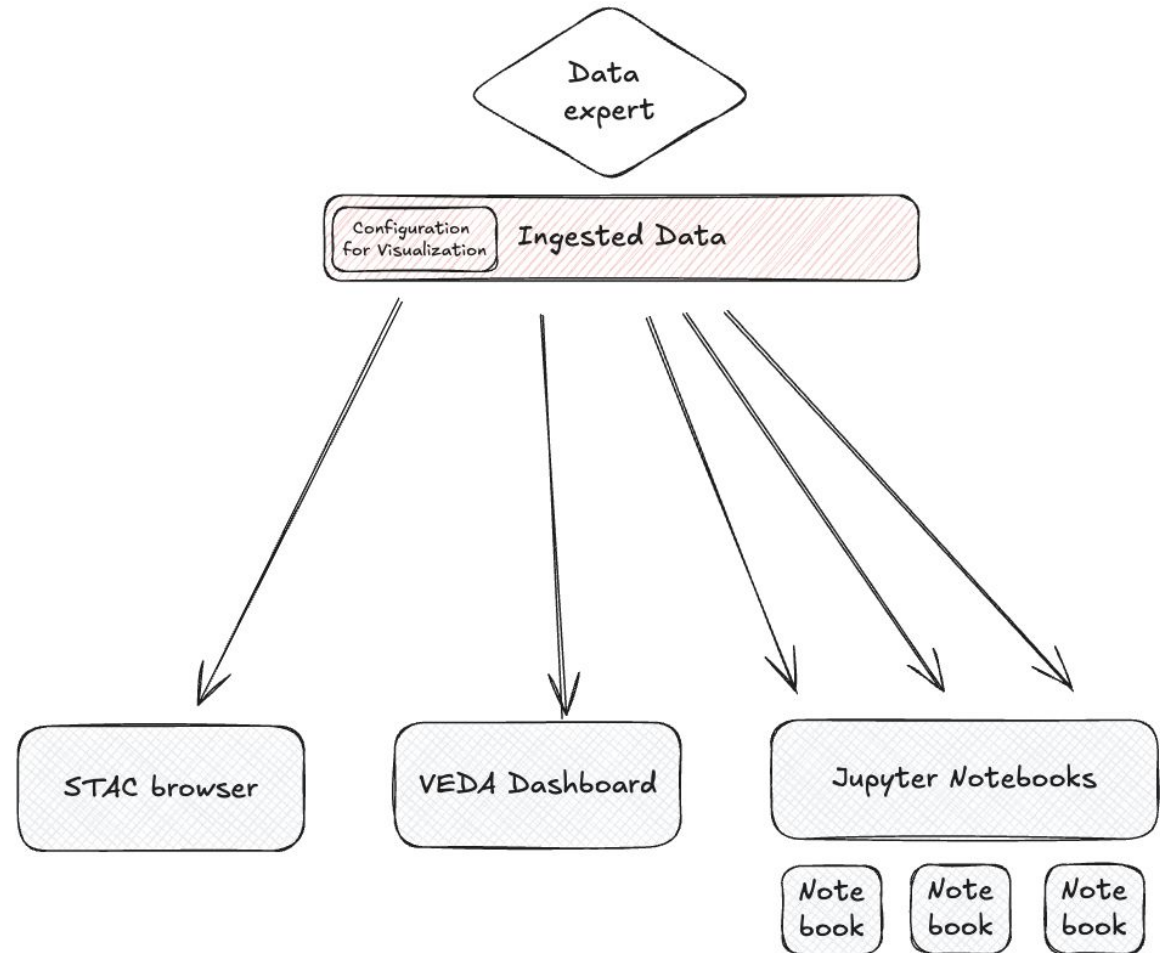
Render parameters (dashboard namespace)

```
sourceParams:  
  resampling: bilinear  
  assets: cog_default  
  bidx: 1  
  color_formula: gamma r 1.05  
  colormap_name: rdbu_r  
  rescale:  
    - 0  
    - 1.5e16
```

VEDA Configuration

Centralized metadata

UI clients and users can now produce visually appealing maps without the need to maintain configurations for each dataset.



Render Extension Use Cases

Web application - VEDA Dashboard

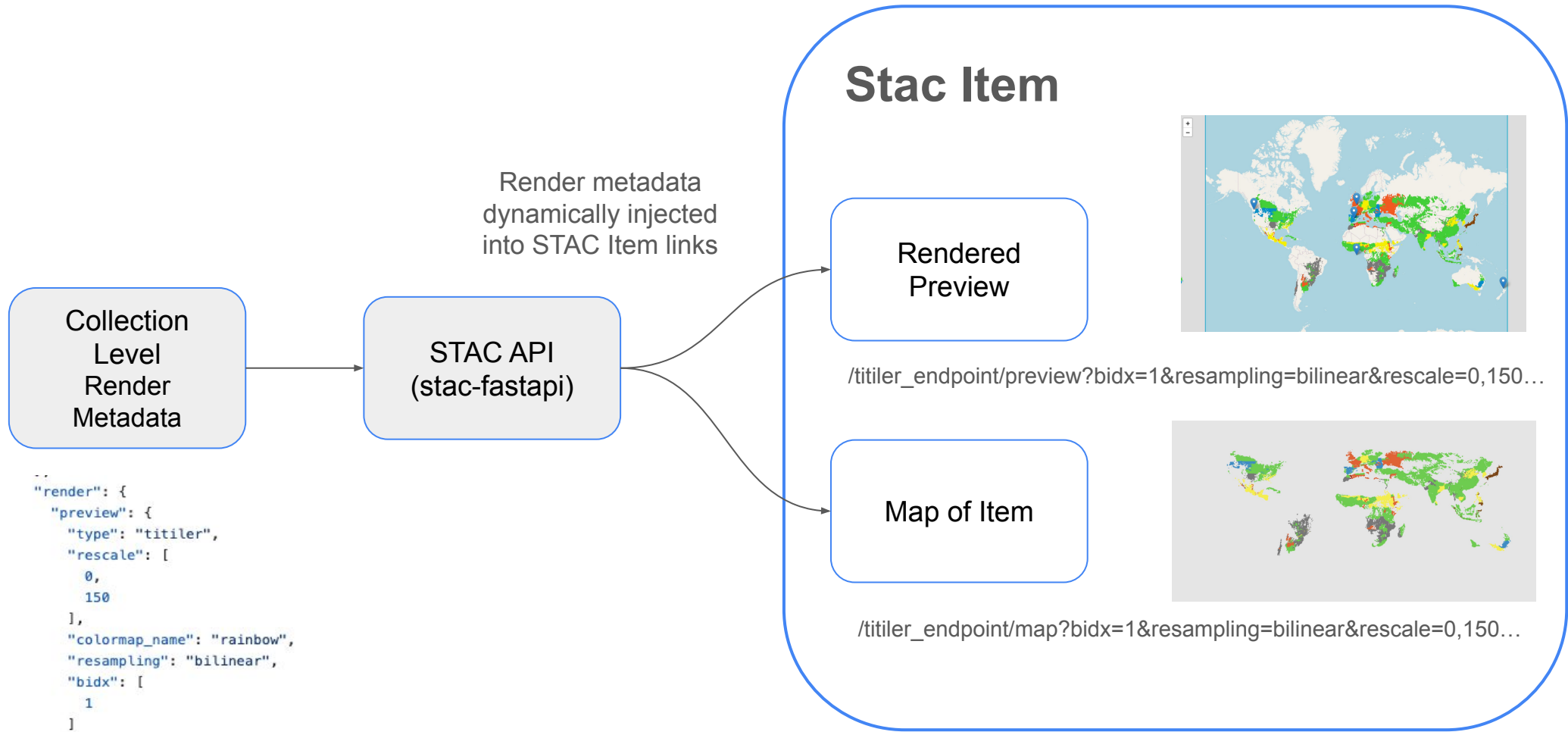
```
7 datasets/CMIP-winter-median-ta.data.mdx
@@ -21,13 +21,6 @@ layers:
21 name: 'SSP2-4.5, Change to winter average air temperature'
22 type: raster
23 description: 'Difference in projected winter (January, February, March) average air temperature, relative to a historical timeframe be
ensembles from CMIP6 (SSP 2-4.5) with downscaling performed by NASA Earth Exchange'
24 - sourceParams:
25 -   resampling: bilinear
26 -   bidx: 1
27 -   colormap_name: rdbu_r
28 -   rescale:
29 -     - -5.5
30 -     - 5.5
31 compare:
32   datasetId: CMIP-winter-median-ta
33   layerId: CMIP245-winter-median-ta
```

```
6 datasets/aerosol-difference.data.mdx
@@ -25,12 +25,6 @@ layers:
25 zoomExtent:
26 - 0
27 - 20
28 - sourceParams:
29 -   colormap_name: bwr
30 -   rescale:
31 -     - -0.1
32 -     - 0.1
33 -   nodata: 0
34 compare:
35   datasetId: houston-urbanization
36   layerId: houston-urbanization
```

```
5 datasets/bangladesh-landcover-2001-2020.data.mdx
@@ -24,11 +24,6 @@ layers:
24 zoomExtent:
25 - 3
26 - 20
27 - sourceParams:
28 -   resampling: nearest
29 -   colormap: '{"0": [0, 0, 0, 128], "100": [0, 130, 0, 255], "200": [17, 131, 226, 255], "300": [199, 43, 32, 255], "400": [98, 234, 37, 255]}'
30 -   bidx: 1
```

- A drop-in replacement for manual configuration
- Saves editors' time and efforts while increasing the consistency of the data visualization

Backend use-case: Dynamic preview generation



VEDA STAC Browser with dynamically generated previews

Gridded 2012 EPA Methane Emissions - Stationary Combustion (monthly)

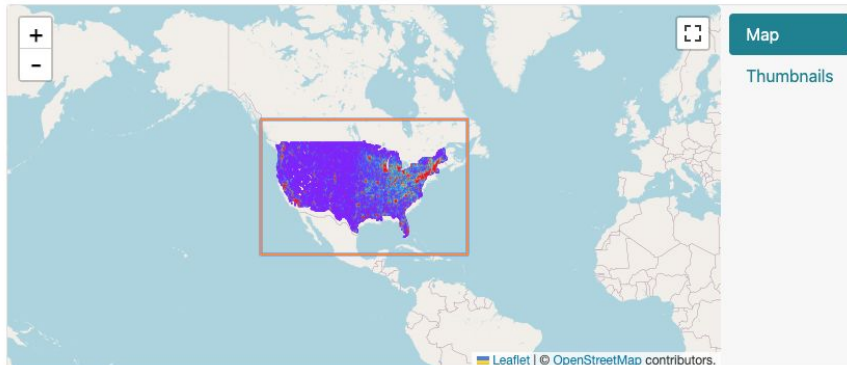
Source Share Language: English

in [openveda.cloud](#) [Up](#) [Browse](#) [Search](#)

Description

Stationary (non-mobile) emissions from sector 1A, including boilers, heaters, furnaces, kilns, ovens, flares, thermal oxidizers, dryers, and any other equipment or machinery that combusts carbon bearing fuels or waste stream materials.

License **CC0-1.0**
Temporal Extent **1/1/2012, 12:00:00 AM UTC - 12/31/2012, 12:00:00 AM UTC**



Asset

> Thumbnail **THUMBNAIL** **JPEG**

Providers

EPA **PRODUCER** **PROCESSOR** **LICENSOR**
NASA VEDA **HOST**

Items

Show Filters

EPA-monthly-emissions_1A_Combustion_Stationary_2_01201
GeoTiff
1/1/2012, 12:00:00 AM UTC - 1/31/2012, 12:00:00 AM UTC

EPA-monthly-emissions_1A_Combustion_Stationary_2_01207
GeoTiff
7/1/2012, 12:00:00 AM UTC - 7/31/2012, 12:00:00 AM UTC

EPA-monthly-emissions_1A_Combustion_Stationary_2_01202

EPA-monthly-emissions_1A_Combustion_Stationary_2_01208

Jupyter notebook

Example Notebook (icesat2-boreal visualization) :

<https://github.com/MAAP-Project/maap-documentation/pull/423/files>

Render extension works well with dynamic tilers, but can be used for general purpose.

```
# load the render parameters from the collection metadata
agb_render_params = collection_info["search"]["metadata"]["defaults"][RENDERING].copy()

# add some zoom limits to avoid overloading the tile server
agb_render_params["minzoom"] = -7
agb_render_params["maxzoom"] = 14

pprint(agb_render_params)
✓ 0.0s
```

```
{'assets': ['tif'],
 'bidx': [1],
 'colormap_name': 'gist_earth_r',
 'maxzoom': 14,
 'minzoom': 7,
 'nodata': 'nan',
 'rescale': [[0, 400]]}
```



Resources

Render Extension repo: <https://github.com/stac-extensions/render>

VEDA Dashboard: <https://www.earthdata.nasa.gov/dashboard/>

GitHub Repository: <https://github.com/NASA-IMPACT/veda-config>