General Server for Rapid Publishing of OGC-Compliant Earth Science Data Products

Mattheus Ueckermann[1] and Jerry Bieszczad[1]

[1]Creare LLC

November 21, 2022

Abstract

To make timely decisions for weather -and climate-related disasters and vulnerabilities, decision makers need current information that can be readily shared and communicated to stakeholders. To date, geospatial data is distributed using monolithic storage architectures and formats best suited for traditional research applications. Thus, everyday decision-makers face significant "barriers to entry" when trying to access, explore, and modify vast historical archives and real time data feeds. To address this need, we are developing a server architecture for rapidly creating and publishing data products. Privileged users can rapidly create or change products by operating on another product or combining multiple disparate data sources together. Users can then consume these new products using OGC-compliant WMS/WCS clients such as ArcGIS, QGIS, or Leaflet. This enables decision makers to effectively communicate with stakeholders using customized maps. Moreover, this capability enables products to be rapidly updated in cases where timely information is important. Our server architecture is containerized, making it easy to deploy on various architectures including serverless cloud resources. It is implemented in Python, leverages the plug-and-play data wrangling capabilities of the PODPAC library, and uses a custom library for serving OGC-compliant data. The result is an easy-to-use architecture for rapidly publishing custom geospatial products that exploit vast earth science data resources. We will demonstrate our server capabilities by showing how privileged users can build a set of products that are computed on-demand starting from a fresh server. Using Jupyter Lab notebooks, we will create products that modify single data sources as well as products that combine multiple disparate sources. We will then show how users can consume these products using OGC-compliant clients. Next, we will detail our cloud-based, serverless deployment of this technology using Amazon Web Services. Finally, we will discuss the advantages of our approach along with any caveats. Enabling everyday decision makers to rapidly create and share geospatial data will revolutionize their productivity and effectiveness for assessing and remediating weather- and climate-related vulnerabilities and disasters.

IN45C-0469
AGU FALL MEETING
New Orleans, LA & Online Everywhere
13–17 December 2021

General Server for Rapid Publishing of OGC-Compliant Earth Science Data Products

Creare LLC
J. Bieszczad
M. Ueckermann

Motivation

- **Decision makers** need current data to **make timely decisions** for weather and climate-related vulnerabilities
- **Geospatial** information needs to be easily shared and communicated to stakeholders
- To date, geospatial data is distributed using **monolithic storage architectures** and formats best-suited for traditional **research applications**
- Everyday **decision-makers** face **significant barriers** when trying to **access, explore, and modify** vast historical archives and real-time data feeds

Project Objectives

- Develop **server architecture** for **rapidly creating** and **publishing** geospatial data products
- Privileged users can **publish** new or **update** existing geospatial products by **combining multiple disparate data sources** together, **post-processing**, and **styling** results
- Users can consume these products using OGC-compliant WMS/WCS clients such as ArcGIS, QGIS, or Leaflet
- Server architecture is containerized, making it easy to deploy on various architectures including local networks or serverless cloud architectures

Project Status

- Server architecture is fully functional and containerized
- Development of web-based interface of product creation is in progress
- The code has not yet been published

Open-Source Development

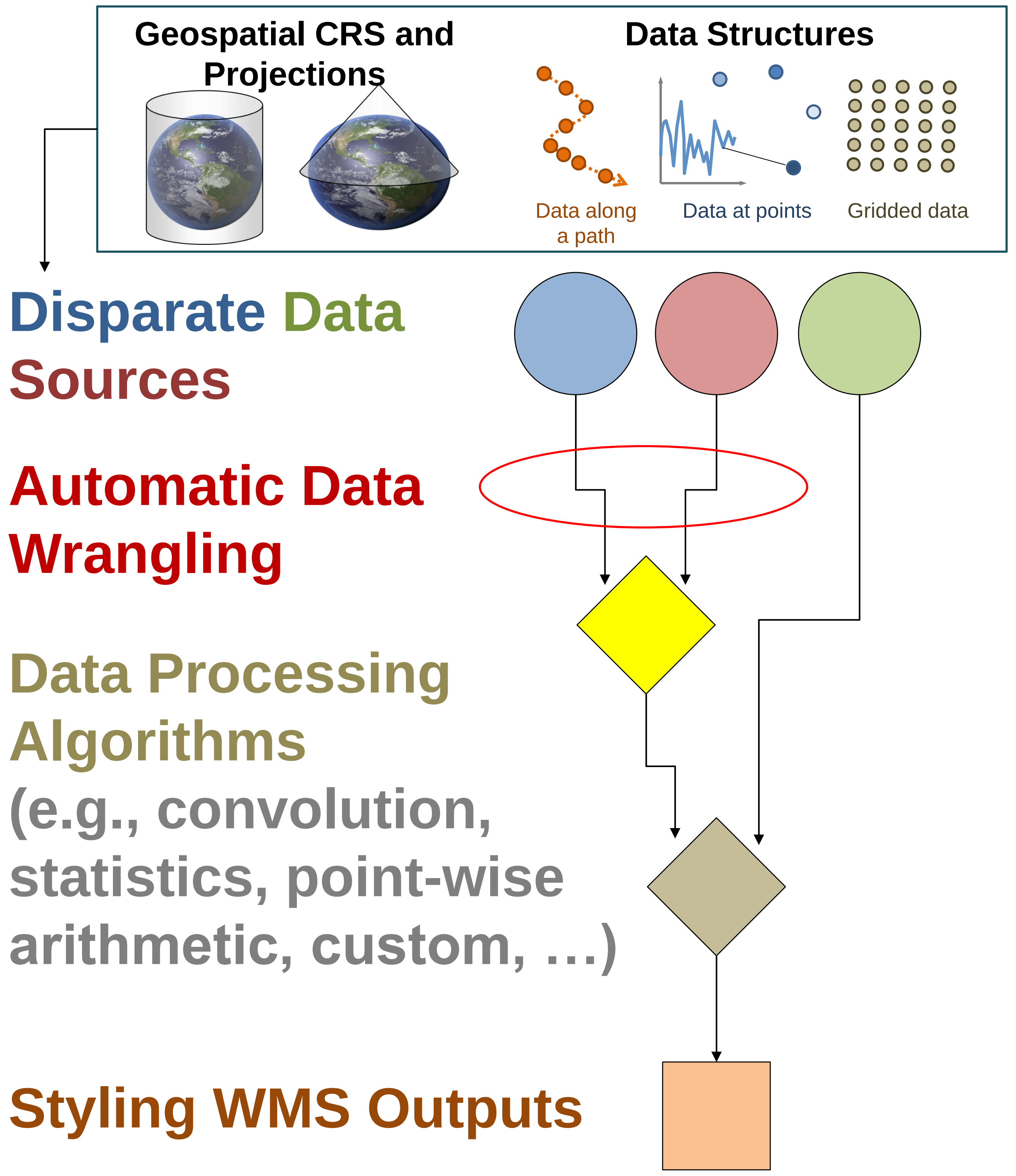
- Our server is built using PODPAC, which is open-source software available at https://podpac.org

Acknowledgment

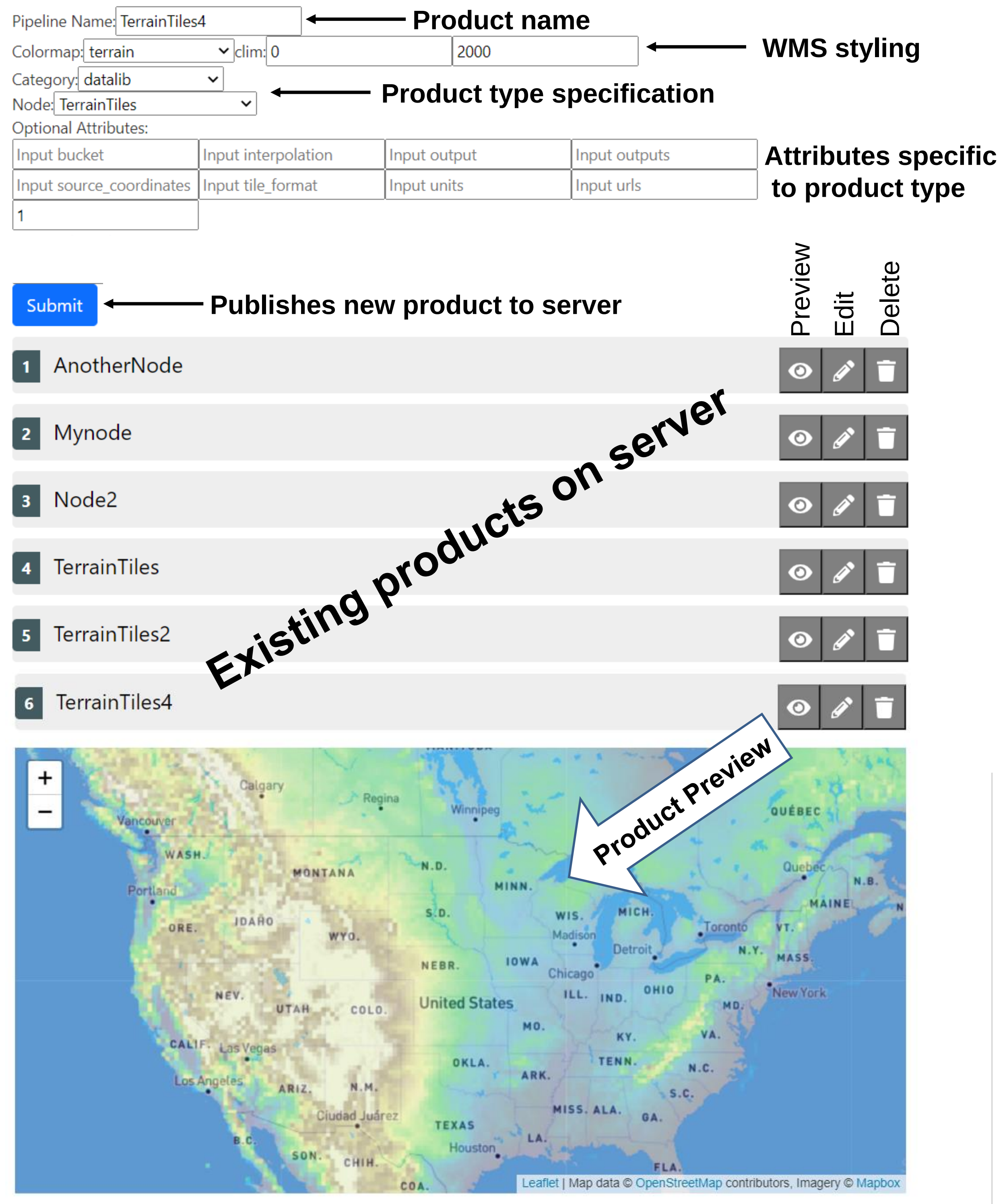
- This research is supported by the US Army ERDC under SBIR Phase II Contract No. W9132V19C0002

Geospatial Data Product Creation Workflow

Capabilities available when creating new products



Creating a node using our custom web interface



Pipeline Name: TerrainTiles4 ← Product name

Colormap: terrain clim: 0 2000 ← WMS styling

Category: datalib

Node: TerrainTiles ← Product type specification

Optional Attributes:

Input bucket	Input interpolation	Input output	Input outputs
Input source_coordinates	Input tile_format	Input units	Input urls
1			

← Attributes specific to product type

Submit ← Publishes new product to server

Creating a node using SoilMAP and PODPAC (Python)

Initialize publisher class

```
publish = Publish(
    source="https://example-server.com/api",  # Server API endpoint
    secret_key="<user-secret-token>",  # Privileged user’s credentials
    name="<my-node-name>",  # OGS “Layer” name
    expiration_date=None,  # Optional expiration date
)
```

Style and create a pipeline

```
style = podpac.style.Style(
    colormap="terrain", clim=(800**2, 1000**2))

dem_data = podpac.datalib.TerrainTiles (zoom=3)
# Limit DEM between 800 and 1000 m, and compute the square
node = podpac.algorithm.Arithmetic(
    eqn="(dem > 800) * (dem < 1000) * dem**2",
    dem=dem_data,
    style=style,
)
```

Publish

```
publish(node)
```

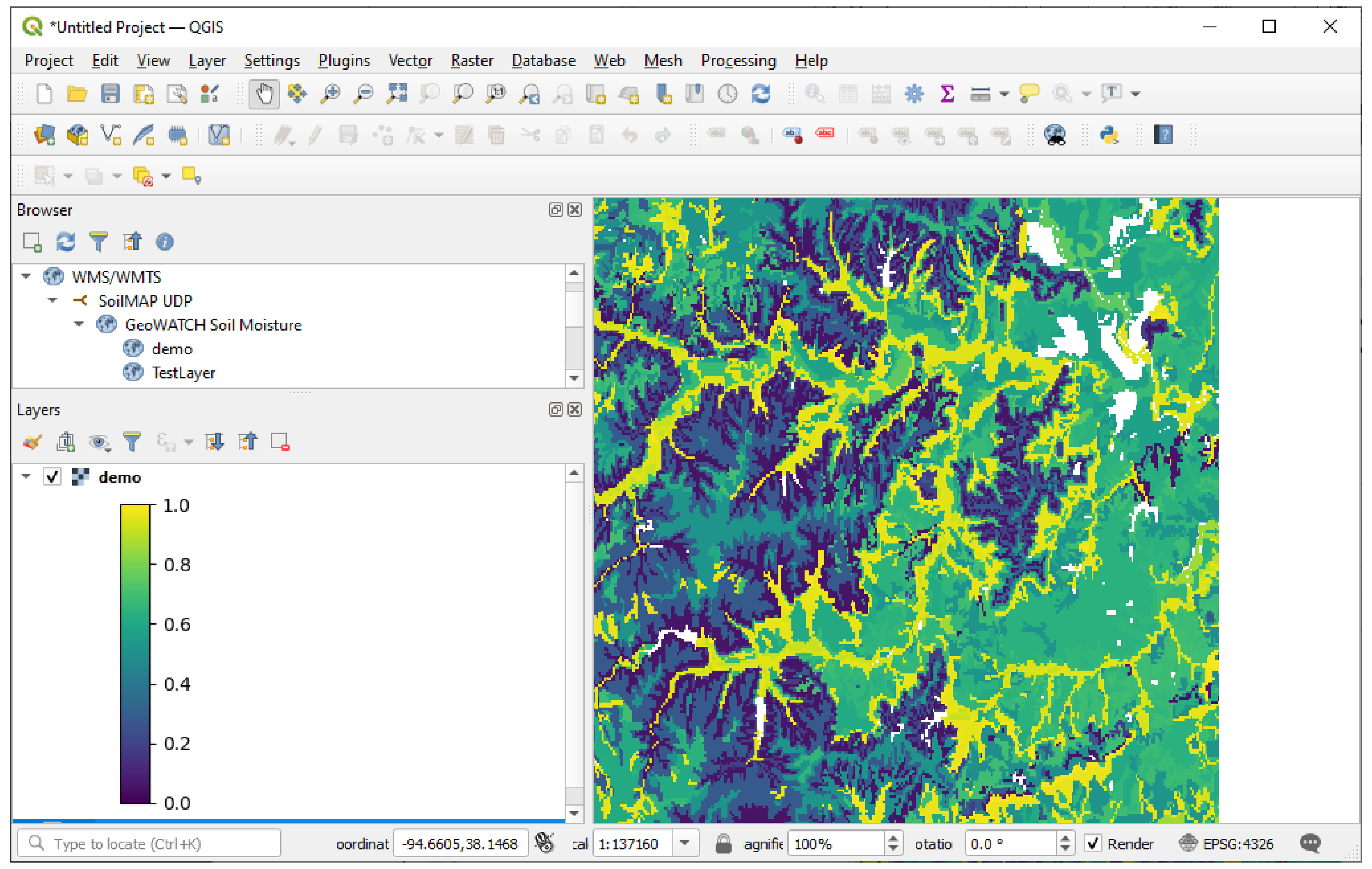
← Could also be plain JSON (used by web-based UI)

How Does it Work?

- Our server leverages the open-source PODPAC Python library’s automated data wrangling and “Node” serialization features
- PODPAC describes geospatial processing pipelines using a light-weight JSON format
- This JSON description is sent to the server using an HTTP POST request by a privileged user and this definition is saved along with a product name
- Users can then request products using the same name, and the server will recreate the data product from its definition, serving the results to the user

Product Consumption Workflow

QGIS used as a WMS/WCS client



- Our server automatically creates an OGC-compliant endpoint that is updated dynamically as new data products are created
- Users point their OGC client to the server endpoint and available products are automatically discovered
- Users select desired data product to browse it (WMS endpoint), or further modify the results (WCS endpoint) and our server fetches and processes data on-the fly with optional server-side caching

Technology

- Jupyterlab to build interactive, customizable UIs
- Leaflet + custom web development for UI that requires no software installation to create data products
- Open-source scientific Python stack

ipyleaflet
Traitlets
ipywidgets
Leaflet
Rasterio
PROJ
jupyterlab
matplotlib
SciPy
NumPy
xarray