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

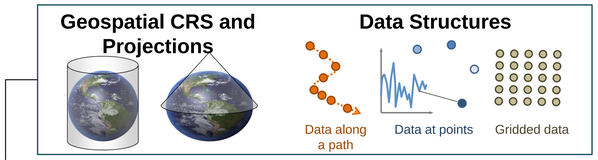
- Our server is built using PODPAC, which is open-source software available at https://podpac.org

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

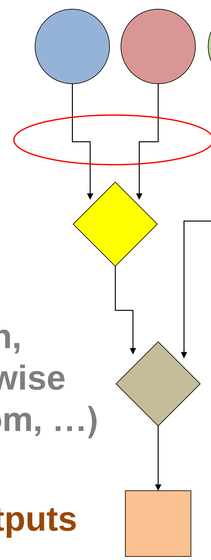


Disparate Data Sources

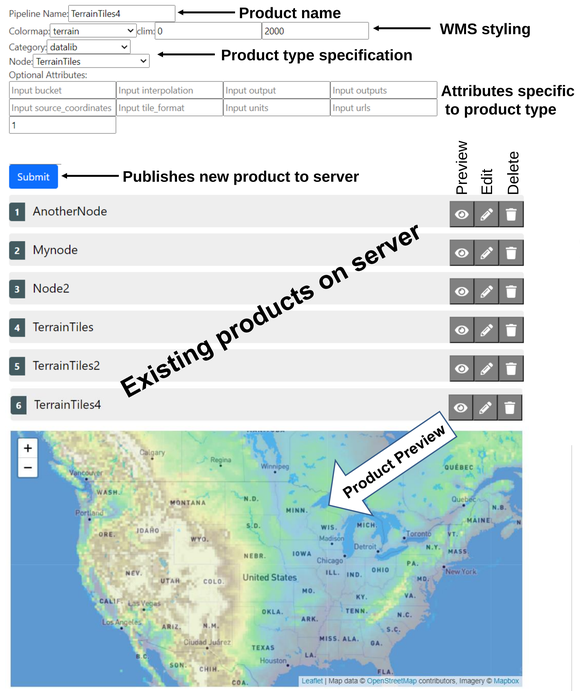
Automatic Data Wrangling

Data Processing Algorithms (e.g., convolution, statistics, point-wise arithmetic, custom, …)

Styling WMS Outputs



Creating a node using our custom web interface



Creating a node using SoilMAP and PODPAC (Python)

Initialize publisher class

```python
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

```python
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

```python
publish(node)
```

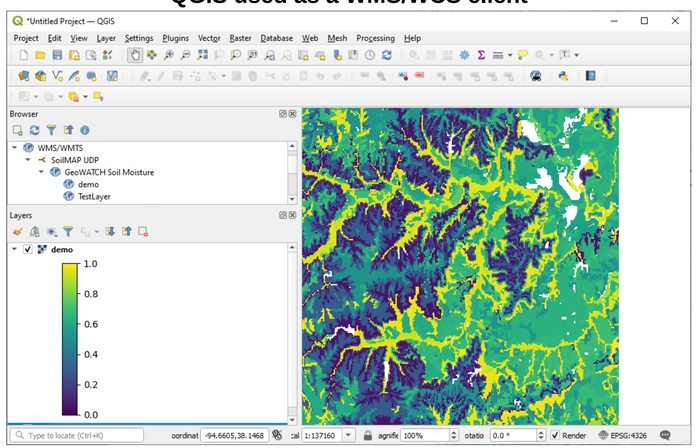
Could also be plain JSON (used by web-based UI)

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 · [ipy]widgets · Leaflet · Rasterio · PROJ · jupyterlab · matplotlib · SciPy · NumPy · xarray