AN OGC API TO GEOSPATIAL DATA

pygeoapi.io

This presentation available at https://pygeoapi.io/presentations/default

TABLE OF CONTENTS

GEOSPATIAL API EVOLUTION

1990S

- Spirit of XML-RPC/CORBA
- SOAP/WSDL/UDDI
- Service Oriented Architecture (SOA)
- Strong concept of RDBMS as the backend
- OGC WMS (1999)

2000s

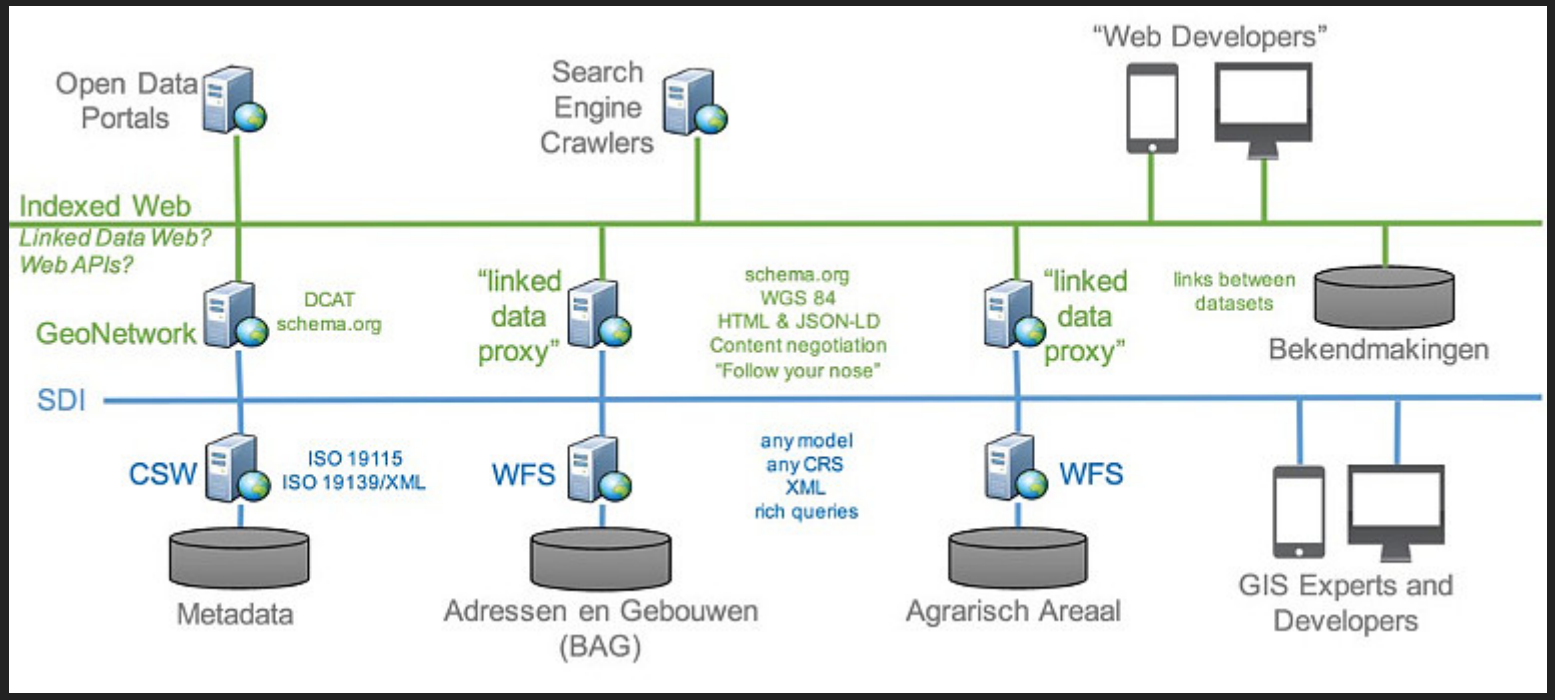
- Web 2.0
- JavaScript/AJAX/Google Maps
- Slippy maps, tiles
- OGC WFS (2002), WCS (2003), WPS (2005), CSW (2007)

REALITIES OF OGC WEB SERVICES

- Key Value Parameters (KVP) or XML encodings
- Requests using HTTP GET, HTTP POST and SOAP
- Responses as XML (GML)
- Error handling as custom *ExceptionReport* within an HTTP 200 response
- Exhaustive specifications/requirements

```
<wfs:FeatureCollection
    xmlns:ms="http://mapserver.gis.umn.edu/mapserver"
    xmlns:gml="http://www.opengis.net/gml" [..]>
    <gml:boundedBy>
        <gml:Envelope srsName="EPSG:32661">[..]
        </gml:Envelope>
    </gml:boundedBy>
    <gml:featureMember>
        <ms:north_poles_wfs gml:id="north_poles_wfs.0">
            <gml:boundedBy>[..]</gml:boundedBy>
            <ms:msGeometry>[..]</ms:msGeometry>
            <ms:Id>0</ms:Id>[..]
    </gml:featureMember>[..]
</wfs:FeatureCollection>
```

MORE REALITIES



- Complex machinery/architectures (e.g. ldproxy) for services to be crawlable
- Challenging for web developers to implement
- Challenging for mainstream web integration (search indexing)
- Geonovum testbed showed OGC world can easily be

REST/JSON/OPENAPI/ SWAGGER

REST

- REpresentational State Transfer (REST)
- HTTP verbs (GET/PUT/POST/DELETE)
- HTTP codes (200, 201, 404, etc.)
- Uses URI to identify resources
- Content negotiation (media types)
- Stateless

JSON

- JavaScript Object Notation
- Easy to write and understand
- Very popular among web developers
- JSON is first class in RESTful web services
- No rigid standards (for better or worse)

OPENAPI

- Specification on how to describe a REST API
- Defines endpoints, request parameters, response
- Programming language agnostic
- Formerly Swagger

```
"info": {
    "contact": {
        "email": "you@example.org",
        "name": "pygeoapi Development Team",
        "url": "https://pygeoapi.io"
    },
    "description": "pygeoapi provides an API to geospatial data",
    "license": {
        "name": "CC-BY 4.0 license",
        "url": "https://creativecommons.org/licenses/by/4.0/"
    },
    "termsOfService": "None",
    "title": "pygeoapi Demo instance - running latest GitHub version",
    "version": "3.0.2",
    "x-keywords": [
        "geospatial",
        "data",
        "api"
    ]
},
"openapi": "3.0.2",
```

SWAGGER

- OpenAPI client tooling
- Robust, web developer focused
- Automated documentation, testing, code generation

OGC ACTIVITY

https://ogcapi.ogc.org

W3C SPATIAL DATA ON THE WEB BEST PRACTICES (2017)

- Recommendations on formats, identifiers, access, licensing, provenance
- Being webby
- https://www.w3.org/TR/sdw-bp

DESIGN PATTERNS

- Being webby (humans, search engines)
- Developer friendly
- Lightweight specification development
- Removing HTTP use as a tunnel
  - ~~/ows?~~
    ~~request=GetFeature&typename=roads&featureid=5~~
  - /api/collections/roads/items/5
- Modular specification development
  - Core and extensions

TIMELINE

- 2017: W3C Spatial Data on the Web Best Practices
- 2017: OGC API Whitepaper
- 2018: WFS3 Hackathon, Weather on the Web API Hackathon
- 2019-2020: OGC API Hackathons
- OGC API Roadmap

FUTURE OGC STANDARDS

OGC ACTIVITY ON GITHUB

SAY HELLO TO PYGEOAPI

pygeoapi

- A Geopython project
- Created by Tom Kralidis in 2018 on Valentine's Day with ♡
- OSGeo Community Project

PROJECT OVERVIEW

- Geospatial data API framework
- OGC Compliant (already!)
- OSGeo Community Project
- International team (Canada, Netherlands, Greece, Italy, New Zealand)
  - Numerous core contributors
  - 21000 upstream contributors (dependencies)

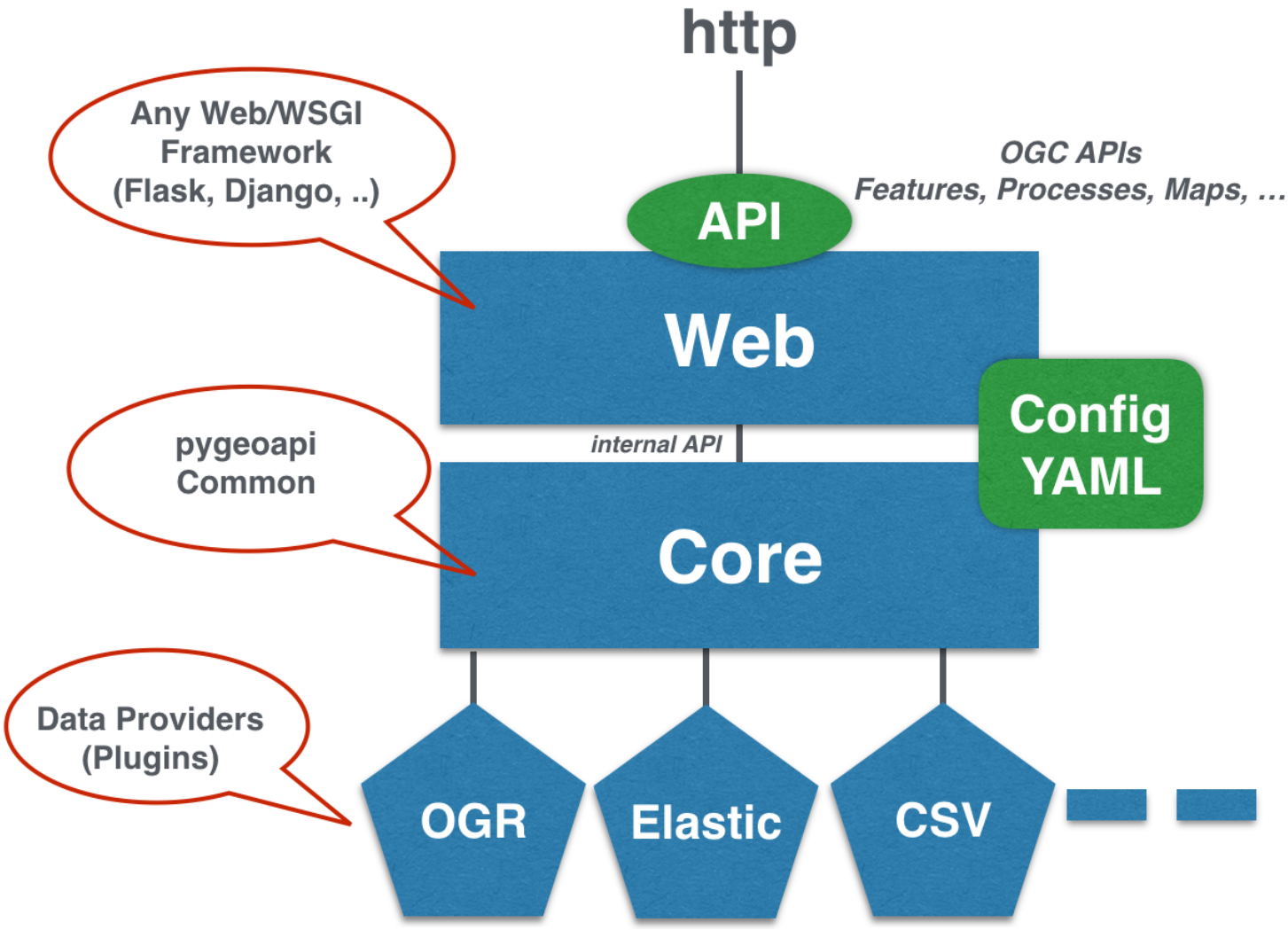
TECHNICAL OVERVIEW

- Publishing (leaves data pristine/alone)
  - Fits to the data (not the inverse)
- Leverages powerful ecosystem of Python packages

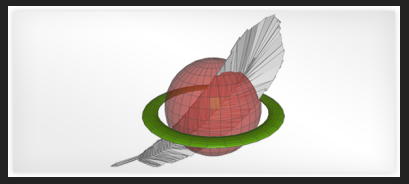
TECHNICAL OVERVIEW

- Core abstract API; web framework agnostic (default Flask)
- YAML configuration (metadata, dataset connections
- Automated OpenAPI document generation (data binding)
- Robust plugin framework (data connections, formats, processing)
- Easy deployment (pip install, Docker, conda, UbuntuGIS)
- Minimal dependencies

pygeoapi - Architecture



FEATURE DATA PROVIDERS



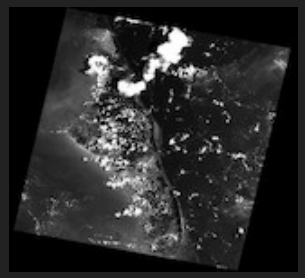
Elasticsearch CSV Spatialite* GeoJSON*

* also via GDAL-OGR Provider

FEATURE DATA PROVIDERS - GDAL/OGR

- Via OGR-Python bindings
- Unleashes ~100 Vector Formats! (WFS 1/2, GeoPackage, Shapefile, GeoJSON, PostGIS, ...)
- Paging, reprojection, spatial/attribute query
- Unlock existing WFS 1/2s!

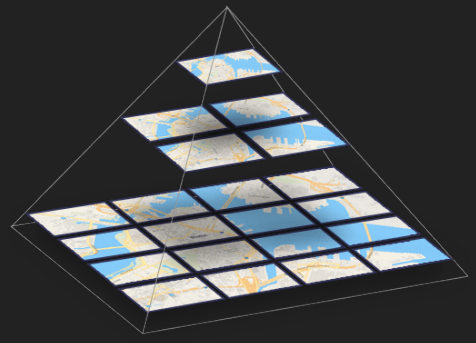
COVERAGE DATA PROVIDERS



xarray rasterio

- NetCDF, Cloud Optimized GeoTIFF (COG)

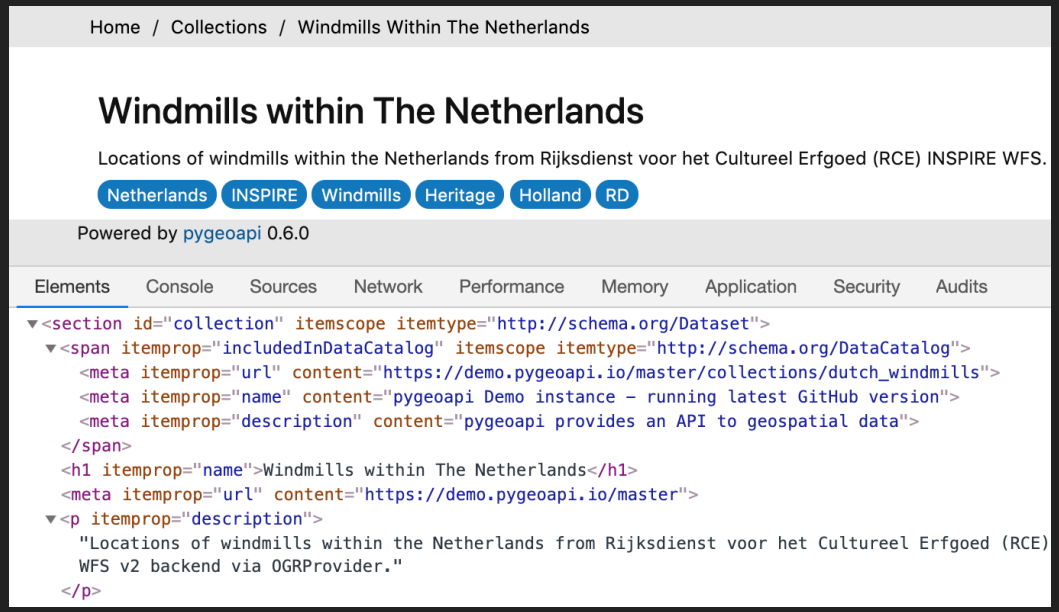
TILE DATA PROVIDERS



minio ZXY directory tree

- MapBox Vector Tiles via tippecanoe

SCHEMA.ORG

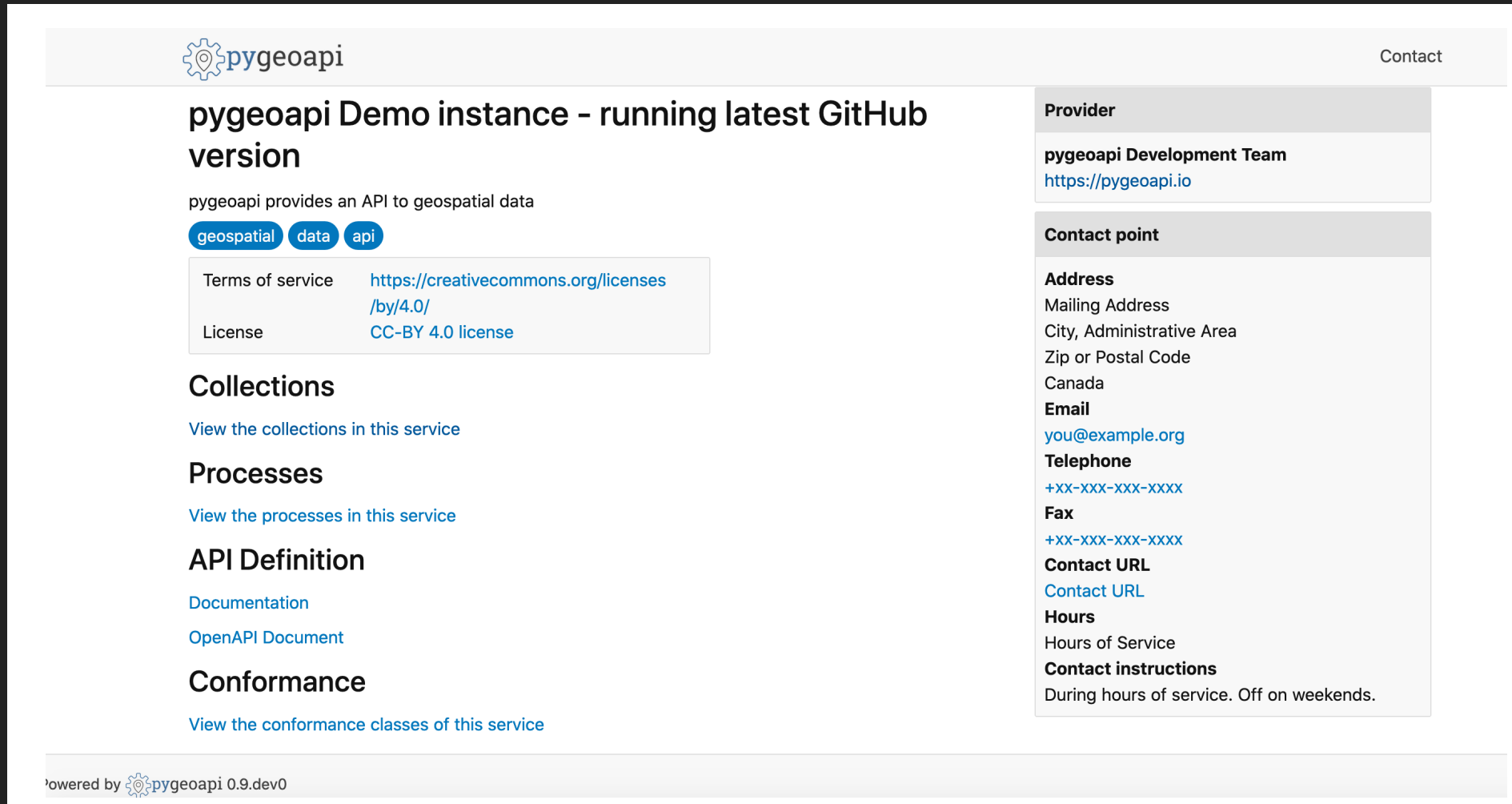


- Lower the barrier to data search
- HTML encoding is annotated as schema.org/Dataset to facilitate search engines
- Google dataset search locates pygeoapi datasets

LINKED DATA - JSON-LD

- JSON-LD provided as additional representation
- Default out of the box functionality for most pygeoapi resources / endpoints
- Additional configuration

CORE CAPABILITIES



OPENAPI

Request URL

```
https://demo.pygeoapi.io/master/collections?f=json
```

Server response

Code	Details
200	Response body


```
{
  "collections": [
    {
      "links": [
        {
          "type": "text/csv",
          "rel": "canonical",
          "title": "data",
          "href": "https://github.com/mapserver/mapserver/blob/branch-7-0/msautotest/wxs/data/obs.csv",
          "hreflang": "en-US"
        },
        {
          "type": "text/csv",
          "rel": "alternate",
          "title": "data",
          "href": "https://raw.githubusercontent.com/mapserver/mapserver/branch-7-0/msautotest/wxs/data/obs.csv",
          "hreflang": "en-US"
        },
        {
          "type": "application/json",
          "rel": "self",
          "title": "This document as JSON",
          "href": "https://demo.pygeoapi.io/master/collections/obs?f=json"
        },
        {
          "type": "application/ld+json",
          "rel": "alternate",
          "title": "This document as RDF (JSON-LD)",
          "href": "https://demo.pygeoapi.io/master/collections/obs?f=jsonld"
        },
        {
```


Download

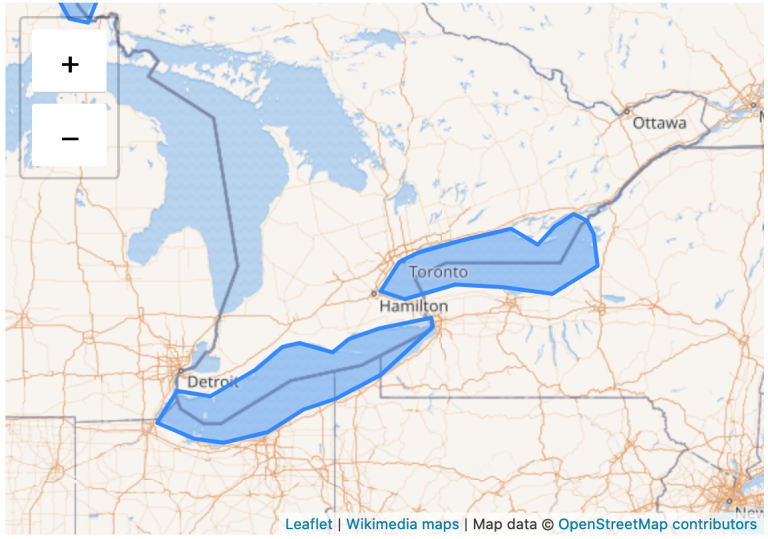
Response headers

```
access-control-allow-origin: *
content-length: 48471
content-type: application/json
date: Tue08 Sep 2020 00:22:43 GMT
server: gunicorn/20.0.4
x-firefox-spdy: h2
x-powered-by: pygeoapi 0.9.dev0
```

OGC API - FEATURES

Large Lakes

Items in this collection.



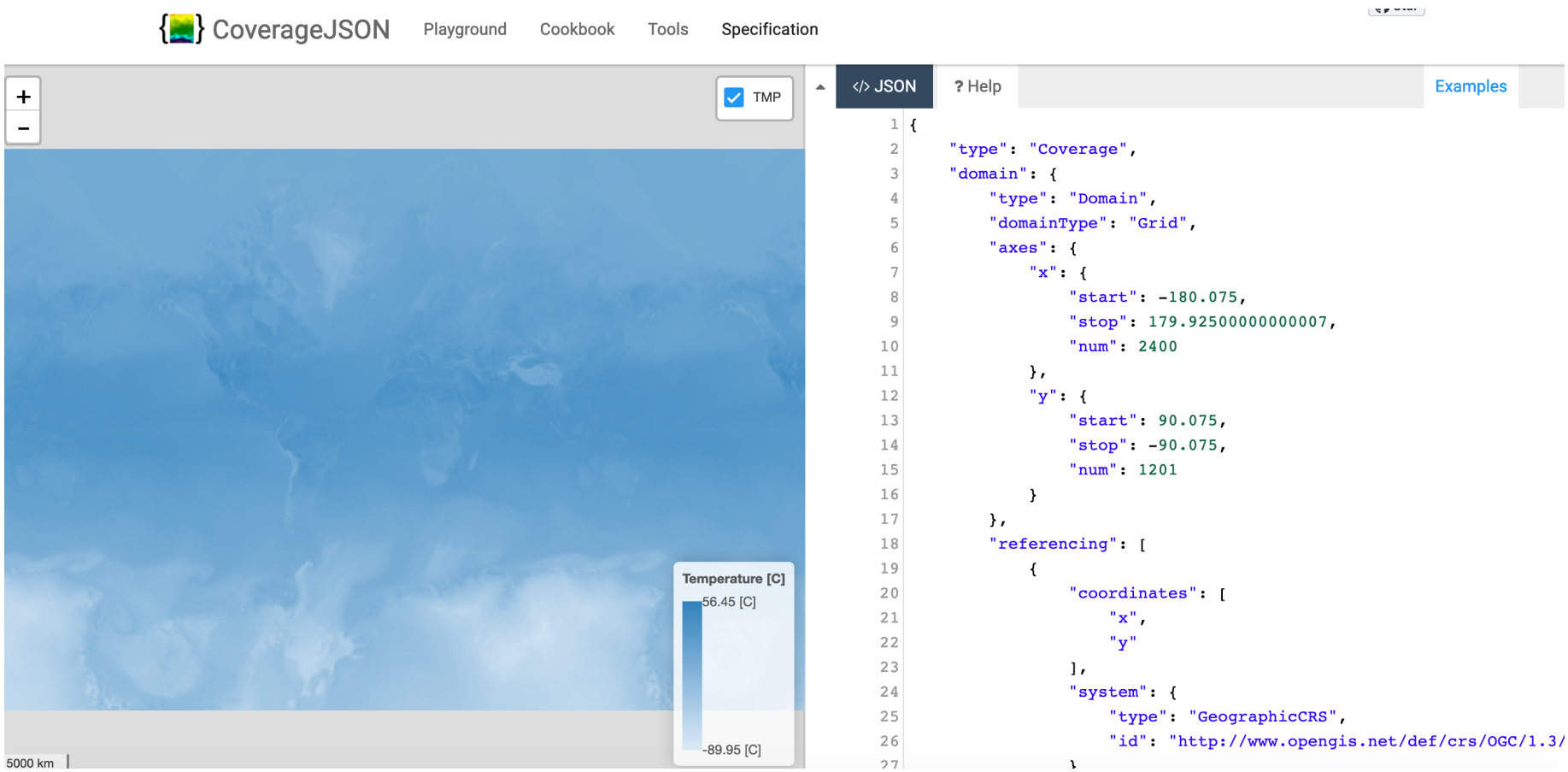
Warning: Higher limits not recommended!

Limit: 10 (default)

Next

id	id	scalerank	name	name_alt
0	0	0	Lake Baikal	https://e...
1	1	0	Lake Winnipeg	https://e...
2	2	0	Great Slave Lake	https://e...
3	3	0	L. Ontario	https://e...
4	4	0	L. Erie	https://e...
5	5	0	Lake Superior	https://e...
6	6	0	Lake	https://e

OGC API - COVERAGES



OGC API - TILES

Large Lakes

lakes of the world, public domain

Tiles

Tile Matrix Set: WorldCRS84Quad

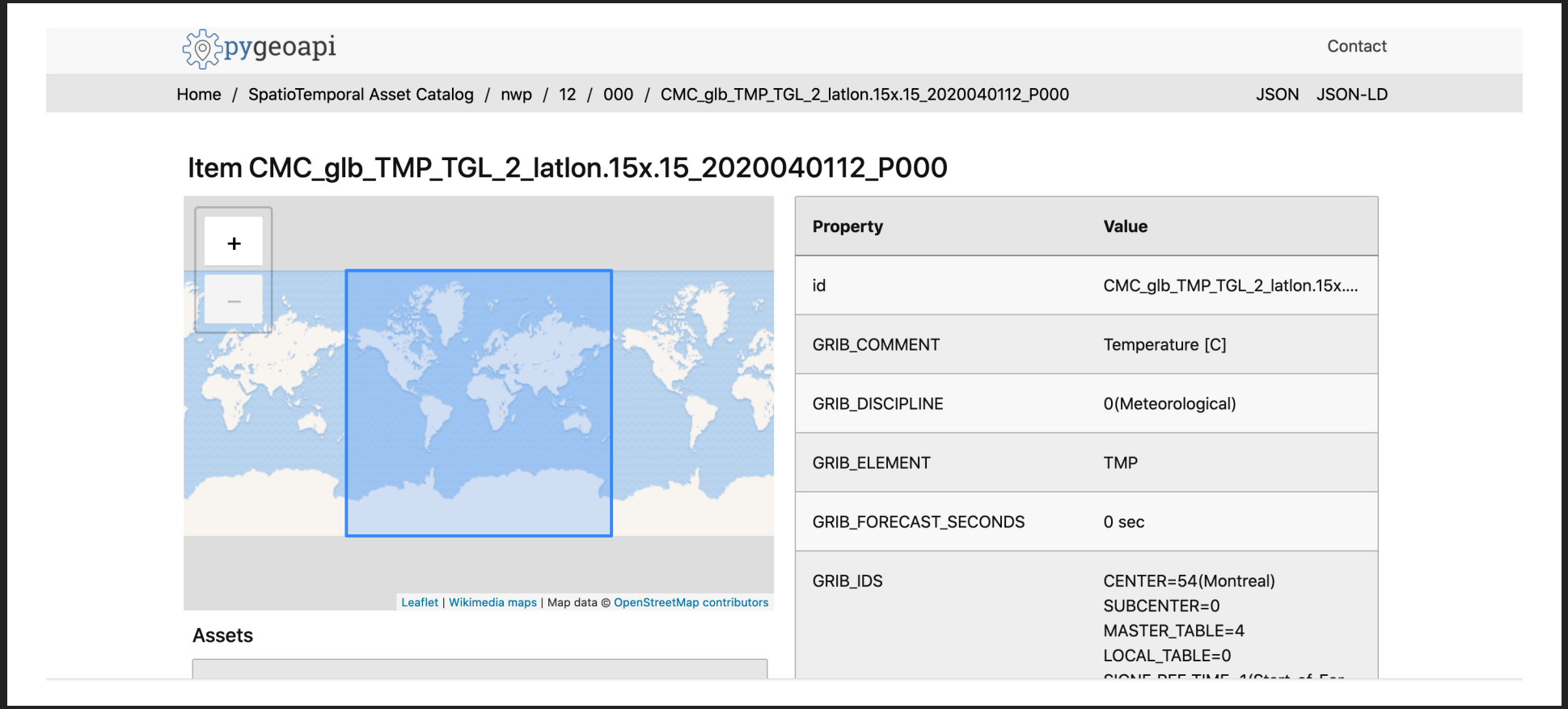
Metadata: Tiles metadata in tilejson format

Map

OGC API - PROCESSES


```
version:                "0.1.0"
id:                     "hello-world"
title:                  "Hello World process"
description:            "Hello World process"
keywords:
  0:                    "hello world"
links:
  0:
    type:               "text/html"
    rel:                "canonical"
    title:              "information"
    href:               "https://example.org/process"
    hreflang:           "en-US"
inputs:
  0:
    id:                 "name"
    title:              "name"
    input:
      literalDataDomain:
        dataType:       "string"
        valueDefinition:
          anyValue:     true
    minOccurs:          1
    maxOccurs:          1
outputs:
  0:
    id:                 "hello-world-response"
    title:              "output hello world"
    output:
      formats:
        0:
          mimeType:     "application/json"
example:
```

SPATIOTEMPORAL ASSET CATALOG (STAC)



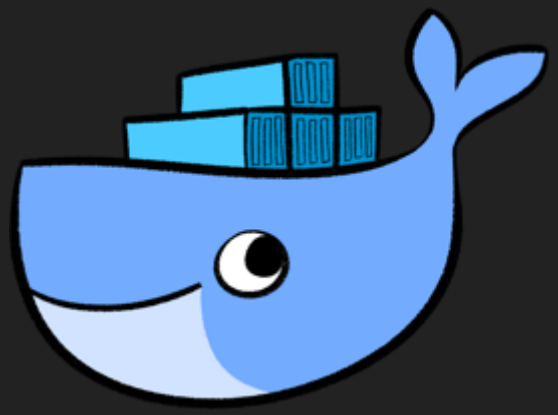
DEPLOYING

PACKAGING

Numerous deployment options

- PyPI
- UbuntuGIS
- Docker
- Conda
- FreeBSD
- Coming to OSGeoLive

DOCKER OVERVIEW



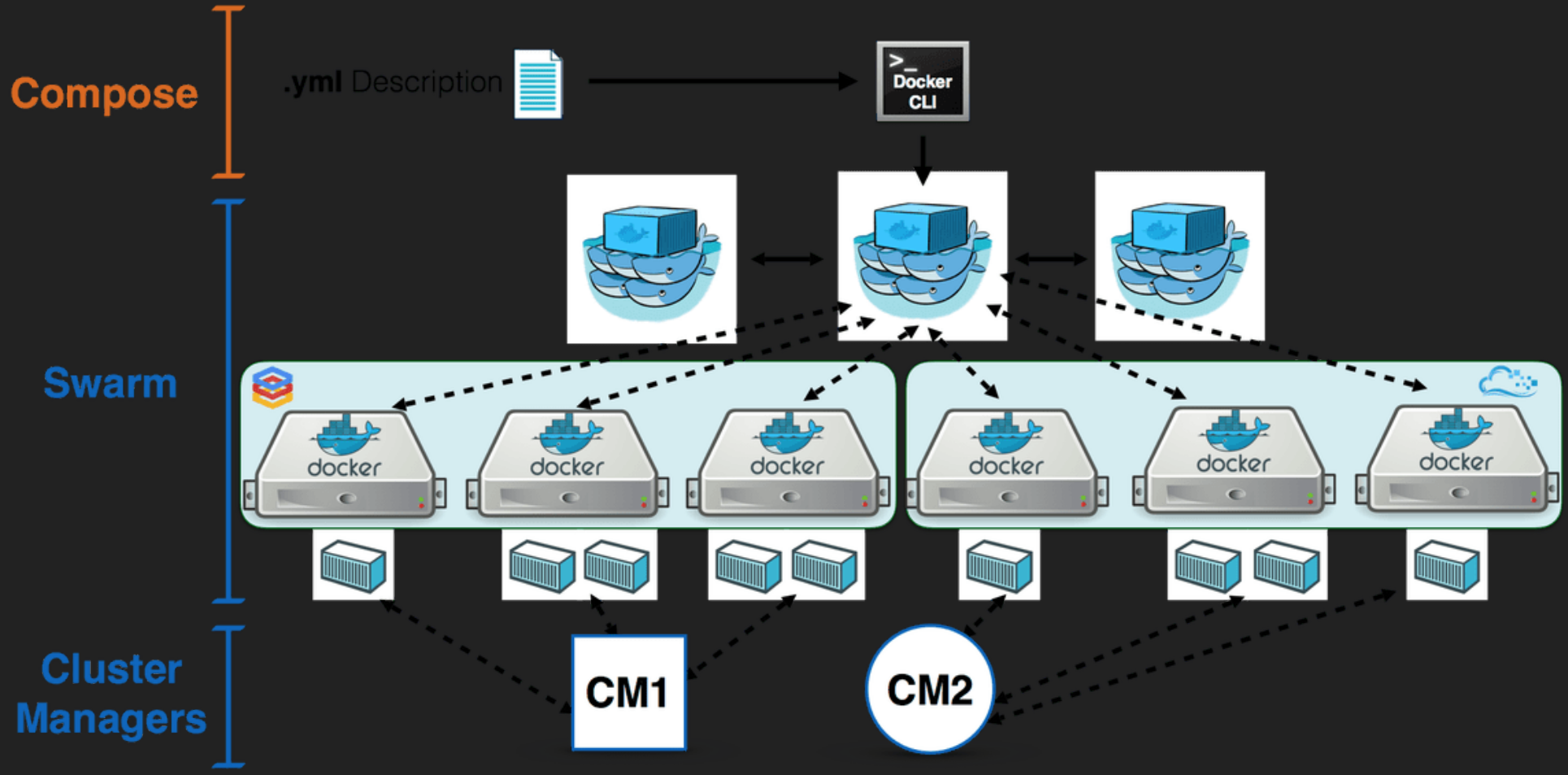
- pygeoapi Docker Images on DockerHub
- Images create Linux containers running content
- Use Docker Images to test locally or deploy in server

```
# Painless deployment
# pull from DockerHub
docker pull geopython/pygeoapi:latest
# run (default config)
docker run -p 5000:80 -it geopython/pygeoapi:latest
# test
curl http://localhost:5000
curl http://localhost:5000/collections
curl http://localhost:5000/collections/lakes/items?limit=2
curl http://localhost:5000/collections/lakes/items?f=html
```

DOCKER OPTIONS

- Custom config via Docker Volume Mapping
- Run with Docker Compose
- Subpath running via Flask `SCRIPT_NAME` environment variable
- Examples

SCALING PYGEOAPI



Cloud clustering with high availability

PRODUCTION INSTANCES

CANADA

METEOROLOGICAL SERVICE OF CANADA GEOMET PLATFORM

- OGC web services platform of Canadian weather, climate and water data
- Realtime weather
- Numerical weather prediction
- Hydrometric and Climate archives

MSC PYGEOAPI IMPLEMENTATION

- Climate services data portal
- climatedata.ca
- https://geo.weather.gc.ca/geomet/features

GLOSIS - GLOBAL SOIL INFORMATION

- Global soil information system under development by FAO and ISRIC
- Using pygeoapi as an external API providing OGC API - Features
- Testbed for JSON-LD for soil data distribution
- https://glosis.isric.org/ogc

ROADMAP

- More OGC APIs: Maps, Records
- OGC API Improvements: Processes
- More data providers
- Advanced filters (CQL)
- Transactions
- Content negotiation (e.g. Response as GeoPackage, GML,...)
- OSGeoLive

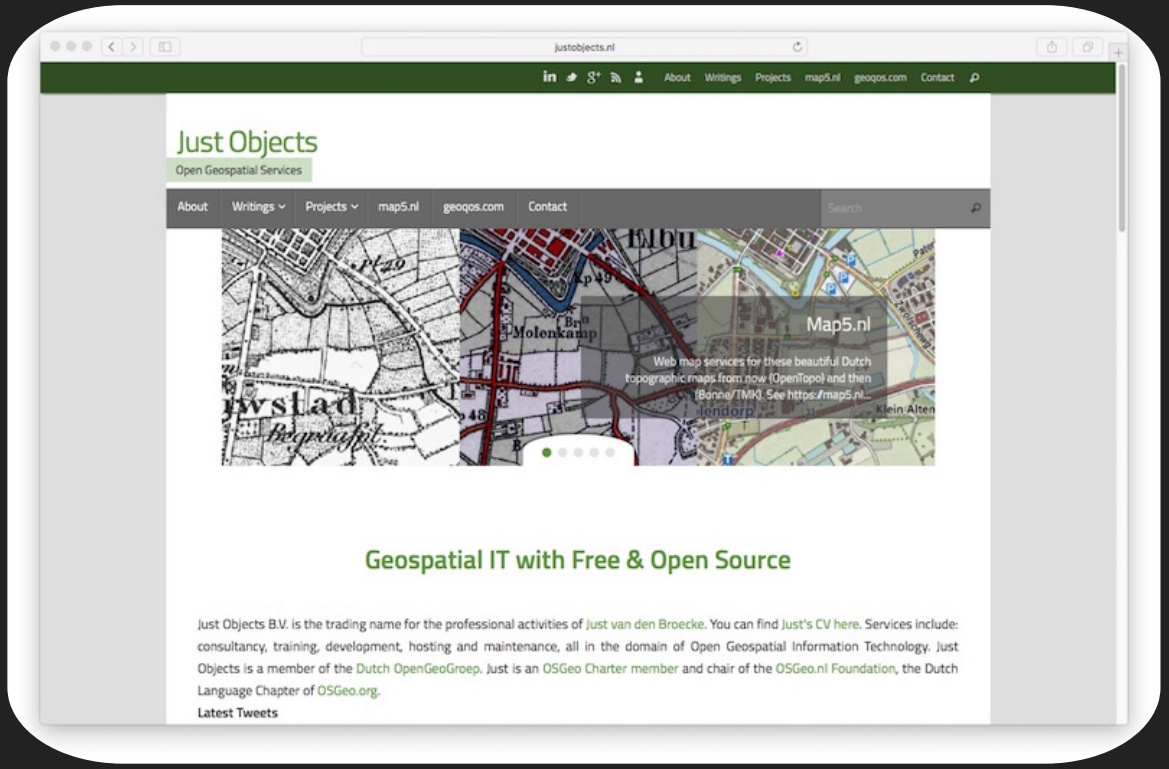
SUPPORT

Need support for WFS3.0???

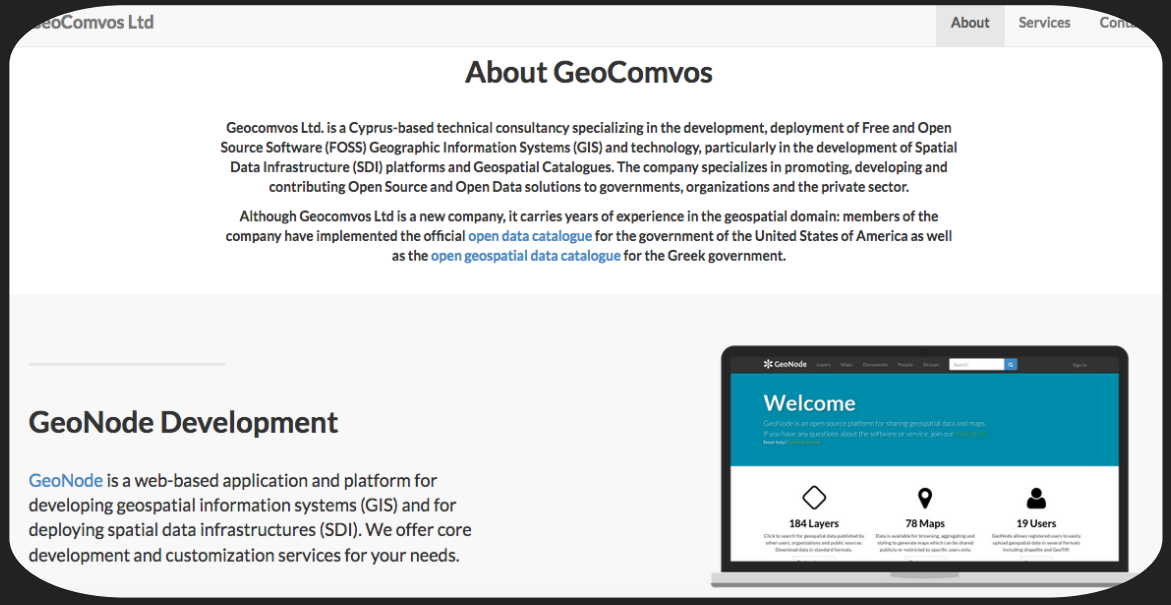
GeoCat can help you

GEOCAT

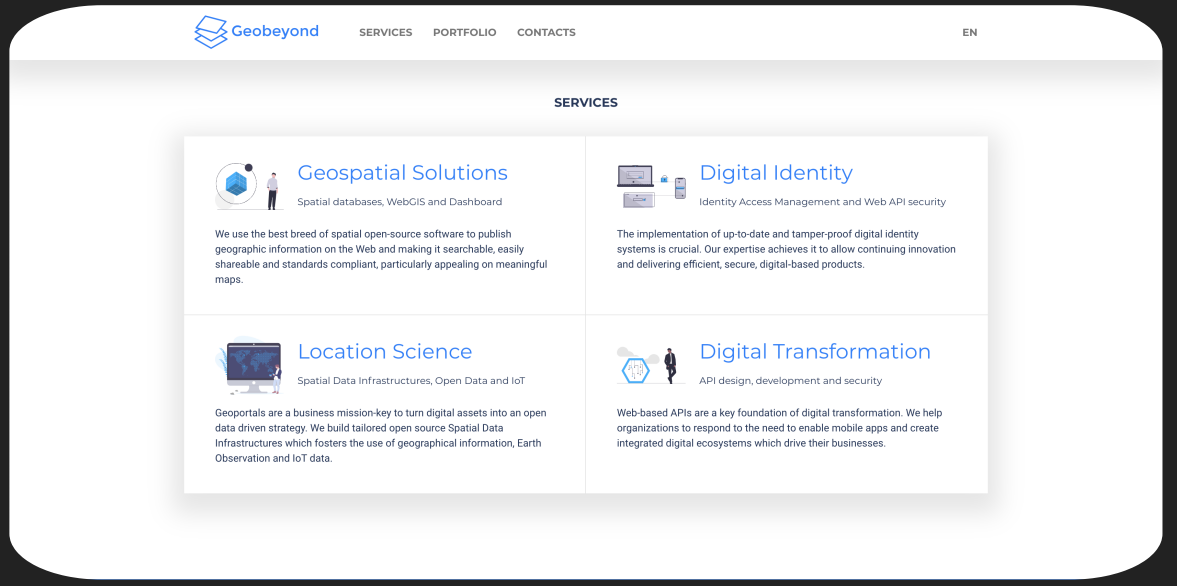
GOVERNMENT GEOGRAPHIC DATA PUBLISHING



justobjects.nl



GeoComvos ltd



Geobeyond Srl

LINKS

- Home: https://pygeoapi.io
- Docs: https://docs.pygeoapi.io
- Demo: https://demo.pygeoapi.io
- GitHub: https://github.com/geopython/pygeoapi
- Docker: https://hub.docker.com/r/geopython/pygeoapi
- Chat: https://gitter.im/geopython/pygeoapi
- Mail: https://lists.osgeo.org/mailman/listinfo/pygeoapi

THANK YOU!

@tomkralidis @JMendesDeJesus @normanbarker @perrygeo @justb4 @pvgenuchten @tzotsos @alpha-beta-soup @francbartoli