



AN OGC API TO GEOSPATIAL DATA

pygeoapi.io



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

TABLE OF CONTENTS

- Geospatial API evolution
- REST/JSON/OpenAPI/Swagger
- OGC activity
- Say hello to pygeoapi
- Core capabilities
- Deploying
- Production instances
- Roadmap
- Support

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)

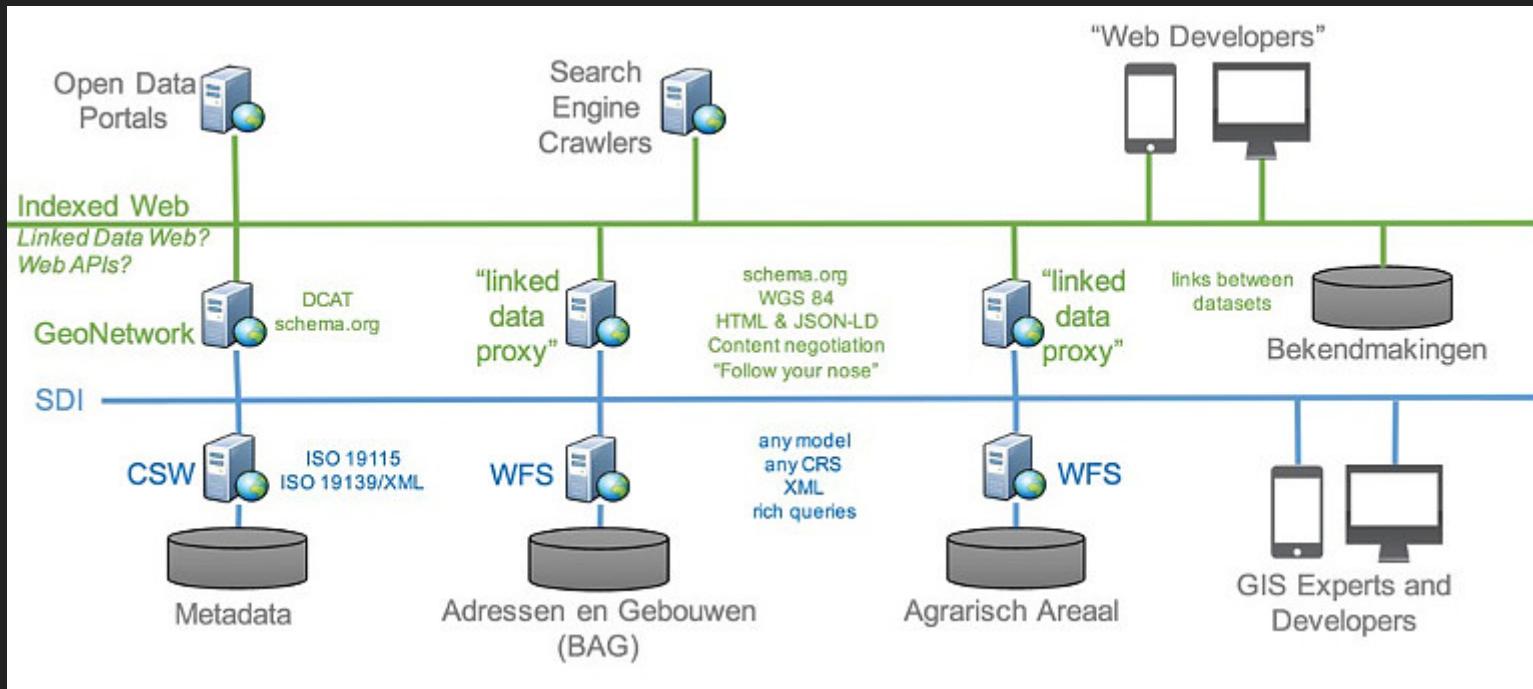
2000S

- 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

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

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



- 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)

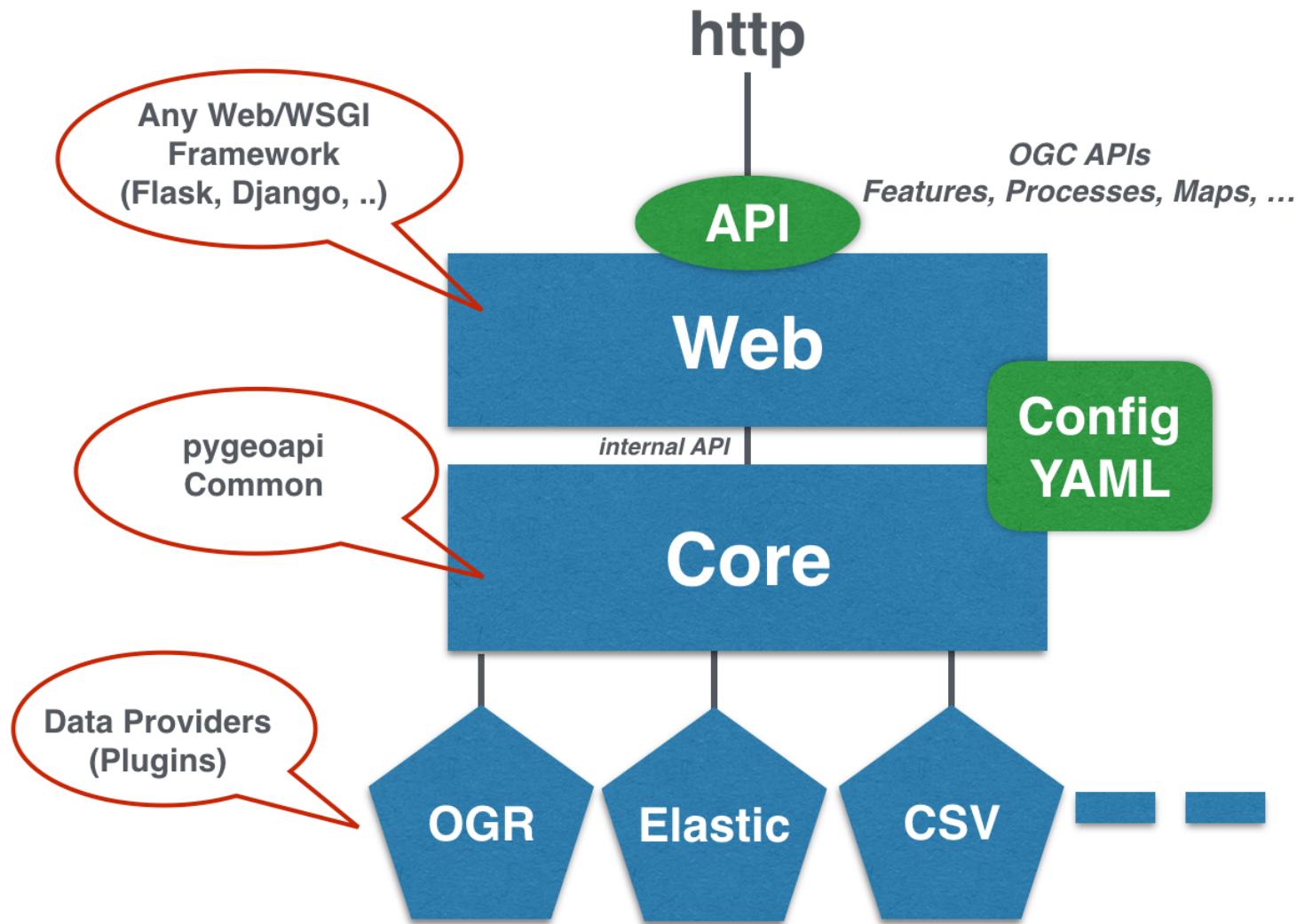
TECHNICAL OVERVIEW

- 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



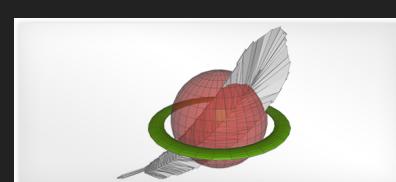
FEATURE DATA PROVIDERS



Elasticsearch



CSV



Spatialite*



GeoJSON*

* also via GDAL-OGR Provider

FEATURE DATA PROVIDERS - GDAL/OGR

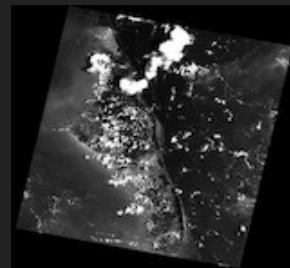


- Via OGR-Python bindings
- Unleashes ~100 Vector Formats! (WFS 1/2, GeoPackage, Shapefile, GeoJSON, PostGIS, ...)
- Paging, reprojection, spatial/attribute query
- Unlock existing WFS 1/2s!

COVERAGE DATA PROVIDERS



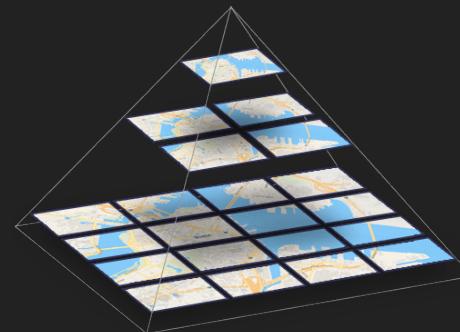
xarray



rasterio

- NetCDF, Cloud Optimized GeoTIFF (COG)

TILE DATA PROVIDERS



minio

ZXY directory tree

- MapBox Vector Tiles via tippecanoe

SCHEMA.ORG

The screenshot shows a web application interface for managing datasets. At the top, there's a navigation bar with links to 'Home', 'Collections', and 'Windmills Within The Netherlands'. Below this is a section titled 'Windmills within The Netherlands' with a subtitle: 'Locations of windmills within the Netherlands from Rijksdienst voor het Cultureel Erfgoed (RCE) INSPIRE WFS.' A row of buttons includes 'Netherlands', 'INSPIRE', 'Windmills', 'Heritage', 'Holland', and 'RD'. A note below says 'Powered by pygeoapi 0.6.0'. A horizontal menu bar has tabs for 'Elements' (which is selected), 'Console', 'Sources', 'Network', 'Performance', 'Memory', 'Application', 'Security', and 'Audits'. The main content area displays the following JSON-LD code:

```
<section id="collection" itemscope itemtype="http://schema.org/Dataset">
  <span itemprop="includedInDataCatalog" itemscope itemtype="http://schema.org/DataCatalog">
    <meta itemprop="url" content="https://demo.pygeoapi.io/master/collections/dutch_windmills">
    <meta itemprop="name" content="pygeoapi Demo instance - running latest GitHub version">
    <meta itemprop="description" content="pygeoapi provides an API to geospatial data">
  </span>
  <h1 itemprop="name">Windmills within The Netherlands</h1>
  <meta itemprop="url" content="https://demo.pygeoapi.io/master">
  <p itemprop="description">
    "Locations of windmills within the Netherlands from Rijksdienst voor het Cultureel Erfgoed (RCE) WFS v2 backend via OGRProvider."
  </p>
```

- 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



Contact

pygeoapi Demo instance - running latest GitHub version

pygeoapi provides an API to geospatial data

[geospatial](#) [data](#) [api](#)

Terms of service <https://creativecommons.org/licenses/by/4.0/>

License [CC-BY 4.0 license](#)

Collections

[View the collections in this service](#)

Processes

[View the processes in this service](#)

API Definition

[Documentation](#)

[OpenAPI Document](#)

Conformance

[View the conformance classes of this service](#)

Provider

pygeoapi Development Team

<https://pygeoapi.io>

Contact point

Address

Mailing Address

City, Administrative Area

Zip or Postal Code

Canada

Email

you@example.org

Telephone

[+XX-XXX-XXX-XXXX](tel:+XX-XXX-XXX-XXXX)

Fax

[+XX-XXX-XXX-XXXX](tel:+XX-XXX-XXX-XXXX)

Contact URL

[Contact URL](#)

Hours

Hours of Service

Contact instructions

During hours of service. Off on weekends.

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"
        },
        {
          "type": "application/xml",
          "rel": "alternate",
          "title": "This document as XML",
          "href": "https://demo.pygeoapi.io/master/collections/obs?f=jsonxml"
        }
      ]
    }
  ]
}
```



Download

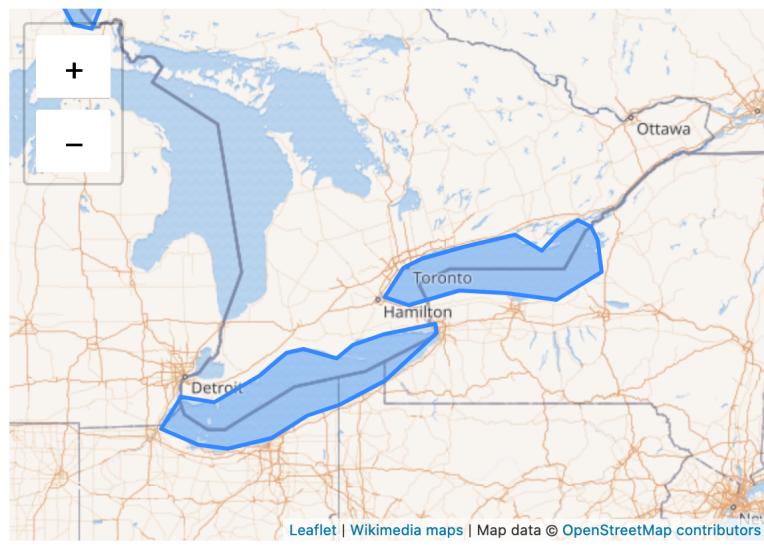
Response headers

```
access-control-allow-origin: *
content-length: 48471
content-type: application/json
date: Tue, 08 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://en.wikipedia.org/w/index.php?title=Lake_Baikal&oldid=9131111
1	1	0	Lake Winnipeg	https://en.wikipedia.org/w/index.php?title=Lake_Winnipeg&oldid=9131111
2	2	0	Great Slave Lake	https://en.wikipedia.org/w/index.php?title=Great_Slave_Lake&oldid=9131111
3	3	0	L. Ontario	https://en.wikipedia.org/w/index.php?title=L._Ontario&oldid=9131111
4	4	0	L. Erie	https://en.wikipedia.org/w/index.php?title=L._Erie&oldid=9131111
5	5	0	Lake Superior	https://en.wikipedia.org/w/index.php?title=Lake_Superior&oldid=9131111
6	6	0	Lake	https://en.wikipedia.org/w/index.php?title=Lake&oldid=9131111

OGC API - COVERAGES

 CoverageJSON [Playground](#) [Cookbook](#) [Tools](#) [Specification](#)

[Examples](#)



5000 km

TMP

1 {
2 "type": "Coverage",
3 "domain": {
4 "type": "Domain",
5 "domainType": "Grid",
6 "axes": {
7 "x": {
8 "start": -180.075,
9 "stop": 179.92500000000007,
10 "num": 2400
11 },
12 "y": {
13 "start": 90.075,
14 "stop": -90.075,
15 "num": 1201
16 }
17 },
18 "referencing": [
19 {
20 "coordinates": [
21 "x",
22 "y"
23],
24 "system": {
25 "type": "GeographicCRS",
26 "id": "http://www.opengis.net/def/crs/OGC/1.3/
27 }

OGC API - TILES

Large Lakes

lakes of the world, public domain

Tiles

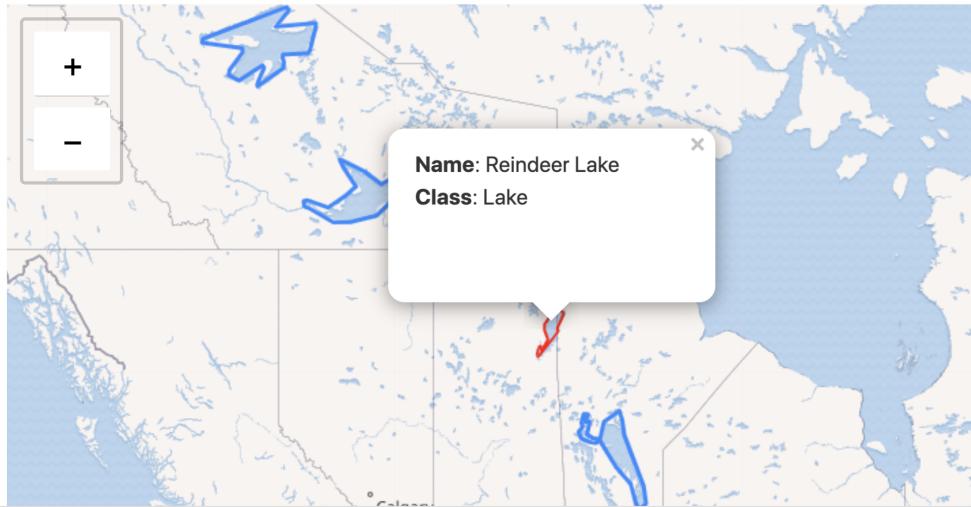
Tile Matrix Set

WorldCRS84Quad ▾

Metadata

[Tiles metadata in tilejson format](#)

Map



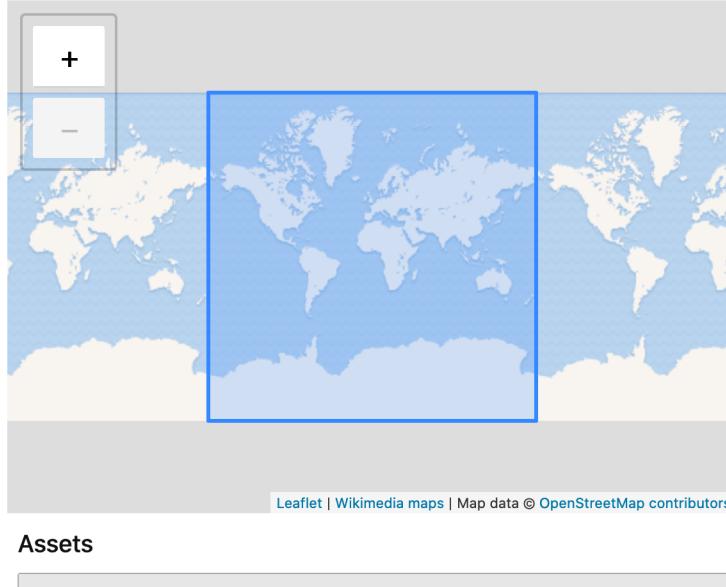
OGC API - PROCESSES

```
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)

[Contact](#)[Home](#) / [SpatioTemporal Asset Catalog](#) / [nwp](#) / [12](#) / [000](#) / [CMC_glb_TMP_TGL_2_latlon.15x.15_2020040112_P000](#)[JSON](#) [JSON-LD](#)

Item CMC_glb_TMP_TGL_2_latlon.15x.15_2020040112_P000



Property	Value
id	CMC_glb_TMP_TGL_2_latlon.15x....
GRIB_COMMENT	Temperature [C]
GRIB_DISCIPLINE	0(Meteorological)
GRIB_ELEMENT	TMP
GRIB_FORECAST_SECONDS	0 sec
GRIB_IDS	CENTER=54(Montreal) SUBCENTER=0 MASTER_TABLE=4 LOCAL_TABLE=0 SIGNIFICANT_TIME_1/Start of For

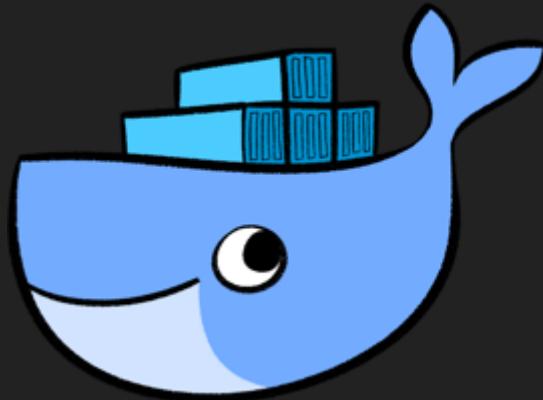
DEPLOYING

PACKAGING

Numerous **deployment options**

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

DOCKER OVERVIEW

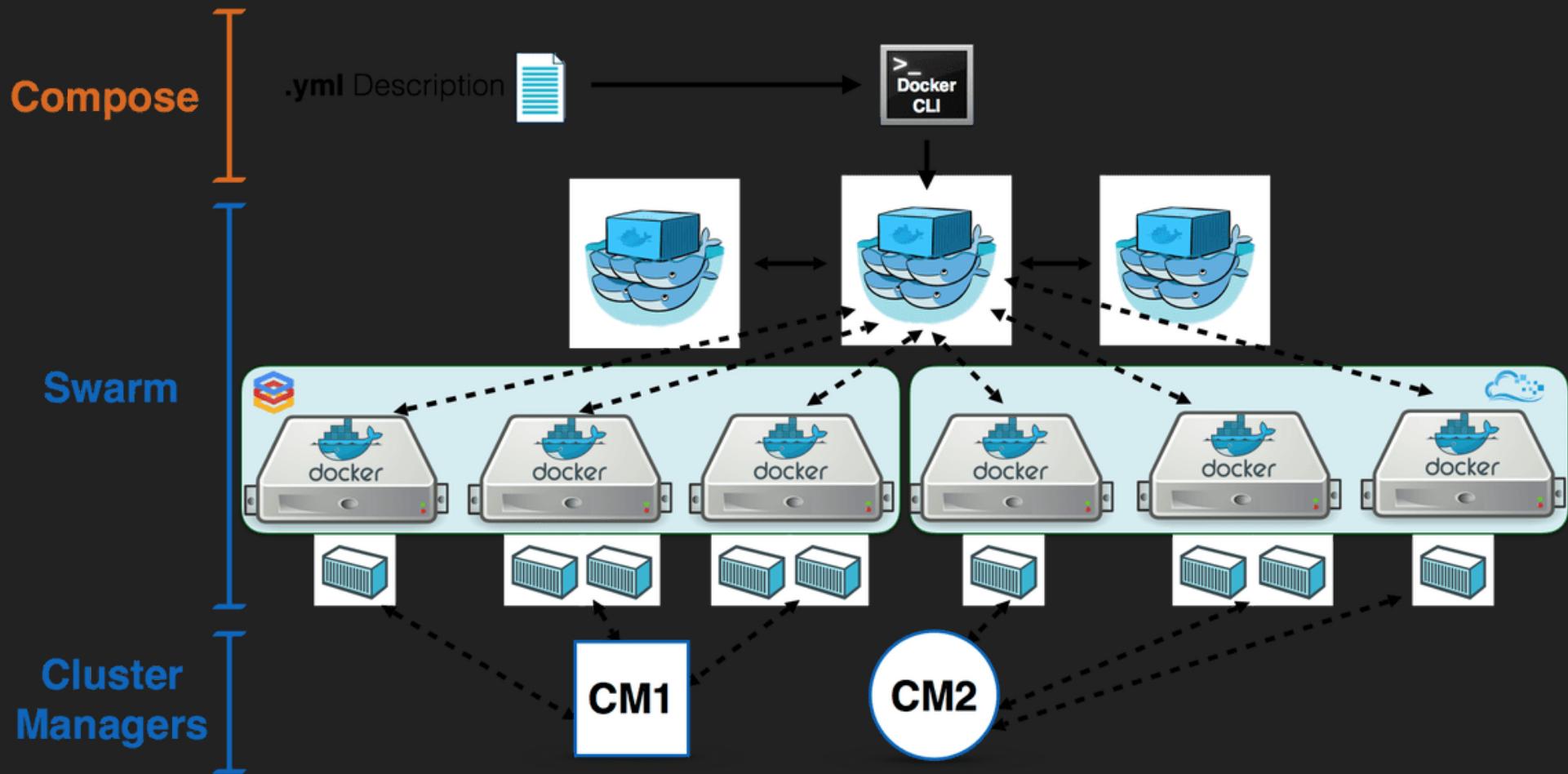


- pygeoapi [Docker Images](#) on DockerHub
- Images create Linux containers running content
- Use Docker Images to test locally or deploy in server

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**

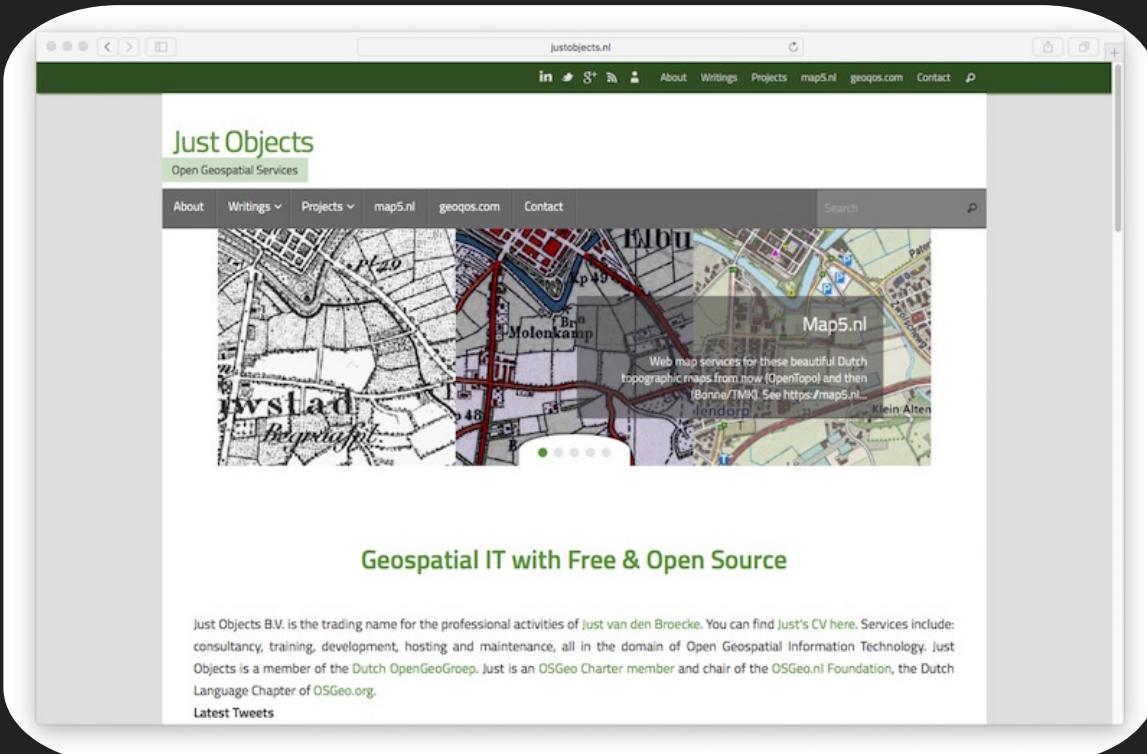
SUPPORT



Need support for WFS3.0???

GeoCat can help you





justobjects.nl

The image shows a screenshot of the GeoComvos Ltd website. At the top, there is a navigation bar with links for 'About', 'Services', and 'Contact'. The main content area has a title 'About GeoComvos' and a paragraph describing the company's focus on Free and Open Source Software (FOSS) Geographic Information Systems (GIS) and technology, particularly Spatial Data Infrastructure (SDI) platforms and Geospatial Catalogues. Below this, another paragraph notes the company's experience in geospatial domains, mentioning its role in implementing official data catalogues for the US government and the Greek government. A section titled 'GeoNode Development' follows, with a description of GeoNode as a web-based application for developing geospatial information systems (GIS) and spatial data infrastructures (SDI). To the right, there is a screenshot of the GeoNode platform's 'Welcome' page, which displays statistics such as 184 Layers, 78 Maps, and 19 Users.

GeoComvos Ltd

About Services Contact

About GeoComvos

Geocomvos Ltd. is a Cyprus-based technical consultancy specializing in the development, deployment of Free and Open Source Software (FOSS) Geographic Information Systems (GIS) and technology, particularly in the development of Spatial Data Infrastructure (SDI) platforms and Geospatial Catalogues. The company specializes in promoting, developing and contributing Open Source and Open Data solutions to governments, organizations and the private sector.

Although Geocomvos Ltd is a new company, it carries years of experience in the geospatial domain: members of the company have implemented the official [open data catalogue](#) for the government of the United States of America as well as the [open geospatial data catalogue](#) for the Greek government.

GeoNode Development

GeoNode is a web-based application and platform for developing geospatial information systems (GIS) and for deploying spatial data infrastructures (SDI). We offer core development and customization services for your needs.

Welcome

GeoNode is an open source platform for sharing geographical data and maps. If you have any questions about the software or service, join our [GeoNode Slack channel](#).

184 Layers

78 Maps

19 Users

GeoComvos ltd

The screenshot shows the 'SERVICES' section of the Geobeyond Srl website. The top navigation bar includes the logo 'Geobeyond' with a blue icon, followed by 'SERVICES', 'PORTFOLIO', 'CONTACTS', and 'EN'. Below the navigation, the 'SERVICES' heading is centered. The page is divided into four main service offerings, each with an icon, title, subtitle, and a brief description.

Service	Description
 Geospatial Solutions Spatial databases, WebGIS and Dashboard	We use the best breed of spatial open-source software to publish geographic information on the Web and making it searchable, easily shareable and standards compliant, particularly appealing on meaningful maps.
 Digital Identity Identity Access Management and Web API security	The implementation of up-to-date and tamper-proof digital identity systems is crucial. Our expertise achieves it to allow continuing innovation and delivering efficient, secure, digital-based products.
 Location Science Spatial Data Infrastructures, Open Data and IoT	Geoportals are a business mission-key to turn digital assets into an open data driven strategy. We build tailored open source Spatial Data Infrastructures which fosters the use of geographical information, Earth Observation and IoT data.
 Digital Transformation API design, development and security	Web-based APIs are a key foundation of digital transformation. We help organizations to respond to the need to enable mobile apps and create integrated digital ecosystems which drive their businesses.

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