contact@odatis-ocean.fr
www.odatis-ocean.fr

ODATIS
OCEAN DATA INFORMATION AND SERVICES

Atelier Technique "Accès aux Données"
10-11 octobre 2024

OGC API

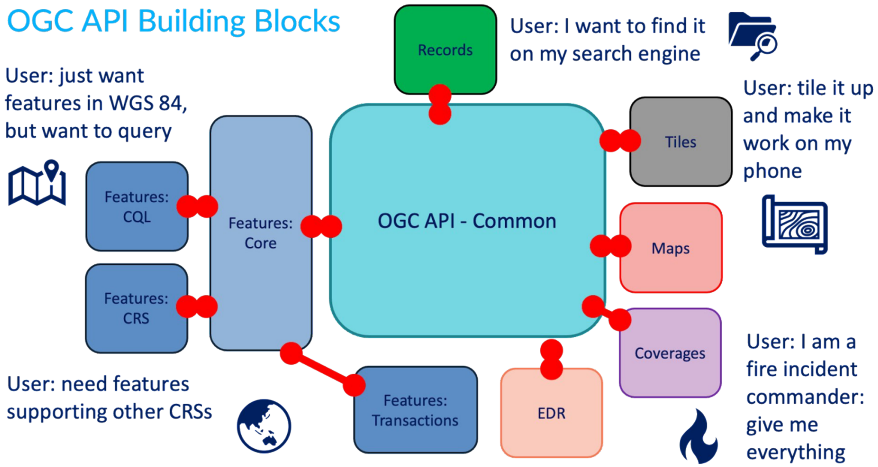
Dimitry Khvorostyanov

En bref...

- OGC API : c'quoi et pourquoi ?
- État actuel
- Prospective

OGC API : c'quoi et pourquoi ? (1/4)

- *"...designed to make it easy for ANYONE to provide and use geospatial data on the web, and to integrate this data with ANY other type of information"* - https://ogcapi.ogc.org/

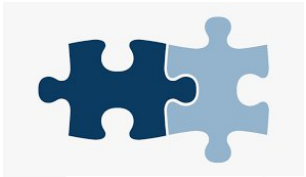


OPENAPI INITIATIVE

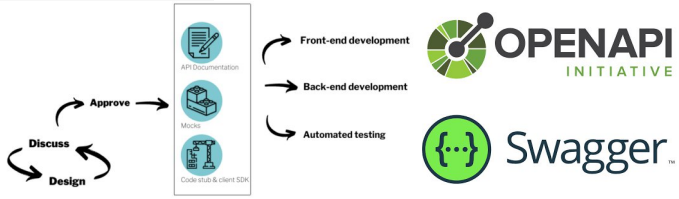
OGC API : c'quoi et pourquoi ? (2/4)

Building blocks simplifiant les "3 I" :

- Intégration



- Implémentation



- Interopérabilité

OGC API : c'quoi et pourquoi ? (3/4)

Exemples requêtes :

Service	OGC API	OGC WCS, WMS, etc.
Coverages	/collections/{collectionId}/coverage /collections/sst/coverage?subset=time("2020-01-01")	GetCoverage&coverageId={coverageId} GetCoverage&coverageId=sst&subset=time("2020-01-01")
Maps	/collections/{collectionId}/map /collections/temperature/map?bbox=-10,10,-5,15	GetMap&layers={layer} GetMap&layers=temperature&bbox=-10,10,-5,15

OGC API : c'quoi et pourquoi ? (4/4)

Exemples requêtes et réponses:

Service	OGC API	OGC WCS, WMS, etc.
Requête	https://example.com/ogc-api/collections/roads/items? bbox=-10,10,-5,15& f=application/geo+json	https://example.com/wfs? SERVICE=WFS& VERSION=2.0.0& REQUEST=GetFeature& TYPENAME=roads& BBOX=-10,10,-5,15& OUTPUTFORMAT=application/gml+xml

Réponse	{
	"type": "FeatureCollection",
	"features": [
	{
	"type": "Feature",
	"geometry": {
	"type": "Point",
	"coordinates": [-7.5, 12.5]
	},
	"properties": {
	"id": 1 }
	}]}
		<wfs:FeatureCollection>
		<gml:featureMember>
		<feature:Roads>
		<feature:id>1</feature:id>
		<gml:Point srsName="EPSG:4326">
		<gml:coordinates>-7.5,12.5</gml:coordinates>
		</gml:Point>
		</feature:Roads>
		</gml:featureMember>
		</wfs:FeatureCollection>

OGC API : les services

The core OGC API standards include:

- OGC API - Features (equivalent to WFS)
- OGC API - Maps (equivalent to WMS)
- OGC API - Coverages (equivalent to WCS)
- OGC API - Processes (~WPS)
- OGC API - Tiles (~WMTS)[1]
- OGC API - Records (~CSW)[2]
- OGC API - Styles (~SLD & SE)[3]

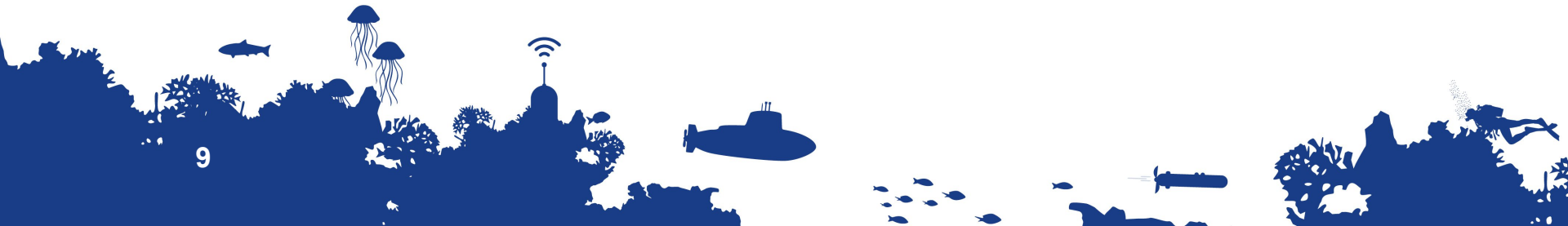
[1]Inclut vector tiles, coverage tiles, map tiles (pas que map comme WMTS)
[2]Effort en cours pour la compatibilité avec STAC
[3]La lisibilité et souplesse entre xml et json – pas photo !

OGC API : État actuel (1/2)

Status : draft

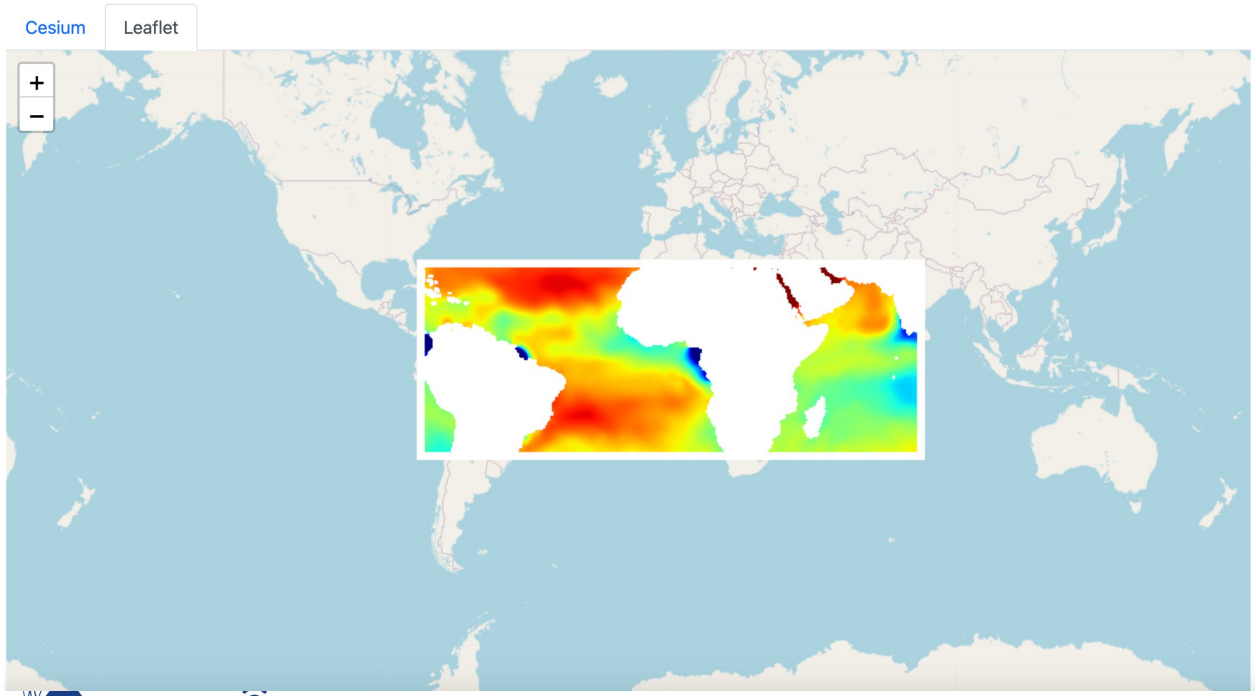
OGC API : État actuel (2/2)

OGC API Service	pygeoapi	MapServer	GeoServer
Features	Full	Partial	Full
Maps	Partial	Partial	Partial
Tiles	Partial	Partial	Partial
Processes	Full	None	Partial
Records	Full	None	None
Styles	None	None	Partial



OGC API : Prospective (1/4)

- Implémentation des briques manquantes : exemple OGC API - Maps NetCDF



OGC API : Prospective (2/4)

- Implémentation des briques manquantes :
  - Exemple OGC API - Maps NetCDF

- Micro-services "maison"

- Intégration avec *pygeocapi*

pygeoapi

  - Directe
  - Routage

OGC API : Prospective (3/4)

- Intégration *directe* dans pygeocapi

- Créer un nouveau "*Provider*" :

```
class NetCDFMapProvider(BaseProvider):
    """NetCDF provider for OGC API - Maps"""

    # Class attributes for configuration
    DATA_DIR = '/path/to/your/netcdf/files/'
    FILE_PREFIX = 'report_'
    FILE_EXTENSION = '.nc'
```

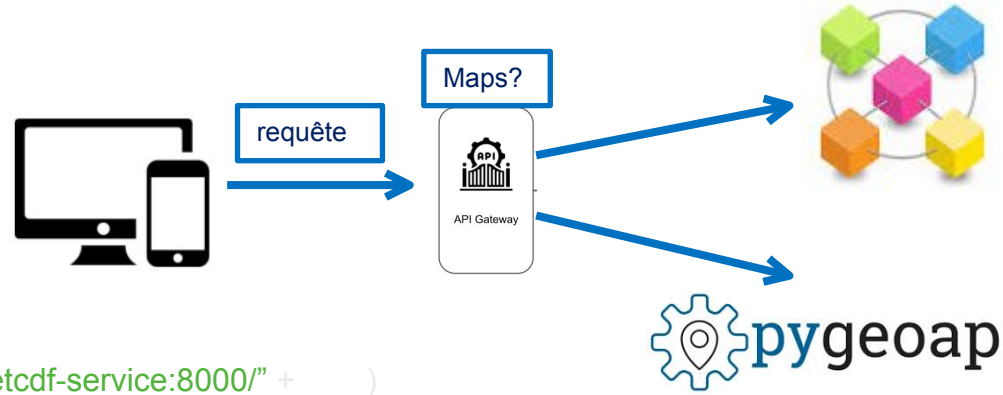
- Le décrire dans la config de pygeoapi :

```
providers:
    - type: map
      name: NetCDFMapProvider
      data: /path/to/your/netcdf/file.nc
      options:
          variable: SSS
          format:
            name: png
            mimetype: image/png
```

OGC API : Prospective (4/4)

- Intégration dans *pygeocapi :* routage

  - Créer un "*Gateway*" :

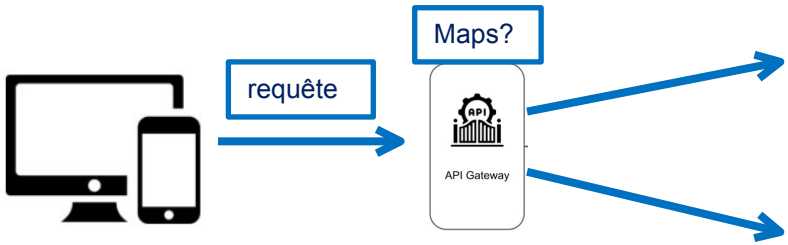



```
@app.get("/{path:path}")
async def gateway(path: str):
    if path.startswith("collections") and "map" in :
        return await forward_request("http://ogc-maps-netcdf-service:8000/" + )
    else:
        return await forward_request("http://pygeoapi:5000/" + )
```

Conclusion

- Perspectives prometteuses, c'est l'avenir
- Il faut être patients...
- ...ou compléter des services existants par des solutions "maison"
- Le routage entre des micro-services "maison" et des logiciels existants est une façon plus directe à mettre en place – à privilégier ?

OU



pygeoapi

contact@odatis-ocean.fr
www.odatis-ocean.fr

ODATIS

OCEAN DATA INFORMATION AND SERVICES

Merci de votre attention !