

- API and indexing
- The power of RUST and C
- Output in NetCDF, Json, Parquet
- Demonstrated for WOD, SeaDataNet, ARGO, NL Rijkswaterstaat



BEACON

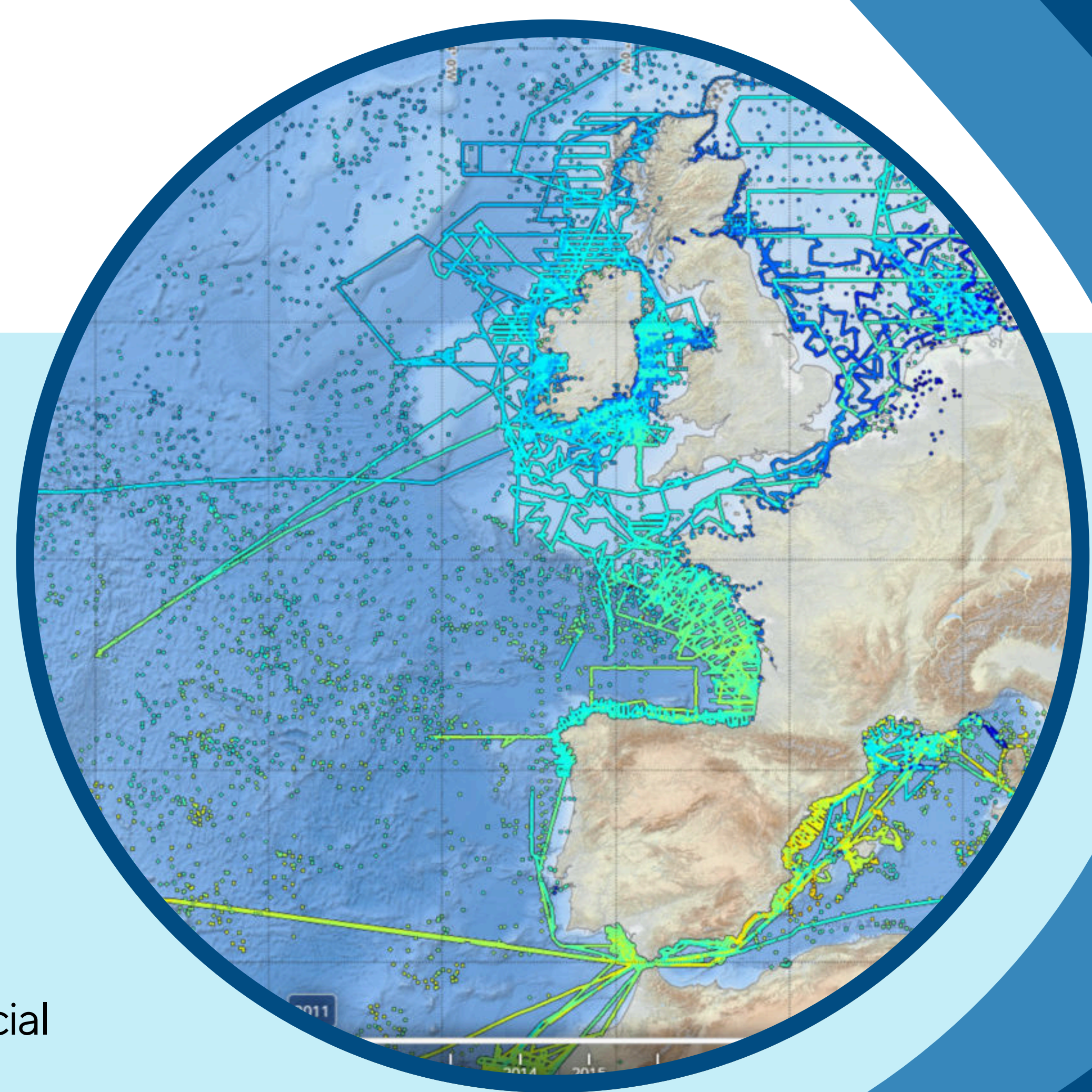
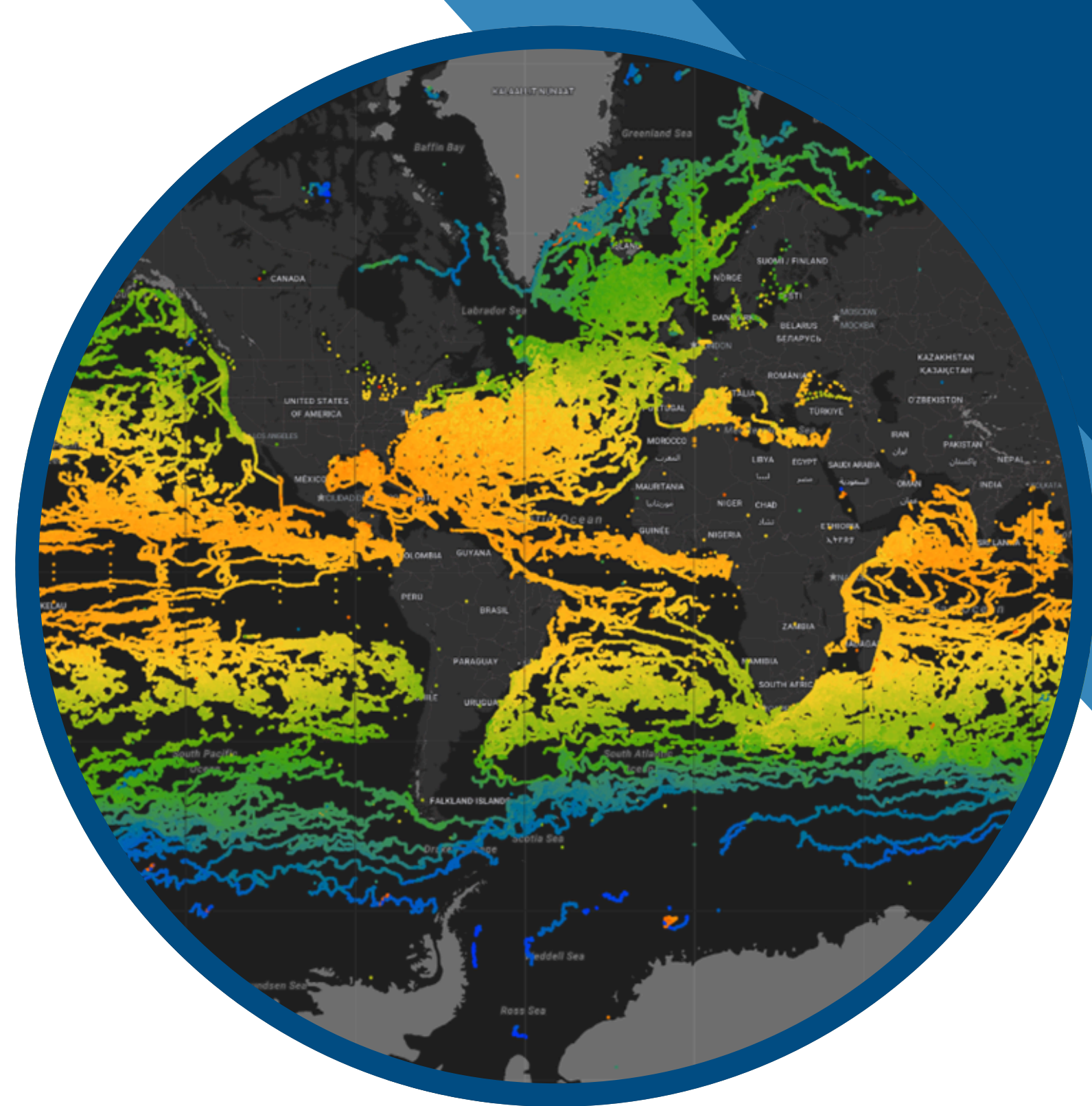
The high performance datalake

Unlock the Power of Your Data!

How to go from files to serving applications as an actual "Data lake"?

BEACON provides an easy-to-use, fast, reliable, and scalable solution for storing, processing, and retrieving large amounts of climate data.

Data federation, unified output to support models and virtual research!

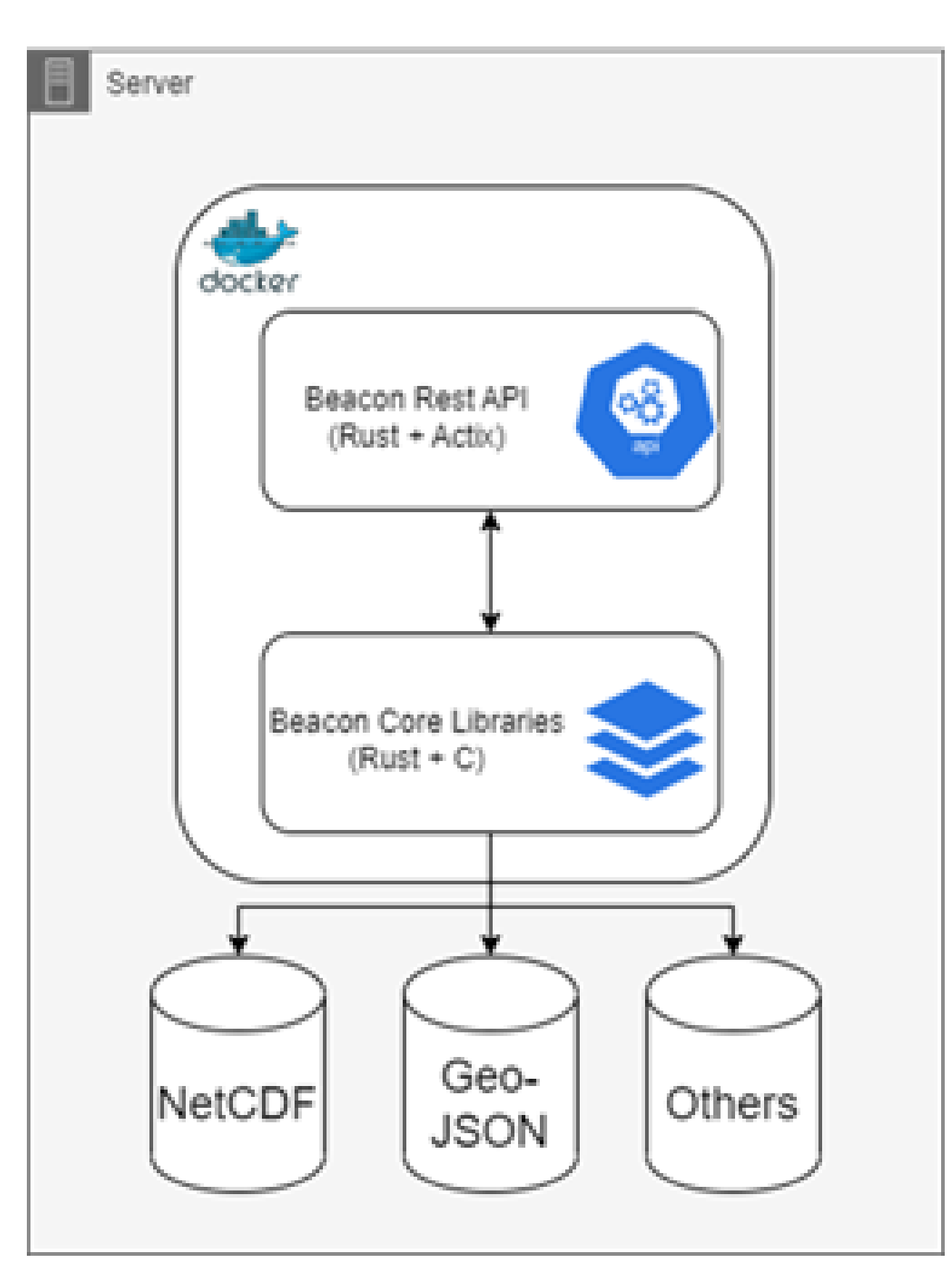


Specs

- ✓ Ultra-fast subsetting in millions of files
- ✓ Easy to deploy and open source
- ✓ Compact data storage
- ✓ Free for research and non-commercial users
- ✓ Powerful data harmonisation
- ✓ Effortless querying from notebooks and applications

Try it for your data →

Contact MARIS
 beacon.maris.nl

```

import requests

url = "https://beacon.maris.nl/api/query"
headers = {"Content-Type": "application/json"}

data = {
    "query_parameters": [
        {"data_parameter": "sea_water_temperature", "unit": "degrees_Celcius"},
        {"data_parameter": "time", "unit": "days since 1970-01-01 00:00:00 UTC", "alias": "TEMPORAL"},
        {"data_parameter": "sea_water_pressure", "unit": "decibar"},
        {"data_parameter": "latitude", "unit": "degrees_north"},
        {"data_parameter": "longitude", "unit": "degrees_east"},
    ],
    "filters": [{"for_query_parameter": {"alias": "TEMPORAL", "min": 21900, "max": 22265 }},
    "output": {"format": "netcdf"},
}

response = requests.post(url, json=data, headers=headers)
    
```

