

Beacon

A high-performance ARCO data lake for accessing & sub-setting large quantities of climate data

*Robin Kooyman
Paul Weerheim*

Origin of BEACON

Challenge:

- Large collections of observation data (e.g. SeaDataNet CDI) can contain millions of files.
- Access to files (datasets) is possible, but data sub-setting proves to be difficult.
- How to optimize the systems for **Machine2Machine access to subsets**, enabling easy access to Jupyter Notebooks and other applications.
- **How to go from files to serving applications as an actual “Data lake”?**

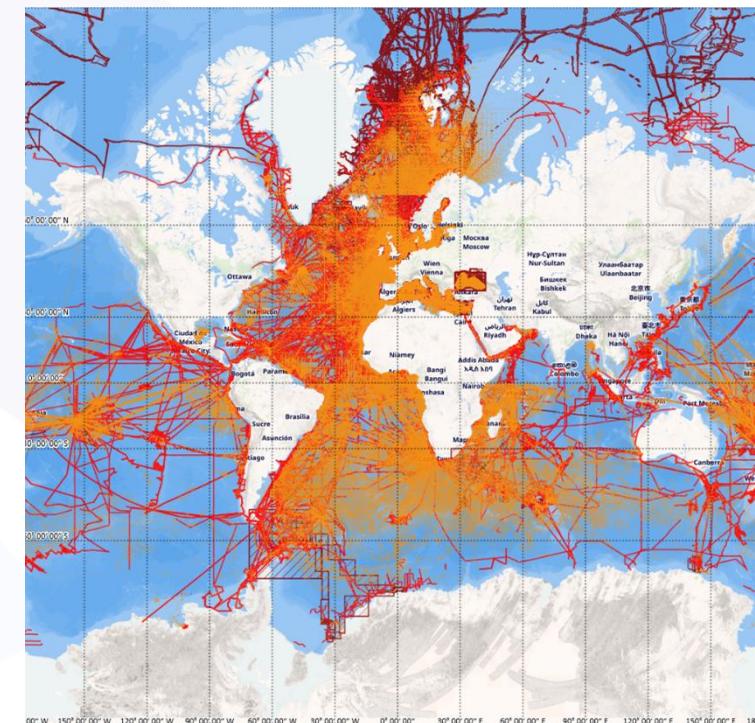
Example:

- Request: Give me all the temperature data in the North Sea, from 2010-2020, in degrees Celsius, at a depth of 0-50 m.
- Response: One NetCDF file containing exactly this data, which is then on the fly, directly usable in a Jupyter notebook and for HPC.

The goal of **BEACON** is to provide an easy-to-use, fast, reliable, and scalable solution for storing, processing, and retrieving large amounts of climate data.

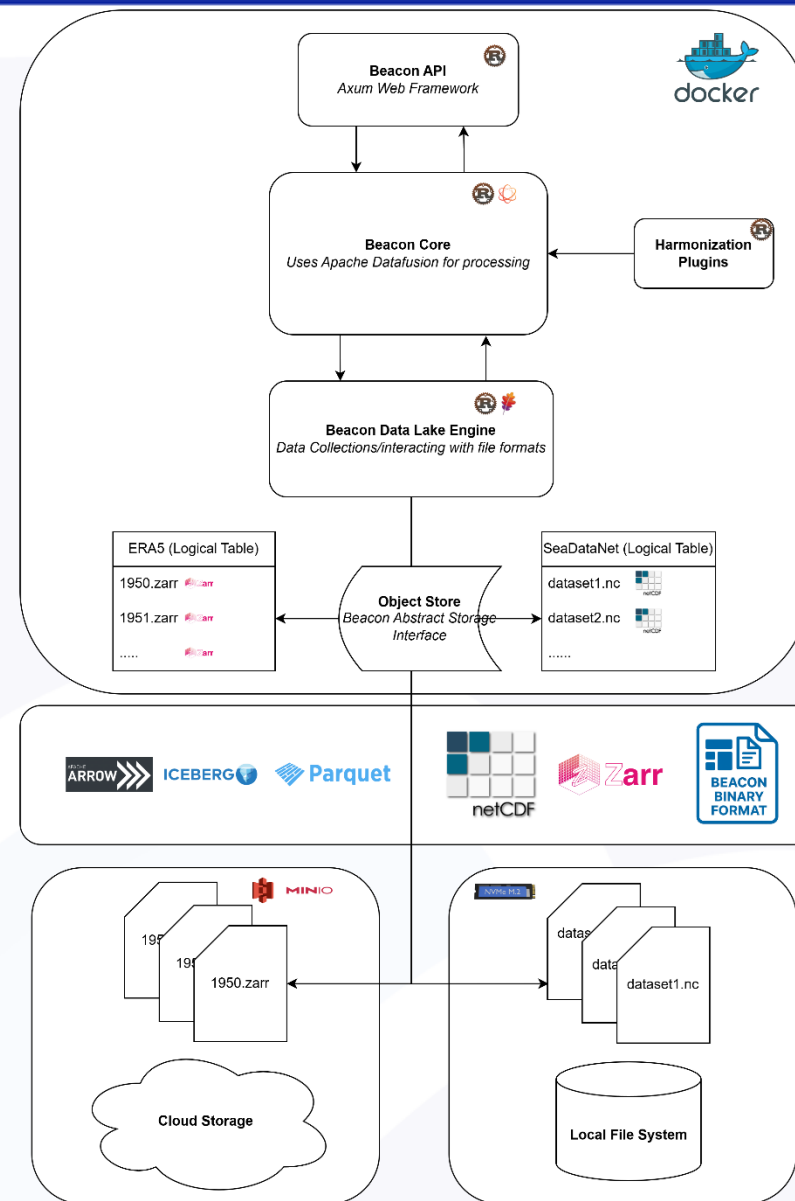
Beacon has been part of various EU projects such as **PHIDIAS HPC** & **EOSC-Future** and is currently part of **FAIR-EASE**, **Blue-Cloud 2026** and national initiatives.

SeaDataNet CDI



Beacon in a nutshell

- **Open-Source High Performance ARCO Data Lake**
- **Written in Rust + C**
- **Runs on any platform using Docker**
- **Consists of:**
 - Rest API
 - Core Libraries (heavy lifting)
 - Data Harmonization Libraries (single output file)
- **Supports Reading & Writing data from/to**
 - Any S3 Compatible Storage
 - AWS S3
 - MinIO
 - Any other S3 bucket
 - Local File System
- **Supported Data Formats (s3, local, nas)**
 - CSV
 - Zarr (any structure, timeseries, gridded, cruises & supports parallel chunked reading)
 - Parquet
 - Apache Iceberg
 - Arrow
 - NetCDF (any structure, timeseries, gridded, cruises & supports chunked reading)
 - Beacon Binary Format (also ARCO)
 - Icechunk (Work in Progress)
- **Creating Real Time Collections (Logical Tables)**
 - On top of existing files:
 - Eg: seadatanet/*.nc
 - Eg: era5/*.zarr



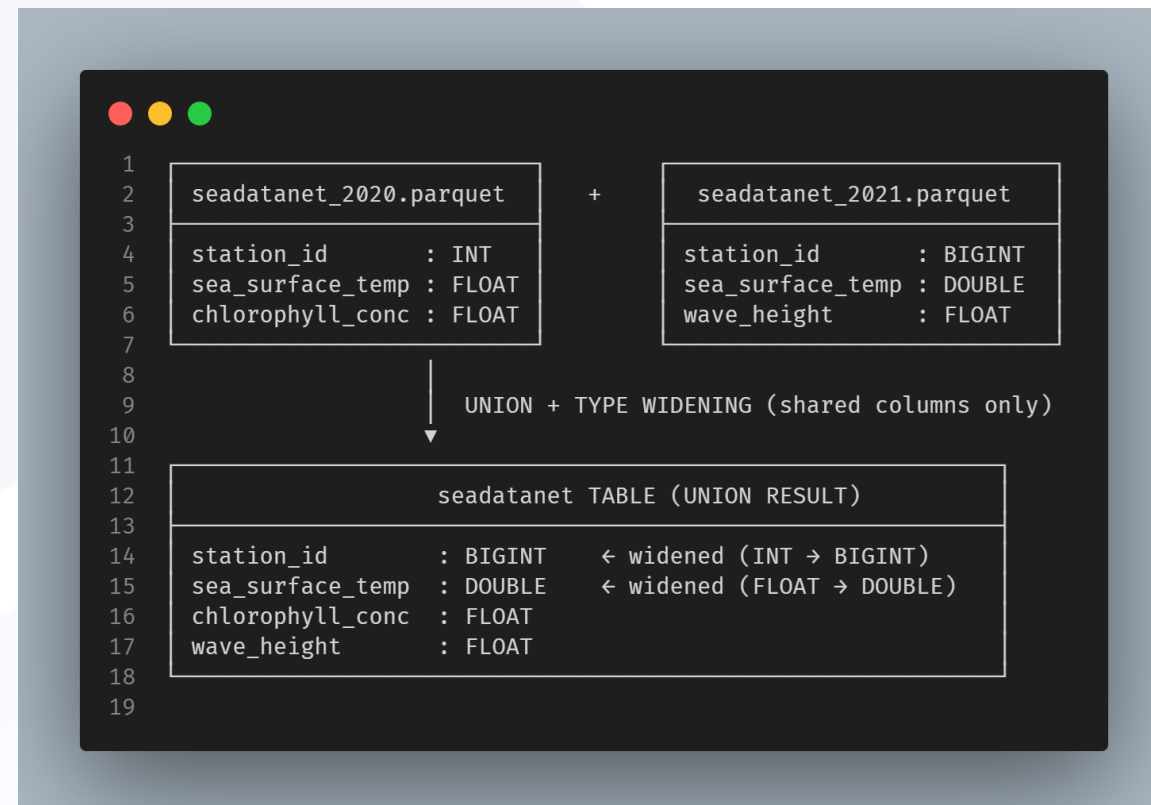
Beacon in a nutshell 🤖

• Harmonization







- N-dimensional structure alignment
- Schema Harmonization of datasets
 - Merge Schema with safe upcasting
 - Type Widening
- Unit Conversions
- Parameter Aggregations

• Performance

- Apache Arrow
 - Apache DataFusion
- Projection Pushdown
 - Only read columns that are being queried
- Pruning Predicates
 - Filter out not relevant datasets/rows/row-groups before reading
- Filter Pushdown
 - Filter data at the scan/read level



Beacon in a nutshell

- **Query using JSON**
 - Select relevant columns
 - Filter using:
 - Value Ranges / Equalities
 - Metadata
 - Polygons
- **Query using Python Library**
 - PyPi: <https://pypi.org/project/beacon-api/>
 - GitHub: <https://github.com/maris-development/beacon-py>
 - Documentation: https://maris-development.github.io/beacon-py/getting_started/
- **Query using SQL**
 - Full SQL support (read-only)
 - Write for admins only
- **Output formats:**
 - ODV ASCII
 - CSV 
 - NetCDF 
 - Arrow 
 - Parquet 
 - GeoParquet 
 - Zarr 

Create and execute a query

To create and execute a query on a specific table, you can use the `query` method of the `Table` object. Here's an example of how to create a simple query that selects specific columns and applies filters:

```
df = (
    tables['default'] # Select the 'default' table as our data source
    .query() # Create a new query on the selected table
    .add_select_column("LONGITUDE") # Select the LONGITUDE column
    .add_select_column("LATITUDE") # Select the LATITUDE column
    .add_select_column("JULD")
    .add_select_column("PRES")
    .add_select_column("TEMP")
    .add_select_column("PSAL")
    .add_select_column(".featureType") # Select the .featureType column
    .add_select_column("DATA_TYPE")
    .add_range_filter("JULD", "2020-01-01T00:00:00", "2021-01-01T00:00:00") # Filter
    .add_range_filter("PRES", 0, 10) # Filter for pressure between 0 and 10 dbar for
    .to_pandas_dataframe() # Execute the query and return the results as a Pandas DataFrame
)
df
```

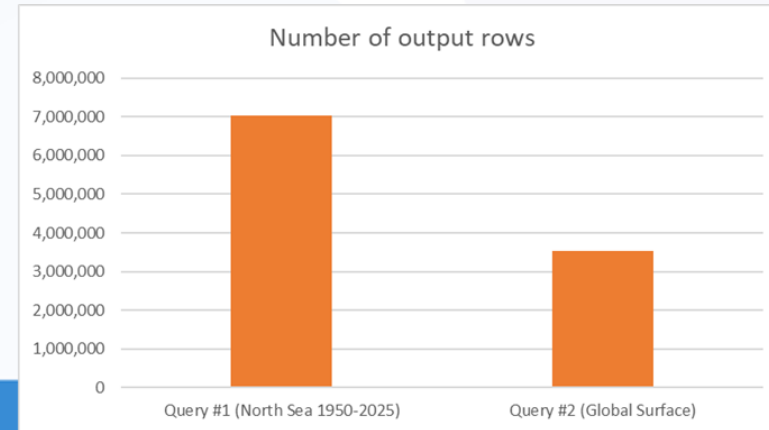
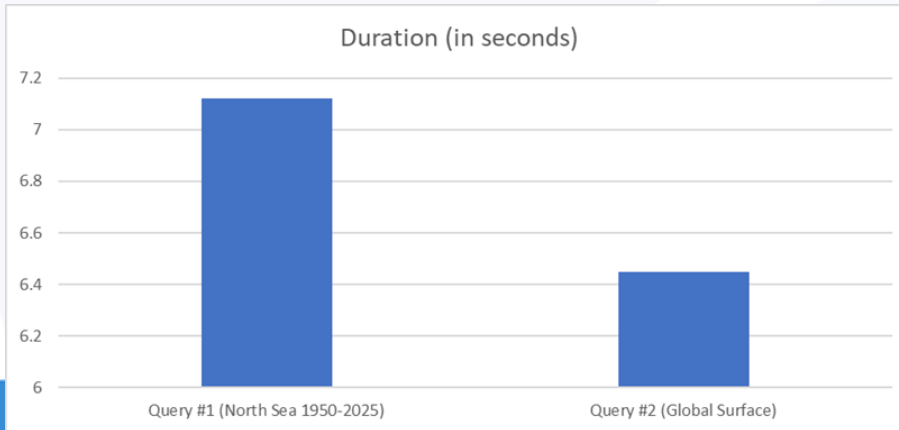
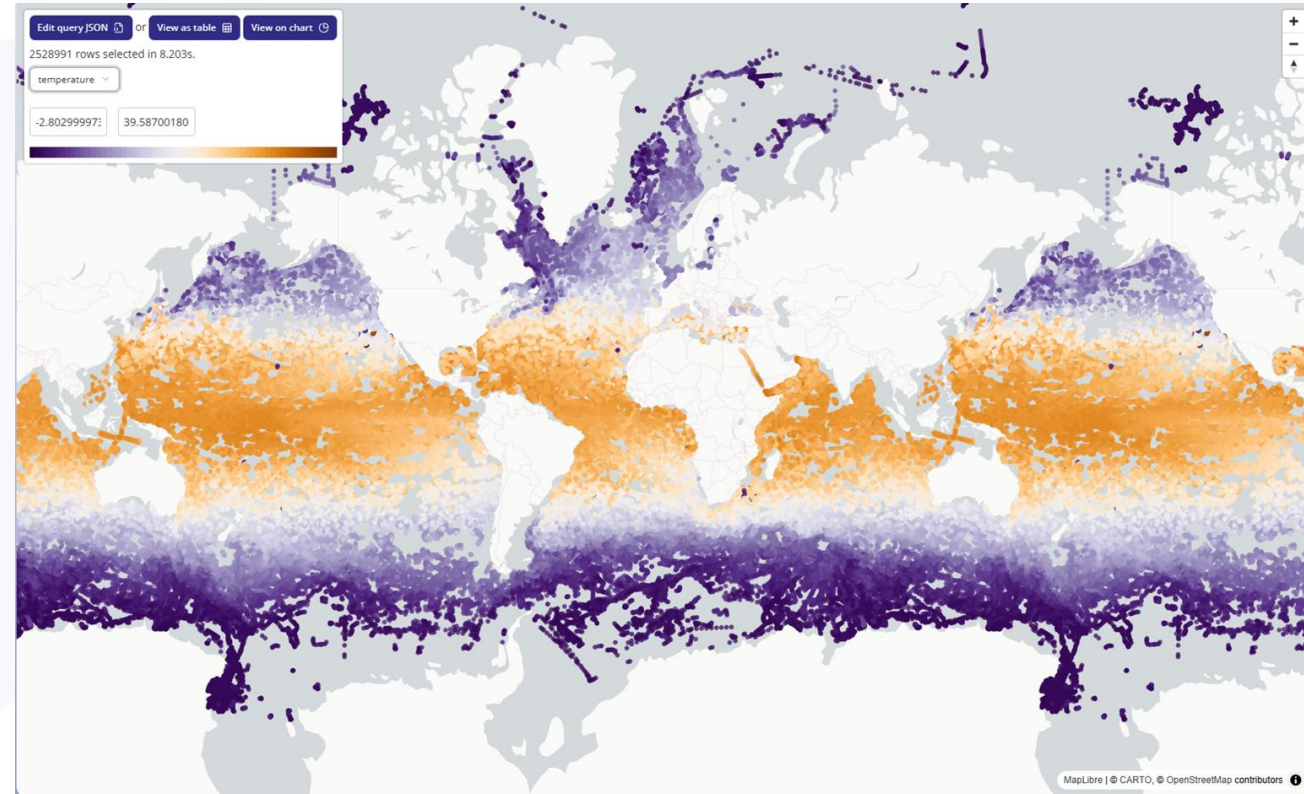
Note

The `to_pandas_dataframe` method executes the query and returns the results as a Pandas DataFrame.

Beacon In-Situ Performance



- **World Ocean Database**
 - ~20 Million individual NetCDF Files
 - ~600GB of data, compressed to 50GB BBF files inside a Cloud Bucket
- **Benchmark Queries**
 - Selected Columns:
 - Temperature + QC
 - Longitude
 - Latitude
 - Depth
 - Time
 - Platform
 - Institute
 - Query #1 Range: North Sea, 1950 - 2025
 - Query #2 Range: Global Surface Level (0-10m)
- **Performance:**



Beacon 🤝 Zarr Performance ⚡

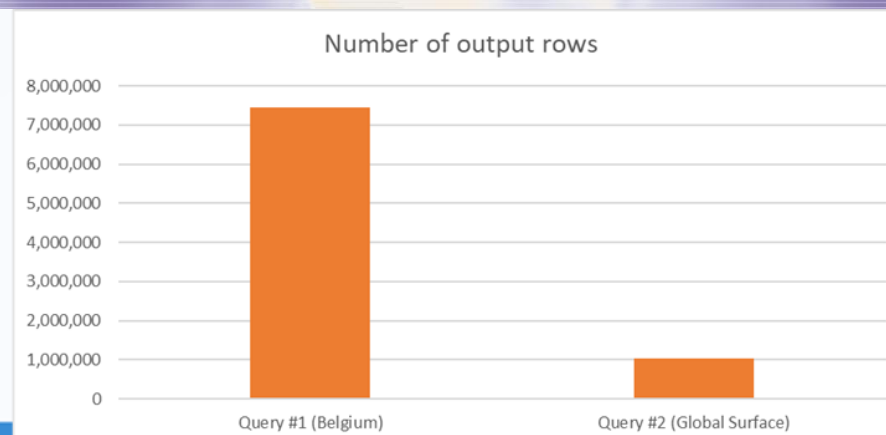
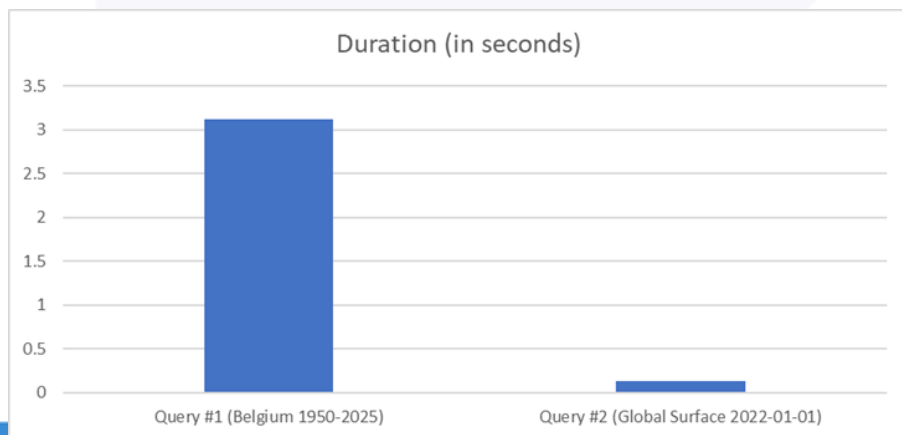
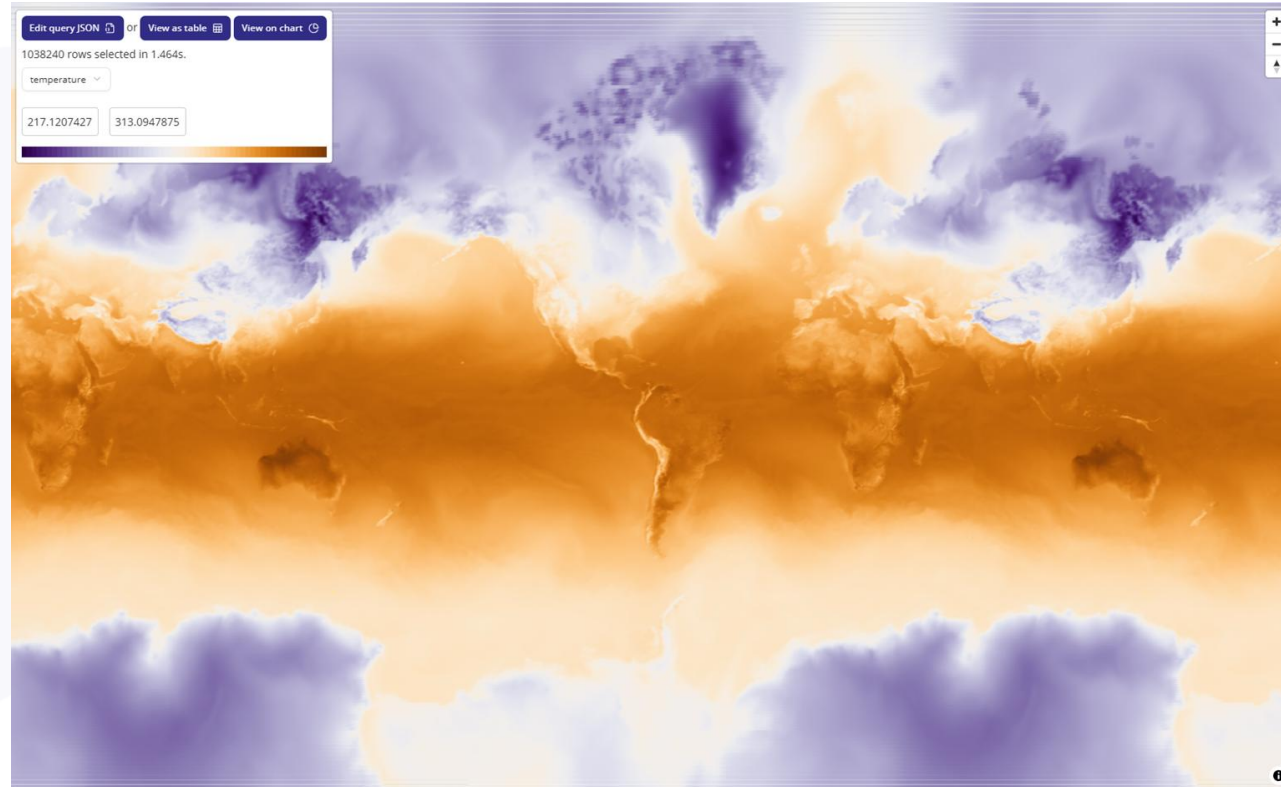
- **ERA5 Daily Reanalysis**
 - Stored as ZARR files inside a Cloud Bucket
 - Multiple TB's (Global, 1950 - 2025)
- **Benchmark Queries**
 - Selected Columns:
 - Temperature
 - Longitude
 - Latitude
 - Time
 - Query #1 Range Belgium, 1950 - 2025
 - Query #2 Range Global for a Single Day
- **Performance:**

Edit query JSON or View as table or View on chart

1038240 rows selected in 1.464s.

temperature

217.1207427 313.0947875



Beacon Blue Cloud instances/nodes

- Euro-Argo data
 - *retrieved from S3 bucket*
- CORA Profile data
 - *retrieved from CMEMS, product: INSITU GLO PHY TS DISCRETE MY 013 001*
- CORA Timeseries data
 - *retrieved from CMEMS, product: INSITU GLO PHY TS DISCRETE MY 013 001*
- EMODnet Chemistry data
 - *retrieved from EMODnet Chemistry WebODV (Eutrophication Atlantic profiles 2024 unrestricted.odv)*
- WOD data
 - *retrieved from ncei.noaa.gov*
- SeaDataNet CDI TS data
 - *retrieved from EGI-ACE webODV*
- SeaDataNet CDI Incremental
- CMEMS BGC data
 - *retrieved from CMEMS, product: INSITU GLO BGC DISCRETE MY 013 046*

Demo

wod-north-sea-trendline.ipynb

0 (latest) > wod-north-sea-trendline.ipynb > M+ Install the beacon_api package to interact with the Beacon Data Lake API > from beacon_api import * # Import the Beacon API client

Generate + Code + Markdown | Run All Restart Clear All Outputs | View data Jupyter Variables Outlinevenv (Python 3.13.2)

Install the beacon_api package to interact with the Beacon Data Lake API

You can find the package on PyPI: <https://pypi.org/project/beacon-api/>

If you run into any issues, please refer to the GitHub repository: <https://github.com/maris-development/beacon>

Documentation for the beacon_api package can be found here: <https://maris-development.github.io/beacon/docs/1.2.0/query-docs/getting-started.html#python>

Documentation for the Beacon Data Lake technology can be found here: <https://maris-development.github.io/beacon/>

```
1 from beacon_api import * # Import the Beacon API client
2 import os
```

Wildcard import from a library not allowed

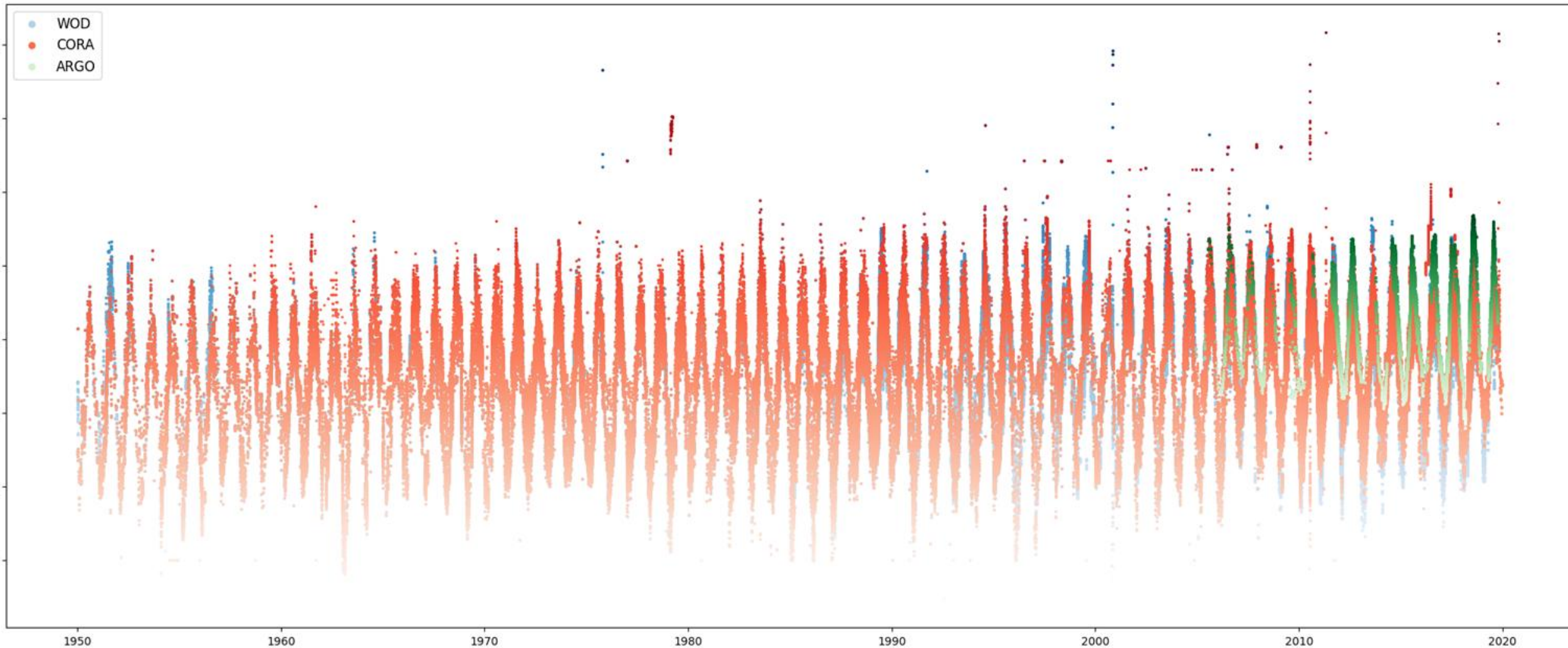
MagicPython

In order to get access to the Beacon endpoint, you will need a token.

The notebook will fetch the Token for you when running in the D4science VRE. If you are running the notebook outside the D4Science VRE you will need to get the token from the D4science VRE and fill it in manually.

```
1 TOKEN = os.getenv('D4SCIENCE_TOKEN') # This will fetch the token from the VRE environment.
2 # If you want to run this notebook locally, you can set the TOKEN variable directly here:
3 #TOKEN = "<your_token>" # Get your token from https://blue-cloud.d4science.org/group/blue-cloudlab/authorization. Simply press "Get Token" and copy the token here.
4 BEACON_INSTANCE_URL = "https://beacon-wod.maris.nl" # Can optionally be replaced with the Beacon running on D4Science: https://
```

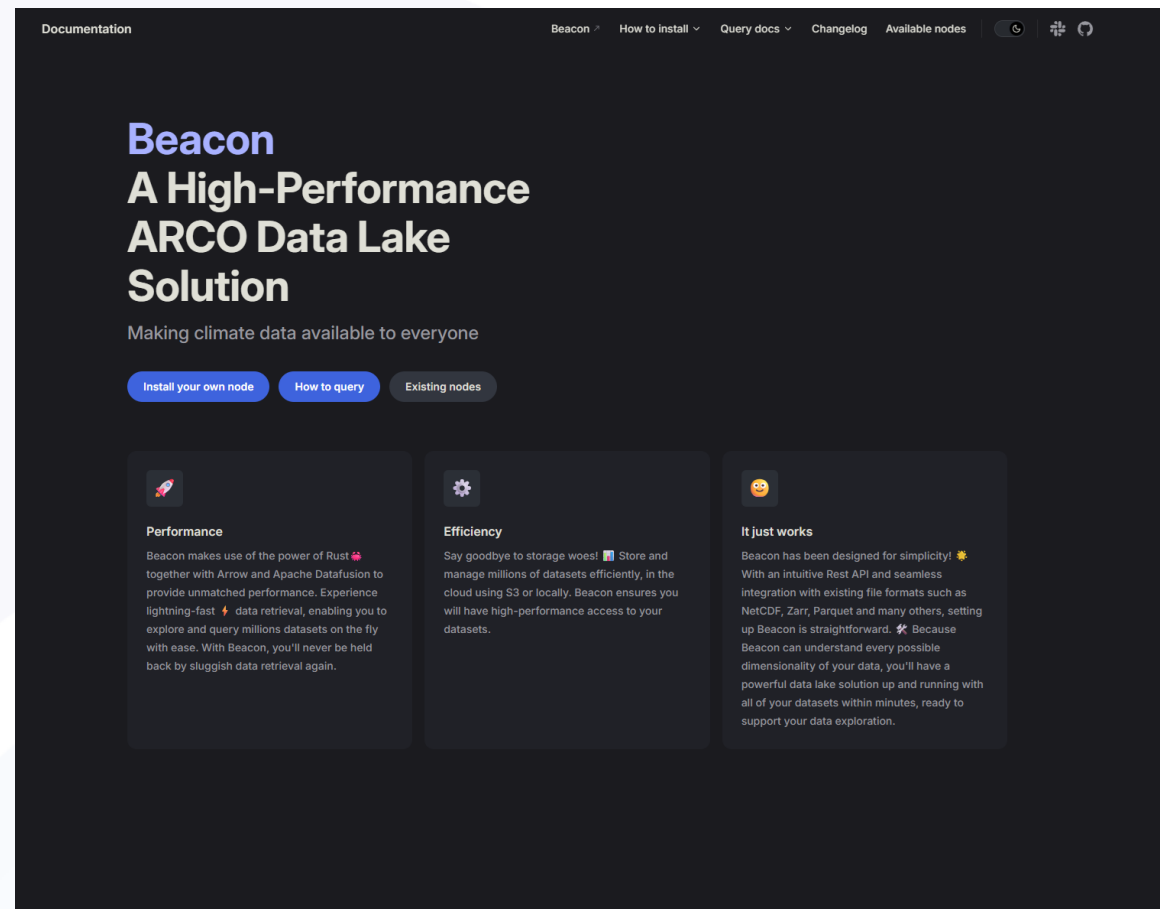
Example: 1950 - 2020 North Sea Temperature



Available Open Source



- Beacon available at :
 - Docs: <https://maris-development.github.io/beacon/>
 -  GitHub Repository: <https://github.com/maris-development/beacon>
- Example setup:
 - A guide for setting up a Beacon instance
 - <https://github.com/maris-development/beacon-example>
- Set-up multiple Beacon instances
 - With example notebooks



Beacon Studio

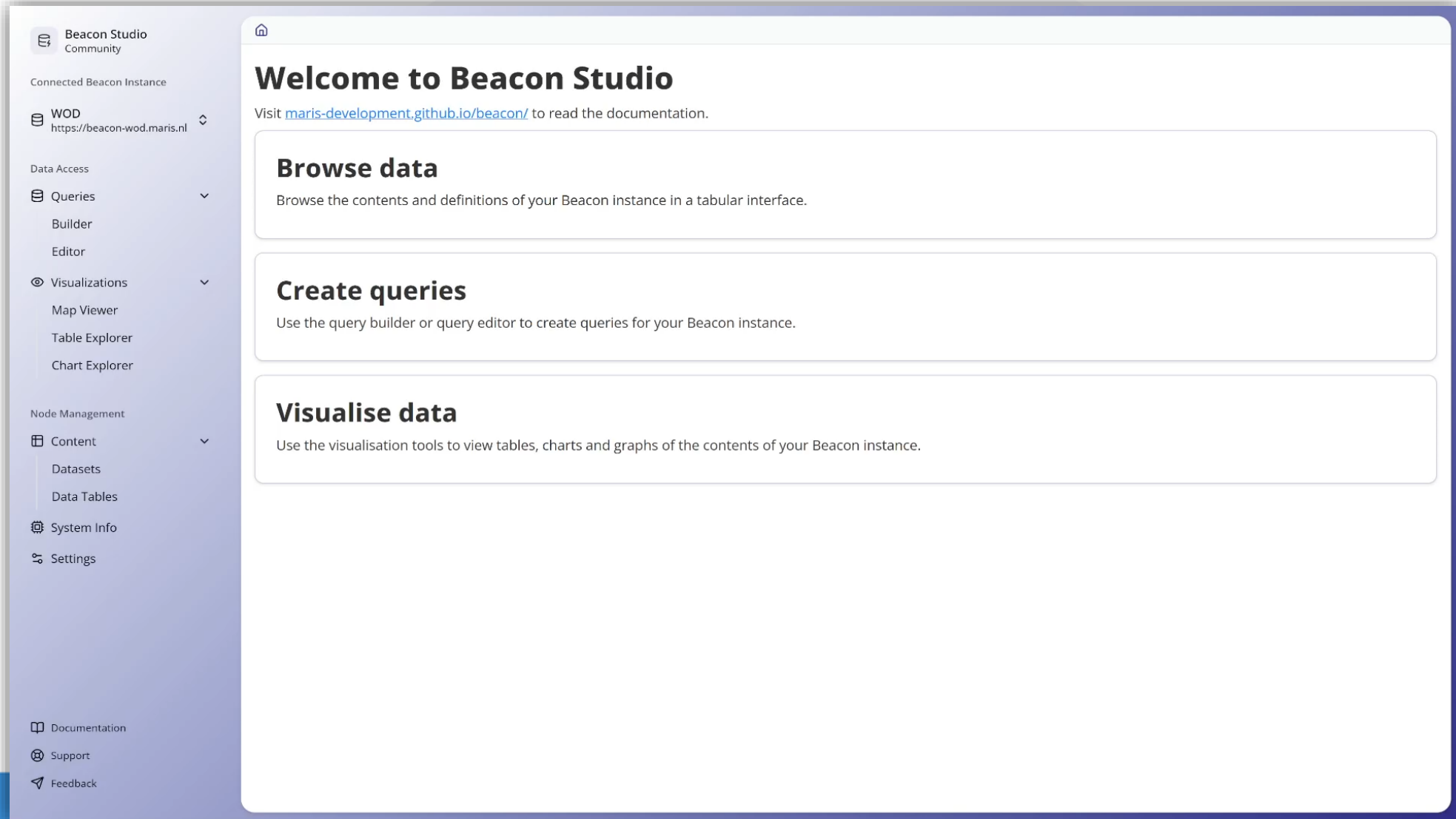
A graphical user interface on top of the Beacon

What is Beacon Studio?

- Graphical User Interface on top of Beacon
- All the functionality of the Beacon API, but made easier



Browse the Beacon contents



The screenshot displays the Beacon Studio Community web application. On the left is a dark blue sidebar with a navigation menu. The main content area on the right is white and features a 'Welcome to Beacon Studio' header, a link to documentation, and three large white boxes with rounded corners containing instructions on how to browse data, create queries, and visualise data.

Beacon Studio Community

Connected Beacon Instance

- WOD <https://beacon-wod.maris.nl>

Data Access

- Queries
 - Builder
 - Editor
- Visualizations
 - Map Viewer
 - Table Explorer
 - Chart Explorer

Node Management

- Content
 - Datasets
 - Data Tables
- System Info
- Settings

Documentation

Support

Feedback

Welcome to Beacon Studio

Visit maris-development.github.io/beacon/ to read the documentation.

Browse data

Browse the contents and definitions of your Beacon instance in a tabular interface.

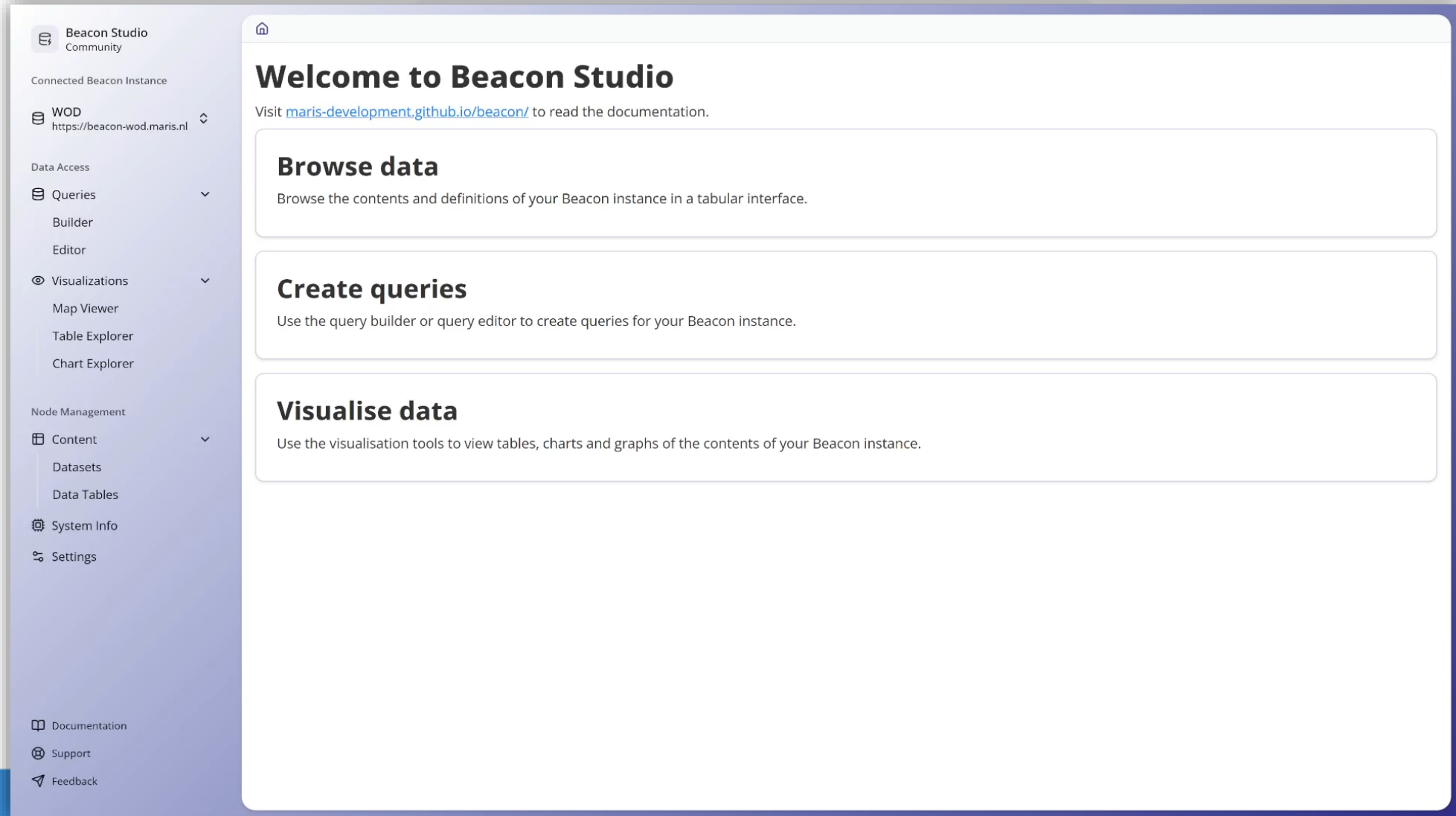
Create queries

Use the query builder or query editor to create queries for your Beacon instance.

Visualise data

Use the visualisation tools to view tables, charts and graphs of the contents of your Beacon instance.

Build a Beacon query (Easy Tables)



View your query results

The screenshot displays the Beacon Studio Community interface. On the left is a sidebar with navigation options: Beacon Studio Community, Connected Beacon Instance (WOD), Data Access (Queries, Builder, Editor), Visualizations (Map Viewer, Table Explorer, Chart Explorer), Node Management (Content, Datasets, Data Tables), System Info, and Settings. At the bottom of the sidebar are links for Documentation, Support, and Feedback.

The main content area is titled 'Table explorer' and shows the breadcrumb 'Visualisations > Table explorer'. It includes three buttons: 'Edit query JSON', 'View on chart', and 'View on map'. Below these, it states '0 rows selected in 0.000s.' and shows a 'Loading data...' spinner. A pagination bar at the bottom indicates 'Page 1 of 0' with 'Previous' and 'Next' buttons. A mouse cursor is visible over the main content area.

View your query results on the map

Beacon Studio Community

Connected Beacon Instance

WOD
https://beacon-wod.maris.nl

Data Access

Queries

Builder

Editor

Visualizations

Map Viewer

Table Explorer

Chart Explorer

Node Management

Content

Datasets

Data Tables

System Info

Settings

Documentation

Support

Feedback

To: 10

☒ **temperature**

Description: Temperature value in degrees kelvin

From: -5

To: 45

☒ **salinity**

Description: Salinity value in practical salinity units

☒ **oxygen**

Description: Oxygen value

Select metadata columns

☒ Include all metadata columns with your query.

Toggle columns display

Select output format

Output format

parquet

Execute query

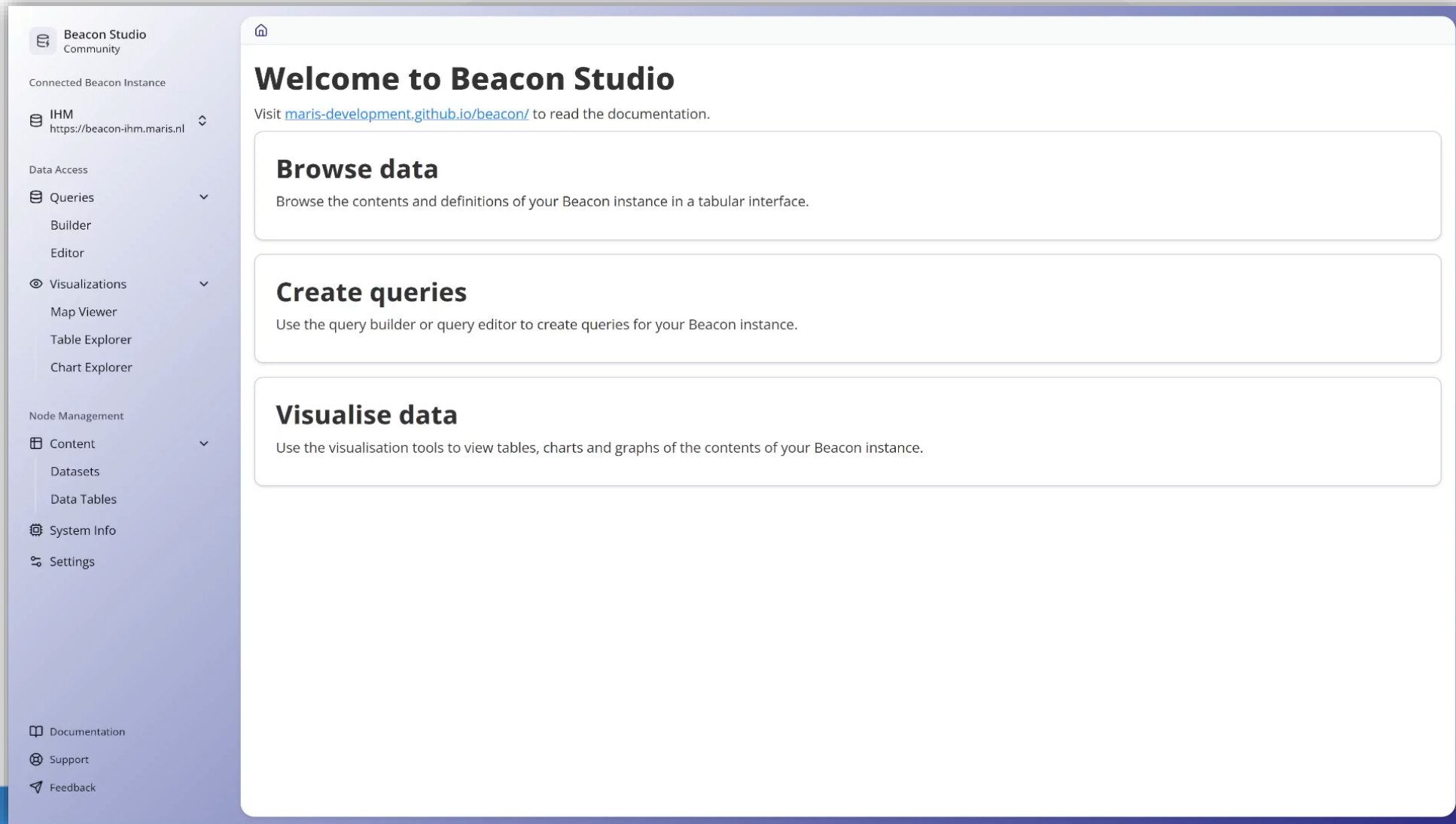
Copy query JSON

View as table

View on map

View on chart

View your query results using charts



**And we've plenty more features
planned for Beacon Studio!**

More about Beacon (& Beacon Studio) at:

- <https://maris-development.github.io/beacon/>
 - <https://beacon.maris.nl/>
- <https://github.com/maris-development/beacon> 

Want to know more?

contact us at:

- peter@maris.nl
- robin@maris.nl
- paul@maris.nl