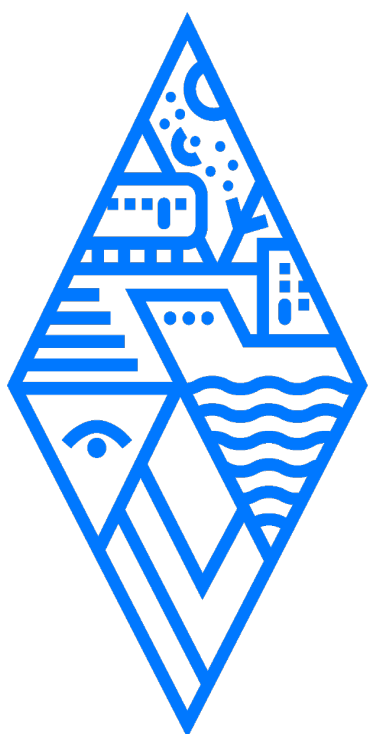




POLARIN

POLAR
RESEARCH
INFRASTRUCTURE
NETWORK

**Deliverable 4.8. Dockerised open-source software
components that cover architecture implementation
requirements**

V2, November 2025

www.eu-polarin.eu



**Funded by
the European Union**

POLARIN: POLAR RESEARCH INFRASTRUCTURE NETWORK

Funding programme: Horizon Europe

Grant Agreement No.: 101130949

Project Start Date: 01/03/2024

Duration: 60 months

Coordinator: Alfred Wegener Institute, Germany

Document information	
Work Package	WP4 Improvement of data services and customised data products
Deliverable No	D4.8
Deliverable title	Dockerised open-source software components that cover architecture implementation requirements.
Version	V1
Dissemination level	☒ PU - Public ☐ PP - Restricted to programme partners ☐ RE - Restricted to a group specified by the consortium ☐ CO - Confidential, only for members of the consortium
Lead Beneficiary	CNR
Lead author	Giulio Verazzo (ISP-CNR)
Contributors	ETT, SIOS, ISP-CNR
Contributing authors	Alice Cavaliere (ISP-CNR), Rachele Bordoni (ETT), Antonio Novellino (ETT)
Due date	31/8/2025
Delivery date	30.08.2024

Document history	
Creation Date	08/08/2025
Version	V1
Version Date	28/10/2025
Status	☐ Draft ☒ WP lead approved ☒ Coordinator approved ☒ Executive Board approved
Status date	30.10.2025

TABLE OF CONTENTS

1. SUMMARY	4
2. Components of the POLARIN Data Hub system architecture	5
2.1 DOCKER	6
2.2 ERDDAP-CMS	7
2.3 GeoNetwork	9
2.3.1 Docker configuration file	9
2.4 ERDDAP	10
2.4.1 Docker configuration file	10
2.5 METADATA Worker	13
2.5.1 Docker configuration file	13
3. References	14

1. SUMMARY

POLARIN's overall aim is to provide efficient and customised research infrastructure (RI) services to address the scientific challenges of the polar regions. This includes offering access to a wide portfolio of complementary and interdisciplinary top level RIs. This also include the access data collected at POLARIN RI that offer Virtual Access (VA).

By integrating polar RI capacities, POLARIN facilitates scientific research to understand and predict key processes in polar regions in the context of climate changes. This effort enhances society's problem-solving capacity and supports evidence-based policy making.

One of POLARIN's four core services is improving access to data and developing online services and data products (WP4), another POLARIN core service is the Virtual Access (VA) to RIs and data collected during the POLARIN project (WP5).

To facilitate the discoverability of the resources, POLARIN is developing a data portal, i.e. the POLARIN Data Hub (DH). On top of this, POLARIN will provide a monitoring framework for data collected at POLARIN RI that offer Virtual Access (VA). The POLARIN DH is developed on top of some open software tools (ERDDAP, GeoServer, GeoNetwork) to catalogue and interoperate with the backend RIs infrastructure service.

This deliverable describes this part of the POLARIN architecture, how these tools are configures in their current state and how they can be deployed by using the Docker technology.

This document does not cover the VA monitoring tools that will be described under WP5 deliverables.

2. Components of the POLARIN Data Hub system architecture

The main goal of the POLARIN Data Hub (DH) is to offer a single easy access point to discover and access the ever-growing heterogeneous collections of (polar) data and products (including observations from satellites, airplanes, drones, in-situ sensors, etc.) for both the North and South Poles. This, obviously, applies at first to the data collected at POLARIN RIs that provide Virtual Access (VA).

The POLARIN DH is central to the overall POLARIN data objectives and represents a key supporting tool for many of the POLARIN activities. As stated in the project's Grant Agreement (Amendment No. 22 to Grant Agreement No. 101130949), the POLARIN DH "establishes a hub where experts as well as non-experts/intermediaries are guided ([..]) to find data and information necessary to match their specific needs" and "develop tools and services for intermediaries and end-users (both experts and non-experts)".

Because of the nature of the federated approach of the DH, the data repositories connected to the Data Hub (sometimes indirectly or through consecutive federation steps) manage the lifecycle of the data. However, wherever this approach is not possible, the data can be cached in the back-end of the DH.

The POLARIN DH backend services and tools include data catalogues, utilities for loading and visualizing data in time and space, tools for processing and comparing results, and new AI-based features that support data navigation and processing. A crucial component of this workflow is the data collected and made available through the VA. As a complementary effort, the POLARIN DH is also developing a framework to consistently monitor access to data via the VA and VA data services.

Currently the tools adopted to implement this DH are:

- GeoNetwork-based central metadata catalogue (<https://geonetwork.s4polarin.eu/geonetwork>)
- POLARIN Content Management System (CMS; Environmental Research Division's Data Access Program (ERDDAP)-based data server; <https://erddap.s4polarin.eu/erddap>)
- Metadata Worker
- POLARIN DH (<https://s4polarin.eu/>)

Through these components, POLARIN ensures that users can easily find and access data using the available discovery metadata. These tools streamline the compilation of data and metadata towards the central system. Metadata can be added to GeoNetwork either through semi-automated submissions (for example, via Transnational Access activities) or through direct machine-to-machine connections from data distributors using standard metadata exchange protocols such as the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) and the Open Geospatial Consortium Catalog Service for the Web (OGC CSW).

Users can interact with the catalogue at the discovery level through these protocols, and directly access data cached via the POLARIN ERDDAP data server. To populate this cache, POLARIN developed and deployed an ERDDAP metadata checker and metadata content manager system (ERDDAP-CMS).

This CMS streamlines the management of datasets in ERDDAP, checks that the (meta)data comply with the requirements stated in the minimum metadata profile of the POLARIN data management plan (D4.1), and makes the data accessible both through a web interface and via data streaming protocols such as OPeNDAP.

Additional contextual information—such as the RI the data were collected at, the component of the climate sphere the data represent, and which scientific challenge can be addressed with the data - is added to the harvested (meta)data using the Metadata Worker. The enriched records and their associated datasets are then published through the POLARIN DH (<https://s4polarin.eu>) a user-centric platform for searching, visualizing, and accessing data.

As described in the GA, these tools are tailored to support the POLARIN DH and in line with the Open Science policy of the project. These tools are made open-source and are available to POLARIN stakeholders and community on the POLARIN GitHub repository (<https://github.com/POLAR-RESEARCH-INFRASTRUCTURE-NETWORK>). Moreover, to facilitate the installation, these tools have been packed in containers (Docker): ready-to-deploy to enable the easy installation of the POLARIN DH at any level.

2.1 DOCKER

All software components of the POLARIN system architecture — including the ERDDAP-CMS, ERDDAP, GeoNetwork, Metadata Worker, and DH — are deployed and orchestrated using Docker. The corresponding containers are published on the POLARIN GitHub repository and provide the specific configurations and customizations used within the project.

Docker is an open-source platform that enables the development, distribution, and execution of applications. It relies on operating system–level virtualization to encapsulate software within units known as *containers*. Each container operates independently of the host system (Ubuntu OS in this case) and integrates all required libraries, dependencies, and configurations.

This approach circumvents the need for complex installation procedures where software components are built and rely on third party software. It mitigates some of the risks associated with incompatibilities between software versions or libraries.

The configuration of containers is defined through a *Dockerfile*, which provides a structured set of instructions for building a container from a Docker image. Docker images constitute blueprints of execution environments. Based on these images, Docker instantiates isolated *containers* that serve as self-contained environments for running the applications they represent.

```
Dockerfile > ...
1  FROM python:3.11.2 AS base (last pushed 2 years ago)
2
3  COPY api_geonetwork /api_geonetwork
4  RUN pip install -r /api_geonetwork/requirements.txt
5
6  WORKDIR /api_geonetwork
7
8  FROM base AS prod
9
10 CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "80", "--reload"]
```

Figure 1. An example of a Dockerfile for building a container hosting a Python application

Software components running in Docker containers can communicate with one another through Docker Compose, a tool that facilitates the definition and management of multi-container applications. The configuration and parameters of each container are specified within a docker-compose.yml file, which provides a declarative description of the services, networks, and volumes required by the system. Applications encapsulated in containers are subsequently instantiated and executed by issuing the docker compose up command within a terminal environment. This procedure ensures the coordinated startup of all defined services, thereby enabling seamless interaction among the containers.

2.2 ERDDAP-CMS

ERDDAP is a powerful data server that provides a simple and consistent way to access, subset, and download gridded and tabular scientific datasets in a variety of common file formats, as well as generate graphs and maps. To enhance and simplify the management of ERDDAP servers, the ERDDAP-CMS web application has been developed. This tool streamlines dataset administration by allowing users to upload new datasets from CSV, NetCDF, or other ERDDAP instances; edit ACDD-compliant metadata through a complete interface with CF-convention standard name support; and validate and publish datasets with a single click. It also supports authentication via ORCID, GitHub, and OAuth, enables user permission management, and offers ISO 19139 (gmd prefix) metadata export fully compatible with GeoNetwork Opensource.

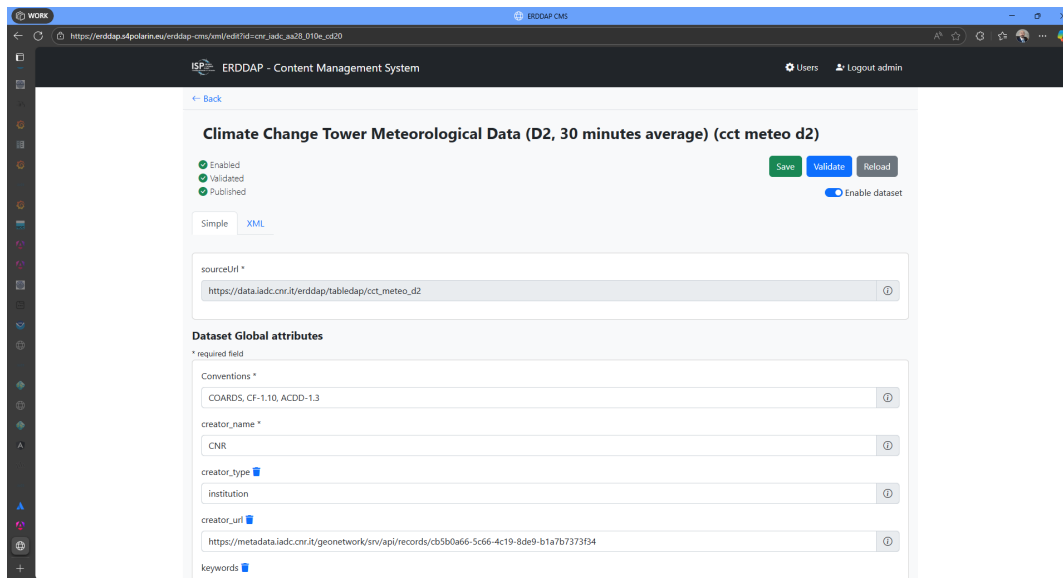
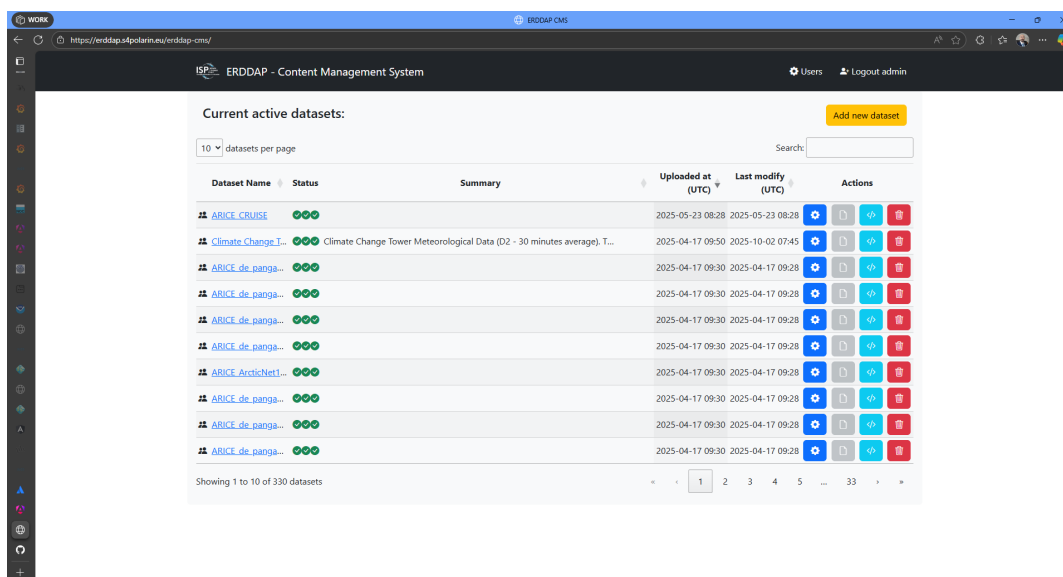


Figure 2. Metadata editor

ERDDAP-CMS is built with Python Flask and can be launched using Docker Compose. By running docker compose up -d, ERDDAP becomes available at <http://server-host:8080/erddap>, while ERDDAP-CMS can be accessed at <http://server-host:5000/erddap-cms>. The default CMS administrator credentials are username: admin and password: admin. Dataset configuration files and data are stored outside of version control: data reside within the Docker volume datasets_data, and configuration files are saved in the datasets_xml folder.



Dataset Name	Status	Summary	Uploaded at (UTC)	Last modify (UTC)	Actions
ARICE_CRIUSE	Validated		2025-05-23 08:28	2025-05-23 08:28	[Edit] [Delete] [Refresh] [Add]
Climate Change J...	Validated	Climate Change Tower Meteorological Data (D2 - 30 minutes average). T...	2025-04-17 09:50	2025-10-02 07:45	[Edit] [Delete] [Refresh] [Add]
ARICE de paoga...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]
ARICE de paoga...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]
ARICE de paoga...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]
ARICE de paoga...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]
ARICE ArcticNet...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]
ARICE de paoga...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]
ARICE de paoga...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]
ARICE de paoga...	Validated		2025-04-17 09:30	2025-04-17 09:28	[Edit] [Delete] [Refresh] [Add]

Figure 3. Metadata editor – validation workflow

2.3 GeoNetwork

GeoNetwork is a Free and Open-Source Software (FOSS) cataloguing application for spatially referenced resources. GeoNetwork provides a web interface to search geospatial data across multiple catalogues. The search provides full-text search as well as faceted search on keywords, resource types, organisations, scale, etc. The catalogue is able to describe geospatial layers, services, maps, and non-geographic datasets. GeoNetwork implements WxS, OGC, ISO 19115/119/110 standards used for spatial resources, and the Dublin Core format usually used for open data portals.

The POLARIN architecture is running a GeoNetwork 3.12.11.

2.3.1 Docker configuration file

```
1  services:
2    # GeoNetwork
3    geonetwork:
4      image: geonetwork:3.12-postgres
5      container_name: "polarin-geonetwork"
6      ports:
7        - "12083:8080"
8      env_file:
9        - "geonetwork-compose.env"
10     volumes:
11       # GeoNetwork data volume - custom configuration, uploaded files, ecc...
12       - type: bind
13         source: /geonetworkData
14         target: /geonetworkData
15       # Host data directory accessible from GeoServer
16       - type: bind
17         source: /opt/extdata/polarin/geonetwork
18         target: /geonetworkUserData
19       # change default language geonetwork (config.properties)
20       - type: bind
21         source: /config.properties
22         target: /usr/local/tomcat/webapps/geonetwork/WEB-INF/config.properties
23     restart: unless-stopped
24
```

Figure 4. An overview of the Docker-compose configuration file of the GeoNetwork software

The GeoNetwork container is built from the [official GeoNetwork Docker image](#) publicly available to DockerHub, the Docker image repository. The HTTP port 8080 exposed by the container is mapped to port 12083 on the host server and an external reverse proxy makes the service available at:

<https://geonetwork.s4polarin.eu/geonetwork/>

Environmental variables are stored into geonetwork-compose.env file and passed to the container while two host folders are bind-mounted inside the container to allow storage persistence across

container restarts. Lastly, a modified version of config.properties file is mounted inside the container to set English as the default language.

In the context of POLARIN, GeoNetwork is used as a collector of metadata from the other catalogues of the POLARIN partners. Collection is realized by leveraging the GeoNetwork ability to harvest metadata using several different protocols. The GeoNetwork central metadata catalogue uses the OAI-PMH and OGC CSW v2.0.2 exchange protocols for discovery metadata

2.4 ERDDAP

ERDDAP data server is an OFFS written in Java that was built using the open-source codebase of the Web Coverage Service (WCS), Sensor Observation Service (SOS), Ocean Biodiversity Information System (OBIS), and OPeNDAP. ERDDAP supports both human interaction and machine-to-machine (M2M) interoperability. The ERDDAP data server supports several common data file formats (HTML table, netCDF, CSV, TXT, MAT, JSON, etc.) and output files are created on-the fly in any of these formats. ERDDAP implements Federal Geographic Data Committee Web Accessible Folder (FGDC WAF) with FGDC-STD-001-1998 and ISO 19115 WAF with ISO 19115-2/19139.

The POLARIN runs a ERDDAP v2.24 with the ERDDAP-CMS (ISP-CNR) on top of it.

2.4.1 Docker configuration file

The POLARIN ERDDAP docker image includes the ERDDAP-CMS from ISP-CNR. This modified image adds a custom web-app on top of the original ERDDAP image, which aims to simplify the usage and management of ERDDAP data server. Its main features are:

- Upload of local datasets from CSV or netCDF
- Linking datasets from other ERDDAP instances
- Complete ACDD-based metadata editor
- Metadata keywords from CF-convention standard names table
- One-click dataset validation and publication
- ISO 19139 (gmd prefix) metadata export fully compatible with GeoNetwork
- Management of user permissions on dataset

The docker-compose configuration file deploys two services, a PostgreSQL database needed to store POLARIN datasets, data of ERDDAP-CMS users, and datasets and the ERDDAP-CMS service. Docker managed-volumes and bind-mounted volumes are used to persist data across container restarts. Environmental variables are stored in the 2-no-multiauth.env file and ports 8080 (ERDDAP HTTP), 8443 (ERDDAP HTTPS), and 5000 (CMS HTTPS) are mapped to external host ports and linked via a reverse proxy to the following addresses:

- <https://erddap.s4polarin.eu/erddap/index.html> (ERDDAP HTTP)
- <https://erddap.s4polarin.eu/erddap/index.html> (ERDDAP HTTPS)
- <https://erddap.s4polarin.eu/erddap-cms> (CMS)

```

1  services:
    ▷ Run Service
2  db:
3      image: postgres:latest
4      container_name: isp-erddap-db
5      environment:
6          POSTGRES_USER: erddap
7          POSTGRES_PASSWORD:
8      healthcheck:
9          test: pg_isready -d erddap -U erddap
10         interval: 30s
11         timeout: 60s
12         retries: 5
13         start_period: 80s
14     volumes:
15         - pg_data:/var/lib/postgresql/data:rw
16     restart: always
17
18     ▷ Run Service
19     erddap-cms:
20         image: bo.isp.cnr.it/erddap-cms-no-multiauth:1.3
21         container_name: erddap-cms
22         ports:
23             - 12086:8080 #erddap
24             - 12090:8443 #erddap https
25             - 12087:5000 #cms
26         restart: always
27         platform: linux/amd64
28         env_file: 2-no-multiauth.env
29         depends_on:
30             db:
31                 condition: service_healthy
32         volumes:
33             - type: bind
34               source: erddap-self.localhost_2122.pfx
35               target: erddap-self.localhost_2122.pfx
36             - type: bind
37               source: server.sslconnetor.xml
38               target: server.sslconnetor.xml
39             - datasets_xml_parts/active:/datasets_xml_parts/active:rw
40             - erddap_data:/erddapData:rw
41             - HCLIM43_ALADIN_2000-2020.png:/logo.png:rw #logo for erddap
42             - datasets_data:/datasets_data:rw
43

```

Figure 4. An overview of the Docker-compose configuration file of the ERDDAP-CMS software

In POLARIN, ERDDAP is used as a data server. Datasets can be stored on the local data server or can be federated from partner servers: in this case, ERDDAP acts as a bridge between the source and the user.

ERDDAP > List of All Datasets

320 matching datasets, listed in alphabetical order.

Grid DAP Data	Sub-set	Table DAP Data	Make A Graph	W M S	Source Data Files	Title	Summary	FGDC, ISO, Metadata	Background Info
	set	data	graph			* The List of All Active Datasets in this ERDDAP *	?	M	background
	set	data	graph			Aerosol scattering and absorption coefficients at the Gruebadet Aerosol Laboratory (Svalbard)	?	F I M	background
	set	data	graph			Aerosol scattering and absorption coefficients at the Gruebadet Aerosol Laboratory (Svalbard), from 2022	?	F I M	background
	set	data	graph			AIRQino artic stations	?	F I M	background
	set	data	graph		files	ArcticNet 0501a, Northern Baffin Bay CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0501b, Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0502a, Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0502b, Hudson Bay CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0602a, Labrador CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0602b, Northern Baffin Bay CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0602c, Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0603a, Beaufort Sea CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0603b, Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0603c, Hudson Bay and Foxe Basin CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0603d, Labrador Sea and fjords CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0704a, Labrador fjords and sea CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0704b, Hudson Bay CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0806, Baffin Bay CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 0904a, Canadian Arctic CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet 0904b, Labrador Fjords CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet 1001a, Hudson Bay CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet 1001b, Northwest Passage CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet 1003a, Beaufort Sea CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet 1003b, Northwest Passage CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet 1003c, Labrador Fjords CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet 1101, Baffin Bay and Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 1103a, Beaufort Sea CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 1103b, Baffin Bay and Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 1304, Baffin Bay and Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet 1305, Northwest Passage CTD data	?	F I M	background
	set	data	graph		files	ArcticNet, Arctic SOLAS 0706a, Northern Baffin Bay CTD data	?	F I M	background
	set	data	graph		files	ArcticNet/BP 1002a, Beaufort Sea CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet/BP 1002b, Beaufort Sea CTD Data	?	F I M	background
	set	data	graph		files	ArcticNet/Geotraces 0903a, Beaufort Sea CTD Data	?	F I M	background

Figure 5. A screenshot of the ERDDAP index.html webpage

As per GeoNetwork, datasets from POLARIN partners already available on other data storage systems are retrieved (if possible) using M2M protocols such as OPeNDAP, preserving original metadata and data ownership.

2.5 METADATA Worker

The **Metadata worker** consists of a Python RESTful API and facilitate the harmonization of metadata, maps which resource is offering data and which are listing other type of resources and to support the presentation into the POLARIN DH tools. More specifically this component. This mapping is managed by an internal JSON file.

2.5.1 Docker configuration file

```
1  services:
2      db:
3          image: mongodb/mongodb-community-server:6.0-ubi8
4          container_name: polarin-api-db
5          restart: always
6          environment:
7              - MONGO_INITDB_ROOT_USERNAME=polarin
8              - MONGO_INITDB_ROOT_PASSWORD=
9          ports:
10             - 10.0.0.1:27017:27017
11          volumes:
12             - db_data:/data/db
13      api:
14          build:
15              context: ./
16              target: dev
17          container_name: polarin-api
18          env_file: 0-api.env
19          ports:
20             - 10.0.0.1:8000:80
21             - 10.0.0.1:5678:5678 #debugpy port
22          restart: always
23          volumes:
24             - ./api_geonetwork:/api_geonetwork
25
26  volumes:
27      db_data:
28      pg_data:
```

Figure 6. The Docker-compose configuration file of the Metadata Worker

The metadata worker is composed of two services, a MongoDB database and a Python application. The database is used to store collected metadata in JSON format. The Python application is built using a Dockerfile starting from the official Python 3.11.2 image. The software is developed using the Python FastAPI library. Environmental variables are stored in the 0-api.env file, and ports 80 (HTTP) and 5678 (DEBUGPY) are mapped to external host ports. Python source code files are mounted via bind-mount inside the container.

Once installed and configured, the API exposes the following endpoints:

Method	URL	Description
GET	/download	Download metadata XML to the server
GET	/load	Load datasets from GeoNetwork to the db
POST	/datasets	Fetch paginated datasets
GET	/datasets/{id}	Fetch a single dataset
GET	/test/{uuid}	Test connection to db
GET	/ris	Get the list of all Research Infrastructures
GET	/distributors	Get the list of all distributors
GET	/owners	Get the list of all owners
GET	/variables	Get the list of all variables in datasets

3. References

- Deliverable D4.1 – POLARIN DMP
- POLARIN GitHub repository: <https://github.com/POLAR-RESEARCH-INFRASTRUCTURE-NETWORK>
- Erddap.com
- Geonetwork.com
- Docker.com