D2.2 Data Tools

Project Number	
Classification	Public
Deliverable No.	D2.2
Work Package(s)	WP2
Document Version	1.0
Issue Date	June 30, 2019
Document Timescale	Project Start: June/July 1st (Poland), 2018
Final version due	2019-06-30
Compiled by	PUEB
Authors	Dominik Filipiak, Milena Stróżyna, Krzysztof Węcel, Witold Abramowicz
Issue Authorisation	Sebastian Feuerstack, OFF

HANSA is funded by the MarTERA partners German Federal Ministry of Economic Affairs and Energy (BMWi), Polish National Centre for Research and Development (NCBR) and Research Council of Norway (RCN) and co-funded by the European Union.

DISTRIBUTION LIST

Copy type[1]	Company and Location	Recipient
T	HANSA Consortium	all HANSA Partners

RECORD OF REVISION

Date	Status Description	Author
2019-06-19	First draft	PUEB

1 Tools

This document presents the code of the methods and tools developed in the HANSA project related to the data retrieval, pre-processing and fusion. The code itself is available in the separate file. The code is organised in two folders, each represented by one of the following subsections.

1.1 Apache Spark Docker Image

The folder Docker-SparkJupyter contains a docker image used as an Apache Spark container within the project. It is used mainly in the process of the development of the recommended corridors and mesh generation. It contains two files: Dockerfile and docker-compose.yml.

Dockerfile The first file describes the docker image – it is based on the jupyter/pyspark-notebook image.

docker-compose.yml The second file describes the service. In this file, high-level features are set, such as Apache Spark options, external libraries, or available ports.

1.2 CMEMS Downloader

This folder contains all the files and services necessary to handle the COPERNICUS weather data.

Dockerfile The docker image for wrapping the downloader.

Dockerfile-pypy The docker image for wrapping the downloader (with PyPy insetad of Python).

build_docker.sh A script for building the docker image from Dockerfile (Python version).

build_docker_pypy.sh A script for building the docker image from Dockerfile PyPy version).

run_docker.sh A shell script for running the process of the data download (Python).

run_docker_pypy.sh A shell script for running the process of the data download (PyPy).

run_docker_test.sh A shell script for running the process of the data download (for test purposes).

requirements.txt Python requirements needed to build the docker image.

avro-tools-1.8.2.jar External avro library, needed for schema generation.

avro_schemas Contains Avro schemas for *ice_surface*, *sea_level*, and *wind* data. To generate avro schemas (.avsc) from avro protocol (.avdl) execute following command

```
java -r avro-tools-1.8.2.jar idl2schemata avro_schemas/ice_surface/
   IceSurfaceProtocol.avdl
```

Serialisation is done after successful download of an .nc file. Avro serialiser for particular CMEMS products are located in *avro_serializers.py*

avro_serializers.py Serialisers for all the handled data types.

cmems.py Contains the code for handling the download process.

do_the_thing.py This main script, responsible for downloading .nc files for CMEMS products from CMEMS ftp and serialising particular variables from this files with Avro. It works in two steps:

1. traverses ftp directory tree, stores ftp paths to .nc files in local database (sqlite) and flags them as *not_synced*
2. download *not'_synced* files from local database and flags them *synced* after successful download

One need to create *credentials* file with ftp authorisation data in form

```
{"login": "your_cmems_login", "password": "your_cmems_pass"}
```

To run this script alone, execute *python do_the_thing.py*.

models.py Database initialiser.

nc2avro_performance_test.py Performance tests for serialisers.