

interTwin

D7.1 Report on requirements and thematic modules definition for the environment domain

Status: FINAL

Dissemination Level: public



Funded by the
European Union

Disclaimer: Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them

Abstract

Key Words

Earth Observation, Digital Twin, Environment, Satellite


interTwin Work Package 7 will provide a set of reusable components, called thematic modules, for the digital twins defined in WP4. In this document we provide a high-level description of the environmental domain use cases and of the designed thematic modules. We also provide a set of common requirements derived from the analysis of the use cases and thematic modules that will need to be available in the Digital Twin Engine (DTE) core, designed in WP6.



Document Description

D7.1 Report on requirements and thematic modules definition for the environment domain

Work Package number WP7

Document type	Deliverable		
Document status	FINAL	Version	1
Dissemination Level	Public		
Copyright Status	 <p>This material by Parties of the interTwin Consortium is licensed under a Creative Commons Attribution 4.0 International License.</p>		
Lead Partner	EURAC		
Document link	https://documents.egi.eu/document/3955		
DOI	https://doi.org/10.5281/zenodo.8036991		
Author(s)	<ul style="list-style-type: none"> • Michele Claus (EURAC) • Alexander Jacob (EURAC) • Björn Backeberg (Deltares) • Frederique de Groen (Deltares) • Joost Buitink (Deltares) • Roel de Goede (Deltares) • Donatello Elia (CMCC) • Gabriele Accarino (CMCC) • Sandro Fiore (UNITN) • Christian Pagé (CERFACS) • Matthias Schramm (TU Wien) • Bernhard Raml (TU Wien) • Christoph Reimer (EODC) 		
Reviewers	<ul style="list-style-type: none"> • Thomas Geenen (ECMWF) • Miruna Stoicescu (EUMETSAT) 		
Moderated by:	<ul style="list-style-type: none"> • Charis Chatzikyriakou (EODC) • Sjomara Specht (EGI) 		

D7.1 Report on requirements and thematic modules definition for the environment domain

Approved by	Andrea Manzi (EGI), on behalf of the TCB
--------------------	--



Revision History			
Version	Date	Description	Contributors
V0.1.1	30/03/2023	Near complete draft of Section 2.3	Björn Backeberg, (Deltares), Frederique de Groen, (Deltares), Joost Buitink, (Deltares), Roel de Goede (Deltares),
V0.2	04/05/2023	Complete draft	Michele Claus, and the other authors
V0.3	26/05/2023	Internal reviewed version	Thomas Geenen, (ECMWF) Miruna Stoicescu (EUMETSAT)
V0.4	07/06/2023	Version ready for TCB approval	Michele Claus (EURAC) and the other authors
v0.5	12/06/2023	Version approved by TCB	Andrea Manzi (EGI), Björn Backeberg (Deltares), Donatello Elia (CMCC)
V1.0	12/06/2023	Final	

Terminology / Acronyms	
Term/Acronym	Definition
DT	Digital Twin
DTE	Digital Twin Engine
EWE	Extreme Event Workflow
ML	Machine Learning
AI	Artificial Intelligence
CMIP	Coupled Model Intercomparison Project



D7.1 Report on requirements and thematic modules definition for the environment domain

ERA5	Fifth generation ECMWF reanalysis for the global climate and weather
TM	Thematic Module
NetCDF	Network Common Data Form
CSV	Comma-Separated Values
C3S	Copernicus Climate Change Service
ESGF	Earth System Grid Federation
API	Application Programming Interface
IBTrACS	International Best Track Archive for Climate Stewardship
SSP	Shared Socio-economic Pathways
EO	Earth Observation
STAC	Spatio Temporal Asset Catalog
OGC	Open Geospatial Consortium
JSON	JavaScript Object Notation
GeoJSON	Geographic JSON
SQL	Structured Query Language
CEOS	Committee on Earth Observation Satellites
CARD4L	CEOS Analysis Ready Data for Land
ARD	Analysis Ready Data
DEM	Digital Elevation Model
EAD	Expected Annual Damages
GIS	Geographic Information System
OSM	OpenStreetMap
IDF	Intensity Duration Frequency
CLI	Command Line Interface

D7.1 Report on requirements and thematic modules definition for the environment domain

CPU	Central Processing Unit
GPU	Graphics Processing Unit
MPI	Message Passing Interface
OS	Operating System
CI/CD	Continuous Integration/Continuous Deployment
URL	Uniform Resource Locator

Terminology / Acronyms: <https://confluence.egi.eu/display/EGIG>



Table of Contents

1	<i>Introduction</i>	11
1.1	Scope	11
1.2	Document Structure	11
2	<i>Initial design of thematic modules in the environment domain</i>	12
2.1	T7.4 Climate analytics and data processing	12
2.1.1	Overview of the digital twins on extreme events (T4.5, T4.6, T4.7)	12
2.1.2	Example workflow of the digital twins on extreme events	12
2.1.3	Description of thematic modules for climate analytics and processing	17
2.2	T7.5 Earth observation modelling and processing	22
2.2.1	Overview	22
2.2.2	Thematic Modules.....	24
2.3	T7.6 Hydrological model data processing	26
2.3.1	Overview	26
2.3.2	Models and data	29
2.3.3	Components.....	44
2.3.4	User interaction.....	51
3	<i>Requirements for the thematic modules in the environment domain</i>	54
3.1	General Description and Categorization of requirements	54
3.2	Storage I/O	55
3.2.1	Input data requirements.....	55
3.3	Databases	56
3.4	Computing	56
3.5	OS and execution framework	57
3.6	Machine Learning	57
3.7	Real-time data acquisition and processing	57
3.8	Data formats	57
3.9	Workflow tools	57
3.10	Visualisation	58
3.11	Data Sharing	58
4	<i>Conclusions</i>	59
5	<i>References</i>	60



Table of Figures

Figure 1 - C4 container-level diagram about the thematic modules used for the digital twin applications on extreme events on climate projections (Task 4.5).....	13
Figure 2 - High-Level workflow of the T4.7 Digital Twin on climate extremes generic event detection.	13
Figure 3 - Context of the planned flood and drought detection workflows that will be used in digital twins for Early Warning of Extreme Events (T4.6).	22
Figure 4 - High-level overview, showing the components of the planned flood detection workflow and their relation to the data storage and other containers.	23
Figure 5 - High-level overview, showing the components of the planned drought detection workflow and their relation to the data storage and other containers.	23
Figure 6 - Calculating bottom of the atmosphere surface reflectances from Sentinel-2 by using the openEO web editor.	25
Figure 7 - Calculating backscatter from Sentinel-1 ARD by using the openEO web editor.	26
Figure 8 - High-level overview of the data and components of the Digital Twin for Flood early warning (FloodAdapt early warning; T4.6) and climate change impact (FloodAdapt climate impact; T4.7; blue box) in coastal and inland regions.	28
Figure 9 - Overview of the workflow components mapped against developments in other work packages and tasks of the project in C4-Model Level 3 format.....	29
Figure 10 - Definition of risk (Kron, 2005).	37
Figure 11 - High-level workflow of Delft-FIAT.....	37
Figure 12 - The context of risk analysis for road infrastructure (Bles et. al 2019).	41

Table of Tables

Table 1 - Data sources required for extreme events on climate projections	15
Table 2 - Requirements for the DT on climate extremes generic events detection	16
Table 3 - Static data for model building	30
Table 4 - Dynamic data for boundary conditions.....	32
Table 5 - Static data for model building	33
Table 6 - Dynamic data for boundary conditions to a SFINCS model.....	35
Table 7 - Data input sources to HydroMT-FIAT to build a Delft-FIAT model.....	38
Table 8 - Delft-FIAT model data description.....	40
Table 9 - Potential data input sources to HydroMT-RA2CE (to be developed) to build a RA2CE model.	42
Table 10 - RA2CE model data description.	43
Table 11 - Available preprocessing tools for each model.....	47
Table 12 - Available run options for each model.....	48
Table 13 - Available postprocessing tools for each model.	50

Executive summary

This document is the deliverable 7.1 of the interTwin project, part of work package 7. It is a report collectively written by the partners of tasks 7.4, 7.5 and 7.6, who are directly involved in the design of digital twins for the environment domain. In this report the focus is towards the definition of the requirements and thematic modules necessary for the creation of the digital twins.

Starting from a description of the use case scenarios, several thematic modules are defined. Those reusable components will be integrated into the Digital Twin Engine (DTE) and could be used by multiple scientists in different workflows.

Based on the provided textual and schematic digital twin descriptions and thematic modules, all the requirements are gathered and subdivided in several categories: storage I/O, databases, computing, operating system and execution framework, machine learning, real-time data acquisition and processing, data formats, workflow tools, visualisation, and data sharing.

1 Introduction

1.1 Scope

The scope of this document is to give an overview of the environment domain thematic modules (T7.4, T7.5, T7.6) and their requirements developed in the interTwin project.

The thematic modules will enhance the capabilities of the core engine running the digital twins by adding functionalities to several fields:

- Machine Learning: data gathering, filtering, cleaning, harmonisation, augmentation. Event detection and attribution.
- Workflow automation: triggering of workflows upon arrival of new data.
- openEO ([Schramm et al., 2021](#)): processes for vector data processing and weather station data filtering and harmonisation.

The design of the thematic modules will follow the specific requirements provided by the WP4 – Technical co-design and validation with research communities, where the use cases are being defined. Additionally, specific requirements for each thematic module will be defined and listed in this document.

1.2 Document Structure

The organisation of the document is as follows:

- [Section 2](#) includes the description of the preliminary design of the thematic modules developed for each individual environment use case. This section starts with a description of the climate analytics and data processing (T7.4), which includes the overview of the digital twins on extreme events and the related ML thematic modules. The following subsection, earth observation modelling and processing (T7.5) introduces thematic modules based on openEO and how they will be used in the global flood and drought monitoring tools. Lastly, the third subsection describes the hydrological model data processing components (T7.6), describing in detail the required data, hydrological models, and processing environment.
- [Section 3](#) presents a collection of general requirements for the thematic modules and use cases previously described. The requirements have been harmonised and subdivided in several categories: storage I/O, databases, computing, operating system and execution framework, machine learning, real-time data acquisition and processing, data formats, workflow tools, visualisation, and data sharing.

2 Initial design of thematic modules in the environment domain

2.1 T7.4 Climate analytics and data processing

2.1.1 Overview of the digital twins on extreme events (T4.5, T4.6, T4.7)

The goal of T7.4 is to develop key components that will enable a fast and flexible design and implementation of Digital Twin (DT) applications based on climate data (e.g., CMIP, ERA5, and potentially DestinE data). Such components can be composed and linked together in order to support DTs development in the context of WP4 (T4.5, T4.6 and T4.7).

In particular, for the DT application setup in T4.5, the focus will be on handling data for extreme events in weather and climate. The main idea is to support DT applications for storms and fires as well as generic extreme events (T4.7) that use Machine Learning to project the likelihood of occurrence of such events in future scenarios (e.g., CMIP6) with the aim of giving an indication about their temporal trend and geographical occurrence across the globe based on different levels of anthropogenic forcing. In this sense, these DTs will enable what-if and sensitivity analyses for these extreme events by exploiting the capabilities of the thematic modules from WP7 (and in particular those for climate analytics and processing - T7.4) and the core modules from WP6.

Two main levels of users interacting with the DTs have been identified:

- **Developers:** which will be using the DT engine components (thematic and core modules) to build a new DT application. Developers will directly access the thematic modules described here and can customise them, as well as the workflow, to meet specific needs for new users.
- **End-users** (scientists, policy makers, stakeholders): which will be directly using the DTs applications to be implemented in WP4. These users will access the data and visualisation capabilities offered by the application with the aim of collecting relevant findings from a scientific standpoint. Furthermore, the application will enable what-if and sensitivity analyses to understand how anthropogenic forcings can impact on Climate Extreme Events in future projection data.

2.1.2 Example workflow of the digital twins on extreme events

This section provides a high-level view of the workflows followed by the WP4 DT applications exploiting the climate analytics and data processing thematic modules.

D7.1 Report on requirements and thematic modules definition for the environment domain

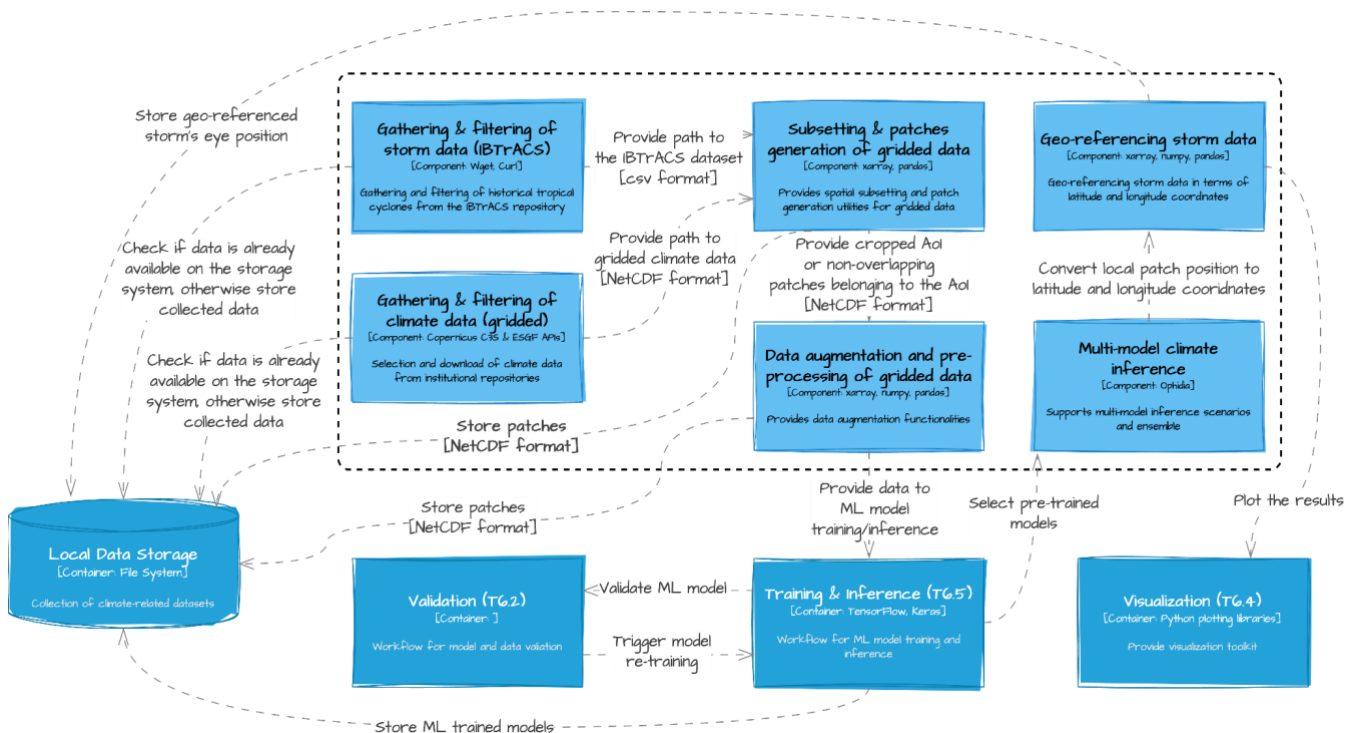


Figure 1 - C4 container-level diagram about the thematic modules used for the digital twin applications on extreme events on climate projections (Task 4.5)

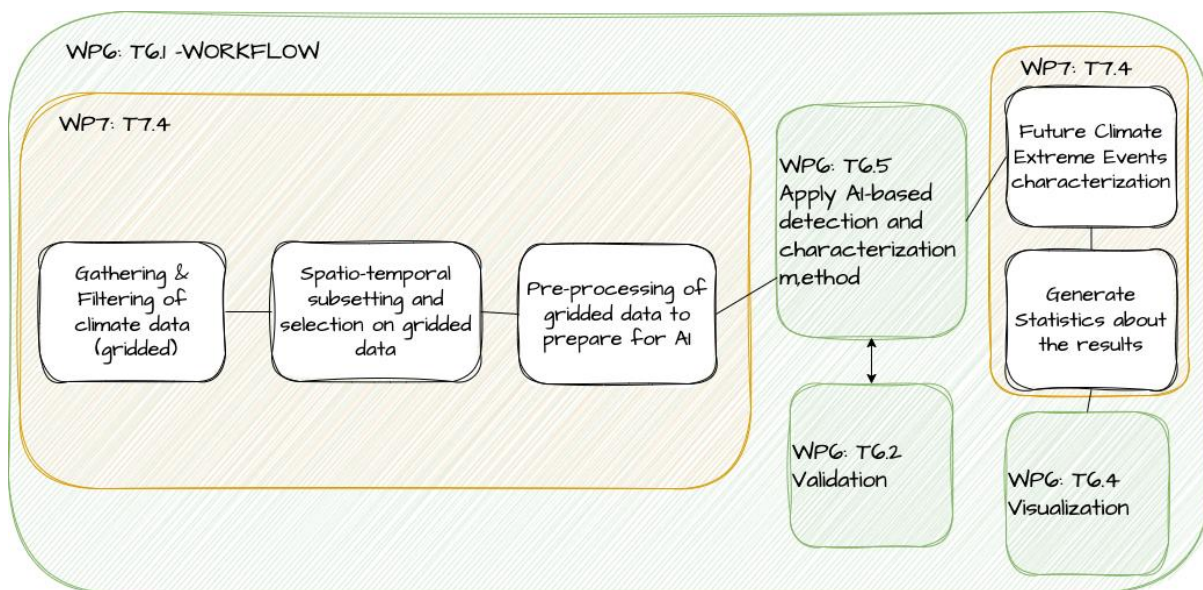


Figure 2 - High-Level workflow of the T4.7 Digital Twin on climate extremes generic event detection.

The two figures above show workflow examples for two digital twins (from T4.5 and T4.7 respectively) on extreme events composed by thematic modules (TM) from T7.4 and core modules from WP6. It must be noted that these do not describe workflow implementations of any specific application, but rather aim to provide an overview of the main steps that would be followed by a potential application from T4.5 and T4.7.



For example, concerning a potential application from T4.5 (sketched in **Figure 2**), the application would comprise the following logical steps:

1. Users can select the proper data for running their analysis; i.e., specifying the temporal and spatial extensions to be considered or the models from CMIP experiments;
2. The selected data will be pre-processed so that it can be used as input to pre-trained ML model for inference (e.g., for detecting the presence of a Tropical Cyclone);
3. Re-training of a ML model can be triggered in order to use different hyper-parameters, if results from inference are not satisfactory in terms of evaluation metrics (e.g., error and classification metrics). At this stage, patch generation and data augmentation procedures of ERA5 data are also required to increase the number of training examples.
4. Results from the inference stage on projection data will then be post-processed to build the final results (e.g., to build the spatial grid);
5. Results on different CMIP experiments can be combined together to reduce the uncertainties of the results from the ML models on a single CMIP output;
6. Final results can be provided to users either as output files or as customizable visualisations tailored to the use case.

The following TMs for climate analytics and processing required by the DT applications have been identified:

- Data gathering and filtering of climate data: to gather gridded data (e.g., in NetCDF format) from multiple climate repositories and filter the results
- Data gathering and filtering of storms data: to gather storms data in CSV format from the IBTrACS repository¹ and filter the results
- Subsetting and patches generation on gridded data: to spatially subset climate data on a geographical domain, and generate patches of given latitude and longitude size
- Data augmentation and pre-processing of gridded data: to perform augmentation and pre-processing operations on climate/gridded data (e.g., for a ML model)
- Multi-model climate inference scenarios: to compute a multi-model ensemble on climate/gridded data (e.g., results from ML model)
- Future Climate Extreme Events characterization: to perform the characterization of the changes of extreme events in the future climate, according to the detected extreme events.

¹ <https://www.ncei.noaa.gov/products/international-best-track-archive>

D7.1 Report on requirements and thematic modules definition for the environment domain

- Geo-referencing of storm data: for geo-referencing storms data in a local domain with respect to the global domain. This step could be applied before visualisation for plotting the results on the global map.
- Statistics of extreme events changes: statistics and end-user products will be calculated from the output (e.g., ML model).

The following subsections will present in detail the aforementioned components.

In terms of data requirements, multiple data sources will be considered in the use cases for EWEs. The following table reports about the main data sources used. All data will be gathered and pre-processed by the specific thematic modules ([see 2.1.3.1](#) and [2.1.3.2](#)).

Table 1 - Data sources required for extreme events on climate projections

Datasets	ERA5	CMIP6	IBTrACS	Fire Danger Indices
Data types	Reanalysis-Gridded	Projections-Gridded	Observative records	Reanalysis - Gridded
Data formats	NetCDF	NetCDF	CSV	NetCDF
Spatial resolution and coverage	~0.25°x0.25°/global	~0.25°x0.25°/global	geographical coordinates/global	~0.25°x0.25°/global
Temporal resolution and extent	Hourly up to daily/ 1979-2020 (past data)	6-hourly up to daily /2015-2100 (projections)	3-hourly/1979-2020 (past data)	Daily at 12:00 / 1979-2020 (past data)
Update frequency	Daily with 5-day latency	Very rarely	Twice a week (at least a year to gather significant data)	Monthly
Storage Requirements	O (100) GB	O (1) TB (under evaluation)	O (100) MB	O (10) GB



D7.1 Report on requirements and thematic modules definition for the environment domain

Usage	Model training, validation, and test	Multi-model inference	Model training, validation, and test	Model training, validation, and test
APIs/Tools	Copernicus CDS	Synda	wget/curl	Copernicus CDS
DT Application	Storms/Fires	Storms/Fires	Storms	Fires

Table 2 - Requirements for the DT on climate extremes generic events detection

Datasets	CMIP6	CMIP6 Climate Indices	ERA5
Data types	Projections-Gridded	Historical and Projections of Climate Indices - Gridded	Reanalysis-Gridded
Data formats	NetCDF	NetCDF or zarr	NetCDF
Spatial resolution and coverage	~0.25°x0.25°/global	~0.25°x0.25°/global	~0.25°x0.25°/global
Temporal resolution and extent	6-hourly up to daily /1850-2014 (historical) and 2015-2100 (projections)	Annual 1850-2014 (historical) & 2015-2100 (projections)	Daily (near real time data, seasonal or event based)
Update frequency	Very rarely	Very rarely	Daily with 5-day latency
Storage Requirements	O (1) TB (under evaluation)	O (300) GB (under evaluation)	O (100) GB
Usage	Model training and calculations	Model training and calculations	Data execution Analysis



APIs/Tools	Synda	wget/curl	Copernicus CDS
DT Application	Generic events characterization and identification	Extreme events characterization and identification	Extreme events attribution

2.1.3 Description of thematic modules for climate analytics and processing

Gathering & Filtering of climate data (gridded)

The gathering & filtering thematic module for climate data (e.g., NetCDF gridded data) provides support to the selection and download of data, such as ERA5 reanalysis and CMIP6 projections, from institutional climate repositories (i.e., C3S and ESGF data nodes) through specific APIs and protocols. Users of these thematic modules are allowed to select the required variables, grid resolution, projection scenarios (i.e., in the case of projection data), and the temporal extent of such data, also specifying the temporal frequency. DT applications will exploit this module specifying the variable required for its implementation. Furthermore, the module allows checking if the requested files are already present on the storage system, thus skipping the download, or, discretionally, the user might force an update of the selected data if needed, in such case the module will synchronise the local version with the remote data. As an example, for the storms use case, a user might use this module for retrieving climate data that are related to the cyclogenesis of tropical storms from the ERA5 repository with a temporal resolution of 6 hours, from 1980 to present as the temporal extent.

Input of the module

- The specific climate repository from which data should be retrieved (e.g., ERA5 or CMIP6)
- Temporal extent (e.g., 1980 to 2022)
- Temporal resolution (e.g., hourly, or 6-hourly)
- List of variables to be retrieved
- Grid resolution
- Projection scenarios (i.e., ssp)
- force_download [Bool]: whether to force the download of requested data even if it is present on the storage system

Output of the module

- Returns a reference (e.g., object, variable) to the retrieved or existing data in NetCDF format

Computational and software requirements



- Standard python libraries
- Synda
- intake/intake-esm
- Conda env

Gathering & Filtering of storm data (IBTrACS)

This module is specifically designed for gathering historical up to more recent tropical cyclones best track data across the globe from the International Best Track Archive for Climate Stewardship (IBTrACS) repository. Furthermore, the module allows filtering records based on user-defined conditions and queries and in addition allows checking if the requested files are already present on the storage system, thus optionally skipping the download..

Input of the module

- The type of repository to be downloaded (e.g., ALL tracks or tracks belonging to a specific formation basin, such as East Pacific, North Atlantic, etc.)
- `force_download` [Bool]: whether to force the download of requested data even if it is present on the storage system
- Type of filtering and selection to be performed (e.g., discard provisional tracks, select only tracks every 6 hours, etc.)

Output of the module

- Returns a reference (e.g., object, variable) to the cleaned IBTrACS dataset in csv format

Computational and software requirements

- Wget/curl
- Standard python libraries
- Conda env

Subsetting & patches generation on gridded data

This module provides spatial subsetting and patch generation utilities for gridded data. In particular, information gathered through the Gathering & Filtering of climate grid-based data thematic module ([2.1.3.1](#)) can be further subsetted to a specific geographical domain of interest which is defined by the user. Moreover, for the storm's application, the geographical location of storms provided in IBTrACS through the Gathering & Filtering of storm data module ([2.1.3.2](#)) can be used to split climate data into non-overlapping patches of customizable size.

Input of the module

- The output of the Gathering & Filtering of climate grid-based data thematic module (NetCDF format)
- The output of the Gathering & Filtering of storm data thematic module

D7.1 Report on requirements and thematic modules definition for the environment domain

- geographical bounds for spatial subsetting (CSV format)
- Size of the patches to be generated
- patches_generation [Bool]: whether to generate non overlapping patches of fixed size

Output of the module

- Cropped area of interest or non-overlapping patches belonging to the area of interest (in NetCDF files)

Computational and software requirements

- Standard python libraries
- Xarray, Pandas
- Conda env

Data augmentation and pre-processing of gridded data

This module allows performing different data augmentation procedures on input gridded data in order to increase the number of examples for training ML models for climate applications. Data augmentation positively affects training performance by mitigating overfitting issues. The module implements augmentation techniques for 2-dimensional gridded data, such as horizontal and vertical flip or both, 180 degrees rotation, as well as supports the same operations on input patches that can be the output of the Subsetting & patches generation thematic module ([2.1.3.3](#)). If input data is labelled, the module will update the label(s) according to the selected augmentation procedure(s).

Input of the module

- Reference (e.g., object, variable) to gridded data (NetCDF format) that should be augmented
- [Optional] List of labels related to input gridded data
- List of data augmentation type (e.g., horizontal, vertical, 180 rotations)

Output of the module

- Returns a reference (e.g., object, variable) to the augmented data in NetCDF format

Computational and software requirements

- Standard python libraries
- Xarray, Numpy, Pandas
- Conda env

Multi-model climate inference

This module supports multi-model inference scenarios where output from the processing of different models can be combined together into an ensemble and aggregated according to different metrics (e.g., avg, min, max, ...). The module can be applied as one



of the final stages in more complex workflows to data produced from previous stages on different ML data-driven models and/or on models from multiple climate projections with different SSP-based scenarios (Shared Socio-economic Pathways: that embed different levels of anthropogenic forcings). The goal is to integrate different results to ameliorate the uncertainties.

Input of the module

- References (e.g., object, variable) to the NetCDF files to be considered in the multi-model inference (e.g., the output from the ML model)
- Type of operation to be applied on the data ensemble

Output of the module

- Reference (e.g., object, variable) to the NetCDF files produced as output of the model ensemble

Computational and software requirements

- Standard python libraries
- PyOphidia/Ophidia
- Conda env
- 10s to 100s of GB of memory for processing multiple datasets

Geo-referencing storm data

This module is specifically designed for geo-referencing storm data in terms of latitude and longitude coordinates. Starting from the row-column position of the storm's eye within the patch (e.g., a prediction from a ML model), the module retrieves the global latitude and longitude coordinates rounded at the spatial resolution of the gridded data. This module can be used before visualisation (from WP6 modules).

Input of the module

- Reference (e.g., object, variable) to patches to be geo-referenced
- List of local row-column position of the storm's eye related to the input patches
- Indices (row-column) of the patches within the entire geographical domain of interest

Output of the module

- List of geo-referenced storm's eye position in terms of global latitude and longitude

Computational and software requirements

- Numpy, Pandas, Xarray
- Conda env

Future Climate Extreme Events characterization

This module is designed to perform the characterization of the changes of extreme events in the future climate, according to the detected extreme events. The specific AI method that will be used is currently in development, but several ML-based methods were already pre-identified. The module will retrieve climate indices gridded data for the future climate in a specific bounded-box spatial region and in a specific time period, in order to apply an AI-based model to characterise the change compared to the historical period, in several climate models independently, and the results will be combined together to assess the uncertainties in the expected changes. This module can be used before visualisation (from WP6 modules).

Input of the module

- Climate indices (gridded data) based on CMIP6 data: Paths to the NetCDF files to be considered.
- Basic variables (gridded data) based on CMIP6 data: Paths to the NetCDF files to be considered, depending on the climate index considered (temperature, precipitation, winds, etc.)
- Spatial region, time period, climate index to consider (iclim python software implemented climate index only).

Output of the module

- Multi-model output of the AI model: change of characteristics of the climate index.

Computational and software requirements

- Standard python libraries
- Numpy, Pandas, Xarray
- Conda env
- 10s to 100s of GB of memory for processing multiple datasets

Statistics of extreme events changes

This module is designed to calculate statistics on the output of the AI-based model that characterise the changes of extreme events in the future climate.

Input of the module

- Multi-model output of the AI model: change of characteristics of the climate index.

Output of the module

- Multi-model statistics of the change of characteristics of the climate index.

Computational and software requirements

- Numpy, Pandas, Xarray

2.2 T7.5 Earth observation modelling and processing

2.2.1 Overview

The interTwin thematic module T7.5 Earth Observation Modelling and Processing will develop the necessary building blocks to run Digital Twins based on EO data, with **openEO** as the driving technology.

Digital twin implementations using the modules developed in this task are global flood and drought monitoring tools, which will be firstly described in D4.1 - First Architecture design of the DTs capabilities for climate change and impact decision support tools.

These digital twins will be used as a reference to demonstrate the necessary modules and how we plan to implement them. **Figure 3** to **Figure 5** show in form of a C4-Model a technical overview (Context) of the flood and drought related developments and in more detail the cores of both implementations (Containers). These workflows, implemented using the openEO syntax, will represent parts of digital twins for early warning of extreme events (T4.6).

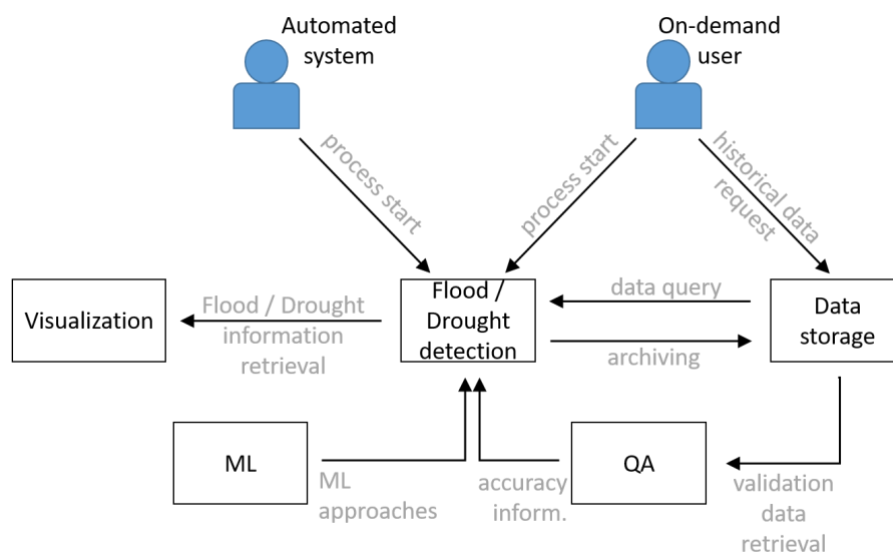


Figure 3 - Context of the planned flood and drought detection workflows that will be used in digital twins for Early Warning of Extreme Events (T4.6).

The containers regarding flood and drought are shown in more detail in **Figure 4** and **Figure 5**.

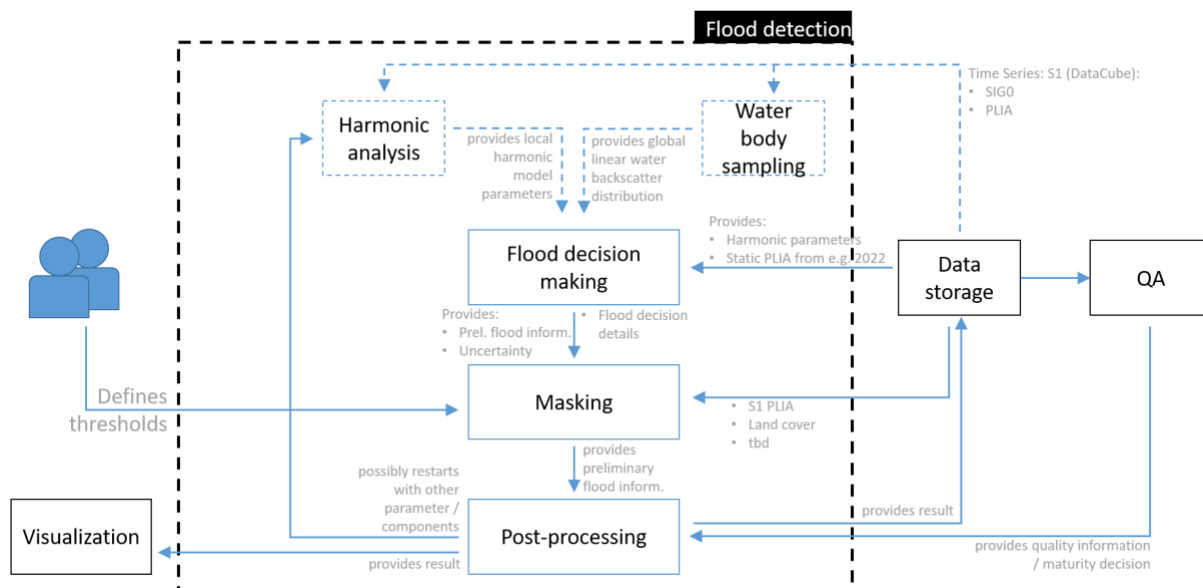


Figure 4 - High-level overview, showing the components of the planned flood detection workflow and their relation to the data storage and other containers.

Components in dashed rectangles are defined as stretched goals, to be implemented in an iterative manner. Until then, their expected outcomes will be delivered as existing static products.

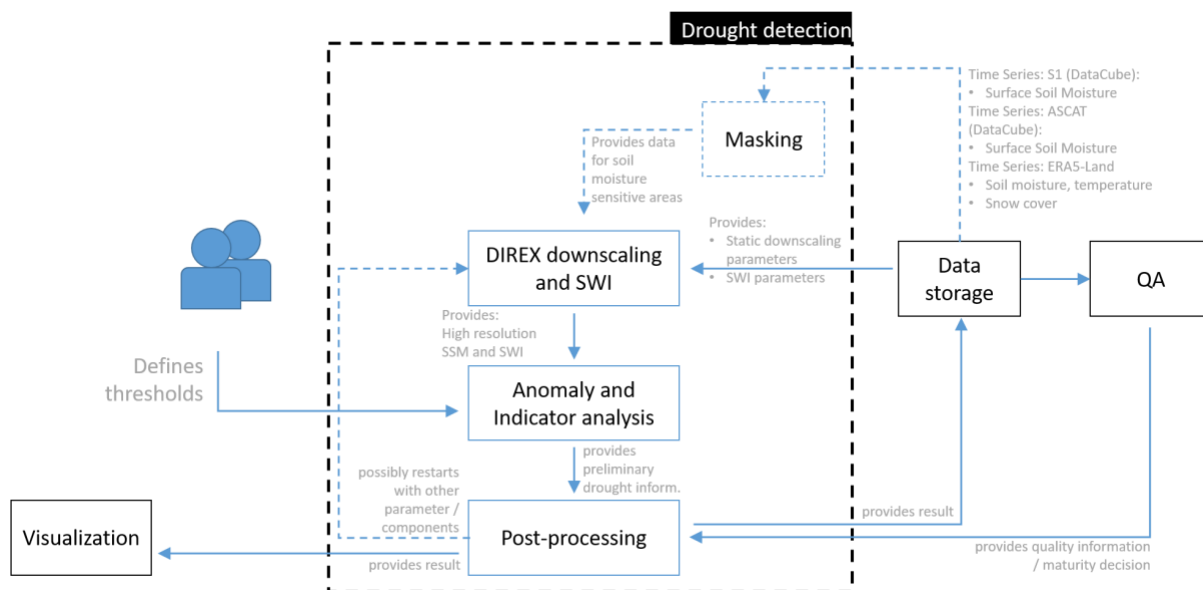


Figure 5 - High-level overview, showing the components of the planned drought detection workflow and their relation to the data storage and other containers.

Components in dashed rectangles are defined as stretched goals, to be implemented in an iterative manner. Until then, their expected outcomes will be delivered as existing static products.



2.2.2 Thematic Modules

Index raster and vector data in openEO

Each digital twin requires some input data that will be provided by the infrastructure of WP5. To use this data in the most efficient way, we want to offer a thematic module that defines how this data will be indexed and served via openEO. openEO is based on the theoretical concept of datacubes, multi-dimensional arrays with additional information about their dimensionality, on which a set of predefined processes can be applied. The HTTP API specification of openEO requires a Spatio Temporal Asset Catalog (STAC) and [OGC API - Features - Part 1: Core compliant endpoint](#) (/collections) to be served. Accordingly, data to be used within openEO has to be made available following either STAC specifications or the OGC API - Features standard.

The core atomic unit known by STAC is the so-called STAC Item, representing a single spatiotemporal resource as a GeoJSON feature plus datetime and links. A spatiotemporal resource is composed of a set of digital assets such as the actual data file, auxiliary files, and metadata. Hence, data to be processed with openEO have to be preprocessed to be accompanied with STAC items (GeoJSON), if not already available. Those STAC items can be stored next to the actual data on any filesystem or cloud storage often referred to as static STAC collection / catalogue. STAC items can be logically grouped into a STAC catalogue or STAC collection. For further details about the STAC specifications the official documentation, <https://stacspec.org>, shall be consulted. Software tools to create the required STAC metadata for certain earth observation datasets is made available by the community under <https://github.com/stactools-packages>.

The STAC family of specifications incorporates the so-called STAC API representing a dynamic version of a SpatioTemporal Asset Catalog. The STAC API can be used to retrieve STAC Catalog, Collection, Item, or STAC API ItemCollection objects from various endpoints. STAC catalogue and collection objects are returned as JSON, while Item and ItemCollection objects are GeoJSON-compliant entities with foreign members. Typically, a single Feature is used when returning a single Item object, and FeatureCollection when multiple Item objects (rather than a JSON array of Item entities). In order to serve a STAC API to users a server implementation is needed. The STAC community provides a number of different implementations based on different technology stacks (Java, Python, Node.js, PostgreSQL, OpenSearch, etc.). Those are maintained and available via <https://github.com/stac-utils>. The `stac-fastapi`, <https://stac-utils.github.io/stac-fastapi/>, represents a mature implementation of the STAC API as a Python FastAPI application. The served STAC objects, JSON, can be stored in a PostgreSQL (PostGIS) database connected via one of the supported backend drivers.

Indexing STAC objects, JSON, into the a given PostgreSQL backend is done via client libraries either tailored to a given backend, `pyPgSTAC` (<https://stac-utils.github.io/pgstac/pypgstac/>), or via standard SQL insert commands.

Run openEO process graph

openEO ([Schramm et al., 2021](#)) is a flexible communication standard between users and cloud services, providing predefined processes for Earth Observation data that standardised the use of custom processing pipelines on diverse cloud platforms. Since openEO will be used in different digital twins and for different objectives, we have to define how the openEO workflows interact with the other components of interTwin. For this reason, we want to create a thematic module specialised in running openEO process graphs. The input data will be provided by the thematic module described in [Section 2.2.2.2](#) and the result will be as well indexed in the same way, so that it will be possible to use it further from another component/thematic module.

Data pre-processing

Leveraging the openEO processes we can pre-process optical and radar data:

- Optical data pre-processing: we can apply atmospheric correction on optical data (such as Sentinel-2 or Landsat) via the process [ard_surface_reflectance](#), which computes CARD4L compliant surface (bottom of atmosphere/top of canopy) reflectance values from optical input.

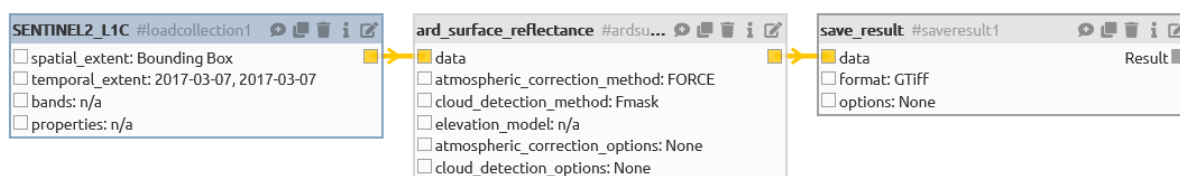


Figure 6 - Calculating bottom of the atmosphere surface reflectances from Sentinel-2 by using the openEO web editor.

Data is queried from a data cube (SENTINEL2_L1C), transferred into surface reflectance information (ard_surface_reflectance) and stored in a GTiff format (save_result) for its further use.

- Radar data pre-processing: The Sentinel-1 microwave datasets are provided at the participating storage systems as Analysis-Ready-Data (ARD): they are gridded, formatted and stored into a data cube. Thus, this data can be queried, accessed, and processed from different storage platforms / at processing service providers in a comparable way. Possible differences between the storage strategies of different cloud platforms (e.g., different grid / projection / data format etc.) will be compensated by using the openEO communication standard. This allows the user (here: the workflow) to generally request for data, while an openEO backend API at the service provider’s side translates this request into the locally needed syntax. The preprocessed Sentinel-1 data will be used in a first step to compute backscatter information, defining several parameters (as e.g., the use of specific exclusion masks, of a specific digital elevation model, or the choice between the radar backscatter conventions “Sigma” and “Gamma”. This backscatter

information can be calculated via the openEO syntax, using the process **“sar_backscatter”** (Figure 7).

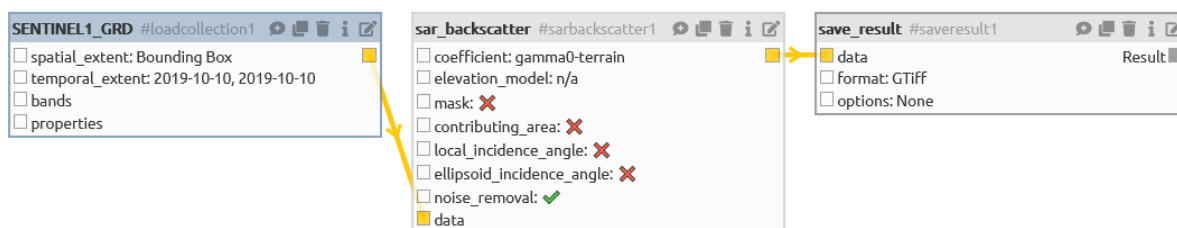


Figure 7 - Calculating backscatter from Sentinel-1 ARD by using the openEO web editor.

Data are queried from a data cube (SENTINEL1_GRD), transferred into backscatter information (sar_backscatter) and stored in a GTiff format (save_result) for its further use.

Data masking

The used optical and microwave satellite data are not suitable to identify flood or drought events in all regions; environmental factors as specific land cover types (e.g., forestry, snow / ice, or urban regions), clouds (optical data) or subsurface signal backscattering (microwave data) can reduce the accuracy of the targeted outcome up to its unusability. Therefore, those areas will be masked out within the calculation processes; the models will not provide any results. In a first step, static information is provided by TU Wien on known exclusion masks; they will be stored manually at the used cloud storage systems and can be applied to the result via the openEO process **“mask”**. In further iterations, those masks will be dynamically created, using commands in the openEO syntax.

Drought and flood monitoring workflows

Both workflows can be developed completely by using existing processes of the openEO syntax. Main scope will lay on the processes **“reduce dimension”** and **“aggregate temporal”**: the first to use specific functions to calculate pixel values from time series; the latter to group pixels to single information. The specific scripts will be developed throughout the project; they will be further used within the digital twins for drought and flood early warning systems (T4.6).

2.3 T7.6 Hydrological model data processing

2.3.1 Overview

In WP4 the objective is to demonstrate that the architecture, tooling, and capabilities of the Digital Twin Engine (WP5 and WP6) can support the implementation of digital twins in different domains.

Here (T7.6) the focus is on developing the necessary components to facilitate automatically setting up local flood hazard and impact models anywhere on Earth.

Figure 2.3.1a provides a high-level overview of the workflow for the process-based models used in the implementation of the Digital Twin for coastal and inland flooding. The workflow will support the implementation of two Digital Twins in WP4:

1. Digital Twin for flood early warning in coastal and inland regions (hereafter referred to as *FloodAdapt early warning*).
2. Digital Twin for climate change impact in coastal and inland regions (hereafter referred to as *FloodAdapt climate impact*)

The *complete* workflow will:

1. Allow a Digital Twin developer / implementer to specify an area of interest,
2. Once an area of interest is specified, **HydroMT (the model builder)** is triggered, which automatically creates model schematisations for **WFLOW (hydrological model)**, **SFINCS (Super-Fast INundation of CoastS)**, **Delft-FIAT (Fast Impact Assessment Tool)** and **RA2CE (Resilience Assessment and Adaptation for Critical infrastructurE Toolkit)**.
3. Run WFLOW and SFINCS to produce flood maps, which will be augmented with Earth-Observation-based flood maps that represent a part of the **Global Flood Monitoring** processing chain of the Copernicus Emergency Mapping Service (CEMS) (T7.5).
4. Post-process the flood maps to create deterministic and probabilistic flood maps and serve these as input data for Delft-FIAT and RA2CE which determine the impact of the flood on buildings, utilities, roads, and accessibility.
5. Post-process the data for visualisation and analysis (e.g., Webapp or Jupyter Notebooks)
6. For FloodAdapt climate impact, the user will be able to define mitigation measures or what-if scenarios and rerun the simulations to e.g., understand the impact of building flood protection in specific areas, or understand the impact of a scenario where twice the amount of rainfall occurs.

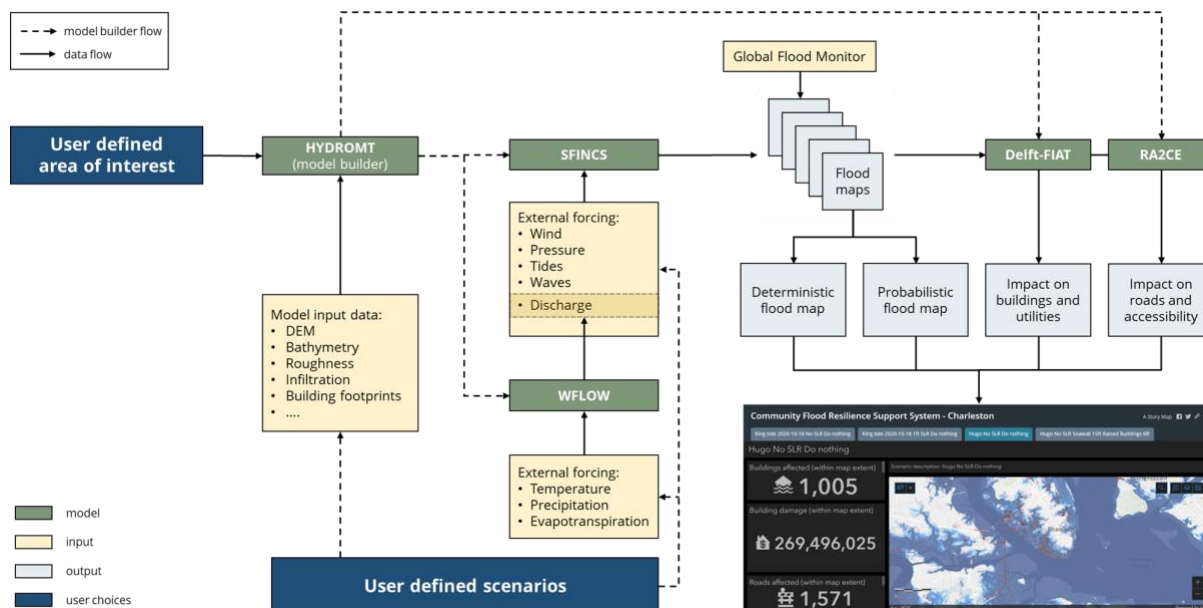


Figure 8 - High-level overview of the data and components of the Digital Twin for Flood early warning (FloodAdapt early warning; T4.6) and climate change impact (FloodAdapt climate impact; T4.7; blue box) in coastal and inland regions.

The focus for T7.6 is to develop the necessary components enabling:

1. Model building: Setting up a model schematisation (grid) for a user defined area of interest for WFLOW, SFINCS, Delft-FIAT and RA2CE.
2. Preprocess boundary condition data: Preprocess the dynamic data to the format needed by the models for boundary conditions.
3. Running the models: Building the docker and Singularity containers to run the models on heterogeneous compute and data infrastructures.
4. Postprocessing the output data: Postprocessing the output data from the models for visualisation and analysis interfaces.

A more technical overview of the components for the Digital Twin for coastal and inland flooding is depicted in **Figure 8**. Arrows from the “Run Model” boxes to “Prepare model run” boxes indicate that the output of one model serves as (direct or to-be-processed) input data to another model. It is still to be determined if the Area of Interest is defined via a Graphical User Interface (GUI) or via another method. All HydroMT components and models are described in the following sections.

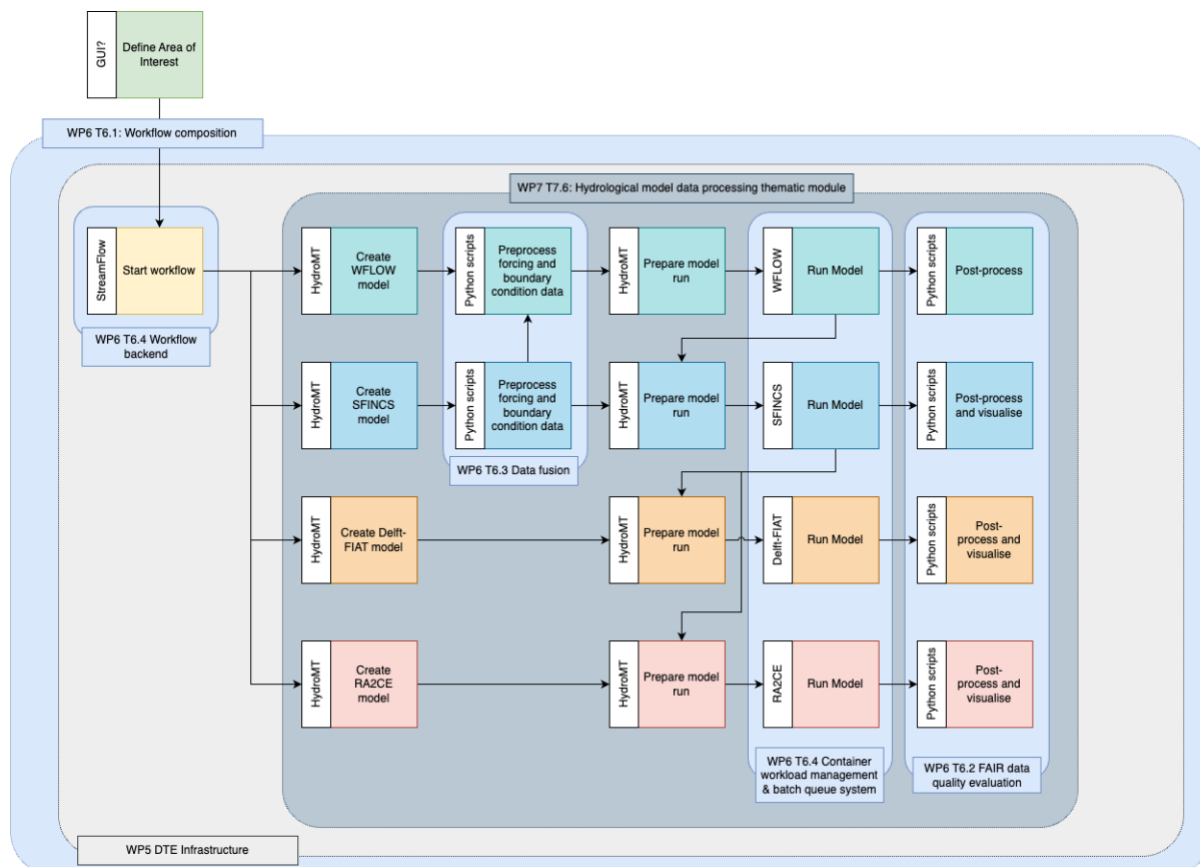


Figure 9 - Overview of the workflow components mapped against developments in other work packages and tasks of the project in C4-Model Level 3 format.

This chapter is structured as follows. [Section 2.3.2](#) describes the process-based models and data specifically needed for the models. [Section 2.3.3](#) describes the model building, preprocessing, model running and postprocessing components in development as part of T7.6. [Section 2.3.4](#) describes the anticipated user interaction from the perspective of two users: (1) a Digital Twin developer / implementer, and (2) a Digital Twin end-user / decision maker.

2.3.2 Models and data

WFLOW

WFLOW models hydrological processes, allowing users to account for precipitation, interception, snow accumulation and melt, evapotranspiration, soil water, surface water and groundwater recharge in a fully distributed environment. Based on gridded topography, soil, land use and climate data, WFLOW calculates all hydrological fluxes at any given grid cell in the model at a given time step. WFLOW has been successfully applied worldwide for analysing flood hazards, drought, climate change impacts and land use changes.

WFLOW is conceived as a framework, within which multiple distributed model concepts are available, which maximises the use of open earth observation data, making it the hydrological model of choice for data scarce environments.

For more information see <https://deltares.github.io/Wflow.jl/stable/>.

The WFLOW model can run using Docker (<https://hub.docker.com/r/deltares/wflow>) and to run WFLOW using Singularity it is possible to pull and build a Singularity image from DockerHub by doing ``singularity pull wflow_latest.sif docker://deltares/wflow:latest``

Static data for model building

Table 3 describes the possible raw data input sources to HydroMT-WFLOW to build a WFLOW model. Note, that for different locations the datasets may have varying degrees of accuracy.

Table 3 - Static data for model building

Name	Scale / coverage	Description
MERIT Hydro	Global	MERIT Hydro is a global hydrography dataset, developed based on the MERIT DEM and multiple inland water maps. It contains flow direction, flow accumulation, hydrologically adjusted elevations, and river channel width. http://hydro.iis.u-tokyo.ac.jp/~yamada/MERIT_Hydro/
HydroLAKES	Global	HydroLAKES aims to provide the shoreline polygons of all global lakes with a surface area of at least 10 ha. https://www.hydrosheds.org/products/hydrolakes
GRanD	Global	Global Reservoir and Dam Database (GRanD) v1.1 is a product of the Global Water System Project, which initiated a collaborative international effort to collate existing dam and reservoir datasets with the aim of providing a single, geographically explicit, and reliable database for the scientific community. https://www.globaldamwatch.org/grand
CORINE Land Cover	Europe	The CORINE Land Cover (CLC) inventory was initiated in 1985 (reference year 1990). Updates have been produced in 2000, 2006, 2012, and 2018. It consists of an inventory of land cover in 44 classes. CLC uses a Minimum Mapping Unit (MMU) of 25 hectares (ha) for aerial phenomena and a minimum width of 100 m for linear phenomena. https://land.copernicus.eu/pan-european/corine-land-cover

D7.1 Report on requirements and thematic modules definition for the environment domain

GlobCover	Global	GlobCover contains global coverage of land use, categorised in 22 classes, at a spatial resolution of 300m. Maps were produced in 2010. http://due.esrin.esa.int/page_globcover.php
VITO	Global	VITO Land Cover is a global land cover dataset, at a spatial resolution of 100m. The land cover classes are categorised in 23 classes and receive yearly updates. https://land.copernicus.eu/global/products/lc
ESA Worldcover	Global	WorldCover provides the first global land cover products for 2020 and 2021 at 10 m resolution, developed and validated in near-real time based on Sentinel-1 and Sentinel-2 data. https://esa-worldcover.org/en/data-access
SoilGrids	Global	SoilGrids™ (hereafter SoilGrids) is a system for global digital soil mapping that uses state-of-the-art machine learning methods to map the spatial distribution of soil properties across the globe. https://www.isric.org/explore/soilgrids
GLIMS	Global	GLIMS (Global Land Ice Measurements from Space) is an initiative designed to monitor the world's glaciers primarily using data from optical satellite instruments. Read about the main features of the GLIMS Glacier Database. https://www.glims.org/
Randolph Glacier Inventory	Global	Randolph Glacier Inventory contains a global overview of the glacier outlines. https://www.glims.org/RGI/
GRDC	Global	The Global Runoff Data Centre collects discharge time series for gauges around the globe. The GRDC dataset contains both the location and time series of the discharge gauges.



Dynamic data for boundary conditions

Table 4 describes the possible raw data input sources for boundary conditions to a WFLOW model. Note, that for different locations the datasets may have varying degrees of accuracy.

Table 4 - Dynamic data for boundary conditions

Name	Scale / coverage	Description
MOD15A2	Global	The Level-4 MODIS global Leaf Area Index (LAI) and Fraction of Photosynthetically Active Radiation (FPAR) product is a 8-day 1-kilometre resolution product on a Sinusoidal grid. https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/MOD15A2/#overview
ERA5	Global	ERA5 provides hourly estimates of a large number of atmospheric, land and oceanic climate variables. The data cover the Earth on a 30 km grid and resolve the atmosphere using 137 levels from the surface up to a height of 80km. ERA5 includes information about uncertainties for all variables at reduced spatial and temporal resolutions. https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5
SEAS5	Global	This dataset provides hydrological seasonal forecasts of monthly mean river discharge across Europe. Two hydrological model ensembles are provided. The first is an E-HYPE multi-model system comprising eight model realisations using a catchment-based resolution. The second comprises the E-HYPEgrid, VIC-WUR and EFAS (LISFLOOD) hydrological models at a 5km gridded resolution. https://cds.climate.copernicus.eu/cdsapp#!/dataset/sis-hydrology-variables-derived-seasonal-forecast?tab=overview
EOBS	Europe	Gridded 1km datasets with meteorological data, covering Europe. Dataset is created by interpolation of observed station data. Data is available on a daily timestep. https://www.ecad.eu/download/ensembles/download.php

CHIRPS	Global	Global precipitation dataset, available on a daily timestep, at a spatial resolution of 0.05°. https://www.chc.ucsb.edu/data/chirps
--------	--------	--

SFINCS

SFINCS (Super-Fast Inundation of CoastS) is a reduced-complexity model capable of simulating compound flooding with a high computational efficiency balanced with an adequate accuracy. In SFINCS a set of momentum and continuity equations are solved with a first order explicit scheme based on [Bates et al. \(2010\)](#). Traditionally SFINCS neglects the advection term (SFINCS-LIE) which is generally justified for sub-critical flow conditions. For supercritical flow conditions or when modelling waves, the advection term needs to be solved. For this purpose, the SFINCS-SSWE version can be used (including advection). For more information see [Leijnse et al. \(2020\)](#).

For more information see <https://sfincs.readthedocs.io/en/latest/index.html>

SFINCS can be run on Windows, Linux or by using Docker (<https://sfincs.readthedocs.io/en/latest/example.html#using-docker>) and Singularity (<https://sfincs.readthedocs.io/en/latest/example.html#using-singularity>).

Static data for model building

Table 5 describes the possible raw data input sources to HydroMT-SFINCS to build a SFINCS model. Note, that for different locations the datasets may have varying degrees of accuracy.

Table 5 - Static data for model building

Name	Scale / coverage	Description
<i>Elevation: Topography (land) and bathymetry (water)</i>		
MERIT Hydro	Global	MERIT Hydro is a global hydrography dataset, developed based on the MERIT DEM and multiple inland water maps. It contains flow direction, flow accumulation, hydrologically adjusted elevations, and river channel width http://hydro.iis.u-tokyo.ac.jp/~yamadai/MERIT_Hydro/
Copernicus	Global	The Copernicus DEM is a Digital Surface Model (DSM) that

DEM		<p>represents the surface of the Earth.</p> <p>https://spacedata.copernicus.eu/en/web/guest/collections/copernicus-digital-elevation-model</p>
GEBCO	Global	<p>GEBCO is a global terrain model for ocean and land, providing elevation data, in metres, on a 15 arc-second interval grid.</p> <p>https://www.gebco.net/data_and_products/gridded_bathymetry_data/</p>
Local elevation datasets	Regional to Local	<p>If available, local datasets are preferred over global, because:</p> <p>(1) Vertical accuracy of a Digital Elevation Model based on SRTM (e.g., MERIT Hydro and Copernicus DEM) is typically 1-10m, while a higher vertical accuracy is preferred (e.g., 10-30cm).</p> <p>(2) Horizontal resolution of global datasets is typically in the order of 1-3 s (~30-90 m), where higher horizontal resolution would be preferred (e.g., 1-10m)</p>
<i>Land Use</i>		
ESA worldcover	Global	<p>WorldCover provides the first global land cover products for 2020 and 2021 at 10 m resolution, developed and validated in near-real time based on Sentinel-1 and Sentinel-2 data.</p> <p>https://esa-worldcover.org/en/data-access</p>
CORINE Land Cover	Europe	<p>The CORINE Land Cover (CLC) inventory was initiated in 1985 (reference year 1990). Updates have been produced in 2000, 2006, 2012, and 2018. It consists of an inventory of land cover in 44 classes. CLC uses a Minimum Mapping Unit (MMU) of 25 hectares (ha) for aerial phenomena and a minimum width of 100 m for linear phenomena.</p> <p>https://land.copernicus.eu/pan-european/corine-land-cover</p>
<i>Infiltration</i>		

D7.1 Report on requirements and thematic modules definition for the environment domain

GCN250	Global	The GCN250 is a globally consistent, gridded dataset defining CNs at the 250 m spatial resolution from new global land cover (300 m) and soils data (250 m). https://figshare.com/articles/dataset/GCN250_global_curve_number_datasets_for_hydrologic_modeling_and_design/7756202
Other	Global	Curve numbers could eventually also be derived based on higher resolution land use datasets combined with soil data
<i>Structures</i>		
Structures	Local	Local data on dunes/dikes and weirs can be used to improve model performance. Also, other important line elements in the landscape, such as roads and railways might influence flow patterns.

Dynamic data for boundary conditions

Table 6 describes the possible raw data input sources for boundary conditions to a SFINCS model. Note, that for different locations the datasets may have varying degrees of accuracy.

Table 6 - Dynamic data for boundary conditions to a SFINCS model

Name	Scale / coverage	Description
Water levels	Global	Water levels are a combination of tide and surge. We have the Global Tide and Surge model (GTSM) that provides both in a 10-day forecast. Also GTSM has reanalysis data based on ERA5 conditions available at the Copernicus CDS https://cds.climate.copernicus.eu/cdsapp#!/dataset/10.24381/cds.8c59054f . Alternatively, we could also use an offshore SFINCS model, add wind and atmospheric pressure (see meteo) to compute the surge, and use the FES database to get tidal water levels at the boundary of the offshore SFINCS model.



Waves	Global	<p>There are global wave models like Wavewatch 3 that we can use, both for operational forecasting and reanalysis data. If available, local wave models can be used.</p> <p>Methods to solve the (infragravity)-waves within SFINCS are still under development, but empirical relations or look-up table metamodels can be used to already estimate wave set-up.</p>
Discharges	Global	The upstream discharges will follow from the Wflow model
<i>Meteo: wind, pressure, and precipitation</i>		
ERA5	Global	<p>ERA5 is the fifth generation ECMWF reanalysis for the global climate and weather for the past 8 decades. Data is available from 1940 onwards.</p> <p>https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview</p>
GFS	Global	<p>The Global Forecast System (GFS) is a National Centers for Environmental Prediction (NCEP) weather forecast model that generates data for dozens of atmospheric and land-soil variables, including atmospheric pressure, winds, and precipitation.</p> <p>https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast</p>

Delft-FIAT

Delft-FIAT builds upon the general concept of risk-based decision making, which combines information about hazard, exposure, and vulnerability (**Figure 10**). Delft-FIAT uses water depth input and user-specified depth-damage curves to rapidly evaluate and output damages per asset, such as buildings and utilities, or per grid, and aggregates damages to user-specified aggregation scales (**Figure 11**). Delft-FIAT has been used in dozens of projects worldwide to help prioritise investments by quantifying the benefits of flood mitigation and adaptation measures.

$$\text{Risk} = \text{Probability} \cdot \text{Consequences}$$

$$\text{Risk} = \text{Hazard} \cdot \text{Exposure} \cdot \text{Vulnerability}$$

Figure 10 - Definition of risk (Kron, 2005).

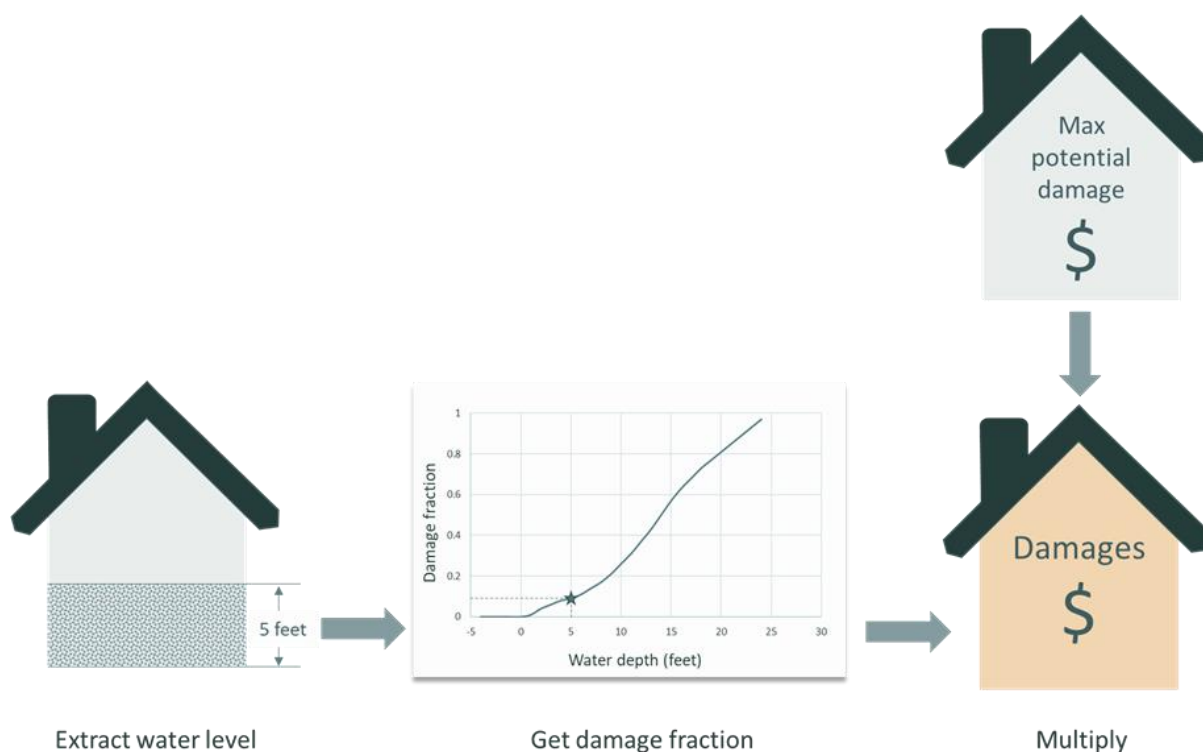


Figure 11 - High-level workflow of Delft-FIAT.

For each object, the water level is extracted within the building, the damage fraction is determined using damage functions, and the combination of the damage fraction and the maximum potential damages returns the damage to the building.

Inputs to Delft-FIAT are flood maps, depth-damage functions, and locations and maximum potential damages of assets. The calculation engine in Delft-FIAT uses the input data to derive aggregated and asset-level damages and risk. For each asset specified in the exposure dataset, the water depth or elevation is subtracted from the flood map at the location of the assets; water elevations are converted to water depths using the ground elevation of each asset. When calculating partial flooding, Delft-FIAT will extract either the average or maximum water depth and the fraction of the building that is flooded. The inundation depth within buildings is obtained by subtracting from the water depth the ground floor height. Delft-FIAT derives the damage fraction for each asset using its inundation depth and interpolating over its depth-damage curve. The damage to the asset is then calculated as the product of the maximum potential damage and the damage fraction. When calculating partial flooding, the damages will be reduced by the



fraction of the building that is dry. When the user inputs return-period flood maps, Delft-FIAT will calculate the associated return-period damages, and then integrate these to derive the expected annual damages (EAD).

To create a Delft-FIAT model, the data should be in the required format. The section ‘Static data for model building’ elaborates on the data that is required to create a Delft-FIAT model with the HydroMT-FIAT plugin (see [Section 2.3.3.4](#)). In the section ‘Model data format’, the result of the HydroMT-FIAT plugin - the Delft-FIAT model schematization - is described.

Static data for model building

Table 7 describes the data input sources that can be used to build a Delft-FIAT model. Options are suggested for different scales and coverages.

Table 7 - Data input sources to HydroMT-FIAT to build a Delft-FIAT model.

Name	Scale coverage	Description
<i>Exposure</i>		
Open Street Maps	Global (but not uniform)	Building footprints, roads, and possibly asset classification: https://www.openstreetmap.org/#map=7/52.154/5.295
Google Buildings	Open Africa, South Asia, and South-East Asia	Buildings footprints and buildings points: https://sites.research.google/open-buildings/#download
Microsoft GlobalMLBuildingFootprints	Parts of North, Middle and South America, Europe, Africa, Asia, SIDS, and more	Building footprints: https://github.com/microsoft/GlobalMLBuildingFootprints
Other building footprints, building locations, road infrastructure	From global to local	Building footprints, building points, and road infrastructure locations.
Other asset	From global to	(Spatial) data that can be used to classify

D7.1 Report on requirements and thematic modules definition for the environment domain

classification data	local	the building footprints or points to align with the damage function categories (e.g. residential, commercial, industrial). This can also be data about the construction material of the assets for buildings or roads.
<i>Vulnerability</i>		
JRC Depth Damage Curves	Global	A globally consistent database of depth-damage curves and maximum damage values for residential buildings, commercial buildings, industrial buildings, transport, road infrastructure, and agriculture. The curves are available for each continent and the damage values are available for almost all countries globally. https://publications.jrc.ec.europa.eu/repository/handle/111111111/45730
Other damage functions	From global to local	Depth-damage curves relating water depth to a damage fraction for a type of asset classification (e.g., residential, and commercial buildings, and roads).
<i>Hazard</i>		
Water depth map	From global to local	A NetCDF water depth map produced by SFINCS.
Water elevation map	From global to local	A NetCDF water elevation map produced by SFINCS.
<i>Configuration</i>		
HydroMT-FIAT configuration file	N/A (not spatial)	An yml file containing the configuration settings for setting up a Delft-FIAT model from raw data sources.



Region	From global to local	The area of interest defined by the Digital Twin developer / implementer.
--------	----------------------	---

Model data format

An overview of the Delft-FIAT data requirements is given in Table 8. This is the output from Hydro-MT FIAT.

Table 8 - Delft-FIAT model data description.

Name	Data type	Description
Exposure database	CSV	This file contains the required exposure information to calculate damages and optionally risk, like the <i>maximum potential damage value</i> and associated <i>depth-damage function</i> , for each asset in the area of interest. The file can contain point locations (X, Y coordinates) or the information in the file can be linked to the geometries and locations in an asset locations/building footprints Geopackage.
Asset building footprints	Geopackage	This is an <i>optional</i> file for when the user would like to link building footprints to their exposure database. It contains a linking ID and geometries of the building footprints.
Damage function database	CSV	This file contains the water depths related to damage fraction per asset category, i.e., the depth-damage functions. The columns contain the names of the damage functions, which are linked to the assets in the Exposure database CSV.
Flood map(s)	NetCDF	One (for an assessing event damages) or multiple probabilistic return period (for assessing risk/EAD) flood maps including the required metadata.
Delft-FIAT configuration file	toml	This file contains the configuration settings for running a Delft-FIAT model.

The flood maps can be water elevation or water depth raster files. The depth-damage function database file contains for each damage function the water depths and the associated fraction of total damage. The exposure data contains the location, maximum potential damage value, ground floor height, ground elevation, and associated depth-damage function for each asset in the area of interest. It can optionally also contain aggregation labels (like neighbourhood or district), aggregation variables (like a social vulnerability index), and a shapefile/gpkg location and “join ID” to connect the asset to an aerial (building) footprint. This latter allows the user to calculate the damages for partial flooding in large buildings, like shopping malls.

RA2CE

RA2CE (phonetically pronounced: race) stands for Resilience Assessment and Adaptation for Critical infrastructurE and is a Python-based model developed by Deltares. It can be used to quantify resilience of critical infrastructure networks, prioritise interventions and adaptation measures and select the most appropriate action perspective to increase resilience considering future conditions. RA2CE is globally applicable and has been applied for resilient investment planning in several regions such as the Netherlands, Philippines, Myanmar, Dominican Republic, and Albania.

RA2CE has been developed by Deltares to produce risk and resilience maps specifically focused on infrastructure networks. RA2CE uses different sources of information and just like Delft-FIAT, builds upon the general concept of risk-based decision making, which is characterised by all components of risk: hazard, exposure, and vulnerability. Exposure is defined as assets that may be at risk from potential (natural) hazards. Vulnerability is related to the physical damage of the infrastructure and used to identify how much damage to the road this will cause. In addition, in RA2CE we also include the cascading effects on society due to disruption of the infrastructure network (**Figure 12**).

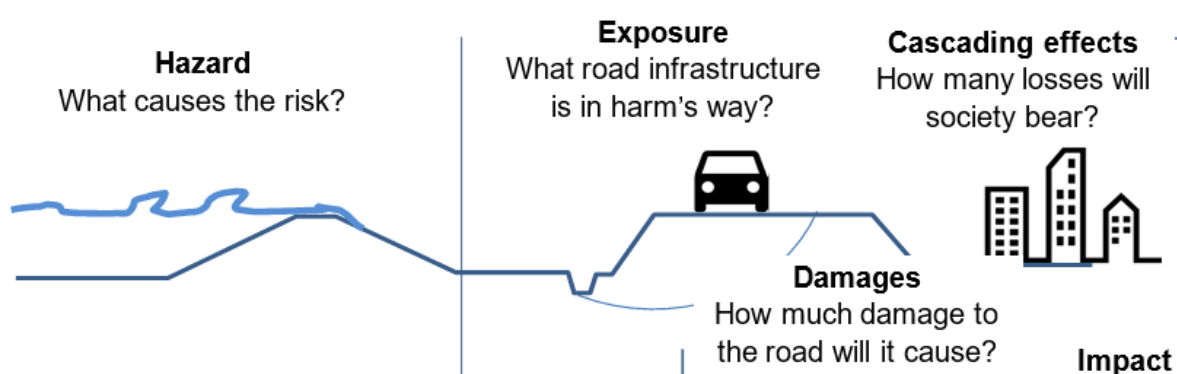


Figure 12 - The context of risk analysis for road infrastructure (Bles et. al 2019).

Table 9 describes the data input sources that can be used to build a RA2CE model with the to-be-developed Hydro-MT RA2CE plugin. Options are suggested for different scales and coverages.

Static data for model building

Table 9 - Potential data input sources to HydroMT-RA2CE (to be developed) to build a RA2CE model.

Name	Scale / coverage	Description
<i>Exposure</i>		
OSM	Global	Road and other infrastructure networks, this can be directly converted into graphs which is the format needed for the accessibility assessments in RA2CE. Points of interest (e.g., hospitals, education locations) can also be obtained from this source.
WorldPop	Global	Population maps that can be used to generate origin points with a population count.
Other roads	From global to local	Vector GIS line data that describes the road network. It should meet certain requirements, such as: the lines should be cut at the intersections, not contain duplicate lines, not contain any dangles, and should be fully connected where the road network is connected in reality.
Other points of interest	From global to local	GIS point data with important locations for routing analysis and/or isolation analysis.
<i>Vulnerability</i>		
European road damage functions	Europe	Depth-damage curves developed for the European road network with OSM road typologies (van Ginkel et. al, 2021).
JRC Depth Damage Curves	Global	A globally consistent database of depth-damage curves and maximum damage



D7.1 Report on requirements and thematic modules definition for the environment domain

		values for residential buildings, commercial buildings, industrial buildings, transport, road infrastructure, and agriculture. The curves are available for each continent and the damage values are available for almost all countries globally. https://publications.jrc.ec.europa.eu/repository/handle/111111111/45730
--	--	--

Model input data

An overview of the RA2CE data requirements is given in Table 10. This will be output from the to-be-developed Hydro-MT RA2CE.

Table 10 - RA2CE model data description.

Name	Data type	Description
<i>Required</i>		
Road network	GeoPackage, shapefile, or pickle (*.p)	A fully connected and clean road network. Ideally this file is a graph saved in pickle format because the RA2CE model can then immediately start the analysis.
<i>Optional</i>		
Flood map(s)	GeoTIFF	One (for an assessing event damages) or multiple probabilistic return period (for assessing risk/EAD) flood maps including the required metadata.
Origins / destinations / points of interest	Shapefiles or Geopackages	Or multiple GIS files describing the locations, category and potentially the number of people per point of interest.



2.3.3 Components

HydroMT-Core

HydroMT-Core (<https://deltares.github.io/hydromt/latest/>) is an open-source Python package that facilitates the process of building and analysing spatial geoscientific models with a focus on water system models. It does so by automating the workflow to go from raw data to a complete model instance which is ready to run and to analyse model results once the simulation has finished.

HydroMT-Core is commonly used in combination with a model plugin which provides a HydroMT implementation for a specific model software. Using the plugins allows users to prepare a ready-to-run set of input files from raw geoscientific datasets and analyse model results in a fast and reproducible way. HydroMT plugins will be used to build models for WFLOW, SFINCS, Delft-FIAT and RA2CE. The following paragraphs elaborate on these plugins.

Dependencies:

- Python 3.8 or greater
- Package manager such as conda or mamba, including the conda-forge channel
- Libraries:
 - pandas
 - numpy
 - xarray
 - dask
 - scipy
 - geopandas
 - pyproj
 - rasterio
 - pyflwdir
 - click

HydroMT-WFLOW

Setting up distributed hydrological models typically requires many (manual) steps to process input data and might therefore be time consuming and hard to reproduce. Especially improving models based on global-local geospatial datasets, which are rapidly becoming available at increasingly high resolutions, might be challenging. HydroMT-WFLOW aims to make the WFLOW model building and updating processes fast, modular, and reproducible and to facilitate the analysis of the model results.

The HydroMT-WFLOW plugin can be used as a command line application, which provides commands to build, update and clip a WFLOW model with a single line, or from python to exploit its rich interface. You can learn more about how to use HydroMT-WFLOW in its [online documentation](#).

For a smooth installation experience we recommend installing HydroMT-Wflow and its dependencies from conda-forge in a clean environment, see [installation guide](#).

Additional plugin dependencies:

- toml
- pcraster (optional): <https://pcraster.geo.uu.nl/>
- gwwapi (optional): <https://github.com/global-water-watch/gww-api>
- hydroengine (optional): <https://github.com/openearth/hydro-engine>

HydroMT-SFINCS

Setting up hydrodynamic models typically requires many (manual) steps to process input data and might therefore be time consuming and hard to reproduce. Especially improving models based on global geospatial datasets, which are rapidly becoming available at increasingly high resolutions, might be challenging. Furthermore, analysing a SFINCS model schematization which uses model-specific binary data formats, can be time consuming. HydroMT-SFINCS aims to make the model building process fast, modular, and reproducible and to facilitate the analysis of SFINCS model results.

The HydroMT-SFINCS plugin can be used as a command line application, which provides commands to build, update the SFINCS model with a single line, or from python to exploit its rich interface. You can learn more about how to use HydroMT-SFINCS in its [online documentation](#). For a smooth installation experience we recommend installing HydroMT-SFINCS and its dependencies from conda-forge in a clean environment, see [installation guide](#).

Additional plugin dependencies:

- No additional dependencies according to https://deltares.github.io/hydromt_sfincs/latest/getting_started/installation.html#installation-guide.

HydroMT-FIAT

This [plugin](#) provides an implementation for the FIAT flood impact assessment tool. It details the different steps and explains how to use HydroMT to easily get started and work on your own fiat model.

With the `hydromt_fiat` plugin, users can easily benefit from the rich set of tools of the HydroMT package to build and update Delft-FIAT models from available global and local data.

This plugin assists the fiat modeller in:

- Quickly setting up a Delft-FIAT model based on existing hazard maps, global exposure layers and a database of vulnerability curves.
- Making maximum use of the best available global or local data.
- Adjusting and updating components of a fiat model and their associated parameters in a consistent way.

Additional plugin dependencies:

- HydroMT-FIAT plugin is currently only available from PyPi. We are working on a release from conda-forge.

HydroMT-RA2CE

The plan is to develop the RA2CE plugin in interTwin. With the input data mentioned in [section 2.3.2.4](#), a road network will be potentially cleaned, prepared, and potentially populated with additional data on for example points of interest, number of people per origin point, or segmented into 100-metre segments for assessing direct road damage.

Pre processing model input data

Preprocessing boundary condition data for compound flood modelling typically involves the following steps:

1. Data quality assessment: Review the collected data to identify any gaps, inconsistencies, or errors. This step may involve cross-referencing multiple data sources, interpolating missing values, and identifying outliers.
2. Coordinate system conversion: Ensure that all spatial data is in a consistent coordinate system, such as a local or global coordinate reference system. This may involve reprojecting the data and confirming that units are consistent (e.g., metres or feet).
3. Data interpolation and downscaling: Interpolate and downscale the data to a common spatial resolution appropriate for the modelling domain. This step may involve resampling elevation data or disaggregating meteorological data using interpolation techniques such as bilinear interpolation or nearest-neighbour methods.
4. Data preprocessing for specific flood drivers:
 - a. For pluvial (rainfall-driven) floods, preprocess rainfall data by calculating rainfall intensity-duration-frequency (IDF) curves for the study area.
 - b. For fluvial (riverine) floods, preprocess river discharge data by extracting flow and stage data and develop rating curves that relate river discharge to water levels.

- c. For coastal floods, preprocess tide and storm surge data by analysing extreme sea level events and their return periods.
5. Boundary condition definition: Define the boundary conditions for the model domain, such as inflow and outflow points, no-flow boundaries, and initial water levels.
6. Data integration: Combine the preprocessed data for each flood driver into a single dataset or model input file. This step may involve creating a joint probability distribution of the different flood drivers or assigning weights to each driver based on their relative contributions to the compound flood risk.
7. Model calibration and validation: Use a subset of the preprocessed data to calibrate and validate the compound flood model, adjusting model parameters as needed to minimise the difference between observed and simulated flood events.
8. Scenario development: Create a set of future scenarios or design events for the compound flood model, which may include changes in climate, land use, or flood management strategies.

Table 11 provides links to available tools for each model to do the above tasks.

Table 11 - Available preprocessing tools for each model.

WFLOW	<p>All preprocessing of the boundary condition data and preparation of the required model file is taken care of by HydroMT-WFLOW. The typical steps within HydroMT-WFLOW include:</p> <ul style="list-style-type: none"> - Basin delineation based on user-defined region of interest and model resolution (reprojecting and regriding of data occurs in all steps) - River network derivation - Converting glacier, lake, and reservoirs data into corresponding WFLOW configurations - Convert land use classes and soil data to WFLOW model parameters using mapping tables (provided by HydroMT-WFLOW) - Calculate potential evaporation from dynamic data and create WFLOW forcing file together with precipitation and temperature data.
SFINCS	<p>In principle, all preprocessing of the SFINCS model is taken care of by HydroMT-SFINCS. The typical steps within HydroMT-SFINCS include:</p> <ul style="list-style-type: none"> - Interpolate gridded spatial data like topography/bathymetry, roughness and infiltration on a model grid covering the area of interest.

	<ul style="list-style-type: none"> - Adding boundary conditions to the SFINCS model: <ul style="list-style-type: none"> - Discharge from Wflow - Offshore water levels (e.g., from GTSM) - Meteo forcing (e.g., from GFS) <p>https://github.com/Deltares/hydromt_sfincs/blob/main/hydromt_sfincs/sfincs.py</p> <p>These scripts use Xarray for raster data and geopandas for vector data. Depending on the source, scripts have to be adjusted.</p>
Delft-FIAT	Water levels computed by SFINCS, stored in th sfincs_map.nc, are interpolated onto the underlying higher-resolution DEM. This results in a water level map with flood depths in geotiff format. These are used as input in Delft-FIAT.
RA2CE	The HydroMT-RA2CE plugin will convert all data in the correct format.

Running the models

Table 12 -Available run options for each model

Model / tool	Run options and URLs
HYDRO MT	<ul style="list-style-type: none"> • Command Line Interface: a high-level interface to HydroMT. It is used to run HydroMT methods such as build, update or clip. • Python Interface: offers more flexibility for advanced users. It allows you to e.g. interact directly with a model component Model API and apply the many methods and workflows available. Please find all available functions API reference
WFLOW	<ul style="list-style-type: none"> • Docker: https://hub.docker.com/r/deltares/wflow • Singularity: <code>singularity pull docker://deltares/wflow:latest</code>
SFINCS	<ul style="list-style-type: none"> • Windows: https://sfincs.readthedocs.io/en/latest/example.html#on-windows-standard

	<ul style="list-style-type: none"> Linux: https://sfincs.readthedocs.io/en/latest/example.html#on-linux Docker: https://sfincs.readthedocs.io/en/latest/example.html#using-docker Singularity: https://sfincs.readthedocs.io/en/latest/example.html#using-singularity
Delft -FIAT	<ul style="list-style-type: none"> Command Line Interface: a high-level interface to Delft-FIAT. It is used to run Delft-FIAT with user-configured input. Currently the GitHub repository is private but will be made public during the duration of the project: https://github.com/Deltares/Delft-FIAT.
RA2CE	<ul style="list-style-type: none"> Command Line Interface: a high-level interface to RA2CE. It is used to run RA2CE with user-configured input. Currently the GitHub repository is private but will be made public during the duration of the project: https://github.com/Deltares/RA2CE.

Post processing the model output data

Postprocessing the output data from compound flood models involves converting the raw model results into formats that can be easily visualised, interpreted, and used for decision-making. Typical steps for post processing compound flood model data include:

1. Data extraction: Extract relevant output variables from the model results, such as water levels, flow velocities, inundation extents, and flood depths.
2. Data aggregation: Aggregate the extracted data over relevant spatial and temporal scales, depending on the intended visualisation or decision-making context. For example, you might calculate average or peak flood depths over specific time periods or within particular geographic areas.
3. Data validation: Compare the model outputs to observed flood data (e.g., historical flood events, gauging station records) to validate the model's performance and ensure that the post processed data is reliable and accurate.
4. Derived variables calculation: Calculate derived variables that can help convey the impacts of compound flooding more effectively, such as flood return periods, flood duration, or economic and social impacts (e.g., damages, affected population).
5. Data transformation: Transform the data into a format suitable for visualisation, such as raster grids, vector polygons, or point datasets. This may involve converting continuous data (e.g., water depths) into discrete classes or categories

D7.1 Report on requirements and thematic modules definition for the environment domain

based on thresholds relevant to decision-making (e.g., minor, moderate, and severe flooding).

6. Geospatial analysis: Perform geospatial analysis of the post processed data, such as overlaying flood extents with land use, infrastructure, or population data to identify areas at highest risk or most vulnerable to compound flooding.
7. Visualisation creation: Create visualisations that effectively communicate the results of the compound flood model, such as static or interactive maps, charts, and graphs. These visualisations should be clear, easy to interpret, and tailored to the needs of the target audience (e.g., decision-makers, stakeholders, or the general public).
8. Uncertainty assessment: Quantify and communicate the uncertainty associated with the compound flood model outputs and the post processed data. This may involve calculating confidence intervals, displaying uncertainty bands in visualisations, or discussing the limitations and assumptions of the model.
9. Decision support: Present the post processed data and visualisations to decision-makers, stakeholders, and other relevant parties to inform decisions related to flood risk management, land use planning, emergency response, and mitigation measures. Encourage discussion and feedback to ensure that the compound flood model results are accurately interpreted and appropriately used in decision-making processes.

Table 13 provides links to available tools and frameworks used to do the above tasks.

Table 13 - Available postprocessing tools for each model.

WFLOW	WFLOW can produce time series and gridded output data. The time series can be either written as a comma separated value (csv) file, or as NetCDF. The gridded output data is always written as a NetCDF file. For typical post processing steps, the python libraries pandas (for csv files), Xarray (for NetCDF files) and matplotlib (for plotting) are used.
SFINCS	SFINCS produces gridded map data and time series, both in NetCDF format. The maximum water levels in the map data are downscaled to floodmaps on the resolution of your input DEM using hydromt-sfincs scripts; <code>downscale_floodmap</code> and <code>read_sfincs_map_results</code> in https://github.com/Deltares/hydromt_sfincs/blob/main/hydromt_sfincs/utils.py
Delft-FIAT	Delft-FIAT produces per-asset and aggregated tables and geospatial data, in CSV, excel, and geopackage format which can directly be loaded into GIS software.

RA2CE	RA2CE produces per-road segment and aggregated tables and geospatial data in CSV and geopackage format which can directly be loaded into GIS software.
-------	--

2.3.4 User interaction

We anticipate two types of users:

1. A researcher or developer that builds and develops digital twins using components of the Digital Twin Engine.
2. A decision maker that uses the digital twin to make a decision.

In the below two sections we describe the anticipated user interaction for both types of users.

Digital Twin developer / implementer

For a user that builds a digital twin using components of the Digital Twin Engine on a federated computing and data infrastructure using command line interfaces, the interaction process can be broken down into the following steps:

1. **Accessing the Federated Infrastructure:** Initially, users can log into the federated computing and data infrastructure using the appropriate credentials through a command line interface (CLI), such as SSH. However, at the end of this project, access will be primarily through a Workflow Engine GUI, or a similar module designed for developers, thereby obscuring direct access to individual infrastructure nodes while still maintaining functionality..
2. **Retrieve the Virtual Machine Image from the Digital Twin Engine:** Users can pull pre-packaged virtual machine (VM) images from an image repository. This VM image includes all the necessary components, models, and tools required for the digital twin, ensuring they have the most recent and complete version of the platform.
3. **Configure the Digital Twin:** The user customises the configuration files for the compound flood modelling digital twin, including specifying data sources, model parameters, and output formats. This may involve editing text-based configuration files, JSON files, or other formats supported by the Digital Twin Engine.
4. **Integrate application-specific components:** The user integrates any application-specific digital twin components, such as custom hydrodynamic models or data preprocessing scripts, by adding them to the appropriate directories and updating the configuration files as needed.
5. Via the workflow manager, the following steps will be executed sequentially:
 - a. **Load required modules and dependencies:** The user loads any necessary modules or libraries required for the Digital Twin Engine, such as

programming languages (e.g., Python, R, or Julia), geospatial libraries (e.g., GDAL, Proj), or specific Digital Twin Engine modules.

- b. **Execute the Digital Twin:** Using the command line interface, the user executes the Digital Twin with the specified configuration, running the compound flood modelling digital twin. This may involve running a series of scripts or executing a single command, depending on the design of the Digital Twin Engine.
 - c. **Monitor progress and troubleshoot:** The user monitors the progress of the digital twin execution through the command line interface, examining log files and console output for any errors or warnings. If issues arise, the user may need to modify the configuration files, update the application-specific components, or adjust the federated infrastructure resources to resolve the problem.
6. **Validate the digital twin output:** After the digital twin has completed execution, the user validates the output data to ensure the compound flood modelling results are accurate and consistent with the intended design. This may involve comparing the digital twin outputs to observed data, performing sensitivity analyses, or examining summary statistics.
 7. **Version control and documentation:** The user maintains version control for the Digital Twin Engine and its components, using a system such as Git, and updates any relevant documentation to reflect the implemented changes and improvements.
 8. **Share and collaborate:** The user shares the digital twin with other researchers, stakeholders, or collaborators by providing access to the federated infrastructure, distributing the configuration files, or sharing the digital twin components through an online repository.

With the above functionality, a user can effectively build a digital twin for compound flood modelling and impact assessment using the Digital Twin Engine on a federated computing and data infrastructure with command line interfaces. The process involves configuring, executing, and validating the digital twin, while also ensuring efficient collaboration and version control.

Digital Twin end-user (e.g. decision maker)

For a user who uses the output from a digital twin to make decisions, the interaction process can involve accessing online dashboards or developing custom analytics in Jupyter Notebooks. The intent is to enable a decision-maker to effectively use the output from a digital twin to make informed decisions, either by exploring pre-built visualisations and data analytics through an online dashboard or by developing custom analyses in Jupyter Notebooks. Below we summarise possible user interactions for this user:

1. **Access the online dashboard or data portal:** The user logs into the online dashboard or data portal hosting the digital twin output data. The user may need to enter their credentials to access the platform and its features.

2. **Explore pre-built visualisations:** The user can explore pre-built visualisations, such as maps, charts, and graphs, that display the digital twin output data. These visualisations can be interactive, allowing the user to filter data, change display settings, or select specific regions or time periods to view.
3. **Customise dashboard views:** The user can customise the dashboard to display specific metrics or indicators relevant to their decision-making needs. This may involve selecting different data layers, adjusting colour schemes, or reorganising the layout of the dashboard.
4. **Access and download data:** The user can access and download the raw digital twin output data or derived datasets in various formats, such as CSV, JSON, or GeoJSON. This data can be used for further analysis, reporting, or sharing with other stakeholders.
5. **Develop custom analytics in Jupyter Notebooks:** The user can create, or access Jupyter Notebooks hosted within the dashboard platform or in a separate environment, such as a local machine or a cloud based JupyterHub.
 - a. **Import data:** The user imports the digital twin output data into the Jupyter Notebook, using appropriate data manipulation libraries (e.g., Pandas for Python).
 - b. **Data preprocessing:** The user preprocesses the data as needed, such as filtering, aggregating, or transforming the data to suit their specific analytical requirements.
 - c. **Develop custom analyses:** The user develops custom analytical scripts to process the digital twin output data, using relevant programming languages (e.g., Python or R) and libraries (e.g., NumPy, SciPy, or matplotlib).
 - d. **Visualise results:** The user creates visualisations within the Jupyter Notebook to display the results of their custom analyses, using libraries such as Matplotlib, Seaborn, or Plotly.
6. **Decision-making:** The user uses the insights gained from the dashboard visualisations and custom Jupyter Notebook analyses to inform their decisions related to compound flood risk management, land use planning, emergency response, or mitigation measures.
7. **Provide feedback:** The user can provide feedback on the digital twin output data, visualisations, or analyses through the dashboard platform or directly to the digital twin developers, helping improve the accuracy and relevance of the digital twin for future decision-making.

3 Requirements for the thematic modules in the environment domain

Section 3 is dedicated to reporting the requirements concerning the thematic modules to be developed for the environment domain thematic modules. The three environment domain thematic modules are Climate analytics and data processing (T7.4), Earth Observation Modelling and Processing (T7.5) and Hydrological model data processing (T7.6).

Thematic modules requirements consolidation resulted from the activities performed so far under work package 4 (WP4), which concerns the technical co-design and validation of the digital twin engine among research communities. Under this scope the objective was to introduce use-case specific requirements for the thematic modules, based on the DTE infrastructure (WP5) and core modules (WP6). Digital twin engine core modules described in work package 6 are responsible for capabilities concerning workflow compositions, real-time acquisition of data and processing, quality and uncertainty tracing, data fusion, big data analytics, as well as AI/ML workflow.

3.1 General Description and Categorization of requirements

With respect to the DTE core modules and after research studies and analysis, institute, and community wise, the environment domain thematic modules requirements were gathered, agreeing that use cases present similar processing operations throughout their workflows. If a requirement category is not applicable to any of the use case's, this will be specifically mentioned in the respective subsection. Each requirement category is represented by a dedicated subsection following the description. The requirement categories compiling use cases' capabilities components are introduced as follows:

- **Input and output storage:** this subsection describes the input and output data requirements concerning data storage architecture utilised, HPC centres where data are available and stored, and any pre-processing methods and steps required. Moreover, the expected data volumes will be reported with respect to both input and output data.
- **Databases subsection** includes the form of databases and the database management systems that are being utilised for storage of data and metadata during use case processes.
- **Computing** is a general term that we use here to describe all the requirements related to computation resources in terms of CPUs and/or GPUs. This subsection describes the computing set up that each use case has been provisioning for each digital twin, from local computation resources to cloud and from High Performance Computing to High Throughput Computing and MPI infrastructure.

- OS and execution framework includes the DT requirements for the operating system and the OS-level virtualization framework for delivering DT software.
- Machine Learning subsection includes requirements concerning the exploitation of machine learning by each environment use case, in terms of software development language, ML frameworks, machine/deep learning models, statistical learning models, monitoring, (re)training and validation.
- Real-time data acquisition and processing refers to the use cases' capability of data/metadata being processed in real-time. This subsection also includes the platforms/frameworks being used for that purpose and how real-time processing is approached by the concerned use cases.
- Data formats subsection describes the different formats that data and metadata coming from the environment use cases present. As well as, to which formats are the original data being converted to in case of pre-processing and post-processing.
- The subsection of workflow tools describes all the workflow software stack requirements for each digital twin thematic module.
- Visualisation subsection includes the different forms and methods of visualising the results in terms of quality verification and validation.
- In the data sharing subsection, the processes of making data resources available are presented.

3.2 Storage I/O

Support for distributed file systems and object stores.

High-performance disk I/O for storing and retrieving large volumes of data.

Support for data compression and decompression to reduce storage costs and improve I/O performance.

3.2.1 Input data requirements

Global datasets required:

- Sentinel-1 GRD data from 2016 onwards
- HydroLAKES
- GRanD
- GlobCover
- VITO Land Cover
- SoilGrids
- GLIMS
- Randolph Glacier Inventory
- GRDC

- MOD15A2
- ERA5
- SEAS5
- CHIRPS
- MERIT Hydro
- Copernicus DEM
- GEBCO
- GCN250
- ESA worldcover
- GFS
- Water levels
- Waves
- OpenStreetMap (OSM)
- WorldPop
- JRC Depth Damage Curves
- CMIP6
- IBTrACS

European datasets required:

- EOBS
- CORINE Land Cover
- European road damage functions.

Other areas:

- Google Open Buildings for Africa, South Asia, and South-East Asia.
- Microsoft GlobalMLBuildingFootprints for parts of north, middle and south America, Europe, Africa, Asia, SIDS, and more.

3.3 Databases

- Support for databases, such as PostgreSQL (with PostGIS).
- STAC catalogue for raster datasets.

3.4 Computing

- Support for distributed computing and parallel processing to handle large-scale datasets.

- Availability of GPUs for machine learning workloads.
- Ability to scale up or down computing resources based on workload demands.

3.5 OS and execution framework

- Linux operating system with command line interface.
- Integration with containerization platforms, such as Docker and Singularity, for efficient deployment and management of applications.
- Python 3.8 or greater, with a package manager such as conda, mamba and/or pip.

3.6 Machine Learning

- Support multiple machine learning frameworks, such as TensorFlow (jointly with higher-level Python libraries such as Keras) and PyTorch.
- Integration with data processing and visualisation tools for data preparation and model evaluation (e.g. TensorBoard).
- Support for model versioning and management.

3.7 Real-time data acquisition and processing

Low-latency data processing capabilities for near real-time analytics and decision-making. Data acquisition and processing could be handled by openEO. There should be an automatic trigger of the openEO workflow based on it.

3.8 Data formats

Support for multiple data formats, such as CSV, JSON, geoJSON, GeoPackage, shapefile, pickle (*.p), GeoTIFF, NetCDF.

3.9 Workflow tools

- Access to an openEO back-end, for generating and managing EO workflows.
- Ophidia back-end for managing multi-model workflows.
- Support for version control and collaboration tools, such as Git and Jupyter Notebooks, for code sharing and reproducibility.
- Compatibility with continuous integration/continuous deployment (CI/CD) tools for efficient deployment and testing of applications.
- Ability to monitor and track workflow progress and performance, using a log management system.

3.10 Visualisation

- Compatibility with open-source visualisation libraries, such as matplotlib and Plotly, for customised visualisations, as well as cartopy for maps production
- Support for interactive data visualisations and dashboards for real-time monitoring and decision-making, with the ability to create customised dashboards and reports.

3.11 Data Sharing

Support for metadata management tools for tracking data lineage and ensuring data quality. This can be achieved with the support of data sharing as STAC items.

Ability to share data across different teams and organisations securely (signed URLs).

4 Conclusions

The thematic modules for the Digital Twins concerning the environmental domain have been developed following the initial design of WP4 during this first period of the project. The researchers analysed step by step the possible interactions between multiple components, summarising them in schematic views following the C4 design model principles, so that the harmonisation among the different models is as its best.

From a common evaluation of the different applications, a set of common minimum requirements has been defined, differentiating them among several categories. This will be necessary for WP6, when defining and implementing the DTE core components.

5 References

Reference	
No	Description / Link
R1	The openEO API–Harmonising the Use of Earth Observation Cloud Services Using Virtual Data Cube Functionalities. M. Schramm et al. In Remote Sensing, vol 13(6), p 1125, 2021, DOI: 10.3390/rs13061125
R2	Flood Risk = Hazard • Values • Vulnerability. Kron, W., 2005. International Water Resources Association, Water International, 30(1), 58-68. March 2005.
R3	Progressing road infrastructure resilience from different institutional development perspectives. Bles, T., Costa, A. L., Hüsken, L., Woning, M., Espinet, X., van Muiswinkel, K., & Page, S. (2019). In 26th World Road CongressWorld Road Association (PIARC).
R4	Modeling compound flooding in coastal systems using a computationally efficient reduced-physics solver: Including fluvial, pluvial, tidal, wind- and wave-driven processes. Tim Leijnse, Maarten van Ormondt, Kees Nederhoff, Ap van Dongeren, Coastal Engineering, Volume 163, 2021, 103796, ISSN 0378-3839, DOI: 10.1016/j.coastaleng.2020.103796
R5	A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. Paul D. Bates, Matthew S. Horritt, Timothy J. Fewtrell, Journal of Hydrology, Volume 387, Issues 1–2, 2010, Pages 33-45, ISSN 0022-1694, DOI: 10.1016/j.jhydrol.2010.03.027
R6	Flood risk assessment of the European road network. van Ginkel, K.C.H., Dottori, F., Alfieri, L., Feyen, L., Koks, E.E., 2021. Nat. Hazards Earth Syst. Sci. 21, 1011–1027. DOI: 10.5194/nhess-21-1011-2021