D4.5 Final Architecture design of the DTs capabilities for climate change and impact decision support tools

Status: Under EC Review
Dissemination Level: public

Funded by the European Union

Abstract

Key Words	Architecture, design, capabilities, Digital Twin Engine, climate change, impact decision, support tools

This deliverable describes the final architecture design of the digital twin application for climate change and impact decision support tools. For each application it depicts their main capabilities and requirements, as well as the different steps included in their workflows. It highlights, thus, the needs of the climate change use cases from the interTwin Digital Twin Engine (DTE). To this end, links with the different components from the DTE are also presented in the document and showcased in workflow diagrams.

Document Description

D4.5 Final Architecture design of the DTs capabilities for climate change and impact decision support tools			
Work Package number 4			
Document type	Deliverable		
Document status	UNDER EC REVIEW	**Version**	1.0
Dissemination Level	Public		
Copyright Status	This material by Parties of the interTwin Consortium is licensed under a Creative Commons Attribution 4.0 International License.		
Lead Partner	CMCC		
Document link	https://documents.egi.eu/document/3941		
DOI	https://zenodo.org/records/15096734		
Author(s)	• Donatello Elia (CMCC) • Davide Donno (CMCC) • Emanuele Donno (CMCC) • Iacopo Ferrario (EURAC) • Massimiliano Fronza (UNITN) • Björn Backeberg (DELTARES) • Willem Tromp (DELTARES) • Christian Pagé (CERFACS)		
Reviewers	• Mohammad Hussein Alasawedah (WWU) • Zdeněk Šustr (CESNET)		
Moderated by:	• Donatello Elia (CMCC) • Andrea Cristofori (EGI) • Andrea Anzanello (EGI)		
Approved by	• Germán Moltó (UPV) on behalf of TCB		

Revision History

Version	Date	Description	Contributors
V0.1	05/12/2024	Template creation from the 1st architecture design	Andrea Cristofori (EGI)
V0.2	03/02/2025	ToC updated and sections assigned	Donatello Elia (CMCC)
V0.3	07/03/2025	All main sections updated	All contributors
V0.4	24/03/2025	Internal revision provided and comments addressed	Reviewers & contributors
V0.5	26/03/2025	TCB review	Germán Moltó
V1.0	28/03/2025	**Final**	

Terminology / Acronyms

Term/Acronym	Definition
AI	Artificial Intelligence
CMIP	Coupled Model Intercomparison Project Phase
CNN	Convolutional Neural Network
CSV	Comma Separated Values
CVAE	Convolutional Variational Auto-Encoder
CWL	Common Workflow Language
DNN	Deep Neural Network
DT	Digital Twin
DTE	Digital Twin Engine
ECMWF	European Centre for Medium-Range Weather Forecasts
EMO	European Meteorological Observations
EO	Earth Observations
ERA5	Fifth generation ECMWF reanalysis for the global climate and weather
FESOM2	Finite-Element/volumE Sea ice-Ocean Model 2
GCNN	Graph Convolutional Neural Network
IBTrACS	International Best Track Archive for Climate Stewardship
LAI	Leaf Area Index
LSTM	Long Short-Term Memory
MODIS	Moderate Resolution Imaging Spectroradiometer
ML	Machine Learning
NetCDF	Network Common Data Form
RCP	Representative Concentration Pathways
SEAS5	Seasonal Ensemble Prediction System 5
SFINCS	Super-Fast INundation of CoastS

SSH	Sea Surface Height
SSP	Shared Socioeconomic Pathways
TC	Tropical Cyclone
VAE	Variational Auto-Encoder
Wflow	hydrological modelling framework
WP	Work Package
xtclim	Generic Climate Extreme characterization and detection AI-based tool

Terminology / Acronyms: **https://confluence.egi.eu/display/EGIG**

Table of Contents

List of Figures

List of Tables

Executive summary

This report presents the final architecture design of the different Digital Twin applications for the environmental domain (tasks 4.5, 4.6 and 4.7 in interTwin). Its main objective is to describe the motivation behind the development of specific DT applications, as well as the targeted user and the key capabilities envisioned. In this respect, it shows an updated view of the architecture design, main requirements and workflows with respect to what was presented in D4.1 “First Architecture design of the DTs capabilities for climate change and impact decision support tools”.

The DT applications address a variety of aspects from the point of view of climate change analysis and impact decisions systems in various geographical regions and for different events including, among the others, tropical cyclones, wildfires, floods, drought and other extreme events. For each application a set of user stories considering various users, such as scientists, developers, decision makers or policy makers are presented, together with the key requirements. Moreover, each DT application dives into the workflow needed to implement the user stories, highlighting the main steps and the interactions with the interTwin DTE component.

In order to simplify the usage of the DT applications, these are designed to provide the end users with either a graphical tool or a Jupyter Notebook-based interface to visualize the results from the DTs. Overall, the applications described in the document are designed to focus on specific climate change impacts and provide valuable insights for assessing climate risk, identifying early warning signals, and implementing mitigation measures.

1 Introduction

1.1 Aim of this deliverable

The goal of deliverable 4.5 is to present the final architecture design for the environmental domain Digital Twin Applications (T4.5, T4.6, T4.7) and their key requirements for the interTwin DTE. It provides an updated overview of the DT applications features and design concerning what was initially presented in D4.1 **[R7]**. In particular, as the developments on the DT applications have advanced in the meantime, the requirements and the workflow description have been refined and extended in several cases. To this end, the workflow steps and diagrams have been specialized to include links to the actual interTwin DTE components. Moreover, in some cases the user stories have also been further extended to better describe additional case studies not yet clearly identified in the first design. It is also worth mentioning that an additional DT application, the eddies one, has been introduced in the document.

1.2 Intended audience

Deliverable 4.5 is useful for both developers and end users as described below:

For **developers**: it provides them with insight into different components, data integration strategies, and computational models required to build an effective digital twin. It would allow them to incorporate new features, leverage components and workflows, improve scalability, support evolving problems over time, and ensure interoperability.

For **end users**: it facilitates data sharing, integration, and analysis among various stakeholders and scientists. By establishing a common framework, researchers and stakeholders will be able to exchange information, validate models, and collaboratively address climate change impacts and suggest mitigation measures.

1.3 Structure of the document

The structure of this deliverable is as follows. **Section 2** describes the user stories and requirements for each digital twin application: a table is provided to describe the user goals and their requirements following the MoSCow (M - Must have, S - Should have, C - Could have, W - Won't have) method[1]. **Section 3** presents the architecture design and illustrates the workflow followed by each DT application. It depicts sequential or parallel steps involved in each DTs operations, highlighting the input, processing, and expected outcomes. Links with components of the DTE are also highlighted. Finally, **section 4** provides the conclusions for the document.

[1] https://en.wikipedia.org/wiki/MoSCoW_method

2 DT Applications User Stories

2.1 DT Application: Changes in Tropical Storms in response to climate change

Geographical region of interest: North Pacific

Summary:

Tropical Cyclones (TCs) are among the most impactful weather phenomena posing significant risks to ecosystems and human life. Every year, an average of 90 TCs occur over tropical waters [R11] and global warming due to climate changes is making them larger, more intense and destructive [R12, R13, R14].

The accurate detection and tracking of tropical cyclones (TCs) is challenging and a key research area in climate science. With the recent advancements in ML several research efforts have been started towards the development of data-driven TCs detection. Data-driven models can learn non-linear correlations between the cyclogenesis weather variables and the occurrence of a TC and can be used to address the TC detection in an efficient and cost-effective manner.

This DT will focus on the detection and tracking of tropical cyclones that consists of localizing, given some input drivers, the cyclone centre (or "eye") in terms of latitude and longitude coordinates using a ML model. A deterministic tracking scheme is subsequently used for joining the different TCs centers in tracks. In particular, a "hybrid" approach is proposed [R9] by linking the data-driven model for detection with the deterministic tracker.

ML models, such as CNNs and GCNNs, are used to learn the mapping between climatic variables and the positions that storms follow during their lifetime in historical records. Trained models will be then applied to predict the occurrence of storms using future projection data in order to give an indication across both space and time about how changes in climate affect the frequency and duration of this phenomena.

The requirements have been refined in this updated version of the design document to better describe user interactions, targeting the configurability of the digital twin (DT) application, as reported in the following. To this end, the previous set of requirements reported in Table 1 in D4.1 [R7] has been extended with a new requirement (4.5-1), while the two requirements previously highlighted have been merged together in (4.5-2) as these were quite similar.

Use case

The aim is to support scientists and researchers in:

1. configuring, training and testing pre-defined ML models (through scripts) for the detection of tropical cyclones using, for example, different driver variables.
2. studying and analyzing changes in tropical cyclones due to climate change through interactive Jupyter notebooks.

Preconditions

Users have access to DT data, ML models, DTE software infrastructure with core and thematic components (e.g., Docker images), scripts for training and notebooks for test/inference.

To train and test a ML model for TC detection:

1. Users can select from a configuration file the ML model hyperparameters (e.g., batch size, epochs, activation functions, loss, optimizer, scheduler, etc.), the driver variables, and the type of model (e.g., CNN, GCNN, Transformers);
2. The user runs the training of the ML model for the DT application with the selected hyperparameters;
3. The resulting trained model can be stored in a repository of models and tested through a notebook. The new model could then be used within the ensemble in the analysis case.

To run the analysis on TCs with the trained ML models:

1. Users can specify from the notebook:
    a. future projection, past and reanalysis climate data from a given list (e.g., CMIP6 or ERA5);
    b. temporal extents;
    c. ML models to be used in the ensemble from a set of pre-trained models;
2. The user runs the DT workflows on the selected input data and ML models;
3. The output of the DT (lists of detections, tracks and related plots and maps) can be downloaded as CSV file and images (e.g., png, jpeg). Maps and plots can be visualized in the notebook. Some examples include: Probability of Detection and False Alarm Rate (only on historical data), frequency/number of TC occurrences on a seasonal/annual basis, track duration, spatial distribution of TC tracks/detections.

Table 1 – User stories for DT Application: Changes in Tropical Storms in response to climate change

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.5-1	Scientist/ Researcher with good technical expertise	Specify the hyperparameters and type of ML model to be trained	Can run training and validation of the data-driven model for TC detection	The resulting trained model is stored on the ML models repository	**Must have:** - Access to ML training environment and scripts/software - Access to necessary data - Access to the ML model repository **Should have:**

					- Functionalities to log and track the training process
4.5-2	Scientist/ policy maker with some or little technical expertise	Specify a period of interest, the climate data to be used for TC analysis and (optionally) the pre-trained ML models to be used	The DT can perform TC detection and tracking on the input climate data	The resulting data with TC detections, tracks and plots are generated	**Must have:** - Access to Jupyter Notebook as a service - Access to necessary data - Access to the ML models **Could have:** - Possibility to adjust the visualisation interactively - Possibility to download the results **Won't have:** - An operational system, this is a demonstrator only

2.2 DT Application: Changes in wildfires in response to climate change

Geographical region of interest: Global

Summary

Several studies show that the effects related to climate change will affect both the frequency and severity of wildfires [R16, R18]. Modelling fire regimes represents, thus, an important tool for assessing future potential impacts on ecosystem functioning and society. Machine learning algorithms have emerged recently as effective alternatives for the prediction of wildfire occurrences compared to the traditional approaches (e.g., dynamic global vegetation models). Indeed, data driven models can learn complex interactions providing accurate predictions and unraveling potential relationships between variables.

The DT application related to wildfires focuses on global-scale wildfire projections, predicting the extent of burned areas using ML-based techniques. The ML model is trained on historical burned areas and validated against the actual burned area at the given time.

ML models, such as the UNet++ [R10], are trained to learn the non-linear relationship between different climatic, weather, and vegetation variables provided as input and the likelihood of wildfires in the geographical domain of interest. Users can configure the model setup, for example, to assess the fitness of the model in predicting burned area with respect to different driver variables. The trained models are then applied to standardized climate projections to identify regions most vulnerable to wildfires due to escalating climate change under different scenarios such as the Shared Socio-economic Pathways (SSPs) [R8].

Also in this case, with respect to the initial version of the design, the requirements have been refined to also support a higher extensibility of the DT application. Thus, a new requirement 4.5-3 has been formalized in Table 2, while the two requirements previously highlighted in D4.3 have been merged together in 4.5-4 since these shared several commonalities.

Use case

The aim is to provide scientists and researchers with:

1. tools (e.g., scripts, notebooks) to configure, train and test pre-defined ML models concerning the prediction of burned area maps on a global scale, for example with a subset of the driver variables.
2. configurable Jupyter notebooks to analyse the results concerning burned area projection using climate data.

Preconditions

Users have access to DT data, ML models, DTE software infrastructure with core and thematic components (e.g., Docker images), scripts/configuration files for training and notebooks for testing/inference.

To train and test a ML model for burned areas prediction:

1. Users can select the ML model hyperparameters (e.g., batch size, epochs, loss functions) and the driver variables;
2. The user runs the training of the ML model for the DT application with the selected hyperparameters;
3. The resulting model can be stored in a repository of models and tested through a notebook with different metrics.

To run the analysis on wildfires with the ML models:

1. Users can specify from the notebook:
    a. future climate scenarios from a given list (e.g., CMIP6 model/scenarios);
    b. temporal and geographical extent (global/regional);
    c. pre-trained ML model (or use the default one);
2. The user runs the DT workflows on the selected input data and trained model;
3. The output of the DT can be downloaded/saved and visualized in the notebook. Maps and charts can be customized through widgets. Some examples include: seasonal/annual burned area maps; trends/interannual variability of burned areas; burned areas aggregated by region.

Table 2 – User stories for DT Application: Changes in wildfires in response to climate change

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.5-3	Scientist/ Researcher with good technical expertise	Specify the hyperparameters and drivers to configure the ML model to be trained	I can run training and validation of the ML model for burned areas prediction	The resulting trained model is stored on the ML models repository	**Must have:** - Access to ML training environment and scripts/software - Access to necessary data - Access to the ML model repository **Should have:** - Functionalities to log and track the training process **Could have:** - Functionalities to evaluate the ML model quality
4.5-4	Scientist/ policy maker with some/little technical expertise	Specify a period of interest, the climate projections for the burned area prediction maps and (optionally) the pre-trained ML model to be used	The ML model can predict the burned areas	The resulting maps/plots of the burned areas are produced	**Must have:** - Access to Jupyter Notebook as a service - Access to necessary data - Access to the ML model repository **Could have:** - Options to interactively adjust the visualisation - Possibility to download the results **Won't have:** - An operational system, this is a demonstrator only

2.3 DT Application: Eddies prediction

Geographical region of interest: South Atlantic Ocean, Gulf Stream

Summary

This DT will focus on the detection of oceanic mesoscale eddies and their classification between cyclonic and anticyclonic, respective of the rotation verse. This consists in identifying, given the SSH information on an interpolated FESOM2 grid, the segmentation mask that contains the pixel information about presence or absence of eddies, and their type. This kind of approach involving trained Convolutional Neural Networks allows to greatly improve the detection speed of oceanic eddies with respect to the classical mathematical models that have been used so far.

Use case

The goal is to provide scripts and notebooks to the scientists who will conduct analysis of oceanic eddies data.

Preconditions

Users have access to DT data, models, thematic components and scripts/notebooks.

To train and test a ML model for eddies detection:

1. Users can select:
    a. temporal extents;
    b. ML models and hyperparameters;
2. The user runs the training of the ML model for the DT application with the selected hyperparameters;
3. The resulting model can be stored on a repository of models and tested through a notebook.

Table 3 – User stories for DT Application: Eddies prediction

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.5-5	Scientist with good technical expertise	Specify a period of interest, ML model to use and hyperparameters to be used for eddies analysis	I can train a ML model and/or use a pretrained one to perform eddies detection on the input FESOM2 climate data	The ML model is trained and the segmentation masks are generated	**Must have:** - Access to Jupyter Notebook as a service - Access to necessary global data - Access to the ML models

2.4 DT Application: Post-flood analysis in coastal regions

Geographical region of interest: Baltic Coast, Germany

Summary

The DT for post-flood analysis in coastal regions will focus on the generation of flood risk maps that trigger early warning alerts when a flood is predicted. The system will be demonstrated for historical flood events of the Baltic Coast, Northern Germany.

Output flood risk maps are produced from SFINCS, a reduced-complexity model for super-fast dynamic modelling of compound flooding. SFINCS will be forced by example (historical) weather forecasts. Additionally, the DT will combine the SFINCS flood maps with Sentinel-1 based flood maps generated by the openEO implementation of the Global Flood Monitor. The mentioned software components are described in D7.1 [R6].

Use case

The goal is to provide Jupyter Notebooks for scientists and decision-makers to:

1. Set up the necessary models for a user-defined region of interest.
2. Run the necessary models and Earth Observation processing pipelines to produce deterministic and probabilistic flood maps for a user-defined region of interest and validate the resultant output data against observations.
3. Prepare the data for easy ingestion into an early warning system.

Preconditions

Users have access to DT data, models, thematic components and Jupyter Notebooks.

1. Users can:
    a. specify a region of interest;
    b. specify a temporal period to simulate;
    c. select local data for the models if available;
2. User runs the DT workflows for the specified region and period using default global data or selected local data if available;
3. The output of the DT can be visualised in the Jupyter Notebooks and the data can be downloaded/saved as NetCDF data.

Table 4 – User stories for DT Application: Post-flood analysis in coastal regions

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.6-1	Decision maker with little technical expertise	Specify a geographic region and temporal period of interest	I can set up the automated processing of data for flood inundation models and hydrological models and access relevant EO data for flood monitoring and forecasting	When the system simulates a historic flood event and data is automatically prepared, so an automated early warning system can access it in a standardised form	**Must have:** - Access to Jupyter Notebook as a service - Access to necessary global data **Should have:** - Example visualisations of output to support validation - trigger sending alert based on flood extent **Could have:** – option to upload local data – interactive Solara[2]-based front-end **Won't have:** – Operational early warning system. This is a demonstrator only.
4.6-2	Expert user with good technical expertise but little domain knowledge	Process and combine modelled and EO-based flood-related data for specific regions of interest	I can get tailored information on flood monitoring and forecasting	I can provide decision makers with a thorough overview of the expected flood event	

2.5 DT Application: Alpine droughts early warning

Geographical region of interest: Alps

Summary

This DT aims to develop a prototype of a seasonal hydrological forecasting system for the Alps at the river basin scale. A deep learning model (surrogate model) is trained to replicate the Wflow_sbm process-based hydrological model. The static input parameters of the surrogate model will be calibrated by minimising a loss function between model output and targets (i.e., satellite surface soil moisture) over the area of interest. Once the model is trained, seasonal forecasts from ECMWF will be used to produce hydrological forecasts and identify regions affected by low water availability. To increase the spatial resolution of hydrological forecasts, a machine learning algorithm to downscale seasonal forecasts is also developed.

[2] https://solara.dev/docs/

Use case

The prototype of a drought early warning system is developed and integrated into openEO as a user interface for researchers and decision makers. Through openEO the user should be able to:

1) Run a trained model for a specific area of interest and temporal extent. The detailed information about openEO is already described and is available in Deliverable D6.1[R3].
2) Validate results using historical observations.
3) Run the model driven by seasonal forecasts to identify areas affected by low water availability.

Preconditions

1) Users have to access the DT using openEO, available global data, or upload their regional data.
2) The module for downscaling climate data and the trained surrogate model should be implemented and available in openEO.
3) Modules for coregistering and formatting all the required input data have to be available in openEO.
4) The output from DT can be directly examined using openEO and can also be downloaded for later use.

Table 5 – User stories for DT Application: Alpine droughts early warning

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.6-3	Researchers	Use openEO to build a process graph to load different inputs and train a surrogate model that emulates a hydrologicAl model	I can study the drought generating processes I can investigate the uncertainty of the model output I can perform sensitivity of model output to different inputs I can leverage cloud-based processing capabilities	When simulated results are in good agreement with historical observations and the trained model is able to reproduce observations	**Must have:** - Access to openEO Authentication and Access Control - Access to modules to ingest and pre-process EO and climate data - Access to the trained surrogate model - Graphical visualization of the output **Should have:** - Estimation of model uncertainty **Could have:**

4.6-4	Local/ Regional public authority in the field of agriculture, hydrology and river basin management	Know the water available on the use case's basins for the next 7 months	I can have an overview of the regions with low(er) water availability	It is possible to visualise maps of the output The accuracy of the prediction is high	- Access to hydrological data for validation. **Won't have:** - Operational drought early warning system

2.6 DT Application: Extreme rainfall, temperature and wind weather event changes in response to climate change

Geographical region of interest: Europe, but could be applied anywhere in the world

Summary

The DT for impacts of extreme weather event changes in response to climate change focus on providing to the users information on the changes of the characteristic of those events and impacts compared to a reference period and a specific region. The focus is on temperature, but can be applied as well to other climate variables such as precipitation and wind. The change of characteristics is assessed, such as intensity (if relevant), duration, and frequency of occurrence. This DT uses an innovative Artificial Intelligence (AI) method based on a Convolutional Variational Auto-Encoder (CVAE) method to detect anomalies.

Use Case

The goal is to provide Jupyter Notebooks for scientists and advanced users to:

1. Assess the changes of characteristics of specific weather extreme events and assess their impacts.
2. Assess over a selected geographical region, a specific climate reference period, a specific future time period with one or several greenhouse gas scenarios.

Preconditions

1. Set up the necessary CVAE model for a user-defined extreme weather event, geographical region of interest, climate reference time period (historical 20-30 years), season, future climate scenarios (RCPs) and climate time period of interest.
2. Train the CVAE model according to end users' choices of step 1, on separate climate models from the CMIP6 archive, using several climate model ensemble members.
3. Run the CVAE model on the end users' choices of extreme weather events, geographical regions, future climate scenarios (RCPs), and climate time periods of

interest, for several climate models using specific training information according to each climate model and specific reference period.

4. Generate end users' products related to the changes in characteristics of those events: intensity (if relevant), duration, and frequency of occurrence. Those products will be specific plots and maps. Output data is also available to end users for further data processing. Multiple climate models and RCPs can be used to provide uncertainties evaluation with a range of characteristic changes.

Table 6 – User stories for DT Application: Extreme rainfall, temperature and wind weather event changes in response to climate change

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.7-1	Policy maker with little technical expertise	Specify an extreme weather event, a geographical region, a reference historical climate time period, a future time period of interest, and the CMIP6 SSP scenarios to be used for extreme event analysis	The CVAE model will be automatically trained and applied on the input climate data	The resulting datasets with extreme events change of characteristics will be calculated and plots are generated	**Must have:** – Access to Jupyter Notebook as a service – Access to necessary global data – Access to the CVAE model **Could have:** – Possibility to adjust the visualisation interactively – Possibility to download the results
4.7-2	Scientist with some technical expertise	Specify an extreme weather event, a geographical region, a reference historical climate time period, a future time period of interest, and the CMIP6 RCP scenarios as well as CMIP6 specific	Can run complex analysis with specific CMIP6 climate models to better assess uncertainties	The resulting datasets with extreme events characteristics and plots are generated	**Must have:** – Access to Jupyter Notebook as a service – Access to necessary global data – Access to the VAE model **Could have:** – Interfaces to adapt the visualisation – Possibility to download the results

		climate models to be used for extreme event analysis			

2.7 DT Application: Flood climate impact in coastal and inland regions

Geographical region of interest: Humber, United Kingdom

Summary

The DT for flood climate impact in coastal and inland regions will focus on the generation of flood maps and quantifying impacts on buildings, utilities, roads and accessibility under future climate conditions. Additionally, end-users can select flood mitigation and adaptation measures and test their effectiveness. The system will be demonstrated for flood scenarios under climate change for Humber, United Kingdom.

Flood maps under future climate scenarios are produced from SFINCS, a reduced-complexity model for super-fast dynamic modelling of compound flooding, which receives river discharge data from Wflow, a hydrological model. The flood maps are then used by Delft-FIAT, a flood impact assessment tool, and RA2CE, a Resilience Assessment and Adaptation for Critical infrastructurE – model, to quantify impacts and damages to buildings, utilities, roads and accessibility.

Additionally, end-users will be able to select flood mitigation and adaptation measures and re-run flood scenarios to test their effectiveness in reducing flood-related impacts.

Use case

The goal is to provide Jupyter Notebooks for scientists and decision-makers to:

1. Set up the necessary models for a user-defined region of interest.
2. Run the necessary models to produce baseline flood maps for a user-defined region of interest and quantify impacts and damages to buildings, utilities, roads and accessibility.
3. Select flood mitigation and adaptation measures and re-run flood scenarios to test their effectiveness at reducing flood-related impacts.

Preconditions

The user has access to DT data, models, thematic components and Jupyter Notebooks.

1. Users can:
    a. specify a region of interest;
    b. specify a temporal period to simulate;
    c. select local data for the models, if available;

    d. select and specify mitigation and adaptation measures;

2. The user runs the DT workflows for the specified region and scenario using default global data or selected local data if available;
3. The output of the DT can be visualised in the Jupyter Notebooks and the data can be downloaded/saved as NetCDF and GeoPackage data.

Table 7 – User stories for DT Application: Flood climate impact in coastal and inland regions

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.7-3	Decision maker or planner with little technical expertise	Specify a geographic region, a climate change scenario and select mitigation / adaptation measures of interest.	I can set up the flood inundation and hydrological models and run flood scenarios under future climate conditions and test the impact of the selected mitigation / adaptation measures	When the system simulates a flood scenario and quantifies the impacts and damages to buildings, utilities, roads and accessibility.	**Must have:** - Access to Jupyter Notebook as a service - Access to necessary climate projection data **Should have:** - Flood and related impact visualisations and data to support decision-making - Options to select future climate change scenarios - Options to select mitigation and adaptation measures **Could have:** - Option to upload local data - Interactive Solara-based front-end **Won't have:** - An extensive list of adaptation and mitigation measures - An operational system, this is a demonstrator only
4.7-4	Expert user with good technical expertise but little domain knowledge	Process and combine different models and tools needed for flood related adaptation planning for specific regions of interest	I can get tailored information on flood scenarios under future climate conditions and make decisions on what adaptation and mitigation measures to invest in	I can provide decision makers with a thorough overview of the expected flood scenarios and their impact under future climate conditions	

3 DT Applications Design

3.1 DT Application: Changes in Tropical Storms in response to climate change

3.1.1 ML Model Requirements

The application input consists of a set of 2-dimensional data and each variable can be easily considered as a 2D image, where each pixel corresponds to a cell of the lat-lon grid. Based on this consideration the ML architecture identified for the DT is a Convolutional Neural Network (CNN) or a Graph Convolutional Graph Neural Network (GCNN). Besides traditional convolutional networks, Transformer-based models are considered too. In the following, Deep Neural Network (DNN) is used to refer to either approach.

The input climatic drivers are linked with the records provided by IBTrACS. In order to feed the DNN with the images, the input variables are stacked together and tiled into non-overlapping patches of fixed size, generating an input of dimension $H \times W \times C$ (where H and W are height and width, respectively, and C is the number of input drivers). Each patch containing a cyclone is associated with the corresponding (row,col) local coordinates, while each patch without a cyclone is associated with a negative value, e.g., (-1,-1). In the case of Graph-based networks, an additional step is performed to convert the patches in a graph structure. This partitioning is used to improve the model efficiency and, more importantly, to ensure that each patch is more likely to contain at most a single TC. In terms of software infrastructure, solutions like Lightning and PyTorch are exploited.

The list of identified data variables is available in Table 8. Note that different subsets of variables can be used for training.

Table 8 – List of data variables for the TC DT Application

Variable name	Temporal Resolution	Spatial Resolution	Unit	ERA5 Name	CMIP6 Name
10 m wind gust since previous post-processing	6-hourly	0.25°×0.25°	m/s	fg10	N.A.
10 m instantaneous wind gust	6-hourly	0.25°×0.25°	m/s	i10fg	wsgmax10m
temperature at 500 mb	6-hourly	0.25°×0.25°	K	t_500	ta
temperature at 300 mb	6-hourly	0.25°×0.25°	K	t_300	ta
relative vorticity at 850 mb	6-hourly	0.25°×0.25°	1/s	vo_850	rv850 (can be derived from eastward “ua” and westward “va” wind

					components)
mean sea level pressure	6-hourly	0.25°X0.25°	Pa	msl	psl

3.1.2 Workflow Description

This section describes the logical flow of operations the DT on TCs needs to perform to carry out user requests. The links WP5 (infrastructure), WP6 (core) and WP7 (thematic) modules are highlighted also in the text. With respect to the workflow presented in D4.1 [R7], the description has been specialized for the two different use cases presented in Section 2.1.

In terms in computing workflow, the DT on TCs includes the following steps (refer to Section to 2.1 for the use case description):

To train and test a ML model for TC detection:

1. Users can define the training hyperparameters, the type of ML model and the subset of variables used as drivers from the training dataset (ERA5 single and pressure levels). To this end solutions from WP5 could be used to manage access to the training and validation data;
2. Data is partitioned in patches and normalised based on the training set. In addition data augmentation procedures of ERA5 data are also required to increase the number of training examples. Steps for generating the graph structure could also be applied. The capabilities from the ML TC detection thematic module are used (WP7);
3. Training of the ML model can be offloaded on HPC resources (WP5) and distributed on multiple GPUs and nodes. During training stage, metrics could be logged on MLFlow and provenance could be tracked by using components from WP6 (e.g., yProv4ML, itwinai);
4. At the end of the training stage, the trained ML model together with other useful artifacts can be stored on MLFlow (WP6).

For the analysis test case:

1. Users can select the proper data for running their analysis; i.e., specifying the temporal extensions to be considered and the CMIP dataset. Specific thematic module (WP7) jointly with the WP5 infrastructure can be used for accessing the data;
2. Selected data is pre-processed (e.g., subsetting, re-gridding, partitioning in non-overlapping patches, conversion to graphs for GNNs, etc.) so that it can be used as input to the pre-trained ML models. Furthermore, the data will be normalized with the training scaler, using components from the ML TC detection thematic modules (WP7);
3. One or more ML models from the repository of pre-trained models (WP6) is executed on the data from step 2, potentially on a HPC infrastructure (WP5);

4. Results from different models can be combined together with an ensemble approach, according to the user's input. When using the ensemble of ML models, the resulting detections from the ensemble are then aggregated;
5. Results from the inference stage are then post-processed to build the final results. A deterministic tracking scheme is applied to the detections in order to align them and produce the TC tracks, using again the capabilities from WP7 (i.e., ML TC detection);
6. Final results can be stored as output files or visualised in the notebook interface.

Figure 1 shows the TC DT application workflow diagram.

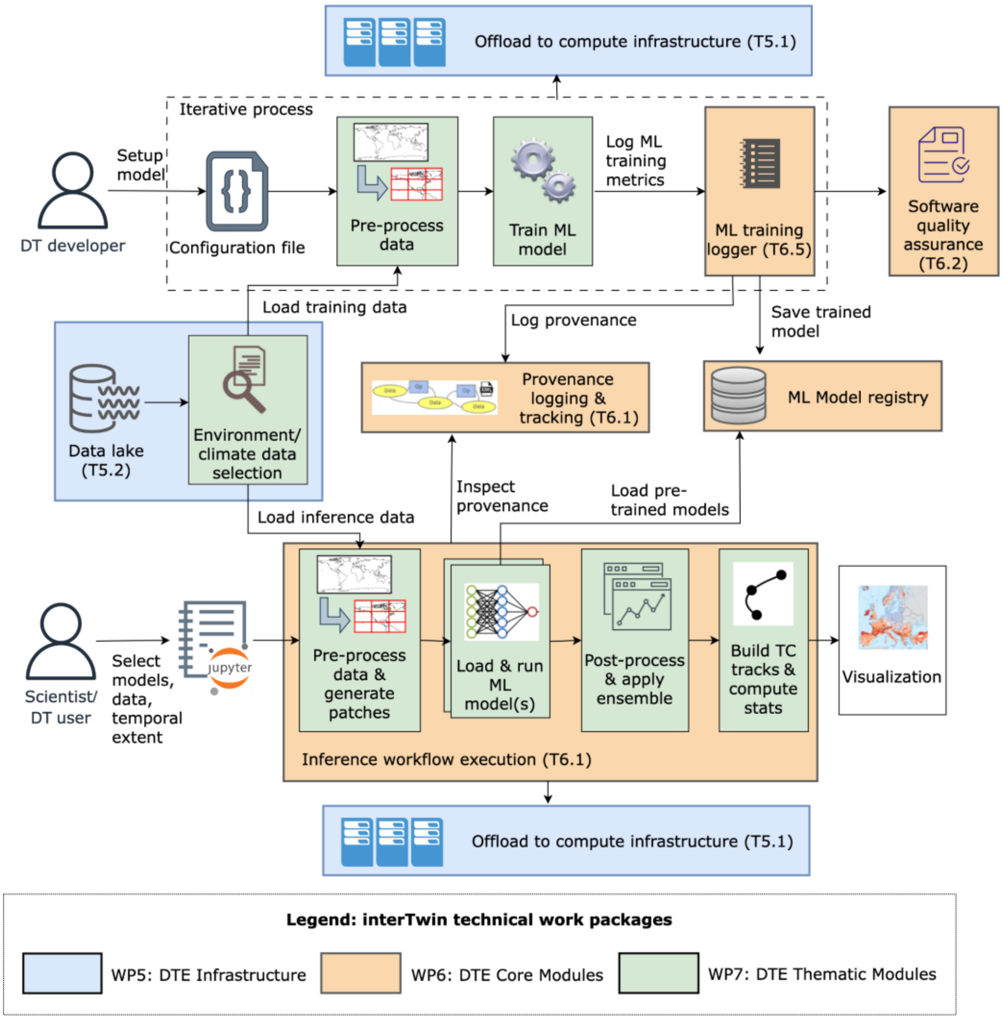

Figure 1 – Overview of the workflow for the DT application on TCs

3.2 DT Application: Changes in wildfires in response to climate change

3.2.1 ML Model Requirements

The main goal of the wildfires DT case study is to develop Convolutional Neural Network-like architectures capable of learning complex relationships between chosen input variables and and the burned areas on a global scale. These networks output the burned areas in hectares, which indicate the severity of wildfires in different regions on the global map.

Multiple architectures have been investigated, such as UNet **[R15]** and UNet++ **[R10]**. The proposed ML architectures take as input a stack of climatic variables of dimension $H \times W \times C$, where H and W are, respectively, the height and width of the input map and C is the number of input drivers, and provide as output a map of dimension $H \times W$ with the percentage of burned areas per each pixel of the map. Also in this case solutions like Lightning and PyTorch are used to implement the ML models.

List of currently identified data variables is available in **Table 9**.

Table 9 – SeasFire Cube and corresponding CMIP6 data variables identified to carry out the wildfires prediction case study

Full name	SeasFire Cube name	Unit	CMIP6 name
ERA5 Meteo Reanalysis Data			
Total precipitation	tp	m	pr
Relative humidity	rel_hum	%	hur
Sea Surface Temperature	sst	K	tos
Temperature at 2 meters – Min	t2m_min	K	tasmin
Land-Sea mask	lsm	0-1	sftlf
Nasa MODIS MOD11C1, MOD13C1, MCD15A2			
Land Surface temperature at day	lst_day	K	ts
Leaf Area Index	lai	m2m-2	lai

Global Wildfire Information System (GWIS)			
Burned Areas from FCCI	fcci_ba	ha	Used only for training

3.2.2 Workflow Description

Similarly to the previous DT application, the workflow description has been specialized to better describe the two different use cases presented in Section 2.2. In terms of computing workflow, the DT on wildfires comprises these steps:

To train and test a ML model:

1. Users can define the training hyperparameters, the subset of variables used as drivers and the time period of the training data (e.g., SeasFireCube [R17]) to be used for training a ML model. Training/validation data can be accessed from WP5 infrastructure;
2. Data is normalised based on the training set. In addition data augmentation techniques can be applied using the features from the thematic modules (WP7), i.e. ML4Fires;
3. Training of the defined ML model can be executed on HPC resources and distributed on multiple GPUs/nodes. During training stage, metrics can be logged on solutions like MLFlow and provenance can be tracked by using the DTE core component from WP6 (e.g., yProv4ML, itwinai);
4. At the end of the training stage the trained ML model, together with scaler and provenance documents can be stored on MLFlow (WP6);
5. The Software Quality Assessment core module (WP6) can be used to evaluate different skills/metrics of the trained model on a small test set.

For the analysis test case, the workflow of the application will be orchestrated by one of the workflow engine solutions from WP6, and comprises the following steps:

1. Users can select the proper data for running their analysis; i.e., specifying the temporal extensions to be considered or the models from CMIP experiments. Thematic modules such as esgpull_rucio (WP7) can be used for uploading, beforehand, the climate projection data on the infrastructure (WP5);
2. Selected data is pre-processed so that it can be used as input to the ML models. Capabilities from thematic modules (WP7), i.e., ML4Fires, are used for running pre-processing functions. In particular, the input data is normalised according to the training scaler and pre-processed to be compatible with the training data structure;
3. A ML model from the repository of pre-trained models (WP6) will be retrieved, deployed and executed, potentially on a HPC infrastructure (WP5), on the data from step 2;

4. Results from the inference stage on projection data (e.g., CMIP6) will then be post-processed to build the final results. Results from multiple models can be also combined;
5. Final results can be stored as output files or visualised in the notebook interface.

Figure 2 shows the overall DT workflow. As it can be seen it shares some similarities with the one shown in **Figure 1**.

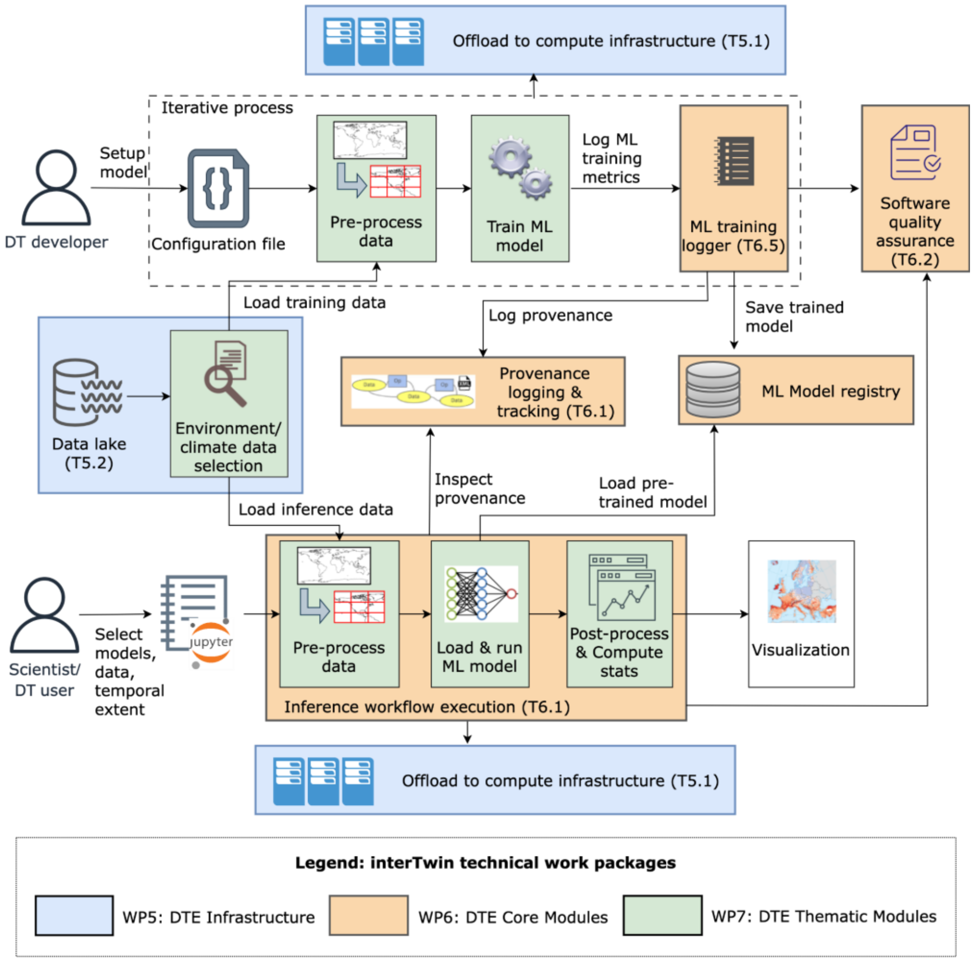

Figure 2 – Overview of the workflow for the DT application on wildfires prediction

3.3 DT Application: Eddies prediction

3.3.1 ML Model Requirements

The eddies DT application goal is to provide climate scientists and, in general, people with technical knowledge with the tools to quickly obtain segmentation masks for FESOM2 oceanic data. Due to the nature of the 2-dimensional input data, the main deep architecture included with the application is the Convolutional Neural Network (CNN). Specifically, the chosen model was the UNet **[R15]**, but many convolutional combinations may be used to learn the complex relationships between the input variable Sea Surface Height and the presence of eddies on the ocean surface.

The proposed ML architecture will take as input a series of 2-dimensional maps with dimension $H \times W$ and provide as output a segmentation mask of dimension $H \times W$ with each pixel being classified as:

- 0 – background
- 1 – cyclonic eddy
- 2 – anticyclonic eddy

In terms of software infrastructure, solutions like Keras/Tensorflow will be exploited.

3.3.2 Workflow Description

This section describes the logical flow of operations the eddies DT needs to perform to carry out user requests. The links to WP6 core modules and WP7 thematic modules are highlighted.

In terms of computing workflow, the DT on oceanic eddies comprises these steps:

1. Users can select the proper data for running their analysis; i.e., specifying the temporal extensions to be considered. The *Interpolation to regular grid* (part of the eddiesML thematic module – WP7) will be used to convert the unstructured FESOM2 grids into structured images. Selected data will be pre-processed via the *Ground Truth Generation (py-eddy-tracker)* (WP7) in order to generate the segmentation masks that will be used as the label information during the ML process. Alternatively, the user can start the process by taking already pre-processed data (WP5);
2. Once the SSH information and the segmentation masks are put together, the training phase can begin. The U-Net model defined inside the eddiesML module (WP7) will be exploited for this step. Within the same module, yProv4ML will be used to track ML metrics (WP6);
3. Only for the training workflow: training of a ML model can be triggered in order to use different hyper-parameters if results from testing are not satisfactory in terms of evaluation metrics (e.g., error and classification metrics). The trained models will be stored in a ML model repository for future usage (MLFlow – WP6);
4. Only for the inference stage: a ML model will be executed on the data from the first step;
5. Final results can be visualised in the notebook interface using different Python visualisation modules.

The overall workflow of the application will be executed on the WP5 infrastructure (e.g., Vega HPC system). **Figure 3** shows a diagram with the DT workflow main steps.

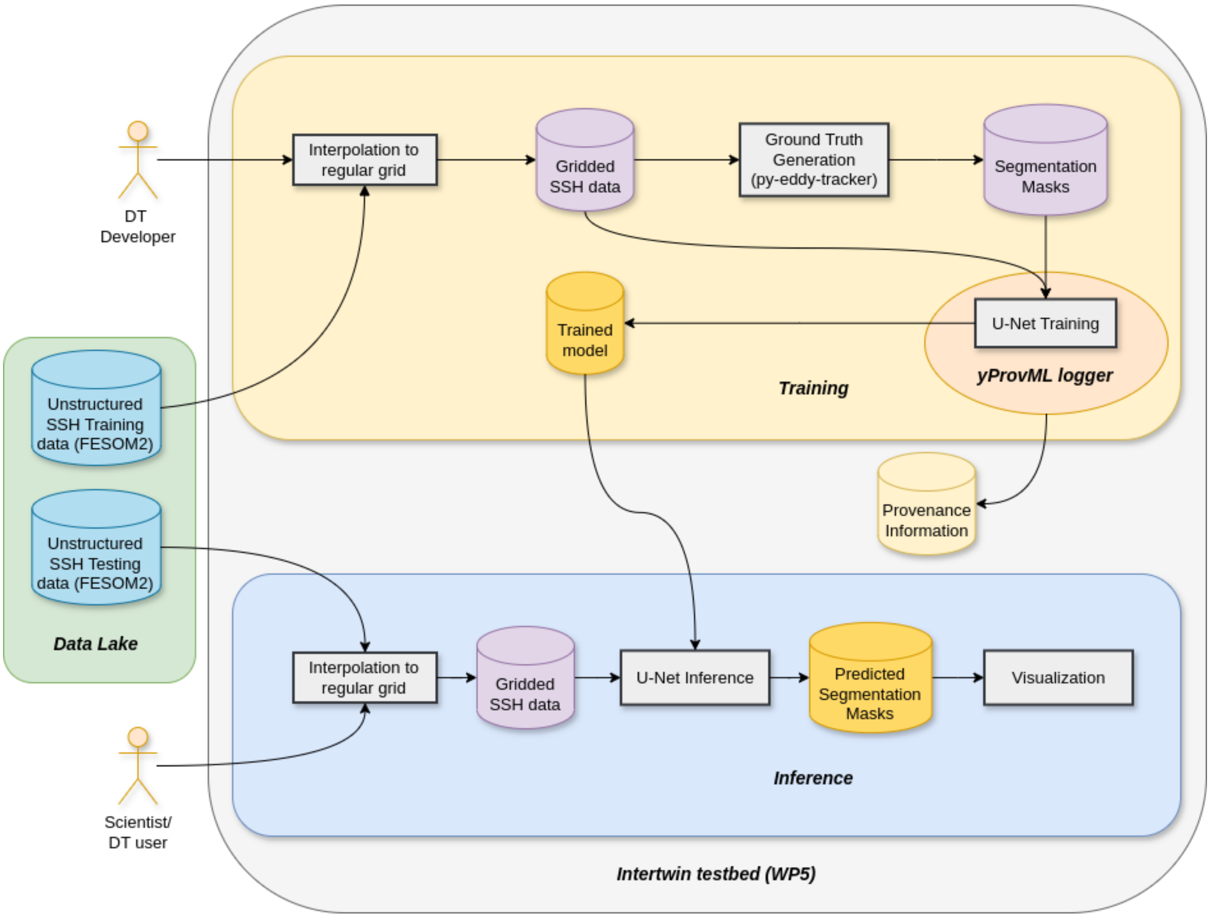

Figure 3 – Overview of the workflow for the DT application on eddies detection

3.4 DT Application: Post-flood analysis in coastal regions

3.4.1 Model Requirements

The post-flood analysis DT for coastal regions relies on a process-based model combined with satellite observations of floods:

1. Super-Fast INundation of CoastS (SFINCS): a reduced-complexity model designed for super-fast modelling of compound flooding in a dynamic way.
2. openEO satellite-based flood monitoring: An existing workflow for flood monitoring [R4] will be re-developed in the openEO syntax [R5] for being usable on several platform backends. The process graph will be gradually enhanced to create a fully automatic processing chain, based on Sentinel-1 σ_0 and Projected Local Incidence Angle image collections.

The flood inundation model (SFINCS) simulates how extreme coastal water levels might spread during flood events. The flow and spread of water is affected by the geometry of the floodplain (usually derived from Digital Elevation Models), the upstream inflow, downstream water levels, location of levees, dams, and channels and how these are operated.

For coastal flooding factors and data need to be considered, including tides, storm surges, sea level rise projections, waves, beach morphology and dynamics, coastal infrastructure and defenses, wind speed and direction and atmospheric pressure.

An overview of data requirements to run SFINCS can be found in **Table 4** and **Table 6** of interTwin D7.1 **[R6]**.

To produce deterministic flood maps for a given scenario or event, a specific (extreme) rainfall or surge event is selected. These, together with the aforementioned data representing relevant processes associated with coastal flooding, are fed into the flood inundation model (SFINCS) which simulates the flood extent and depth. The combined EO-model workflow is then used to do post-flood assessments, for example to run what-if scenarios to understand where dike breaches might have occurred.

To ensure accurate simulations models are usually calibrated and validated against historical data from in situ measurements and / or satellites. Those value-added satellite data will be processed using a global flood monitoring workflow that is being adapted to project specific requirements and that is being automated. In the first iteration the flood monitoring workflow is based on Bayesian decision making, exploiting data cubes of Sentinel-1 data with its orbit repetition and a-priori generated probability parameters for flood and non-flood conditions. Therefore, local seasonal non-flood conditions for each day-of-year are defined by pre-processing harmonic parameters of each pixel's full time series. As a stretched goal, processing this firstly static information can be embedded as a dynamic workflow into the operational process chain, only analysing, e.g., the recent two years and thus considering possible changes in the backscatter by, e.g., changes of land use / land cover. Another stretched goal is the usage of ML-based training instead of the lightweight Bayesian approach, making use of the available big data processing capabilities within the project. A data cube based masking of no-sensitivity resulting from ill-posted satellite geometries or impeding land cover further enhances the process' robustness. Implementing the described stretched goals will depend on 1) analyses on their potential to increase the product's accuracy, and 2) on decisions throughout the project regarding implementing redundant processing libraries, usable for this and for other workflows.

To set up SFINCS, the Digital Elevation Model is used to define the floodplain, i.e. the geometry of the rivers and floodplains. Boundary condition points are determined for the upstream inflows, downstream water levels or tidal conditions. The influence of bridges, dams, levees, and other infrastructure on flow dynamics should also be included.

An overview of data requirements to set up SFINCS can be found in Tables 3 and 5 of interTwin D7.1 **[R6]**.

In this digital twin application functionality is being developed that enables an end user

1. Easily set up SFINCS for a user-defined region of interest;
2. Easily run the models and produce deterministic flood maps.

3.4.2 Workflow Description

Two workflows are described in this section:

1. Setting up the necessary SFINCS models
2. Producing a deterministic flood map

Setting up SFINCS models for a specific region of interest comprises the following steps:

1. Set up SFINCS
    a. A user defines a geographic region of interest;
    b. A user defines the SFINCS model resolution;
    c. A user selects available global datasets from which to build the model. Optionally a user can upload and use local data instead of the global data.

Once the SFINCS model has been set up they can be run to produce deterministic and probabilistic flood maps as follows.

For deterministic flood maps:

1. A user selects a time period to simulate, e.g., a specific historical event;
2. A user selects the preferred forcing data for SFINCS, e.g.
    a. SFINCS – in addition to precipitation, for a coastal flood, select data for tides, surges, waves, and sea level.
3. A user then runs SFINCS;
4. The water levels produced by SFINCS are interpolated to the digital elevation model selected by the user in the Set up SFINCS step 1c (see above), to estimate flood depths and extent.

In the above workflows, this DT application will leverage the following capabilities from other WPs of interTwin (summarised in **Figure 4**):

- The DT will leverage capabilities developed in WP7, T7.6: Hydrological model data processing thematic module.
- To run the models using containers, this DT will rely on functionality from WP6, specifically OSCAR.
- For workflow composition and execution, this DT will leverage developments from WP6, T6.1 Workflow composition and WP6, T6.4 Workflow backend, specifically the CWL workflows developed for this application.
- Finally, all data and compute resources are leveraged from WP5, the DTE infrastructure including Notebooks as a Service.

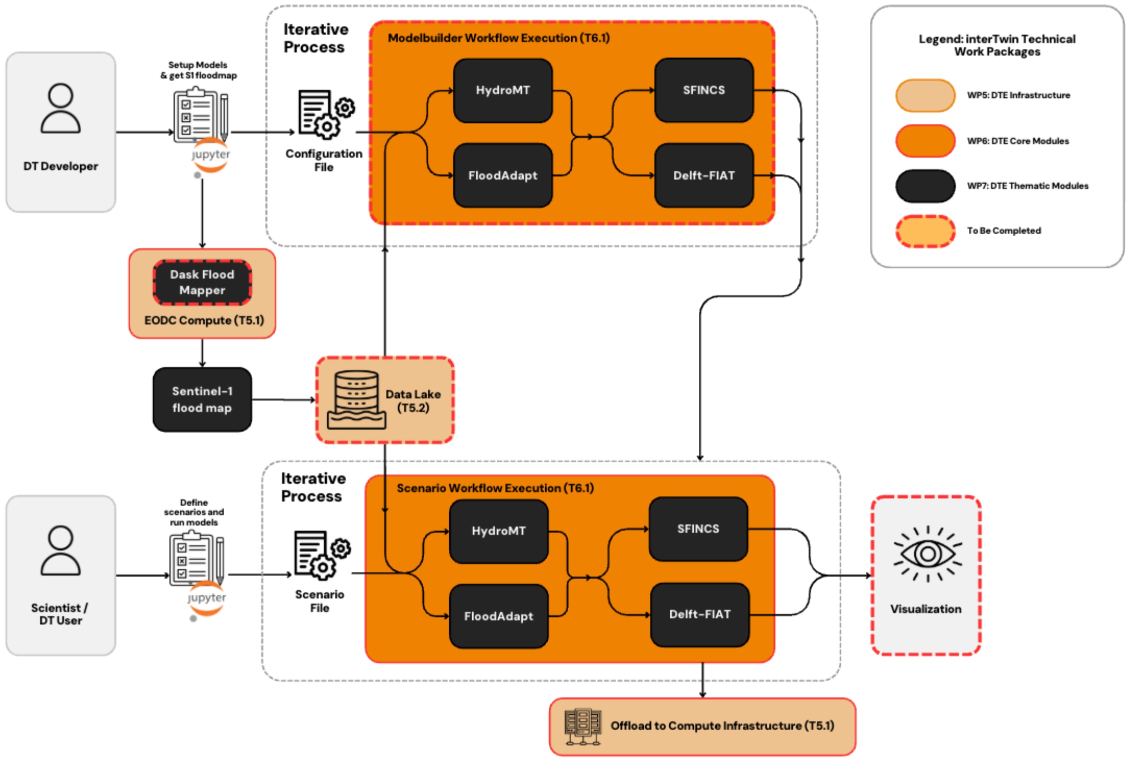

Figure 4 - Workflow components for the post-flood analysis in coastal regions

3.5 DT Application: Alpine droughts early warning

3.5.1 Model Requirements

The aim is to develop a deep learning model capable of emulating the hydrological processes of the physical based Wflow_sbm model. The surrogate is used to simulate hydrological states and fluxes (soil moisture, evapotranspiration and snow water equivalent). To train the surrogate the required inputs are static, such as the Wflow_sbm parameters that should be calibrated, and dynamic, such as the meteorological variables (temperature, precipitation and potential evapotranspiration). Meteorological daily inputs are derived from EMO1 (~1.5 km). Both the dynamic and static data are resampled to the resolution of the hydrological model grid (~1 km). Once trained, the deep learning surrogate can be used to calibrate the less computationally efficient physical based model by using spatially distributed estimates derived from EO. Currently, the DT enables calibrating the model using the satellite surface soil moisture estimates provided by TU Wien. After obtaining calibrated static parameters, these can be used as inputs to the surrogate, or to the original Wflow_sbm model, for inference. The hydrological forecasts are generated by forcing either models with SEAS5 seasonal forecasts. The SEAS5 forecasts are first downscaled to match statistics and spatial resolution of EMO1. The downscaling of forecast fields is performed through a GAN-based neural network specialized for each input variable. **Table 10** and **Table 11**presents the type of dynamic and seasonal forecast data used in this study. The description of static data is already available in D7.1[R6].

Table 10 – Details of dynamic data used in this study and their sources

Dynamic Data	Data Source	Time	Temporal Resolution	Spatial Resolution	Product Description
2m_temperature Total precipitation Surface solar radiation downwards	EMO1	2000-2022	Daily	~1.5X1.5 km	Climate
Potential evapotranspiration	derived from EMO1	2000-2022	Daily	~1.5X1.5 km	PET computed using the Jensen-Haise method
surface soil moisture	TU Wien	2017-2024	Sub-daily	~1 km	RT0

Table 11 – Details of seasonal forecast data used in the study and their sources

Variable	Spatial Resolution	Temporal Resolution	Product Description
2m_temperature Total precipitation Surface solar radiation downwards	36 X 36 km	Daily	Seasonal Forecast ECMWF

3.5.2 Workflow Description

The purpose of this section is to describe the DT developer and user workflows.

DT developer workflow

Four primary steps are involved:

i) Setting up physical based model Wflow_sbm;

ii) Pre-training the surrogate model to emulate Wflow_sbm;

iii) Calibrate static parameters of the Wflow_sbm model through the surrogate;

iv) Predict water availability using a trained surrogate model or wflow_sbm with calibrated static input parameters and downscaled seasonal forecast.

This activity will use the spatially semi-distributed hydrologic model Wflow_sbm [R1, R2] to estimate hydrological states and fluxes including actual evapotranspiration, soil moisture and snow water equivalent. One model is developed for the entire Alpine region with a 0.008333-degree cell resolution, which corresponds to approximately one kilometer. In this study, a surrogate model is developed in two phases as shown in Figure 5. The first phase involves training the LSTM to reproduce the performance of the process-based model Wflow_sbm by minimising the loss function (RMSE). The LSTM surrogate model is trained using Wflow_sbm's dynamic forcings, static attributes and dynamic targets such as soil moisture, evapotranspiration and snow water equivalent. As soon as the model is trained and the parameters are calibrated, seasonal forecast data from ECMWF at daily scale will be used in order to forecast hydrological conditions over the next 7 months

DT user workflow

1. Defining Geographical Extent

The user must specify the geographical area of interest.

2. Data Selection and Preprocessing

The DT Application provides flexibility to their users to choose globally available data, such as digital elevation model, forcings, etc. The user must preprocess and harmonise the data according to model resolution. This process ensures that the surrogate model receives clean and standardised data for simulation.

3. Model Selection

Following the preprocessing of the data, the user can select the trained surrogate model or the wflow_sbm model, with calibrated parameters, to simulate historical hydrological conditions or forecast hydrological conditions for the next 7 months. The current surrogate model is trained and validated on the Alps. If the user wishes to run the model on a different region it will be necessary to retrain the model. If the user wants to recalibrate the parameters using additional datasets, e.g., snow water equivalent, he has to rerun the calibration.

4. Visualisations

Final results can be visualised thanks to dedicated user defined openEO processes. Results can also be downloaded for further analysis or communication.

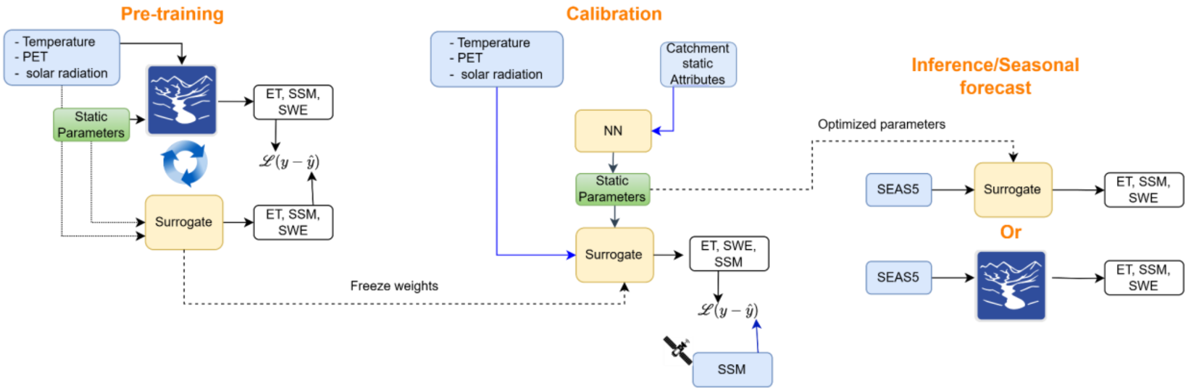

Figure 5 – 1) Pre-training: A deep learning model is trained to emulate the Wflow_sbm physical based hydrological model 2) Calibration: The static parameters are calibrated using satellite soil moisture 3) Inference: the surrogate model or the original Wflow_sbm with calibrated parameters, is forced with downscaled seasonal forecasts to provide seasonal forecast of hydrological conditions.

3.6 DT Application: Extreme rainfall, temperature, and wind weather event changes in response to climate change

3.6.1 ML Model Requirements

The application inputs are sets of 2-dimensional data for each climate variable. Each of them can therefore be considered as a 2D image, where each pixel coincides with a cell of the lat-lon grid. Considering three climate variables (temperature, precipitation, and wind), the images take the form of "RGB" (Red Green Blue) images, carrying three components for each pixel when compound extreme events are analysed. In the current implementation only one variable is chosen to be analysed, which is temperature, but any variable can be chosen depending on the application. Based on this consideration, the deep learning model identified for the DT is based on convolutional layers. Given the unsupervised learning conditions, the model used for anomaly detection is a Convolutional Variational Auto-Encoder – an image-compressing/rebuilding neural network (CVAE). Its inputs are daily squares over Western Europe, with the three climate variables values for each pixel – dimension $n_{lat} \times n_{lon} \times 3$. The tool can be extended to any geographical zone. The CVAE compresses (encodes) the input with convolutional layers to a smaller latent space. Distribution parameters are sampled from the encoded space, and the image is rebuilt with symmetrically transposed convolutional layers. A backward loop assesses the loss between the original and reconstruction image. The model is trained on historical data (about 30 years), assuming that history is the "normal" situation. The trained network is then applied to projection data: when the reconstruction loss is unusually high, the situation is considered an anomaly. Post-processing these findings helps characterise the events with their duration, frequency, and intensity.

The software framework is PyTorch.

A list of used data variables is available in **Table 12**. Any variable can be used and also at any spatial resolution. In the current implementation extreme hot days were explored using the Daily Maximum Near-Surface Air Temperature.

Table 12 – Preliminary list of data variables for the DT Application

Variable name	Temporal Resolution	Spatial Resolution	Unit
Daily Maximum Near-Surface Air Temperature	daily	100km × 100km	K
Daily Precipitation	daily	100km × 100km	kg m^{-2} s^{-1}
Daily-Mean Near-Surface Wind Speed	daily	100km × 100km	m s^{-1}

3.6.2 Workflow Description

This section describes the logical flow of operations the DT needs to perform to carry out user requests.

In terms of computing workflow, the DT on extreme events follows those steps:

1. **Data Selection** Users select the relevant data to run their analysis; i.e., specify the geographical region of interest, climate reference time period (~30 years), time period of interest, season, and future climate scenarios (RCPs).
2. **Data Preprocessing** Selected data will be pre-processed so that it can be used as input to the ML model. In particular, data will be normalised with respect to the full dataset, and split into four season datasets.
3. **Model Training** The network will be pre-trained for Western Europe on a specific climate model, but for any other location or model it will have to be trained again. The training is based on 30 years of historical data. The hyperparameters are tuned to reach a trade-off between various applications.
4. **Projection Data Inference** The weights of the trained network are saved and applied to projection data of the same climate model – to avoid detecting biases between climate models as anomalies.
5. **Post-processing** The daily reconstruction errors will be analysed with respect to history or other IPCC scenarios, for each season. Percentiles and other statistical methods are used to determine the behaviour changes of historically rare events: their frequency of occurrence, duration, intensity (when relevant).
6. **Improving results** Only one climate model can be studied at a time, but variants of the same model (ensemble members) can be used simultaneously and

therefore augment available data by providing uncertainty quantification. A trained network on more data will generate more accurate results and reduce uncertainty – it can be evaluated depending on climate models and RCPs. Results from different models (and therefore different networks) can also be aggregated.

7. **Visualising** Final results can be stored as output files or visualised in the notebook interface using different Python visualisation modules, e.g., plots, maps.

The script of the core model is available on GitHub (in progress): https://github.com/cerfacs-globc/xtclim.

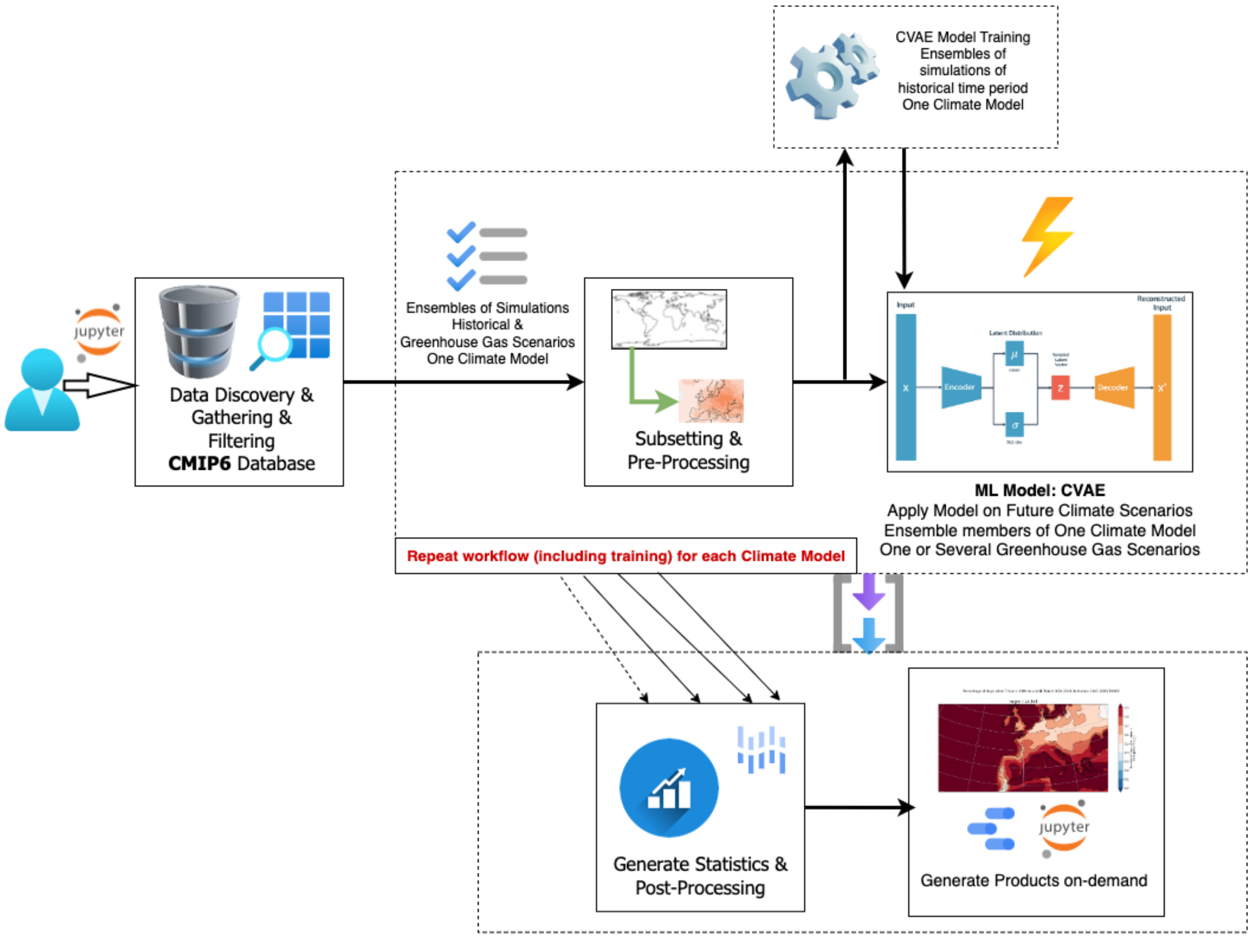

Figure 6 – Schematic overview of the main components of the DT Application: Extreme weather event changes in response to climate change. It uses a Convolutional Variational Auto-Encoder method to detect anomalies

3.7 DT Application: Flood climate impact in coastal and inland regions

3.7.1 Model Requirements

In addition to the flood inundation models and workflow described in **Section 3.4**, the flood *climate impact* DT for coastal and inland regions uses models and tools to quantify impacts on buildings, utilities, roads and accessibility under different future climate scenarios applying different mitigation and adaptation strategies.

The climate impact DT relies on an additional process-based model:

1. WFLOW: A framework for modelling 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.

Wflow simulates the volume and timing of water flow (runoff) from a catchment into a river system, based on meteorological inputs and the catchment characteristics. To do so a user has to define catchment boundaries, set up hydrological response units based on combining, e.g., soil, land cover and topography data. Simulations are forced using precipitation data, and results are affected by land use / land cover, soil types, topography and evapotranspiration rates.

Instead of using the meteorological data referred to in Section 3.4 to run the hydrological models (Wflow) and the flood inundation model (SFINCS), here climate projection data will be used to simulate flood depth and extent under future climate conditions. The requirements and workflow to arrive at the deterministic are the same as in **Section 3.4**, but using climate projection data instead of weather forecasts.

Once the deterministic flood maps have been generated, these are fed into the flood impact assessment tool (Delft-FIAT), and the Resilience Assessment and Action perspective for Critical infrastructurE (RA2CE) model.

Using the flood maps, and additional inputs such as depth-damage functions, asset locations and their maximum potential damages, Delft-FIAT derives asset-level and aggregated 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.

Similarly to Delft-FIAT, RA2CE uses the flood maps combined with road network data, road damage functions, road depth damage curves, population data, and important locations to calculate damages to road networks including the cascading effects on society due to disruptions of the infrastructure network.

Additionally for this DT, users can select and input flood mitigation and adaptation strategies such as flood walls, levees, pumps and culverts, raising properties and flood proofing properties and run scenarios to estimate the impact said strategies have on mitigating damages related to floods.

Setting up Delft-FIAT for the purposes described above requires exposure data including building footprints, roads, and asset classification data, as well as vulnerability data including depth damage curves and functions.

To set up RA2CE, additionally requires road infrastructure data as well as data on population and important locations.

An overview of data requirements to set up Delft-FIAT and RA2CE can be found in **Table 7** and **Table 10** of interTwin D7.1 [R6].

3.7.2 Workflow Description

The workflow to set up SFINCS is described in **Section 3.4.2**. For the flood climate impact DT, there are 3 additional workflows, namely:

1. Setting up the WFLOW, Delft-FIAT and RA2CE models;
2. Producing a baseline damage and impact assessment, based on flood depth and extent from running Wflow and SFINCS with future climate projection data;
3. Selecting flood mitigation and adaptation strategies and rerunning the models.

Setting up the Delft-FIAT and RA2CE models for a specific region of interest comprises the following steps:

1. Set up WFLOW
    a. The area of interest defined for the SFINCS model is used to define the upstream catchment area for WFLOW;
    b. A user defines the WFLOW model resolution;
    c. A user selects available global datasets from which to build the model. Optionally, a user can upload and use local data instead of the global data.
2. Set up Delft-FIAT
    a. A user defines a geographic region of interest;
    b. A user selects available global datasets from which to build the model.

       Optionally a user can upload and use local data instead of global data;
    c. A user checks and/or links the classification of the assets to the asset damage functions.
3. Set up RA2CE
    a. A user defines a geographic region of interest;
    b. A user selects available global datasets from which to build the model. Optionally a user can upload and use local data instead of global data;
    c. A user selects the road types to include;

   d. A user checks and/or links the road types to the road damage functions.

Once the SFINCS, Wflow, Delft-FIAT and RA2CE models have been set up, they can be run to produce baseline damage assessments under future climate scenarios as follows:

1. A user selects a time period to simulate, e.g., a specific historical event;
2. A user selects the preferred forcing data for SFINCS and Wflow, e.g.,
   a. Wflow – temperature, precipitation, and potential evapotranspiration
   b. SFINCS – in addition to precipitation, select data for tides, surges, waves, and sea level
3. A user then runs Wflow;
4. A user then runs SFINCS with river discharges from Wflow;
5. The water levels produced by SFINCS are interpolated to the digital elevation model selected by the user in the Setup SFINCS step 1c (**Section 3.4.2**), to estimate flood depths and extent;
6. A user then runs Delft-FIAT with the flood depth maps from SFINCS;
7. A user then runs RA2CE with the flood depth maps from SFINCS;
8. A user can visualise the resulting damages and impacts spatially in the Jupyter Notebook or in any GIS software.

Once the baseline damage assessments under a future climate scenario have been run, the workflow to assess the impact of selecting mitigation and adaptation strategies can be run as follows:

1. A user selects the measures and specifies the properties of the measures that they want to test;
2. A user runs a FloodAdapt module that implements the changes in the models corresponding to the measures. It is currently possible to implement measures that make changes to the SFINCS and Delft-FIAT models;
3. Depending on the choice of measures, the user must rerun the following models with the updated model data:
   a. If the user selected a flood wall, pump, levee, and/or culvert, the user must rerun SFINCS, Delft-FIAT and RA2CE with the updated models.
   b. If the user selected buyouts, flood proofing, and/or raising properties, the user can just rerun Delft-FIAT.
4. A user can visualise the resulting changes in damages and impacts spatially in the Jupyter Notebook or in any GIS software.

In the above workflows, this DT application will leverage the following capabilities from other WPs of interTwin (summarised in **Figure 7**)

- The DT will leverage capabilities developed in WP7, T7.6: Hydrological model data processing thematic module.

- To run the models using containers, this DT will rely on functionality from WP6, specifically OSCAR
- For workflow composition and execution, this DT will leverage developments from WP6, T6.1 Workflow composition and WP6, T6.4 Workflow backend, specifically the CWL workflows developed for this application
- Finally, all data and compute resources are leveraged from WP5, the DTE infrastructure including Notebooks as a Service.

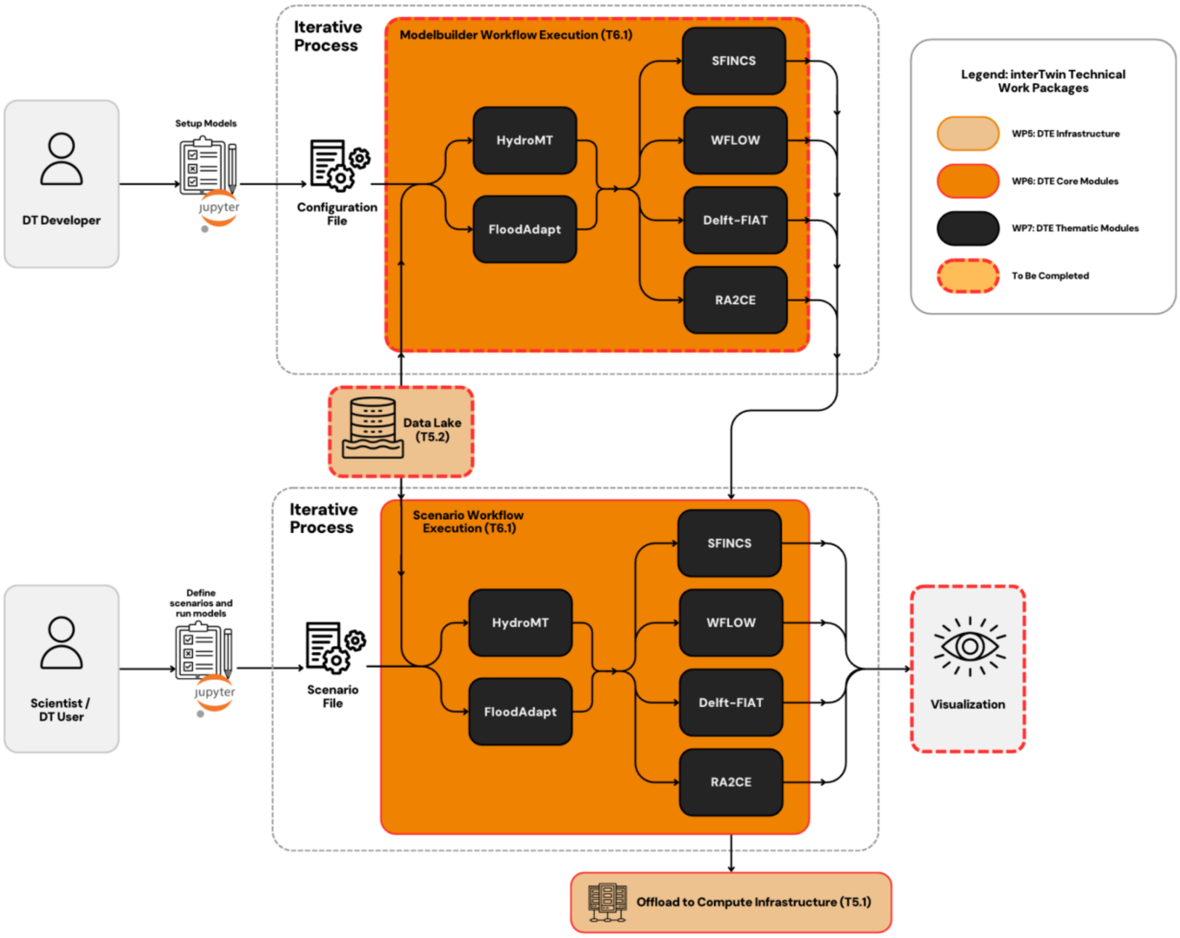

Figure 7 – Overview of the workflow components for the DT application: Flood climate impact in coastal and inland regions

4 Conclusions

This document presented the final version of the architecture design of interTwin DTs applications (WP4) from the environmental domain. It updates the first version of the capabilities presented in D4.1 **[R7]**, by refining the initial design and providing a revised version of the workflow and their links with the DTE. More specifically, D4.5 defines the different user stories and key requirements, as well as the workflow and possible beneficiaries for each DTs application. Moreover, the necessary steps that should be considered when designing an individual DT application are also outlined, together with a visual representation of the workflow and the connection with the components provided by the DTE.

As next steps the DT application development and integration with the DTE will be finalized and presented in D4.7 “Final version of the DTs capabilities for climate change and impact decision support tools including validation reports” due by the end of the project.

5 References

Reference No	Description / Link
R1	Imhoff, R., et al., Scaling point-scale (pedo) transfer functions to seamless large-domain parameter estimates for high-resolution distributed hydrologic modelling: An example for the Rhine River. Water Resources Research, 2020. 56(4): p. e2019WR026807. DOI: **https://doi.org/10.1029/2019WR026807**
R2	Schellekens, J., Van Verseveld, W., Visser, M., Winsemius, H., Euser, T., Bouaziz, L., Boisgontier, H, Openstreams/wflow:. 2021. DOI: **https://doi.org/10.5281/zenodo.4291730**
R3	Isabel Campos, Donatello Elia, Germán Moltó, Ignacio Blanquer, Alexander Zoechbauer, Eric Wulff, Matteo Bunino, Andreas Lintermann, Rakesh Sarma, Pablo Orviz, Alexander Jacob, Sandro Fiore, Miguel Caballer, Bjorn Backeberg, Mariapina Castelli, Levente Farkas, & Andrea Manzi. (2023). interTwin D6.1 Report on requirements and core modules definition (V1 Under EC review). Zenodo DOI: **https://doi.org/10.5281/zenodo.8036987**
R4	Bauer-Marschallinger, B., Cao, S., Tupas, M. E., Roth, F., Navacchi, C., Melzer, T., Freeman, V., & Wagner, W. (2022). Satellite-Based Flood Mapping through Bayesian Inference from a Sentinel-1 SAR Datacube. Remote Sensing, 14(15), Article 3673. DOI: **https://doi.org/10.3390/rs14153673**
R5	Schramm, M., Pebesma, E., Milenković, M., Foresta, L., Dries, J., Jacob, A., Wagner, W., Mohr, M., Neteler, M., Kadunc, M., Miksa, T., Kempeneers, P., Verbesselt, J., Gößwein, B., Navacchi, C., Lippens, S., & Reiche, J. (2021). The openEO API – Harmonising the Use of Earth Observation Cloud Services Using Virtual Data Cube Functionalities. Remote Sensing, 13(6), 1125. DOI: **https://doi.org/10.3390/rs13061125**
R6	Michele Claus, Alexander Jacob, Björn Backeberg, Frederique de Groen, Joost Buitink, Roel de Goede, Donatello Elia, Gabriele Accarino, Sandro Fiore, Christian Pagé, Matthias Schramm, Bernhard Raml, & Christoph Reimer. (2023). interTwin D7.1 Report on requirements and thematic modules definition for the environment domain (V1 Under EC review). Zenodo DOI. **https://doi.org/10.5281/zenodo.8036991**
R7	Muhammad Usman Liaqat, Mariapina Castelli, Donatello Elia, Gabriele Accarino, Davide Donno, Sandro Fiore, Bjorn Backeberg, Matthias Schramm, Christian Pagé, Frederique de Groen, Albrecht Weerts, Kathryn Roscoe, & Atef Ben Nasser. (2023). interTwin D4.1 First Architecture design of the DTs capabilities for climate change and impact decision support tools (Final). Zenodo. Zenodo DOI: **https://doi.org/10.5281/zenodo.10417135**
R8	O'Neill, B. C., Tebaldi, C., van Vuuren, D. P., Eyring, V., Friedlingstein, P., Hurtt, G., Knutti, R., Kriegler, E., Lamarque, J.-F., Lowe, J., Meehl, G. A., Moss, R., Riahi, K., and Sanderson, B. M.: The Scenario Model Intercomparison Project (ScenarioMIP) for CMIP6, Geosci. Model Dev., 9, 3461–3482, 2016. DOI: **https://doi.org/10.5194/gmd-9-3461-2016**

R9	Gabriele Accarino, Davide Donno, Francesco Immorlano, Donatello Elia, Giovanni Aloisio. "An ensemble machine learning approach for tropical cyclone localization and tracking from ERA5 reanalysis data." Earth and Space Science 10.11 (2023): e2023EA003106. DOI: **https://doi.org/10.1029/2023EA003106**
R10	Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J. (2018). UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In: Stoyanov, D., et al. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA ML-CDS 2018 2018. Lecture Notes in Computer Science, vol 11045. Springer, Cham. DOI: **https://doi.org/10.1007/978-3-030-00889-5_1**
R11	Emanuel, K. A., & Nolan, D. S. (2004). Tropical cyclone activity and the global climate system. In Preprints, 26th conf. on hurricanes and tropical meteorology, Miami, FL (Vol. 10). Amer. Meteor. Soc. A.
R12	Elsner, J. B., Kossin, J. P., & Jagger, T. H. (2008). The increasing intensity of the strongest tropical cyclones. Nature, 455(7209), 92–95. DOI: **https://doi.org/10.1038/nature07234**
R13	Mendelsohn, R., Emanuel, K., Chonabayashi, S., & Bakkensen, L. (2012). The impact of climate change on global tropical cyclone damage. Nature Climate Change, 2(3), 205–209. DOI: **https://doi.org/10.1038/nclimate1357**
R14	Sun, Y., Zhong, Z., Li, T., Yi, L., Hu, Y., Wan, H., et al. (2017). Impact of ocean warming on tropical cyclone size and its destructiveness. Scientific Reports, 7(1), 8154. DOI: **https://doi.org/10.1038/s41598-017-08533-6**
R15	Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. DOI: **https://doi.org/10.1007/978-3-319-24574-4_28**
R16	Cunningham, C.X., Williamson, G.J. & Bowman, D.M.J.S. Increasing frequency and intensity of the most extreme wildfires on Earth. Nat Ecol Evol 8, 1420–1425 (2024). DOI: **https://doi.org/10.1038/s41559-024-02452-2**
R17	Alonso, L., Gans, F., Karasante, I., Ahuja, A., Prapas, I., Kondylatos, S., Papoutsis, I., Panagiotou, E., Mihail, D., Cremer, F., Weber, U., & Carvalhais, N. (2024). SeasFire Cube: A Global Dataset for Seasonal Fire Modeling in the Earth System (0.4) [Data set]. Zenodo DOI: **https://doi.org/10.5281/zenodo.6834584**
R18	Jones, M. W., Abatzoglou, J. T., Veraverbeke, S., Andela, N., Lasslop, G., Forkel, M., et al. (2022). Global and regional trends and drivers of fire under climate change. Reviews of Geophysics, 60, e2020RG000726. DOI: **https://doi.org/10.1029/2020RG000726**