Copernicus Climate Change Service

Final Service Report

Issued by: VITO / Nele Veldeman

Date: 15/12/2025

Ref: C3S2_D461BE-1.2.1.2_202512_Final Service Report

Official reference number service contract: 2024/C3S2_461-1_BE_VITO

Contributors

VITO
Nele Veldeman
Hendrik Wouters
Niels Souverijns
Filip Schouwenaars
Denis Caeyers
Robin Houdmeyers

Belgian Climate Centre
Rozemien De Troch

Change Log

Version	Date	Description
v0.1	21/10/2025	First draft = Prototype Service Report (D461BE-1.2.1.1) by Nele Veldeman (VITO)
V0.2	30/10/2025	Revision by Rozemien De Troch (BCC)
V0.2	24/11/2025	Inclusion of validation: description and results by Hendrik Wouters (VITO)
V0.3	09/12/2025	Revision by Nele Veldeman (VITO)
V0.4	11/12/2025	Revision by Robin Houdmeyers (VITO-IT)
V0.5	12/12/2025	Revision by Hendrik Wouters (VITO)
v1.0	15/12/2025	D461BE-1.2.1.2 (Nele Veldeman)

Table of Contents

Introduction

This report documents the **pilot implementation** of the Climate Service for Vector-Borne disease (VBD) management in Belgium. While the ***Service Specification & Implementation Plan*** (D461BE-1.1.2.1, June 2025) detailed the conceptual and technical design, the ***Prototype Service Report*** (D461BE-1.2.1.1, October 2025) reports on the realization of the prototype service, including implemented workflows and integration at Sciensano. The current document, i.e., the ***Final Service Report*** (D461BE-1.2.1.2, December 2025), details the realization of the **final service**. In comparison to the Prototype Service Report, it includes **additional information on service validation and provides further elaboration on the technical components**.

The service was co-developed by VITO, the Belgian Climate Centre (BCC), and Sciensano. It aims to provide **daily, high-resolution (100 m) climate maps and derived indicators** relevant for ticks and exotic mosquitoes, to support disease surveillance, early warning, and long-term public health studies of Sciensano and other Belgian public authorities from the health sector.

1 Service specification

1.1 Introduction

Climate change has profound implications for public health, particularly in the context of **vector-borne diseases (VBDs)**, which are transmitted by arthropod vectors such as mosquitoes and ticks. These diseases, including Lyme borreliosis, dengue, and chikungunya, are influenced by various environmental factors, with **climate being a key driver** of vector distribution, abundance, and activity. Rising temperatures, changing precipitation patterns, and increasing extreme weather events are creating more favorable conditions for vector populations to thrive and expand their geographical range.

Despite the clear link between climate and VBDs, **real-time and structured climate data integration into public health monitoring systems remains limited in Belgium**. The current use of climate information at Sciensano, Belgium's national public health institute, is **project-based, retrospective, and often reliant on ad-hoc data retrieval processes**. This lack of real-time, automated access to climate data presents a critical gap in the ability to anticipate vector activity, improve risk assessments, and implement timely public health interventions.

To address this need, VITO and the Belgian Climate Centre (BCC), in collaboration with Sciensano, are developing a **Climate Service tailored to the surveillance and management of VBDs in Belgium**.

This service will provide:

- **Automated daily climate data retrieval** from the Copernicus Climate Change Service (C3S) and Royal Meteorological Institute of Belgium (RMIB) to support real-time monitoring.
- Downscaled **climate indicators at high spatial resolution (100 m)** to integrate urban effects and provide location-specific insights.
- A **structured and user-friendly data pipeline** that seamlessly integrates with Sciensano's existing epidemiological workflows.

The proposed Climate Service will bridge the gap between climate science and public health by:

- **Improving early warning systems** – Enabling Sciensano to anticipate and communicate seasonal peaks in tick and mosquito activity, based on near-real-time climate conditions.
- **Enhancing data accessibility** – Automating climate data retrieval and processing for operational integration into Sciensano's surveillance systems.
- **Increasing accuracy in risk assessments** – Providing climate-driven insights to better predict vector emergence, hotspots, and high-risk periods.
- **Supporting long-term epidemiological studies** – Facilitating the assessment of climate change's impact on vector distribution and disease trends over time.

By leveraging state-of-the-art climate modeling, high-resolution datasets, and automation, this Climate Service will strengthen Belgium's capacity to manage VBDs proactively and provide a scalable use case for climate-health services in Europe.

1.2 Overview of the services' conceptual design

In the ***Service Specification and Implementation Plan*** (D461BE-1.1.2.1, June 2025), the Climate Service (CS) for vector-borne disease management was conceptually designed as a **dual-flow system combining an AI-based workflow and a regression-based workflow**, see Figure 1. Both workflows were intended to operate independently on a daily base, ensuring continuous climate data coverage despite the five-day latency of the ERA5 dataset. The original description is reproduced from D461BE-1.1.2.1 in Text Box 1.

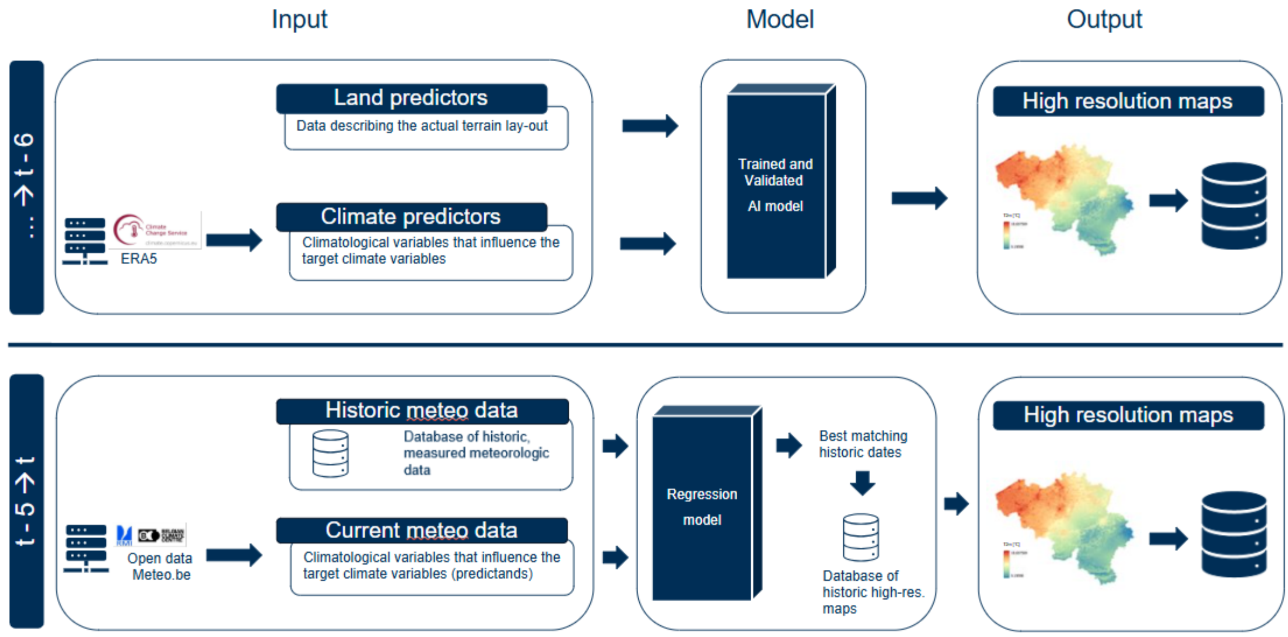

Figure 1: Schematic overview of the Climate Service for vector-borne disease management.

In the ***Prototype Service Report*** (D461BE-1.2.1.1, October 2025), changes to the design introduced during the Pilot Development phase were described. Essentially,

- the overall design philosophy remained valid,
- the **AI-based workflow (Flow 1,** Figure 1 – top) was implemented as originally designed,
- and the **regression-based workflow (Flow 2,** Figure 1 – bottom), relying on a **pre-generated archive of high-resolution maps**, was replaced by an **on-the-fly map generation** mechanism.

The **final conceptual design** is presented in Figure 2. **Flow 2** first identifies the most similar historical day by comparing recent Automatic Weather Station (AWS) observations with the multi-year archive of AWS data. However, instead of retrieving a corresponding map from a static pre-generated database (Design phase, Figure 1, Text Box 1) **the system now re-runs Flow 1 (the AI-based model) for the selected analogue date.**

Text Box 1 - Original Conceptual Design

*The system comprises **two distinct data processing and modeling workflows**, both operating on a daily base, to deliver high-resolution (100 m) maps for different climate variables for Belgium. These workflows function independently.*

***The first workflow** (Figure 1, top) applies to historical dates up to six days before the current date (t-6) and, in operational mode, will process data for day 't – 6'. It generates output using a trained and validated AI model that leverages ERA5 climate data from the Copernicus Climate Change Service (C3S).*

***The second workflow** (Figure 1, bottom) operates for the period from five days before the current date (t-5) up to today (t). Instead of AI-based processing, it selects the most relevant high-resolution climate maps from a pre-configured archive, based on a regression between (near) real-time and historic climate data from Automatic Weather Stations (AWS) of the Royal Meteorological Institute of Belgium (RMIB).*

Source: Service Specification & Implementation Plan (D461BE-1.1.2.1, June 2025).

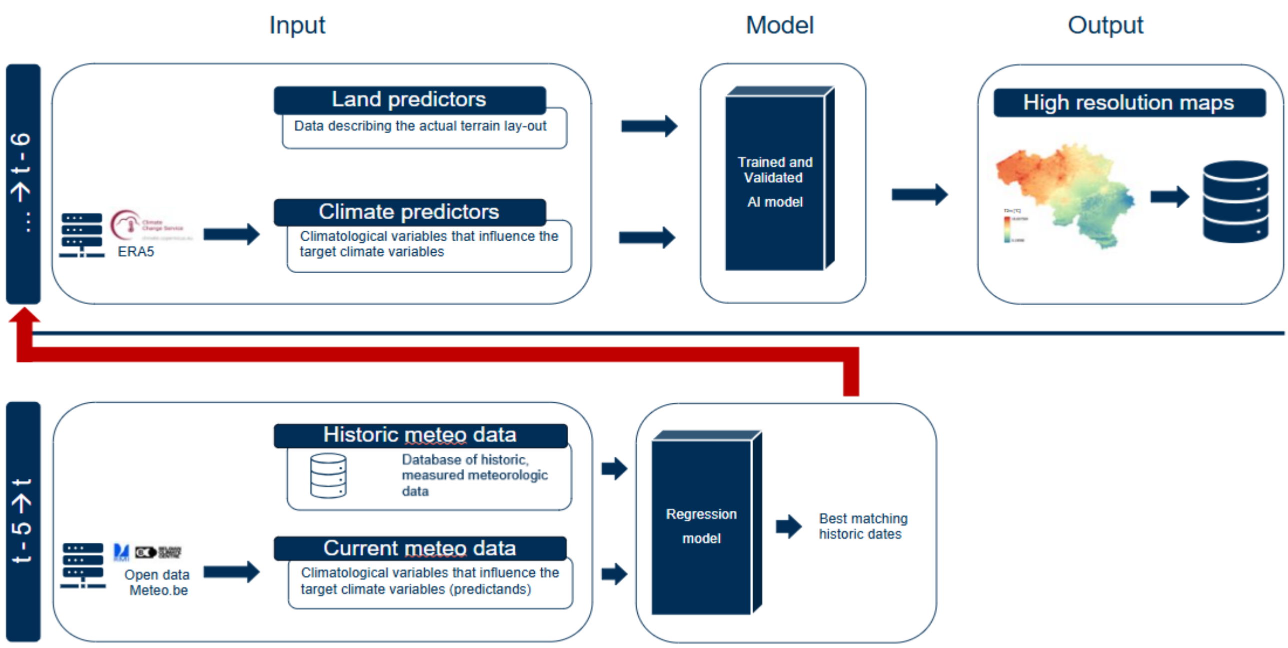

Figure 2: Adapted schematic overview of the Climate Service for vector-borne disease management. **The resulting output** remains identical to the original approach—a high-resolution climate map for the analogue day—but it is now **generated dynamically** rather than **retrieved from storage**.

This modification offers **several advantages**:

- **Reduced data storage requirements**, as no full archive of daily high-resolution maps needs to be maintained.
- **Greater consistency** between both flows, since all maps (whether for t–6 or t–1) are now generated using the same AI model and processing chain.
- **Simplified maintenance**, because improvements or retraining of the AI model automatically propagate to both workflows.

In summary, the conceptual structure defined in the design phase remains valid—**two complementary workflows ensuring continuous climate coverage**—but the regression-based flow has transitioned from a database retrieval approach to an **on-demand AI generation approach**, improving efficiency, coherence, and long-term sustainability of the operational service.

1.3 Workflow based on AI model leveraging ERA5 data from C3S

A detailed overview of the first operational workflow (Figure 2, top), based on an **AI model leveraging ERA5 data from C3S**, was presented in the *Service Specification & Implementation Plan* (D461BE-1.1.2.1, June 2025). The schematic overview is copied below (Figure 3), and a concise summary of the concept is provided in Text Box 2. Further details on the scientific concept and model architecture are available in D461BE-1.1.2.1.

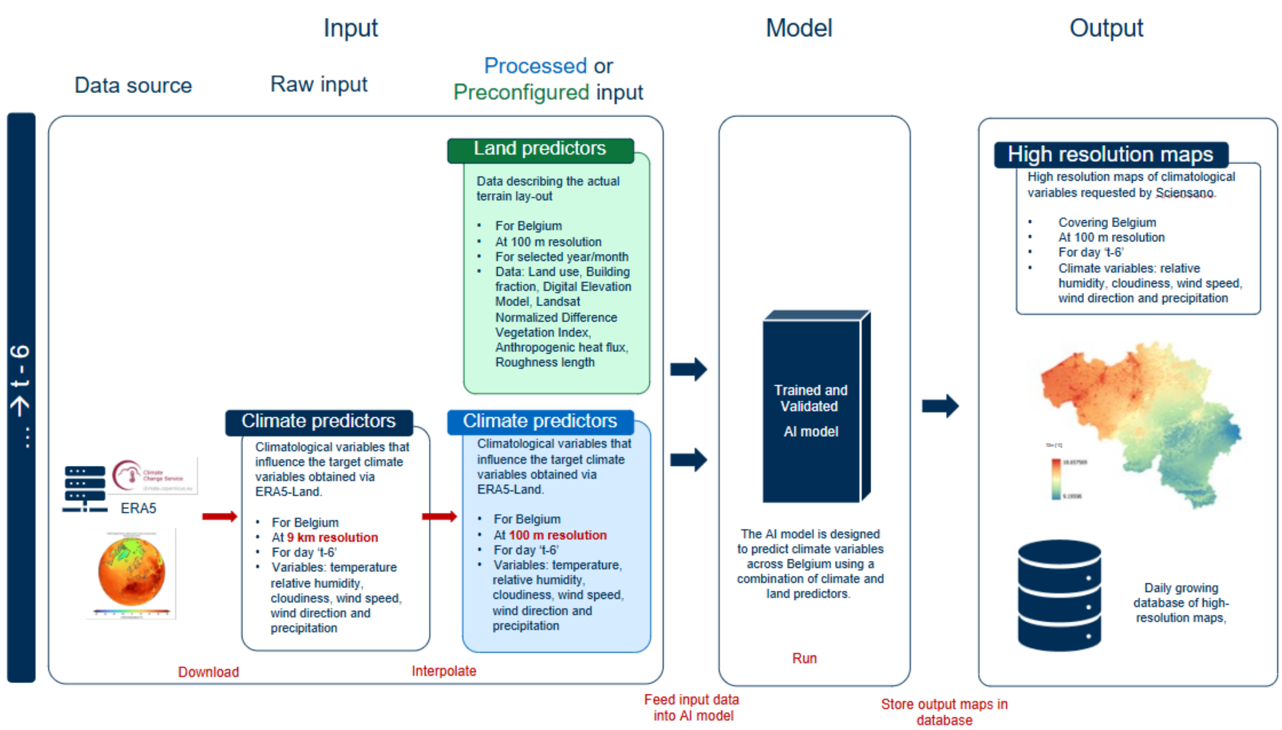

Figure 3: Comprehensive breakdown of the operational AI model flow, utilizing ERA5 data.

Text Box 2. Original concept of the AI-based workflow (from D461BE-1.1.2.1)

The AI-based workflow forms the core of the Climate Service. It was designed to generate high-resolution (100 m) climate maps for Belgium daily, using a trained and validated machine-learning model that downscales Copernicus ERA5 data.

Model concept
- The model predicts multiple climate variables (e.g. temperature, humidity, wind, precipitation) by combining climate predictors and land predictors.
- Training data consist of simulated climate variables from 13 representative UrbClim patches (predictands) and corresponding predictors.
- The model learns spatial relationships between predictors and predictands and generalizes them to national scale.

Input data
- Land predictors: elevation (Copernicus DEM), land cover (Corine LC), NDVI (Sentinel), anthropogenic heat flux, etc.
- Climate predictors: ERA5 reanalysis data (temperature, humidity, wind, radiation, precipitation).

Operational use
- The model runs automatically for day t – 6, processing ERA5 data and preconfigured land predictors on-the-fly.
- The workflow produces daily 100 m-resolution climate maps for the variables requested by Sciensano.

Output
Validated, high-resolution maps of climatological variables for Belgium, suitable for epidemiological modelling and long-term public-health applications.

Source: Service Specification & Implementation Plan (D461BE-1.1.2.1, June 2025).

The **actual implementation** of this AI-based model flow, including training, validation, and operational integration, is described in detail in the *Implementation* section of this report (Section 2).

1.4 Workflow based on regression analysis of AWS data

1.4.1 Design based on retrieving pre-generated maps

A detailed description of the regression-based workflow was included in the *Service Specification & Implementation Plan* (D461BE-1.1.2.1, June 2025). That document presented a concept in which the regression module would identify the most similar historical day based on AWS observations and subsequently **retrieve the corresponding high-resolution map** from a **pre-generated database** of daily climate maps produced by the AI-based workflow. This concept ensured continuity of near real-time climate information for the five most recent days (t – 5 to t), for which ERA5 data are not yet available. The original description is summarized in Text Box 3, and the conceptual schematic from the service specification report is reproduced in Figure 4.

Text Box 3. Original concept of the regression-based workflow (from D461BE-1.1.2.1)

The regression-based workflow was designed to ensure continuous near real-time climate data availability for the most recent days (t – 5 to t), for which ERA5 data are not yet accessible.
In the conceptual design, the workflow operated independently from the AI-based flow and relied on the following key steps:

Analogue-day selection: *recent meteorological observations from the Royal Meteorological Institute's AWS network were compared with a multi-year archive of AWS data to identify the historical day most like current conditions.*

Retrieval of high-resolution maps: *once the analogue day was identified, the corresponding high-resolution climate map was retrieved from a pre-generated database of daily maps that had been produced in advance using the AI-based workflow.*

Map assembly and output generation: *retrieved maps were merged and formatted to provide seamless spatial coverage for Belgium at 100 m resolution, forming the near real-time component of the climate service.*

Source: Service Specification & Implementation Plan (D461BE-1.1.2.1, June 2025).

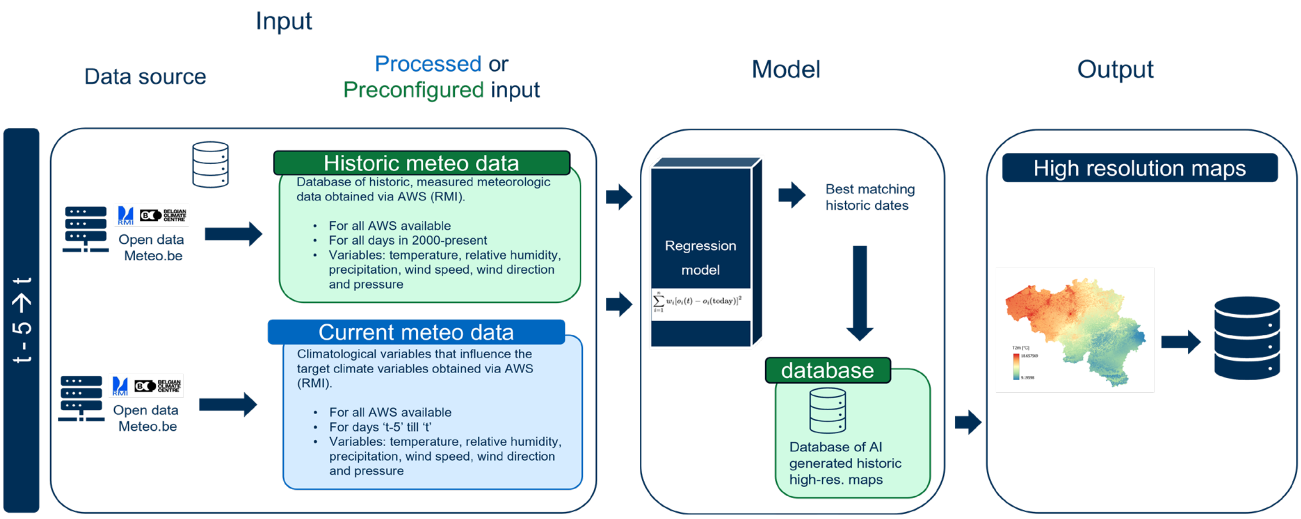

Figure 4: Detailed overview of the AWS data-based regression model flow.

1.4.2 Updated implementation: on-the-fly map generation

During the pilot development, the overall logic of the regression-based approach was retained — the system still **identifies the analogue date that best represents current meteorological conditions** — but the implementation strategy was fundamentally improved.

Instead of relying on a pre-generated database of historic maps, the current implementation performs **on-the-fly generation of the analogue-day maps by dynamically re-running the AI-based workflow for the selected date**. This modification ensures that both flows now share the same code base, model version, and data-processing chain. The updated schematic is shown in Figure 5.

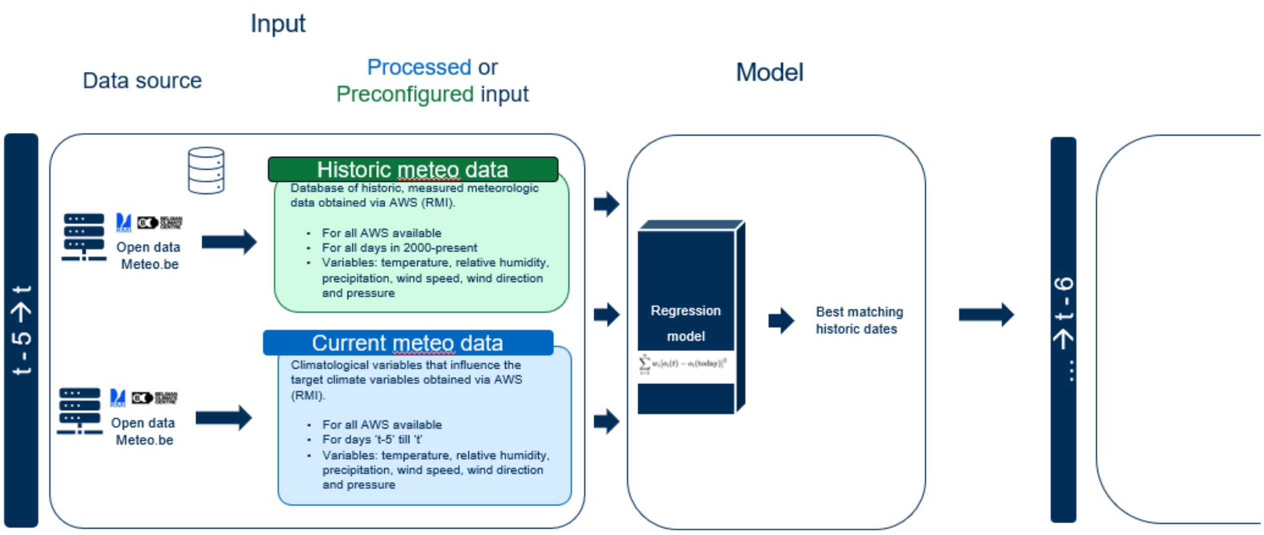

Figure 5: Updated schematic of the regression-based workflow, showing on-the-fly generation of the analogue-day map using Flow 1.

2 Service Implementation

2.1 Introduction

The implementation phase translated the conceptual design of the Climate Service into **a functioning service, integrating the AI-based and regression-based workflows within a unified processing framework**. All elements described in the *Service Specification and Implementation Plan (D461BE-1.1.2.1)* were realized, with refinements introduced during the pilot phase to improve performance, consistency, and maintainability.

The **service implementation** focused on:

- Developing, training, and validating the AI-based model (Flow 1).
- Establishing automated data pipelines for ERA5 and AWS data.
- Implementing the revised regression-based flow (Flow 2) with on-the-fly AI map generation.
- Integrating both flows into the CLIMSERVE Python package for deployment in Sciensano's POSIT Enterprise environment.

2.2 Workflow based on AI model leveraging ERA5 data from C3S (Flow 1)

A schematic breakdown of the operational AI model flow was presented in Figure 3. The core of the flow **is a trained and validated AI model** that predicts high-resolution (100 m) daily climate variables for Belgium, using a combination of static land predictors and dynamic ERA5 climate predictors (~25 km). Training is performed on UrbClim-derived daily climate fields that represent the fine-scale interactions between land surface characteristics and atmospheric processes.

2.2.1 The UrbClim model

The UrbClim model forms the scientific backbone of the AI-based workflow. UrbClim is an **urban boundary layer model** designed to simulate local meteorological conditions at high spatial resolution, while maintaining computational efficiency. UrbClim explicitly models the lowest 1.2 km of the atmosphere (De Ridder et al., 2015) and couples a **land-surface scheme with simplified urban physics** to a **3D atmospheric boundary-layer module**, thereby capturing dynamic interactions between the land surface and the atmosphere. The model is driven by hourly **ERA5 synoptic forcing fields**, ensuring meteorological consistency with large-scale conditions. The land-surface scheme, based on the soil–vegetation–atmosphere transfer model of De Ridder and Schayes (1997), represents energy and moisture exchanges among soil, vegetation, built surfaces, and the atmosphere. Each 100 m grid cell is characterized by a composition of vegetation, bare soil, and urban fractions, each parameterized by albedo, emissivity, roughness length, heat capacity, and other properties derived from detailed land-surface descriptors. This physically based framework allows UrbClim to simulate fine-scale meteorological variables (temperature, humidity, wind, pressure, precipitation) with high realism, particularly in heterogeneous environments such as urban areas. A schematic representation of the UrbClim model structure is shown in Figure 6.

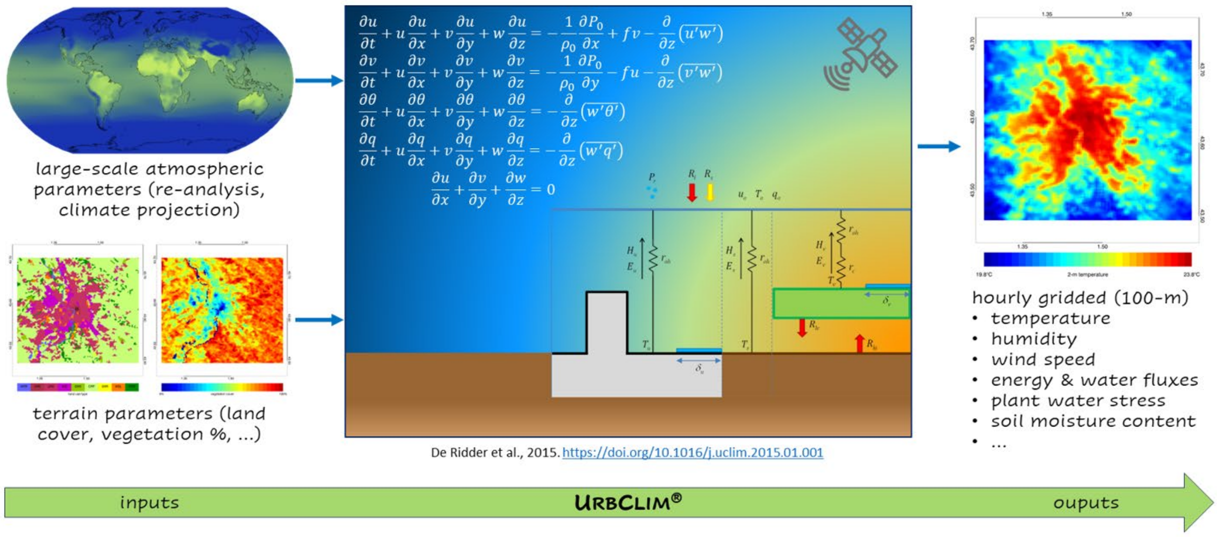

Figure 6: Schematic representation of the UrbClim model (after De Ridder et al., 2015).

2.2.2 UrbClim simulations for 13 Belgian patches

To train (and validate) the AI model, **UrbClim simulations** were performed **for 13 representative patches across Belgium** (Figure 7), covering both urban and rural environments and a wide range of climatic and geographic conditions.

1. Antwerp
2. Ghent
3. Charleroi
4. Liège
5. Brussels
6. Polders
7. Kempen
8. Haspengouw
9. Fagne-Famenne
10. Lotharingen
11. Ardennen
12. Westhoek
13. Henegouwen

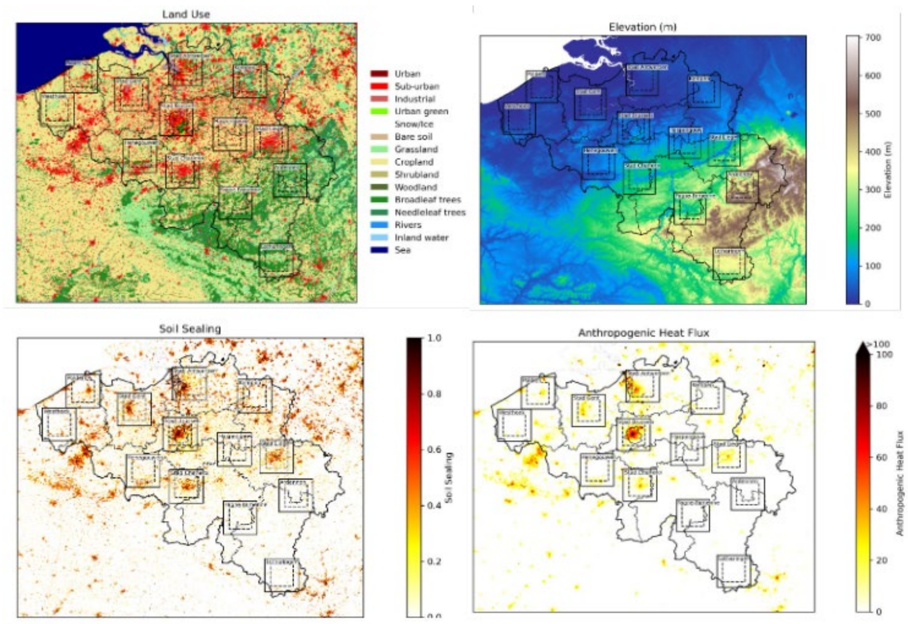

Figure 7: Selection of 13 UrbClim patches across Belgium, ensuring environmental and climatic diversity.

The first five urban patches align with the **“100 European Cities”** project (Lauwaet et al., 2022, *Urban Climate* 54, 101850), where UrbClim simulations were previously conducted for 2008–2017. These existing datasets provided valuable benchmarking for model verification.

Eight additional patches were selected to capture the full range of Belgian environmental conditions — from the flat coastal plains (Polders) to the hilly Ardennes — ensuring that the AI model would be trained on data representative of the country’s spatial heterogeneity.

Selection criteria included:

- Urban vs. rural gradients
- Coastal vs. inland climate zones
- Flat vs. elevated terrain
- Variability in land-cover classes and vegetation density

For each patch, UrbClim simulations were run for the **period 2010–2020**, generating daily maps for temperature, relative humidity, wind speed and direction, surface pressure, and precipitation. These outputs serve as the **predictand** dataset for training the AI model and/or as validation data set (see 2.2.5).

2.2.3 Input data for the AI model

The input data required by the AI model is referred to as **predictor data**. Predictors are (input) variables that influence or explain the target climate variables. In this case, they include **land predictors and climate predictors**, which serve as explanatory variables for the climate variables to be computed.

2.2.3.1 Land predictors (static)

Land predictors describe **the actual terrain lay-out of every (100 m) grid cell** of the output grid, including characteristics such as elevation, soil type, and land cover. A detailed overview of the land predictor data together with meta-data is given in Table 1 [Source: Lauwaet, D., et al., High resolution modelling of the urban heat island of 100 European cities, Urban Climate, 54, 101850 and updates* after publication thereof].

All datasets were pre-processed to 100 m resolution on a unified Belgian grid and harmonized through cropping, resampling, and interpolation, ensuring consistency with UrbClim’s physical representation.

Note that the pre-configured land predictor data is a **static dataset**, used consistently for every run in the operational service, regardless of the date. Any future updates to this dataset will require re-training of the AI model.

Table 1: Overview of required land predictor data and meta-data. Information marked with an asterix differs from the above publication as an update took place after publication of the paper.

Land predictor	Data source	Year	Spatial Resolution	Temporal Resolution
Land use - Spatial distribution of land use types	Corine Land Cover	2018*	100m	year
Building fraction - percentage of urban land cover in each grid cell	Global Human Settlement Layer*	2018*	10m*	year
Digital Elevation Model (DEM) – Terrain heights	Copernicus DEM*	https://dataspace.copernicus.eu/explore-data/data-collections/copernicus-contributing-missions/collections-description/COP-DEM (different sources from different times)	30m	-
Sentinel* Normalized Difference Vegetation Index (NDVI) – amount of vegetation in each grid cell	Sentinel derived from OpenEO*	2019-present*	10m*	Estimated for each month of the year separately
Soil sealing – Imperviousness Density (i.e. artificial sealed surfaces) in the range from 0% to 100% for the 2021 reference year	Copernicus imperviousness density https://doi.org/10.2909/34ef6334-d432-4041-a3da-67e156d6501d	2021	10m	year

2.2.3.2 Climate predictors (dynamic)

Climate predictors represent **dynamic atmospheric conditions that vary over time**, such as temperature, precipitation, wind speed, etc. The following variables were included: temperature, relative humidity, cloudiness (downward shortwave radiation), wind speed and direction, and precipitation.

ERA5 data is retrieved daily, interpolated to 100 m, and aligned with the corresponding predictor grid.

2.2.4 AI model architecture: 2D Neural Network

To emulate UrbClim's fine-scale output efficiently, a **2D Neural Network, UrbNet,** was developed.

This architecture was chosen for its capacity to capture complex, non-linear spatial relationships between predictor variables and climate outputs. The model operates on gridded data and thus retains the spatial context of meteorological and land-surface features. During training, the UrbNet model learns from the paired UrbClim (predictand) and predictor datasets, identifying the functional relationships that govern climate variability across Belgium. A schematic overview of the network architecture is shown in Figure 8.

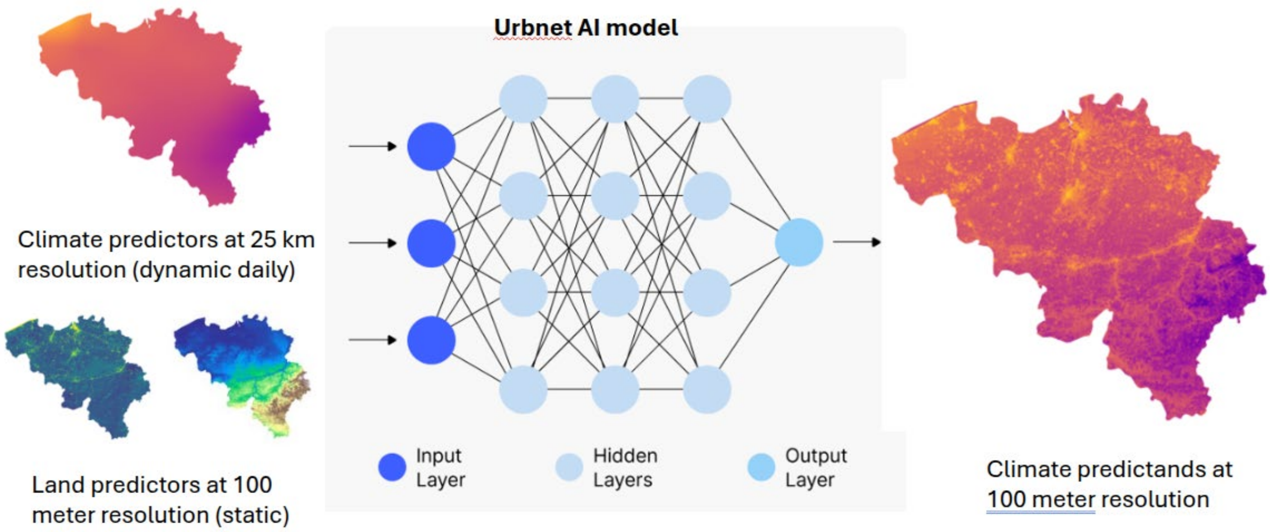

Figure 8: Schematic visualisation of the 2D Neural Network architecture applied in this study.

Key architectural features:

- **Input layer:** concatenates static (land) and dynamic (climate) predictor maps for 9 training patches (see also 2.2.5).
- **Hidden layers:** multiple convolutional layers operating at different spatial scales to represent local and advective processes.
- **Activation functions:** non-linear functions applied after each convolutional layer to capture non-linear dependencies.
- **Output layer:** predicts high-resolution (100 m) daily fields of target climate variables (e.g. temperature, humidity, wind).
- **Loss function:** mean squared error between predicted and UrbClim-simulated values.
- **Optimization:** adaptive gradient-based optimization with early stopping to prevent overfitting.
- **Batch Normalization**: it accelerates and stabilizes neural network training by normalizing the outputs of intermediate layers

This architecture allows the AI model to infer realistic high-resolution climate fields from coarse-scale ERA5 predictors while preserving spatial consistency.

The AI model is developed in Python and uses the PyTorch package as a basis. To deal with the high amount of input data, a Just-In-Time strategy is applied, allowing for efficient memory mapping. The training of the model is executed on a GPU.

A thorough testing of the quality of the 2D Neural Network in reproducing the UrbClim model results has been executed. Based on these results, the following input layers were retained to provide accurate information for the 2D Neural Network to reproduce UrbClim results. The final data input channels include static land predictors at 100 meter resolution:

- Digital Elevation Model
- Building fraction
- NDVI (monthly average)
- Land use classes (each land use class is considered as a separate channel for the neural network with values of 0 or 1)

and dynamic climate predictors at 25 km resolution:

- Lowest model level ERA-5 temperature, relative humidity, wind speed, and wind direction components (northing and easting)
- Surface-level ERA-5 for surface downwards shortwave radiation

The diurnal timing of climate predictor depends on the intended predictand:

- For predictand 'daily average temperature', daily average climate predictors are taken for (coarse) temperature, wind speed and direction components, and total downwards shortwave radiation.
- For 'mean relative humidity' predictand, we also add (coarse) mean relative humidity as climate predictor.

- For predictand 'daily maximum (minimum) temperature', the climate predictors (coarse) daily maximum (minimum) temperature are taken but also wind speed and direction components on the moment when hourly temperature reaches a maximum (minimum), and the total downwards short-wave radiation is taken from the same day (previous day; since the short-wave radiation of the same day isn't expected to influence the daily minimum temperature that usually occurs before sunrise).

2.2.5 AI model training and validation

A detailed schematic overview of the training and validation process required to develop the trained and validated AI model is presented in Figure 9.

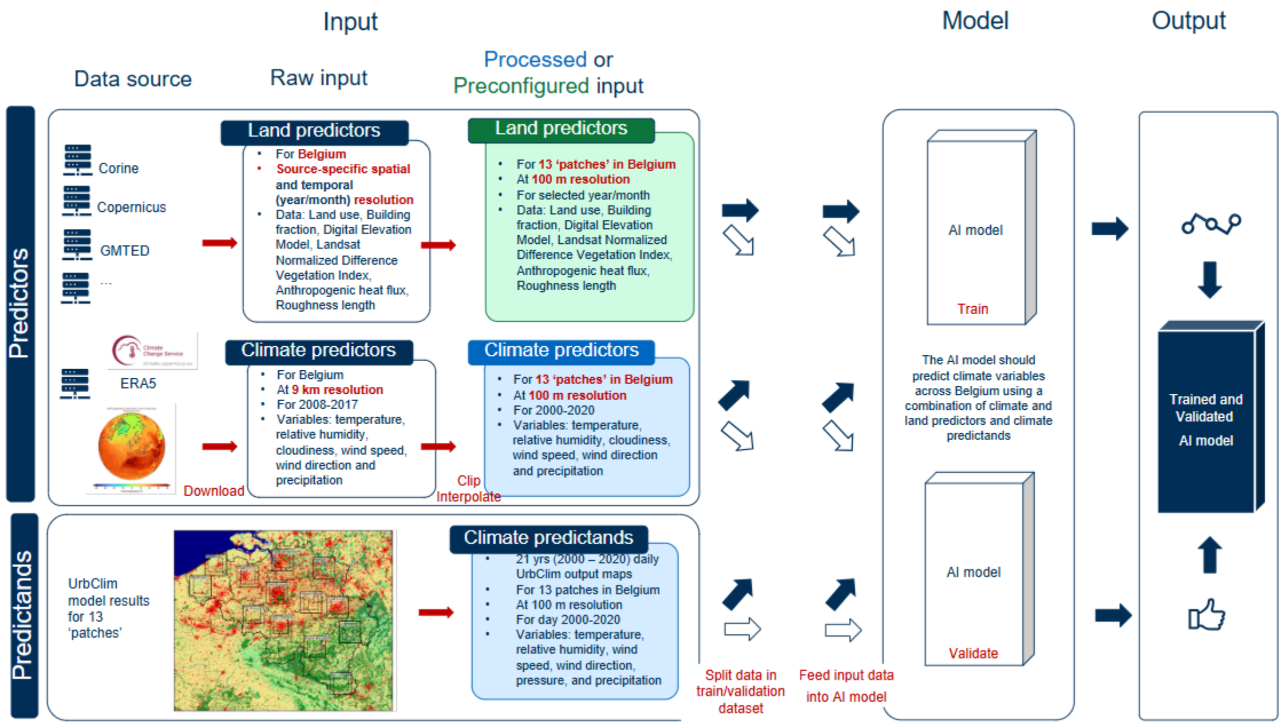

Figure 9: Schematic overview of the training and validation of the AI model, core of the workflow leveraging Copernicus ERA5 data.

Briefly stated, the training and validation dataset consists of UrbClim data for 13 Belgian patches for the 2010–2020 period (predictands), along with corresponding (land and climate) predictor data (see respectively 2.2.2 and 2.2.3). The model learns the relationships between predictors and predictands from (a selection of) these patches and then applies this knowledge to generate high-resolution climate maps for the entire country. By leveraging spatial patterns in the input data, the AI model enables area-wide climate estimations beyond the original training locations.

2.2.5.1 Training and validation strategy for the AI model

To train and validate the AI model, a representative subset of the Belgian domain was selected, both spatially and temporally.

Spatial domain selection.
Thirteen patches were chosen to capture the diversity of Belgian landscapes, ensuring that the model is exposed to a broad range of climatic and geographical conditions. The selection covers contrasts such as urban vs. rural, flat vs. elevated, and coastal vs. inland environments (see Section 2.2.2). This spatial heterogeneity was considered essential for obtaining an AI model that generalizes well across Belgium.

Temporal domain selection.
A 11-year period (2010–2020) was used. For each of the 13 patches, full UrbClim simulations were performed for all days in this period, providing a consistent and high-quality dataset for model development.

Training and validation datasets.
The UrbClim output for the 13 patches was split into **two subsets** following a commonly used 70%–30% division for training-validation (https://the-examples-book.com/tools/data-modeling-cross-validation-train-valid-test), while ensuring diversity in terms of spatial coverage, land-use characteristics and availability of measurement data.

The selected patches are (see also Figure 10):

- **Training patches:** Ardennen, Brussels, Charleroi, Haspengouw, Henegouwen, Kempen, Liège, Polders, and Westhoek
- **Validation patches:** Antwerp, Fagnes, Gent, and Lotharingen

This approach ensured that both datasets remain representative of Belgian variability while still allowing the model to be evaluated on regions unseen during training.

Predictors and predictands.
To train the AI model, the same predictor fields used to run UrbClim (see Section 2.2.3) were supplied as model inputs. The corresponding UrbClim outputs for the selected training patches served as predictands. In other words, the AI model is trained to reproduce UrbClim-like high-resolution climate fields from the same input information that drives the UrbClim simulations.

Validation strategy.
Validation was performed in three complementary pathways:

1. **Comparison with UrbClim output in the validation patches.**
   For the four validation patches, the AI-based results (UrbNet) were compared against the UrbClim simulations. This assesses the ability of the AI model to reproduce UrbClim dynamics in regions not included in the training set.

A methodological nuance is that the UrbClim output from all 13 patches was also used during model training to optimise the internal model fit and reduce overfitting, meaning that this validation is not strictly independent.

2. **Comparison with AWS observations in the validation patches.**
   For the four validation patches, the AI-based results (UrbNet) were compared against Automatic Weather Station (AWS) observations in the patches. Location of the measurement stations is presented in Figure 10 (within the validation patches (blue boxes)).

3. **Comparison with AWS observations outside the 4 validation patches.**
   To ensure a fully independent evaluation, an additional validation step was included: AI-based results were compared against Automatic Weather Station (AWS) observations located outside the validation patches. This provides an independent benchmark that is unaffected by the internal UrbClim-based training/validation split.

Together, these validation pathways provide a robust assessment of the AI model's performance, combining internal consistency (against UrbClim) and real-world accuracy (against AWS measurements).

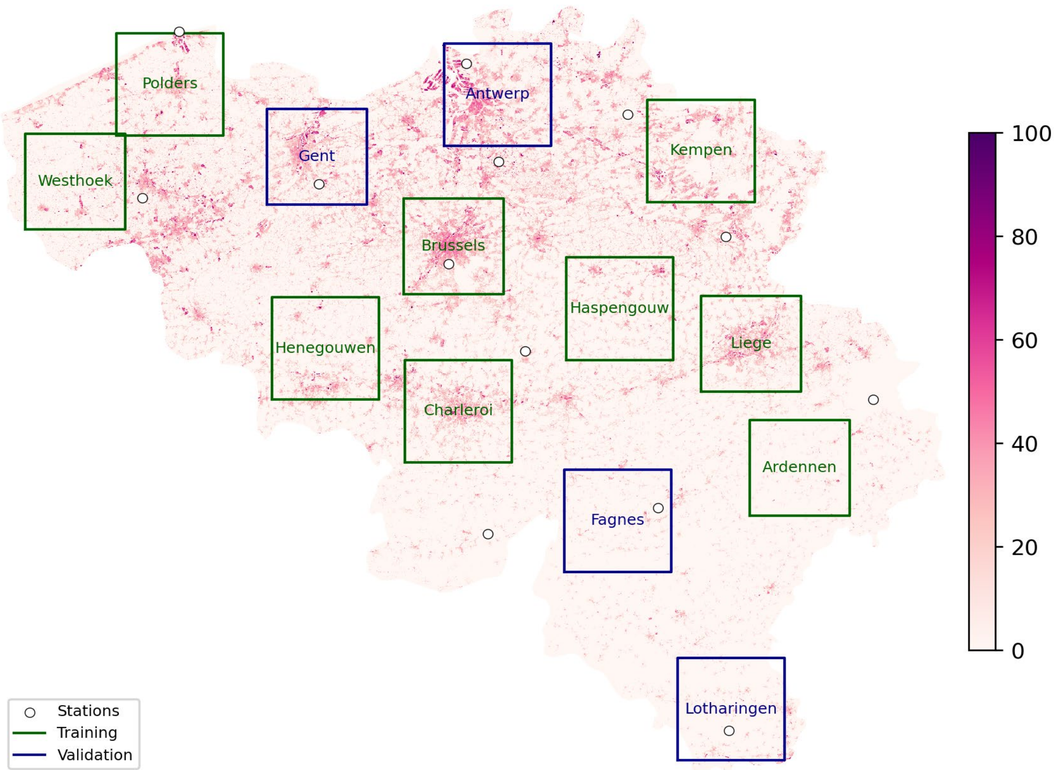

Figure 10: Overview of patches for training (green) and validation (blue), and of locations of automatic weather stations (dots). The legend indicates the percentage buildings, which is one of the high-resolution input parameters of the AI model, UrbNet.

2.2.5.2 Validation results

For the validation, the UrbNet AI model was applied over Belgium and evaluated against UrbClim output and/or observations from Automatic Weather Stations (AWS) available on opendata.meteo.be for the period 2010–2020. The results are presented in Figure 11 (daily minimum and maximum temperature) and Figure 12 (daily mean temperature and relative humidity).

Validation within the validation patches

Validation within the four validation patches (Gent, Antwerpen, Fagnes and Lotharingen) includes two complementary evaluation steps, corresponding to validation pathways 1 and 2.

Pathway 1: Consistency with UrbClim in the validation patches

Pathway 1 assesses whether UrbNet reproduces the behaviour of UrbClim in regions not used during training. In the left panels of Figure 11 and Figure 12, UrbClim outputs are compared with AWS observations inside the validation patches. In the middle panels, UrbNet outputs are compared with the same AWS observations. By comparing the left and middle panels, we observe that:

- The performance of UrbNet relative to AWS observations closely matches that of UrbClim.
- This holds for all considered variables (daily minimum and maximum temperature, daily mean temperature, and relative humidity).

This demonstrates that UrbNet succeeds in replicating the UrbClim model dynamics in the validation patches, confirming that the AI model generalises well beyond the training regions.

Pathway 2: Direct comparison of UrbNet with AWS within the validation patches

Pathway 2 focuses specifically on the middle panels of Figure 11 and Figure 12, where UrbNet outputs are directly compared with AWS measurements in the validation patches. These results show that:

- UrbNet accurately reproduces the observed spatial and temporal variability captured by the AWS data.
- The model performs reliably for all four validation regions and across all meteorological variables.

Together, the pathway 1 and pathway 2 analyses confirm that the AI model performs consistently and robustly within the validation patches—both in comparison to UrbClim and to real-world measurements.

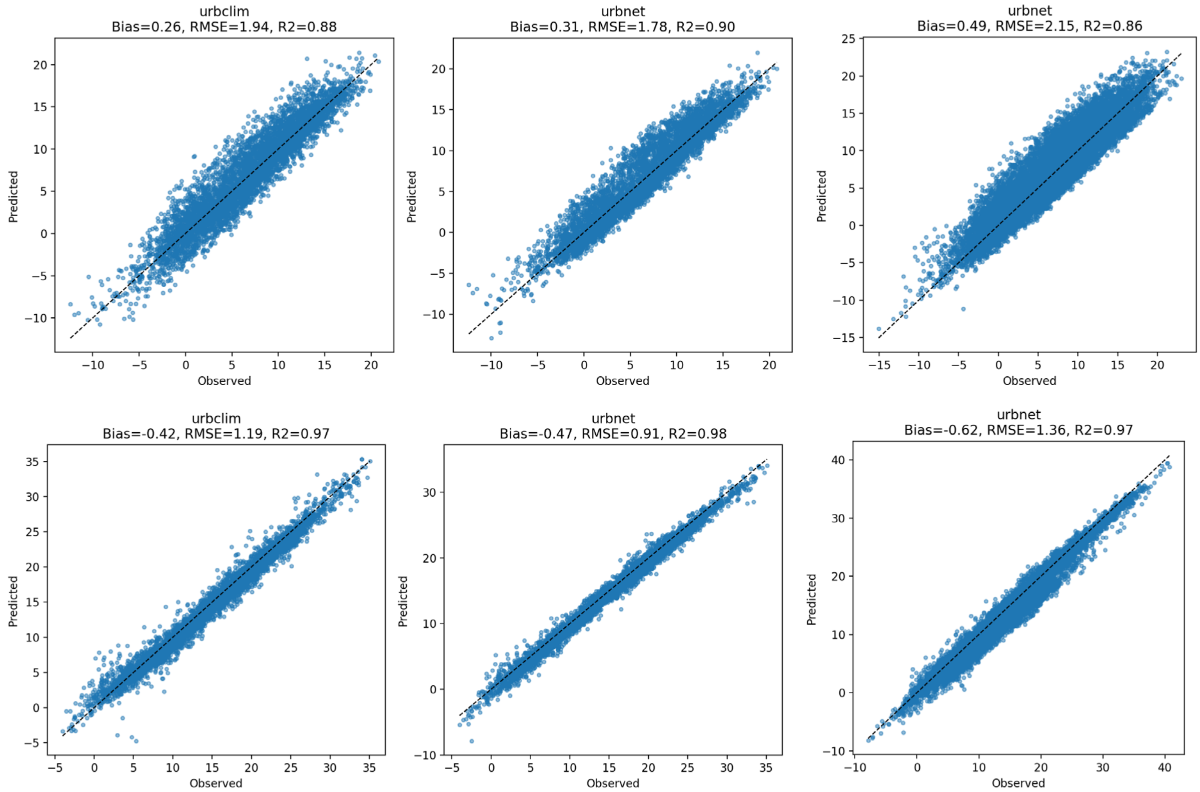

Figure 11: Validation of daily minimum (top panels, °C) and maximum temperature (bottom panels, °C) against AWS observations (2010–2020). UrbClim reference results are shown in the left panel; UrbNet results are shown in the middle and right panels. The left and middle panels relate to the four stations within the validation patches, while the right panels provide independent validation using seven stations outside all training and validation areas (see Figure 10 for station locations).

Validation outside the validation patches (fully independent validation)

Pathway 3: Comparison of UrbNet with AWS outside the validation patches

To provide an independent evaluation beyond all regions used for training or UrbClim-based optimisation, UrbNet was compared with AWS observations located **outside all training and validation patches** (right panels of Figure 11 and Figure 12). The results show that:

- UrbNet achieves performance levels comparable to those observed within the validation patches.
- The model maintains accuracy even in areas with no exposure during either training or internal validation.

This confirms that the AI model is **spatially transferable** and capable of providing reliable high-resolution climatic estimates across the entire Belgian domain.

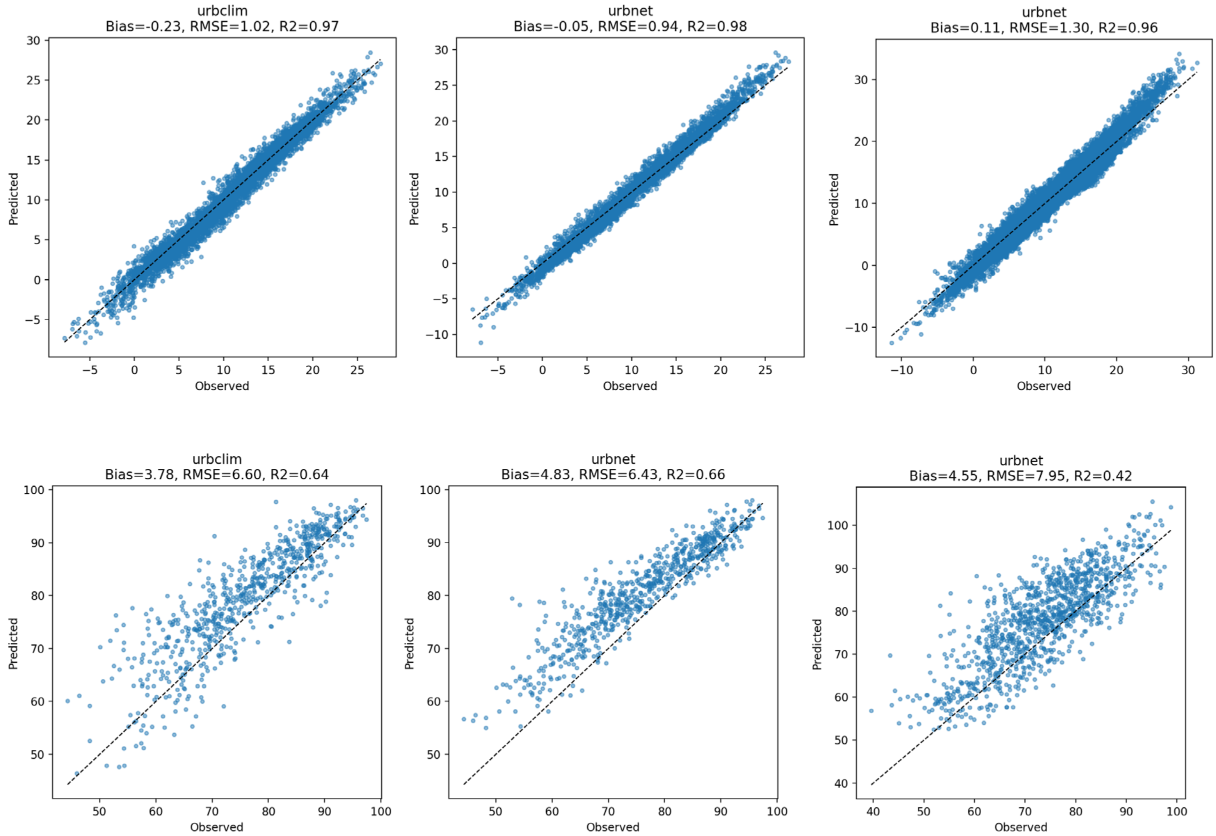

Figure 12: Same as Figure 11, but for daily mean temperature (top panels, °C) and daily mean relative humidity (bottom panels, %).

2.2.5.3 UrbNet output

UrbNet, the trained and validated AI model, forms the **operational engine of Flow 1**, automatically generating high-resolution (100 m) daily maps for day t – 6, based on ERA5 inputs and the pre-configured land predictor dataset.

In Figure 13, a comparison between ERA5 output (used as UrbNet input) and UrbNet output is presented. In Figure 14, averaged UrbNet output over Belgium for 2010–2020 is presented for different climate variables.

The UrbNet model also provides **the computational foundation for the regression-based flow (Flow 2)**, which reuses the same trained model for on-the-fly generation of analogue-day maps.

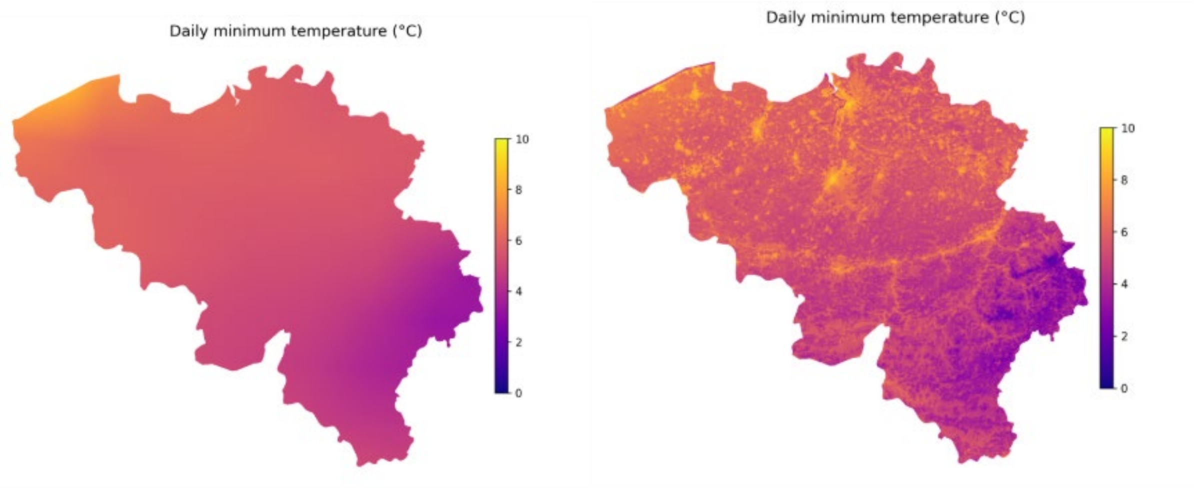

Figure 13: Comparison between ERA5 output (left) and UrbNet output (right) for daily minimum temperature averaged for the year 2000. For this visual comparison, ERA5 (originally at 25 km resolution) was interpolated to the same grid as UrbNet at 100 meter resolution.

2.2.6 Software implementation

The AI-based workflow was implemented as part of the modular CLIMSERVE Python package, developed by VITO.

Within this package, Flow 1 is encapsulated as a function named `Urbnet_flow_latest_era5` and can be configured using a configuration class. This function automatically retrieves ERA5 predictor data for day t – 6, combines them with the pre-configured land predictors, and executes the trained neural-network model to produce high-resolution maps.

The module is fully reusable and interoperable with Sciensano's POSIT Enterprise environment, ensuring transparent, automated, and reproducible processing.

Further technical details on the CLIMSERVE architecture and integration are provided in Section 2.4.

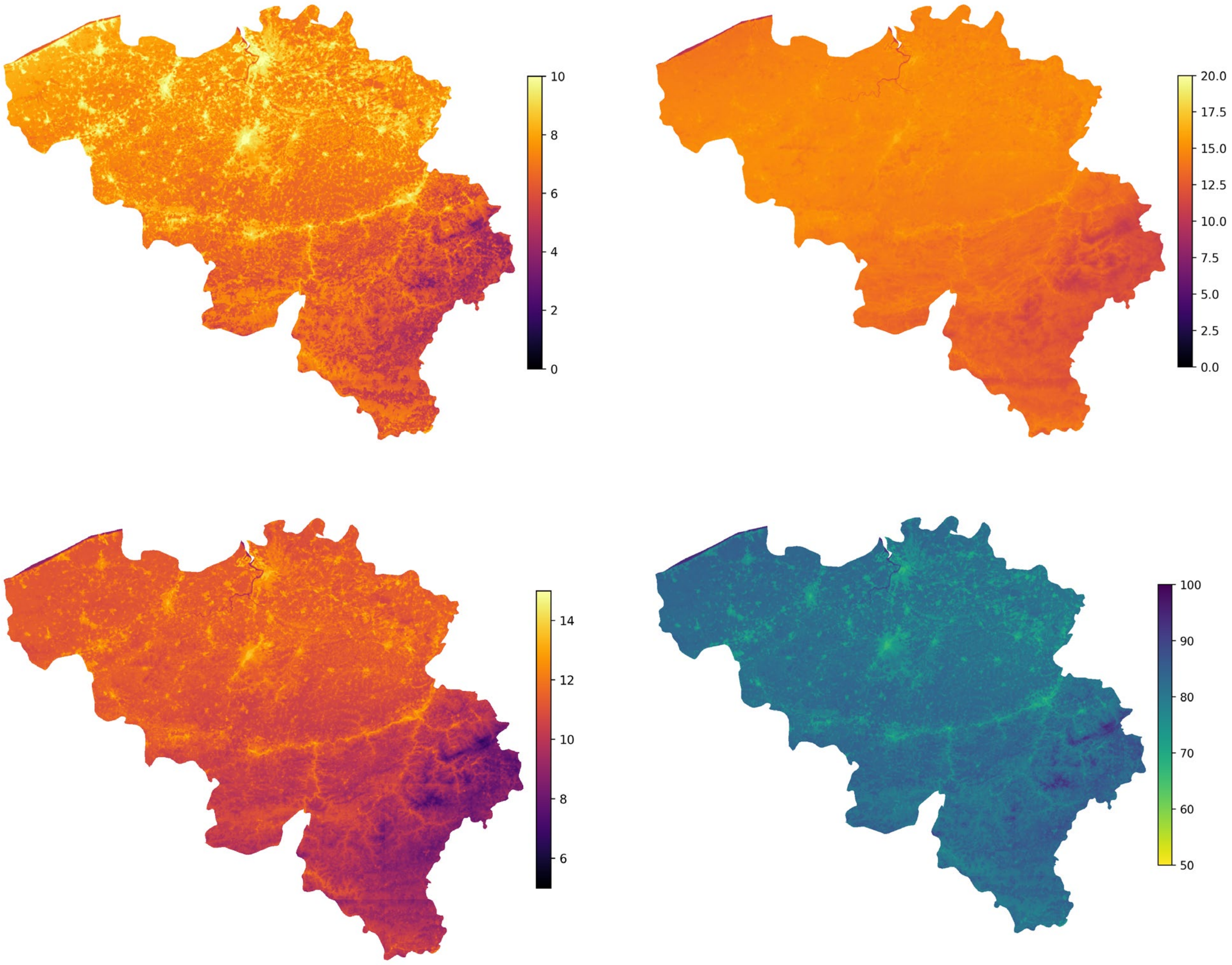

Figure 14: Average UrbNet output over Belgium for 2010–2020, showing daily minimum temperature (upper left), daily maximum temperature (upper right), daily mean temperature (lower left) and daily mean relative humidity (lower right). Temperature is expressed in °C and relative humidity in %.

2.2.7 Operational output

In operational mode, Flow 1 produces daily 100 m-resolution maps of the requested climatological variables for Belgium, for day t – 6. The outputs are stored in NetCDF format, enabling seamless ingestion by Sciensano's analysis tools.

Each product includes metadata on variable definitions, model version, and processing date, ensuring full traceability.

Examples of output maps and integration within Sciensano's POSIT environment are discussed in Section 2.4.

2.2.8 Summary of the AI-based workflow implementation

A concise summary of the implemented AI-based workflow is provided in Text Box 4.

Text Box 4. Summary of AI-based workflow implementation (Flow 1)

The AI-based workflow (Flow 1) forms the operational core of the Climate Service. It downscales ERA5 reanalysis data to 100 m resolution, producing daily high-resolution climate maps for Belgium.

Scientific foundation
• Built upon UrbClim, a physically based urban boundary-layer model that explicitly simulates the lowest 1.2 km of the atmosphere and resolves land–atmosphere interactions at high spatial detail.
• UrbClim simulations (2000–2020) for 13 representative Belgian patches provided the training dataset for the AI model.

Model and inputs
• The AI model is a 2D neural network trained on paired UrbClim outputs (predictands) and predictor data derived from ERA5 and high-resolution land-surface descriptors.
• Static land predictors include DEM, land cover, NDVI, building fraction, and anthropogenic heat flux; dynamic climate predictors include temperature, humidity, radiation, wind, and precipitation.

Implementation
• Training and validation were conducted on UrbClim-simulated fields, confirming high spatial and temporal correspondence.
• In operational mode, the model runs automatically for day t – 6, using pre-configured land predictors and daily ERA5 data.
• Outputs: validated 100 m-resolution maps of core climate variables, delivered in NetCDF format for integration into Sciensano's POSIT environment.

Source: Pilot implementation, VITO (2025); based on D461BE-1.1.2.1 Service Specification & Implementation Plan.

2.3 Workflow based on regression analysis of AWS data

The regression-based workflow (Flow 2) complements the AI-based workflow by extending the temporal coverage of the Climate Service to the most recent five days (*t – 5 to t*), for which ERA5 data are not yet available.

While the original design proposed retrieving pre-generated maps from an archived database, the pilot implementation introduced an **on-the-fly generation mechanism**, dynamically invoking the AI-based workflow (Flow 1) to produce the required high-resolution maps for analogue days identified through AWS-based regression analysis (see Figure 2).

2.3.1 Scientific principle

The workflow is grounded in a **pattern-matching regression approach** that identifies the historical day most like current meteorological conditions observed across Belgium. The core assumption is that days with comparable large-scale meteorological signatures will exhibit similar fine-scale spatial patterns in temperature, humidity, and other climate variables. By combining real-time observations from the **Royal Meteorological Institute's (RMI) Automatic Weather Station (AWS) network** with a long-term historical archive of AWS measurements, the method reconstructs high-resolution near-real-time maps that are physically consistent with past climate states.

2.3.2 Input data

The regression-based workflow relies on three primary input sources:

1. **Real-time AWS observations** – Daily measurements of temperature, relative humidity, wind speed and direction, and precipitation from RMI's AWS network (13 stations, see Figure 10).
2. **Historical AWS records** – A multi-year database (2000–present) used as the reference for similarity matching.

   Note: For both real-time and historical data, we used stations from the same AWS network to ensure consistency between the measurements.

3. **Trained AI model and land predictors** – The same trained AI model and pre-configured static land predictors used in Flow 1, ensuring methodological consistency.

These datasets are automatically retrieved and processed through dedicated CLIMSERVE modules for AWS data handling and pre-processing.

2.3.3 Similarity analysis and analogue-day selection

For each climate variable independently (e.g., temperature, humidity, wind speed, wind direction, pressure), the model identifies the **historical date t*** by solving the following optimization problem, i.e. by minimizing the following function over t, where n represents the number of measurement stations:

$$E(t, \mathbf{w}) = \sum_{i=1}^{n} w_i [o_i(t) - o_i(\text{today})]^2$$

where:

- $o_i(t)$ represents the **historical observation** for station i on day **t**,
- o_i(today) is the **current observation** for station i,
- w_i is the **weight** assigned to station i (in this case, all stations receive equal weights),
- the sum runs over **all AWS stations** providing observations for the given climate variable.

Since this optimization is performed **separately for each climate variable**, the best-matching historical date **t*** may differ between daily minimum, maximum and mean temperature. To minimize inconsistencies, such as winter days being mapped to summer days, the matching for relative humidity is based on the observations of daily average temperature. The latter is regarded as a proxy for overall weather conditions, including relative humidity.

2.3.4 On-the-fly generation of high-resolution maps

Once the analogue day t* is identified, the system **dynamically invokes the AI-based model (Flow 1)** to generate a corresponding 100 m-resolution climate map for that date. This replaces the pre-generated-map retrieval proposed in the design phase, offering a unified and maintainable solution.

The **on-the-fly generation process** comprises:

1. **Triggering Flow 1** with ERA5 data for the analogue date t* and the static land predictors.
2. Generating **high-resolution maps** for all required variables.
3. **Storing the results** in a standardized NetCDF format for downstream integration.

By reusing the trained AI model, this approach ensures that both workflows rely on the same model version, preprocessing routines, and metadata structure.

2.3.5 Software implementation

The logic of Flow 2 was implemented within the **UrbNet-based flow** and integrated into the modular **CLIMSERVE Python package.**

Two dedicated functions encapsulate the regression and analogue-generation operations:

Function	Description
`Urbnet_flow_latest_era5(config)`	Executes Flow 1 for *t – 6* using the most recent ERA5 data.
`Urbnet_flow_cherrypick_similarity(config)`	Identifies analogue days for *t – 1 to t – 5* via AWS similarity matching and triggers Flow 1 to generate corresponding maps.

The configuration parameters include the climate variable, bounding box, spatial resolution (100 m), coordinate reference system (EPSG: 3035), and temporal offsets.

The functions can be executed manually from within Sciensano's POSIT environment or scheduled for automatic operation once full integration is completed.

2.3.6 Operational output

The regression-based workflow delivers near-real-time, 100 m-resolution climate maps for the five most recent days, seamlessly extending the ERA5-based archive. Outputs are identical in format and structure to those from the AI workflow, ensuring interoperability and enabling downstream analysis in Sciensano's epidemiological tools.

2.3.7 Summary of the regression-based workflow implementation

A concise summary of the implemented regression-based workflow is provided in Text Box 5.

Text Box 5. Summary of regression-based workflow implementation (Flow 2)

The regression-based workflow (Flow 2) complements the AI-based flow by providing near-real-time climate information for the most recent days (t – 5 to t), when ERA5 data are not yet available.

Concept and refinement
• Originally designed to retrieve analogue-day maps from a pre-generated database of AI-derived climate fields.
• During implementation, replaced by an on-the-fly generation mechanism: the system now re-runs the trained AI model (Flow 1) for the analogue date, ensuring methodological consistency and reducing storage demands.

Operational logic
• Daily AWS data from the Royal Meteorological Institute (temperature, humidity, wind, precipitation) are compared with historical AWS records.
• A similarity function identifies the historical day that best matches current conditions for each variable.
• For each selected analogue day, Flow 1 is triggered to produce a corresponding high-resolution climate map.
• The resulting set of maps (t – 1 to t – 5) provides seamless, near-real-time climate coverage.

Implementation
• Encapsulated in two dedicated functions within the stand-alone UrbNet flow:
– Urbnet_flow_latest_era5(config) – runs Flow 1 for t – 6 using the latest ERA5 data.
– Urbnet_flow_cherrypick_similarity(config) – generates maps for t – 1 to t – 5 based on AWS-selected analogue days.
• Integrated into the modular CLIMSERVE Python package and tested within Sciensano's POSIT environment.

Source: Pilot implementation, VITO (2025); based on D461BE-1.1.2.1 Service Specification & Implementation Plan.

2.4 Overview of the services' implementation (technical design)

This section outlines the technical design behind the development and operational deployment of the climate service. It describes both the creation of the **CLIMSERVE Python package** and its **integration into Sciensano's existing analytical workflows**.

Essentially, the service is structured around two main pillars:

- The development of a modular Python package to generate and process climate data.
- The integration of this package within Sciensano's POSIT Enterprise environment, enabling automated workflows and operational use.

The subsections below elaborate on these components in detail.

2.4.1 Climserve Python Package Overview and Workflow

A central component of the envisioned Climate Service (CS) is the development of a modular Python package, specifically designed to process input data and execute model inference tasks in support of high-resolution climate mapping. This package encapsulates the full technical workflow and forms the backbone of the CS data processing logic.

The package is organized around a set of independent, reusable modules, each responsible for a distinct operation within the broader processing chain. This modular approach ensures flexibility, scalability, and ease of maintenance while supporting the integration of future extensions.

In its current standalone version, the package provides two key functions that illustrate the operational workflow:

The first function, **`urbnet_flow_latest_era5(config)`**, runs the UrbNet flow for today minus six days using the most recent ERA5 reanalysis data. It automatically retrieves the latest available ERA5 dataset corresponding to 'today – 6' and executes the UrbNet model workflow. This functionality is particularly useful for maintaining near-real-time updates based on freshly released ERA5 data.

The second function, **`urbnet_flow_cherrypick_similarity(config)`**, performs UrbNet flow simulations for today minus one through five days by selecting the most similar historical analogue day. The selection is based on observed station temperature data, ensuring that each recent day is represented by a closely matching historical counterpart. This "cherry-picking" approach enables realistic reconstructions and scenario testing based on observed local conditions.

The following parameters outline the structure and capabilities.

Core Configuration

A required setting is the UrbnetVariable, which must be one of the following:

- Daily minimum temperature
- Daily maximum temperature
- Daily mean temperature
- Daily mean relative humidity

The selected UrbnetVariable determines which ERA5 climate variables are retrieved, indeed:

- For **daily minimum temperature:**
    - Daily minimum temperature
    - Wind speed value at the hour of daily minimum temperature
    - Northing wind direction at the hour of daily minimum temperature
    - Easting wind direction at the hour of daily minimum temperature
    - Incoming mean solar radiation from the previous day
- For **daily maximum temperature:**
    - Daily maximum temperature
    - Wind speed value at the hour of daily maximum temperature
    - Northing wind direction at the hour of daily maximum temperature
    - Easting wind direction at the hour of daily maximum temperature
    - Incoming mean solar radiation from the current day
- For **daily mean temperature:**
    - Daily mean temperature
    - Daily mean wind speed
    - Daily mean northing wind direction
    - Daily mean easting wind direction
    - Incoming mean solar radiation from the current day
- For **daily mean relative humidity:**
    - Daily mean temperature
    - Daily mean relative humidity
    - Daily mean wind speed
    - Daily mean northing wind direction
    - Daily mean easting wind direction
    - Incoming mean solar radiation from the current day

Additional Configuration Options

- **Experiment name**: default VECTORBORNE
- **SRID (spatial reference ID)**: default 3035
- **Bounding box (min/max)**: [[2913063, 3202063], [3755765, 4099165]]
- **Grid offset**: [63, 65]
- **Grid resolution**: [100, 100]
- **UrbNet build archive start**: date or null (default null)
- **Stations buffer time**: PT4H (4 hours)
- **Days backwards UrbNet**: 6

Collectively, these components form the technical foundation for automated, data-driven UrbNet workflows within the Climate Service framework, paving the way for scalable high-resolution climate simulations and continuous operational updates.

The modularity of the package ensures:

- **Reusability** across research and operational environments
- **Transparency and reproducibility** of all processing steps
- **Ease of integration** into POSIT Enterprise for automated deployment
- **Scalability**, allowing future adaptation to other geographic regions or variables

By abstracting the technical complexities of climate data handling and model inference, this package enables streamlined, daily production of actionable climate information, supporting public health applications such as vector-borne disease risk assessment.

2.4.2 Integration into Sciensano's workflows using POSIT Enterprise

The integration of the Climserve Python package into Sciensano's operational workflows was shaped through collaboration between Sciensano and VITO. Initially, the idea was that VITO would deliver a fully operational service, including the orchestration of daily tasks. However, during in-depth technical discussions, several key considerations led to a revised approach that better aligned with both organizations' capacities and constraints:

- **VITO's orchestration framework is proprietary:** The orchestration layer VITO uses for internal automation is based on in-house developed software that falls under VITO's intellectual property. As such, it cannot be transferred to external partners or made openly available.
- **General-purpose orchestration, not fit for specific external use**: The orchestration tooling in use at VITO is designed to support a wide range of internal data pipelines in a generalized manner. Its complexity and abstraction level exceed what is needed for the specific Climate Service workflows, making it unnecessarily difficult to operate, maintain, or adapt outside VITO.
- **Incompatibility with Sciensano's existing infrastructure**: The VITO software stack is not readily compatible with the tools and systems currently used at Sciensano, such as POSIT Enterprise. Adopting the VITO tooling would require additional maintenance effort and training, while introducing integration and security challenges.

Based on these considerations, **the following division of responsibilities was established:**

- **Sciensano** would take responsibility for implementing the orchestration layer of the Climate Service.
- **VITO** would focus on delivering the modular CLIMSERVE Python package and supporting Sciensano during the integration process.

This **collaborative model** ensures that operational ownership lies with Sciensano, while maintaining alignment with the technical design delivered by VITO.

As a result, the Climserve package will be integrated into Sciensano's POSIT Enterprise environment, a robust and widely used platform for managing analytical workflows in R and Python.

This implementation approach combines the technical expertise of VITO in climate modeling with Sciensano's existing analytical infrastructure and operational needs. It results in a streamlined, maintainable, and organization-owned Climate Service, ready for long-term integration into public health surveillance and response systems.

ECMWF – **Robert-Schuman-Platz 3, 53175 Bonn, Germany**

Contact: https://support.ecmwf.int/