HELLENIC REPUBLIC
MINISTRY OF DEVELOPMENT
GENERAL SECRETARIAT FOR RESEARCH AND INNOVATION
HELLENIC FOUNDATION FOR RESEARCH AND INNOVATION

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN
"BASIC RESEARCH FINANCING" (Horizontal support for all Sciences)
ID 16618 – Subproject 1 (MIS: 5163923)

WP2
D2.2 Digital Twin and Documentation

Spatially Explicit Digital Twin of the Greek Agro-Hydro-System

ID 14815

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN
Funded by the European Union NextGenerationEU

Plan Details

Digital Twin and Documentation
Deliverable number: D2.2
Creation Date:01/02/2025
Last modification date: 10/11/2025
Dissemination Level: Public

1. Introduction

The objective of Work Package 2 (WP2) is the development of DT-Agro, a spatially explicit Digital Twin of the Greek Agro-Hydro-System. DT-Agro constitutes the core computational component of the project, integrating modelling algorithms, Earth Observation (EO) data and processing workflows into a unified, distributed system capable of simulating and assessing agricultural and hydrological processes across Greece. Its overarching aim is to provide a flexible and efficient digital environment that can:

- describe the current state of the agro-hydro-system,
- explore climate and management scenarios, and
- support data-driven decision-support tools for sustainable agriculture and water management.

The first stage of this work, documented in D2.1, focused on the development and evaluation of the core algorithms and workflows that underpin the Digital Twin. This included the adaptation and implementation of simplified but robust algorithms for the main components of the soil–plant–atmosphere system (hydrological balance, crop water use, soil erosion), as well as the recoding and optimisation of the AgroHydroLogos model core in C++ and Python to ensure computational efficiency and interoperability. In parallel, Python-based workflows were developed to automate the acquisition, preprocessing and harmonisation of key EO and geospatial datasets, such as AgERA5 meteorological data, Copernicus Land Monitoring Service (CLMS) products, and soil and land-cover data from ISRIC, ESDAC, CORINE Land Cover and CLC Backbone.

Building on these developments, D2.2 presents the integration of these components into the operational DT-Agro system. This deliverable describes the complete system architecture, the data flow between modules, the model configuration procedures and the operational framework that enables seamless interaction between EO data, the AgroHydroLogos-based modelling core and the processing tools developed under WP2. It documents how simulations are initialised, how the Digital Twin is executed over historical and recent periods, how outputs are stored and visualised, and which mechanisms ensure scalability, reproducibility and interoperability with other work packages.

In addition, the report provides the technical documentation of the DT-Agro system, including the organisation of the code base, the description of the main modules and interfaces, the database and file structures, and the software and hardware requirements for deployment. In this context, WP2 provides the computational and data infrastructure that underpins the analysis tasks of WP5. While D2.2 documents the implementation and operational readiness of the DT-Agro Digital Twin, D5.1 focuses on the methodological framework for exploiting the data it generates to extract meaningful indicators and knowledge. Subsequent deliverables (D5.2 and D5.3) will build upon the outputs of the Digital Twin to assess agro-hydrological processes, evaluate climate impacts and design spatially explicit decision-support services.

2. System Architecture and Design

The DT-Agro Digital Twin is implemented as a modular, layered system that integrates the C++ simulation core, Python-based data workflows, EO and ancillary datasets, and user-facing tools into a single operational framework. The architecture is designed to (i) support large-scale daily simulations over the entire territory of Greece, (ii) ingest and update multiple dynamic data streams (meteorology, soil moisture, vegetation, land cover, etc.), and (iii) remain open, extensible, and interoperable with other data streams and external platforms.

At the highest level, DT-Agro is organised into four main layers:

1. **Data acquisition and pre-processing layer**, comprising the EO and ancillary data workflows developed in WP2 for meteorological forcing, vegetation indices, land cover/imperviousness, soils, topography and (in later phases) soil moisture, actual evapotranspiration and other state variables.
2. **Data management layer**, which stores static and dynamic data in harmonised geospatial formats and structured directories, and exposes these datasets to the modelling core through standardised interfaces.
3. **Modelling and simulation layer**, built around the AgroHydroLogos-derived hydrological, crop growth and soil-erosion modules, fully re-implemented in C++ and optimised for dual-resolution and parallel execution.
4. **Orchestration and interaction layer**, implemented in Python, which manages configuration, job scheduling, pre- and post-processing, and provides interfaces for users and for other WPs.

The following subsections describe the main components and data flows of this architecture.

2.1 Overall concept of the Digital Twin

Conceptually, DT-Agro goes beyond a conventional agro-hydrological model. It is designed as a **digital twin of the Greek Agro-Hydro-System**, i.e. a dynamic numerical replica that:

- maintains a consistent representation of the current state of soils, crops and water resources,

- can be updated as new EO and other data become available, and
- can be used to explore past conditions, current situations and future scenarios within a single framework.

To support this concept, the system architecture explicitly separates:

- **Static components** (e.g. DEM and derivatives, soils, long-term land cover, irrigation infrastructure), which change rarely and are stored as project-wide reference layers;
- **Slowly varying components** (e.g. crop patterns from IACS, imperviousness, multi-annual land-cover products), which are updated every few years; and
- **Fast-varying components** (meteorology, NDVI/vegetation indices, surface soil moisture and other EO-derived variables), which are updated on daily to seasonal time scales.

This separation allows DT-Agro to run long historical simulations, to perform "state updates" when new information becomes available (e.g. updated NDVI fields or soil-moisture maps), and to support near-real-time or seasonal applications with minimal reprocessing.

2.2 Main software components

From an implementation viewpoint, the DT-Agro architecture comprises three principal software components:

1. **C++ simulation engine**
    - Implements the computational core of AgroHydroLogos, including:
        - the **water balance of the reference soil volume** (infiltration, soil moisture, deep percolation, baseflow),
        - **runoff generation** using the impervious-aware SCS-CN formulation (pervious and impervious separation at cell level),
        - **actual evapotranspiration and crop-water stress**,
        - **irrigation water requirements**, and
        - the **RUSLE-based soil-erosion and sediment-delivery module**.
    - Includes a dedicated **runoff-routing component**, which:
        - uses pre-computed flow directions, slopes, channel masks and flow-accumulation grids derived from the DEM,
        - converts cell-scale surface runoff and baseflow into **discharge time series along the drainage network**,
        - routes delivered sediment along the same network to obtain **sediment-load time series** at control points,

        - relies on travel-time / velocity relationships (for overland and channel flow) to move water and sediment downstream
        - provides daily streamflow and sediment loads at user-defined outlets and internal control points for validation and linkage with other WPs.
    - Operates on a **dual-resolution grid** (1 km for meteorological forcing, 100 m for agro-hydrological and erosion processes), using pre-computed meteorological interpolation weights and efficient in-memory data structures.
    - Integrates the **impervious-aware SCS-CN formulation** developed in DT-Agro by computing, for each grid cell, separate contributions from impervious and pervious fractions derived from Copernicus imperviousness products and land-cover datasets (CLC Backbone, IACS, etc.). In this way, the hydrological response at cell level explicitly accounts for small impervious patches embedded in otherwise permeable landscapes.
    - Includes the **serialisation algorithm** (Soulis, 2013) and **domain-decomposition techniques** that allow hydrologically independent regions to be processed separately, facilitating both parallel execution and targeted calibration or real-time update runs over sub-domains.

2. **Python orchestration, EO and utilities package**
    - Provides high-level Python modules to configure and run the C++ engine, prepare input files, and manage outputs.
    - Hosts the **EO data pipeline** described in D2.1 and D2.2, including meteorological data retrieval, AgERA5 sampling and station-wise bias correction, NDVI and vegetation-index processing, land-cover and imperviousness preparation, soil and topographic pre-processing, and the creation of **virtual meteorological stations** and dynamic interpolation weights.
    - Implements job-management routines to split the national domain into tiles or hydrological regions, submit and monitor simulation tasks, and recombine outputs.
    - Contains utilities for basic diagnostics and validation (e.g. comparison with station data, hydrographs at outlets, consistency checks, summary statistics).
3. **User interface and external interfaces**
    - Replaces the legacy ArcGIS/ArcMap extension of AgroHydroLogos with an **open-source, Python-based interface**, which can be used both interactively (e.g. via notebooks or simple GUIs) and in batch mode.
    - Maintains the connection to GIS functionality through libraries such as rasterio, rioxarray, geopandas and shapely, ensuring that raster and vector

data can be read, processed and visualised without dependence on proprietary desktop software.

- Uses standard geospatial formats (GeoTIFF, NetCDF/xarray, CSV/Parquet, ESRI Shapefile/GeoPackage) throughout, allowing straightforward integration with other tools in WP3 and WP5 or with external spatial platforms.

2.3 Data flows and execution modes

The DT-Agro architecture is designed around clear, reproducible data flows from external providers to model outputs. A typical workflow consists of the following stages:

- **Data ingestion and pre-processing**
  - EO and ancillary datasets (AgERA5, station observations, CLMS NDVI, Copernicus land-cover and imperviousness products, soils, DEM, surface soil-moisture products, etc.) are acquired and processed using the modular Python workflows documented in D2.1 and D2.2.
  - For meteorology, AgERA5 is sampled at the locations of approximately 140 historical stations and regression models are calibrated for each station and variable to generate **bias-corrected virtual-station series**, which provide continuous daily time series from the start of the AgERA5 record to the present.
  - All rasters are reprojected to EGSA87 (EPSG:2100) and resampled to the target grid (100 m or 1 km), while time series are stored in standard tabular or NetCDF formats with consistent naming conventions.
- **Model configuration**
  - A configuration layer specifies the simulation period, spatial domain (full country or sub-regions), the selection of static and dynamic input datasets, the parameter sets to be used, and the desired outputs (variables, aggregation levels, temporal frequency).
  - The configuration is expressed in human-readable files (e.g. YAML/JSON) and passed to the Python orchestration layer, which automatically locates the corresponding input files in the data repository and prepares the input folders expected by the C++ core.
- **Execution and parallelisation**
  - The orchestration layer splits the domain into hydrologically independent regions or tiles and prepares input bundles for each.
  - For each region, the C++ engine is invoked (as an executable or shared library) and performs daily simulations, reading meteorological fields from the 1 km

grid (interpolated from the bias-corrected virtual stations) and computing water-balance, crop and erosion variables at 100 m resolution.

  - As part of the same run, the **runoff-routing component** converts cell-scale surface runoff, baseflow and delivered sediment into discharge and sediment-load time series along the drainage network, at user-defined outlets and internal control points.
  - Parallel execution is supported at two levels: (i) spatial domain decomposition (multiple regions processed concurrently), and (ii) internal multi-threading within the C++ engine, where appropriate. This ensures that national-scale runs remain computationally feasible even when additional processes (e.g. erosion, future nutrient modules) are activated.

- **Post-processing and export of results**
  - Raw outputs (time series at cell and aggregated levels, maps of long-term means or extremes, indicators relevant to WP5) are written in standard formats.
  - The Python layer then performs optional post-processing steps such as aggregation to administrative or hydrological units, linkage to the IACS parcel layer, and calculation of derivative indicators (e.g. irrigation abstractions by water source, soil-loss risk classes, discharge statistics at control sections).
  - These processed outputs are made available to WP5 through well-defined directories and naming conventions and can also be served to external tools or visualisation platforms.

Within this architecture, different **execution modes** can be supported with minimal changes in configuration:

- **Historical mode**, where the model is run for multi-decadal periods using static or slowly varying land-use conditions and bias-corrected meteorology;
- **Current-state mode**, where the most recent EO-derived NDVI, soil-moisture and land-cover layers are used to update the state variables before simulating the current season; and
- **Scenario mode**, where alternative land-use, irrigation or climate scenarios are substituted at configuration level to explore future conditions and management options.

2.4 Openness, interoperability and extensibility

A key requirement of WP2 is that DT-Agro remains open, transparent and easy to extend. The system architecture supports this through several design choices:

- All core algorithms have been recoded in C++ with a clear separation between numerical routines and input/output handling, so that new processes (e.g. nutrient cycling, salinity, soil-health indicators, pollutant transport) can be added as additional modules without rewriting existing components.
- Communication between layers is based on open, documented formats, rather than proprietary project files. This makes it straightforward to plug DT-Agro into external workflows, cloud services or web-based decision-support tools.
- The Python orchestration and EO modules are organised as reusable packages, allowing other WPs to call DT-Agro simulations programmatically (e.g. from WP5 analysis scripts) and to re-use the EO processing routines in other contexts.
- The architecture is platform-independent: the same code base can run on a desktop workstation, an institutional HPC cluster or a cloud environment, depending on the needs of the application and the availability of computing resources.

Overall, the system architecture and design of DT-Agro provide a robust backbone for the Digital Twin, ensuring that the hydrological, crop and erosion algorithms developed under D2.1 can operate in a scalable, EO-driven, and policy-relevant environment. Subsequent sections detail the database structures, the integration of EO and (future) IoT data, and the validation activities that demonstrate the operational readiness of this architecture.

3. Database

The DT-Agro Digital Twin is supported by a dedicated data repository that stores all input, intermediate and output datasets required for model configuration, execution and analysis. In the context of this deliverable, the term “database” refers to this hybrid geospatial data store, which is primarily file-based (rasters, NetCDF, CSV/Parquet and vector files) but organised and documented as a coherent, project-wide geodatabase.

The database plays a central role in WP2 because it:

- provides a single, harmonised source for all EO, meteorological and ancillary data used by DT-Agro,
- ensures traceability between raw products, pre-processed layers and the datasets actually used in simulations, and
- exposes well-defined entry points for other work packages (especially WP5), which consume DT-Agro outputs and indicators.

3.1 Objectives and design principles

The design of the DT-Agro database follows a set of pragmatic principles:

- **Consistency and harmonisation.** All spatial datasets are stored in a common projection (EGSA87 / EPSG:2100) and at a limited set of resolutions (100 m for agro-

hydrological and erosion processes, 1 km for meteorological forcing). Variables derived from different sources (e.g. soil maps, Copernicus land cover, IACS, NDVI) are harmonised to the same grid and coordinate system.

- **Separation of concerns.** Raw EO products, intermediate pre-processed datasets and final model-ready inputs are stored in separate, clearly labelled locations. This allows reprocessing or updating of individual stages without confusion with others.
- **Reproducibility and versioning.** Filenames and folder structures encode dataset type, spatial resolution, time period and version (e.g. product vintage or processing script version). This makes it possible to reproduce simulations later, even after updating to new data versions.
- **Scalability and performance.** The repository is designed so that large national-scale rasters and long time series can be streamed efficiently by the C++ core (e.g. using serialisation algorithm binary files).
- **Openness and interoperability.** Only open, widely supported formats are used (GeoTIFF, NetCDF/xarray, CSV/Parquet, Shapefile/GeoPackage). This makes the database usable not only by DT-Agro code, but also by partner tools and external GIS/analysis platforms.

3.2 Logical data structure

Logically, the DT-Agro database is organised into a small number of **data domains**, each of which groups related datasets:

1. **Static geospatial reference data**
    - Digital Elevation Model (DEM) and derivatives: slope, aspect, flow direction, flow accumulation, channel network masks.
    - Soil properties: texture, hydraulic parameters, erodibility factors (K), soil depth, etc.
    - Long-term land cover and land use from Copernicus (CLC, CLC Backbone, HRLs), supplemented by IACS parcel information in agricultural areas.
    - Imperviousness fraction layers from Copernicus HRL, resampled to 100 m.
2. **Meteorological data**
    - Raw station observations (as collected, with original metadata).
    - Raw AgERA5 or similar reanalysis products (NetCDF grids).
    - **Bias-corrected virtual-station time series** for each of the approximately 140 station locations (one file per station/variable), generated by regression between observed data and sampled AgERA5.
3. **Dynamic EO-derived variables**

- NDVI and other vegetation indices at 100 m, stored as time stacks.
- Derived crop coefficients (Kc) and cover factors (C) at 100 m resolution, computed from NDVI and land-cover information.
- Surface soil-moisture products at 1 km or 100 m, when available.

4. Additional EO layers (e.g. LAI/FAPAR, snow cover), which the system architecture is designed to accommodate.**Model parameter maps**
   - Raster layers representing model parameters on the 100 m grid, such as:
     - SCS-CN base values for pervious areas,
     - impervious fraction per cell and derived effective CN,
     - rooting depths and water-holding capacity,
     - RUSLE factors (R, K, LS, C, P) for erosion.
   - These maps are derived from the static reference data and dynamic EO layers and represent the **bridge between EO/geospatial data and the C++ model core.**
5. **Simulation configurations and runs**
   - Configuration files (YAML/JSON) describing simulation settings, model options and input datasets for each run.
   - Run metadata (log files, start/end times, code version, domain, processes activated).
6. **Model outputs and indicators**
   - Raw outputs: time series at grid-cell level or at control points (e.g. daily runoff, soil moisture, ET, erosion).
   - Aggregated outputs: maps of long-term means, extremes or indicators (e.g. mean annual irrigation requirement, long-term soil loss, water balance components by sub-basin).
   - Derived indicators for WP5: spatially aggregated or post-processed variables at administrative or hydrological unit scale.

These domains are logically distinct but strongly linked: for example, the RUSLE C factor in the parameter maps domain depends on both static land cover and dynamic NDVI, while the meteorological interpolation weights depend on station locations and on the DEM-derived topographic gradients.

3.3 Physical organisation and directory structure

Physically, the database is implemented as a **hierarchical directory structure** on disk (and/or network storage), with top-level folders corresponding to the domains above. A typical structure is:

- static/ – DEM, soils, long-term land cover, imperviousness, irrigation infrastructure.
- meteo/ – raw station data, raw AgERA5, virtual stations.
- eo/ – NDVI and other vegetation indices, soil moisture and auxiliary EO layers.
- parameters/ – model parameter rasters on the 100 m grid (CN, Kc, RUSLE factors, rooting depth, etc.).
- configs/ – configuration files for DT-Agro runs.
- runs/ – outputs organised by run ID, containing maps and time series for each simulation.
- metadata/ – documentation, lookup tables (e.g. crop codes ↔ parameter sets), station lists, processing logs.

Within each folder, naming conventions encode key attributes:

- dataset type (e.g. ndvi, cn_pervious, meteo_prcp),
- spatial resolution (e.g. _100m, _1km),
- time period (e.g. _2000_2024, _2015),
- version (e.g. _v1, _v2 when reprocessing yields updated products).

For time-dependent rasters, NetCDF/xarray files or multi-band GeoTIFFs are used, with the time dimension clearly defined. For large collections of station or virtual-station time series, CSV or Parquet files are used with standard column names (station ID, date, variable, value).

This layout is aligned with the Python orchestration layer, which expects inputs under known folder names and patterns, and with the code documentation that describes how each script reads and writes data.

Table 1. Main datasets stored in the DT-Agro database and their role in the Digital Twin.

Domain	Dataset / Variable	Main source(s)	Native / stored resolution	Temporal coverage	Main use in DT-Agro
Static geospatial	DEM, slope, aspect, flow accumulation, streams	Copernicus DEM / EU-DEM	Native ~25 m / stored 100 m	Static	Topography, flow routing, LS factor, delineation of

					hydrological units
Static geospatial	Soil properties (texture, hydraulic, K, SOC)	ISRIC, ESDAC, national soil maps	Native 250–1 000 m / stored 100 m	Static / multi-year	Rooting depth & water storage, runoff parameters, RUSLE K factor, soil health
Land cover / use	Land cover, crop types	Copernicus CLC & CLC Backbone, IACS	Native 10–100 m / stored 100 m	Multi-annual (1990–present); annual IACS	Definition of agricultural areas, crop types, management practices, C and P factors
Imperviousness	Imperviousness density	Copernicus HRL Imperviousness	Native 10–20 m / stored 100 m	Multi-annual (e.g. 2006–2021)	Impervious fraction per cell, impervious-aware SCS-CN, urban runoff
Meteorology – stations	Observed station time series	National station networks (≈140 stations)	Point locations	Station-dependent historical periods	Calibration of bias correction; QC; evaluation of reanalysis products
Meteorology – reanalysis	AgERA5 (P, T, etc.)	Copernicus Climate Data Store (AgERA5)	Native 0.1° / sampled at stations	~1979–present	Basis for virtual stations and gap filling where observations are missing
Meteorology – virtual	Bias-corrected	Stations + AgERA5	Point locations	AgERA5 period	Forcing of interpolation scheme; main input for 1 km

	virtual-station series	(regression-based)		(continuous daily)	meteorological grids
Meteorology – gridded	Interpolated daily P, T, etc.	Virtual stations (interpolation)	Stored 1 km	AgERA5 period (continuous daily)	Meteorological forcing of DT-Agro core (dual-resolution framework)
EO – vegetation	NDVI (and derived Kc, C)	Copernicus Land Monitoring Service (CLMS)	Native 10–300 m / stored 100 m	Product-dependent (10-daily / monthly)	Dynamic crop condition, crop coefficients, RUSLE C factor, vegetation diagnostics
EO – soil moisture	Surface soil moisture	Copernicus/global EO products	Native 1–25 km / stored 1 km/100 m	Product-dependent (daily / multi-day)	Evaluation of soil moisture; potential state updates and calibration
Model parameter maps	CN, rooting depth, WHC, RUSLE R,K,LS,C,P, etc.	Derived from DEM, soils, land cover, EO	Stored 100 m (some 1 km)	Static / periodically updated	Bridge between EO/geospatial data and DT-Agro C++ core parameterisation
Simulation configurations	Run configuration files	WP2 (DT-Agro configuration)	Text files (YAML/JSON)	One per simulation	Definition of simulation domain, period, inputs, parameters, outputs
Model outputs	Gridded outputs & indicators	DT-Agro simulations	100 m / 1 km; point time series	Simulation period (daily /	Water balance, ET, irrigation, erosion, discharge,

				aggregate d)	indicators for WP5

3.4 Metadata, provenance and version control

To ensure **traceability and reproducibility**, the database includes basic metadata and provenance information:

- For each dataset, a small metadata file (e.g. JSON or text) summarises:
  - source (e.g. “CLMS NDVI”, “Copernicus DEM”, “ISRIC SoilGrids”, “HNMS station network”),
  - processing steps applied (e.g. reprojection, resampling method, masking, regression correction),
  - date of processing and the corresponding script or code version.
- Look-up tables are maintained for:
  - station IDs and coordinates,
  - mapping between IACS crop codes and DT-Agro crop parameter sets,
  - mapping between land-cover classes and hydrological/erosion parameters.
- When datasets are updated (e.g. new CLC version, updated NDVI series, recalibrated virtual stations), the new versions are stored alongside the old ones, with version identifiers in filenames. The configuration files for each model run point explicitly to a particular version, so that past simulations remain reproducible.

Where appropriate, lightweight version control (e.g. Git) is used for configuration files and small text metadata, while large rasters and time series are versioned through their filenames and directory structure.

3.5 Access, backup and integration with other WPs

Access to the DT-Agro database is provided through:

- direct file-system access for the DT-Agro code,
- documented directory structure and naming conventions for WP5 and other parts who wish to use inputs or outputs,
- and, where needed, simple wrapper scripts (e.g. in Python or R) that facilitate loading standard datasets.

Regular **backups** of the database are performed at the level of:

- raw and pre-processed EO and meteorological datasets, which are costly to regenerate, and
- calibration and validation results for key regions.

The organisation described in this section makes it straightforward to:

- plug new data sources into DT-Agro (e.g. additional stations, new Copernicus generations, IoT sensors),
- provide clear entry points for WP5 analyses (e.g. reading model outputs or indicators), and
- keep a clear separation between data (this section) and code (documented in subsequent sections of D2.2 and in the accompanying code documentation).

4. Integration with EO and IoT data

A key characteristic of DT-Agro, and the main difference from a conventional agro-hydrological model, is its ability to interact with external data streams in a systematic way. Earth Observation (EO) products and, in later phases, data from IoT sensing networks are used not only to parameterise the model, but also to drive, constrain and update the simulations over time.

In this context, WP2 has designed DT-Agro as an open system that can:

- ingest EO and later IoT data in a controlled, documented way,
- translate them into model parameters, forcings and state updates, and
- provide feedback (e.g. indicators, anomalies) back to decision-support tools.

The following subsections describe how EO and IoT data are integrated in the current version of DT-Agro and how the architecture prepares the ground for more advanced data-assimilation and real-time applications.

4.1 Conceptual roles of EO and IoT within the Digital Twin

Within DT-Agro, EO and IoT data fulfil complementary roles:

- EO data (satellite- and reanalysis-based) provide spatially explicit information on a national scale:
    - static or slowly varying layers (DEM, soils, land cover, imperviousness),
    - gridded meteorological forcings (via AgERA5 + virtual stations),
    - dynamic vegetation indices (NDVI and derivatives), and
    - surface soil-moisture and other auxiliary products.
- **IoT data** (ground-based sensors and smart devices) provide high-frequency, local measurements for selected locations, such as:
    - soil-moisture sensors at different depths,
    - local weather stations,

- flow meters and pressure sensors in irrigation networks,
- water-quality or EC sensors in canals and wells.

EO provides the wall-to-wall picture, while IoT provides high-resolution truth points. The DT-Agro architecture links the two: EO products are used to build and run the Digital Twin everywhere, and IoT data are used to support validation, calibration and state updating of the Digital Twin at specific locations, with the resulting effects propagating to surrounding areas through the model structure.**4.2 Integration of EO data**

The pre-processing and harmonisation of EO datasets are described in detail in D2.1 and in Section 3 of this report. Here, we focus on how these EO products are actually used by the Digital Twin.

EO data interact with DT-Agro along four main pathways:

1. **Forcing**
   - Meteorological forcing is based on a hybrid EO–station approach: AgERA5 is sampled at approximately 140 station locations, station-wise regressions are used for bias correction, and the resulting virtual stations are interpolated to a 1 km grid.
   - These 1 km meteorological fields (precipitation, temperature, radiation, humidity, wind) provide the daily forcing for the hydrological, crop and erosion algorithms (Sections 3.1–3.3).
2. **Parameterisation**
   - Static EO and geospatial layers are used to derive spatially distributed model parameters at 100 m resolution:
     - DEM → slopes, flow directions, flow accumulation, LS factors for RUSLE.
     - Land cover (CLC, CLC Backbone) combined with IACS → crop types, irrigation/non-irrigation, management classes, C and P factors.
     - Imperviousness (HRL) → impervious fraction per cell and impervious-aware SCS-CN parameters.
     - Soils (ISRIC, ESDAC, national maps) → hydraulic properties, rooting depth, water-holding capacity, erodibility (K).
   - These parameter rasters are the “bridge” between EO and the C++ simulation core.
3. **Dynamic vegetation and crop condition**
   - Time series of NDVI (and, in future, LAI/FAPAR) are used to derive:
     - NDVI-based crop coefficients (Kc) for evapotranspiration, and

        - time-varying cover factors (C) for erosion.
    - This allows DT-Agro to represent inter-annual and intra-seasonal changes in crop development and canopy cover, rather than relying on fixed Kc or C curves.

4. **Evaluation and state updating**
    - EO soil-moisture products and vegetation indices are used to:
        - evaluate modelled soil moisture and vegetation dynamics at regional scale,
        - identify systematic biases (e.g. too wet / too dry), and
        - design simple state-updating procedures, such as adjusting initial soil-moisture conditions and nudging modelled Kc towards NDVI-derived values.

At this stage, most EO integration is performed offline (i.e. products are pre-processed and then used as inputs or references for simulations). However, the architecture (Section 2) and database (Section 3) are already structured so that EO layers can be updated and re-ingested routinely, enabling a gradual move towards near-real-time operation.

4.3 Concept and role of IoT data

While EO provides spatially exhaustive information with revisit periods from days to weeks, IoT networks can deliver high-frequency, in-situ measurements that complement EO and model outputs in several ways:

- **Soil moisture sensors** in fields can provide direct information on the dynamics of soil water in the root zone, helping to:
    - validate and refine the soil-water balance,
    - adjust soil hydraulic parameters (e.g. water-holding capacity), and
    - test irrigation strategies at farm scale.
- **Local weather stations** (when connected through IoT interfaces or APIs) can supply near-real-time rainfall, temperature, humidity and wind data, which can be used to refine or temporarily replace the interpolated meteorological fields at high-priority sites.
- **Flow and pressure sensors** in irrigation infrastructure can monitor actual water abstractions and distribution losses, allowing DT-Agro to:
    - compare modelled irrigation demands with actual withdrawals, and
    - improve assumptions on conveyance and application efficiencies.

- **Water-quality sensors** (e.g. EC, nitrates, turbidity) in canals and streams provide an empirical basis for future design of modules on nutrient loads, agrochemical pollution and salinity, which are foreseen in later phases of DT-Agro.

In the current phase of WP2, the emphasis is on defining the interfaces that will allow such IoT streams to be integrated, rather than on large-scale deployment. The system is designed to treat IoT data as additional input layers or time series, stored in the same database structure as EO and station data, with explicit spatial references (coordinates, parcels, network nodes).

4.4 Technical integration and interfaces

From a technical perspective, the integration of EO and IoT data follows the same general pattern:

1. **Ingestion and storage**
    - IoT data streams (e.g. from gateways or web APIs) are ingested via dedicated Python scripts and stored in the meteo/ or eo/ domains of the DT-Agro database, using a consistent format (CSV/Parquet or NetCDF) and including all necessary identifiers (sensor ID, location, depth, variable, units).
    - Simple quality-control routines (range checks, spike detection, completeness checks) are applied to flag suspicious values and maintain a separate “quality” field.
2. **Linkage to model units**
    - Each IoT sensor is associated with a **model unit**: grid cell, IACS parcel, or network element (e.g. canal reach, pumping station).
    - Lookup tables (in the metadata/ domain) record this mapping, enabling the model or post-processing scripts to find, for any cell or parcel, the corresponding sensors (where available).
3. **Use within DT-Agro workflows**

Depending on the variable and application, IoT data can be used in three main ways:

- **Calibration / parameter refinement:**
    - Using multi-year soil-moisture time series to tune soil hydraulic parameters or irrigation efficiencies in representative locations.
    - Using flow-meter data to calibrate conveyance losses or on-farm application efficiency.
- **Validation:**
    - Comparing modelled vs. observed soil moisture, local ET estimates or flows to quantify model performance at high-resolution sites.

- **State updating (data assimilation in a broad sense):**
  - Adjusting modelled soil moisture (or other state variables) towards observed values at sensor locations at predefined intervals (e.g. weekly) and propagating this influence to neighbouring cells via simple spatial kernels.

4. **Interfaces for future services**
   - The Python orchestration layer supports the exposure of selected DT-Agro outputs (e.g. soil-moisture or irrigation-need maps) back to IoT platforms or farm-management systems, enabling two-way interaction: sensors feed the Digital Twin, and the Digital Twin returns actionable information to the field.

In practical terms, most of these mechanisms can be implemented without changing the C++ core: they are handled at the pre-processing and post-processing levels of the Python layer, which reads and writes IoT-augmented datasets within the existing database structure.

4.5 Current status and planned developments

At the time of this deliverable, the **EO integration** within DT-Agro is operational to a large extent:

- EO-driven parameter maps (DEM derivatives, soils, land cover, imperviousness) are available for the whole of Greece at 100 m resolution.
- The meteorological forcing chain (stations + AgERA5 → virtual stations → 100 m or 1 km grids) is implemented and is used in national-scale simulations.
- NDVI-based vegetation layers are processed and ready to be used for Kc and C factor estimation.
- Soil-moisture and auxiliary EO products are currently used for validation and simple state-update experiments.

The **IoT integration** is, by design, at an earlier stage:

- The architectural interfaces and database locations for IoT data have been defined, ensuring that sensor time series can be ingested and linked to model units with minimal changes to the existing codebase.
- Pilot use-cases (e.g. soil-moisture sensors and flow meters in specific irrigation schemes) are under prpeparation, in coordination with other WPs, to evaluate calibration and state-update procedures.

Future developments, beyond the scope of D2.2 but aligned with the overall project plan, include:

- operational use of IoT data for near-real-time updating of soil-moisture and irrigation-need estimates in selected areas,

- extension of the Digital Twin to include nutrient and pollutant loads, where IoT water-quality sensors and EO products (e.g. turbidity, chlorophyll proxies) will play a central role, and
- integration with farm-management tools and regional platforms, where DT-Agro will serve as a back-end engine using EO and IoT data to generate decision-support indicators.

In summary, the integration of EO and IoT data constitutes a core design element of DT-Agro. EO provides the spatial backbone and a consistent representation of the agro-hydrosystem across Greece, while IoT data provide local, high-frequency detail where available. The architecture developed in WP2 allows both types of data to be incorporated in a modular way, enabling DT-Agro to evolve beyond the project lifetime from an EO-driven model into a **data-enriched Digital Twin**.**4.6 Challenges and limitations**

Although DT-Agro has been designed as an open and extensible Digital Twin, its performance is inevitably constrained by the quality, density and consistency of the underlying data sources. Work conducted through parallel scientific studies and abstracts has highlighted a number of important limitations that directly affect the current implementation and guide the priorities for further development.

Meteorological reanalysis datasets.

A major challenge is the low accuracy of global metanalysis datasets when used directly for small-scale agricultural and hydrological applications in Greece. Recent evaluations of AgERA5 and MERRA-2 in the Nemea viticultural area showed substantial biases and errors for key variables such as precipitation and reference evapotranspiration, with performance strongly depending on local topographic and climatic conditions. Even AgERA5, which generally outperforms MERRA-2, was found to be insufficiently accurate for precision irrigation scheduling when used without local correction (Soulis et al., 2025).

These findings motivated the development of the virtual-stations concept and the spatially distributed rainfall and temperature gradients used in DT-Agro: reanalysis time series are first sampled at station locations, statistically corrected against local observations, and only then interpolated in space. While this hybrid approach significantly improves meteorological forcing, it still inherits uncertainties from both the station network (gaps, representativeness) and the reanalysis products.

Global soil datasets and soil information gaps.

A second critical limitation concerns the global and pan-European soil datasets (e.g. ISRIC SoilGrids, ESDAC), which are widely used for large-scale modelling but show very low accuracy when compared to detailed national data in Greece. A recent assessment against the Greek Soil Map, based on more than 10,000 field samples, found low overall accuracy for soil-texture classes (≈19–21%), low explanatory power ($R^2 < 0.2$) and high RMSE values (13–19% for

texture fractions), with errors systematically clustered in specific geomorphological and lithological settings (Gerontidis et al., 2025b).

Complementary work has demonstrated that these discrepancies propagate into key modelling outcomes, such as Curve Number (CN), RUSLE K-factor and texture-based indices like the Texture Quality Index (TQI), leading to substantial differences in erosion risk, runoff estimates and desertification assessments when global datasets are used instead of national soil data (Gerontidis et al., 2025a)

In response, the team has initiated the development of a national soil data hub that already integrates more than 17,000 georeferenced soil sampling points from multiple sampling campaigns and sources, and continues to expand. Using this data hub together with AI-based methods, new improved soil-property layers are being generated for Greece, aiming to replace or locally correct global products for DT-Agro parameterisation. This work is ongoing and, until the improved layers reach full coverage and maturity, soil-related parameters (e.g. hydraulic properties, erodibility) remain a significant source of uncertainty in the Digital Twin.

Gaps and uncertainty in EO- and IoT-based monitoring.

Even where local monitoring networks exist, **data gaps and sensor limitations** pose additional challenges. Studies on reconstructing missing solar-radiation measurements using empirical methods and machine learning over the Nemea station network showed that, although ML approaches can improve accuracy, they require additional effort and are not a universal solution; method selection remains case dependent and constrained by available predictors and computational resources (Soulis et al., 2024).

Similarly, the ongoing work on Sentinel-1/2-based soil-moisture mapping, supported by an IoT network of in-situ sensors, has underlined the complexity of reliably estimating near-surface soil moisture in a highly heterogeneous Mediterranean landscape and the importance of careful calibration and validation of machine-learning models combining SAR, multispectral and ground data (Kalivas et al., 2025).

At present, IoT coverage remains limited to specific pilot areas, and soil-moisture and other EO-derived products are mainly used for evaluation and exploratory state-updating experiments rather than for fully operational data assimilation across Greece.

Implications for DT-Agro and future work.

These limitations have several implications for the current DT-Agro implementation:

- Our original plan was delayed as we had to face alternative solutions for all this limiting factors and to face all these challenges. However, very important and innovative work resulted in this effort that has multiple benefits and increases DT-Agro project impact.
- Model performance is constrained by the quality of forcing and parameter datasets, especially in regions with sparse station coverage, complex terrain or poorly represented soils.

- Some components of the Digital Twin (e.g. erosion, future nutrient and pollution modules, soil-health indicators) are still affected by the uncertainties of global soil datasets and will only reach their full potential once the national soil data hub and AI-enhanced products are fully integrated.
- The EO/IoT integration is, by necessity, being developed incrementally: first as an EO-driven Digital Twin supported by hybrid meteorological forcing and improved soil data, and progressively enriched by Sentinel-based soil moisture and in-situ IoT measurements where available.

Overall, building a **national-scale Digital Twin of the Greek agro-hydrosystem is an ambitious, very important and long-term process.** The work carried out so far has already identified and quantified critical limitations in existing global datasets and has led to concrete solutions (virtual stations, spatial gradients, national soil data hub, Sentinel-IoT soil-moisture framework). Nevertheless, these challenges will continue to shape the roadmap for future developments and must be kept in mind when interpreting DT-Agro results and designing subsequent extensions of the system.

5. Graphical User Interface (GUI) and Run Control

The DT-Agro Digital Twin has been implemented with a deliberately simple, lightweight graphical user interface (GUI). The goal is not to provide a complex desktop environment, but a minimal wizard that:

- guides the user through the basic configuration steps,
- verifies that all required inputs are provided, and
- can be easily embedded into, or replaced by, other platforms and front-ends (e.g. web services, dashboards, external decision-support tools).

All advanced functionality (data management, parallelisation, logging) is handled by the Python orchestration layer and the C++ core. The GUI acts as a thin layer that exposes the main options in a transparent way and ensures that the Digital Twin can also be used interactively on a standard workstation.

5.1 Step-wise workflow

The GUI follows a stepwise workflow, where the user moves through a sequence of dialog windows using the *Previous*, *Next* and *Cancel* buttons:

1. selection of the model/project folder,
2. definition of input datasets and file formats,
3. specification of the simulation period and interpolation options,
4. selection of the model functions to be executed,

5. choice of which results will be written as rasters, and
6. confirmation of successful completion.

Each step is self-contained and shows the information required for the specific task, keeping the interface compact and easy to understand.

5.2 Main GUI windows

5.2.1 Project and model folder selection

The first step is to define the model directory, i.e. the folder that contains the utility files, station data and gridded inputs in the expected sub-folders. The GUI provides a simple text field and a *Pick address* button to browse to the desired folder. A short description below the field reminds the user of the expected sub-folder structure (utility files in _util, station data in _station, grid matrices in _matrix, etc.) (Figure 1).

Figure 1 illustrates the empty folder-selection window and the same window after a valid folder has been chosen and the expected sub-folders have been detected.

5.2.2 Input datasets and file formats

Once the model folder is set, the user is asked to specify the input datasets required for the run. The corresponding window lists all required input categories (DEM, flow direction, flow accumulation, Curve Number, rainfall, the crop coefficient Kc, reference evapotranspiration, soil water-content at saturation, shape factor, saturated hydraulic conductivity-Ks, imperviousness density, etc.).

For each category, the user can:

- select the file path using *Pick address*, and
- indicate the file type (raster, grid binary, or serialized grid) via radio buttons.

This design reflects the internal flexibility of the DT-Agro core, which can work either directly with standard rasters or with pre-serialized grid binaries for faster execution. An example of this window is shown in Figure 2.

Figure 2. DT-Agro GUI, selection of input datasets for each required category (DEM, flow direction, rainfall, Kc, etc.) and choice of file format (raster, grid binary or serialized).

5.2.3 Simulation period and interpolation settings

In the next step, the GUI prompts the user to define the starting and ending dates of the simulation, as well as the number of stations to be used in the interpolation of meteorological variables. The window provides simple date fields and a numeric entry for the number of stations (Figure 3).

These settings are passed to the meteorological-interpolation module, which then extracts the appropriate daily series from the bias-corrected virtual stations and builds the gridded forcing fields for the specified period.

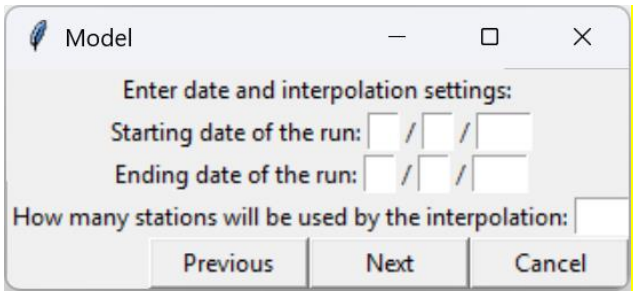


Figure 3. DT-Agro GUI – definition of simulation period and number of stations used in the meteorological interpolation.

5.2.4 Selection of model functions

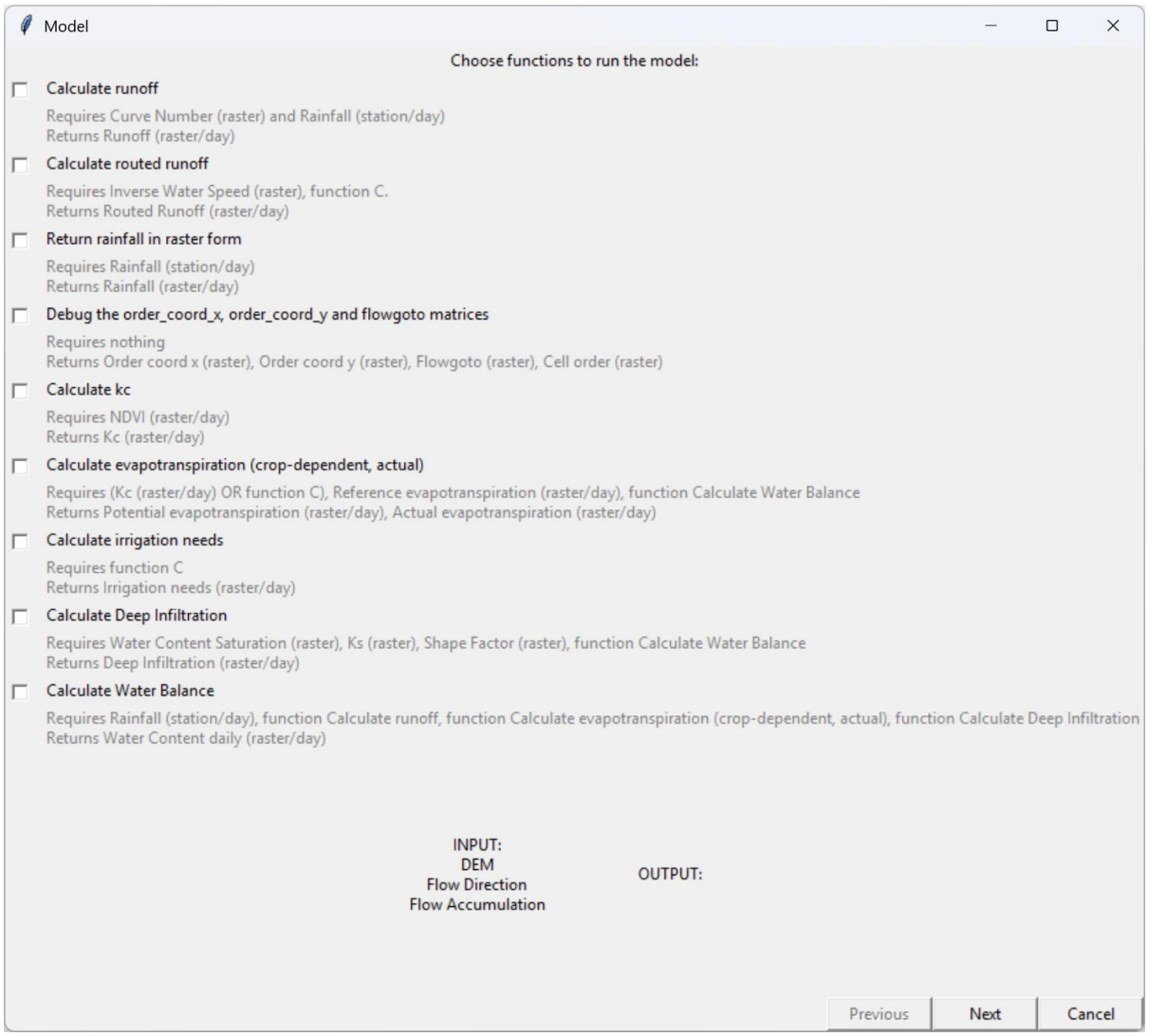
A central element of the GUI is the window where the user can **select which DT-Agro functions to execute**. The list includes options such as:

- *Calculate runoff*,
- *Calculate routed runoff*,
- *Return rainfall in raster form*,
- *Calculate Kc*,
- *Calculate evapotranspiration (crop-dependent, actual)*,

- *Calculate irrigation needs*,
- *Calculate Deep Infiltration*,
- *Calculate Water Balance*,

as well as debugging functions (e.g. *inspection of the order_coord_x, order_coord_y and flowgoto matrices*).

For each function, the window indicates the required inputs and the corresponding outputs produced. A summary at the bottom displays the minimal set of input grids required and the type of outputs generated. Figure 4 shows an example where the user selects only the function *Return rainfall in raster form*, while Figure 5 illustrates the message box that appears if the user attempts to proceed without selecting any function. This built-in check ensures that a run is launched only when at least one valid process has been selected.



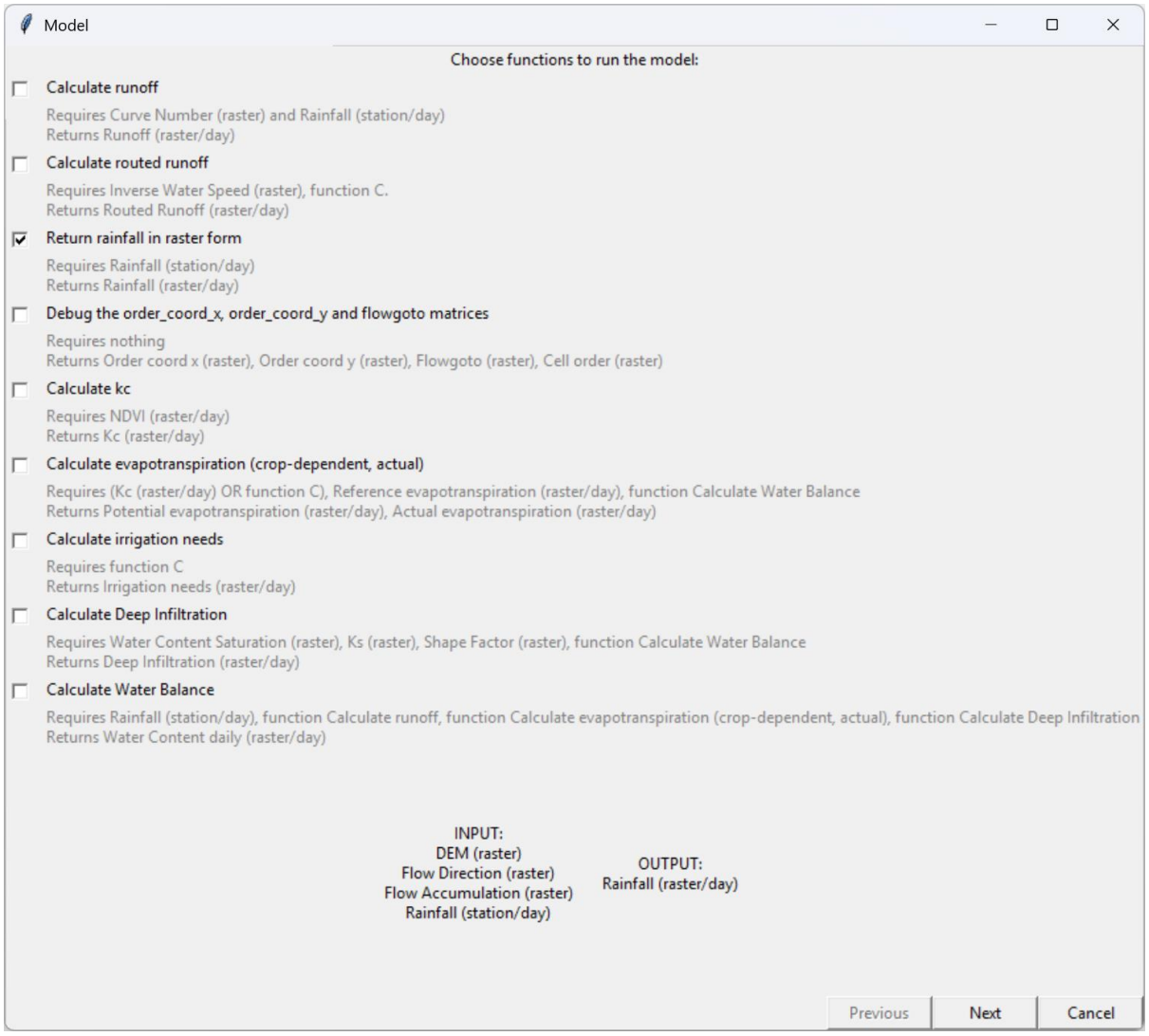


Figure 4. DT-Agro GUI, selection of model functions to be executed (runoff, evaporation, irrigation needs, water balance and others), with a summary of required inputs and outputs.

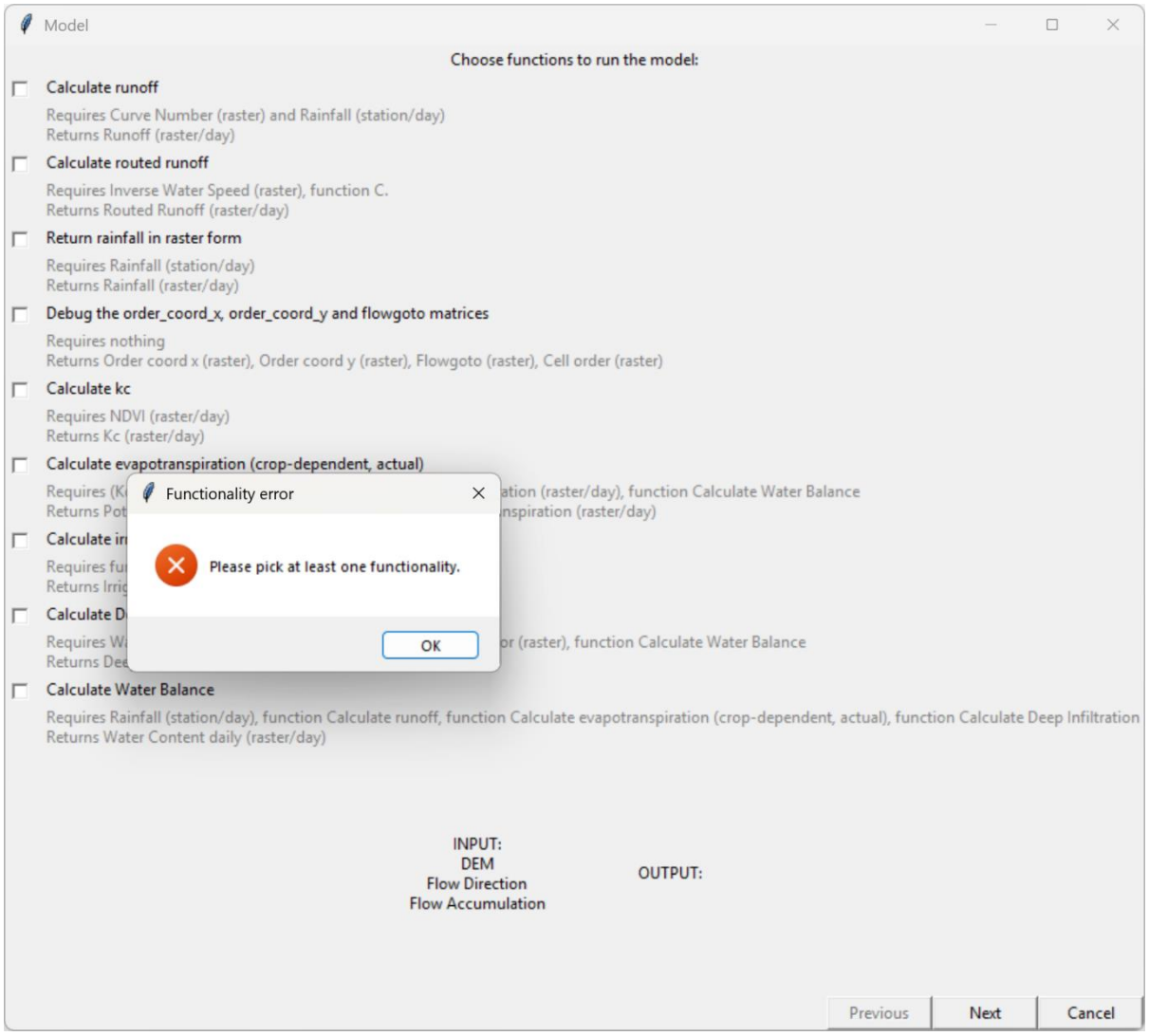


Figure 5. DT-Agro GUI, example of an internal consistency check: the system prompts the user to select at least one functionality before proceeding.

5.2.5 Output selection and completion

After the functions have been defined, a small window allows the user to specify which of the computed variables should be written as rasters. For example, in a run where rainfall has been converted from station data to a grid, the user can choose whether to export daily rainfall as raster outputs (Figure 6).

Once the computation is completed successfully, a final dialog box simply reports **"Done!"**, with the option to close the wizard (Figure 7). All outputs are written to the corresponding sub-folders in the model directory, where they can be further processed by the Python scripts or visualised using external GIS tools.

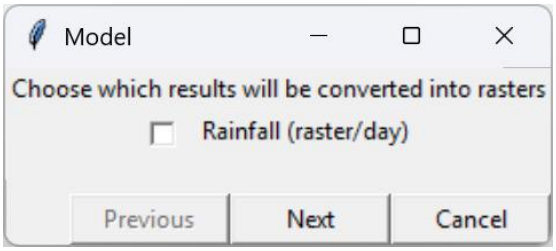


Figure 6. DT-Agro GUI, window for choosing which computed variables will be saved as raster outputs (e.g. daily rainfall).

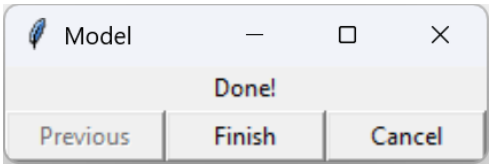


Figure 7. DT-Agro GUI, completion window indicating that the run has finished successfully.

5.3 Design philosophy and integration in other platforms

The examples above illustrate that the DT-Agro GUI is intentionally minimalistic:

- it exposes only the most essential options needed to configure and run the Digital Twin;
- it employs standard dialog windows and simple controls, minimising dependencies on specific libraries or operating systems; and
- it closely mirrors the underlying command-line and configuration-file options, so that any workflow defined through the GUI can also be scripted and automated.

This design makes it easy to:

- embed DT-Agro within other platforms (e.g. web applications, dashboards, cloud services) that may provide their own user interfaces, and
- use the current GUI as a reference implementation or a debugging tool for model runs during development and testing.

In this way, the GUI serves primarily as a lightweight front-end to a much richer and more powerful back-end, ensuring that the DT-Agro Digital Twin is both accessible for interactive use and ready for integration into broader digital-agriculture ecosystems.

6. Testing

The testing of DT-Agro builds on more than a decade of development and testing of the AgroHydroLogos modelling framework and extends it to the new Digital Twin configuration. Testing is pursued at three complementary levels:

- the hydrological and agro-hydrological core algorithms,
- the EO- and data-driven input layers (meteorology, soils, land cover, soil moisture), and
- system-level performance, with a particular focus on irrigation-water requirements and policy-relevant indicators at national and parcel scales.

Model performance is assessed using standard hydrological and agro-hydrological metrics such as Nash–Sutcliffe Efficiency (NSE), its logarithmic form (lnNSE) and Percent Bias (PBIAS) for streamflow, as well as RMSE, MAE and $R^2$for continuous variables (precipitation, soil properties, soil moisture, etc.).

6.1 Validation of the hydrological core

The hydrological algorithms implemented in DT-Agro are derived from the AgroHydroLogos model, which has been extensively tested under Greek and Mediterranean conditions.

The original spatially distributed continuous model was first applied at catchment scale, simulating daily water-balance components (soil moisture, runoff, deep percolation, actual evapotranspiration) and comparing them with observed streamflow and available soil-water information. These studies demonstrated a realistic partitioning between evapotranspiration, recharge and runoff under Mediterranean seasonality and provided the initial calibration of key parameters (e.g. SCS-CN, Brooks–Corey percolation, baseflow coefficients).

The same modelling core was subsequently upscaled and applied at national scale, producing a multi-decadal climatology of precipitation, reference and actual evapotranspiration, soil moisture, deep percolation and runoff for Greece. Simulated monthly flows at gauged stations showed satisfactory agreement with observations (NSE and lnNSE generally above 0.5; overall R^2~ 0.87; PBIAS mostly within ±25%), confirming that the simplified daily water-balance and routing schemes capture the dominant hydrological behaviour at basin and national scales.

These results underpin the configuration adopted in DT-Agro. The new component for sub-cell impervious areas, based on a simplified two-CN decomposition, was tested in parallel studies and shown to preserve classical SCS-CN behaviour for fully pervious or impervious cells, while improving runoff estimates in mixed cells where small impervious fractions dominate runoff generation. This approach has been integrated into the DT-Agro runoff module.

6.2 Validation of EO- and data-driven components

Because DT-Agro depends strongly on external EO and global datasets, a substantial part of the validation effort focuses on assessing and correcting these inputs before they are used by the model.

6.2.1 Meteorological forcing and “virtual stations”

Detailed evaluation of AgERA5 and related climate datasets for Greek conditions has revealed substantial biases and errors when used directly at local scales, especially for daily precipitation and reference evapotranspiration. Even where AgERA5 performs better than other products, its accuracy is insufficient for precision irrigation appliacations without local correction.

In response, DT-Agro uses a hybrid forcing strategy based on:

1. extracting complete AgERA5 time series at the locations of ~140 stations;
2. fitting station-specific regressions between observed and AgERA5 variables;
3. generating long, bias-corrected daily series for each “virtual station”; and
4. dynamically interpolating these virtual stations to the forcing 100 m or 1 km grid using topography-aware weighting and lapse-rate adjustments.

Internal cross-checks against withheld stations show clear improvements in bias, RMSE and correlation relative to raw AgERA5, making the virtual-station fields a more reliable meteorological forcing dataset for the Digital Twin.

This framework is currently being extended by incorporating additional stations, further increasing the representativeness and robustness of the forcing dataset.**6.2.2 Soil properties and erosion parameters**

Independent work comparing global/pan-European soil datasets (e.g. SoilGrids, ESDAC) with the Greek National Soil Map and an expanding soil data hub (currently >17,000 sampling points) showed very low accuracy for texture classes, high RMSE for texture fractions and weak explanatory power (R^2 often < 0.2). These errors propagate strongly to derived quantities such as soil hydraulic properties, CN, RUSLE K and texture-based indices, with clear implications for runoff and erosion modelling.

For DT-Agro, soil-related parameters (hydrologic soil group, water-holding capacity, K-factor, etc.) are therefore derived primarily from the Greek soil datasets and the soil data hub, rather than from uncorrected global products. Ongoing work applies AI methods to generate improved national soil maps from this data hub, which will further reduce parameter uncertainty in future model versions.

6.2.3 Land cover, crop mapping and irrigation information

The methodology adopted in DT-Agro for land cover and crops follows a novel approach merging different scales. Sentinel-2 time series, IACS parcel information and CORINE / CLC Backbone land-cover maps are combined to derive crop-type distributions and to link each IACS parcel to a representative grid cell with the same crop and similar conditions. This algorithm was shown to provide highly detailed and consistent information at farm level, while keeping computational costs manageable at national scale.

6.2.4 Surface soil moisture and state variables

In parallel, a Sentinel-based soil-moisture framework has been developed, combining Sentinel-1 SAR, Sentinel-2 vegetation indices and in-situ IoT soil-moisture sensors. Initial results show promising predictive skill for top-soil moisture across representative agricultural regions in Greece. These products are already being used to benchmark DT-Agro soil-moisture simulations and to test simple state-updating strategies (e.g. nudging initial soil moisture before the irrigation season).

6.3 System-level tests and evaluation of irrigation-water requirements

Beyond process-level tests, DT-Agro (and its predecessor configurations) has been evaluated at system level, particularly for estimating irrigation-water requirements and abstractions under realistic policy and data constraints.

In the national-scale application for Greece, the model was run on a daily time step for a 54-year period (1971–2024), producing gridded outputs of all water-balance components and crop water deficit. From these, net irrigation requirements and corresponding water abstractions were computed for every irrigated grid cell and aggregated to parcel, regional and national scales.

Key evaluation findings include:

- **Plausible national volumes and strong temporal variability.**

  Simulated total water abstractions in irrigated agriculture ranged from about 6000 hm^3 (≈490 mm) in a wet year to 7800 hm^3 (≈665 mm) in a dry year, with an average of ~6600 hm^3. These ranges and their inter-annual variability were consistent with expert expectations for Greek irrigation under current practices and climate variability.

- **Realistic crop-wise patterns.**

  Average net irrigation requirements for major crops (e.g. maize, cotton, alfalfa) were found to cluster around 380–420 mm, with clear spatial structure related to regional climate and soils. These values align with experimental evidence and local know-how, providing additional confidence in the combined ET, soil-water and irrigation modules.

This demonstrates that the DT-Agro framework can capture both the magnitude and variability of irrigation demand and directly support policy-relevant indicators.

- **Case-study comparisons for calibration and plausibility.**

  Where data were available from monitored farms, local irrigation networks (collective schemes) and farms equipped with water meters, simulated abstractions were broadly comparable with observed or reported volumes, confirming that the selected crop coefficients, stress thresholds and loss percentages are reasonable at system level. Some systematic overestimation for specific crops (e.g. deficit-irrigated olives and vineyards) has been identified, highlighting the need for crop-specific refinement of stress parameters and efficiency factors in future work.

Taken together, these results show that the DT-Agro approach reproduces key features of the national irrigation-water regime (magnitudes, spatial patterns, temporal variability and response to crop-pattern changes) in a way that is consistent with available data and expert knowledge, despite the scarcity and limitations of direct abstraction measurements.

6.4 Current status and outlook

The present DT-Agro implementation has been tested in offline mode for multi-decadal simulations over the entire Greek territory using:

- bias-corrected AgERA5-based meteorology (virtual stations and dynamic interpolation),
- EO-derived land cover and crop information consistent with IACS,
- soil parameters from available global and national datasets
- routing over the EU-DEM-derived drainage network.

These tests confirm that the recoded C++ core and the Python orchestration reproduce the large-scale water-balance patterns and irrigation-water indicators reported in previous national studies, while substantially improving openness, modularity and computational efficiency. At the same time, the validation exercises emphasise that performance remains constrained by the accuracy of input datasets (particularly soil and meteorology) and by the limited availability of high-quality abstraction and monitoring data.

Ongoing and planned work, supported by the expanding soil data hub, Sentinel-based soil-moisture products and IoT networks, will focus on more extensive calibration and validation in representative basins and irrigation schemes, explicit quantification of uncertainty stemming from global datasets and parameter choices, and tighter integration of EO- and IoT-based observations into the Digital Twin (e.g. through routine state updates and targeted local calibration).

This stepwise development strategy ensures that DT-Agro remains scientifically robust and transparent while progressively evolving into a fully operational Digital Twin for agricultural water management and policy evaluation in Greece.

References

Gerontidis, S., Soulis, K. X., Kairis, O., Palli-Gravani, S. Kopanelis, D., Kalivas, D. P. & Stavropoulos, A. (2025a). Assessing the impact of uncertainty in global soil property datasets on soil erosion predictions. [Poster presentation]. XIIth Scientific Assembly of the International Association of Hydrological Sciences (IAHS 2025), Roorkee, India. https://iahs2025.com/pdf/IAHS2025_Program_Book.pdf

Gerontidis, S., Soulis, K. X., Nikitakis, E., Kalivas, D. P., Kairis, O., Kopanelis, D., & Palli-Gravani, S. (2025b). Assessment of the accuracy of ISRIC and ESDAC soil texture data compared

to the Soil Map of Greece: A statistical and spatial approach to identify sources of differences. Soil Systems, 9(4), 133. https://doi.org/10.3390/soilsystems9040133.

Kalivas, D., Dosiadis, E., and Soulis, K. (2025). Estimating top-soil moisture at high spatiotemporal resolution in a highly complex landscape, EGU General Assembly 2025, Vienna, Austria, 27 Apr–2 May 2025, EGU25-11864, https://doi.org/10.5194/egusphere-egu25-11864.

Soulis, K., Dosiadis, E., Nikitakis, E., Charalambopoulos, I., Kairis, O., Katsogiannou, A., Palli-Gravani, S., & Kalivas, D. (2025). Assessing AGERA5 and MERRA-2 Global climate datasets for Small-Scale Agricultural Applications. Atmosphere, 16(3), 263. https://doi.org/10.3390/atmos16030263.

Soulis, K. X., Nikitakis, E. E., Katsogiannou, A. N., & Kalivas, D. P. (2024). Examination of empirical and Machine Learning methods for regression of missing or invalid solar radiation data using routine meteorological data as predictors. AIMS Geosciences, 10(4), 939–964. https://doi.org/10.3934/geosci.2024044.