PHISHES

PHysically-based Integrated
Soil HEalth Simulation platform

Predicting soil health to protect ecosystems

PHISHES
Digital Platform
Conceptual Design

Authors: Anthonj, Jannik; Meneses, Elbys; Jensen, Roar Askær; Falkena, Imke; Leirião, Sílvia

Disclaimer

This report was written as part of the PHISHES project under EC grant agreement 101157438. The information, documentation and figures available in this deliverable were written by the PHISHES project consortium and do not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

How to quote this document

Anthonj, J.; Meneses, E., Jensen, R.A., Falkena, I., Leirião, S. (2026). Outline of PHISHES Digital Platform (PHISHES Deliverable 1.6)

This deliverable is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Project Acronym: **PHISHES**

Project Full Title: PHysically-Based Integrated Soil HEalth Simulation Platform

Grant Agreement Number: Project: 101157438 - HORIZON-MISS-2023-SOIL-01

Type of Action: **HORIZON-RIA**

Deliverable number	**X.Y**
Deliverable name:	PHISHES Digital Platform Conceptual Design
Dissemination level:	PU - Public
Type of document:	R – Document, report
Deliverable Leader:	DHI
Delivery due date:	31/08/2026
Actual date of submission:	21/09/2026
Deliverable description:	Conceptual design of the PDP including mapping of data sources and flows, as well as description and workflow for the modelled scenarios. Technical specifications in Appendix.
WP / WP number:	WP1
Lead author(s):	Anthonj Jannik, Elbys Meneses and Roar Askær Jensen
Contributors:	Imke Falkena and Sílvia Leirião
Reviewers:	All beneficiaries

Version log

Version	**Date**	**Issued by**	**Description**	**Summary of changes**
1.0	31/08/2025	DHI	Original	
1.1	31/08/2026	DHI	Revised	Updates resulting from the technical review process. Inclusion of the system's architecture and other technical specifications.

Table of Content

Table of Content 4
List of Figures 5
List of Tables 5
Executive Summary 6
Keywords 7
List of acronyms and abbreviations 7
1 Introduction 9
1.1 Objectives and overall functionality of the PDP 9
1.2 The contents of this report 10
2 Introduction to PHISHES Digital Platform, functionality and main elements 12
2.1 General application 12
2.2 Main elements 12
2.3 Functionality of PDP during and beyond PHISHES 14
2.4 The model trains and their overall functionality 15
2.5 The preprocessing part of the PDP 15
2.5.1 The need for a geographically oriented platform 15
2.5.2 Structure of the site-specific data repository for model input data 16
2.5.3 The simulation part of the PDP 16
2.5.4 The post-processing part of the PDP 17
2.5.5 The script and model code repository 17
2.5.6 Expansion of the PDP 17
2.6 Work and data flow for a model scenario 17
3 Focus of the test cases and the addressed pollution 19
4 The model trains 22
4.1 Train 1, Hydrus-PHREEQC-MODFLOW (Örnsköldsvik, lab. soil tank) 22
4.2 Train 2, Daisy–MIKE SHE ECO Lab (Zelivka, Koprzywianka) 23
4.3 Train 3, MODFLOW6-UZF6 (Rotterdam) 25
5 Data needs and data structure 26
5.1 Introduction 26
5.2 Inventory of data sources relevant to the PDP 26
6 Outline of IT architecture and user interface 30
APPENDICES 37
Appendix A PDP System Architecture: Technical Specifications 37
Appendix B The selected model trains 56
Appendix C Data requirements for model trains 61
Appendix D Scenarios PHISHES (Milestone 3) 73

List of Figures

Figure 1.1 Connections between soil properties, soil functions and eco-system services. PHISHES targets quantitative simulation of the blue connection arrows. 9

Figure 2.1 Workflow and main elements of the PHISHES Digital Platform (PDP) 13

Figure 2.2 Main data flows and repositories of the PDP 14

Figure 4.1 Train 1D Hydrus – PHREEQC – MODFLOW-2005 with inter-model data flows (SZ = saturated zone, UNZ = unsaturated zone, SWI = surface water interface, AWI = air water interface) 23

Figure 4.2 Model train 2 with inter-model data flows and train outputs (GW = groundwater, LAI = leaf area index) 24

Figure 4.3 Model train 3, coupling with inter-model data flows between MF6 saturated and unsaturated zone + external coupling to ponding reservoir model. With UZ and SZ for unsaturated zone flow and transport respectively, UZF and UZT for unsaturated zone flow and transport package respectively and GW for groundwater flow. 25

Figure 6.1 Selection and view of data for download from the Data repository (example from other existing application) 32

Figure 6.2 Menu for selection of scenarios for I or description (example from other existing application) 33

Figure 6.3 Menu for illustration of indicator results and comparison of key indicator results to those of other alternative scenarios (example from other existing application) 34

Figure 6.4 Multi criteria numerical comparison of selected indicators or groups of indicators (example from other existing application) 35

Figure 6.5 Graphical Multi criteria comparison of four scenarios (example from other existing application) 36

Figure 6.6 PDP context: platform components, user-supplied model trains, and external services. 38

Figure 6.8 PDP component relationships across the data, compute, contract, tooling, and presentation planes. 41

Figure 6.9 Backend elements and workflow of the PDP platform hosted tier. 43

List of Tables

Table 2.1 Identification of which software used to build model trains is proprietary and requires a license. 12

Table 3.1 Test cases focusing on agricultural soils 19

Table 3.2 Test cases focusing on heavily contaminated soils 19

Table 3.3 Test cases focusing on urban soils 21

Table 5.1 Data types supported by the PDP with alternatives to local data 27

Table 6.1 Overview of the five logical planes and the PDP components assigned to each. . 39

Table 6.3 Principal PDP interfaces, the information exchanged, and the applicable contract and access model. 44

Executive Summary

The project PHysically-based Integrated Soil HEalth Simulation platform ("PHISHES") is a cooperation project between 9 European research organizations. It focuses on physically-based integrated soil health simulation and is financed through the Horizon Europe Mission: Soil deal for Europe.

This report constitutes Deliverable 1.6 of the project and articulates the first overall layout of the PHISHES Digital Platform (PDP), which will be one of the main outputs of the project.

During the project execution, the PDP will provide a protected data exchange platform and data repository shared by the nine participating research institutions. The PDP will work as a common test bench for the methodology and software tools being developed and applied during the four-year project.

The PDP layout reflects and supports the common approach to analyses that will be applied in five selected practical test cases, where a range of measures to improve soil functions are being tested and their efficiency in benefiting the targeted ecosystem services are being evaluated and quantified. The test cases focus on various soil functions of agricultural soils, heavily polluted soils, and urban soils, and on some of the important ecosystem services associated to these soil functions.

The methodology embedded in the PDP is centred around the applications of predeveloped, well proven and generally applicable software packages. These packages are being linked together in model trains to best analyse the particular challenges at the test sites.

This leads to a layout of the PDP with four main elements:

- A preprocessing and input data repository
- A simulation repository holding the specific files for scenario simulations and the results of these simulations
- A post-processing and decision support section with overall results and conclusions
- A script repository holding the pre- and post-processing tools developed.

The presented layout is expandable to other application sites by downloading and can hereby facilitate the application of the developed technology elsewhere in Europe after the project finalisation.

To ease such applications, an inventory of data sources holding general European or global data relevant to the use of the methodology in the PHISHES test cases or at other locations is included in the report.

Furthermore, the report provides:

- an overview of the addressed pollutants and measures to be tested in the cases studies
- specifics on the selected model trains that constitute the fundamental elements in the approach to modelling, and
- descriptions of the scenarios planned to be analysed in the various test cases.

While the data structures, the processing software and the conclusions will be publicly available, some of the data included in the test cases are confidential. The access to the populated data repositories and model input and output related to the test cases will therefore be restricted to the project participants.

Keywords

Web-portal design, soil functions, ecosystem services, scenario simulation, hydrological models, soil chemistry

List of acronyms and abbreviations

AWI	Air water interface
DMP	Data Management Plan
EAB	External Advisory Board
EC	European Commission
GA	Grant Agreement
GLA	General Assembly
GUI	Graphical User Interface
PC	Project Coordinator
PCT	Project Coordination Team
PDP	PHISHES Digital Platform
PMC	Project Management Committee
PO	Project Officer
QA	Quality Assurance
REA	European Research Executive Agency – European Commission
SWI	Surface Water interface
SZ	Saturated Zone
TL	Task Leader
UNZ	Unsaturated Zone
UZ	Unsaturated Zone
WP	Work Package

WPL	Work Package Leader

1 Introduction

The PHysically-based Integrated Soil Health Simulation platform (PHISHES) is a four-year research project under EU's Horizon Europe programme. The project focuses on providing quantitative simulation tools for analysing the efficiency of various measures to improve soil functions and their impact on key ecosystem services such as food production, sufficient and sufficiently clean water supply and retention of storm water (the blue arrows in Figure 1.1).

Figure 1.1 Connections between soil properties, soil functions and eco-system services[1]. PHISHES targets quantitative simulation of the blue connection arrows.

This report constitutes Deliverable 1.6 of the project and describes the overall layout and functionality of the PHISHES Digital Platform (PDP) – a soil and water platform for sharing of data, tools and scenario results to be used both by project partners, stakeholders and for dissemination purposes.

1.1 Objectives and overall functionality of the PDP

The PDP will provide a customised digital platform to facilitate the application of soil and water scenario models. Such models are central elements in the methodology and will be applied in Work Package 4 (the case studies) for testing the methodology.

The user requirements for the PDP elements stems from the pre- and post-processing needs of the planned scenario simulations to be applied in the test cases to quantify the efficiency of various

[1] Calvaruso et al. (2020) in "The Soil Fresco, 2023"

soil pollution alleviation measures and land use management practices in improving soil functions and associated ecosystem services.

The PDP will allow for integration of local monitoring data with data streams from generally available international data sources, such as global weather and climate models, soil databases and Earth Observations. These data will facilitate quantitative spatial analysis at different scales (from laboratory to water catchment scale) and ensure interoperability of databases and models.

During the PHISHES project, the PDP will serve as:

- a “test bench” for the developed technology
- a protected data repository for the partner institutions’ internal sharing of data
- a repository for the model input and output, and
- a library for the developed processing programs.

In its final version the PDP will also host a number of tested post-processing tools for evaluation and ranking of model scenario results for decision-making. The latter parts of the PDP will be further described in the next version of PDP documentation (Deliverable 1.1).

During the project and beyond, the PDP will serve as a container for the developed methodologies outcomes and conclusions and will be designed to be accessible and applicable to any site in Europe.

Model simulations of the types dealt with in PHISHES are normally most accurate when based on high quality, locally observed data. However, it must be accepted that such data are not always available at all sites where it would be relevant to analyse how the local soil functions and associated ecosystem services may be improved and where results from the simulation tools introduced and tested under PHISHES should therefore be relevant and appreciated. In addition to its role as a repository for local site data for the PHISHES test cases, the PDP will therefore also provide links to data bases with generally available data with a European coverage. This will facilitate upscaling of the developed technology as well as its application at more data sparse locations.

Since some of the handled data are sensitive, the access to the PDP will be restricted to the project partners during the project lifetime, and no restricted data will be accessible to people outside of the project staff as described in more details in the Data Management Plan[2] .

1.2 The contents of this report

Section 2 below describes the overall layout and the main elements of the PDP. It introduces the model trains that constitute the core of the system along with the data repositories with pre- and post-processing tools that facilitate the modelling workflows and the evaluation of scenario results in the various test cases. The aim of section 2 is merely to ease the reading of the following more detailed sections:

- Section 3 summarising the focus and the pollution components addressed
- Section 4 describing the model trains to be applied in the various test cases and the flows of data between the individual models
- Section 5 specifying the identified data needs to the PDP and its structure
- Section 6 giving examples of user interfaces of similar platforms.

[2] Deliverable D7.2 of the PHISHES Project, Jan 2025

It has been the intention to keep the individual sections rather short and concise to ease reading while large tables and more detailed explanations are included as appendices to the report.

Appendix A - Has extensive information on the facilities of the selected model trains

Appendix B - Lists the data requirements of the models

Appendix C - Describes the planned test case model scenarios

More detailed information on the model trains, the scenarios considered, and the nature of the simulated soil functions will be provided in upcoming deliverables.

2 Introduction to PHISHES Digital Platform, functionality and main elements

This section introduces the focus and functionality of the main elements of the PDP. More detailed descriptions of each element are given in the following sections.

2.1 General application

The PDP is designed for preparation, execution, and evaluation of numerical modelling scenarios quantifying how certain possible measures in various soil contexts may improve typical soil functions and their associated ecosystem services. It is a main goal of the PDP to develop a generally applicable yet transparent methodology that eases the use of advanced numerical modelling tools for such purposes at any location in Europe. During the PHISHES project the applicability of the methodology will be demonstrated through four selected case studies representing typical agricultural, contaminated and urban soil contexts.

The computationally intensive analyses will take place on the user's own IT platform, which will also host a copy of the PDP repositories (described below) for the user's focus area and the PDP scripting library easing the population of these repositories.

This means that the various applied modelling packages shall be installed on the user's computer and potential licenses for their application obtained. The PDP will include links for downloads of software and information on how to obtain software licences. The software that is proprietary and requires a license is presented in Table 2.1.

Table 2.1 Identification of which software used to build model trains is proprietary and requires a license.

Software	License requirement
MIKE SHE	Commercial licence required
MIKE+ ECO Lab	Commercial licence required (add-on to MIKE SHE)
Hydrus 1D	No licence required
PHREEQC	No licence required
MODFLOW 6	No licence required
MODFLOW 2005	No licence required
MT3D	No licence required
Daisy	No licence required
TKTD	No licence required

The PHISHES project will create a central version of the PDP including the data and analyses for all the selected case study areas and the overall conclusions and decision support aspects. Access to the full central version will be restricted to the project partners, as some data used in the analyses are confidential.

The central version of the PDP will be maintained throughout the project period, and its open elements will be accessible as further described below.

2.2 Main elements

The model trains constitute the fundamental elements of the PHISHES methodology. Despite the differences in focus and specific data needs of the model trains, they all handle input data in the

same overall manner and generate similar types of output. The PDP provides a shared general approach to scenario analysis that will be used in all case studies to quantify the impacts of certain drivers and measures on targeted soil functions and ecosystem services.

Therefore, the PDP's data repositories and data processing facilities will be designed to ease the application of the model trains in scenario analyses, both in the scheduled case studies and beyond the PHISHES project for similar studies at other locations. The concept has resulted in the following four main elements of the PDP as illustrated in Figure 2.1 and Figure 2.2:

1. **A pre-processing part**
   for gathering, storage and preprocessing of site- and scenario-specific input data for the modelling trains
2. **A scenario simulation part**
   containing the input and output of scenario runs in model-specific formats
3. **A post-processing part**
   holding and storing result evaluations and calculated indicators of the efficiency of the various analysed measures
4. **A script and model code repository part**
   holding download links to external data and to the necessary software packages as well as pre- and post-processing scripts and codes for interlinking of the various models in each train.

During the course of the PHISHES project, the PDP will furthermore function as a protected platform for storage of (sometimes sensitive) data and for general data sharing within the partners.

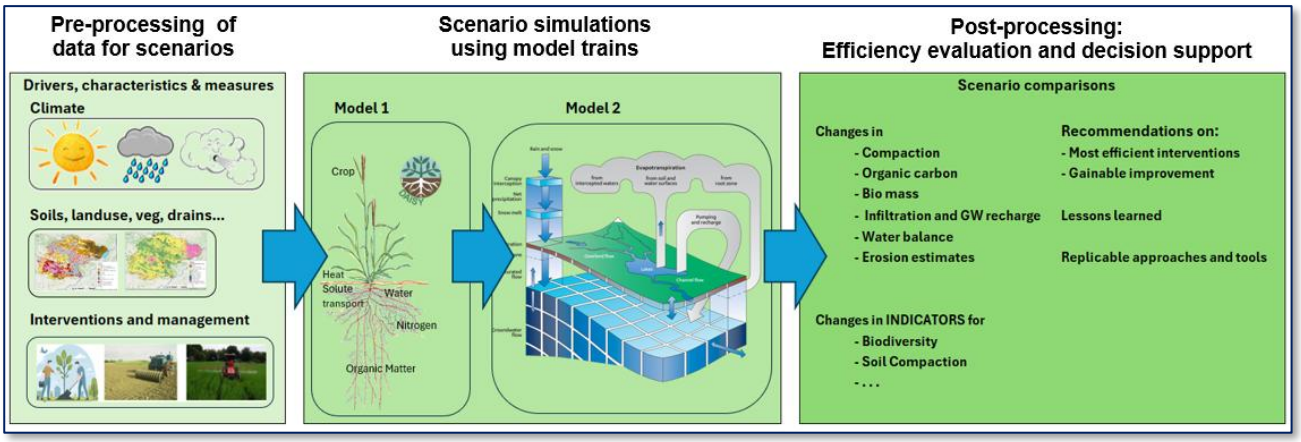

Figure 2.1 Workflow and main elements of the PHISHES Digital Platform (PDP)

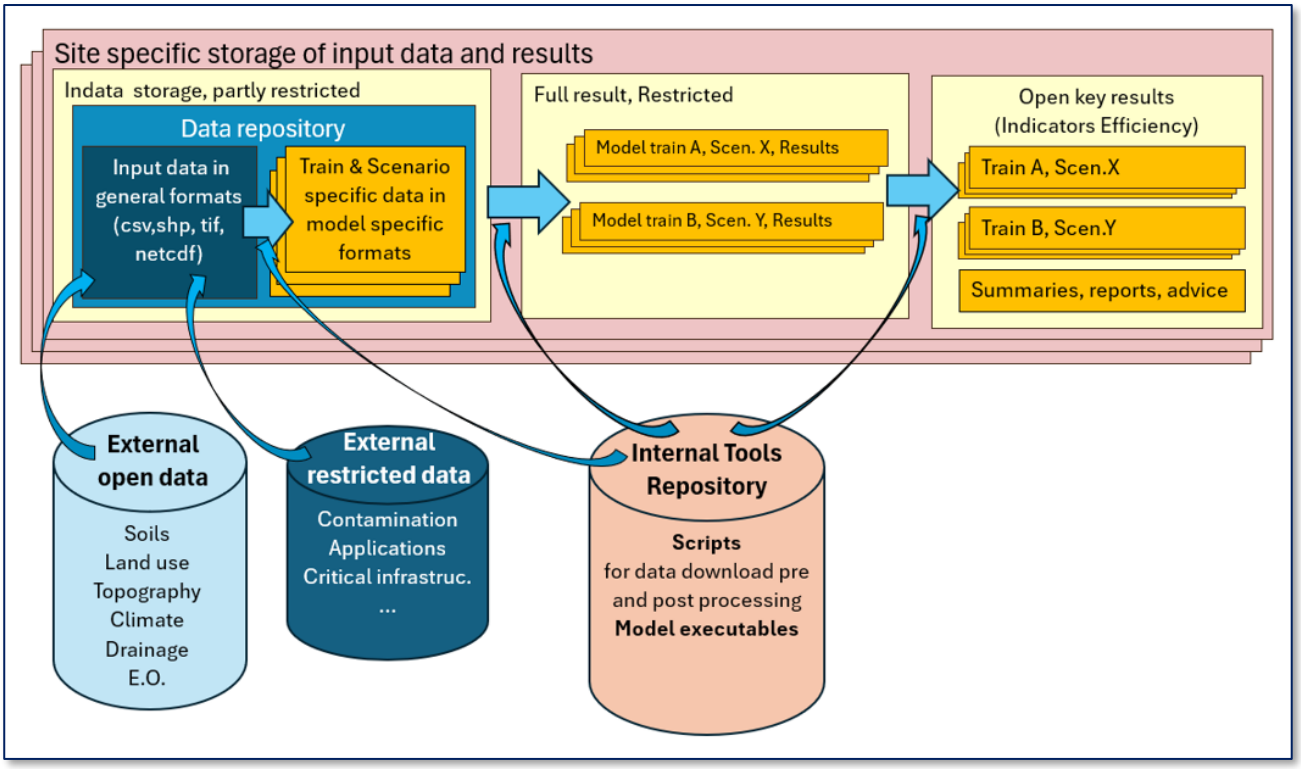

Figure 2.2 Main data flows and repositories of the PDP

2.3 Functionality of PDP during and beyond PHISHES

During the project period, the PDP will function as a common work and test bench for the PHISHES methodology and developments on the selected test cases.

The PDP will be hosted on Azure with the bulk of the computational work taking place on the partners' own hardware, with access to the data stored in PDP. However, some computational capacity will be allocated in Azure in connection with the PDP implementation to facilitate central testing of workflows and data processing.

The PDP will be designed to ease the download of software and free data to users' own premises for reuse of the methodology, the implemented workflows and the data structures allowing analyses of other locations. However, maintenance beyond the PHISHES project period ensuring compatibility between the PDP scripts libraries, external databases and model input and output formats is not within the budgeted project scope and will require extra funding.

Upon the launch of the first version of the PDP, August 2026, a dedicated discussion will be initiated with EUSO on the IT requirements for transferring the system, including maintenance and operational costs, for which we will have an initial estimate.

Ongoing maintenance is essential to ensure the PDP remains secure, performant and aligned with operational requirements. The platform integrates data, services and software components from multiple sources and therefore requires periodic maintenance to ensure continued functionality, compatibility and security. The associated costs and IT prerequisites will be presented and discussed with EUSO.

As some of the local data used by PHISHES are strictly confidential, external users of the methodology will not have access to all input data used in the test case scenarios or to the detailed scenario results. However, access will be provided to overall results and conclusions that cannot be traced back to sensitive input data.

Furthermore, public access will be provided to studies, methodologies and findings via planned research publications.

2.4 The model trains and their overall functionality

The model trains constitute interlinked sets of pre-developed, proven and generally applicable physically and biologically based simulation codes - all with unique facilities for simulating water and solute flows and bio-chemical processes in the actual field or catchment being analysed.

The model trains are capable of simulating the temporal variation of a number of internal state variables such as water contents and flows, evapotranspiration, nutrient uptake, pollution degradation, solute transport, etc. The results of such simulations can be used directly or indirectly to quantify how certain soil functions and associated ecosystem services may change in consequence of alterations in model drivers, vegetation, land use/land management or other human interventions.

To produce such results, the models need information on the model area in the form of lists or arrays of site-specific information holding:

- **Static or semi-static parameters**, such as soil and vegetation characteristics, topography, geology, drains and initial conditions.
- **Time varying conditions** in the model area. Examples of such inputs are climatic variables, such as precipitation and temperature, but also management conditions in the form of fertilizer application, cropping patterns, land management practice, etc.

Some of these model inputs are considered drivers, i.e. elements that cannot be changed by local management strategies, while others are considered measures, i.e. elements that may be altered by local managers to improve the targeted soil functions and ecosystem services. The efficiency of various management alterations (the measures) will be evaluated by comparing the resulting status of the soil functions or ecosystem services in focus, as simulated with and without the alteration. Likewise, the impacts from changes in the drivers can be predicted by comparing simulations with and without taking these changes into account.

As described in more details below, the PDP will facilitate the execution of such methodology by assisting the setup of input files for the model trains (the Pre-processing part), providing a structure for storing and managing the input and output of the scenario simulations (the Scenario simulation part) and availing post-processing tools for validation of the impacts of the various optional measures on the targeted soil functions and eco-system services and illustrating the results (the Post-processing part).

For application of the PDP new locations, the end user will have to select the most appropriate model train for the soil functions and ecosystem service challenges in question.

2.5 The preprocessing part of the PDP

2.5.1 The need for a geographically oriented platform

Although the simulation tools applied in PHISHES are generally applicable computer codes, their *practical* application relies entirely on a detailed set of site-specific input data that, in their entirety, describe the characteristics of the particular field or the catchment being analysed. Hence, the setup and the calibration of the model trains will be site-specific and will likely involve locally observed field data in combination with generally available data from European or even global data sources.

Due to the large computational intensity of detailed soil health simulations, the foreseen scenario analyses will be constrained to localized areas rather than national or even continental scales.

Furthermore, some of the general data sets with high temporal and spatial resolution (such as climate data sets) are quite voluminous, and it is practical to work with subsets of such datasets that cover only the localized project area.

To make the PDP geographically flexible and generally applicable it is therefore organised according to geographical locations, the size of which can vary from field to catchment size or even beyond. In the PHISHES activities, the case study areas are examples of such locations.

2.5.2 Structure of the site-specific data repository for model input data

As part of the PHISHES project, the PDP data repositories will be established for each of the four selected test case locations. Even simulations under the fifth test case (the BRGM Test Facility - Soil Laboratory) *could* be stored in the PDP framework, although the experiments in the lab will not necessarily relate to specific physical location.

The data repository will hold data for a specific location (test case) and will be a folder structure organised after data types relevant for the specific model train(s) to be applied at this location.

Sub-folders will be organised after types of data such as:

- Climate
- Hydrological catchment characteristics (topography, artificial and natural drainage systems, flow regulation, etc.)
- Soils (soil type distribution, soil characteristics (texture, mineralogy, hydraulic characteristics nutrients, etc.))
- Hydrogeology and groundwater
- Vegetation (land use, crop rotation patterns, vegetation characteristics (LAI, Rooting depths))
- Initial conditions
- Calibration data (stream flow and water level records, groundwater levels, observations of actual evapotranspiration and Leaf Area indices, cropping and land use patterns, vegetation status, etc.)

If some of these data types are not relevant, the directories may be kept empty or deleted.

As illustrated in Figure 2.2, the data repository has two parts. **Part 1 of the data repository** holds general data for the site in general data formats (comma separated text files , GIS layers as shape or Geo tiff files or as 2D-3D time varying netCDF files. Data in part 1 will often be shared by several scenarios.

Where download of data takes place from external data bases using preprocessing scripts from the PDP script library, the data will be downloaded to Part 1 of the repository. Development of scripts converting data in the standard formats to the model-specific formats required by the various model trains and inclusion of these scripts in the Script repository will be the responsibility of the model owner (the partner organisation developing or applying the particular model). If the data for a location already exist in the model-specific format, the user may also choose to keep the data format and skip the general formats.

2.5.3 The simulation part of the PDP

Part 2 of the data repository is a complete set of model input data for a given scenario in model-specific formats (parameter files, time series data, 2D, 3D input files, etc.). It is stored in a scenario-specific folder that will also hold the model results when the simulation is completed. The series of such folders of input and output data for the simulated scenarios constitute the **Simulation part of the PDP**. To analyse the impact of any scenario its results must be compared to those of an associated baseline. Hence, both baseline and alternative will normally be stored and available for the evaluation routines at the time of comparison. The structure therefore allows

for storing families of scenarios, each consisting of a baseline scenario and several possible measure-alteration or driver scenarios. As numbers of alternatives tend to grow during a project, the structure also allows for deleting certain scenarios if they become obsolete.

2.5.4 The post-processing part of the PDP

The post-processing part of the PDP is where the various scenarios are compared and where the comparisons are stored for analysis and evaluations. Typically, the results of a certain management (or planning) scenario are compared to a baseline to quantify the effects of the implemented measures which, in turn, may be compared to those found for different measure implementations. Scripts from the scripting library will be used to compare results in the same standard formats as used in the input data repository.

Some result comparisons will be straightforward such as differences in simulated compound concentrations at a certain location while other impacts such as changes in biodiversity may have to be addressed more implicitly via sets of predefined indicators. Such sets of indicators and how they will support policy decisions will be elaborated in other work packages, and their implementation in the PDP will be reported in deliverable D1.1.

2.5.5 The script and model code repository

This repository will hold all the data processing scripts produced under PHISHES. It will likely be in the form of Python notebooks, built on top of pre-existing Python libraries for interaction with the specific input and output data of the various modelling packages.

The PDP is designed for all simulations to run on the users' premises, and all modelling packages included in a selected model train should therefore be installed on these premises. Links to installation guides and information on how to obtain the necessary application licenses will be included in the script library.

Furthermore, the repository will contain links to installation of the various modelling packages as well as details on how to obtain the necessary licences for model application for the commercial software packages.

2.5.6 Expansion of the PDP

When the need for analysis of a new location arises, a new location entity will be created with the same repository structure as for other locations. Hence, the PDP will be organised as a set of identical data structures each covering only one of the application areas (a field or a catchment).

The data repository will be a folder structure organised after data types relevant for the specific model train(s) to be applied in this location.

2.6 Work and data flow for a model scenario

A typical workflow for population of the data repository and establishment of a model scenario will be:

1. **Identify soil health problem and geographical extent**
   - Evaluate the soil functions and ecosystem service(s) in question including which processes should be represented in the analysis
   - Select an appropriate model train
   - Create a georeferenced Shapefiles with boundaries for download of general data
2. **Establish a location entity of the PDP on users' computer**
   - Download data structure and scripts using the boundary shape
   - Download and test model packages

- Get licenses – if needed

3. **Identify the data needs for the selected model train**
   - Identify availability of local data and load these into the PDP data repository
   - Identify relevant generally available data and populate the repository (via PDP links and scripts)
4. **Establish model parameter files for the baseline scenario**
   - Through the Graphical User Interfaces (GUIs) of the involved models: Generate model-specific input files (configuration, parameters, time series, etc.) in a specific scenario folder (typically a baseline folder)
   - Populate the input files with data from the general data repository (manually and via available scripts)
   - Fill in possible data gaps with assumed values
   - Calibrate the models and evaluate their performance
5. **Simulate the baseline scenario**
   - Run the models with the tools and methodology developed in PHISHES
   - Use scripts from the repository to convert the results to supported standard formats
   - Activate the post-processing scripts to quantify the indicators of the state of the soil functions or ecosystem service(s) in focus under the baseline scenario
6. **Establish and simulate the alternative scenario**
   - Identify the measure(s) to be tested and how they influence the model input
   - Copy the baseline to a new scenario folder
   - Introduce the necessary changes in the input files to represent the alteration(s) in the measures
   - Simulate the alternative scenario and convert the results to the supported standard formats
   - Activate the post-processing scripts to quantify the indicators of the state of the soil functions or ecosystem service(s) in focus under the alternative scenario
7. **Extract and analyse impacts**
   - Activate the post-processing scripts to compare the state of the soil functions or ecosystem services (the baseline vs. the alternative)
   - Compare the impacts with those of other alternatives, possibly by multicriteria analyses
   - Rank/prioritize the alternatives
8. **Conclusion and upload of key findings**
   - Draw conclusions
   - Upload conclusions and key results to the open decision support part of the PDP
   - Document the analyses

3 Focus of the test cases and the addressed pollution

Together, the PHISHES test cases cover three different soil environments:

- Agricultural soils (the Zelivka and Koprzywianka cases)
- Contaminated soils (the Örnsköldsvik and BRGM Soil Laboratory cases)
- Urban soils (the Rotterdam case)

The various test cases focus on different aspects of soil health and on different ecosystem services associated with soil functions and catchment processes. Table 3.1, Table 3.2 and Table 3.3 provide a summary including the challenges and ecosystem services addressed, the targeted soil functions, the key drivers considered and the measures to be analyzed. A more extensive description of the various test cases and the planned scenarios are given in Appendix C.

Table 3.1 Test cases focusing on agricultural soils

	Zelivka	Koprzywianka
Main pollution challenge	Drinking water quantity and quality for the city of Prague and close vicinity threatened by agricultural and urban pollution	Freshwater quality in the river which is threatened by pollution from agriculture (orchards, fields) and old mining sites
Most important ecosystem services	• Regulation of water quality • Biodiversity conservation • Regulation of erosion and landslides • Food and water supply	• Regulation of water quality • Biodiversity conservation • Regulation of erosion and landslides • Food and water supply
Target soil functions	• Stable physical support for vegetation • Storage, recycling and transformation of organic matter • Habitat for soil organisms • Interaction with chemical composition of atmosphere • Retention and supply of nutrients • Filtering, buffering and degradation of pollutants • Water retention, circulation, infiltration and groundwater recharge • Impact of soil drainage on transport of pollutants and nutrients	• Stable physical support for vegetation • Storage, recycling and transformation of organic matter • Habitat for soil organisms • Interaction with chemical composition of atmosphere • Retention and supply of nutrients • Filtering, buffering and degradation of pollutants • Water retention, circulation, infiltration and groundwater recharge
Key drivers to consider	• Climate change	• Climate change
Measures to analyse	• Agricultural practices • Pesticide use • Stream modification and adaptation • Drainage scenarios • Land use change (afforestation)	• Agricultural methods • Pesticide use • Stream modification • Land use change • Sludge recycling (on fields)
Model trains proposed	• DAISY + MIKE SHE + MIKE ECO Lab • MIKESHE SHE + MIKE ECO Lab	• MIKE MSHE + MIKE ECO Lab

Table 3.2 Test cases focusing on heavily contaminated soils

	Örnsköldsvik	BRGM Lab
Main pollution challenge	Soil, wetland and groundwater threatened by a PFAS plume	(Near 'real'-scale laboratory experiment and remediation technology experiments) Soil-groundwater continuum contaminated by PFAS plume
Most important ecosystem services	• Regulation of water quality • Biodiversity conservation • Non-food agriculture (forestry)	(Laboratory used to conduct pollution experiments)
Target soil functions	• Filtering, buffering and degradation of pollutants • Habitat for soil organisms • Water retention, circulation and infiltration and groundwater recharge • Storage, recycling and transformation of organic matter • Retention and supply of nutrient	(Laboratory used to conduct pollution experiments) • Filtering, buffering and degradation of pollutants • Water retention, circulation and infiltration and groundwater recharge • Storage, recycling and transformation of organic matter • Retention and supply of nutrient
Key drivers to consider	• Climate change	• Climate change
Measures to analyse	• Remediation technologies (NNF) • Vegetation growth • Bare soil	• Bare soil • Vegetation growth • Remediation technologies (NNF)
Model trains proposed	MODFLOW + Hydrus-1D + PHREEQC	MODFLOW + Hydrus-1D + PHREEQC

Table 3.3 Test cases focusing on urban soils

	Rotterdam
Main pollution challenge	Public health risk due to pollution (in water and soil) caused by increased infiltration (rainwater infiltrates instead of going into sewage system)
Most important ecosystem services	• Regulation of water flows and stocks • Biodiversity conservation • Support for urbanization • Regulation of erosion and landslides • Regulation of risks for health and ecosystems
Target soil functions	• Water retention, circulation and infiltration • Habitat for soil organisms • Filtering, buffering and degradation of pollutants • Stable physical support for infrastructure
Key drivers to consider	• Climate change
Measures to analyse	• Urban greening • Increase infiltration through drainage-infiltration-transport sewage system • Biodiversity conservation
Model trains proposed	MODFLOW6 + UZF6

4 The model trains

The model trains are cornerstones in the PHISHES methodology supported by the PDP. They consist of combinations of inter-linked existing and well proven mathematical simulation models dedicated to simulating impacts of soil functions on typical ecosystem services in one or more of the selected test cases.

All modelling systems for running a selected model train must be installed on the computer hosting the PDP. Links to installation guides will be included in the PDP as well as information on how to obtain a license to run them (licenses are required for some of the software packages).

4.1 Train 1, Hydrus-PHREEQC-MODFLOW (Örnsköldsvik, lab. soil tank)

The contamination test sites, Örnsköldsvik in Sweden and BRGM Test Facility - Soil Laboratory in France, focus on the PFAS fate and transport from soil to groundwater. Model train 1 has been selected for these two test cases and consists of a combination of the software packages HYDRUS-1D, PHREEQC, and MODFLOW-2005/MT3DMS. The train allows for simulation of water flow and reactive solute transport from field scale to watershed scale in the soil as well as unsaturated and saturated zones. For simulating geochemical reactions in the unsaturated zone, we will use the version of HYDRUS-1D coupled to the geochemical modelling program PHREEQC.

The HYDRUS-1D is a public domain Windows-based modelling environment for analysis of water flow, heat and solute transport in variably saturated porous media using a finite element resolution[3]. PHREEQC is a free, versatile geochemical modelling computer program developed by the United States Geological Survey (USGS). It is used to simulate chemical reactions and transport processes in natural or polluted water systems, laboratory experiments, or industrial processes. PHREEQC is based on equilibrium chemistry, but it can also model kinetic reactions and one-dimensional transport[4].This software will be used to develop model simulating transport of contaminant, soil functions and the impacts of the different measures or drivers on the soil functions.

The MODFLOW-2005/MT3MDS software will be used for simulation of the three-dimensional water flow and solute transport in ground- and surface water to assess the impacts of soil functions on related ecosystem services. MODFLOW-2005 simulates steady and unsteady flow in an irregularly shaped flow system in which aquifer layers can be confined, unconfined, or a combination of confined and unconfined[5]. Reactive pollutants are supported by MT3DMS (linked to MODFLOW) and by PHREEQC.

The data flows between the individual models are indicated in Figure 4.1 and will be established by data exchange scripts to be included in the data processing library of the PDP.

[3] https://www.pc-progress.com/en/Default.aspxhydrus-1d
[4] https://www.usgs.gov/software/phreeqc-version-3
[5] https://www.usgs.gov/software/modflow-2005-usgs-three-dimensional-finite-difference-ground-water-model

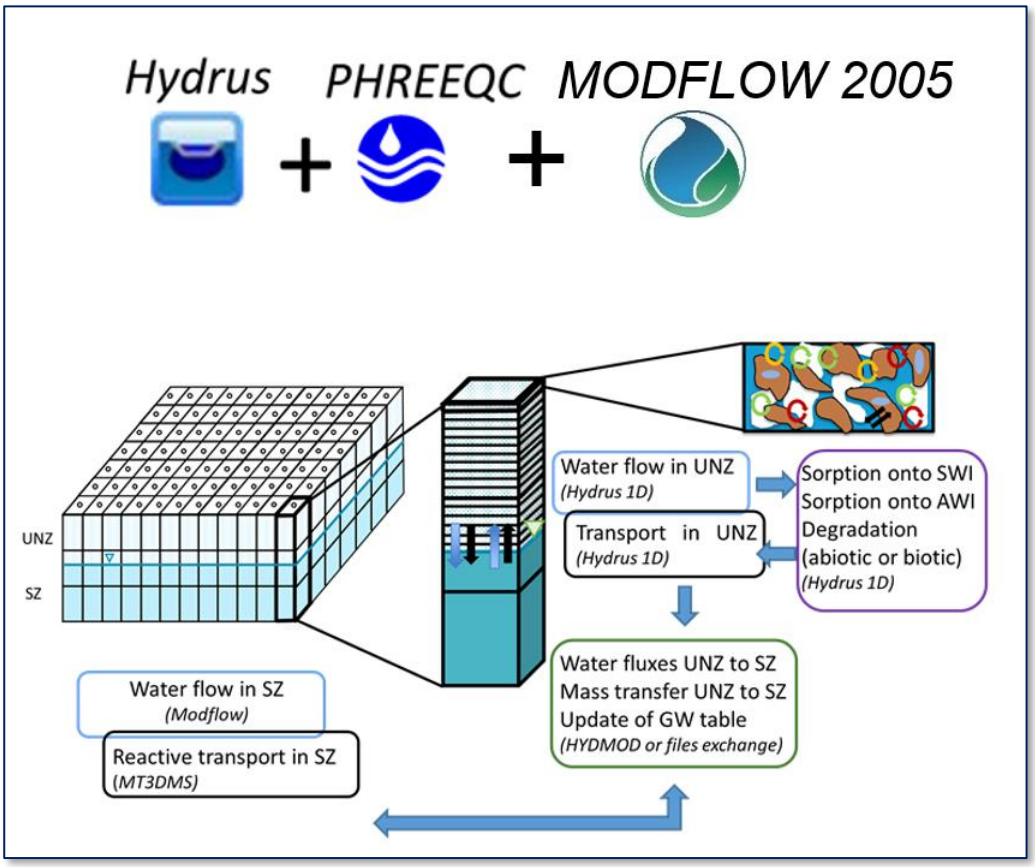

Figure 4.1 Train 1D Hydrus – PHREEQC – MODFLOW-2005 with inter-model data flows (SZ = saturated zone, UNZ = unsaturated zone, SWI = surface water interface, AWI = air water interface)

4.2 Train 2, Daisy–MIKE SHE ECO Lab (Zelivka, Koprzywianka)

The DAISY model is a mechanistic and detailed simulation model developed to simulate the physical and biological processes within an agricultural field. It focuses on the soil-plant atmosphere system, tracking the fate of water, energy, carbon, and other relevant components. The model is used to optimize crop yields, minimize nutrient losses, and assess the risk of environmental impacts like nitrate leaching[6].

In this train DAISY is used for detailed simulation of the effects of changed agricultural practices (tillage, fertilizing, crop rotation, etc.) on the soil health and the leakage of nutrients and pesticides.

The simulation will be made for selected representative soil profiles. To incorporate the interaction with groundwater and streams it will be linked with the groundwater component of the fully integrated hydrological modelling system MIKE SHE as illustrated in Figure 4.2.

MIKE SHE is a comprehensive hydrological modeling system that simulates the entire land phase of the hydrologic cycle. It integrates surface water and groundwater flow, allowing for detailed modeling of various hydrological processes like infiltration, evapotranspiration, and runoff. MIKE SHE is used for a wide range of applications, including water resource management and understanding the impacts of land use and climate change on water resources[7].

To enable the development of a climate-responsive vegetation growth module under PHISHES, MIKE SHE has been linked with the template-based general equation solver MIKE ECO Lab[8],

[6] https://daisy.ku.dk/technical-manual/
[7] https://www.dhigroup.com/technologies/mikepoweredbydhi/mike-she
[8] MIKE ECO Lab | Ecological Modelling Software

which is dedicated to incorporating ecological processes in aquatic simulation models and will be used to expand the present simulation of vegetation in MIKE SHE. Improved growth modules will be developed for various types of vegetation including conventional agricultural crops, forests, wetlands, scrublands, etc. This will improve MIKE SHE's present simulation of evapotranspiration, root water and nutrient uptake, add simulation of carbon sequestration and expand the plant growth simulation to more types of land use. Hence, the new vegetation module will supplement DAISY's simulations of agricultural fields with similar results for types of land use. The new module will be simpler and computationally faster than DAISY and is targeted for upscaling simulations to larger and more diverse catchments.

The data flows between the three models will take place during the simulation and are illustrated in Figure 4.2.

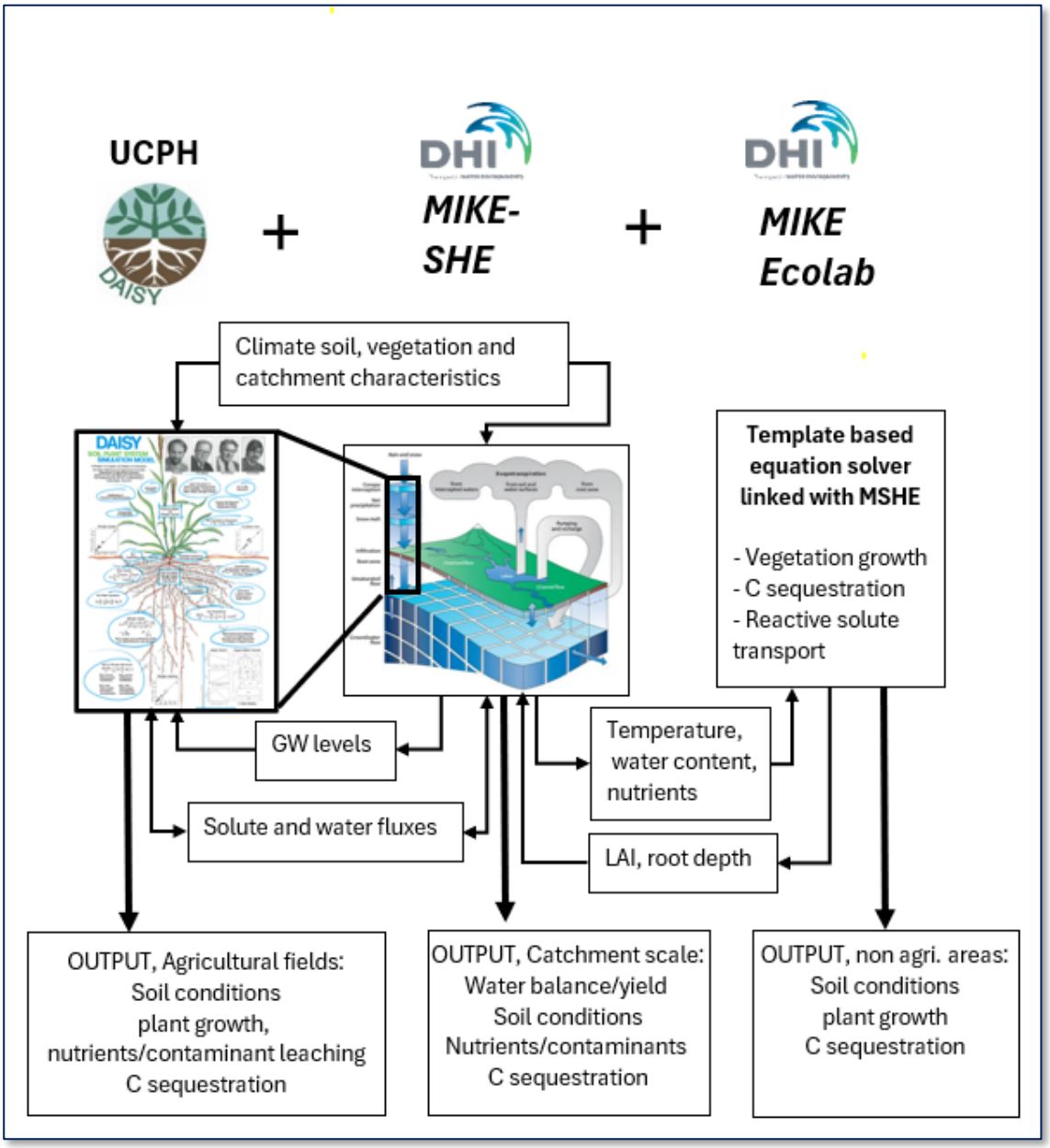

Figure 4.2 Model train 2 with inter-model data flows and train outputs (GW = groundwater, LAI = leaf area index)

4.3 Train 3, MODFLOW6-UZF6 (Rotterdam)

For the Rotterdam case, a MODFLOW 6 based model framework is used. MODFLOW 6 is the most recent core version of the MODFLOW modelling framework, developed by the United States Geological Survey (USGS).

For the unsaturated zone, the UZF-package is used, which is a 1D Richards based model taking only the gravitational potential into account. For the saturated zone, MODFLOW 6 solves Darcy's equation for saturated and constant density groundwater flow. To deal with the wetting and drying of cells in the phreatic aquifer, the Newton formulation is used.

For surface water ponding, a reservoir model was developed by Deltares. This model keeps track of the water balance at the subsurface and relates this to a ponding water level and inundation zones. This is coupled online to the Unsaturated Zone Flow (UZF) package via the available API. The coupling concept is shown in Figure 4.3.

During this project the workflow will be extended with the groundwater solute transport functionalities of MODFLOW 6. This deals with both advection and diffusive transport in the unsaturated (1D) and saturated zone (3D).

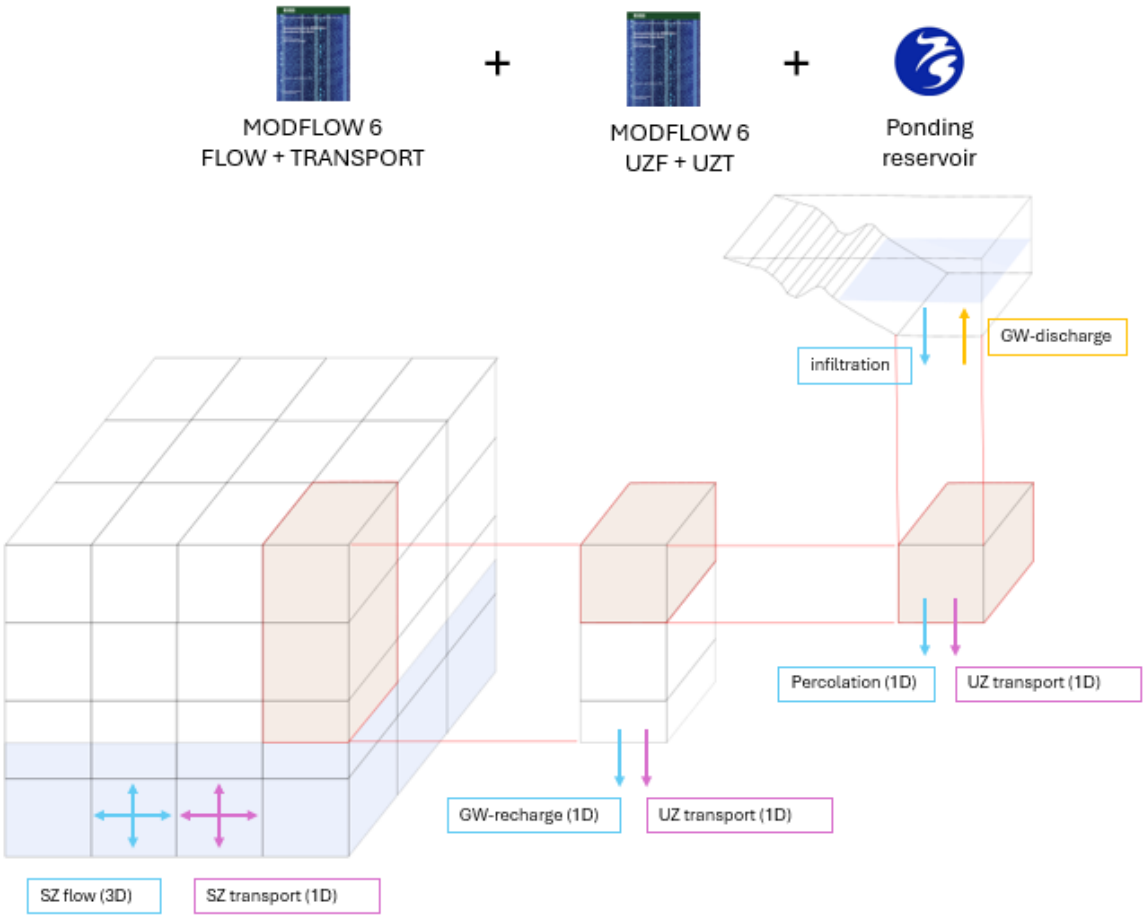

Figure 4.3 *Model train 3, coupling with inter-model data flows between MF6 saturated and unsaturated zone + external coupling to ponding reservoir model. With UZ and SZ for unsaturated zone flow and transport respectively, UZF and UZT for unsaturated zone flow and transport package respectively and GW for groundwater flow.*

5 Data needs and data structure

5.1 Introduction

The PDP is intended to be applicable both for the specific case studies analysed as part of the PHISHES project and for other potential European project sites.

In Europe, local environmental data are normally found to be at a high-quality level and therefore (when available) assessed to provide the best model input. However, new earth observations (E.O.) based data streams may supplement traditional local data and provide alternative channels, particularly for use in calibration of certain elements of model trains such as actual evapotranspiration from natural or forest vegetation.

The PHISHES test cases are based on models developed prior to the start of the project. As local data are to a large extent already incorporated in these models, the PDP data repositories will contain these data in their model specific formats. Such formats are described in the specific model manuals and will not be further described in this report.

In general, the PDP will not include download and processing facilities for specific local or national databases. However, to the extent that local data have been downloaded from national databases in the case studies and tested programmes exist for such downloads and conversions to the specific model formats, they may be incorporated in the PDP Script Library for use at other locations in the same country.

To ease the establishment of the selected model trains for other site studies, the PDP will have facilities for downloads and storage from some of the generally available European and global databases. Where such data are voluminous, scripts for download to the user's computer will be availed while less voluminous data, such as soil characteristic, land use types, etc., will be stored in the PDP repository as GIS files (shape or Geo tiff formats) for easy processing.

An inventory of the data needs of the individual models is included in Appendix B.

All the models involved in PHISHES focus on environmental and hydrological aspects, and large parts of their data needs are therefore shared between the models.

Table 5.1 describes the various data types dealt with in the PDP and possible alternative data sources for consideration if local data are not available.

5.2 Inventory of data sources relevant to the PDP

The case study models are being largely based on data from local institutions (meteorological services, geological services, agricultural institutions and water resources authorities).

Furthermore, an inventory has been made of European alternatives to such local data. The results of this work are listed in Table 5.1. The local data applied in the various detailed model setups will be reported under the various test cases.

Table 5.1 Data types supported by the PDP with alternatives to local data

Type of data	Required data format	Alternative if local data are not available	How the alternative data may be included in PDP
Climate			
Precipitation	Gridded file (NetCDF or dfs2)	Satellite-based observation or re-analysed weather model results, GPM, ERA5, gridded local data from local meteorological institutes	Scripts for download will be included
Reference evapotranspiration and "crop" coefficients	Gridded file (NetCDF or dfs2)	Penmann or Preistly Tailor estimate from weather model output	Scripts for calculation will be included
Temperature Air	Gridded file (NetCDF or dfs2)	Re-analysed weather model results, ERA5, gridded local data from local meteorological institutes	Scripts for download will be included
Soil temperature	Gridded file (NetCDF of dfs2)	Calculation from air temperature	Scripts for calculation will be included
Climate change time series	Gridded file (NetCDF of dfs2)	Developed downscaling tools from Global Climate Models (WP1)	Links to downscaling tools will be included. Scripts for alternative downscaling approaches will be included.
Catchment			
Digital elevation model	Gridded file (GeoTIFF)	Copernicus Digital Elevation Model (ESRIN,10m res)[9]	Scripts for download of a local area will be included
Surface drainage network	Vector format (Geopackage [.gpkg] or Geodatabase [.gdb])	EU-Hydro River Network Database 2006-2012, Europe[10]	Scripts for download will be included
River cross section and hydraulic characteristics	Tables	Simplified cross sections elevated according to Digital Elevation Model	To be included through the GUIs of the applied models

[9] DEM: https://sentinels.copernicus.eu/-/copernicus-digital-elevation-model-dem-now-available-via-copernicus-data-space-ecosystem

[10] River net work: https://land.copernicus.eu/en/products/eu-hydro/eu-hydro-river-network-database#download

Type of data	Required data format	Alternative if local data are not available	How the alternative data may be included in PDP
River structures (ponds, reservoirs, weirs) location and characteristics	CSV	Assessed from E.O., field inspections	To be included through the GUIs of the applied models
Soils			
Spatial soil profile distribution (soil map)	Shapefile of soil profile + interpolated gridded layers	LUCAS	Unless restricted, the data will be included in the data repository, in GIS formats. Alternatively, links to the source will be provided
Soil types at the various soil calculation layers	Tables	European Soil Database v2 Raster Library 1kmx1km	As above.
Soil characteristics texture, bulk density, plant water holding capacity, etc.	**Shapes, Grids**	ESDAC/LUCAS	As above
Depth to impermeable layer		ESDAC	As above
For each soil type:			
Soil water retention	Tables	UNSODA or Van Genuchten or ESDAC/HYPRES	Through links to source databases and processing scripts.
Hydraulic conductivity table	Tables	UnSODA or Van Genuchten or ESDAC/HYPRES	As above
Depth of drainage pipes (if any)	Parameter	National information	No tools for inclusion will be provided.
Hydrogeology			
For each calculation column: Thickness of the various hydrogeological formations	Shapefiles	National databases	As above
For each formation:			
Water storage capacity	Shapefiles	National databases	As above
Saturated hydraulic conductivity	Shapefiles	National databases	As above
Land use and vegetation			
Map of land use	Vector or gridded (.gpkg/.gdb/GeoTiff)	CORINE (100m); ESA World Cover (10m)	Scripts for download will be included
Type of vegetation or crop rotation	NetCDF/	ESA, crop types	Scripts for download will be included
For each vegetation type:			
LAI typical variation	netCDF of def2	Literature	No tools for inclusion will be provided
Root depth typical variation	netCDF of def2	Literature	As above
Root distribution (parameter)	netCDF of def2	Literature	As above
Initial conditions			
Groundwater levels	netCDF (or dfs2)		As above

Type of data	Required data format	Alternative if local data are not available	How the alternative data may be included in PDP
Initial flows and water levels river boundaries			As above
Calibration data			
Flow records	Time series	Links to European databases with streamflow information, e.g. GRDC[11]	Links to external databases will be provided
Water levels in rivers (if available)	Time series (.CSV)	E.O. analyses, hydroweb[12]	Scripts for download may be provided
LAI (optional)	NetCDF, GeoTiff or dfs2	ESA FAPAR[13]	Scripts for download may be provided
Obs Actual evapotranspiration (optional)	NetCDF, GeoTiff or dfs2	ET4FAO[14], LSA SFA[15], MODIS ET Product[16]	Scripts for download may be provided
Soil moisture obs (optional)	NetCDF, GeoTiff, HDF5 or dfs2	ESA CLMS[17] Soil Moisture, SMAP[18], SMOS[19]	Scripts for download may be provided
Ground water levels (if available)	netCDF or dfs2	National database, GSEU	No tools will be provided

[11] Flow Records: https://portal.grdc.bafg.de/applications/public.html?publicuser=PublicUser#dataDownload/Stations
[12] W.L: https://www.theia-land.fr/en/hydroweb/
[13] LAI: https://land.copernicus.eu/en/products/vegetation?tab=technical_summary
[14] ET4FAO: https://www.esa-sen4et.org/project
[15] LSA SFA: https://lsa-saf.eumetsat.int/en/data/products/evapotranspiration-and-turbulent-fluxes/
[16] MODIS: https://modis.gsfc.nasa.gov/data/dataprod/mod16.php https://www.esa-sen4et.org/project
[17] Soil moisture: https://land.copernicus.eu/en/products/soil-moisture/daily-surface-soil-moisture-v1.0
[18] Soil moisture: https://nsidc.org/data/spl3sma/versions/3
[19] Soil moisture: https://earth.esa.int/eogateway/missions/smos/data

6 Outline of IT architecture and user interface

This chapter presents a high-level summary of the system architecture. For technical specification consult Appendix A.

The platform architecture is composed of four primary components that work together to streamline user interaction, data access, and model preparation:

1. **Centralized Data Repository (Blob Storage[20])**
   A centralized, persistent storage system that hosts all essential datasets needed throughout the project lifecycle. This includes large and historically deep datasets with European coverage that are otherwise cumbersome for users to download, e.g. ERA5 data (precipitation, temperature, wind, humidity, etc.). In addition, the repository will hold relevant GIS data such as gridded soil data from the ESDAC and LUCAS databases, land use data, etc., to the extent that the use of such data are not restricted.

2. **Local Data repository**
   A repository holding local data that has been downloaded to the users' premises.
   In the case of PHISHES, local repositories will be established for each case study at the partners premises that will all be copied to protected section of the central project server.
   The local repositories will hold all input data to the model scenarios along with the scenario results. The data will be organized in folder structures and formats making the input readable by the model trains and the scenario results available for the evaluation tools to be developed during the project.

3. **Python Repository (STAC[21] to Model)**
   A specialized codebase designed to automate the retrieval and preparation of datasets for modeling purposes. It connects both to the above centralized and local data repositories and to external data sources (holding other or more recently updated data than those in the Centralized Repository).
   Since external database formats and addresses may change, the efficacy of the scripts linking to external sources cannot be guaranteed beyond the project period.
   Scripts will be included for download of free data and model software to the user's computer and for evaluation and comparison of the scenario results.

4. **Web User Interface (Project Portal)**
   A browser-based interface that serves as the main entry point to the platform. It guides users through the platform's purpose, providing:

   - Information on the PHISHES project and its main results and findings.
   - Description of the data available in the Centralized Data Repository (Point 1 above) and GUI's[22] for downloading them (Figure 6.1).
   - Description of the processing facilities included in the Python Repository developed under PHISHES (Point 3 above) and how to use it in a project context.
   - Description of the directory structure provided by PHISHES and how to organize simulation scenarios to ease the post-processing of scenario results.

[20] Blob storage is a cloud storage solution for data, that does not conform to a specific structure or format. It is designed to store massive amounts of data and is used for applications that require scalable and cost-effective storage

[21] STAC, which stands for SpatioTemporal Asset Catalog, is a standard specification for describing and cataloging geospatial data. It provides a consistent way to organize and expose collections of geospatial data, like satellite imagery, point clouds, elevation data, and more.

[22] Graphical User Interfaces

- A scenario management section for organizing, describing, and illustrating results as well as for evaluation and extraction of conclusions of the PHISHES scenarios.

Screenshots from similar scenario management portals are shown in the following figures. It is emphasized that these figures are from other projects with different objectives and indicators than those used in PHISHES. Therefore, the nomenclature used in these projects may also be different. Nevertheless, the figures still represent examples of how bulky results from distributed numerical environmental models may be organized, evaluated and presented.

Figure 6.1 illustrates a data view and download selection screen from DHIs Flood and Drought Portal. With a polygon surrounding the focus area (in this case France) it is possible to browse through a number of free globally available data sets. In this example, ERA5 rainfall has been selected and overall information on the data set is shown. It is possible to browse through the gridded data and display the spatial distribution at any timestep and hereby initially evaluate data's relevance for driving hydrological models. It is possible to download the data within the polygon and to calculate aggregated timeseries for the polygon as a whole or for sub-polygons.

Figure 6.2 to Figure 6.5 illustrate options to manage, describe compare scenarios.

Figure 6.2 shows a list with main characteristics, which is useful to keep track of the analyses and to select scenarios for intercomparison. In this example, the list indicates the scenarios' names, dates of simulation, the models applied, the climate scenarios used to drive the scenario and the interventions (measures) implemented. The scenario list may also be used to select certain scenarios for intercomparison.

Figure 6.3 shows the information and findings from a particular scenario. In this case carbon emissions at different distances from the intervention is evaluated and benchmarked against alternative scenarios. In PHISHES scenarios be different degrees of forestation could be benchmarked against discharged nutrients amounts load to a reservoir. Typically, several evaluation indicators are applied and subjected to a multicriteria analysis to determine the best solution.

Figure 6.4 displays a list the key indicators used in the comparative evaluation of scenarios and Figure 6.5 shows simple bar charts of such performance indicators.

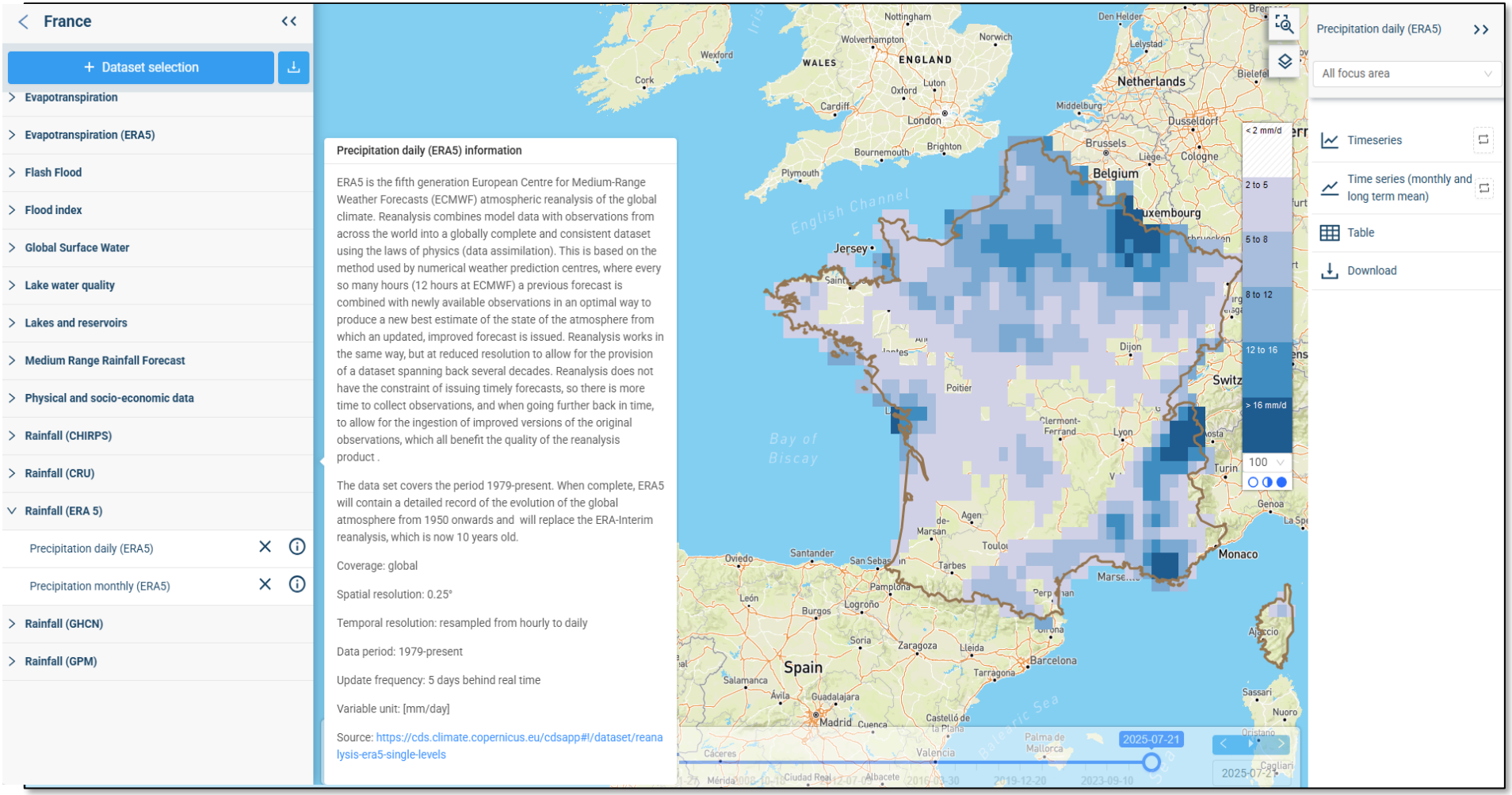

Figure 6.1 Selection and view of data for download from the Data repository (example from other existing application)

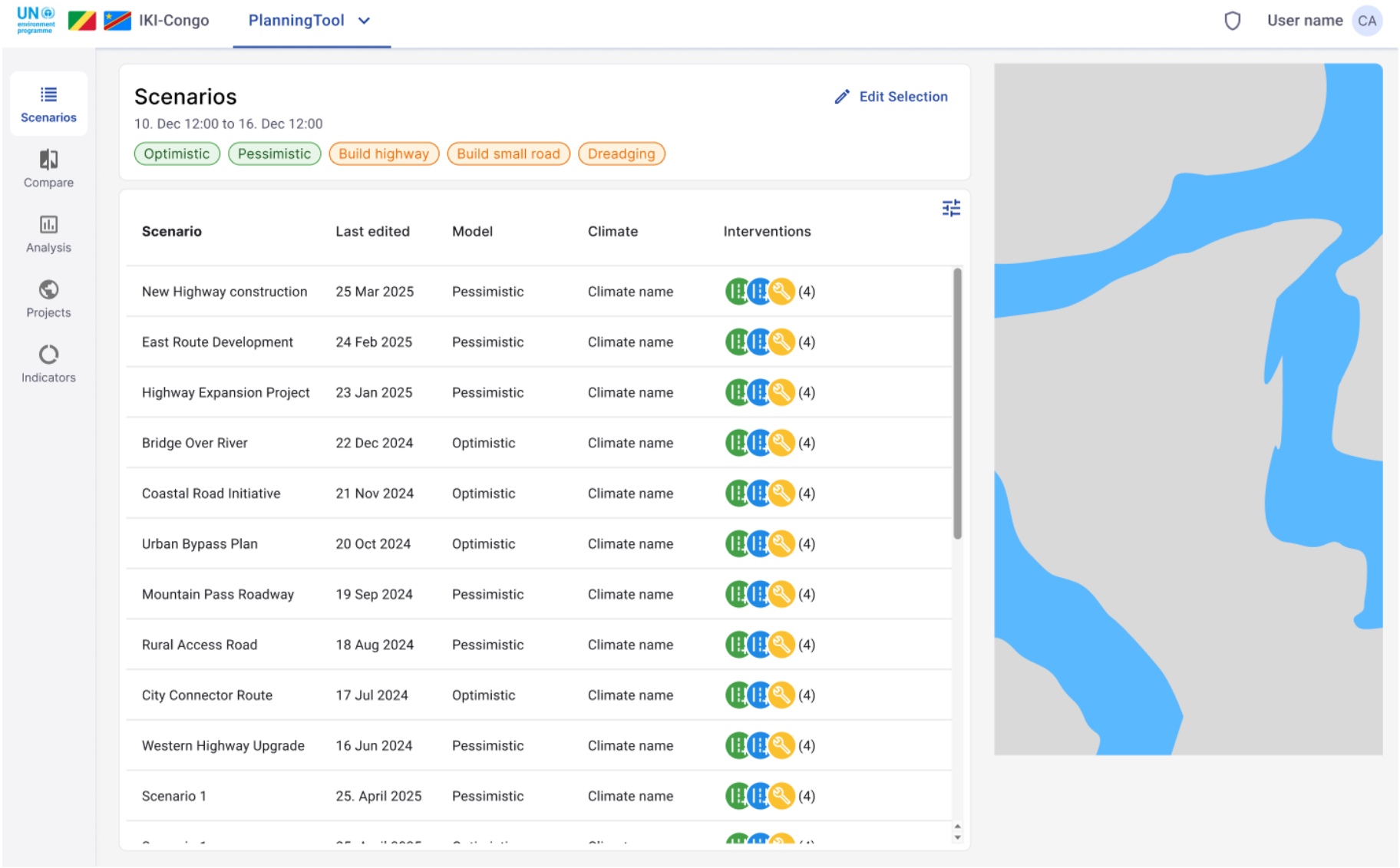

Figure 6.2 Menu for selection of scenarios for l or description (example from other existing application)

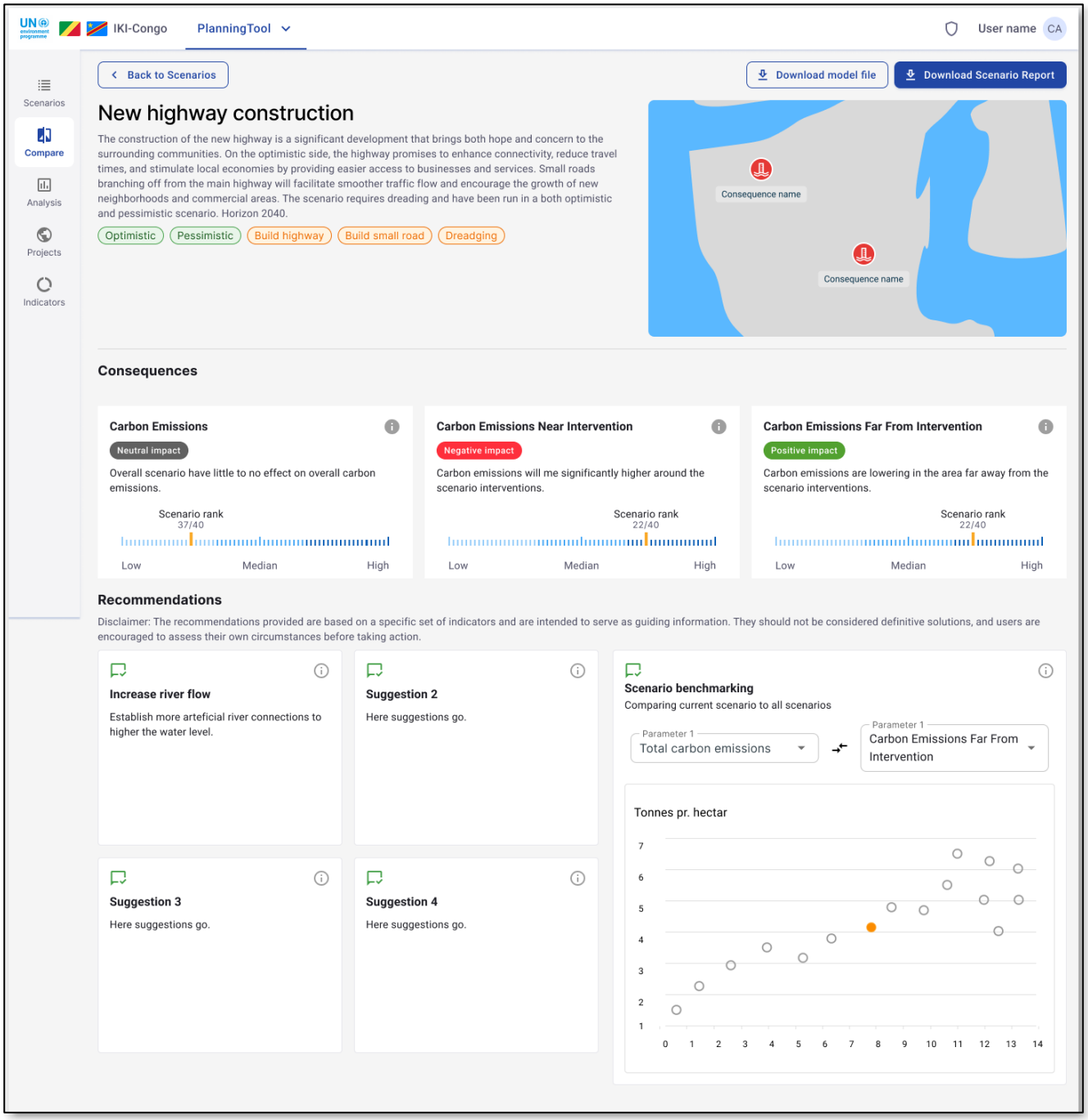

Figure 6.3 Menu for illustration of indicator results and comparison of key indicator results to those of other alternative scenarios (example from other existing application)

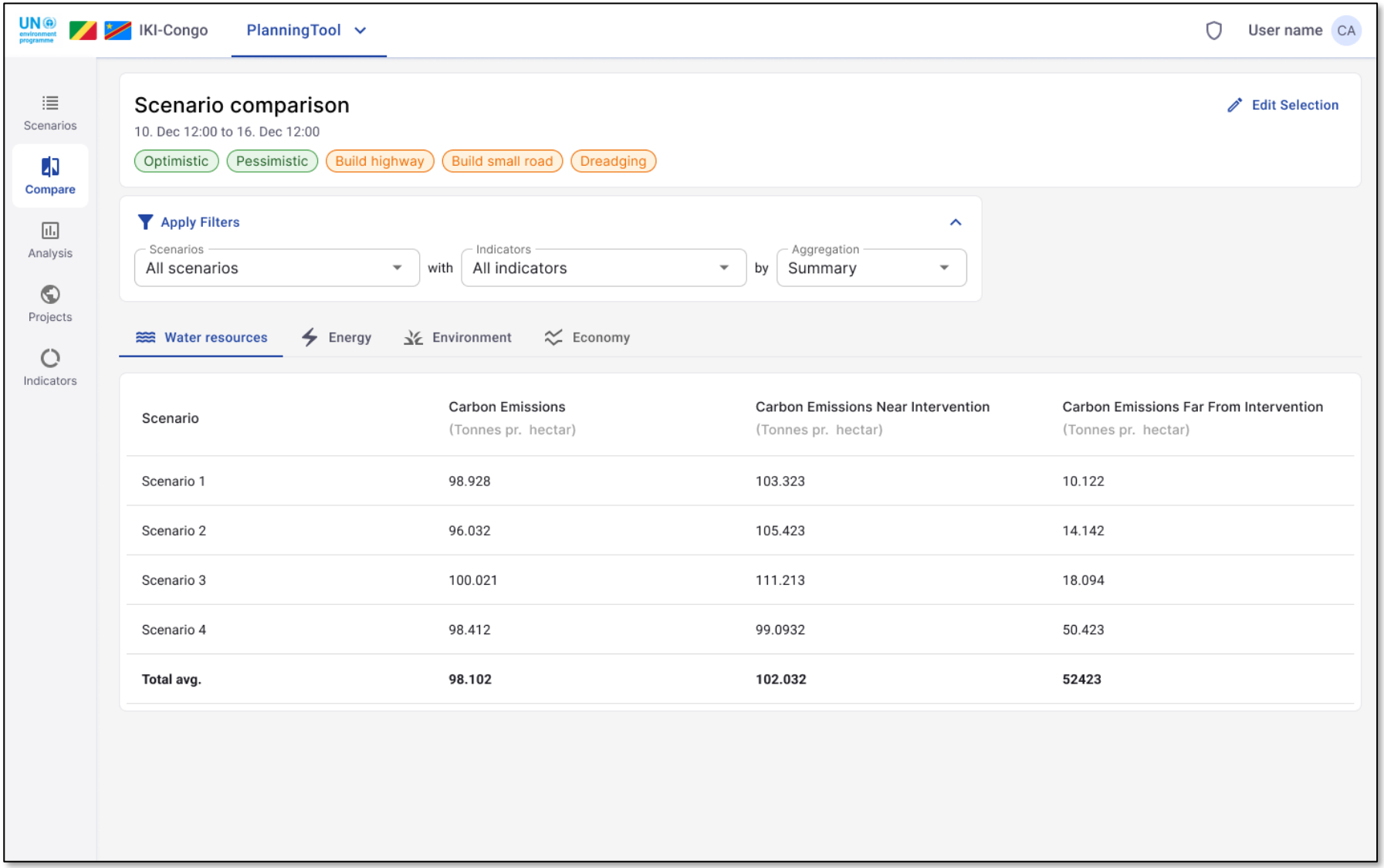

Figure 6.4 Multi criteria numerical comparison of selected indicators or groups of indicators (example from other existing application)

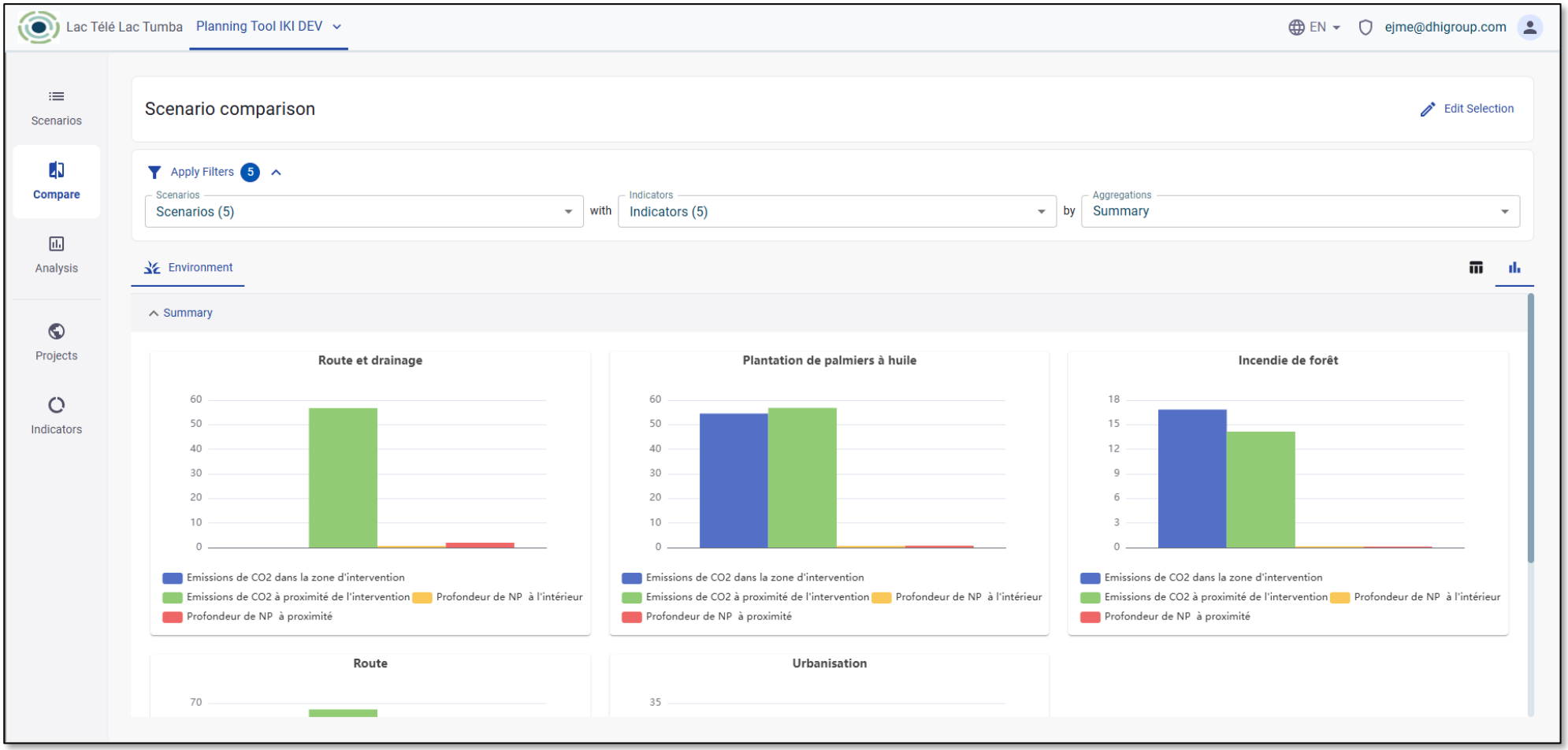

Figure 6.5 Graphical Multi criteria comparison of four scenarios (example from other existing application)

APPENDICES

Appendix A PDP System Architecture: Technical Specifications

The PHISHES Digital Platform (PDP) follows a hybrid architecture. Computationally intensive process modelling is performed on infrastructure operated by users and project partners. The hosted tier provides curated input data, controlled publication of derived indicators, interactive comparison of published scenarios, and decision-support functions.

The division of responsibility between the tiers is governed by one architectural rule:

The hosted tier performs no process simulation and does not orchestrate user-owned model execution. All process modelling remains on user or partner infrastructure. The hosted tier operates only on curated input data and published, validated indicators.

This separation allows licensed or open source and computationally demanding modelling systems to remain under institutional control while the published tier remains comparatively simple, low-operation, auditable, citable, and suitable for long-term preservation.

Appendix A.1 Architecture objectives and principles

The architecture is designed to satisfy the following objectives:

- support heterogeneous, user-licensed as well as open-source model trains without embedding those models in the PDP;
- provide consistent access to shared input datasets;
- standardise the preparation, description, validation, and publication of scenario indicators;
- prevent raw or confidential model inputs and outputs from being published unintentionally;
- provide reproducible comparison and decision-support functions over published results;
- minimise central operational dependencies and recurring hosting costs;
- retain a clear provenance chain from published indicators to their source datasets, model run, processing tools, and contract versions;
- preserve published states as immutable, citable snapshots; and
- allow the physical storage and query implementation to evolve without changing the scientific definitions exchanged between components.

The design applies the following principles.

Appendix A.1.1 User-owned simulation

Model execution, calibration, sensitivity analysis, and scenario production remain the responsibility of the participating organisation. The PDP does not distribute the model engines, provide model licences, schedule work on user infrastructure, or require a specific institutional scheduler.

Appendix A.1.2 Indicator-based publication

The public platform operates on derived indicators and supporting provenance rather than on native simulation output. Model grids, profiles, internal states, calibration files, and confidential source inputs remain on user premises unless separately approved for publication.

Appendix A.1.3 Clear rules for data sharing

The modelling and dissemination parts of the system follow the same agreed set of rules. These rules define which datasets, indicators and scenarios are used, how they are checked, and when results can be compared.

The two parts remain separate and only exchange data through clearly defined interfaces. This makes the system easier to maintain, update and integrate.

Appendix A.1.4 Controlled and immutable publication

Published content is created only by the publication pipeline. Each published snapshot is immutable and individually addressable. Corrections, additions, and supersessions create new publication records and new snapshots rather than modifying previously published content.

Appendix A.2 System context and boundaries

The PDP consists of eight platform components and integrates with user-supplied model trains and external services. The model trains are not PDP components; they are external systems operated and licensed by users or partners.

Appendix A.2.1 System context

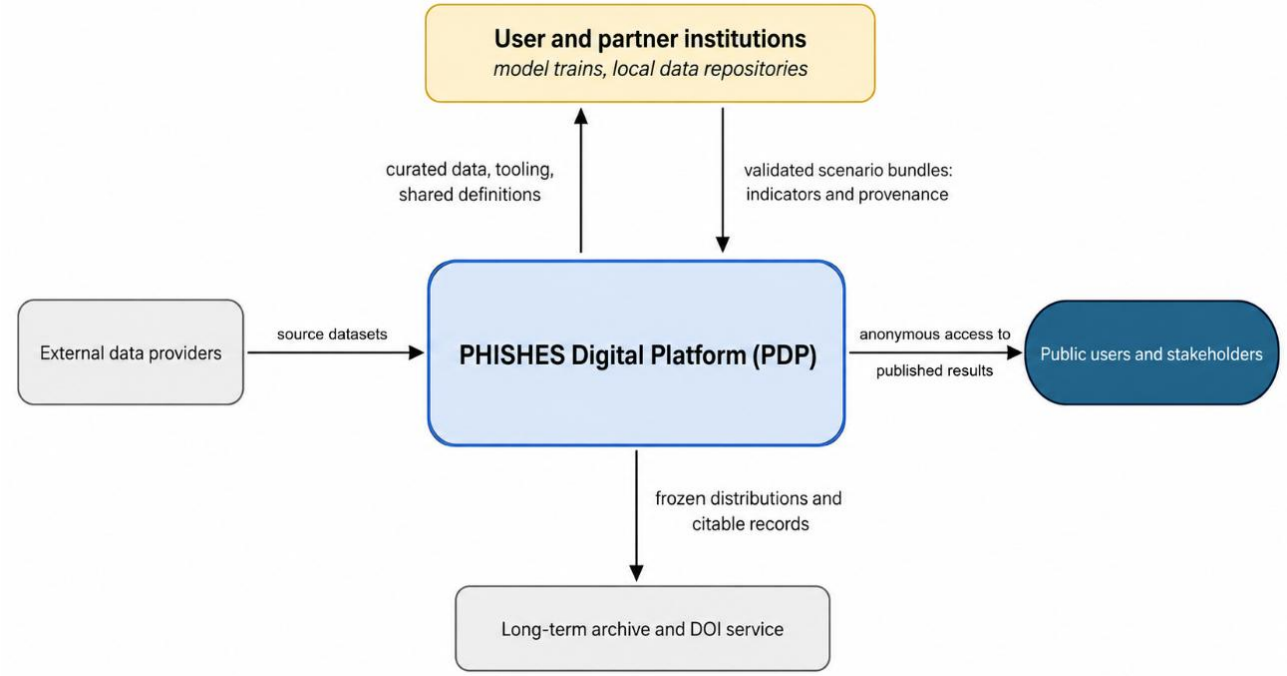

Figure 6.6 PDP context: platform components, user-supplied model trains, and external services.

Appendix A.2.2 Architectural boundary

The PDP boundary includes:

- the centralized repository of curated input datasets;
- the local data-repository conventions used by the PDP tooling;
- the dataset, indicator, and scenario contracts;
- the Python scripting library;
- the results publication pipeline;

- the published indicator store;
- the planning and comparison frontend; and
- the decision-analysis layer.

The following are outside the PDP boundary:

- user-licensed modelling software and model trains;
- institutional compute, storage, backup, and scheduling used to run the models;
- external source-data services;
- user-specific calibration and model-governance processes;
- general-purpose source-control, identity, cloud, domain, and archival services; and
- optional future assistant or surrogate-modelling services.

Appendix A.2.3 Responsibility boundary

The PDP standardises the interfaces immediately before and after model execution. It does not standardise the internal implementation of every model train.

The inbound boundary is the preparation of model-ready inputs from catalogued datasets. The outbound boundary is the production of contract-conformant indicators and a scenario manifest from model results. Model-specific adapters may be used at both boundaries, but the model engine itself remains external.

Appendix A.3 Logical architecture

The components are organised into five logical planes. The planes describe responsibility and lifecycle rather than physical network segments.

Table 6.1 Overview of the five logical planes and the PDP components assigned to each.

Plane	PDP components	Primary responsibility
Data	Centralized Data Repository; Local Data Repository; Published Indicator Store	Custody of shared inputs, local model data, and public derived results
Compute	External user-supplied model trains	Process simulation and scenario execution
Contract	Dataset, Indicator and Scenario Contract	Shared definitions, schemas, comparability rules, and validators
Tooling and publication	Python Scripting Library; Results Publication Pipeline	Input preparation, indicator generation, validation, review, and promotion
Presentation	Planning and Comparison Frontend; Decision Analysis Layer	Exploration, comparison, provenance inspection, and decision support

Appendix A.3.1 Data plane

Centralized Data Repository. Persistent hosted storage for curated datasets that are shared across the project and are legally permitted to be hosted. The repository may contain large, historically deep environmental and geospatial datasets that would otherwise be difficult for each user to acquire independently. Dataset identity, coverage, version, licence status, and access method are defined by the dataset catalogue.

Local Data Repository. A repository on user or partner premises containing the input data prepared for model execution, native model files, model results, intermediate products, and generated indicators. The repository follows documented conventions so that the PDP tooling can exchange data with the model trains without requiring the platform to control the model software. Local contents are not published automatically.

Published Indicator Store. Hosted storage containing validated indicator tables, scenario manifests, provenance records, map layers, supersession metadata, and other publication

artifacts. It contains no native simulation output unless a separate publication decision explicitly permits it. Content is organised into immutable, dated snapshots and is the authoritative data source for the public frontend.

Appendix A.3.2 Compute plane

User-supplied model trains. Model packages and workflows, including combinations of Hydrus, PHREEQC, MODFLOW, DAISY, MIKE SHE, MIKE ECO Lab, and other approved tools, execute on user or partner infrastructure. The user is responsible for software licences, model configuration, calibration, compute capacity, and institutional execution controls.

The PDP does not invoke these models remotely. Integration takes place through documented input and output contracts implemented by the local repository conventions and model-specific adapters.

Appendix A.3.3 Contract plane

Dataset, Indicator and Scenario Contract. A set of version-controlled, declarative definitions comprising at least:

- the dataset catalogue;
- the indicator catalogue;
- the scenario-manifest schema;
- publication and provenance schemas;
- comparability classifications;
- schema and contract validators; and
- compatibility and deprecation rules.

The contract defines the meaning of information exchanged by the platform. It does not prescribe the complete internal implementation of every component.

New instances of supported dataset and indicator types may be introduced through catalogue entries. New protocols, formats, scientific methods, spatial supports, or processing behaviours require a versioned extension to the tooling and, where necessary, to the contract schemas.

All of this controlled from the python package repository.

Appendix A.3.4 Tooling and publication plane

Python Scripting Library. A notebook-oriented package with an underlying programmatic interface. It provides two principal capabilities:

Data acquisition and preparation: discovery, authenticated retrieval, spatial and temporal subsetting, preparation, and conversion of catalogued data into model-ready local inputs.

Indicator processing and evaluation: transformation of model outputs into contract-conformant indicators, production of scenario manifests and provenance records, local validation, and local comparison of scenarios.

Detailed scientific methods, supported formats, package dependencies, and installation requirements are specified in Section Appendix A.7.

Results Publication Pipeline. The controlled mechanism by which an approved scenario bundle is transformed into public content. It performs central validation, invokes the required review and approval gates, builds query and map artifacts, creates an immutable snapshot, updates append-only publication metadata, and deploys the frontend against an explicitly selected snapshot.

Appendix A.3.5 Presentation plane

Planning and Comparison Frontend. A browser-based public interface providing documentation, test-site context, scenario selection, indicator comparison, spatial exploration, downloads, and provenance inspection. The initial implementation is a static deployment whose query operations execute in the browser against a published snapshot.

Decision Analysis Layer. A component that constructs and evaluates decision contexts from published indicators according to the methods and constraints defined in [Decision-Analysis Methodology, due end of 2027]. It is available in two execution contexts:

- a Python implementation for users analysing local or unpublished results; and
- a browser implementation for public analysis of published results.

A decision specification is represented as a deterministic JSON document. The browser may encode a compact specification in a shareable URL or allow the specification to be downloaded. A specification becomes centrally published and citable only when it is deposited through the controlled publication or archival workflow. The static frontend does not claim to provide an unrestricted general-purpose persistence service.

Appendix A.3.6 Component relationships

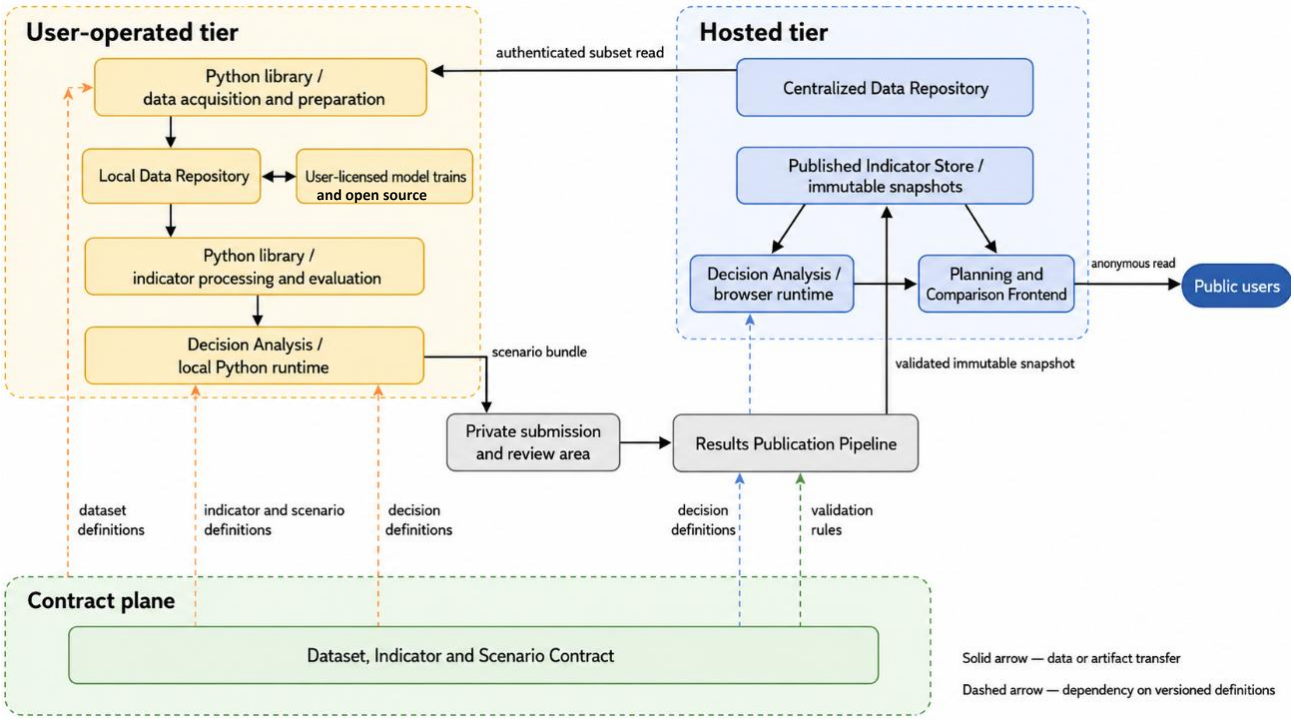

Figure 6.7 PDP component relationships across the data, compute, contract, tooling, and presentation planes.

Solid arrows represent data or artifact transfer. Dashed arrows represent dependency on versioned definitions.

Appendix A.4 Deployment architecture

The PDP follows a hybrid deployment model consisting of a hosted tier, a distributed user-operated tier, and external supporting services.

Appendix A.4.1 Hosted tier

The hosted tier provides storage, controlled publication, and public delivery.

- Backend elements of the hosted tier are as follows:

- the Centralized Data Repository;
- the contract and released tooling repositories;
- the submission and review area;
- the Results Publication Pipeline;
- the Published Indicator Store;
- the public frontend; and
- a limited compute allocation for workflow, conversion, publication, and integration testing.

The publication pipeline is the only element that executes on a recurring basis. It runs on submission and on approved merge, not continuously.

No central compute allocation is provided for scenario simulation, workflow testing or data processing. All model execution and all performance assessment take place on user or partner infrastructure, see Figure 6.8 below.

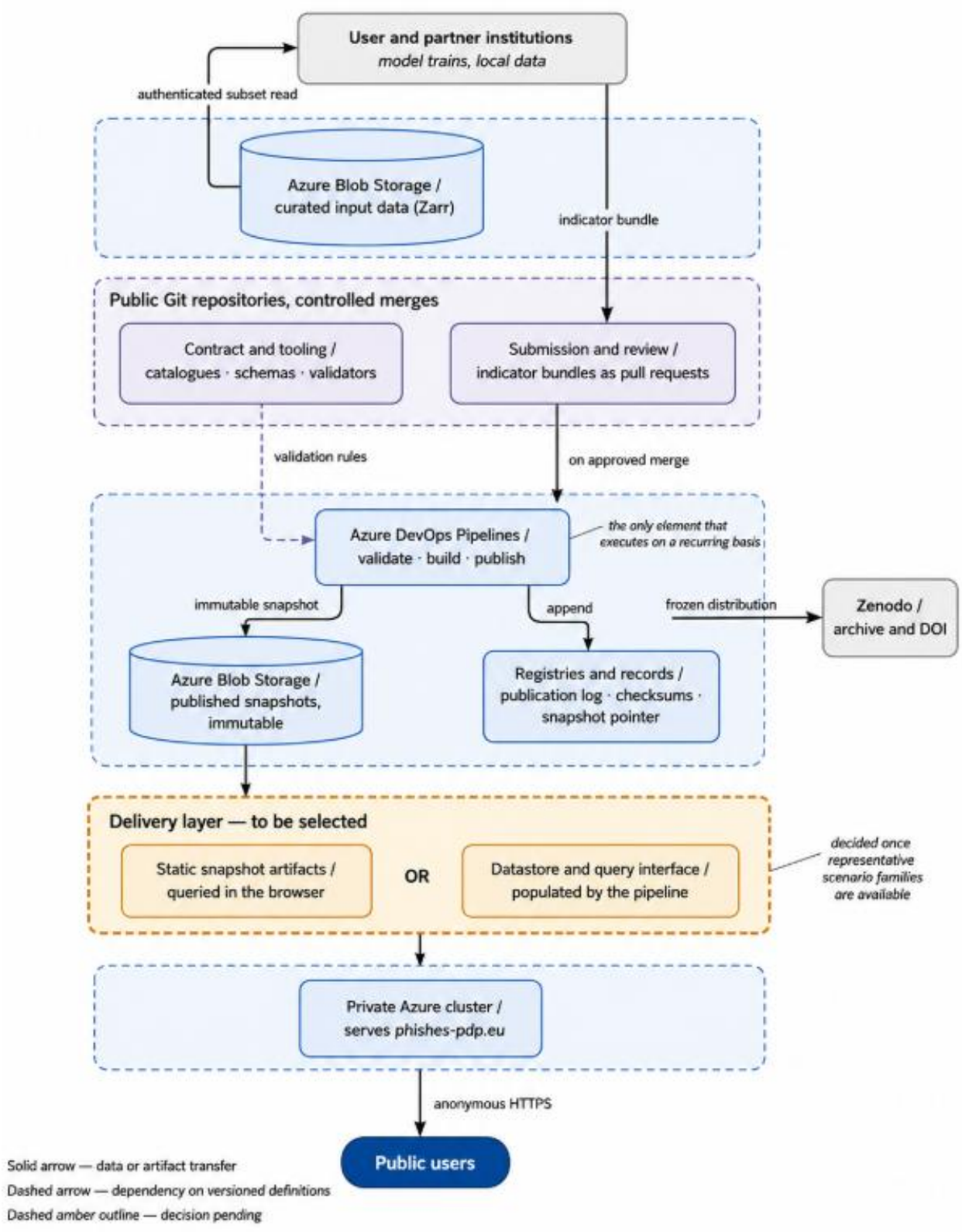

Figure 6.8 Backend elements and workflow of the PDP platform hosted tier.

The publication pipeline produces immutable snapshots in every case. That snapshot is the published record.

The mechanism by which the snapshot is delivered to the phishes-pdp.eu is designed once representative complete scenario families are available.

The elements listed above are the confirmed hosted components. Whatever delivery mechanism is selected is added to them rather than substituted for them and does not alter the snapshot as the archival and citable record.

Appendix A.4.2 User-operated tier

The user-operated tier contains:

- the Local Data Repository;
- the supported PDP Python environment;
- user-licensed and open-source model packages;
- model configurations and calibration material;
- native model input and output files;
- local indicator processing; and
- optional local decision analysis.

The PDP defines a supported execution profile covering operating systems, Python versions, required native dependencies, minimum resources, proxy behaviour, installation validation, and compatibility with supported model versions.

User-operated environments are outside the hosted production boundary. They are nevertheless required to comply with the supported execution profile when users expect PDP support or intend to submit scenario bundles for publication.

Appendix A.4.3 External supporting services

The architecture relies on external services for functions such as:

- source-data provision;
- source control and code review (GitHub);
- identity and access management;
- domain and certificate management;
- DOI assignment and long-term archival; and
- optional institutional high-performance computing.

These dependencies and their owners must be recorded in the operations and sustainability plan. The frozen distribution described in Section 3.12 must minimise reliance on external runtime services.

Appendix A.4.4 Containerisation and orchestration

Containerisation and a centrally managed orchestration layer are not required for the initial release. The user tooling is distributed as a reproducible, locked Python environment with installation and validation instructions.

Institutional schedulers may be used to run model trains, but the PDP does not require or control a particular scheduler. Containerisation may be introduced later as a supported packaging option if it improves portability without creating a dependency that prevents use of licensed desktop or institutional model software.

Appendix A.5 Interfaces and data flows

The principal interfaces are summarised below.

Table 6.2 Principal PDP interfaces, the information exchanged, and the applicable contract and access model.

Interface	Producer	Consumer	Information exchanged	Contract and validation	Access model
Curated dataset access	Centralized Data Repository	Python acquisition module	Requested spatial and temporal subsets plus dataset metadata	Dataset catalogue; checksum and coverage validation	Authenticated outbound HTTPS
External dataset access	External provider	Python acquisition module	Source files or service responses	Dataset catalogue and provider-specific adapter	Provider-specific authentication
Model-input hand-off	Python preparation module	Model train	Model-ready files and run metadata	Model-adapter specification	Local filesystem or institutional storage
Model-output hand-off	Model train	Python indicator module	Native model results and run metadata	Model-adapter specification	Local filesystem or institutional storage
Scenario submission	Python indicator module or authorised user	Private submission area	Manifest, indicators, provenance, validation report, and referenced artifacts	Scenario contract and local validation	Authenticated repository or submission identity
Central publication	Private submission area	Results Publication Pipeline	Approved scenario bundle	Pinned validators and review status	Pipeline service identity
Snapshot access	Published Indicator Store	Browser frontend and public clients	Indicator tables, manifests, map layers, and publication metadata	Published snapshot manifest	Anonymous HTTPS read
Archival deposit	Results Publication Pipeline	Archive or DOI service	Frozen distribution and publication metadata	Archive package manifest and checksums	Controlled archival identity

Appendix A.5.1 Dataset acquisition flow

Dataset access is initiated from the user-operated tier. No inbound connection to user infrastructure is required. Where supported by the source format, reads are spatially and temporally subsetted so that users do not need to transfer complete continental or historical datasets.

Dataset identity and provenance must survive subsetting and transformation. The local run record therefore references the exact source dataset version or immutable source identifier, the contract version, the acquisition time, and the transformation configuration.

Appendix A.5.2 Scenario-submission flow

A scenario submission contains the smallest package necessary to review and reproduce the indicator-processing step. It includes:

- a scenario manifest;
- indicator values or references to immutable indicator artifacts;
- provenance and software-version records;
- validation results;
- publication and confidentiality declarations; and
- checksums for all submitted artifacts.

Small, human-reviewable text artifacts may be stored directly in the private version-controlled submission repository. Larger generated tables, geospatial layers, or time-series artifacts may be stored in private immutable submission storage and referenced by checksum from the review record. The threshold between these mechanisms is defined through performance and repository-growth testing.

Appendix A.6 Environments and release boundaries

Appendix A.6.1 Development

Development consists of developer workstations, controlled branches, development storage, representative catalogue subsets, and limited test compute. Development data must not be treated as published content.

Appendix A.6.2 Integration and test

The integration environment validates end-to-end workflows against representative real datasets and test-site bundles. It includes a non-public staging snapshot and exercises:

- dataset access and conversion;
- scenario validation;
- publication builds;
- Python and browser decision-analysis equivalence;
- frontend compatibility; and
- performance against specified thresholds.

Appendix A.6.3 Hosted production

Hosted production contains the released Centralized Data Repository, immutable Published Indicator Store, selected public frontend build, and controlled publication pipeline. It contains no production-scale model simulation service.

Appendix A.6.4 User-operated execution environments

Each user institution operates its own model execution environment. Such an environment is not part of hosted production, but it becomes a supported PDP environment when it conforms to a documented tooling release and supported execution profile.

A submitted scenario manifest records the relevant tooling release and environment identifier. User institutions remain responsible for model licences, local backup, institutional availability, and the preservation of native model files.

Appendix A.6.5 Release model

Contract, tooling, frontend, and published data are versioned independently but linked through manifests and compatibility rules. A release must identify:

- the compatible contract version range;
- the tooling and frontend versions;
- dependency or environment lock identifiers;
- migration or deprecation requirements; and
- the snapshot against which the public frontend has been verified.

Appendix A.7 Python Repository Component

This repository holds all the data processing scripts produced under PHISHES. It is in the form of Python Jupyter notebooks, built on top of pre-existing Python libraries for interaction with the specific input and output data of the various modelling packages.

The following sections describe each of the technical tools developed for the PDP (purpose, processing steps, input/output overview, etc.).

The python repository is publicly available at: https://github.com/DHI/phishes-pdp/

Appendix A.7.1 Repository design

Scope and Boundaries

In scope

- Overall repository structure and navigation
- Governance and contributor-facing policy files
- Cross-module discoverability conventions
- Documentation layering across root and module documents

Out of scope

- Internal architecture of data-download-tool
- Internal architecture of model-trains, for e.g. MSHE-Ecolab (plant-growth-modul)e
- Detailed module workflows, APIs, and runtime behavior

Design Principles

1. Separation of concerns
   - Each module owns its own source code, dependencies, notebooks, and technical specs.
2. Stable top-level contract
   - Root-level files define governance, onboarding, and orientation for contributors and users.
3. Module autonomy
   - Modules can evolve independently as long as their published interfaces and docs remain coherent.
4. Documentation layering
   - Root docs explain repository composition; module docs explain implementation details.

Repository Responsibilities

The top-level repository is responsible for:

- Presenting a unified project overview and clear entry points
- Defining contribution, security, citation, and code-of-conduct policies
- Maintaining a consistent high-level structure for modules and shared assets
- Linking users to the correct module-specific setup and technical documentation

Documentation Contract

Repository-level documents should describe:

- What exists in the repository
- How the parts relate
- Where to find detailed documentation

Module-level documents should describe:

- How each module works internally
- Module-specific architecture, workflows, interfaces, and constraints

Top-Level Artifact Roles

- README.md: user-facing project overview and navigation
- phishes-pdp.md: agent/context-oriented high-level repository guidance
- REPOSITORY_DESIGN.md: repository design baseline and scope boundaries
- Policy files (SECURITY.md, CONTRIBUTING.md, CODE_OF_CONDUCT.md, LICENSE): governance and compliance context
- Module folders (data-download-tool/, plant-growth-module/): implementation ownership and module documentation

Acceptance Criteria

This repository design is considered effective when:

- Users can identify module entry points from root docs
- Contributors can find governance policies without module-level deep dives
- Repository-level docs avoid duplicating module internals
- Module documentation remains the authoritative source for implementation details

Appendix A.7.2 Data download tool

The PHISHES Data Download Tool is a Python toolkit for downloading and managing PDP datastore datasets by user-defined catchments. It connects to Azure Blob Storage, subsets datasets spatially (and temporally when applicable), and writes standardized outputs locally.

It produces local NetCDF, Zarr, or DFS2 outputs in a standardized folder structure and logs each download.

The PHISHES Data Download Tool works with the following inputs and capabilities for catchment-scale soil science modeling:

- Catchment spatial data (Shapefile format or extent) defining your study area boundary
- PDP datastore containing curated datasets in Zarr format
- Spatial sub setting to extract only data within your catchment
- Automated downloads with proper folder organization

And generates:

- Locally stored datasets clipped to your catchment extent

- Standardized folder structure for organized data management
- Download logs tracking provenance and metadata

The tool is publicly available at: https://github.com/DHI/phishes-pdp/tree/main/data-download-tool

Appendix A.7.3 Repository design

Subrepository Scope and Boundaries

In scope

- Catchment-based dataset download from Azure Blob Storage (Zarr source)
- Spatial and optional temporal subsetting
- Output writing to NetCDF, Zarr, or DFS2
- Download logging and minimal project folder scaffolding
- Post-download analysis helpers (weights, basin averages, anomalies, plotting)
- Notebook-first user workflow

Out of scope

- Parent platform orchestration and workflow scheduling
- Access management and secret rotation infrastructure
- End-user web UI and DSS integration
- Script download feature (documented requirement, not implemented)
- Complex project-run family folder orchestration

Design Goals

- Reproducible catchment downloads from a controlled dataset catalog
- Low-friction usage through Jupyter and Python API
- Storage efficiency via subsetting before local write
- Interoperable outputs for downstream analysis tools
- Deterministic filesystem contract for parent-repository consumers

System Context

Upstream dependencies

- Azure Blob Storage account/container with catalog-backed Zarr datasets
- User-provided catchment geometry (file, GeoDataFrame, or manual extent)

Downstream consumers

- Notebook users and analysis scripts
- Parent repository components that consume files under *data/* and logs under *logs/*

Architecture Overview

The repository follows a layered Python package structure:

- src/core: download pipeline, catalog, I/O, folder and utility primitives
- src/analysis: catchment QA/processing and analytics/visualization helpers
- notebooks: primary user-facing execution entry point

Core components

1. core/downloader.py
   - PDPDataDownloader is the central orchestration class
   - Manages catalog access, Azure connection, processing, writing, and logging
2. core/dataset_catalog.yaml
   - Canonical list of supported datasets and metadata (path, variable, CRS, temporal, DFS typing)
3. core/utils.py
   - Path construction, format-agnostic open, file lock-safe cleanup, variable inference
4. core/dfsio.py
   - Conversion and write bridge from xarray to DFS2 via mikeio
5. core/folder_structure.py
   - Creates minimal shared structure (logs/, readmes, metadata)

Analysis components

1. analysis/catchment.py
   - Load and validate catchment
   - AOI overlap and area constraints
   - Geometry buffering for points/lines
   - Reprojection and grid-cell intersection weights
2. analysis/timeseries.py
   - Area-weighted basin average and anomaly computation
3. analysis/visualization.py
   - Catchment, spatial map, and timeseries plotting helpers

Primary Runtime Flows

1. Single dataset download
   - Initialize PDPDataDownloader
   - Validate catchment and initialize Azure filesystem connection
   - Resolve dataset metadata from catalog
   - Open remote Zarr lazily
   - Spatial subset (bounds or catchment clip)
   - Optional temporal subset
   - Optional variable filtering
   - Write output (.nc, .zarr, .dfs2)
   - Append log entry to logs/download_log.json
2. Batch download

- Iterate all categories/subcategories from catalog
- Run single-download flow per dataset
- Continue on per-dataset failure with error print

3. Analysis flow (post-download)
   - Open local output (open_dataset_any)
   - Compute catchment weights (compute_catchment_weights)
   - Compute basin average and anomalies
   - Visualize with plotting helpers

Data and Storage Contracts

Input contracts

- Catchment: path, GeoDataFrame, or extent+CRS
- Dataset key: category/subcategory from YAML catalog
- Optional: time_range, variables, output format, masking/buffer options

Output contracts

- Dataset path pattern: data/<category>/<subcategory>/<subcategory>.<ext>
- Log file: logs/download_log.json
- Log record fields: timestamp, category, subcategory, description, output_path, bounds, time_range, catchment_shp

Catalog schema (current)

- path, display_name (optional in some entries), description
- variable, temporal, crs
- eumtype, eumunit (for DFS2)

Integration Contract with Parent Repository

This subrepository is designed to be consumed by parent-level orchestration through:

1. Python API contract
   - PDPDataDownloader for download orchestration
   - Analysis helper functions for optional post-processing
2. Filesystem contract
   - Stable output layout under data/
   - Stable run history under logs/download_log.json
3. Configuration contract
   - Parent workflows should pass catchment, output location, dataset keys, and time range
4. Catalog extension contract
   - New datasets are introduced by catalog entry additions, not downloader code changes

Security and Credential Handling

Current state

- Downloader has default Azure account/container values
- A default SAS credential is embedded as a class constant fallback

Design requirement for parent integration

- Logs should be considered operational metadata and may expose local paths

Reliability and Error Strategy

- Input validation uses explicit ValueError/FileNotFoundError
- Azure connection failures are wrapped into ConnectionError
- Spatial subset failures return full dataset with warnings
- Existing output deletion uses retry logic for Windows file-lock behavior
- Batch mode isolates per-dataset failures and continues processing

Performance Characteristics

- Remote Zarr access is lazy through xarray/adlfs
- Spatial slicing reduces transferred data for typical workflows
- Catchment masking and DFS2 conversion are more expensive paths
- Zarr output writes with v2 format and safe_chunks=False for compatibility

Appendix A.7.4 Pipeline

Objective / Rationale:

Once the user selects the datasets needed, they should be able to download them. The download tool should also allow download of any scripts. It should create a folder structure on the user's machine following a predetermined logic that fits the scripts that may need to be run, for one reason or another.

Processing Steps

1. Ingest
   - Inputs: catchment (shapefile, GeoDataFrame, or manual extent), dataset selection, output base, optional time range
   - Outputs: validated catchment geometry and user options
2. Quality checks
   - Inputs: catchment geometry
   - Outputs: validated geometry, rejected with error if AOI overlap or size limits fail
3. Catalog lookup and remote open
   - Inputs: selected dataset category/subcategory
   - Outputs: catalog metadata and remotely opened Azure Zarr dataset
4. Spatial subset / optional clip
   - Inputs: opened dataset and catchment bounds/geometry

- Outputs: spatially subset dataset (with optional catchment mask)

5. Temporal subset (optional)
   - Inputs: subset dataset and optional time range
   - Outputs: time-filtered dataset for temporal products
6. Format conversion
   - Inputs: xarray Dataset
   - Outputs: NetCDF (.nc), Zarr (directory), or DFS2 (.dfs2)
7. Write outputs and append log
   - Inputs: converted output and run metadata
   - Outputs: file written to data/// and entry appended to *logs/download_log.json*

Folder Structure

The downloader writes outputs using a dataset-first structure and logs each run in a shared log folder. Current structure:

```
project_root/
  data/
    <category>/
      <subcategory>/
        <subcategory>.<ext>
  logs/
    download_log.json
```

Implementation notes:

- Current code writes datasets to data/<category>/<subcategory>/<subcategory>.<ext> and logs to logs/download_log.json.
- Advanced project-run folder orchestration is out of scope for this tool.

Input/Output Schema

For each pipeline stage, specify file format, variable names, units, time resolution, and required metadata.

Catalog Entry (YAML)

```yaml
climate:
  era5_precipitation:
    path: climate/era5_precipitation/ERA5_precipitation.zarr
    display_name: ERA5 precipitation
    description: ERA5 precipitation
    variable: tp
    temporal: true
    crs: EPSG:4326
    eumtype: Rainfall
    eumunit: millimeter
```

Download Log Entry (JSON)

```
{
  "timestamp": "2026-02-24T10:15:30.123456",
  "category": "climate",
```

```
  "subcategory": "era5_precipitation",
  "description": "ERA5 precipitation",
  "output_path":
"C:/path/to/project/data/climate/era5_precipitation/era5_precipitation.nc
",
  "bounds": [7.1, 50.1, 8.4, 51.2],
  "time_range": ["2015-01-01", "2020-12-31"],
  "catchment_shp": "C:/path/to/catchment.shp"
}
```

Stage-Specific I/O Summary

- Ingest: Shapefile or GeoJSON; CRS metadata required; manual extent uses EPSG code
- Quality checks: Geometry and CRS; AOI overlap threshold; size limits in km2
- Spatial subset: Zarr dataset with lat/lon or x/y dims; CRS from catalog
- Temporal subset: Time dimension named time/date/t; ISO 8601 date strings
- Format conversion: NetCDF, Zarr (v2), DFS2; variable names from catalog

Dependencies

Required software libraries, scripts, and computing resources:

- Core: numpy, pandas, xarray, zarr, dask, netCDF4
- Geospatial: geopandas, shapely, rasterio, rioxarray, pyproj, fiona
- Azure: adlfs, fsspec, azure-storage-blob, azure-identity
- Optional: mikeio (DFS2), matplotlib/cartopy (visualization)
- Recommended system: >= 2 GB RAM, disk sized to dataset/time range

Validation / QC

Checks to ensure outputs are consistent, physically realistic, and compatible with downstream models:

- Catchment CRS is defined and AOI overlap is validated
- Catchment size is within min/max thresholds
- Dataset spatial dims are detected; clip is only applied to spatial variables
- Time slicing only applied when a time dimension exists

Error Handling

How missing data, corrupt files, or unexpected formats are handled:

- Missing catalog entry: ValueError with dataset not found
- Azure access errors: ConnectionError after connection test fails
- Spatial subset failure: fallback to full dataset with warning
- Output path locked: remove_path_with_retry retries with backoff

Expected failure modes and recovery strategies:

- SAS token expired -> retry with new token; alert user to update credentials
- No AOI overlap -> abort with validation error
- Empty spatial subset -> fallback to nearest-neighbor selection

Sample error messages and alert recipients:

- "Cannot connect to Azure storage" -> user notified in notebook output
- "AOI validation failed" -> user notified in notebook output

Performance Requirements

Time limits, computational constraints, scalability:

- Expected runtime depends on dataset size and network throughput
- Spatial subsetting minimizes data transfer where possible
- DFS2 conversion may be CPU-intensive for large time series

Representative data volume targets (:

- Small catchment (<= 10k grid cells): minutes per dataset
- Large catchment (>= 100k grid cells): tens of minutes per dataset

Internet speed assumptions for the estimates above:

- Fast connection (>= 100 Mbps): runtimes are typically near the lower end of the ranges
- Moderate connection (25-100 Mbps): use the listed ranges as-is
- Slow connection (< 25 Mbps): expect runtimes to increase substantially

Scaling strategy:

- Use smaller time windows for testing
- Avoid mask_on_catchment unless required for accuracy
- Prefer Zarr output when reusing in analysis notebooks

Appendix B The selected model trains

This appendix lists the specifics of the selected modelling trains in tabular forms.

Table A-1 Characteristics of model train 1 HYDRUS-1D – PHREEQC – MODFLOW-2005/MT3DMS

Name of tools	**Hydrus 1D – PHREEQC – MODFLOW-2005/MT3DMS**
Test case application	BRGM Test Facility and Örnsköldsvik airport site
Scale (field/catchment)	Field scale (1D, 2D) (+ Watershed scale, 3D)
Chemical compounds	• Carbon • Nitrogen • Phosphorus • Major chemical element (Ca, Mg, K, Na, Cl, SO4, HCO3, Si, etc.) • Minor chemical element (F, Fe, Mn, etc.) • Trace metals (As, Se, Zn, Cu, Cd, Pb, etc.) • Pesticides (zwiterrionic, non-polar, polar) • Pharmaceutical compounds (zwiterrionic, non-polar, polar) • Nanoparticles and colloids • Dissolved organic compounds • PFAS
Water flow processes	• Variability saturated flow (Hybrid form of the Richards equation) • Saturated flow (Darcy equation + continuum equation) • Van Genutchen and Brooks-Corey laws for variably saturated flow • Permeability laws • Dynamic boundary conditions (Dirichlet, Gauchy) => climate boundary conditions + groundwater table level boundary conditions + surface water flow • Double porosity/permeability flows (matrix and macropores) • Gaseous (CO_2, O_2, N_2) flow transport (diffusive)
Numerical development envisioned	• Numerical framework to parametrize HYDRUS-1D-PHREEQC software for simulating plant transpiration and passive/active plant uptake, carbon cycle based on the modelling scheme developed in Daisy, N nutrient cycle based on the modelling scheme developed in Daisy model and explicit simulations of PFAS sorption onto air-water interface in the unsaturated zone and transformation pathway from PFAS present in aqueous film-forming foam such as precursors to persistent PFAS compounds. • Numerical framework to link HYDRUS-1D software and MODFLOW-2005 software to exchange: i) data on groundwater table level and water flow from soil to groundwater and ii) data on mass fluxes from soil to groundwater • Numerical framework to test and ensure water mass balance calculated between models developed with HYDRUS-1D and MT3DMS software to exchange: i) solute concentration in groundwater and surface water and ii) mass fluxes from soil to groundwater • Updating the coupling scheme between Hydrus software and PEST++ software to estimate simulations uncertainties and improve model calibration • Numerical framework to extent model forecasts accounting for model uncertainties

Name of tools	Hydrus 1D – PHREEQC – MODFLOW-2005/MT3DMS
Test case application	BRGM Test Facility and Örnsköldsvik airport site
Previous or ongoing works that could be linked to PHISHES project scenarios simulated in previous projects	A first version of the coupling scheme between a simplified version of HYDRUS-1D and MODFLOW-2005/MT3DMS has already been made (Beegum et al., 2018, 2020). This version has been used in the PROMISCES project (Grant Agreement N°101036449) to simulate fate and transport of PFAS in the unsaturated-saturated continuum, but simplifications still remain concerning PFAS reactivity in the unsaturated zone.

Table A-2 Characteristics of model train 2a Daisy

Name of tools	Daisy - MIKE-SHE
Test case application	Zelivka, Koprzywianka
Scale (field/catchment)	Field scale (1D, 2D)
Chemical compounds	Carbon (CROP, SMB, SOM, AOM, and DOC), Nitrogen (NH4, NO3, N2O, DON), Pesticides (from PPDB), PFAS (under development), Nitrification inhibitors (under consideration)
Water flow processes	2 domain water flows (Richards equation, macropores), upper boundary defined by weather, lower boundary defined by groundwater or gravity flow (Dirichlet and Neumann). Transpiration governed by dynamic crop model (based on light, temperature, phenology), root density, and radial darcy flow towards roots. Above ground compartments: snow pack, leaves, litter, pond.
Numerical development envisioned	Further work on PFAS and AWI, start working on NI. The main development focus will likely be on the above ground processes, though, which tend to dominate the system. We will look especially at the role and fate of the litter layer, which is expected to change a lot with changes to agronomic practice. Coupling with the MIKE SHE model below the Root-zone with interchange of solutes and water contents and fluxes.
Previous or ongoing works that could be linked to PHISHES project scenarios simulated in previous projects	1) https://daisy.ku.dk/about-daisy/agroeco-hpc/; 2) Klimagødskning (N2O); 3) https://daisy.ku.dk/projects-and-collaboration/projects/gmsr/ (Conservation agriculture)

MIKE

Table A-3 Characteristics of model train 2b MIKE SHE – MIKE ECO Lab

Name of tools	**MIKE-SHE- ECO Lab**
Test case application	Zelivka, Koprzywianka
Scale (field/catchment)	Catchment to basin scale (1D, 2D, 3D)
Chemical compounds	Catchment scale (1D, 2D, 3D) advection dispersion approach for unlimited number of compounds
Water flow processes	Unsaturated zone options: 1D, finite difference multilayer method based on either Richards equation or gravity flow; two-layer root zone model for simple water balance OR Green and Ampt infiltration when capillarity can be ignored. Bypass flow for simple representation of macropores and sub-grid variability of infiltration. Overland flow: 2D diffusive wave method with two-way dynamic exchange between flood plains and river. Conceptual routing of subsurface drains to the river network. Full 1D hydrodynamic simulation of river channels and lakes, including operation of hydraulic structures. Faster and less data-intensive flow routing is also possible. Kristensen and Jensen vegetation-based actual evapotranspiration accounting for interception, soil water content, ponded water and groundwater. Snow is converted to surface water using an extended degree-day method.
Numerical development envisioned	Coupling with Daisy, further development of the ECO Lab coupling. Extensive developments of ECO Lab templates for growth of various types of vegetation linked with climate, nutrients and water uptake.
Previous or ongoing works that could be linked to PHISHES project scenarios simulated in previous projects	MIKE SHE models exist for two of the test cases. Pre-existing ECO Lab templates for nutrient cycling will be used in the development. However, constants would need updating for unsaturated zone (currently developed for saturated sediments). Links and coordination with developments under another project focusing on Global Wetlands takes place particularly for plant growths, emission and carbon sequestration modelling of wetlands.

Table A-4 Characteristics of model train 3 MODFLOW6-UZF6

Name of tools	**MODFLOW6-UZF6**
Test case application	Rotterdam
Scale (field/catchment)	Catchment scale, 1D for unsaturated and 3D for saturated zone flow and transport
Chemical compounds	Catchment scale (1D + 3D) advection dispersion approach for unlimited number of compounds
Water flow processes	Ponding reservoir that keeps water balance of surface system. No surface flow routing within the defined reservoirs. Infiltration and drainage zones are updated dynamically. Simplified 1D Richards formulation taking only the gravitational potential into account. Diffusive driven flow like capillary rise is not taken into account. Evapotranspiration is assumed to be removed instantaneously from storage. 3D Darcy based formulation for saturated flow. Dynamically drying and rewetting of model cells is done using the Newton formulation of MODFLOW 6.
Numerical development envisioned	Extending current workflow with MODFLOW 6 transport functionalities for advection and dispersion in saturated and unsaturated zone.
Previous or ongoing works that could be linked to PHISHES project scenarios simulated in previous projects	Workflow and coupling were validated using field test experiments. This validation is relevant for the validation of the extended transport functionalities.

Appendix C Data requirements for model trains

Appendix C.1 Introduction to data requirements

This appendix contains the data requirements for each of the identified model trains to be used in the PHISHES project.

The tables list the data required by the model train (column 1), the format of the data in the PDP data repository (column 2) and possible external data source/database (column 3) from where the particular data may be downloaded, should they not be available from local data bases. Some of these databases will be included directly in the PDP preprocessing data repository This goes for the soil data from ESDAC, digital elevation models and other less bulky data types. For others, scripts for download of geographical subsets of the data to the user's own premises will be included in the PDP script library or links will be provided to the database for manual access.

The data requirements are specified in tabular form - one table for each model train. A summary table is included in the main part of this report.

Table B-1 Data requirements for model train 1 Hydrus - PHREEQC - MODFLOW-2005 models

Specification of type of data	Format of data	Alternative if local data are not available
Climate:		
Precipitation	Gridded time series (netCDF)	
Temperature	Gridded time series (netCDF)	
Potential evapotranspiration	Gridded time series (netCDF)	
Actual evapotranspiration	(Transient boundary conditions in atmosphere)	
Climate change parameters describing the predicted change in the above climate series.	Gridded file (netCDF of dfs2)	Developed downscaling tools from Global Climate Models (WP1)
Data relative to simulated compounds:	**Applications**	
PFAS amount spilled	Amount and distribution netCDF, Shape	
Initial conditions		
Concentrations of simulation substances in each soil layer	CSV netCDF	
Water content in soil layers	CSV netCDF	
Temperature in soil layers	CSV netCDF	
Groundwater level	netCDF	
Concentrations of simulation substances in ground water	netCDF	
Regional temperature in groundwater	CSV	
Type of vegetation	Shapefile, grid	
Root distribution in soil layers	Tables	
Leaf Area Index	Grid	
Soils		
Texture	CSV netCDF	General DB UNSODA; LUCAS; ESDB
Dry bulk density	CSV	General DB UNSODA; LUCAS; ESDB

Specification of type of data	Format of data	Alternative if local data are not available
Organic matter content	CSV	General DB UNSODA; LUCAS; ESDB
Saturated soil water content	CSV	General DB
Residual soil water content	CSV	General DB
Saturated hydraulic conductivity	CSV	General DB
Tortuosity	CSV	General DB/Literature data
Water retension function	CSV	Pedo transfer functions
Hydraulic conductivity function	CSV	Pedo transfer functions
Transfer function between mobile and immobile fraction		Pedo transfer functions
Effective porosity		Literature data
Gas diffusion properties		Literature data
Biogeochemical processes		
Simple model:		
Sorption process parameters	CSV	Literature data /experimental results
Partition coefficient (Kd)	CSV	Literature data /experimental results
Freundlich equation	CSV	Literature data /experimental results
Langmuir equation	CSV	Literature data /experimental results
Instantaneous and kinetically controlled sorption reactions	CSV	Literature data /experimental results
Advanced model:		
Sorption process parameters:		Literature data /experimental results
Surface complexation reactions (thermodynamic and electrostatic parameters)	netCDF CSV	Literature data /experimental results
Parameters for ion-exchange (thermodynamic data and mineral surface properties)	netCDF CSV	Literature data /experimental results
Sorption reaction onto air-water interface (partition coefficient)	netCDF CSV	Literature data /experimental results

Specification of type of data	Format of data	Alternative if local data are not available
Parameter of Langmuir or Freundlich equation	NetCDF CSV	Literature data /experimental results
Degradation process:		Literature data /experimental results
Equilibrium redox reactions (thermodynamic data)	NetCDF CSV	Literature data /experimental results
Kinetically controlled redox reactions (kinetic rate)	NetCDF CSV	Literature data /experimental results
Thermo-kinetically controlled redox reactions (thermodynamic data and kinetic rates)	NetCDF CSV	Literature data /experimental results
Themo-kinetically controlled redox reactions (with explicit dependence on microbial biomass) (thermodynamic data and kinetic rates)	NetCDF CSV	Literature data /experimental results
Thermo-kinetically controlled redox reactions (with explicit dependence on microbial anabolism and growth) (thermodynamic data and kinetic rates)	NetCDF CSV	Literature data /experimental results
Data on precipitation/dissolution process:	NetCDF CSV	Literature data /experimental results
Equilibrium reactions (thermodynamic data)	NetCDF CSV	Literature data /experimental results
Kinetically controlled reaction (thermodynamic data, kinetic rate)	NetCDF CSV	Literature data /experimental results
Kinetically controlled reactions based on transition state theory (thermodynamic data and mineral surface properties)	NetCDF CSV	Literature data /experimental results
Kinetic rate	NetCDF CSV	Literature data /experimental results
Data on intra aqueous speciation process:	NetCDF CSV	Literature data /experimental results
Equilibrium reactions (thermodynamic data)	NetCDF CSV	Literature data /experimental results
Kinetically controlled reactions (kinetic rates)	NetCDF CSV	Literature data /experimental results
Thermo-kinetically reactions (thermodynamic data and kinetic rates)	NetCDF CSV	Literature data /experimental results

Table B-2 Data requirements for model train 2a the DAISY - MIKE-SHE model

Specification of type of data	Format of data	Alternative if local data are not available
Climate:	**Daily or hourly time series (CSV) for each soil column**	
Location and elevation of weather station	Shapefile	
Precipitation	Time series	Satellite or reanalysed weather models (ERA5) or national models
Reference evaporation (ET0)	Time series, CSV	Temp., humidity, wind, global radiation
Temperature	Time series	Satellite or reanalysed weather models (ERA5) or national models
Solar radiation	Time series	Global radiation (outer atm)
Wind speed (optional)	Time series	Satellite or reanalysed weather models (ERA5) or national models
Humidity (optional)	Time series	Satellite or reanalysed weather models (ERA5) or national models
Climate change parameters describing the predicted change in the above climate series	Gridded file (NetCDF of dfs2)	Developed downscaling tools from Global Climate Models (WP1)
Soil:		
Soil profile divided into horizons. For each:		
Texture	CSV, Shapefile	General DB UNSODA, LUCAS, ESDB
Humus	CSV, Shapefile	General DB UNSODA, LUCAS, ESDB
Dry bulk density	CSV, Shapefile	General DB UNSODA, LUCAS, ESDB
Existence and depth of macropores	CSV, Shapefile	General DB UNSODA, LUCAS, ESDB
Hydraulic retention	Tables	Pedo-transfer functions from above parameters
Hydraulic conductivity	Tables	Pedo-transfer functions from above parameters
Ground water level	Time series (CSV)	
Depth and distance of drain pipes (if existing)	CSV, Shapefile	
Dry and wet deposition of No3 and NH4	CSV, Shapefile	

Specification of type of data	Format of data	Alternative if local data are not available
Land Management:		
Crop rotation	Time series of crop types (CSV)	
Sowing	Date or condition	
Harvest	Date or condition	
Tillage	Table CSV	
Fertilizer type and amount (and dates)	Time series with individual columns, CSV	
Pesticide application type and amount	Time series with individual columns, CSV	
Irrigation application	Timeseries mm/day	
Calibration		
Crop yield	Time series (CSV)	
Drain flow	Time series (CSV)	

Table B-3 Data requirements for model train 2b MIKE SHE – MIKE ECO Lab

Specification of type of data	Format of data (gridded time series, single time series, classification list, xy table	Alternative if local data are not available
Climate		
Precipitation	Gridded file (netCDF or dfs2)	Satellite based observation or reanalyzed weather model results, GPM, ERA5, national Gridded Precipitation
Potential Evapotranspiration	Gridded file (netCDF or dfs2)	Penman or Priestley Tailor estimate from weather model output.
Temperature Air	Gridded file (netCDF or dfs2)	Reanalysed weather model results, GPM, ERA5, national Gridded Precipitation
Soil temperature	Gridded file (NetCDF of dfs2)	Formula from Tair
Climate change parameters describing the predicted change in the above climate series.	Gridded file (NetCDF of dfs2)	Developed downscaling tools from Global Climate Models (WP1)
Catchment		
Digital elevation model	Grid	
Surface drainage network	Shapefiles or similar	
River cross section and hydraulic characteristics	Tables	
River structures (ponds; reservoirs; weirs) location and characteristics.	CSV	
Soils		

Specification of type of data	Format of data (gridded time series, single time series, classification list, xy table	Alternative if local data are not available
Spatial soil profile distribution (soil map)	Gridded file (if spatially distributed)	ESDAC
Soiltypes at the various soil calculation layers	Tables	ESDAC, LUCAS (Texture), UNSODA, etc.
For each soil type:		
Soil water retention	Tables	Pedotranfer funcs (texture), UNSODA
Hydraulic conductivity table	Tables	Pedotranfer funcs (texture), UNSODA
Depth of drainage pipes (if any)	Grid	
Hydrogeology		
For each calculation column: levels of of the various hydrogeological formations.	Grids	
For each formation:		
Water storage capacity	Shapefiles	ESDAC
Saturated hydraulic conductivity	Shapefiles	ESDAC, Hyd. conductivity table (above)
Vegetation		
map of vegetation types	Shapefiles or gridded files	E.O., ESDAC, CORINE
Type of vegetation or crop rotation	netCDF or dfs2	E.O.
For each vegetation type:		

Specification of type of data	Format of data (gridded time series, single time series, classification list, xy table	Alternative if local data are not available
LAI variation	netCDF or dfs2	E.O.
Root depth variation	netCDF or dfs2	Literature
Root distribution (parameter)	netCDF or dfs2	Literature
Initial conditions		
Groundwater levels	netCDF or dfs2	
Initial flows and water levels at river boundaries.		
Calibration data		
Flow records	Time series	Backcalculated reservoir inflows. WMO records, E.O. Analyses.
Water levels in rivers (if available)	Time series	E.O. Analyses
LAI (optional)	netCDF or dfs2	E.O. Analyses
Obs Actual evapotranspiration (optional)	netCDF or dfs2	E.O. Analyses,
Soil moisture obs (optional)	netCDF or dfs2	E.O. Analyses

Table B-4 Data requirements for Model train 3 MODFLOW6-UZF6 (Rotterdam)

Specification of type of data	Format of data (gridded time series, single time series, classification list, xy table	Alternative if local data are not available
Climate		
Precipitation	Gridded file (netCDF or dfs2)	Satellite based observation or Reanalyzed weather model results, GPM, ERA5, National Gridded Precipitation
Potential Evapotranspiration	Gridded file (netCDF or dfs2)	Penman or Priestley Tailor estimate from weather model output
Temperature Air	Gridded file (netCDF or dfs2)	Reanalysed weather model results, GPM, ERA5, National Gridded Precipitation
Soil temperature	Gridded file (NetCDF of dfs2)	Formula from Tair
Climate change parameters describing the predicted change in the above climate series.	Gridded file (NetCDF of dfs2)	Developed downscaling tools from Global Climate Models (WP1)
Catchment		
Digital elevation model	Grid	
Surface drainage network	Shapefiles or similar	
River cross section and hydraulic characteristics	Tables	
River structures (ponds;reservoirs;weirs) lovation and characteristics.	CSV	
Soils		

Specification of type of data	Format of data (gridded time series, single time series, classification list, xy table	Alternative if local data are not available
Spatial soil profile distribution (soil map)	Gridded file (if spatially distributed)	ESDAC
Soiltypes at the various soil calculation layers	Tables	ESDAC, LUCAS (texture), UNSODA, etc.
For each soil type:		
Soil water retention	Tables	Pedotranfer funcs (texture), UNSODA
hydraulic conductivity table	Tables	Pedotranfer funcs (texture), UNSODA
Depth of drainage pipes (if any)	Grid	
Hydrogeology		
Dor each calculation column: levels of the various hydrogeological formations.	Grids	
For each formation:		
Water storage capacity	Shapefiles	ESDAC, retention
Saturated hydraulic conductivity	Shapefiles	ESDAC, Hyd. conductivity table (above)
Vegetation		
Map of vegetation types	Shapefiles or gridded files	E.O., ESDAC, CORINE
Type of vegetation or crop rotation	netCDF (or dfs2)	E.O.
For each vegetation type:		

Specification of type of data	Format of data (gridded time series, single time series, classification list, xy table	Alternative if local data are not available
LAI variation	netCDF (or dfs2)	E.O.
Root depth variation	netCDF (or dfs2)	Literature
Root distribution (parameter)	netCDF (or dfs2)	Literature
Initial conditions		
Groundwater levels	netCDF (or dfs2)	
Initial flows, water levels and river boundaries.		
Calibration data		
Flow records	Time series	Back calculated reservoir inflows. WMO records, E.O. Analyses.
Water levels in rivers (if available)	Time series	E.O. Analyses
LAI (optional)	netCDF or dfs2	E.O. Analyses
Obs Actual evapotranspiration (optional)	netCDF or dfs2	E.O. Analyses,
Soil moisture obs (optional)	netCDF or dfs2	E.O. Analyses

Appendix D Scenarios PHISHES (Milestone 3)

Milestone 3 of the PHISHES project encompasses a selection and definition of scenarios which will serve as the basis for the simulations for the test sites. The scenarios described in this milestone are preliminary. This means that the scenarios might change, based on modelling capabilities or changes in data availability.

In this milestone, scenarios are defined as a combination of measures and drivers. The workflow scheme pictured in Figure C-1 below gives an overview of the relation between the modelling and the scenarios. Measures consist of land use changes, restoration measures, and nature-based solutions that influence soil health and soil pollution or promote soil health and reduce soil pollution. The goal of the drivers is to set boundary conditions within which the measures are tested. Drivers consist of potential changes that will occur in the long-term, regardless of short-term human influence or immediate political decisions.

The starting point for the determination of relevant drivers and measures are the test sites. However, the drivers and measures should also be realistic for other sites that are not part of the PHISHES project. This means that the drivers and measures should not only be relevant for the test sites in this project but could also be translated to other regions in Europe.

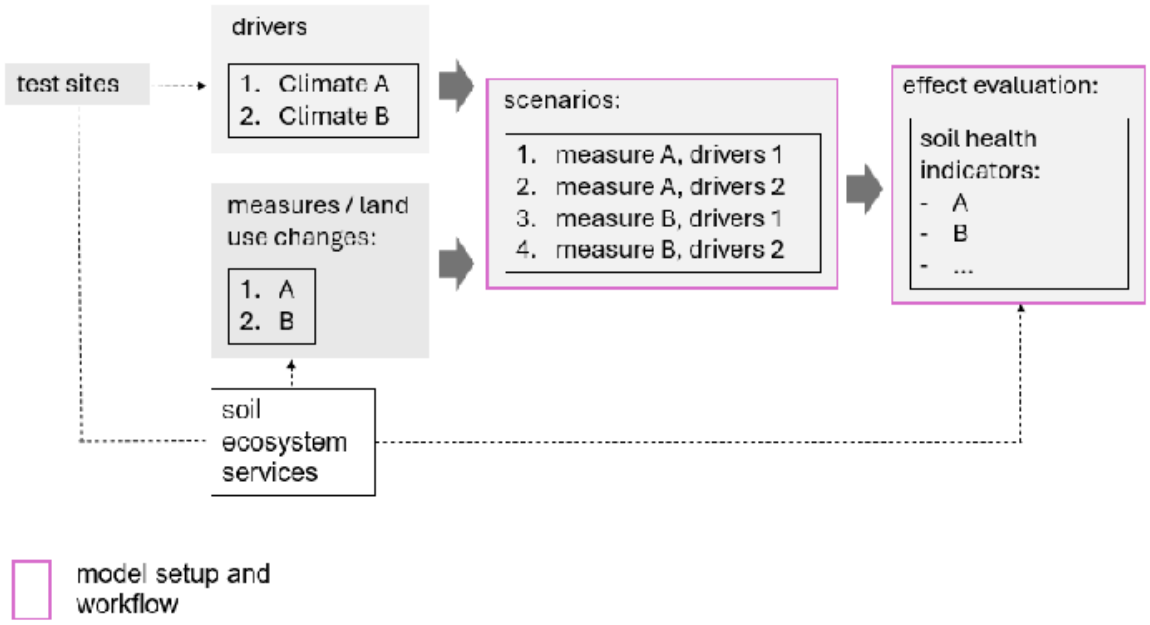

Figure C-1 Overview of the workflow scheme in the PHISHES project

Appendix D.1 Drivers

The identified drivers are the same for all test sites, although the exact implementation of the driver during modelling might differ between test sites. An overview of the drivers can be seen in Table C-1.

The first set of conditions that will be evaluated is the baseline. The baseline is representative of the current situation. This means that the climate and land use are defined by the situation right now. The goal of testing measures with this specific driver is to determine the short-term impact (<5 years) of the identified measures on soil health.

The other two identified drivers are related to climate scenarios. By using the climate scenarios as drivers, the long-term impact of measures on soil health can be evaluated. As the exact way the

climate will change is uncertain, climate scenarios that are on the "extreme" side of potential changes have been chosen. The SSP 1-2.6 (or RCP2.6) and SSP 5-8.5 (or RCP8.5) pathways, as defined by the IPCC (2023), will be used to capture the lower and higher GHG emission scenario for climate change.

Table C-1 Overview of drivers used in the scenarios.

Drivers
Baseline (current situation)
Climate scenario SSP1-2.6
Climate scenario SSP5-8.5

Appendix D.2 Measures

The measures in the scenarios are test site specific. This means that each test site will model the measures that are relevant to that specific test site. For comparable test sites, there will be overlap in measures. For example, many of the measures to be applied in the Zelivka and Koprzywianka test sites are the same. This way, the relevance of location for the success of certain measures can be determined. The exact way the measures will be modelled will be determined by the test site owners and modelers in the coming months.

An overview of the measures is given in Table C-2. In this table, the test sites are divided into the area types Agricultural, Industrial, and Urban, which indicate comparable test sites. Also, the most relevant pollutants per test site are listed. The identification of relevant pollutants aids in translating the eventual modelling results to more general conclusions.

In the accompanying excel file, the scenarios are indicated by an ‘x’ or ‘o’. The ‘x’ means that it is certain that the scenario will be modelled. An ‘o’ indicates that the choice to model the scenario will be based on the outcome of the scenarios indicated with an ‘x’.

Table C-2 Overview of measures used in the scenarios.

Measures					
Agricultural		**Industrial**			**Urban**
Zelivka	**Koprzywianka**	**Örnsköldsvik**	**BRGM**	**BRGM/SGI**	**Rotterdam**
Nitrogen, pesticides, PFAS, Nitrification Inhibitors (+ water balance and crop yield)	*Nutrients, pesticides, heavy metals*	*PFAS*	*PFAS*	*Arsenic*	*Heavy metals*
Spatial reduction in agricultural activity	Spatial reduction in agricultural activity	Injection of non-Newtonian fluid	Greening	Greening	Urban greening
Shift to conservation agriculture (i.e. no-tillage, permanent soil cover and crop diversification).	Shift to conservation agriculture	Injection of non-Newtonian fluid x Greening	Injection of non-Newtonian fluid	Use of reactive barrier (activated carbon/zero valent iron, depending on data availability)	Increase infiltration through drainage-infiltration-transport sewage system
Shift to no-till agriculture	Shift to no-till agriculture	Biochar application in the unsaturated zone	Injection of non-Newtonian fluid x Greening	Use of reactive barrier (activated carbon/zero valent iron, depending on	Increase infiltration through drainage-infiltration-transport sewage

Measures					
Agricultural		**Industrial**			**Urban**
Zelivka	**Koprzywianka**	**Örnsköldsvik**	**BRGM**	**BRGM/SGI**	**Rotterdam**
Nitrogen, pesticides, PFAS, Nitrification Inhibitors (+ water balance and crop yield)	*Nutrients, pesticides, heavy metals*	*PFAS*	*PFAS*	*Arsenic*	*Heavy metals*
				data availability) x Greening	system x Urban greening
Permanent soil organic cover	Permanent soil organic cover	Biochar application in the unsaturated zone x Greening		Phytoremediation	Biodiversity conservation
	Reduce chemical fertilizer			Phytoremediation x Greening	Biodiversity conservation x Urban greening
Crop rotation and diversification (more legumes)	Crop rotation and diversification (more legumes)				Sludge recycling and deposition on soil
Decrease pesticide usage	Decrease pesticide usage				Sludge recycling and deposition on soil x Urban greening
Shift to conservation agriculture x Spatial reduction in agricultural activity	Increased water retention through modification of streams, and development of retention basins and wetlands				
Increased water retention through modification of streams, and development of retention basins and wetlands	Increased water retention through modification of streams, and development of retention basins and wetlands x Shift to conservation agriculture				
Increased water retention through modification of landuse x Spatial reduction in agricultural activity	Increased water retention through modification of streams, and development of retention basins and wetlands x Spatial reduction in agricultural activity				
Increased water retention through modification of	Sludge recycling and deposition on soil				

Measures					
Agricultural		**Industrial**			**Urban**
Zelivka	**Koprzywianka**	**Örnsköldsvik**	**BRGM**	**BRGM/SGI**	**Rotterdam**
Nitrogen, pesticides, PFAS, Nitrification Inhibitors (+ water balance and crop yield)	*Nutrients, pesticides, heavy metals*	*PFAS*	*PFAS*	*Arsenic*	*Heavy metals*
landuse x Conservation agriculture					
Increased water retention through modification of landuse x Spatial reduction in agricultural activity x Conservation agriculture	Reducing atmospheric emissions				

Appendix D.3 References

IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland, 184 pp., doi: 10.59327/IPCC/AR6-9789291691647.

PHISHES

PHysically-based Integrated
Soil HEalth Simulation platform

Predicting soil health to protect ecosystems

For More Information

- Visit PHISHES website:
  www.phishes-project.eu/
- Follow PHISHES on LinkedIn:
  www.linkedin.com/company/phishes-project/

**Funded by
the European Union**

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.