



## D 4.3 Repository infrastructure, components and APIs, v3

M31/MARCH 2026



**Funded by  
the European Union**

This project has received funding from the Horizon Europe research and innovation programme under Grant Agreement No 101112838.

Acronym	SoilWise
Project Full Title	An open access knowledge and data repository to safeguard soils
GA number	101112838
Topic	HORIZON-MISS-2022-SOIL-01-01
Type of Action	HORIZON Innovation Actions
Project Duration	48 months
Project Start Date	1-Sep-2023
Project Website	<a href="https://soilwise-he.eu/">https://soilwise-he.eu/</a>
Deliverable Title	D 4.3 Repository infrastructure, components and APIs, v3
Delivery Time (DOA)	M31
Deliverable Submission Date	30/04/2026
Status	V3
Dissemination Level	PU - Public
Deliverable Lead	Thorsten Reitz (WE)
Author(s)/Organisation(s)	Ronald Kilian (WE), Thorsten Reitz (WE), Rob Lokers (WR), Dajana Snopková (MU), Tomáš Řezník (MU), Paul van Genuchten (ISRIC), Nick Berkvens (EV ILVO)
Contributor(s)	WE, MU, WR, WU, ZALF, EV ILVO, DOMG – VL O, CREA, ISRIC, NP
Peer-Reviewers	Anna Fensel (WU), Tomáš Pavelka (MU)
Contact	<a href="mailto:tr@wetransform.to">tr@wetransform.to</a> , <a href="mailto:rk@wetransform.to">rk@wetransform.to</a>
Work Package	WP4
Dissemination level	Public
Keywords	SWC functionalities, technical components, implementation, source code, data management components, prototype, technical documentation
Abstract	The purpose of this deliverable is to extend and update instance of the SWC infrastructure to v3 and to provide the SWC, with its code, the UI and API interfaces, after the end of the integration and validation phase (cycle 2).

## Disclaimer

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. Neither the European Union nor the granting authority can be held responsible for them.

In this document, the acronym 'DOMG – VL O' is used to refer to the Department of the Environment and Spatial Development, Flanders, Belgium, as per the partner's request for clarification. It is noted that in the grant agreement, the partner is identified by the acronym VL O (Vlaamse Gewest).

## List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CI-CD	Continuous Integration and Continuous Delivery / Development
CIRAD	CENTRE DE COOPERATION INTERNATIONALE EN RECHERCHE AGRONOMIQUE POUR LEDEVELOPPEMENT
CORDIS	Community Research and Development Information Service
CREA	CONSIGLIO PER LA RICERCA IN AGRICOLTURA E L'ANALISI DELL'ECONOMIA AGRARIA
CSV	Comma-separated values
CSW	Catalog Service for the Web
DG	Director General
DM	Data Management
DoD	Definition of Done
DOMG – VL O	VLAAMSE GEWEST
EJP	European Joint Programme
ESDAC	European Soil Data Centre
ETL	Extract, Transform, Load
ETS	Executable Test Suite
EU	European Union

EUSO	EU Soil Observatory
EV ILVO	EIGEN VERMOGEN VAN HET INSTITUUT VOOR LANDBOUW- EN VISSERIJONDERZOEK
FAIR	Findable, Accessible, Interoperable and Reusable
INSPIRE	Infrastructure for Spatial Information in Europe
ISO	International Organization for Standardization
ISRIC	STICHTING INTERNATIONAL SOIL REFERENCE AND INFORMATION CENTRE
JPEG	Joint Photographic Experts Group
JRC	Joint Research Centre
JSON	JavaScript Object Notation
KM	Knowledge Management
M	Month
ML	Machine Learning
MU	Masaryk University
NP	NEUROPUBLIC AE PLIROFORIKIS & EPIKOINONION
OGC	Open Geospatial Consortium
PDF	Portable Document Format
PNG	Portable Network Graphics
PSI	Public Sector Information
PU	Public

---

REST	Representational State Transfer
RDF	Resource Description Framework
SOS	Sensor Observation Service
SQLite	Structured Query Language Lite
STAC	Spatio Temporal Asset Catalogue
SVG	Scalable Vector Graphics
SWC	SoilWise Catalogue
T	Task
UI	User Interface
URL	Uniform Resource Locator
WE	WETRANSFORM GMBH
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WP	Work Package
WPS	Web Processing Service
WR	STICHTING WAGENINGEN RESEARCH
WU	WAGENINGEN UNIVERSITY
XML	Extensible Markup Language

PROJECT NAME – SoilWise



Project Number 101112838

ZALF

LEIBNIZ-ZENTRUM FUER AGRARLANDSCHAFTSFORSCHUNG

# Table of Contents

<b>EXECUTIVE SUMMARY .....</b>	<b>9</b>
<b>1 INTRODUCTION.....</b>	<b>10</b>
1.1 PROJECT SUMMARY.....	10
1.2 DOCUMENT SCOPE.....	10
1.3 DOCUMENT STRUCTURE.....	11
1.4 RELATIONSHIP TO OTHER PROJECT DELIVERABLES.....	11
1.5 RELATIONSHIP TO PROJECT TASKS.....	12
<b>2 TECHNICAL COMPONENTS IMPLEMENTED/DEPLOYED IN V3 .....</b>	<b>13</b>
<b>3 MIGRATION OF SWC INFRASTRUCTURE TO WETRANSFORM .....</b>	<b>20</b>
3.1 CURRENT SETUP AND PRE-V3 BASELINE .....	21
3.2 WORK DONE IN THE V3 ITERATION .....	21
3.3 PLANNING FOR EUSO INTEGRATION.....	22
<b>4 TECHNICAL DOCUMENTATION .....</b>	<b>24</b>
<b>5 RELEASE NOTES.....</b>	<b>25</b>
<b>6 CONCLUSION AND OUTLOOK .....</b>	<b>34</b>
<b>REFERENCES.....</b>	<b>35</b>

---

## List of Tables and Figures

Figure 1: SWC 3<sup>rd</sup> prototype architecture (light-colored elements are under development and not part of 3<sup>rd</sup> prototype)..... 14

Figure 2: Interfaces implemented (available at <https://prototype-3-0.soilwise-architecture.pages.dev/>) ..... 17

Table 1: List of components of the 3<sup>rd</sup> delivered prototypes and their source code ..... 15

Table 2: List of APIs and User Interfaces of the prototype and their endpoints ..... 17

Table 3: Components migrated to the unified wetransform infrastructure in v3, with their pre-v3 and v3 locations and migration status..... 20

Table 4: Configuration of the soilwise (test) and soillive (production) environments deployed from the same OpenTofu codebase..... 21

---

## Executive Summary

The SoilWise project, under the Horizon Innovation Actions, aims to develop an open-access soil knowledge and data repository to support users in safeguarding soil across Europe. With 60-70% of European soils currently considered unhealthy—losing their capacity to support food production, biodiversity, and the regulation of water, nutrient, and carbon cycles—the project addresses the urgent need for reliable and harmonized data. Aligning with the EU Mission 'A Soil Deal for Europe', SoilWise is a 48-month initiative involving 15 partners, designed to create a scalable, modular repository that leverages AI and ML technologies to make soil data Findable, Accessible, Interoperable, and Reusable (FAIR).

This deliverable D4.3 presents significant advancements in the SWC infrastructure during the v3 cycle. Building upon the stakeholder-driven use cases defined in D1.3, and D1.2, this version marks the transition from a distributed prototype to a unified, centrally managed environment. A core milestone is the migration of the SWC infrastructure to wetransform, utilizing OpenTofu/Terraform to manage a scalable Kubernetes-based architecture. While the former setup spanned clusters at distributed locations, v3 establishes the GitOps framework required to unify these deployments into a single, optimized environment to improve efficiency and interoperability. Furthermore, this setup can easily be further migrated to JRC and deployed at EUSO premises at a later stage.

In particular, this deliverable details advancements in the SWC infrastructure, focusing on:

- **Unified Infrastructure Migration:** Transitioning to a GitOps model using OpenTofu and dynamic YAML templating, with separate test (`soilwise`, currently active) and production (`soillive`, provisioned) namespace configurations to support staged deployment.
- **Infrastructure and Platform Services:** Full deployment and lifecycle management of Keycloak for authentication and Prometheus/Grafana for system monitoring.
- **EUSO Integration (status & next steps):** The SWC exposes standardized discovery and access endpoints (e.g., OGC API Records, STAC, CSW, OAI-PMH, SPARQL) intended as building blocks for EUSO integration. The detailed integration guidelines and any EUSO-specific interface adaptations are being specified jointly with JRC.
- **Requirement Fulfilment:** Validation of key acceptance criteria to ensure the architecture supports the 11 identified key functionalities defined for previous iterations and 8 user stories defined for cycle 3.
- **Technical Documentation & Release Notes:** Updated documentation of architecture, component functionality, interfaces/endpoints, and deployment/operations aspects, accompanied by updated release notes for the delivered prototype.

# 1 Introduction

## 1.1 Project summary

Now more than ever, soil health is an issue that needs to be addressed urgently, as recent assessments state that 60-70% of European soils can be considered unhealthy (Bouman, 2022). The EU Mission ‘A Soil Deal for Europe’, the EU Soil Strategy and the proposal for a Soil Monitoring and Resilience Directive (5 July 2023), aims to have 75% of EU soils healthy or significantly improved by 2030 and all soils healthy in 2050. Reaching such an ambition requires, among others, access to reliable, harmonized existing and new data and knowledge collected at local, national and EU levels to allow **informed decision-making at all scales to support the proposed Soil Monitoring and Resilience Directive and the EU Soil Strategy**.

The SoilWise project will provide an integrated and actionable access point to scattered and heterogeneous soil data and knowledge in Europe, making them FAIR (Findable, Accessible, Interoperable and Reusable) and improve trust, willingness, and ability to share and re-use soil data and knowledge. In three project development cycles, **co-creation and co-validation by multi-stakeholder groups are at the centre of project activities**. SoilWise recognises existing workflows and repositories for specific user needs and aims to work with them to enhance their discoverability, approachability, and interconnection. An open, modular, scalable, and extensible knowledge and data repository building on existing and new technologies will be provided while respecting data ownership, access policies and privacy. AI- and ML- techniques will be employed to interlink scattered data and knowledge, automate the processes, infer new knowledge and increase FAIRness. **SoilWise applies infrastructure thinking instead of project thinking to design a repository for at least a decade to support EU SO involvement accordingly**. The SoilWise Catalogue and community are designed to be a joint starting point and common ground for countries, the European Commission, and other stakeholders to jointly guide soil and related spatial policy and informed decision-making towards the 2030 goals of the Green Deal, achieve healthy soils in 2050 and ensure broad uptake and implementation by land managers, policy, research, and industry.

All personal data acquired through SoilWise is processed in strict accordance with the relevant EU privacy regulations, highlighting our dedication to uphold to the highest standards of data privacy and security for our users.

## 1.2 Document scope

The deliverable D4.3 presents the further advancement of the initial implementation and the strategic migration of the SoilWise (SWC) infrastructure to wetransform, developed across cycles v2 and v3. This cycle marked a significant step in establishing the core functionalities of the SWC system and a robust infrastructure and suitable APIs to support continuous operation. A central update in v3 is the transition from a distributed setup across multiple locations to a unified deployment using Kubernetes, ensuring high scalability and reliability.

The migration utilizes a "Provider-Template" pattern managed by OpenTofu, which allows the team to maintain the familiarity of existing Kubernetes YAML manifests while transforming them into dynamic templates. This approach replaces static configurations with a variable-driven model, where environment-specific parameters such as Ingress hosts are injected at runtime. Infrastructure state is securely persisted in an AWS S3 bucket, while credentials remain protected through Gopass and GitHub Secrets.

The third version of this document (v3) has been significantly extended compared to version v2, notably through the addition of a new chapter focused on the Migration of SWC Infrastructure (Chapter 3). This update reflects the evolving development status of the core technical components and the transition toward a more centralized operational model.

In particular, Chapter 2 details the technical components deployed during v3, describing an architecture that currently spans one Kubernetes cluster at wetransform. This host manages identity and access via [Keycloak](#), the SWC and specialized knowledge components, and system observability via [Prometheus](#) and [Grafana](#).

While these components were distributed in the past, version v3 describes the strategic efforts undertaken to unify these deployments into a single, cohesive environment. This consolidation is designed to eliminate cross-site latency, improve resource efficiency, and enhance the overall scalability and interoperability of the SWC system.

The focus of Chapter 3 is to detail the migration of the SWC Infrastructure, transitioning from the previous distributed setup to a unified, automated environment. It describes the technical implementation of the migration, utilizing OpenTofu/Terraform to manage the lifecycle of the unified cluster.

The technical documentation and release notes are updated to v3 in chapters 4 and 5.

Chapters 1.2 (Document scope) and 1.3 (Document structure) have been substantially revised to reflect the v3 architecture and the migration to wetransform; the relationship sections 1.4 and 1.5 received minor updates.

## 1.3 Document structure

This document is comprised of the following chapters:

- **Chapter 1** introduces the project and the document, and outlines the scope and the architecture derived for SWC v2,
- **Chapter 2** provides a list of the technical components implemented/deployed in SWC v3 and links to source code,
- **Chapter 3** describes the SWC infrastructure migration to wetransform,
- **Chapter 4** details the technical documentation,
- **Chapter 5** contains the Fulfilment of the system requirements,
- **Chapter 6** contains the release notes.

Technical documentation is available at <https://prototype-3-0.soilwise-documentation.pages.dev/> and a PDF exported version is also available on demand as a non-editable version saved at the date of the deliverable.

## 1.4 Relationship to other project deliverables

This deliverable relates to and complements the following deliverables:

- D1.1, D1.2 – Usage Scenarios, requirements, v1, v2 (M6, M36)
- D1.3 – Repository architecture, v1, v2 (M08, M42)
- D2.1, D2.2, D2.3, D2.4 – Developed & Integrated DM components, v1, v2, v3, v4 (M13, M18, M31, M47)

- D3.1, D3.2, D3.3, D3.4 – Developed & Integrated KM components, v1, v2, v3, v4 (M13, M18, M31, M47)
- D4.2, D4.3, D4.4 – Repository infrastructure, components and APIs, v1, v3, v4 (M13, M31, M47)
- D1.5, D1.6 – Repository GM, v1, v2 (M21, M42)
- D4.5, D4.6, D4.7 – Repository Data and Knowledge Resources, v1, v2, v3 (M21, M34, M46)
- D5.3, D5.4, D5.6 – Deployment and Evaluation Report, v1, v2, v3 (M21, M34, M46)
- D7.2, D7.3, D7.4 – Open Science and Data Management plan, v1, v2, v3 (M6, M27, M48)

## 1.5 Relationship to project tasks

This deliverable relates to the following project tasks:

- T1.3 Requirements, Validation framework and Rolling plan – will feed and update the design and implementation of DM components (T2.1 and T2.2)
- T1.4 Define SoilWise Architectural Design – will be considered in the design and implementation of DM components (T2.1 and T2.2)
- T1.5 Define SoilWise Multi-Stakeholder governance model – will be considered in the design and implementation of DM components (T2.1 and T2.2)
- T2.1 Design of the data technology components
- T2.2 Implementation and deployment of data components
- T2.3 AI and ML for data findability and accessibility – will extend the design and implementation of DM components (T2.1 and T2.2)
- T2.4 Strategy for FAIRness on soil data – will be followed in the design and implementation of DM components (T2.1 and T2.2)
- T3.1 Design of the KM components – will consider and align with the design of DM components (T2.1)
- T3.2 Implementation and deployment of knowledge component – will consider and align with the implementation of DM components (T2.2)
- **T4.1 Repository digital infrastructure for deployment and delivery – will integrate implementation of DM components (T2.2) and KM components (T3.2)**
- **T4.2 Interfaces for access, sharing, population and integration with EUSO – will be considered in the design and implementation of DM components (T2.1 and T2.2)**
- **T4.3 Solutions & repository validation and population – will validate designed and implemented DM components (T2.1 and T2.2)**
- T5.2 User Cases implementation and demonstration – will demonstrate the functionality of implemented DM components (T2.2)
- T7.2 Technical and Scientific Management – prepares an Open Science and Data Management Plan (D7.2 – D7.4) describing the data management level of handled (meta)data resources

## 2 Technical components implemented/deployed in v3

This chapter lists the technical components implemented and deployed in v3 of the SWC and links to their source code and documentation. The unified-infrastructure migration that hosts these components is described separately in Chapter 3.

Figure 1 depicts the overall architecture of SWC that has been updated to v3. A major change involved rebuilding the SWC frontend with React to facilitate a more user-friendly search experience. Furthermore, the following components have been added to the SWC: Soil Companion, Vocabulary Viewer, Apache Solr (improved search functionality), Data publication support components, and the SoilMission News Feed Harvester.

In this section, a list of the implemented/deployed technical components of the 3<sup>rd</sup> delivered SoilWise prototype is given together with the functionalities fulfilled, links to the source codes, guidelines, and tutorials. Starting with the 2<sup>nd</sup> prototype (delivered after the validation and integration phase of the first project cycle, M18), more elaborate components were also released on Zenodo.

The derived architecture of v3 is shown in Figure 1 and can be accessed via this URL <https://prototype-3-0.soilwise-architecture.pages.dev/>.

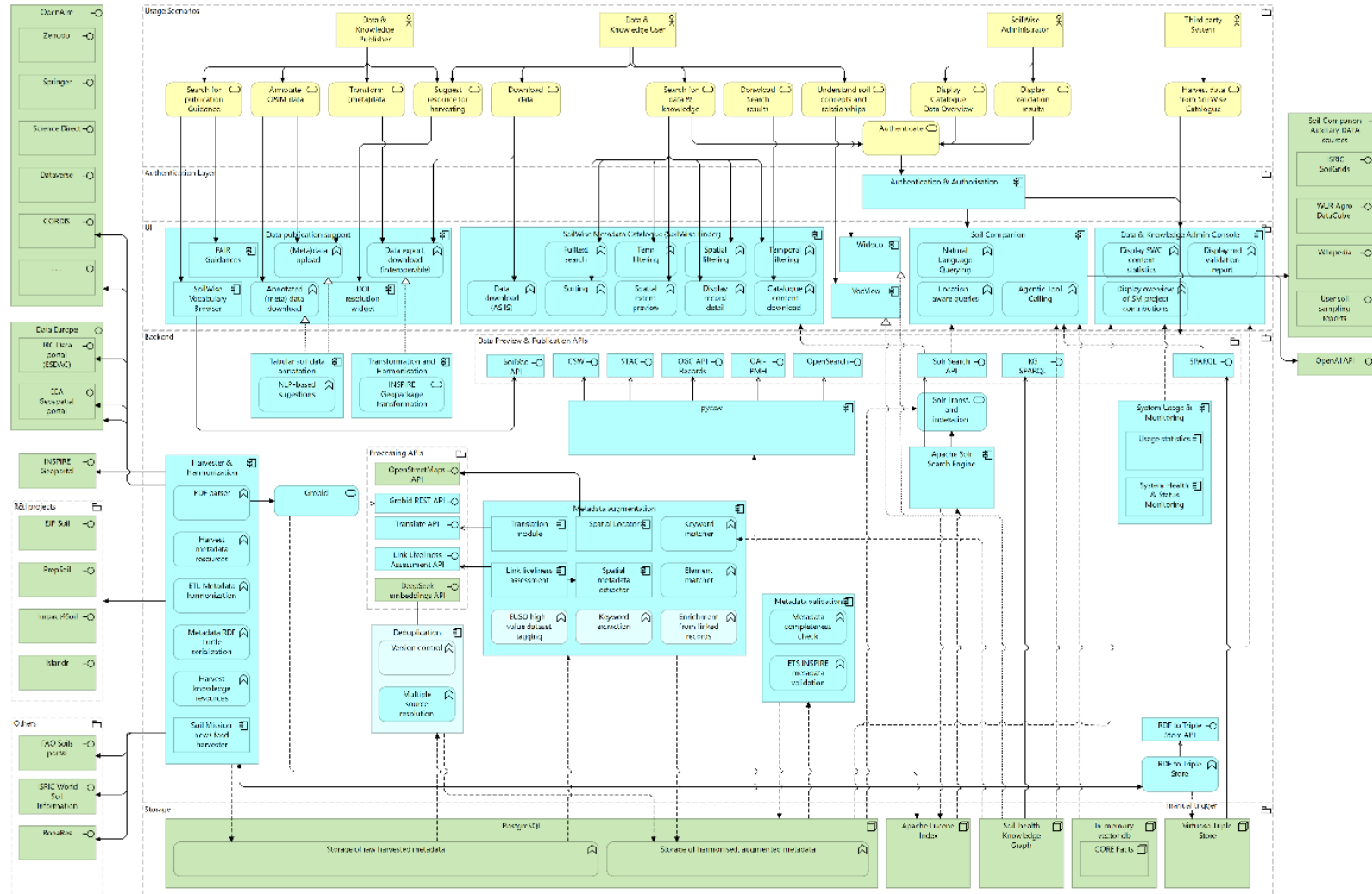


Figure 1: SWC 3rd prototype architecture (light-colored elements are under development and not part of 3rd prototype)



Table 1: List of components of the 3<sup>rd</sup> delivered prototypes and their source code

Component	Function / API	Source code (GitHub)	Source code (Zenodo)	Documentation
<b>Harvester</b>	Harvest metadata resources	<a href="https://github.com/soilwise-he/harvesters/releases/tag/v0.3.0">https://github.com/soilwise-he/harvesters/releases/tag/v0.3.0</a>	<a href="https://doi.org/10.5281/zenodo.14923562">https://doi.org/10.5281/zenodo.14923562</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/ingestion/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/ingestion/</a>
	Harvest knowledge resources			
	Metadata RDF Turtle serialization			
	RDF to Triple store			
	Harvest Soil Mission news feed			
<b>Metadata Harmonisation</b>	ETL Metadata harmonization	<a href="https://github.com/soilwise-he/md-harmonization/releases/tag/0.1.0">https://github.com/soilwise-he/md-harmonization/releases/tag/0.1.0</a>	<a href="https://zenodo.org/records/19909785">https://zenodo.org/records/19909785</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/metadata_harmonisation/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/metadata_harmonisation/</a>
	Duplication identification			
<b>SoilWise Finder (Metadata Catalogue)</b>	Metadata & knowledge NLP transformations and indexation	<a href="https://github.com/soilwise-he/SoilWiseFinder/releases/tag/v2.0.0">https://github.com/soilwise-he/SoilWiseFinder/releases/tag/v2.0.0</a>	<a href="https://doi.org/10.5281/zenodo.19691525">https://doi.org/10.5281/zenodo.19691525</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/catalogue/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/catalogue/</a>
	Search API on Solr			
	Solr configuration			
	Catalogue User Interface			
<b>Apache Solr Search Engine</b>	Search platform	Not applicable	Not applicable	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/catalogue/#search-engine-index-and-search-strategies">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/catalogue/#search-engine-index-and-search-strategies</a>
<b>pycsw</b>	CSW API	<a href="https://github.com/geopython/pycsw/releases/tag/3.0.0-beta1">https://github.com/geopython/pycsw/releases/tag/3.0.0-beta1</a>	Not applicable	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/catalogue/#integrations-interfaces">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/catalogue/#integrations-interfaces</a>
	OGC API-Records			
	STAC API			
	OAI-PMH API			
	OpenSearch			
<b>Metadata validation</b>	ETS INSPIRE validation	<a href="https://github.com/soilwise-he/metadata-validator/releases/tag/v0.4.1">https://github.com/soilwise-he/metadata-validator/releases/tag/v0.4.1</a>	<a href="https://doi.org/10.5281/zenodo.14924543">https://doi.org/10.5281/zenodo.14924543</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/metadata_validation/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/metadata_validation/</a>
	Metadata completeness check			
<b>Metadata Augmentation</b>	Translation module	<a href="https://github.com/soilwise-he/metadata-">https://github.com/soilwise-he/metadata-</a>	<a href="https://doi.org/10.5281/zenodo.14924181">https://doi.org/10.5281/zenodo.14924181</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/tech">https://prototype-3-0.soilwise-documentation.pages.dev/tech</a>
	Link liveliness assessment			

	Keyword Matcher	<a href="https://github.com/soilwise-he/soilwise-he/releases/tag/v0.3.0">augmentation/releases/tag/v0.3.0</a>		<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/metadata_augmentation/">nical_components/metadata_augmentation/</a>
	Element Matcher			
	Spatial Metadata Extractor			
<b>Soil Health Knowledge Graph</b>	Storage of soil health knowledge as a graph representation	<a href="https://github.com/soilwise-he/soilwise-he/soilwise-he/releases/tag/v0.2.6">https://github.com/soilwise-he/soilwise-he/soilwise-he/releases/tag/v0.2.6</a>	<a href="https://doi.org/10.5281/zenodo.14936019">https://doi.org/10.5281/zenodo.14936019</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/knowledge_graph/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/knowledge_graph/</a>
	Knowledge Graph querying (SPARQL endpoint)			
<b>Repository Storage</b>	Storage of raw harvested metadata – PostgreSQL / vector DB	Not applicable	Not applicable	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/storage/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/storage/</a>
	Storage of augmented metadata – PostgreSQL / vector DB			
	Storage of augmented, linked metadata, knowledge graph - Virtuoso			
	Storage of indexed metadata and knowledge content			
<b>Soil Companion (chatbot)</b>	Intelligent conversational user interface using natural language querying	<a href="https://github.com/soilwise-he/soilwise-soil-companion/releases/tag/v1.0.0">https://github.com/soilwise-he/soilwise-soil-companion/releases/tag/v1.0.0</a>	<a href="https://doi.org/10.5281/zenodo.19252205">https://doi.org/10.5281/zenodo.19252205</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/soil_companion/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/soil_companion/</a>
	Location aware queries and responses			
	Access and integration of external data sources through agentic tool calling			
	Load and merge auxiliary DATA resources			
<b>Data publication support</b>	DOI Resolution Widget	<a href="https://github.com/soilwise-he/soilwise-util/releases/tag/v1.2.0">https://github.com/soilwise-he/soilwise-util/releases/tag/v1.2.0</a>	<a href="https://doi.org/10.5281/zenodo.19708262">https://doi.org/10.5281/zenodo.19708262</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/data_publication_support/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/data_publication_support/</a>

	Tabular Soil Data Annotation	<a href="https://github.com/soilwise-he/TabularDataAnnotation/releases/tag/v1.0.1">https://github.com/soilwise-he/TabularDataAnnotation/releases/tag/v1.0.1</a>	<a href="https://doi.org/10.5281/zenodo.19569572">https://doi.org/10.5281/zenodo.19569572</a>	
	SoilWise GeoPackage	<a href="https://github.com/soilwise-he/Geopackage-so/releases/tag/1.0.0">https://github.com/soilwise-he/Geopackage-so/releases/tag/1.0.0</a>	<a href="https://doi.org/10.5281/zenodo.19465866">https://doi.org/10.5281/zenodo.19465866</a>	
<b>Soil Vocabulary Browser (SoilVoc)</b>	Display linked soil vocabulary terms W3id REST API	<a href="https://github.com/soilwise-he/soil-vocabs/releases/tag/v0.1.0">https://github.com/soilwise-he/soil-vocabs/releases/tag/v0.1.0</a>	<a href="https://zenodo.org/records/19710234">https://zenodo.org/records/19710234</a>	<a href="https://main.soilwise-documentation.pages.dev/technical_components/knowledge_graph/#soilvoc">https://main.soilwise-documentation.pages.dev/technical_components/knowledge_graph/#soilvoc</a>
<b>Data &amp; Knowledge Administration Console</b>	Display Metadata Catalogue content statistics	<a href="https://github.com/soilwise-he/soilwise-superset/releases/tag/v1.0.1">https://github.com/soilwise-he/soilwise-superset/releases/tag/v1.0.1</a>	<a href="https://doi.org/10.5281/zenodo.19567030">https://doi.org/10.5281/zenodo.19567030</a>	<a href="https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/admin_console/">https://prototype-3-0.soilwise-documentation.pages.dev/technical_components/admin_console/</a>
	Display metadata validation reports			
	Display overview of Soil Mission project contributions			

Here we provide a deliverable-level inventory of the main interfaces (APIs and user interfaces) exposed by the SWC for discovery, access, and supporting processing/quality services. These interfaces also serve as building blocks for integration with the European Soil Observatory (EUSO); findings and the integration plan agreed with JRC are summarized in Section 3.3. The interface inventory is complemented by the implemented-interface overview in Figure 2 and the endpoint list in Table 2.

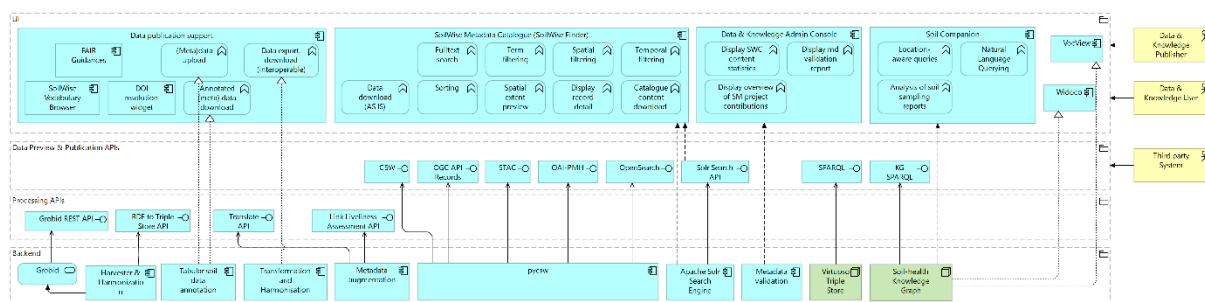


Figure 2: Interfaces implemented (available at <https://prototype-3-0.soilwise-architecture.pages.dev/>)

Figure 2 summarizes the interfaces that are implemented as part of the SWC technical infrastructure (status as of April 2026). It provides a concise overview of the available APIs and user interfaces and serves as the visual anchor for the endpoint inventory listed in Table 2.

Table 2: List of APIs and User Interfaces of the prototype and their endpoints

API / UI	End point
SPARQL	<a href="https://sparql.soilwise.wetransform.eu/sparql">https://sparql.soilwise.wetransform.eu/sparql</a>

<b>Discovery APIs</b>	OGC API-Records	<a href="https://api.soilwise.wetransform.eu/records/openapi">https://api.soilwise.wetransform.eu/records/openapi</a>
	Spatio Temporal Asset Catalog (STAC)	<a href="https://api.soilwise.wetransform.eu/records/stac/openapi">https://api.soilwise.wetransform.eu/records/stac/openapi</a>
	Catalog service for the Web (CSW)	<a href="https://api.soilwise.wetransform.eu/records/csw">https://api.soilwise.wetransform.eu/records/csw</a>
	Protocol for Metadata Harvesting (OAI-PMH)	<a href="https://api.soilwise.wetransform.eu/records/oaipmh">https://api.soilwise.wetransform.eu/records/oaipmh</a>
	OpenSearch	<a href="https://api.soilwise.wetransform.eu/records/opensearch">https://api.soilwise.wetransform.eu/records/opensearch</a>
	Solr Search API	<a href="https://catalogue.soilwise-he.eu/search-api/swagger-ui/index.html">https://catalogue.soilwise-he.eu/search-api/swagger-ui/index.html</a>
	<b>Processing APIs</b>	Translate API
Link Liveness Assessment API		<a href="https://api.soilwise.wetransform.eu/linky/docs">https://api.soilwise.wetransform.eu/linky/docs</a>
DOI Resolution API		<a href="https://api.soilwise.wetransform.eu/util/docs">https://api.soilwise.wetransform.eu/util/docs</a>
Mission Soil News Feed API		<a href="https://api.soilwise.wetransform.eu/util/docs">https://api.soilwise.wetransform.eu/util/docs</a>
<b>End-User Interfaces</b>	SoilWise Metadata Catalogue	<a href="https://catalogue.soilwise-he.eu/">https://catalogue.soilwise-he.eu/</a>
	Soil Companion	<a href="https://soil-companion.containers.wur.nl/app/index.html">https://soil-companion.containers.wur.nl/app/index.html</a>
	Tabular Soil Data Annotation	<a href="https://dataannotator-swr.streamlit.app/">https://dataannotator-swr.streamlit.app/</a>
	Soil Vocabulary Viewer	<a href="https://soilwise-he.github.io/soil-vocabs/">https://soilwise-he.github.io/soil-vocabs/</a>
	Data & Knowledge Administrator Console	<a href="https://superset.soilwise.wetransform.eu/superset/dashboard/p/P52OgRVBGo9/">https://superset.soilwise.wetransform.eu/superset/dashboard/p/P52OgRVBGo9/</a>

<b>Administrative User Interfaces</b>	System Monitoring	<a href="https://grafana.soilwise.wetransform.eu/">https://grafana.soilwise.wetransform.eu/</a>
---------------------------------------	-------------------	---

Table 2 lists the main interface entry points of the v3 prototype. A full component-level interface specification, including OpenAPI documentation where available, is maintained in the technical documentation.

### 3 Migration of SWC infrastructure to wetransform

This chapter summarises the infrastructure consolidation activities for the SoilWise Catalogue (SWC) in the v3 iteration. Prior to v3, SWC services were operated across two partner environments: a WUR-hosted cloud running the catalogue (pycsw and frontend), Apache Solr, the Virtuoso triple store, and the harvester orchestration; and WUR-hosted containers running the SPARQL endpoint and the Soil Health Knowledge Graph. As of M31, all of these components have been migrated to a single Kubernetes cluster operated by wetransform and managed as code via OpenTofu. The cluster is organised into two namespaces — `soilwise` (test) and `soillive` (production) — both defined and provisioned through the same IaC codebase. The migrated services currently run in the test namespace (`soilwise`); the production namespace (`soillive`) is prepared but not yet active and will be activated once integration testing is concluded. The migration is functionally complete for the components in scope, with the exception of items listed under §3.3 (EUSO integration). The Soil Companion chatbot remains hosted at WUR and is out of scope for this migration. Keycloak (authentication) and the Prometheus/Grafana/Loki monitoring stack were not migrated, as both were already operated by wetransform; they have, however, been newly integrated into the unified deployment and extended to cover the migrated services.

The following table summarises the migration scope at component level:

*Table 3: Components migrated to the unified wetransform infrastructure in v3, with their pre-v3 and v3 locations and migration status.*

Component	Pre-v3 location	v3 location	Migration status
<b>Catalogue (pycsw + React frontend)</b>	WUR cloud	wetransform Kubernetes (soilwise / soillive)	Migrated
<b>Apache Solr / Search-API</b>	WUR cloud	wetransform Kubernetes	Migrated
<b>Virtuoso triple store</b>	WUR cloud	wetransform Kubernetes (sparql.soilwise.wetransform.eu)	Migrated
<b>Harvester (Argo CronWorkflows)</b>	WUR cloud	wetransform Kubernetes	Migrated
<b>Metadata Validation (ETS INSPIRE + completeness + link liveness)</b>	WUR cloud	wetransform Kubernetes	Migrated
<b>Util APIs (DOI Resolution, Mission Soil News Feed, History)</b>	WUR cloud	wetransform Kubernetes	Migrated
<b>Soil Health Knowledge Graph / SPARQL endpoint</b>	WUR containers	wetransform Kubernetes	Migrated
<b>PostgreSQL (catalogue database)</b>	WUR cloud	wetransform Kubernetes (postgis/postgis:16-3.5)	Migrated
<b>Keycloak (authentication)</b>		wetransform	Newly integrated
<b>Monitoring (Prometheus / Grafana / Loki / Alertmanager)</b>		wetransform	Newly integrated
<b>Soil Companion (chatbot)</b>	WUR cloud	WUR Kubernetes	Out of migration scope

All migrated components currently run in the test namespace (`soilwise`). The parallel production namespace (`soillive`) has been provisioned via the same IaC codebase but is not yet hosting active services. The technical implementation of the migrated setup — IaC engine, deployment templating, environment separation, and operational tooling — is described in §3.2 *Work done in the v3 iteration*.

### 3.1 Current setup and pre-v3 baseline

Before the v3 iteration, SWC services were operated in a distributed setup across partner environments (e.g., ISRIC/WUR, MU, and Neupublic). As a result, deployment and operations were executed in environment-specific ways and required partner-local coordination for updates and troubleshooting.

For the migration preparation leading into v3, operational configuration handling was aligned towards a unified approach. A test database was available in the cluster; access was initially provided via proxy, with a request pending to enable direct access via port 5432 with IP whitelisting. For CI/CD, Argo CD was included in the target setup and was being set up during the migration planning phase. Deployment assets were consolidated in a dedicated repository (`soilwise-he/k8s-soilwise`), and the SWC test domain was established at <https://catalogue.soilwise-he.eu/>.

### 3.2 Work done in the v3 iteration

The v3 iteration progressed the migration towards a unified, WE-operated setup and introduced an Infrastructure-as-Code (IaC) and GitOps-aligned deployment approach intended to improve repeatability, traceability, and environment parity.

The unified SWC infrastructure runs on a Kubernetes cluster, using the `hcloud-volumes` storage class for all persistent volumes.

The migration approach implemented in v3 adopted OpenTofu as the IaC engine to manage resource lifecycle and state, combined with a template-based deployment strategy to retain native Kubernetes YAML structures while enabling controlled parameterisation. Remote state persistence was implemented using an AWS S3 backend (`eu-central-1`), supporting team collaboration and traceability. Deployment automation followed a GitHub Actions pull-request workflow using a plan/apply pattern (plan on pull request; apply on merge), establishing a reviewable change process as part of the GitOps alignment. In addition, a weekly scheduled run (Sundays, 07:00 UTC) performs drift detection, automatically opening or updating GitHub issues if the live infrastructure state has diverged from the codebase.

Environment separation within the unified setup was implemented through a two-namespace strategy, with environment-specific configuration, hostnames, secrets, and feature flags managed through separate `.tfvars` files:

Table 4: Configuration of the *soilwise* (test) and *soillive* (production) environments deployed from the same OpenTofu codebase.

	Test	Production
Workspace	default	prod
Namespace	soilwise	soillive
Purpose	Integration testing, development	Live service (provisioned, not yet activated)
Configuration	config/test.tfvars	config/prod.tfvars

Both environments use the same container images and OpenTofu modules. Within the deployment templating work, the Virtuoso deployment was consolidated from static files into a unified template folder, with environment-specific differences handled through variables to reduce divergence while maintaining necessary configurability. For the database layer, the baseline container image referenced was `postgis/postgis:16-3.5`, tracking PostgreSQL 16 minor releases.

All SWC services currently run as single-replica deployments; no high-availability configuration is in place. Most services use a `Recreate` update strategy, meaning brief downtime occurs during redeployment. Linky and Virtuoso are exceptions, using `RollingUpdate` for zero-downtime updates.

A monitoring setup based on Prometheus, Grafana, Loki, and Alertmanager is operated by wetransform alongside the Kubernetes cluster. Prometheus scrapes real-time metrics from all services, the cluster, and host nodes; detailed Nginx OWS metrics are enabled per publication and service type. A second Prometheus LTS instance retains metrics for approximately 30 days to support trend analysis and availability reporting. Grafana provides dashboarding and visualisation. Loki with Promtail handles log aggregation, including HTTP status code filtering for 4xx/5xx error identification. Operational alerts are routed via Alertmanager to Slack, with PagerDuty available for critical environments.

In addition, Keycloak has been set up with credentials that have been shared and verified with the partners.

### 3.3 Planning for EUSO integration

Initial discussions with JRC during the v3 iteration established a shared framing for integrating the SoilWise Catalogue (SWC) into the European Soil Observatory (EUSO). The framing covers a division of responsibilities between the two parties, a scope distinction between core and added-value capabilities, an indicative timeline, and a set of open items to be resolved jointly during the remainder of the project.

A division of responsibilities was agreed in principle. SoilWise provides the data-management (DM) and knowledge-management (KM) components and their documentation, supports integration for testing and staging environments, prepares guidelines for EUSO integration, and exposes the interfaces required for discovery, access, sharing, and population of soil data and knowledge. JRC, in turn, is responsible for setting up the hosting environment at JRC premises, performing the installation and deployment of SWC components there, and developing the EUSO front-end interface(s) on top of the SWC backend.

A second framing concerns which parts of the SWC stack are considered core capabilities for EUSO and which are added-value extensions. Core capabilities must be open-source software, built on open-source technologies, conventional and proven, and manageable by JRC operations staff without specialist support; this is a hard requirement on the JRC side. Added-value components, by contrast, are optional and may rely on less conventional solutions — potentially proprietary, AI/ML-based, or both — with variable Technology Readiness Levels (TRL). This distinction will guide which components are mandatory in the initial JRC deployment and which can be considered for later iterations or excluded entirely depending on JRC's operational constraints.

The EUSO integration is being delivered as a dedicated workstream within cycle 3, running through to project end. An initial demonstration and development installation at JRC is foreseen during this period, in coordination with JRC's hosting environment setup, and is intended as a working installation for evaluation rather than a final hand-over. The final hand-over state will be reported in D4.4 (M47).

Three items remain to be clarified jointly with JRC in the run-up to the initial JRC installation. The first is a clearer mapping from the SoilWise technical component list (Table 1) to the logical/functional components used in EUSO planning, so the two architectures can be aligned with confidence. The second is an explicit categorisation of optional and proprietary components, including the functional impact of excluding any of them from the JRC deployment. The third is the definition of the user roles intended for SWC at EUSO, the purpose of each exposed API in that context, and the corresponding onboarding workflow for EUSO end users.

These items will be addressed jointly with JRC during cycle 3.

## 4 Technical documentation

For this deliverable, a stable release of the SWC technical documentation is published at <https://prototype-3-0.soilwise-documentation.pages.dev/>. A PDF exported version is also available on demand as the non-editable version made at the date of deliverable submission. The same documentation is maintained as a living environment in the public GitHub repository at <https://github.com/soilwise-he/SoilWise-documentation>, where it is continuously updated during the development process. It comprises a description of functionality, detailed technical specifications, and interfaces between all modular components, and is structured according to the main technical components.

## 5 Release Notes

The versions highlighted in green are relevant for the third development iteration.

### Infrastructure

#### v0.3

The SWC infrastructure has been migrated from a distributed setup to a unified Kubernetes cluster, managed as code using OpenTofu (Terraform-compatible). The two previously separate clusters (wettransform and ISRIC) have been consolidated under a single deployment repository ([k8s-soilwise](#)):

- Two environments are defined — test (namespace `soilwise`, currently active) and production (namespace `soillive`, provisioned for future activation) — built from the same OpenTofu codebase using environment-specific configuration files
- Infrastructure state is managed remotely via an S3 backend (AWS `eu-central-1`)
- All SWC components are containerised and published to the GitHub Container Registry (`ghcr.io/soilwise-he/*`)
- CI/CD is automated via GitHub Actions: pushes to main deploy to the test environment; the production environment will follow once integration testing is concluded; pull requests trigger plan-only runs; weekly drift detection opens GitHub issues if infrastructure state diverges from code
- Metadata harvesting is orchestrated through Argo CronWorkflows, scheduled with staggered start times to avoid resource contention

#### v0.2

No changes

#### v0.1

The current infrastructure of SWC v1 is distributed across different, independently operating premises. So far, a Kubernetes cluster has been set up in the wettransform premises, including:

- A Keycloak instance
- A hale»connect installation including the required infrastructure and monitoring via Prometheus and Grafana

Keycloak can be used to authenticate with hale»connect, supporting authentication via GitHub user accounts. The hale»connect and hale»studio can be used to transform and validate data/ metadata.

An additional Kubernetes cluster has been set up at ISRIC, which operates the SWC and the knowledge components. It is planned to integrate these distributed infrastructure components into a single setup in the upcoming cycles of v2/v3.

### Harvester

#### v0.3

The harvesting strategy was adjusted to focus primarily on the secondary sources (aggregators of multiple primary repositories such as OpenAIRE, CORDIS, INSPIRE geoportal, and `data.europa.eu`) and harvest primary sources only if there are clear benefits.

The repositories harvested in the current iteration are as follows:

**Major aggregators:**

- [OpenAIRE](#)
- [CORDIS](#)
- [Data.europe.eu](#)
- [INSPIRE](#)

**Directly harvested Mission Soil Projects**

- [EJPSoil](#)
- [Impact4Soil](#)
- [Islandr](#)
- [Prepsoil](#)

**Edge of scope**

- [ISRIC World Soil Information](#)
- [FAO](#)
- [EEA Geoportal](#)

**v0.2**

The harvester component version 0.2.0 expands the list of harvest records from v0.1 by following platforms:

- [EJP Soil](#)
- [PrepSoil](#)
- [FAO Soils portal](#)
- [EEA Datahub](#)
- [ISRIC - World Soil Information](#)

The processing of the harvesters after initial fetch has been improved:

- Processed results are written to a temporary table. Only when all processing is finished, the published results are overwritten.
- While processing, any augmented metadata facts are inserted in the final result.

**v0.1**

The harvester component version 0.1.0 is a set of java and python scripts which harvest records from selected platforms:

- [INSPIRE geoportal](#)
- [CORDIS](#)
- [ESDAC](#)
- [Bonares](#)

These scripts are typically set up to run at intervals, for example in a dedicated container in a GIT CI-CD environment or Google Cloud Run.

Records are initially stored as-is in a PostgreSQL database, identified by their MD5 hash.

After the initial run, the new task parses the records into a common model. For ISO19115 and STAC records the library OwsLib and pygeometa are used to parse records to Dublin Core RDF. ESDAC and Cordis/OpenAire are parsed to Dublin Core.

The common model is stored in a PostgreSQL database and can also be translated to RDF using the Dublin Core ontology. After harmonization the RDF is ingested into the triple store (Virtuoso). PostgreSQL is used as a staging layer for raw and harmonised records because it provides atomic insert/update semantics,

deduplication via MD5 hashing, and efficient incremental processing across harvesting runs. RDF/triple-store ingestion is performed downstream once records have been harmonised against the common model. A manual intervention is currently needed to ingest RDF from a tailored API which exposes the RDF from the PostgreSQL database.

### Metadata Harmonisation

#### v0.1.0

New component created from a subset of Harvester functionalities concerning metadata harmonization to a common database model.

### Catalogue user interface

#### v2.0.0

Pycsw Catalogue frontend was replaced by a custom-made solution using React on top of Apache Solr backend. Upgrade focused on improving the overall look-and feel of the user interface and supporting advanced search and filtering functionalities. Currently the user interface offers:

- **Fulltext search** with autocomplete
- **Resource type filter**
- **Thematic filters** (replacing former faceted search, filter by license, language, project, source, Mission Soil)
- **Temporal filter** focused on date of issue, update date and temporal coverage
- **Spatial filter**, with option to search using location, address search, selection of administrative units, or drawing a polygon
- **Combination of filters** and fulltext search enabling to narrow down the search results
- Paginated search results
- Highlighting of search phrases in the search results
- Preview of spatial extent directly in the search results
- Opening record's detail page in a side panel
- **Possibility to download search results**
- Option to share record's detail

Apache Solr backend was integrated into SWC to support advanced search functionality. Solr denormalises metadata from existing database models into Solr.documents, which are subsequently processed and assigned indexes, relevance scores, and other search-related attributes.

#### v0.2

Faceted search was updated and currently supports the following filters:

- filter by physical soil parameters (soil texture, WRB, soil structure, bulk density, porosity, water holding capacity, soil moisture),
- filter by chemical soil parameters (ph, organic matter, cation exchange capacity, electrical conductivity, nutrient content, soil carbon, soil nitrogen, soil phosphorus, heavy metals concentration),
- filter by biological soil parameters (microbial biomass, soil enzyme activities, soil fauna, soil respiration),
- filter by soil functions (soil fertility, water regulation, soil erosion control, carbon sequestration, soil health, supporting plant growth, contaminant filtration),

- filter by soil degradation indicators (soil erosion, soil compaction, soil salinization, soil acidification, soil contamination),
- filter by environmental soil functions (habitat for organisms, climate regulation, water filtration),
- filter by long-term field experiments (experimental treatments, temporal data, environmental covariates, soil productivity, soil management),
- filter by record's type (dataset, document, publication, software, services, series).

#### v0.1

The SoilWise prototype adopts a frontend, focusing on:

- minimalistic User Interface, to prevent a technical feel,
- paginated search results, sorted alphabetically, by date,
- option to filter by facets,
- preview of the dataset (if a thumbnail or OGC:Service is available), else display of its spatial extent,
- option to provide feedback to publisher/author,
- readable link in the browser bar, to facilitate link sharing.

### PyCSW

#### 3.0.0.beta1

With the redesign of Catalogue user interface, pycsw now serves as a backend component only, providing standardised APIs to access the contents of SWC and support external integration.

#### 3.0.0.beta1

The pycsw catalog is currently in its beta release, as the team is awaiting the official release of the OGC API – Records standard. Pycsw operates with a PostgreSQL database backend, ensuring efficient data management and retrieval. A set of template overrides has been applied to customize the default pycsw user interface, giving it a distinctive SoilWise look and feel.

### Metadata validation

#### v0.4.1

**Link Liveliness Assessment** is now considered to be a part of the Metadata Augmentation components.

**ETS INSPIRE Metadata validation** This component employs the INSPIRE Reference Validator, which is based on INSPIRE ATS and available as a validation service. All metadata records with the source property value equal to INSPIRE are validated against the INSPIRE validation. Metadata is stored in the PostgreSQL database. This validation is performed on non-augmented, non-harmonised metadata records.

#### v0.2

#### **Link Liveliness Assessment, v 1.1.4**

This version of the Link Liveliness checker has the following improvements:

- a number of bug fixes,
- with every check, the tool also captures the size and MIME type of the remote source, returns this information through the API and visualizes it in the catalogue.
- The API has been extended to include the newly tracked metadata fields:
  - link\_type: Shows the file format type of the resource (e.g., image/jpeg, application/pdf)

- link\_size: Indicates the size of the resource in bytes
- last\_modified: Provides the timestamp when the resource was modified

### Metadata Completeness Checker

Completeness is evaluated against a set of metadata elements for each harmonized metadata record in the SWC platform. Completeness is indicated in % of 100 for the following properties: Identification, Title, Abstract, Author/Organization, Date, Type, Rights, and Extent (geographic and temporal). Properties are evaluated by their existence, resulting in a score.

### v0.1

#### Link Liveliness Assessment

The component checks the status of web links in metadata records and consists of 2 parts: (1) an API which can be queried to retrieve link-test results by status or the status of an individual link. The component includes a generic python link checking library. An exception is made for links of type OGC: WMS, WFS, WCS, CSW, WPS, SOS, which return an error 500 if requested without parameters. The OwsLib is used to test these links. Link types are identified by analyzing the type of property of the link (in the OGC API – Records).

#### Metadata validation

In the hale»connect instance for SWC v1 ETF/ETS have been configured based on the INSPIRE soil metadata profile for the ISO 19115 standard, which can be used to validate soil and metadata. In this way, data/metadata can be uploaded, validated on the fly and downloaded as transformed data sets along with reports. Furthermore, data sets can be published using these services.

In addition, ETS profiles based on EJP Soil/EUSO specifications, and a minimum metadata profile are to be configured in the upcoming iteration.

## Data publication support

### FAIR Guidances, v 1.0.0

A first version of the strategy developed within SoilWise to improve the FAIRness of soil data and knowledge across Europe was published as a deliverable D2.5 *Report on strategy for FAIRness on soil data* (M27). On top of it, practical recommendations for data providers, supporting metadata preparation, repository selection, vocabulary use, provenance documentation, and licensing were published on Zenodo: <https://doi.org/10.5281/zenodo.18759085>.

### DOI Resolution widget, v 1.2.0

A tool which can be used to test different DOIs for the presence of the corresponding record inside SoilWise.

### Tabular Soil Data annotation, v 1.0.1

A SoilWise-developed [Streamlit](https://github.com/soilwise-he/TabularDataAnnotator-streamlit) web application (source: <https://github.com/soilwise-he/TabularDataAnnotator-streamlit>; Zenodo release: <https://doi.org/10.5281/zenodo.19569572>) that helps users create semantically correct metadata for tabular datasets (e.g., soil observations and measurements). It combines optional Large Language Model (LLM) assistance with semantic embeddings to annotate data columns with machine-readable descriptions, element definitions, units, methods, and vocabulary mappings. The tool further supports FAIR data publication.

### Geopackage, v 1.0.0

The SoilWise GeoPackage is the relational (SQLite-based) container to enable exchange, storage, and GIS-native use of soil data, with the explicit goal of making them FAIR and reusable across European policies,

research, and land management workflows. (Source: <https://github.com/soilwise-he/Geopackage-so>; Zenodo release: <https://doi.org/10.5281/zenodo.19465866>. Built on the OGC GeoPackage standard.)

### Transformation and Harmonization

Transformation and harmonization are further supported by the GeoPackage component (part of Data publication support).

#### v5.3.0

The first technical functions have been set up utilizing hale»studio and will be further optimized in the following iterations in collaboration with the stakeholders in the soil domain. By using the open-source software hale»studio from wetransform, soil data can be harmonised either by using preset transformation functions or by creating customised functions by the user.

hale»studio can be downloaded from GitHub: <https://github.com/halestudio/hale/>

### Metadata augmentation

#### v0.3.0

##### Link Liveliness Assessment, v1.1.8

LLA now supports on-demand URL validation by returning immediate feedback without storing results in the database. For the OGC web services, it can now additionally extract spatial information (coordinate system + extent bounding box), if available.

##### Spatial Metadata Extraction, v 0.3.0

A new component that uses Named Entity Recognition (NER) to extract location entities from metadata titles and abstracts, thus providing additional information about spatial reference. The component leverages a trained spaCy model.

#### v0.2

##### Keyword matcher

This module helps to facilitate better search user experience when searching for similar datasets or knowledge assets. Existing keywords which are present in metadata records are matched against a set of preferred keywords in the soil domain derived from Agrovoc, Gemet and ISO11704.

##### Metadata interlinker

SoilWise Storage comprises also a triple store, which enables storing links between harvested metadata on data and knowledge assets. SoilWise implements interlinking based on explicit links, which are found directly in metadata (e.g. projects in CORDIS are explicitly linked to documents and datasets). This information can be then displayed in the details of metadata records through Catalogue UI.

#### v0.1

##### Automatic metadata generation

To generate metadata (data set and service metadata), activate the corresponding button(s) when setting up the theme for the transformation process.

Setting up a transformation process in hale»connect

Complete the following steps to set up soil data transformation, validation and publication processes:

1. Log in to hale»connect
2. Create a new transformation project (or upload it)
3. Specify source and target schemas
4. Create a theme (this is a process that describes what should happen with the data)
5. Add a new transformation configuration. Note: Metadata generation can be configured in this step
6. A validation process can be set up to check against conformance classes

#### Executing a transformation process

1. Create a new dataset and select the theme of the current source data, and provide the source data file
2. Execute the transformation process. ETF validation processes are also performed. If successful, a target dataset and the validation reports will be created
3. View and download services will be created if required

#### Translation module

The component consists of 2 parts: (1) an API that can be queried to retrieve a translation and (2) an API endpoint that receives the translated string from the EU translation service. The translated strings are stored in a PostgreSQL database. Whenever a translation is requested for the second time, it is returned from the database.

### Soil Companion

#### v1.0.0

The Soil Companion is an AI chatbot developed in the SoilWise project. It provides an intelligent conversational interface through which users can explore European soil metadata, query global and country-specific soil data services, and receive answers grounded in the SoilWise knowledge repository.

The chatbot uses an agentic tool-calling approach: a large language model (LLM) autonomously decides which external data sources to consult for each question, executes the relevant tool calls, and synthesizes the results into a coherent response. Answers are enriched with auto-generated links to SoilWise vocabulary terms and Wikipedia articles. A sidebar Insight panel displays related SKOS vocabulary concepts and clickable chips that allow users to explore connected topics.

### Knowledge Graph

#### v0.2.6

The knowledge graph content was updated and extended.

Additionally, a user interface (Widoco) was added to enable documentation and exploration of the knowledge graph.

Knowledge Graph was also integrated with the Keyword matcher and Soil Companion.

#### v0.2

The knowledge graph content was updated and extended.

#### v0.1

To provide additional (semantic) functionality for technical components and end users, SWC offers access to a knowledge graph (Wang et al., 2025) exposed through a SPARQL endpoint. This allows reasoning over

relationships between concepts that are part of the SWC (meta)data and connected vocabularies and ontologies.

The current iteration implements an RDF representation of the SWC harvested metadata in the deployed Virtuoso Triple Store, accessible through the SPARQL endpoint.

Besides, it is used for experiments to extend the knowledge graph with external vocabularies and ontologies and connect these with the SWC metadata. In the upcoming iterations this will result in a linked data graph that provides integration of soil health and related domain knowledge.

### Soil Vocabulary Viewer

#### v0.1.0

SoilVoc (SoilWise Vocabulary Browser) is a SKOS-based soil science thesaurus designed to connect soil knowledge with soil data. SoilVoc links soil properties with the procedures underlying those properties, and by enforcing a rigorous, transparent hierarchy that enables controllable navigation from abstract concepts to concrete properties and their procedures.

### Repository Storage

#### Apache Lucene index, release 9.11.1

The SWC newly uses a dedicated index (Apache Lucene) to efficiently index and store the harvested and augmented metadata, as well as the knowledge extracted from documents referred to through the metadata records (currently only supporting PDF format). Access to the index (both indexing and querying) is provided through the Apache Solr search framework.

#### v0.2

No changes

#### v0.1

The following storage components have been deployed on the SWC v1:

- A (**PostgreSQL v12.2**) relational database management system for the storage of the core metadata of both data and knowledge assets,
- A (**Virtuoso v07.20.3239**) triple store to store the metadata of data and knowledge assets as a graph, linked to soil health and related knowledge as a linked data graph,
- **GitHub** for storage of user-enhanced metadata.

Storage of selected soil data, metadata, transformed data etc. into the repository will be covered in v2 once the requirements become clearer after validation of SWC v1.

### Data and Knowledge Administration Console

#### v1.0.1

The newly added Data and Knowledge Administration Console is composed of several dashboard pages implemented using Apache Superset technology. Apache Superset is an open-source data exploration and visualization platform that enables the creation of interactive, web-based dashboards with minimal coding. In SoilWise, it is used to present statistics from the Metadata Catalogue, results of the Metadata Validation, and to provide more insights into the results published by Mission Soil projects. The console is available only for authorised users.

### Authentication & Authorisation

User and organisation management, authentication, and authorisation are handled through an [Keycloak](#) instance operated by wetransform, serving as the central OpenID Connect (OIDC) identity provider for SWC components:

- Keycloak is accessible at `id.soilwise.wetransform.eu`; the admin console is available at `id-admin.soilwise.wetransform.eu`
- User self-registration is disabled; accounts are provisioned by administrators via superuser registration (activation link) or automatically via a trusted authentication token
- Keycloak maps provider roles to internal application roles (e.g. Keycloak role `Redakteur` → `hale»connect` role `themeManager`) and manages user-to-organisation assignments
- The internal role model follows a five-tier structure: `anonymous`, `basicUser`, `dataManager`, `themeManager`, and `orgAdmin`
- Supported authentication mechanisms include OIDC, OAuth, SAML 2.0, and Active Directory via identity brokering
- Integration with EU Login (European Commission authentication service) is planned for a future iteration

### System Usage & Monitoring

All SWC components and services are monitored using a central Prometheus + Grafana stack deployed to the SoilWise Kubernetes cluster:

- Prometheus (v2.55.1) scrapes metrics from all services, the Kubernetes cluster, and host nodes via Node Exporter (v1.9.1) and kube-state-metrics
- Prometheus LTS (v2.18.1) retains metrics for longer periods (~30 days / 4 GB) to support trend analysis and availability reporting
- Grafana (v12.1.1) provides provisioned dashboards covering Kubernetes cluster health, service performance and request metrics, container resource consumption, log analysis, AWS infrastructure, and long-term OWS availability statistics
- Loki (v3.4.1) with Promtail (v3.0.0) handles log aggregation and structured log querying, including filtering by HTTP status code to identify 4xx/5xx errors
- Website-level usage of the SoilWise public website is tracked via Hotjar and Google Analytics, managed separately at the website/CMS level

## 6 Conclusion and outlook

This deliverable concludes the second of three SoilWise development cycles and marks the transition of the SWC from a distributed prototype operated across partner environments to a unified, GitOps-managed deployment hosted on a Kubernetes cluster at wetransform. By the end of cycle 2, the SWC is operating in the unified, GitOps-managed infrastructure: source-code releases for all delivered components are tagged on GitHub and archived on Zenodo, the discovery and processing APIs (CSW, OGC API – Records, STAC, OAI-PMH, OpenSearch, SPARQL, and the Solr Search API) are publicly accessible, and the user-facing interfaces — SoilWise Finder, Soil Companion, Tabular Soil Data Annotation, SoilVoc, and the Data & Knowledge Administration Console — are deployed and available to stakeholders. The corresponding component-level developments are reported in deliverables D2.3 and D3.3.

The third and final development cycle (M32–M47) is now in progress and is organised as a consolidation phase rather than an exploration phase: the work is structured around thirteen component workstreams (Harvester, Repository Storage, Knowledge Graphs, Metadata Validation, Metadata Augmentation, Data & Knowledge Publication Support Tools, SoilWise Catalogue, Soil Companion, System Usage & Monitoring, Dissemination & Guidelines, Access Control, EUSO Integration, and Metadata Authoring), each with a designated lead, an initial team, and a target completion date. Within each workstream, the focus for cycle 3 is on activities driven by Grant Agreement commitments, JRC requirements, and User Case acceptance criteria; activity inventories per workstream and their delivery timelines are tracked in the project backlog and are not reproduced here.

The integration with the European Soil Observatory (IT2-4) is one of the strategic priorities for cycle 3. As outlined in Section 3.3, EUSO Integration is being delivered as a dedicated workstream with target completion towards project end; an initial demonstration and development installation at JRC is planned in coordination with JRC's hosting setup. Open items to be resolved with JRC include the mapping from technical components to the logical/functional components used in EUSO planning, clarification of optional and proprietary components and their functional impact, and definition of EUSO-specific user roles and onboarding workflows. The closing state of the SWC infrastructure, the outcome of the EUSO integration work, and the consolidated set of deliveries available for hand-over will be reported in **D4.4 Repository infrastructure, components and APIs, v4** (M47), the final deliverable in this series.

## References

Bouman, J., & Veerman, C. P. (2022). Developing Management Practices in: “Living Labs” That Result in Healthy Soils for the Future, Contributing to Sustainable Development. *Land*, 11(12), 2178.

Wang, B., De Sousa, L. M., & Fensel, A. (2025, December). Make soil healthy again: Construction of ontology-compliant soil health knowledge graph with large language models. In *Proceedings of the 13th Knowledge Capture Conference 2025* (pp. 56-60).