M13/SEPTEMBER 2024

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

D 4.1 Repository infrastructure, components and APIs, v1

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-23
Project Website	www.soilwise-he.eu
Deliverable Title	D 4.1 Repository infrastructure, components and APIs, v1
Delivery Time (DOA)	M13
Deliverable Submission Date	30/09/2024
Status	V1
Dissemination Level	PU - Public
Deliverable Lead	Thorsten Reitz (WE)
Author(s)/Organisation(s)	Somakanthan Somalingam (WE), Thorsten Reitz (WE), Dajana Snopková (MU), Tomáš Řezník (MU), Paul van Genuchten (ISRIC), Nick Berkvens (EV ILVO)
Contributor(s)	WE, MU, WU, ZALF, EV ILVO, DOMG – VL O, CREA, ISRIC, NP
Peer-Reviewers	Rob Lokers (WR), Anna Fensel (WU)
Contact	tr@wetransform.to, so@wetransform.to
Work Package	WP4
Dissemination level	Public
Keywords	SWR functionalities, technical components, implementation, source code, data management components, prototype, technical documentation
Abstract	The purpose of this deliverable is to set up the first instance of the SWR infrastructure v1 and to provide the source code in Cycle 1 at the end of the development phase. In addition, potential users are provided with instructions and corresponding tutorials and documentation.

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
DM	Data Management
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
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
SWR	SoilWise Repository
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
ZALF	LEIBNIZ-ZENTRUM FUER AGRARLANDSCHAFTSFORSCHUNG

Table of Contents

List of Tables and Figures

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, harmonised 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, automatise 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 EUSO evolvement accordingly**. The SoilWise repository 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 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 SWR v1,
- **Chapter 2** provides a list of the technical components implemented/deployed in SWR v1 and links to source code,
- **Chapter 3** evaluates the fulfilment of the system requirements of the first delivered prototype according to the vision scenarios,
- **Chapter 3** details the technical documentation,
- **Chapter 4** contains the release notes.

Technical documentation is available at https://prototype-1-0.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.

1.3 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, v2, v3, v4 (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, v2, v3, v4 (M13, M18, 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)

1.4 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)

1.5 Scope of D 4.1: Repository infrastructure, components and APIs, v1

The deliverable D4.1 consists of the first instance of the developed/implemented SWR infrastructure and the code in Cycle 1 at the end of the development phase. In Cycle 1, the core functionalities of the SWR (11 functionalities) were defined using use cases based on stakeholder involvement and an architecture was derived from this. Technical components were then identified from the functionalities and architecture and used to build the SWR infrastructure by providing suitable APIs and technical components.

Note: In Cycle 1, components exist in distributed infrastructure environments at different locations. Nevertheless, the core SWR functionalities can be accessed and demonstrated in v1. Overall integration into a single SWR infrastructure environment will be realised in v2/3.

The derived architecture of v1 is shown in Fig. 1 and can be accessed via this URL: https://prototype-1-0.soilwise-architecture.pages.dev/?view=id-e3ae52bba4fb42dfa0b3900e7d3.

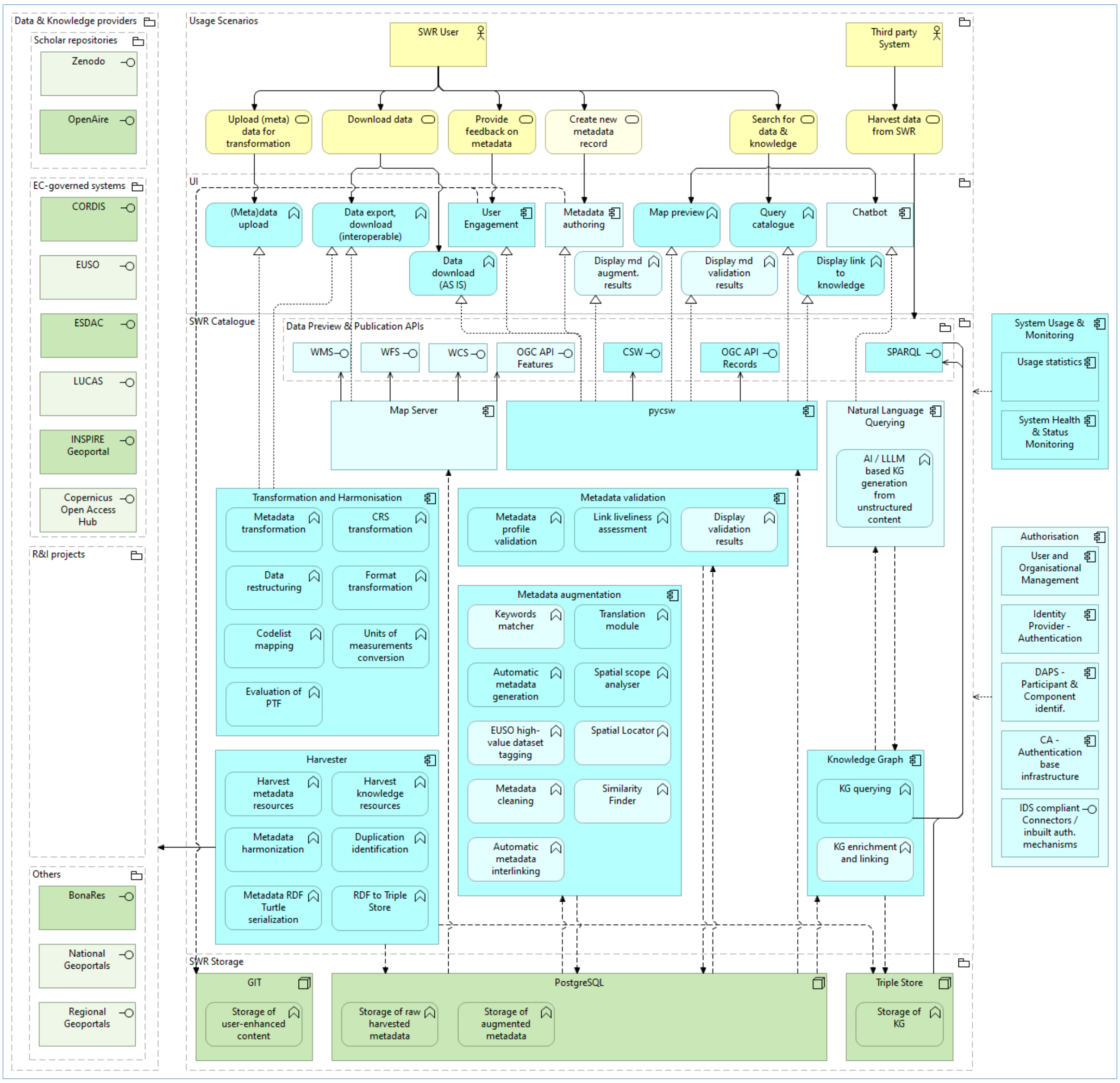

Figure 1 SWR architecture of the first prototype

2 Technical components implemented/deployed in the first SWR prototype

In this section, a list of the implemented/deployed technical components is given together with the functionalities fulfilled, links to the source codes and to guidelines and tutorials.

Table 1 List of components of the first delivered prototype and their source code

Component	Function / API	Source code	Tutorials/ Documentation
Harvester	Harvest metadata resources, v1 Harvest knowledge resources, v1 Metadata harmonization, v1 Metadata RDF Turtle serialization, v1 RDF to Triple store, v1 Duplication identification, v1	https://github.com/soilwise-he/harvesters/releases/tag/v0.1.0	https://prototype-1-0.soilwise-documentation.pages.dev/technical_components/ingestion/
Catalogue	Query Catalogue, v1 CSW API, v1 OGC API Catalogue, v1 Data download (AS IS), v1 Display metadata augmentation results, v1 Map preview, v1 User Engagement, v1	https://github.com/soilwise-he/pycsw/releases/tag/v0.1.0	https://prototype-1-0.soilwise-documentation.pages.dev/technical_components/catalogue/
Metadata validation	Metadata profile validation, v1	https://github.com/soilwise-he/metadata-validator/releases/tag/v0.1.0/	https://www.youtube.com/watch?v=Pxl5hI3FxJg&list=PLoyBfgUelhNOwA_GGkd4hSwDnwNhxGC87&index=1&t=1461s
	Link liveliness assessment, v1	https://github.com/soilwise-he/link-liveliness-assessment/releases/tag/v0.1.0	
Transformation and Harmonization (hale»studio 5.3)	Manual metadata upload Metadata transformation CRS transformation Data restructuring Format transformation	https://github.com/halestudio/hale/releases/tag/v5.3.0	https://www.youtube.com/playlist?list=PLoyBfgUelhNOwA_GGkd4hSwDnwNhxGC87

	Codelist mapping		
	Units of measurements conversion		
	Download interoperable metadata		
Metadata Augmentation	Automatic metadata generation	https://github.com/soilwise-he/metadata-validator/releases/tag/v0.1.0/	https://prototype-1-0.soilwise-documentation.pages.dev/technical_components/metadata_augmentation/
	Translation module	https://github.com/soilwise-he/metadata-augmentation/releases/tag/v0.1.0	
	Spatial scope analyser		
Knowledge Graph	Knowledge Graph enrichment and linking	https://github.com/soilwise-he/soil-health-knowledge-graph/releases/tag/v0.1.0	https://prototype-1-0.soilwise-documentation.pages.dev/technical_components/knowledge_graph/
	Knowledge Graph querying (SPARQL endpoint)		
Repository Storage	Storage of user-enhanced content - Git	Not applicable	https://prototype-1-0.soilwise-documentation.pages.dev/technical_components/storage/
	Storage of raw harvested metadata – PostgreSQL		
	Storage of augmented metadata – PostgreSQL		
	Storage of augmented, linked metadata, knowledge graph - Triple Store		

3 Fulfilment of the system requirements

The SoilWise core product and its functionality delivered in the first prototype were driven by the **Vision scenarios** (presented in chapter 2.5 of deliverable D1.3 Repository architecture). Originally, they were covered by 10 functionality points, but eventually, the eleventh point was added to the list. These 11 functionality points were further specified through the definition of acceptance criteria (the so-called **Definition of done, DoD**). The following table represents a means to evaluate the first delivered prototype through an indication of fulfilled acceptance criteria.

For the second and third iterations, the development will be driven by the Product Backlog comprising the user stories, requirements, acceptance criteria, and functionalities, and the evaluation of delivered prototypes will be covered by D5.4 Deployment and Evaluation Report, v2 and D5.5 Deployment and Evaluation Report, v3. D5.3 Deployment and Evaluation Report, v1 will evaluate the validated first prototype, to be delivered in M13.

The following table describes the acceptance criteria derived from the 11 functionalities and indicates their status in relation to the DoDs.

1. The SWR Catalogue will be powered by a GIT repository, relational database and triple store containing standardised metadata of external data/knowledge resources. Standardised means a unified structure despite various input metadata structures from the underlying repositories, such as CORDIS, INSPIRE Geoportal, Zenodo, BonaRes, OpenAire, etc.

- **Catalogue**
  - ✓ Visibility: Users can see a list of all available documents.
  - ✓ Accessibility: The catalogue is accessible via a clear and intuitive menu on the homepage.
  - ✓ Sorting: Users can sort the catalogue by various criteria (e.g., date, relevance, title).
  - ✓ Filtering: Users can filter the catalogue based on predefined categories (e.g., source, date range, document type).
  - ✓ Pagination: The catalogue supports pagination for easy navigation through large datasets.
- **Dashboard**
- **Search Function**
  - ✓ Relevance: The search function returns relevant results based on user queries.
  - ✓ Speed: Search results are displayed within a few seconds.
  - ✓ Filters: Users can apply filters to refine search results (e.g., by date, document type, source).
  - ✗ Advanced Search: Advanced search options are available for more precise queries (e.g., Boolean operators, exact phrases).
  - ✓ Result Sorting: Users can sort search results by relevance, date, or title.
- **Metadata Store**
  - ✓ Central Repository: All metadata is stored in a centralized repository.
  - ✓ Scalability: The metadata store can handle a large volume of metadata entries.
  - ✓ Accessibility: Metadata is accessible via the search function and the catalogue.
  - ✓ Integrity: Metadata entries are accurate and consistently formatted.
  - ✓ Security: The metadata store is secure from unauthorized access.
- **Standardized Metadata (Unified Structure)**
  - ✓ Consistency: All metadata follows a standardized structure.
  - ✓ Interoperability: Metadata is interoperable with other systems and standards.
  - ✓ Validation: There are validation rules in place to ensure metadata conforms to the standardized structure.
  - ✓ Documentation: Clear documentation is provided for the metadata structure.
  - ✓ Ease of Use: The standardized metadata is easy to use and implement by developers.
- **Metadata of Datasets**
  - ✓ Completeness: Metadata for datasets includes all necessary information (e.g., title, description, date, source).
  - ✓ Accuracy: Metadata entries accurately reflect the content and details of the datasets.
  - ✓ Searchability: Dataset metadata is searchable through the platform's search function.
  - ✓ Linkage: Metadata entries link to the actual datasets for easy access.
  - ✓ Updates: Metadata is updated regularly to reflect any changes in the datasets.
- **Metadata of Knowledge**
  - ✓ Relevance: Metadata captures key information about knowledge documents (e.g., articles, reports).
  - ✓ Comprehensiveness: All essential details are included in the metadata (e.g., author, publication date, keywords).
  - ✓ Accessibility: Knowledge metadata is accessible and searchable via the platform.
  - ✓ Categorization: Metadata entries are properly categorized for easy filtering and retrieval.
  - ✓ Regular Updates: Metadata is kept up to date with the latest knowledge documents.
- **Data from EUSO**
  - ✓ Integration: Data from EUSO is successfully integrated into the central database.
  - ✓ Accuracy: EUSO data is accurately reflected in the platform.
  - ✓ Searchability: EUSO data is searchable via the platform's search function.
  - ✓ Metadata: EUSO data includes standardized metadata entries.

- ✓ Access: Users can easily access and retrieve EUSO data through the platform.
- **Data from INSPIRE Geoportal**
  - ✓ Integration: Data from INSPIRE Geoportal is integrated into the central database.
  - ✓ Accuracy: INSPIRE Geoportal data is accurately captured.
  - ✓ Searchability: Users can search for INSPIRE Geoportal data using the platform's search function.
  - ✓ Metadata: Standardized metadata is provided for all INSPIRE Geoportal data entries.
  - ✓ User Access: Users can access INSPIRE Geoportal data easily through the platform.
- **Data from Zenodo**
  - ✓ Integration: Data from Zenodo is seamlessly integrated into the central database.
  - ✓ Accuracy: Zenodo data is accurately represented in the platform.
  - ✓ Searchability: Zenodo data is searchable via the platform's search function.
  - ✓ Metadata: Zenodo data includes standardized metadata entries.
  - ✓ User Access: Users can access Zenodo data easily through the platform.
- **Data from Cordis**
  - ✓ Integration: Cordis data is integrated into the central database.
  - ✓ Accuracy: Cordis data is accurately reflected in the platform.
  - ✓ Searchability: Cordis data is searchable via the platform's search function.
  - ✓ Metadata: Standardized metadata is provided for all Cordis data entries.
  - ✓ User Access: Users can easily access and retrieve Cordis data through the platform.
- **Data from BonaRes**
  - ✓ Integration: BonaRes data is successfully integrated into the central database.
  - ✓ Accuracy: BonaRes data is accurately represented.
  - ✓ Searchability: BonaRes data is searchable using the platform's search function.
  - ✓ Metadata: BonaRes data includes standardized metadata entries.
  - ✓ User Access: Users can easily access and retrieve BonaRes data through the platform.
- **Data from OpenAire**
  - ✓ Integration: OpenAire data is seamlessly integrated into the central database.
  - ✓ Accuracy: OpenAire data is accurately reflected in the platform.
  - ✓ Searchability: OpenAire data is searchable via the platform's search function.
  - ✓ Metadata: Standardized metadata is provided for all OpenAire data entries.
  - ✓ User Access: Users can access OpenAire data easily through the platform.

2. The Triple store will be extended with links to knowledge assets from external resources, for the 1st iteration extracted from CORDIS. To a user, the knowledge will be made available, among others, based on interlinked metadata, i.e., metadata linked to relevant projects and project deliverables.

- **Triple Store Hosting Knowledge Graph (KG)**
  - ✓ The triple store must be set up and running on the designated server.
  - ✓ It should support SPARQL queries and provide an endpoint accessible to authorized users.
  - ✓ It must store and manage triples effectively, ensuring data integrity and consistency.
  - ✗ The performance of the triple store should be tested to handle the expected volume of data and queries without significant lag.
- **KG Structure Implementing the Chosen SWR Metadata Schema**
  - ✓ The KG must correctly implement the chosen SWR metadata schema.

✓ All required entities, properties, and relationships defined in the metadata schema should be present and correctly structured.

✗The KG should pass validation tests against the schema to ensure compliance.

✓Sample data should be inserted and queried to demonstrate correct implementation.

- **KG Structure Implementing Linkages Between Resources**

  ✓The KG should include defined linkages between various resources as specified.

  ✓These linkages must be navigable through SPARQL queries, demonstrating connections between resources.

  ✗The linkages should be tested to ensure they reflect the relationships accurately and consistently.

  ✗Visualization tools should display these linkages clearly to users.

- **CORDIS Resources Metadata + Links to Knowledge Integrated in KG (link to harvester component)**

  ✓CORDIS metadata should be harvested and integrated into the KG.

  ✓All relevant metadata fields must be accurately mapped and stored in the KG.

  ✓ Links to knowledge (e.g., project deliverables) should be established and verified.

  ✗Queries and visualizations should correctly reflect the integration of CORDIS metadata.

- **Explicit Links from CORDIS to Zenodo Integrated in KG (link to harvester component)**

  ✓Explicit links between CORDIS resources and Zenodo should be established within the KG.

  ✓These links should be navigable and correctly represented in the KG.

  ✗Test cases should confirm that users can traverse from CORDIS resources to corresponding Zenodo entries.

  ✗The linkage data should be visualized correctly in the UI.

- **Zenodo Resources Metadata + Links to Data and/or Knowledge Integrated in KG (link to harvester component)**

  ✓ Zenodo metadata should be harvested and integrated into the KG.

  ✓ All relevant metadata fields must be accurately mapped and stored in the KG.

  ✓ Links to associated data and/or knowledge should be established and verified.

  ✗Queries and visualizations should correctly reflect the integration of Zenodo metadata.

- **Bonares Resources Metadata + Links to Knowledge Integrated in KG (link to harvester component)**

  ✓ Bonares metadata should be harvested and integrated into the KG.

  ✓ All relevant metadata fields must be accurately mapped and stored in the KG.

  ✗Links to knowledge resources should be established and verified.

  ✗Queries and visualizations should correctly reflect the integration of Bonares metadata.

- **Integrated UI Visualization of Data and Knowledge Resources (link to UI component)**

  ✓ The UI should integrate visualizations for all data and knowledge resources within the KG.

  ✓ Visualizations must be clear, interactive, and user-friendly.

  ✓ Users should be able to filter, search, and navigate the data and knowledge resources easily.

  ✗Performance tests should ensure that the UI loads and responds quickly even with large datasets.

- **UI Visualization of (Explicitly) Linked Data and Knowledge (link to UI component)**

  ✗The UI should clearly visualize explicitly linked data and knowledge.

  ✗Users should be able to see the relationships and linkages between different resources.

  ✗The UI must provide intuitive navigation through these links.

  ✗Tests should confirm that the visualized links correspond accurately to the data in the KG.

3. When providing the search results, the SWR Catalogue will display a dataset or knowledge source to a user and show in which external repositories are, e.g., such a dataset available. The SWR also provides metadata, data & knowledge validation, among others, checks automatically persistence of available data/knowledge (e.g. if a dataset uploaded into Zenodo still exists or if a Web service endpoint is still alive and providing data).

- **Display a Dataset**
  - ✓Users can search for datasets using keywords, filters (e.g., date range, author), and advanced search options.
  - ✓Search results for datasets are displayed within 2 seconds for typical queries.
  - ✓Datasets are displayed in a user-friendly format with information about title, source, publication date, and summary.
  - ✓The user shall be able to view detailed information of the dataset by clicking on the search result, which opens a detailed view page.
  - ✓Users can sort datasets by columns (e.g., date, title).
  - ✓Datasets with more than 50 entries are paginated.
  - ✓Users can navigate between pages using 'Next', 'Previous', and direct page number buttons.
- **Display a Knowledge Source**
  - ✓Users can search for knowledge sources using keywords, filters (e.g., publication year, author), and advanced search options.
  - ✓Knowledge source search results are displayed within 2 seconds for typical queries.
  - ✓Each knowledge source is displayed with metadata including title, author, publication date, abstract, and source URL.
  - ✓Users can view the full text of the knowledge source directly or via a provided link.
  - ✓Knowledge sources are categorized (e.g., articles, books, reports).
  - ✓Users can filter search results by category.
- **Show Originating External Repositories**
  - ✓Each document in the search results displays the originating external repository.
  - ✗Users can view a list of all external repositories included in the search.
  - ✗Users can click on the repository name to view more details (e.g., repository description, URL).
  - ✓The user shall be able to filter search results by external repositories.
- **Display Linked Items**
  - ✓Each document displays related documents or references as linked items.
  - ✓Linked items are displayed as clickable links.
  - ✓Clicking a linked item shows its detailed metadata and a content preview.
- **Display Duplicities**
  - ✗The system automatically identifies and flags duplicate documents in the search results.
  - ✗Users are notified of duplicate documents and can choose to view or ignore duplicates.
  - ✗The system shall provide an option to merge duplicate entries and keep the most relevant information.
- **Display Broken Links**
  - ✓ The system automatically checks for broken links during data ingestion and periodically thereafter.
  - ✗Users are notified of broken links with a clear message and the option to report or ignore the issue.
  - ✗Broken links are highlighted in search results, and users can view a list of all broken links.
- **Data Quality Assurance**
  - ✗The system performs automatic checks to ensure data integrity (e.g., correct formats, valid values).
  - ✗Each document displays a quality score or indicator based on predefined quality metrics (e.g., completeness, accuracy).
- **Requirement: Data Completeness (only internal use, no reporting!!)**
  - ✗The system shall ensure that all mandatory metadata fields (e.g., title, author, publication date) are present for each dataset and knowledge source.
  - ✗Internal administrators/actors shall be able to view a completeness score or indicator for each search result.
  - ✓The system shall allow users to submit feedback or report incomplete data for improvement.
- **Requirement: Checks Automatically Persistence of Available Data/Knowledge**
  - ✗The system shall periodically verify the availability of datasets and knowledge sources in their originating repositories.
  - ✗The user shall be notified if a dataset or knowledge source is no longer available, with the option to remove it from the search results.
  - ✗ The system shall log and report the persistence status of each dataset and knowledge source, providing a history of availability checks.

4. The SWR Catalogue will display a map preview of a resource (dataset/knowledge/service/…) from the source graphic/WMS/…, if applicable.

- **Map Preview Dataset**
    - ✗ Users can search for datasets that have associated map previews.
    - ✗ The search results include an indicator showing the availability of a map preview.
    - ✓ When a user selects a dataset with a map preview, a map is displayed alongside the dataset details.
    - ✗ The map preview accurately represents the geospatial data associated with the dataset.
    - ✓ Users can zoom in and out, pan, and interact with the map preview.
    - ✓ Map layers can be toggled on and off for better data visualization.
    - ✗ The map preview loads within 3 seconds after the dataset is selected.
- **Map Preview Knowledge**
    - ✗ Users can search for knowledge sources that include a map preview.
    - ✗ The search results indicate the presence of a map preview.
    - ✓ When a knowledge source with a map preview is selected, a map is displayed alongside the knowledge source details.
    - ✗ The map accurately reflects the geospatial data referenced in the knowledge source.
    - ✓ Users can interact with the map (zoom, pan, toggle layers).
    - ✓ The map interface is intuitive and easy to use.
    - ✗ The map preview loads within 3 seconds after the knowledge source is selected.
- **Map Preview Service**
    - ✗ Users can search for services that offer a map preview.
    - ✗ The search results clearly indicate the availability of a map preview.
    - ✓ When a service with a map preview is selected, a map is displayed alongside the service details.
    - ✗ The map accurately represents the geospatial data or coverage area of the service.
    - ✓ Users can interact with the map (zoom, pan, toggle layers).
    - ✓ The map interface provides tools for better understanding the service area (e.g., markers, polygons).
    - ✗ The map preview loads within 3 seconds after the service is selected.
- **Available from the Source Graphic**
    - ✗ The platform supports integration with graphic sources (e.g., SVG, PNG) for map previews.
    - ✗ Users can view map previews generated from source graphics accurately.
    - ✗ Map previews from source graphics are clear and of high quality, without distortion.
    - ✗ The graphic maintains its resolution and detail when zoomed in or out.
    - ✗ The map preview from the source graphic loads within 3 seconds.
- **Available from the Source WMS (Web Map Service)**
    - ✓ The platform supports integration with WMS for map previews.
    - ✓ Users can view real-time map previews provided by WMS.
    - ✓ The WMS-based map preview includes standard functionalities such as layer selection, zooming, and panning.
    - ✗ Users can access additional WMS features such as querying map layers for more information.
    - ✗ The map preview from the WMS loads within 3 seconds.
    - ✗ Real-time data from the WMS is updated promptly to reflect the latest information

5. Thanks to a Interlinker component, that will be powered by a Triple store, the SWR will identify duplicates based on metadata.

- **Define Strategy for Identification, Processing, and Storing of Duplicates for Iteration 1**
  - ✘Define a clear algorithm or methodology for identifying duplicates based on metadata attributes (e.g., title, author, publication date).
  - ✘Ensure the strategy accounts for variations in metadata across different document types and repositories.
  - ✘Outline the workflow for processing identified duplicates, including how to handle conflicting metadata and determining the primary document.
  - ✘Specify whether duplicates should be merged, flagged, or excluded from search results.
  - ✘Define the storage mechanism for duplicates, ensuring efficient retrieval and management within the central database.
- **Adapt KG Structure to Support Duplicities (Link with Knowledge Graph)**
  - ✘Modify the knowledge graph (KG) schema to accommodate duplicate relationships between documents.
  - ✘Define how duplicate relationships will be represented within the KG (e.g., as edges linking duplicate nodes).
  - ✘Ensure KG queries can retrieve duplicate-related information, allowing users to explore connections between duplicate documents.
  - ✘Test KG queries to verify accurate retrieval of duplicate metadata and relationships.
- **Processing in Metadata Store**
  - ✘Ensure metadata extraction is accurate and robust across different document formats and languages.
  - ✘Enrich metadata with additional attributes that facilitate duplicate identification (e.g., normalized titles, standardized author names).
  - ✘Identify duplicates using the processing capabilities of the SWR metadata store.
- **Identify Duplicates Based on Metadata**
  - ✘Implement a duplicate detection algorithm based on metadata similarity metrics (e.g., Jaccard similarity, Levenshtein distance).
  - ✘Test the algorithm's performance on a diverse dataset to evaluate its accuracy and efficiency.
  - ✘Define thresholds for similarity scores or metadata attributes to classify documents as duplicates.
  - ✘Adjust thresholds based on the desired balance between precision and recall in duplicate identification.
- **Visualization of Duplicates in User Interface (Link with UI Component)**
  - ✘Design intuitive visualizations within the user interface (UI) to represent duplicate relationships.
  - ✘Ensure visualizations are accessible and informative for users of varying expertise levels.
  - ✘Implement interactive features that allow users to explore duplicate relationships (e.g., clicking on a document to view its duplicates).

6. SoilWise Repository will provide data download in two modes, “as is” or in an “interoperable way”. “As is” means the SWR is a broker connecting a user to a relevant data source. The “interoperable way” means the SWR is a mediator that converts external data into, e.g. INSPIRE compliant through an ETL-like tool, e.g. HALE Studio.

- **Data downloaded "as is": SWR is a broker connecting user to a relevant data source**
  - ✓Users can search for documents using various filters (e.g., keywords, date, source).
  - ✓The search results display metadata about each document (e.g., title, source, date of publication).
  - ✓Users can initiate a direct download of the selected document from the original data source.
  - ✓The platform provides a link or button for downloading the document "as is".
  - ✓The download link/button redirects users to the original data source or initiates the download directly.
  - ✓The downloaded document retains its original format and metadata.
  - ✓No modifications are made to the content of the document during the download process.
  - ✘Users receive a clear error message if the document cannot be accessed or downloaded from the original source.
  - ✘The platform logs such errors for troubleshooting and reporting purposes.

✓Users can provide feedback on the download process and report any issues.

✓The feedback mechanism is accessible and easy to use.

- **Data downloaded in an "interoperable way": SWR is a mediator converting external data into, e.g. INSPIRE, compliant data through an ETL-like tool**

✗Users can search for documents using various filters (e.g., keywords, date, source).

✗The search results display metadata about each document (e.g., title, source, date of publication).

✓The platform uses an ETL tool (e.g., HALE) to convert external data into INSPIRE compliant format.

✓An alignment-configuration can be stored for each distribution in a dataset metadata

✓Users can contribute an alignment-configuration

✓A transformation process can process the alignment-configuration to produce an interoperable dataset

✓Users can choose to download the interoperable dataset

✓The INSPIRE compliant data retains all relevant information from the original document.

✓The transformation process does not introduce errors or omissions.

✓The INSPIRE compliant data includes all necessary metadata as per the INSPIRE specifications.

✓Users can view and download this metadata along with the document.

✓Users receive a clear error message if the transformation process fails or if the document cannot be downloaded.

✓The platform logs such errors for troubleshooting and reporting purposes.

✓ Users can provide feedback on the transformation and download process and report any issues.

✓ The feedback mechanism is accessible and easy to use.

7. SoilWise Repository will enable user to manually upload data for their on-the-fly processing within the SWR. Capacity of the SWR will be significantly limited; however, a demonstration of manual data upload and their processing, e.g. transformation of coordinate systems or measurements units.

- **Manually Upload Ingest Data**

✓Users should be able to access a clearly labeled "Upload Data" button or menu option from the main interface.

✓The upload interface should accept common file formats such as CSV, JSON, XML, and Excel.

✓A progress indicator should be displayed during the upload process to inform users of the status.

✓Users can select and upload files from their local system.

✓The system should validate the uploaded file's format and content before processing. If the file is invalid, an informative error message should be displayed.

✓Uploaded data should be ingested and stored in the central database for further processing.

✓Upon successful upload, the system should display a confirmation message.

✓If the upload fails, the system should provide a clear and actionable error message.

✓Ensure that only authenticated users can upload data.

✓Uploaded files should be scanned for viruses and malware.

✓Provide user documentation or tooltips within the UI to guide users on how to use the upload and transformation features.

✓Include detailed error messages and troubleshooting steps.

- **On-the-Fly Processing Within the SWR: Transformation of Coordinate Systems**

✓Users should have the option to specify the desired coordinate system transformation when uploading data or from a settings menu.

✓An alignment-configuration can be stored for each distribution in a dataset metadata

✓Users can contribute an alignment-configuration

✓A transformation process can process the alignment-configuration to produce a transformed dataset

✓The system should accurately transform coordinates from the source coordinate system to the target coordinate system during the upload process.

✓ Transformed data should be stored correctly in the central database with the appropriate coordinate system metadata.

✓Display a message confirming the successful transformation of coordinate systems after processing.

✓If the transformation fails, the system should provide an informative error message indicating the reason for the failure.

✓The transformation process should be efficient and not significantly delay the upload process.

- **On-the-Fly Processing Within the SWR: Transformation of Measurements Units**

✓Users should have the option to specify the desired measurement units for data transformation when uploading data or from a settings menu.

✓An alignment-configuration can be stored for each distribution in a dataset metadata

✓Users can contribute an alignment-configuration

✓A transformation process can process the alignment-configuration to produce a transformed dataset

✓The system should accurately transform measurement units from the source units to the target units during the upload process.

✓Transformed data should be stored correctly in the central database with the appropriate units metadata.

✓Display a message confirming the successful transformation of measurement units after processing.

✓If the transformation fails, the system should provide an informative error message indicating the reason for the failure.

✓The transformation process should be efficient and not significantly delay the upload process.

8. In the first iteration, the SWR will primarily demonstrate functionality available to a non-registered user. Nevertheless, a lightweight proof-of-concept on functionality available to an authorised user will be present.

- **Primarily functionality available to a non-registered user**
- **A lightweight proof-of-concept demonstrating functionality available to an authorised user**

✓Basic forms for user registration and login, including validation.

✓A minimal dashboard page that an authorized user can access after logging in. This will include: Welcome message, Simple user profile information (e.g., username, email)., Basic navigation to additional user-specific features (even if they are placeholders).

✓Authentication: Implement basic authentication using a secure method (e.g., token-based authentication like JWT or EU login alternative).

✓Authorization: Ensure that certain routes/pages are accessible only to logged-in users.

✗Data Storage: Use a lightweight database (e.g., SQLite) for storing user credentials and profiles.

9. The SoilWise Repository functionality will be offered via open, standardised APIs (e.g. CSW, OGC REST API – Records, WMS, etc.) and a demonstration on how SWR can be used by other Soil Health Mission projects will be performed (e.g. BENCHMARKS and BonaRes already offered their participation).

- **Offer SWR Functionality via Open, Standardised APIs: CSW**

✓API Endpoint Availability: The CSW API endpoint is accessible and returns a successful response for valid requests.

✓Metadata Retrieval: The CSW API allows users to retrieve metadata about documents, including title, abstract, keywords, and publication date.

✓Search Functionality: Users can perform search queries using various parameters (e.g., keyword, date range, author) and receive relevant results.

✓Standard Compliance: The CSW API adheres to the latest OGC CSW specification.

✓Response Formats: The API supports multiple response formats (e.g., XML, JSON) as specified in the standard.

✓Error Handling: The API provides meaningful error messages and handles invalid requests gracefully.

- **Offer SWR Functionality via Open, Standardised APIs: OGC REST API**

✓API Endpoint Availability: The OGC REST API endpoint is accessible and returns a successful response for valid requests.

✓Record Retrieval: The API allows users to retrieve records of soil health documents, including detailed metadata.

✓Search Functionality: Users can search for records using various parameters (e.g., spatial extent, time range, keywords) and receive relevant results.

✓Standard Compliance: The OGC REST API adheres to the latest OGC API - Records specification (https://ogcapi.ogc.org/records/).

✓Response Formats: The API supports responses in JSON format.

✓Pagination Support: The API supports pagination to handle large sets of search results.

✓Error Handling: The API provides clear error messages and handles invalid requests appropriately.

- **Offer SWR Functionality via Open, Standardised APIs: WMS**

✗API Endpoint Availability: The WMS API endpoint is accessible and returns a successful response for valid requests.

✗Map Layers: The WMS API provides access to various map layers related to soil health data.

✗Layer Retrieval: Users can retrieve and display specific layers of soil health data on a map interface.

✗Standard Compliance: The WMS API adheres to the latest OGC WMS specification.

✗Image Formats: The API supports multiple image formats for map rendering (e.g., PNG, JPEG).

✗GetCapabilities: The API supports the GetCapabilities request to provide metadata about available layers.

✗GetMap and GetFeatureInfo: The API supports GetMap and GetFeatureInfo requests to retrieve map images and feature information, respectively.

✗Error Handling: The API provides meaningful error messages and handles invalid requests appropriately.

- **Show How SWR Can Be Used by Other Soil Health Mission Projects**

✗Integration Documentation: Clear documentation is provided, detailing how to integrate SWR functionality into other projects using the provided APIs (CSW, OGC REST, WMS).

✗Use Case Examples: Practical examples and use cases are provided, demonstrating the use of SWR functionality for e.g. BENCHMARKS and ISLANDER projects.

✗API Endpoints: Specific API endpoints are highlighted, showing how they can be used to access and retrieve relevant data.

✗Technical Support: A support mechanism (e.g., email, forum) is available for projects needing assistance with integration.

✗Demonstration: A live or recorded demonstration is available, showcasing the integration process and benefits for Soil Health Mission projects.

✗Feedback Loop: A feedback loop is established to gather input from BENCHMARKS and ISLANDER projects to improve the SWR functionality and its integration process.

10. The SoilWise Repository will be equipped with a Usage monitoring & statistics module, automatically counting e.g. which soil datasets are the most searched for, which datasets are the most downloaded, how many visitors the SWR has etc.

- **A Usage Monitoring & Statistics Module for SWR**

✓The Usage Monitoring & Statistics module is installed and integrated with the SWR.

✗The module can be accessed via an administrative dashboard within the SWR.

✓The module's interface is user-friendly and responsive.

✓The collected data is displayed in a visually appealing format, including charts and graphs.

✓The module allows filtering data by date ranges (e.g., daily, weekly, monthly).

✓The module does not significantly slow down the SWR's performance.

✓The data collection continues to function accurately even during high traffic periods.

✘The module has a mechanism to handle data anomalies and errors gracefully.

✘The module complies with relevant data privacy regulations (e.g., GDPR).

✓Access to the statistics and monitoring module is restricted to authorized personnel only.

- **Automatically Reporting Which Soil Datasets Are Most Searched For**

✓The module logs each search query related to soil datasets.

✓The search logs include details such as timestamp and search frequency.

✓The module generates a report listing the most searched soil datasets.

✘The report is updated in real-time and can be viewed on the administrative dashboard.

✘The collected data is displayed in the UI in a visually appealing format, including charts and graphs.

✓The report can be exported in common formats (e.g., CSV, PDF).

✓The report accurately reflects the most searched datasets without discrepancies.

✓The report highlights the top 10 most searched soil datasets.

✘The module allows administrators to set thresholds or time frames for defining "most searched" (e.g., last week, last month).

- **Automatically Reporting Which Datasets Are Most Downloaded**

✘The module logs each download of datasets.

✘The download logs include details such as dataset name, timestamp, and frequency.

✘The module generates a report listing the most downloaded datasets.

✘The report is updated in real-time and is accessible via the administrative dashboard.

✘The collected data is displayed in the UI in a visually appealing format, including charts and graphs.

✘The report can be exported in common formats (e.g., CSV, PDF).

✘The report accurately reflects the most downloaded datasets without discrepancies.

✘The report highlights the top 10 most downloaded datasets.

✘The module allows administrators to set thresholds or time frames for defining "most downloaded" (e.g., last week, last month).

- **Automatically Reporting How Many Visitors the SWR Has**

✓The module logs each unique visitor to the SWR.

✓The visitor logs include details such as timestamp and visitor count.

✓The module generates a report showing the number of visitors to the SWR.

✘The report is updated in real-time and accessible via the administrative dashboard.

✘The collected data is displayed in the UI in a visually appealing format, including charts and graphs.

✓The report can be exported in common formats (e.g., CSV, PDF).

✓The visitor count report accurately reflects the number of unique visitors.

✓The report includes metrics such as total visitors, new visitors, and returning visitors.

✓The module allows administrators to filter visitor data by time frames (e.g., daily, weekly, monthly).

✓The visitor report includes visual elements such as graphs and charts for easy analysis.

11. The SWR should distinguish high-value data sets from other data in the SWR Catalogue. This requirement is tightly linked to the EU Directive 2019/1024 on open data and the re-use of public sector information (PSI). Neither the PSI directive nor the state-of-the-art practices are capable of

delivering information on whether a data set is considered high-value or not within the scope of the PSI directive. The SoilWise research should, therefore, focus on the identification of (PSI) high-value data sets.

- ✗ **An INSPIRE data set is identified as having coverage of a whole EU/EFTA country or a regional coverage.**
- ✗ **An INSPIRE data set is identified as a high-value data set in line with the scope of the PSI directive.**

4 Technical documentation

The documentation of SoilWise Repository architecture is maintained in the public GitHub repository: https://github.com/soilwise-he/SoilWise-documentation. It comprises description of functionality, detailed technical specifications and interfaces between all modular components and is structured according to the main technical components, linking to the implementation of these in Github. Note that the SWR documentation is also a living environment that is continuously updated during the development process. For this deliverable, a stable release is published at: https://prototype-1-0.soilwise-documentation.pages.dev/, and a PDF exported version is also available on demand as the non-editable version made at the date of the deliverable submission.

5 Release Notes

5.1 Infrastructure

The current infrastructure of SWR v1 is distributed across different, independently operating premises. So far, a Kubernetes cluster has been set up in the wetransform 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 SWR catalogue and the knowledge components. It is planned to integrate these distributed infrastructure components into a single setup in the upcoming cycles of v2/v3.

5.2 Harvester, v0.1.0

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

- Cordis + OpenAire
- BonaRes (Catalogue Service for the Web)
- NSPIRE geoportal (GeoNetwork API)
- ESDAC website
- Copernicus Spatio Temporal Asset Catalogue

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 harmonisation the RDF is ingested into the triple store (Virtuoso). A manual intervention is currently needed to ingest RDF from a tailored API which exposes the RDF from the PostgreSQL database.

5.3 Catalogue, v0.1.0

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.

5.4 Metadata validation

5.4.1 Link liveliness assessment, v0.1.0

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 analysing the type of property of the link (in the OGC API – Records).

5.4.2 Meta data validation using INSPIRE and EJP Soil/EUSO profiles, v0.1.0

In the hale>>connect instance for SWR 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.

5.5 Transformation and Harmonization, v5.3.0

The first technical functions have been set up utilizing Hale>>Connect 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/

5.6 Metadata augmentation, v0.1.0

5.6.1 Automating the creation of metadata using Hale>>connect

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

5.6.2 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.

5.7 Knowledge Graph, v0.1.0

To provide additional (semantic) functionality for technical components and end users, SWR offers access to a knowledge graph, exposed through a SPARQL endpoint. This allows reasoning over relationships between concepts that are part of the SWR (meta)data and connected vocabularies and ontologies.

The current iteration implements an RDF representation of the SWR 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 SWR metadata. In the upcoming iterations this will result in a linked data graph that provides integration of soil health and related domain knowledge.

5.8 Repository Storage, v0.1.0

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

- A (PostgreSQL) relational database management system for the storage of the core metadata of both data and knowledge assets,
- A (Virtuoso) 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,
- Git 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 SWR v1.

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.