Scientific Lake

Deliverable D1.2: Initial integrated system

Due Date of Deliverable	30/06/2024
Actual Submission Date	29/06/2024
Work Package	WP1
Tasks	T1.5
Type	Other
Approval Status	Submitted
Version	1.0
Number of Pages	34

Abstract

SciLake introduces an open Scientific Lake concept, featuring customizable components deployed across a federation of machines to create, interlink, and maintain domain-specific, community-managed Scientific Knowledge Graphs (SKGs). This infrastructure provides a unified method for accessing and querying contained assets, enabling the development of value-added services to enhance knowledge discovery, research reproducibility, and other research-related routines. Domain experts can select and tailor components to their specific needs, making the architecture highly customizable.

Apart from the Scientific Lake concept, SciLake is also developing two indicative, discipline-tailored, value-added services that are showcasing the value of this concept in practice. The first service is designed to enhance the exploration of a specific scientific domain's knowledge space by leveraging the content of relevant SKGs. The second service focuses on enhancing research reproducibility within specific domains by utilising the contents of SKGs to provide insights on the reproducibility of associated research products.

This deliverable report outlines the first version of the integrated SciLake system, its distinctive subsystems consisting of individual components, API interfaces and portals, integrated and deployed at the end of June 2024 (M18) .

This project has received funding from the European Union's Horizon Europe framework programme under grant agreement No. 101058573. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the European Research Executive Agency can be held responsible for them.

Revision history

Version	Date	Reason	Revised by
0.1	05/03/2024	First Draft	Marek Horst
0.2	05/06/2024	Updated draft incorporating input provided by partners	Marek Horst
0.4	25/06/2024	Peer review comments addressed	Marek Horst
1.0	28/06/2024	Final Version after proofreading	Thanasis Vergoulis, Marek Horst

Author List

Organisation	Name	Contact Information
ICM	Marek Horst	mhorst@icm.edu.pl
CNR	Miriam Baglioni	miriam.baglioni@isti.cnr.it
ARC	Thanasis Vergoulis	vergoulis@athenarc.gr

Contributor List

Organisation	Name	Contact Information
SIRIS	César Parra Rojas	cesar.parra@sirisacademic.com
ARC	Sokratis Sofianopoulos	s_sofian@athenarc.gr
TUE	Nick Yakovets	N.Yakovets@tue.nl
ARC	Serafeim Chatzopoulos	schatz@athenarc.gr
ARC	Dimitris Pappas	dpappas@athenarc.gr

Table of Contents

List of Figures

Abbreviation List

Abbreviations	
FoS	Field of Science
GDD	Graph Differential Dependency
GGD	Graph Generating Dependency
HIN	Heterogeneous Information Network
IIS	Information Inference Service
ProGGD	Graph Data Profiling with GGDs
R2PG-DM	Relational to Property Graph Direct Mapping
RAA	Research Artefact Analysis
RI	Research Infrastructure
SKG	Scientific/Scholarly Knowledge Graph
TDM	Text and Data Mining
UI	User Interface

1. Executive Summary

Scientific research aims to deepen our understanding of the world and leverage this knowledge to enhance various aspects of our lives. Scientific insights are instrumental in shaping modern society, driving technological innovations, and stimulating economic growth. Furthermore, scientific knowledge lays the groundwork for future discoveries, allowing scientists to build on the work of their predecessors. However, this knowledge is often fragmented and disorganised, hindering discovery and the extraction of valuable insights for informed decision-making.

Transforming domain knowledge into more structured formats (such as a graph or a relational database) is challenging. Domain experts typically lack the specialised technical expertise required, while knowledge management experts alone cannot properly organise the information without domain-specific insights. To address this problem, SciLake is introducing an open Scientific Lake concept, a suite of customisable components that can be used to create a federated infrastructure capable of facilitating various useful applications for the research community, such as scientific knowledge discovery and research reproducibility.

This report presents the first version of the SciLake ecosystem, its distinctive subsystems consisting of individual components, API interfaces and portals, integrated and deployed at the end of June 2024 (M18). In the following sections, all relevant technical details are elaborated. In Section 2, the required background is introduced and the main objectives of the SciLake ecosystem are outlined. Section 3 elaborates on the design approach followed. In Section 4, technical details of the main systems and components of the respective infrastructure are presented. Section 5 outlines the current version of the integrated system. Finally, in Section 6, we summarise the report and discuss our next steps.

2. Introduction

The primary goal of scientific research is to deepen our understanding of the world and leverage this knowledge to enhance our lives in various aspects. Scientific insights play a crucial role in shaping modern society, propelling technological innovations, and stimulating economic growth. Moreover, scientific knowledge serves as a foundation for future scientific discoveries, enabling scientists to build upon the work of their predecessors.

However, scientific knowledge is fragmented and not always organised in a manner that facilitates discovery and the extraction of insights valuable for informed decision-making. Domain knowledge can be encoded in a variety of formats: while some can be structured and easily searchable, such as information encoded in databases, others are completely

unstructured, such as the textual information found in scientific publications. Compounding the challenge, scientific results are frequently published in different languages, which can further impede the process of uncovering valuable information.

Modern data and knowledge management approaches offer potential solutions to these challenges. For instance, Knowledge Graph technologies can help scientists organise domain knowledge into formats that are easily accessible and queryable. This facilitates the development of value-added applications that can enhance various aspects of daily routines for researchers and other relevant stakeholders. However, transforming domain knowledge into a Scientific Knowledge Graph (SKG) presents significant challenges. Domain experts often lack the specialised technical expertise needed to perform this work independently. In addition, this work cannot be done by knowledge management experts alone since domain knowledge is critical to organise the respective information in a proper way.

To address such challenges, SciLake introduces an open Scientific Lake concept, comprising an array of customizable components that can be deployed and executed on a federation of machines (called SciLake nodes) to facilitate the creation, interlinking, and maintenance of domain-specific, community-managed SKGs and offer a unified method for accessing and querying the contained assets enabling the creation of value-added services to enhance knowledge discovery and other routines important to the research community at large. Domain experts can select to leverage only those components that are valuable for their particular use cases and configure or tailor them accordingly. Moreover, each SciLake node can host different SciLake components according to the needs of the particular use case. Hence, the whole architecture can be considered as a federated and highly-customisable computational infrastructure.

Demonstrating the practical application of this concept, SciLake is also developing two indicative, discipline-tailored, value-added services. The first service, delivered by WP3, is designed to enhance the exploration of a specific scientific domain's knowledge space by leveraging the content of relevant SKGs. These SKGs provide metadata for entities that are essential for describing the domain, as well as valuable information on the relationships between these entities. Additionally, indicators of the impact of related research products, such as publications and datasets, that can also be derived from the contents of the SKGs, will be exploited by the service to provide valuable insights into the significance of various domain-specific entities. The second service, the outcome of WP4, focuses on enhancing research reproducibility within specific domains by utilising the contents of SKGs to calculate reproducibility indicators for associated research products. To generate these indicators, the service will leverage detailed information (enriched in the SKGs) from scientific research outputs, such as the semantics of citations pointing to them. Additionally, the service is

tailored to the unique characteristics of each domain and is equipped to handle text in multiple languages (utilising SciLake's automatic translation component).

3. Design

In this section, we elaborate on the process followed during the design of the integrated system of SciLake, also offering quick insights on the functional requirements of the respective subsystems and components. More details about these requirements for each of the components can be found in Deliverables D2.1 (“Initial version of the Scientific Lake service”), D3.1 (“Initial version of the smart impact-driven discovery service”), and D4.1 (“Initial version of the smart reproducibility assistance service”).

A key objective of the SciLake project was to involve researchers closely in the design and development process from the earliest stages. Within the project, use case representatives serve as domain experts actively engaged in these activities. We placed significant emphasis on presenting early prototypes to these pilot representatives and gathering feedback specific to their use cases. The input from SciLake pilot representatives is also crucial, as the project focuses on providing services tailored to different scientific domains. These representatives bring real-life use cases from four diverse scientific fields: Neuroscience, Cancer Research, Transport Research, and Energy Research. This diversity ensures that the developed services will be designed in a way so that they can be configured and customised to meet the needs of multiple large scientific communities, including those engaged in cross-disciplinary research and collaboration.

Based on the previous, we decided to follow a co-design, agile approach that would rely on early and rapid prototyping and receiving feedback from the pilots of the project on many occasions through flexible experimentation and collaborative activities. Design sketches, static mock-ups, and (when possible) working prototypes were used to test new concepts and provide realistic impressions of the respective functionalities. This process is driving software development efforts offering iterative refinements and establishing a common vision for the project and maintaining a coherent focus at all stages. At later stages, where most of the SciLake components are in a relatively mature phase, it is expected to extend this process to engage researchers outside of the project consortium in an attempt to receive additional feedback.

Moreover, we have designed a series of parallel activities to reinforce the aforementioned approach:

- We have collected early feedback from the pilot representatives to better understand the use cases they bring to the project and their special needs for the components and services to be developed by the project. This activity has been fully documented in Deliverable D1.1 ("Initial service requirements")[1].
- In November 2023, we organised a hands-on workshop in Barcelona, for the pilots of the project to better refine their use cases, receiving also feedback by the technical partners of the project about what can be supported by the various components of the architecture.
- We have drafted an internal document outlining a roadmap of the expected SciLake piloting activities in an attempt to improve the communication between technical partners and pilot representatives and make them all aware of the expected joint activities and their rough timeline. This is a living document, providing general concrete guidance on these activities while continuously refining focused subjects as needed.
- We have formalised a way of collecting information about the respective domain knowledge spaces and SKG data models using living documents (the approach is also elaborated in the roadmap document mentioned in the previous bullet).

The aforementioned approach was the one followed during the previous period helping SciLake members to determine and refine the architecture of the respective integrated system and the design of the various services, subsystems, and components.

4. High-level architecture and components eligible for integration

In this section, we provide a high-level, conceptual architecture for the SciLake ecosystem (Section 4.1) and outline its main components (Sections 4.2-4.4).

4.1. High-level, conceptual architecture

Figure 1 illustrates the high-level, conceptual architecture of the SciLake ecosystem. In brief, SciLake components are organised in different bundles based on the type of services they are offering. Each bundle of tools essentially defines a subsystem of the ecosystem. The subsystems can be organised in two tiers: the foundational Scientific Lake tier, which contains the core components related to the Scientific Lake concept, and the application tier, which contains components that can be combined to offer value-added services to researchers and domain experts leveraging the Scientific Lake contents.

[1] https://zenodo.org/records/8403180

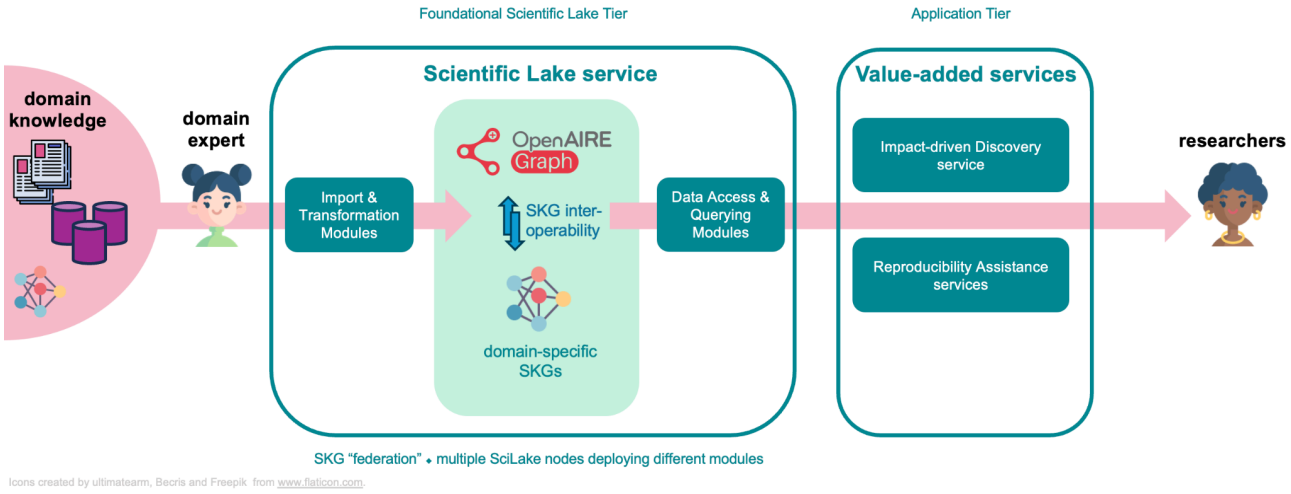


Figure 1: *High-level, conceptual architecture*

The *Scientific Lake service subsystem* implements the basic concept of the Scientific Lake, containing components that assist users in maintaining, updating, and accessing domain-specific and domain-agnostic scientific knowledge. This knowledge can be well-structured (e.g., in the form of Scientific Knowledge Graphs - SKGs) or, even, completely unstructured (e.g., textual data). The subsystem comprises a variety of components including, among others: knowledge graph creation tools, text mining tools, graph mining tools, data transformation tools, graph querying & analytics tools, scientific content acquisition tools, etc. This subsystem is responsible of hosting both all domain-specific SKGs (e.g., those that will be created for each SciLake pilot) and domain-agnostic ones (mainly the OpenAIRE Graph[2]).

The *Impact-driven Discovery service* subsystem aims to leverage scientific impact indicator technologies to facilitate researchers in navigating the vast knowledge space of the respective scientific domains. The subsystem comprises a variety of components including, among others: keyword-based search tools for research products (publications, datasets, etc.), multi-perspective impact-based ranking tools for research products, Fields-of-Science (FoS) classification tools for research products, topic evolution and trend identification tools, and impact propagation technologies. All the components in this bundle make use of the Scientific Lake services to access the required scientific knowledge space.

The *Reproducibility Assistance service* subsystem aims to leverage text and graph mining technologies to assist researchers in making their research more reproducible. To achieve this, the subsystem leverages tools offering a variety of functionalities including, among others: automatic identification of missing links between research products (e.g., publications, datasets, software), classification of links between research products based on their

[2] OpenAIRE Graph: https://graph.openaire.eu

semantics, calculation of the reproducibility level of research works (e.g., offering reproducibility badges or indicators) and so on. All components in this bundle make use of the Scientific Lake services to access the required scientific knowledge space.

It is worth mentioning that some of the components within the presented conceptual architecture for the application-tier subsystems (e.g., the automatic translation component) can also be considered integral parts of the fundamental Scientific Lake service. The current distinction between these components and the core modules was chosen to facilitate the development phase of the SciLake project. However, this categorization is flexible and may evolve in the future to better align with the overall system architecture and functional requirements. The following sections outline the components that have been integrated (or are planned to be integrated) in the SciLake subsystems.

4.2. Scientific Lake service components

In the following paragraphs we outline the components which are eligible for integration in the Scientific Lake service subcomponent.

4.2.1 Knowledge Graph engine

All SKGs are meant to be instantiated by relying on the AvantGraph[3] solution which is the next-generation high-performance graph processing and analytics engine for scientific lake services. AvantGraph is meant to be deployed as a docker image and the full deployment instructions are available at github.com/avantlab/avantgraph.

[3] https://avantgraph.io/

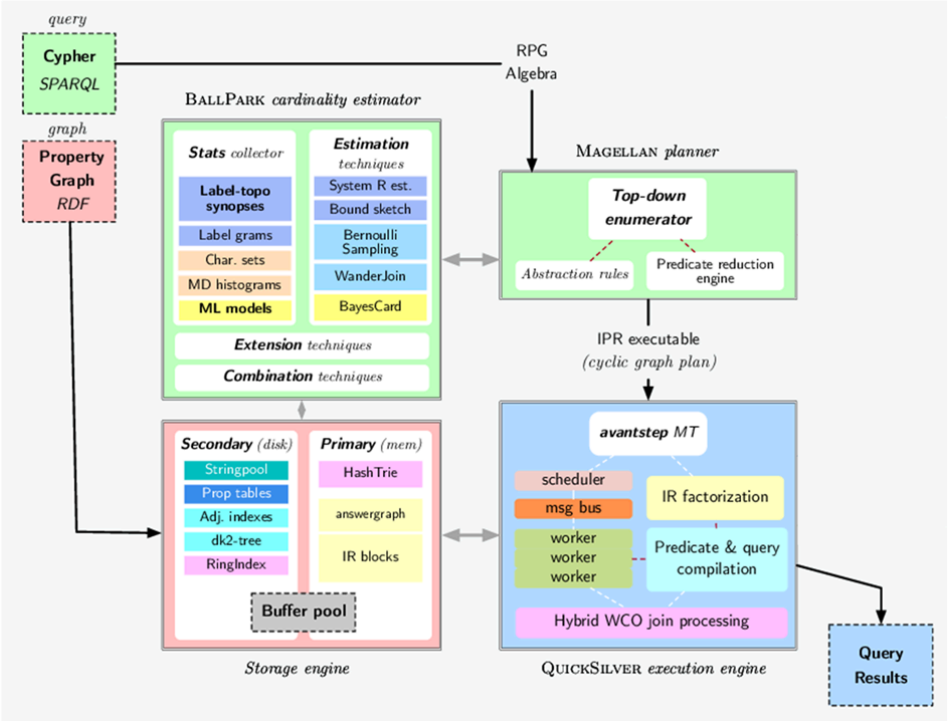


Figure 2: *AvantGraph sub-components*

It supports a wide spectrum of data processing tasks: from subgraph matching to general graph algorithms. AvantGraph allows loading SKGs as neo4j JSON/RDF and querying with Cypher/SPARQL and can be understood as an in-place replacement of neo4j through BOLT[4].

4.2.2 SciLake Catalogue

To aid in graph discovery, a dedicated SciLake Catalogue serves as a central repository for information on all available Scientific Knowledge Graphs (SKGs) and tools. The catalogue is available at scilake-catalogue.d4science.org/. This catalogue acts as a single point of access, fostering data discovery and governance. It achieves this by leveraging rich metadata descriptions for each resource, including details like description, location, provenance, and dependency information, regardless of where the resources are stored or running.

[4] https://neo4j.com/docs/bolt/current/bolt/

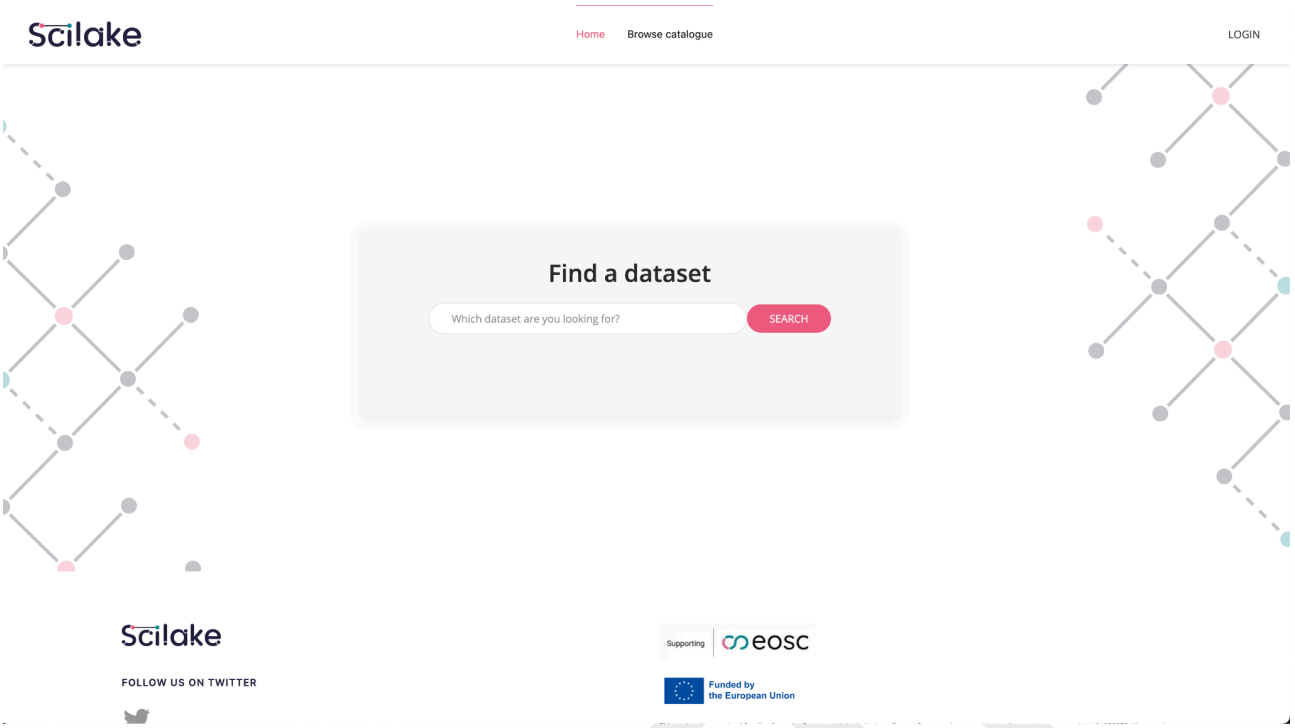


Figure 3: *SciLake Catalogue home page*

The catalogue leverages a two-part architecture: a back-end and a front-end.

- **Back-end:** The back-end infrastructure builds upon the one developed by ARC for the Intelcomp Project[5]. This foundation ensures a robust and proven technical base.
- **Front-end:** The front-end interface is specifically tailored to match the needs of SciLake users. The source code can be currently found in the git repository: code-repo.d4science.org/D-Net/scilake-catalogue-ui.

This combined approach fosters a user-friendly and efficient information resource aligned with the SciLake information space.

[5] https://github.com/IntelCompH2020

4.2.3 Information Inference Service

Information Inference Service (IIS)[6] is a service, developed and maintained by ICM, encapsulating various mining modules including TDM modules based on the Madis framework[7], an extensible relational database system, developed by ARC.

IIS is a flexible data processing system for handling big data based on Apache Hadoop[8] technologies. It uses algorithms to extract new entities and relations from full texts to enrich SKGs. In practice, IIS defines data processing workflows that connect various modules, each one with well-defined inputs and outputs.

4.2.4 PDF Fetcher

This module, developed by ARC, is responsible for managing and optimising the process of obtaining PDF documents associated with publication graph entities. It delivers PDF documents as an input to the IIS where plaintexts are extracted, aggregated into the datasets and sent to the TDM modules for further processing.

4.2.5 Domain-Specific Machine Translation Models

The Machine Translation system ensures accurate and contextually appropriate translations by fine-tuning general-purpose machine translation models with domain-specific scientific data. There are currently 3 translation models supporting the following language pairs: French to English, Spanish to English, and Portuguese to English. The models were generated by fine-tuning large pre-trained MT models for the aforementioned language pairs by employing appropriately curated parallel data from the scientific domain. The produced models are open-source and can be downloaded from the Hugging Face platform:

- **French-English:** huggingface.co/ilsp/opus-mt-big-fr-en_ct2_ft-SciLake
- **Portuguese-English:** huggingface.co/ilsp/opus-mt-pt-en_ct2_ft-SciLake
- **Spanish-English:** huggingface.co/ilsp/opus-mt-big-es-en_ct2_ft-SciLake

It is worth mentioning that these models are also being used by the Article segmentation for multilingual articles component that is being developed in the context of the Reproducibility Assistance service (Section 4.4).

[6] https://github.com/openaire/iis
[7] https://github.com/madgik/madis
[8] https://hadoop.apache.org/

4.2.6 Knowledge Graph creation assistant

The aim of the Knowledge Graph Creation Assistant Tool Bundle is to deliver a set of automated and semi-automated tools to facilitate the creation of a collection of SKGs, following the task of data acquisition. Use-case partners have different data sources which could be structured as well as unstructured and would utilise the tools we provide to create their own pipeline with respect to the creation of a knowledge graph. Unstructured data would be transformed to (semi-)structured data by information (entity, relation, property) extraction. Schema mapping tools are developed to transform (semi-)structured data to a property graph that captures both topological and data artefacts in a flexible-schema data structure.

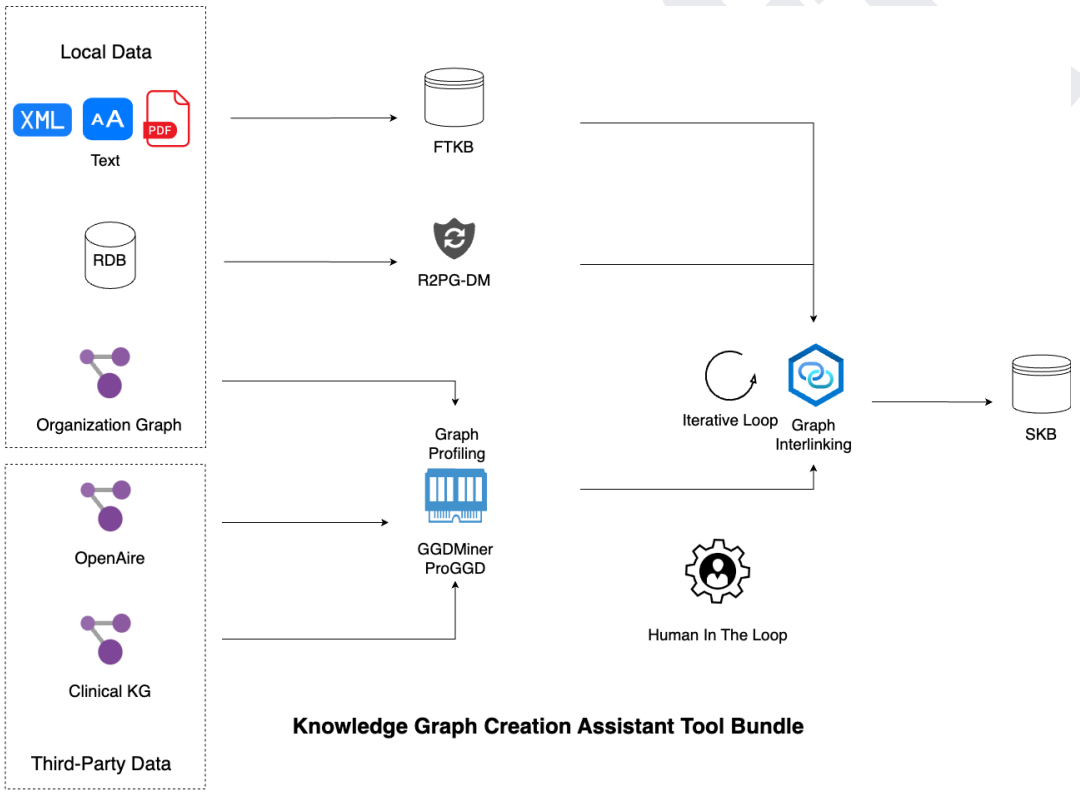


Figure 4: Knowledge Graph Creation Assistant Tool Bundle

To guarantee the quality of its practical implementation, the whole Knowledge Graph creation assistant tool bundle is meant to be used by the use-case partners, aka humans in the loop.

R2PG-DM

R2PG-DM is a direct mapping which follows a natural, logical translation of relation databases to property graphs by preserving the schema and by reasoning over basic properties of a direct mapping.

ProGGD

Graph Differential Dependencies (GDDs) are a novel class of integrity constraints in property graphs for capturing and expressing the semantics of difference in graph data. They are more expressive, and subsume other graph dependencies; and thus, are more useful for addressing many real-world graph data quality/management problems.

ProGGD is a system that uses GGDs to represent information about the graph data. Graph Generating Dependencies can express topological constraints based on two (possibly different) graph patterns and the similarity of property values of nodes and edges within those defined graph patterns[9]. Given a graph and its schema information, ProGGD discovers GGDs from the graph data and displays to the user not only the GGD in itself but also validated/non-validated data-examples for each one of the GGDs.

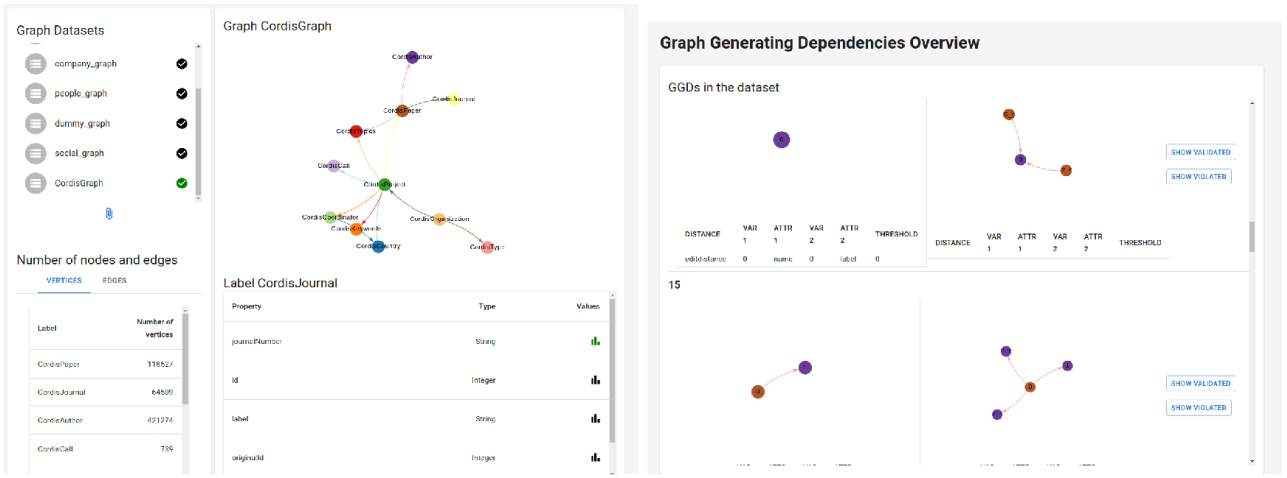


Figure 5: *ProGGD user panel*

GDDMiner

GDDMiner addresses the general discovery problem for GDDs: the task of finding a non-redundant and succinct set of GDDs that hold in a given property graph. It efficiently returns a minimal cover of valid GDDs.

[9] Shimomura, L.C., Yakovets, N. and Fletcher, G., 2022. Reasoning on Property Graphs with Graph Generating Dependencies. arXiv preprint arXiv:2211.00387

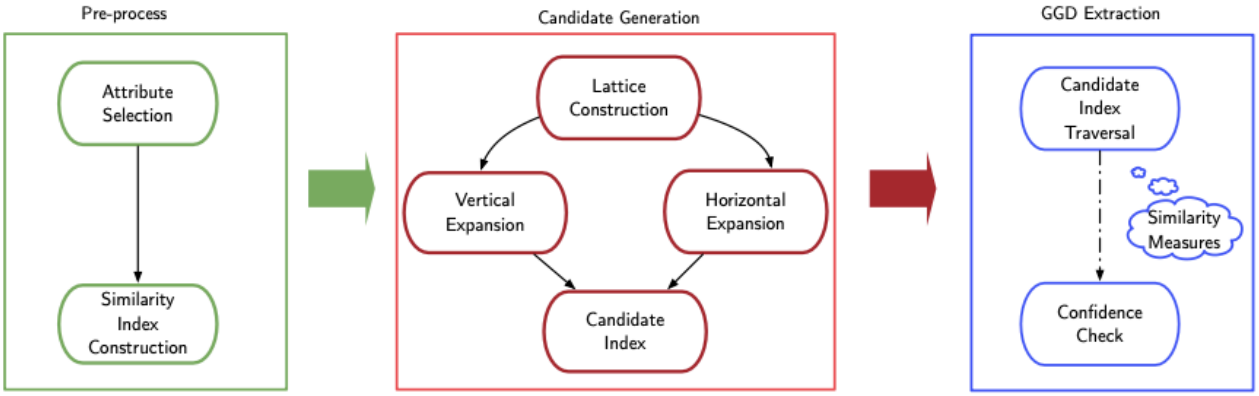


Figure 6: GDDMiner workflow

4.2.7 Data interlinking component

The data interlinking component is mainly based on the functionalities provided by SciNeM and sHINER, two components that are also responsible for the link recommendation functionalities offered in the context of the Reproducibility Assistance service. Hence, the respective descriptions can be found in the respective subsection of Section 4.4.

4.2.8 Community Gateways

The Research Community Gateway is a service which fosters transparent evaluation of results and facilitates reproducibility of science for research communities by enabling a scientific communication ecosystem supporting exchange of research products (publications, research data, research software, methods) and links between them across communities and across content providers. It is a Virtual Research Environment making it easy to share, link, disseminate and monitor publications, research data, research software, methods in one place. It simplifies the discovery of research products within a community, finding a repository to deposit research outcomes and linking research products with a community, funding stream or other research product.

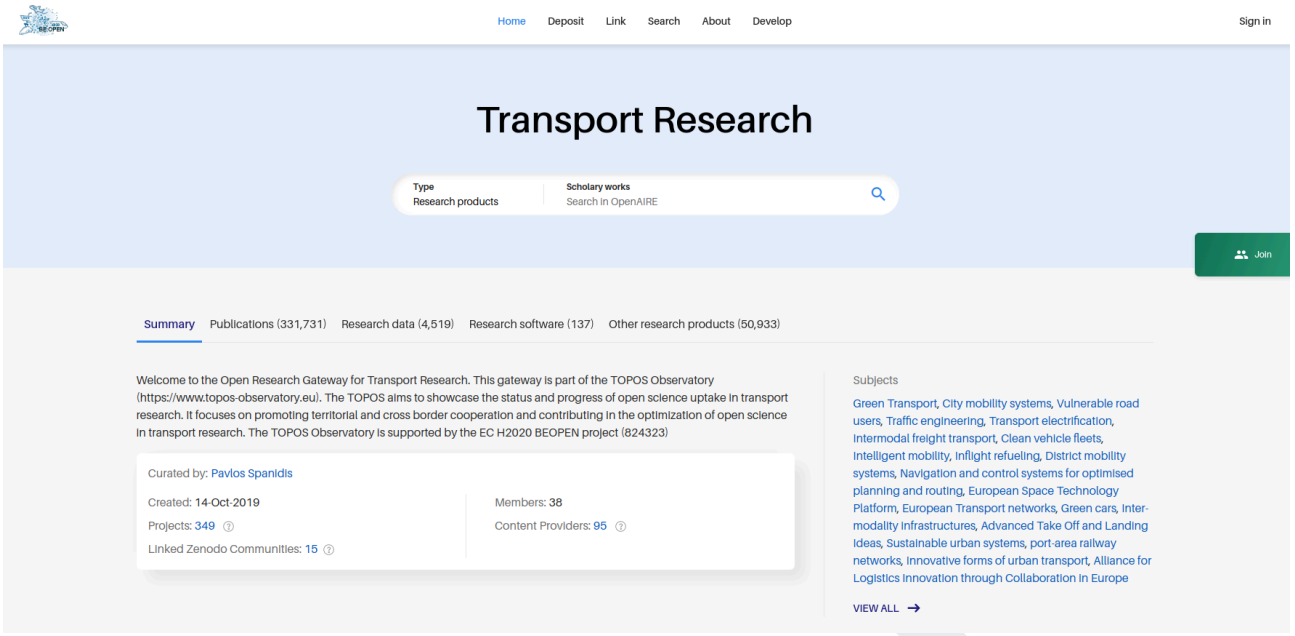


Figure 7: *Transport Research Community Gateway*

4.3. Impact-driven Discovery service components

In the following paragraphs we outline the components which are eligible for integration in the Impact-driven Discovery service subcomponent.

4.3.1 BIP! Services

BIP! Services is a set of services that form a platform offering scientific literature exploration and research assessment services leveraging advanced citation-based impact indicators on top of scholarly knowledge graphs. BIP! aggregates citation data from the OpenAIRE Graph constructing a citation network that contains more than 190 million research products (articles, datasets, software, and other types of products). A set of citation-based indicators is, then, computed on top of this network in a scalable manner (leveraging Apache Spark technologies), capturing distinctly different aspects of scientific impact, such as popularity (current impact), influence (overall impact), and impulse (initial momentum).

The aforementioned impact indicators are calculated by the BIP! Ranker component[10] and, after their calculation, they are used to offer various services to the research community at large. The most important service is related to ranking search results in academic search

[10] BIP!-Ranker, https://github.com/athenarc/Bip-Ranker

engines to facilitate scientific knowledge discovery. BIP! Finder[11] is the main end-user interface to support relevant functionalities, offering impact-based ranking and filtering of research products to assist literature exploration scenarios. In the context of the project, this interface was adapted and extended to create BIP! Spaces[12], a platform to support tailored scenarios that can better accommodate the needs of experts in specific domains, leveraging also domain-specific SKGs. For that need, BIP! Spaces service offers various customisation options and has implemented a connection to the Knowledge Graph engine provided by the SciLake ecosystem. BIP! Spaces offers useful features to the domain experts such as the incorporation of domain-specific annotations on the search results based on the contents inside various domain-specific SKGs. It is also acting as the backbone for the various demonstrations showcasing how the results of the various SciLake components can offer useful, tailored services for domain experts. The code of BIP! Finder and Spaces is open source and available at https://github.com/athenarc/bip-services.

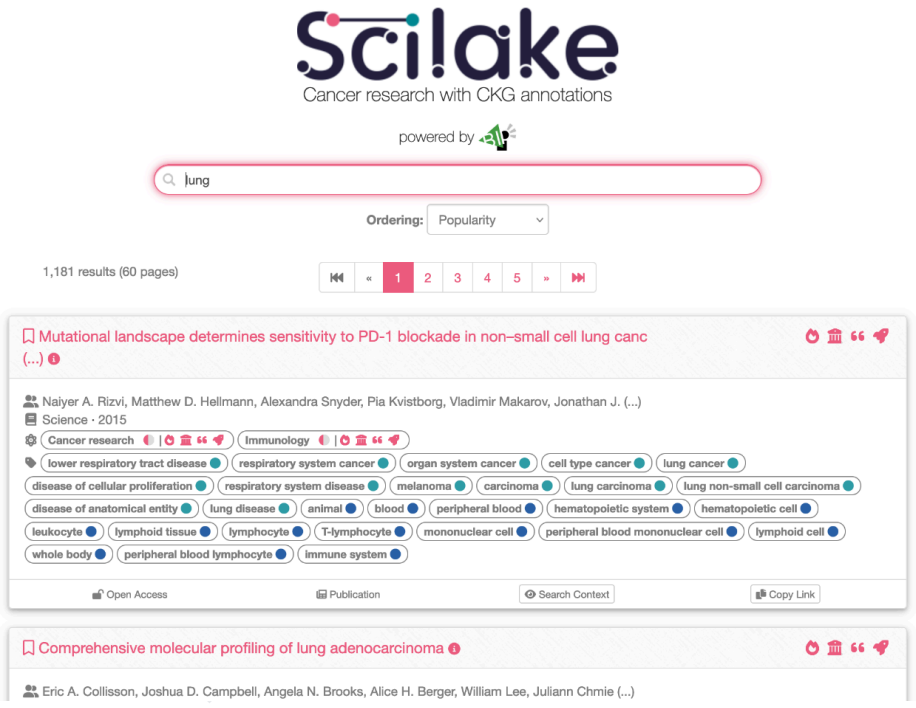


Figure 8: *BIP! Space for the SciLake cancer research pilot*

[11] BIP! Finder, https://bip.imsi.athenarc.gr/search
[12] BIP! Spaces, https://bip.imsi.athenarc.gr/spaces

4.3.2 SciTo

SciTo is a tool for monitoring scientific trends by identifying topics that are expected to attract attention in the near future. It is intended to be used mostly by researchers, fund managers etc. The tool leverages the topics of publications that can come from a topic modelling approach or a well-established scholarly data source (e.g., the OpenAIRE Graph or OpenAlex). The tool offers various visualisations on how the various topics evolve in the course of time. These visualisations exploit the number of publications related to the topics of interest, their impact in the respective scientific domain, and other information, such as the similarity between different topics. Initially the tool was provided as an independent UI but, in the context of the project, efforts are made to integrate the respective functionalities inside the BIP! Spaces tool.

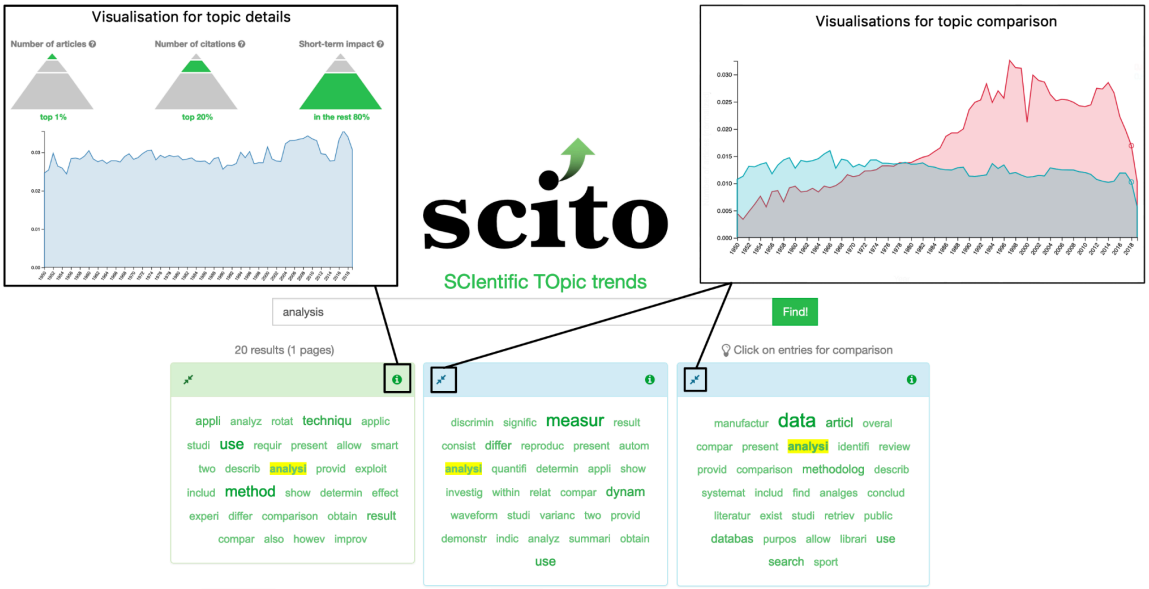


Figure 9: *SciTo Visualisation charts*

4.3.3 SciNoBo FoS (Field of Science)

The SciNoBo Field of Science (FOS) classifier classifies publications into one or more FOS labels and is based on the assumption that a publication mostly cites thematically related publications. It bridges venues (journals/conferences) and publications by constructing a multilayer network (graph) in which venues are represented by nodes, and venue-venue edges reflect citing-cited relationships in their respective publications. The FOS classifier assigns FOS labels through the publishing venues of the publications it references and the

publishing venues of the publications it gets cited by. The FOS taxonomy that is used as a classification scheme is based on an enhanced OECD discipline/Fields-of-Science taxonomy.

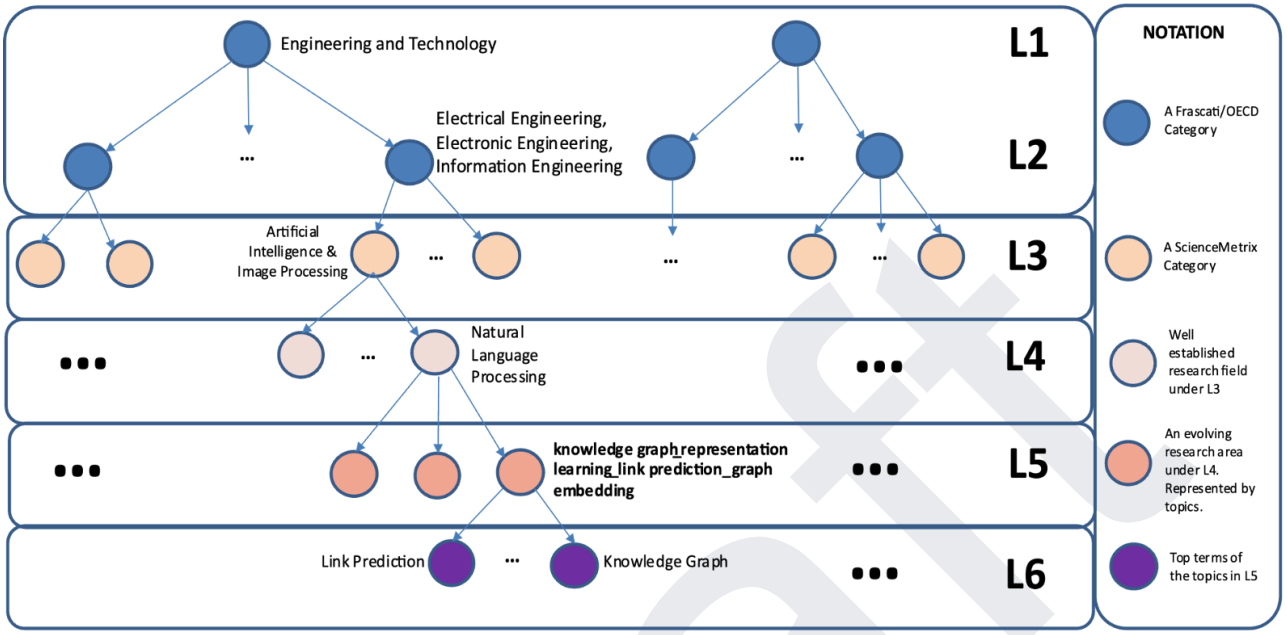


Figure 10: *SciNoBo Field of Science classification levels*

4.3.4 Impact propagation

The impact propagation tool addresses the problem of measuring research object's impact for software and datasets. Leveraging the contents of the Lake, it computes impact scores for research objects by considering different types of mentions in papers and in the textual descriptions of other objects, not limited to structured citations but also including "indirect" citations such as plain mentions or footnotes, which are the dominant forms in which objects other than papers tend to be cited, in addition to taking into account the intent of the citations—e.g., reuse or neutral mention. Current efforts are focused on the appropriate weighing of these different types of relationships into the computation of the impact scores.

4.4. Reproducibility Assistance service components

In the following paragraphs we outline the components which are eligible for integration in the Reproducibility Assistance service subcomponent.

4.4.1 SciNoBo Objects Recommendation

This tool is powered by the SciNoBo Research Artefact Analysis (RAA). It performs RAA on scientific texts to extract mentions of RAs (e.g. datasets, software) along with their metadata, and then deduplicates these mentions to find the unique RAs that were referenced, used or created in the text. The tool leverages fine-tuned large language models (LLMs), a Coreference Resolution (SciCo) Longformer Model and relies on a predefined list of discipline-specific keywords, key phrases, and gazetteers to detect candidate mentions of RAs. It comprises five stages: (i) candidate detection, (ii) RA mention identification and validation, (iii) extraction of RA mention metadata, such as names, versions, licences, and URLs, (iv) classification of RA mentions by usage and ownership, and (v) deduplication of RA mentions to ensure the uniqueness of each identified RA.

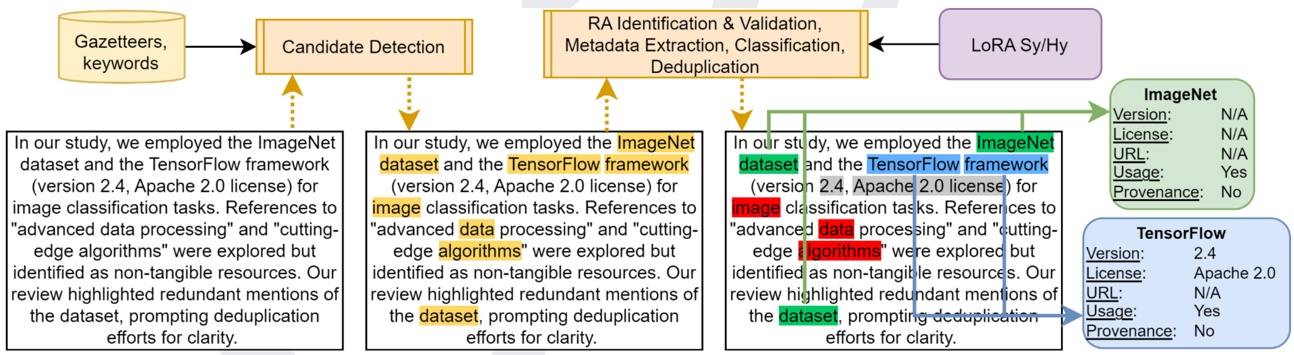


Figure 11: *An overview of how the SciNoBo RAA tool works (pt1)*

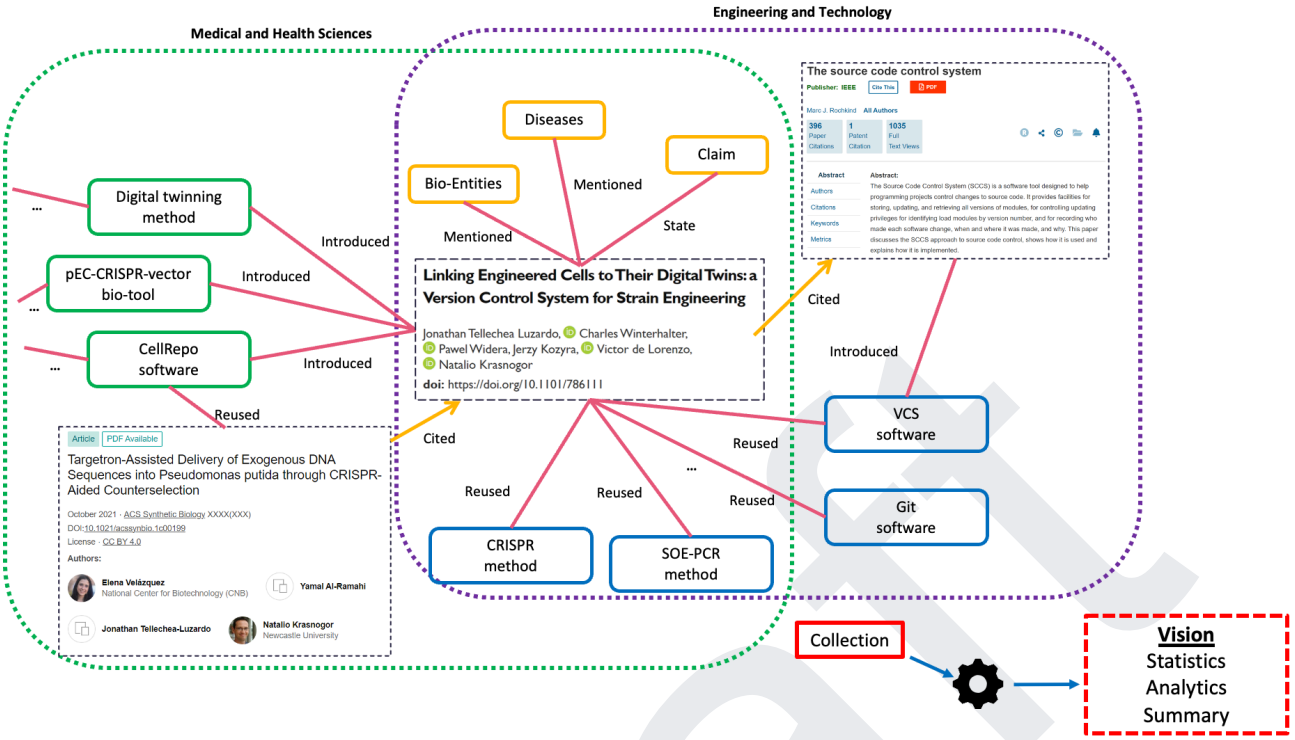


Figure 12: *An overview of how the SciNoBo RAA tool works (pt2)*

4.4.2 Research entity mentions

This service aims to identify mentions of domain-specific entities/concepts of interest in publications and in the textual descriptions of other research outputs, and to normalise these mentions with persistent identifiers through the linking with standard taxonomies. The initial version of the NER module is based on the all-in-one NER (AIONER) annotation scheme and focuses on the identification of biomedical entities, which are relevant to two of the four research communities that are part of the pilots of the project. It will be expanded with additional entities tailored to the other two pilots, with flexibility in mind to accommodate additional communities in the future.

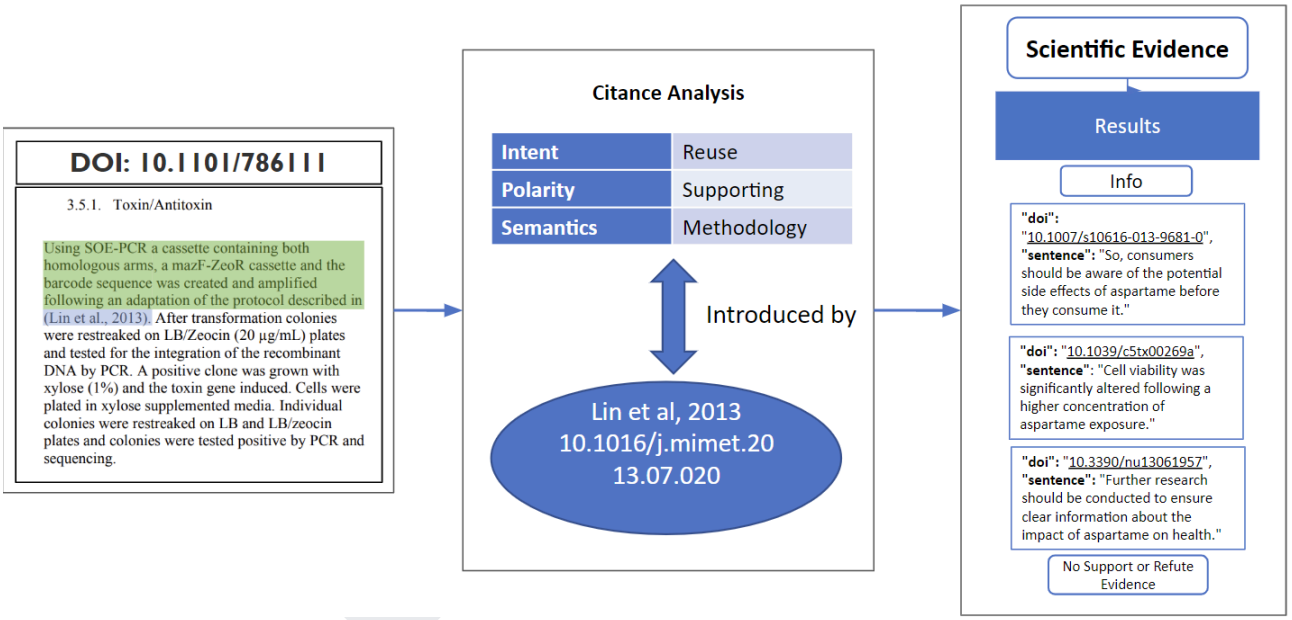
4.4.3 SciNoBo Reproducibility

Supports tracking and monitoring of research artefacts in scientific literature. Provides a list of artefacts used in research areas of interest, enriched with metadata and references. SciNoBo Citance Analysis (CA) analyses citation mentions (citances) and tries to determine their Intent, Polarity, Semantics and further scientific evidence.

The primary objective is to provide a nuanced analysis of citations, aiding in the replication and validation of scientific findings. Specifically, it addresses the following research questions:

- What is the purpose or intent of the citation?
  - Generic References
  - Reuse instances
  - Comparison scenarios
- What stance do the authors take towards the cited work?
  - Supporting
  - Neutral
  - Refuting
- What aspect of the cited work is being referenced?
  - Claim
  - Methodology
  - Results
  - Research Artefact

The tool utilises fine-tuned Large Language Models (LLMs) in an instruction-based Question Answering (QA) setting to perform the classification.



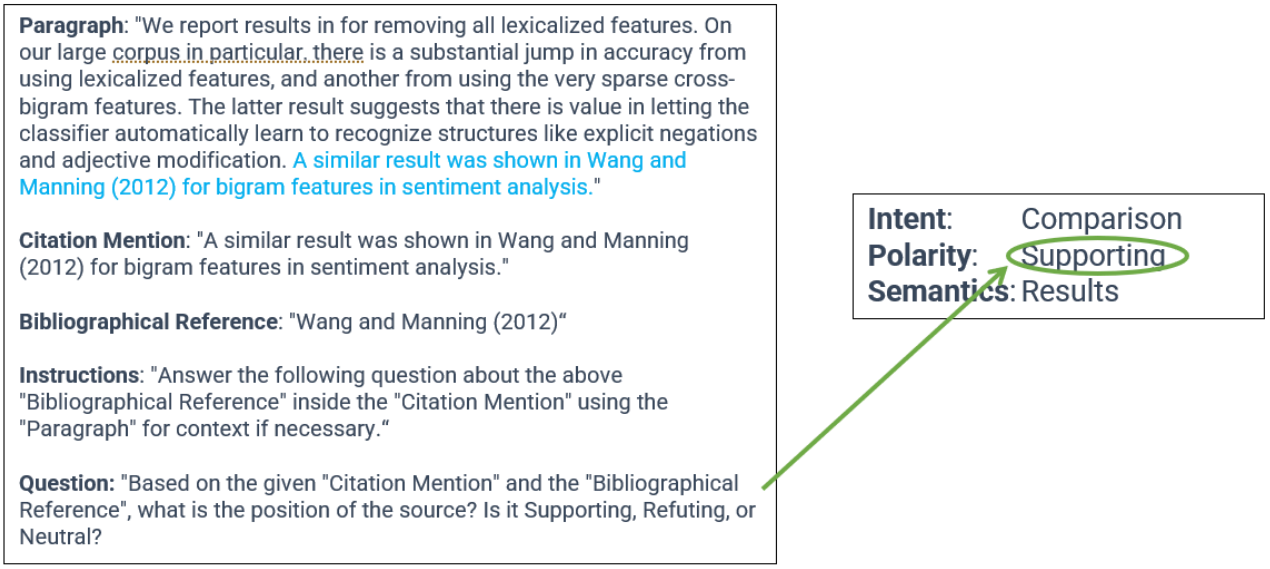


Figure 13: *SciNoBo Citance Analysis Question Answering setting*

4.4.4. Article segmentation for multilingual articles component

This component is mainly based on the DoStRe software, but it also leverages the automatic translation models developed in the context of the Scientific Lake service (Section 4.2).

DoStRe (Document Structure Recognition) is a tool that recognizes the structure of scientific documents. The objective of this tool is the recognition of the structure of documents, specially identifying parts (sections and titles). Apart from the identification of parts of the document, a second functionality is the classification of the parts in predefined section/part types, i.e, the conclusions in a scientific paper could be named in many ways, such as 'Conclusions', 'Conclusions and Future Work', 'Discussion', etc.

4.4.5 Link recommendation component

sHINER for Entity Resolution

The sHINER system is implemented for entity resolution in graph data using GGDs. The following figure shows the sHINER system architecture.

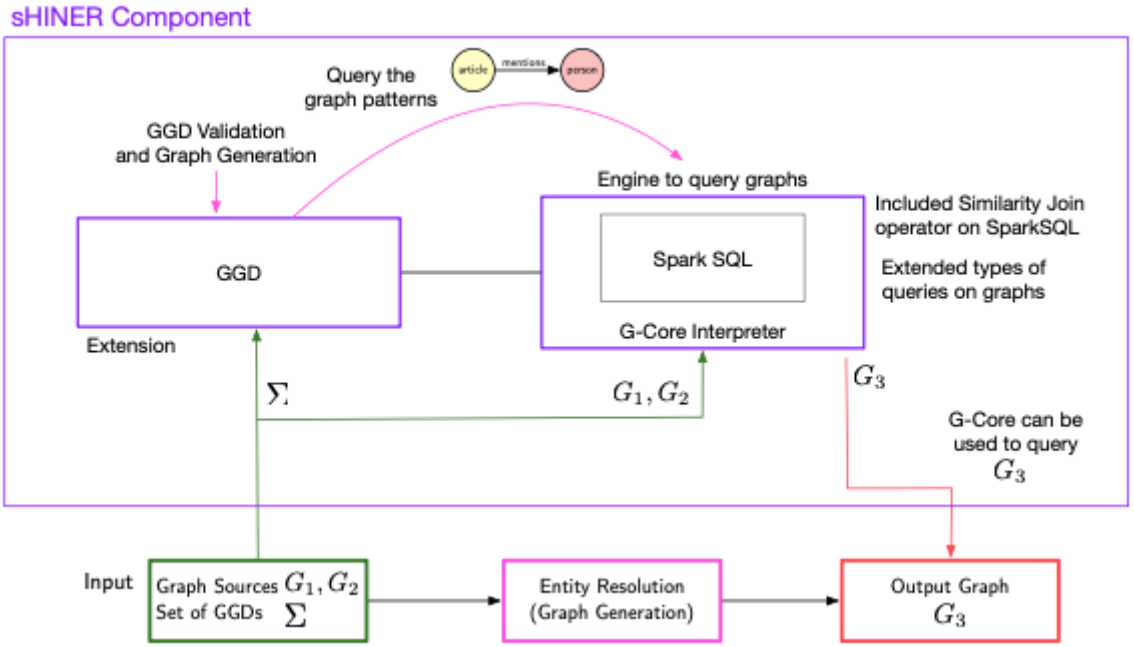


Figure 14: *sHINER Components*

sHINER has mainly two components: the G-Core language interpreter, built over Apache Spark[13], and the GGDs component, responsible for the interpretation and execution of the GGD validation and graph generation algorithms used in the entity resolution.

SciNeM

SciNeM[14] is an open-source tool that offers a wide range of functionalities for exploring and analysing Knowledge Graphs encoded in the HIN (Heterogeneous Information Network) format and utilises Apache Spark[15] for scaling out through parallel and distributed computation. SciNeM also provides an intuitive, Web-based user interface to build and execute complex constrained metapath-based queries and to explore and visualise the corresponding results. Under the hood, all the supported state-of-the-art HIN analysis types

[13] https://spark.apache.org/
[14] https://github.com/athenarc/SciNeM-workflows & https://github.com/athenarc/SciNeM
[15] https://spark.apache.org/

have been implemented in a scalable manner supporting the distributed execution of analysis tasks on computational clusters. SciNeM has a modular architecture making it easy to extend it with additional algorithms and functionalities.

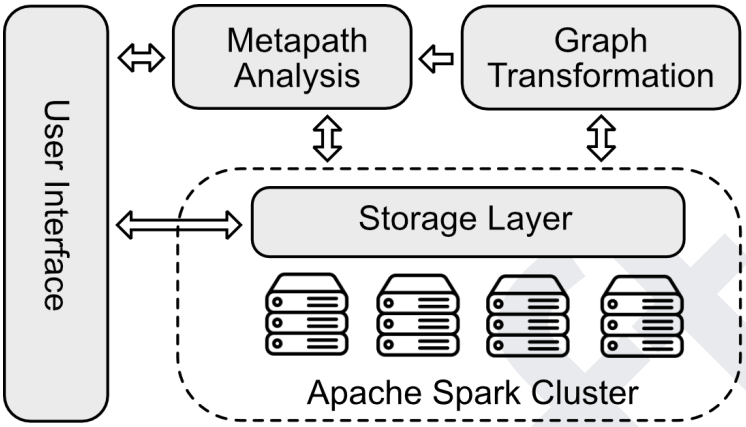


Figure 15: *SciNeM's main components*

Currently, it supports the following operations, given a user-specified metapath:

- ranking entities using a random walk model
- retrieving the top-k most similar pairs of entities
- finding the most similar entities to a query entity
- discovering entity communities

while, in the context of the SciLake project, SciNeM will be adapted to provide recommendations for links between research objects (more details in deliverable D4.1).

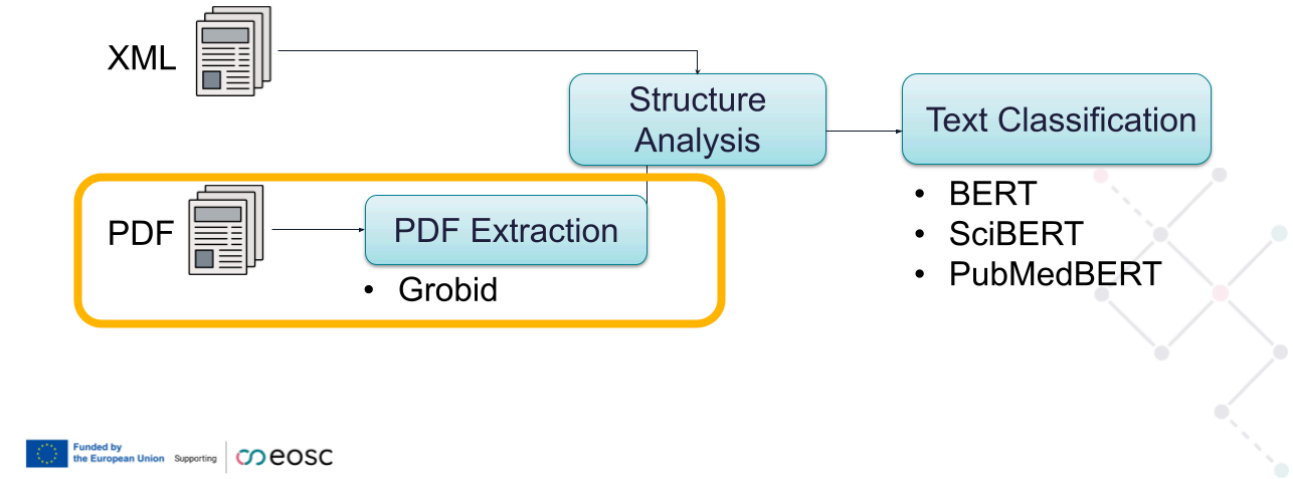


Figure 16: *DoStRe components*

5. Integrated system overview

The Scilake integrated system is designed to incorporate an array of computational nodes, called "SciLake nodes". Each node is expected to deploy a selection of SciLake components (those described in Sections 4.2 - 4.4) and data (mostly SKGs) creating, essentially, a federation of nodes that implements the conceptual architecture described in Section 4.1. The graphs deployed within SciLake Nodes are tailored for the needs of a given community or or group of communities. This flexible and distributed design is also similar to the newly introduced concept of an EOSC Node[16].

In the next subsections we discuss the current integration status within each of the tiers of the SciLake high-level conceptual architecture presented in Section 4.1.

5.1. Fundamental Scientific Lake Tier

The fundamental Scientific Lake tier consists mainly of a deployment of the Scientific Lake service. As mentioned, a such deployment on a computational cluster essentially involves (a) installing the Knowledge Graph engine in various nodes of the cluster, each hosting (and making accessible) a different SKG, and (b) deploying and executing a selection of the various

[16] https://eosc.eu/building-the-eosc-federation/

Scientific Lake components on various nodes of the cluster. The selection of the components is based on the needs of the use case of interest and the particularities of the respective knowledge domain.

The fundamental Scientific Lake tier is organised in a way that facilitates the discovery and consumption of the contents of the Lake. The former is facilitated by the SciLake catalogue component, which offers search and navigation capabilities for the respective ecosystem. The latter is possible due to the interoperability guidelines that are followed for the creation and extension of the SKGs hosted by the Lake and the open API that is designed and implemented following the respective specifications. In the following paragraphs we elaborate more on these subjects.

The starting point for any end-user interested in the Scientific Lake contents is the SciLake Catalogue. This component, positioned at the centre of the federated system (hosted on one of the cluster nodes), allows listing and searching for the Knowledge Graphs hosted on the nodes of the system, as well as for SciLake tools deployed on them. The catalogue makes available useful information for the previous resources, such as basic descriptions, specifications used, documentation, etc. The metadata collected for tools and services are influenced by metadata specifications for research services that have been used in the past by EOSC or similar initiatives, in an attempt to make their integration into the EOSC ecosystem easier.

Regarding offering a convenient environment to create, update, and maintain SKGs, deployments of the various components described in Section 4.2 are configured and ready to be executed on demand on a selection of cluster nodes. These tools either (a) transform knowledge from sources having different formats to create SKGs that can later be hosted on an instance of the Knowledge Graph engine (see also the conceptual diagram in Figure 1) or (b) are capable of further enriching existing SKGs with additional contents. The tools are naturally heterogeneous, however containerisation is used to simplify their deployment process. Furthermore, as explained later in the text, special care is taken to ensure that bundles of tools, which are logically used together, are made more compatible and easier to use in tandem. This includes offering straightforward deployment methods and common programmatic interfaces.

Regarding accessing and querying the Lake contents in a unified and powerful manner, AvantGraph implements the Knowledge Graph engine, which is at the heart of each SciLake node. It is expected that each SciLake ecosystem deployment will include multiple deployments of AvantGraph, each offering accessing and querying capabilities on top of a different SKG. As of today, the following AvantGraph instances were established:

- one for the domain agnostic OpenAIRE Research Graph and

- two for domain specific SKGs dedicated for pilots: one for Cancer Research and one for Transportation.

In general, various tools are expected to be used in bundles and, as a result, some effort is given in making them more interoperable. For instance, the Knowledge Graph Creation Assistant bundle, tightly bound to the AvantGraph engine, was developed to facilitate the process of the creation of a collection of SKGs. It is meant to be used by use-case partners to create their own data processing pipeline following the task of data acquisition and resulting in creation of a knowledge graph. Additionally, PDF Fetcher and the Text and Data Mining component, which are also Scientific Lake components, are currently integrated with the domain agnostic **OpenAIRE Graph** data provision pipeline and both are ready to be utilised within a scope of domain specific SKGs. PDF Fetcher is responsible for obtaining PDF documents associated with publication graph entities while the data mining subsystem relies on plaintexts extracted from those PDF files, along with the research graph data, in order to produce inferred information which comprises the further extension of the graph. The full list of graph-enrichment modules is available on the OpenAIRE Graph documentation page.

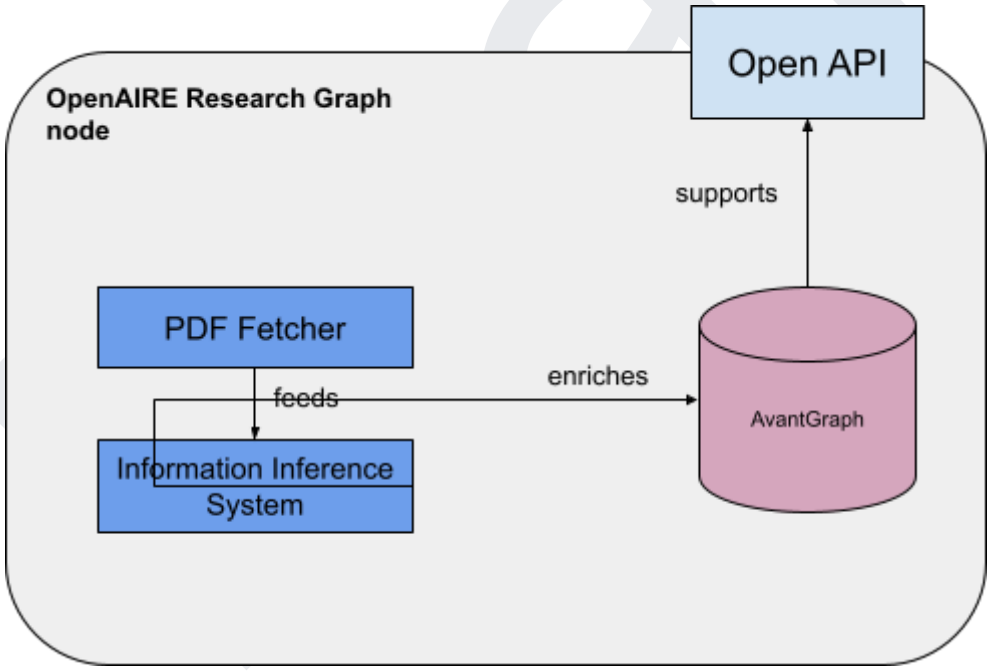


Figure 17: *OpenAIRE Graph node components*

In order to ensure a common way of accessing graph data on every SciLake Node, a common **Lake API** will be deployed on each node. The respective API specification is building upon Research Data Alliance's SKG Interoperability Framework[17] (SKG-IF). This framework offers a

[17] SKG-IF: https://skg-if.readthedocs.io/en/latest/

metadata model for the most common entities covered by domain-agnostic research-related entities (such as publications, datasets, researchers, organisations). SciLake's API endpoints, parameters, and fields will adhere to a consistent nomenclature and structure whenever possible. Simultaneously, given SciLake's focus on addressing the specific needs of various domains, the project will provide extensions to this model to meet those needs. Feedback from domain experts will be instrumental in this process (the process has already started). Since SKG-IF is planned to officially accommodate similar extensions through a "profiles" concept in the near future, the project will also aim to express all developed extensions in a format compatible with the guidelines established by the respective Research Data Alliance interest group.

Based on the previous, SciLake's API is expected to offer a set of SKG-IF-compatible entity-focused endpoints. On top of that, the respective API will also offer a powerful graph query endpoint that will make it possible for its end-users to pose graph queries (e.g., in Cypher). This functionality (built on top of the KG engine) will give a lot of expressiveness to the API end-users in building value-added services. As expected, the LAke API will be utilised by the Application Tier services facilitating the consumption of the Scientific Lake contents that is required.

5.2. Application Tier

Apart from the core components of the Scientific Lake Service described above, each SciLake Node may be supplemented with deployments of components aiming to support the provision of value-added services for the research community. More specifically, the project is focusing on two main categories of such services: impact-driven discovery services (elaborated in D3.1) and reproducibility assistant services (elaborated in D4.1).

These value-added services rely on combining different components that can produce valuable SKG enrichments or offer related end-user functionalities. These components are leveraging the Scientific Lake service via the APIs to access and/or analyse the SKG contents.

For the impact-driven discovery case, the main focus of the project is on implementing BIP! Spaces, a service that supports the creation of customised knowledge discovery portals (called "spaces"), tailored for the needs of particular communities. Each space is capable of considering the contents of the OpenAIRE Graph, but also of one specified SKG (that captures the domain knowledge of interest). The aim of the project is to build one space for each pilot use case (incorporating in the space the respective domain-specific SKG). At the moment, there is only one mature BIP! Space created for the Cancer Research pilot, which is based on the current version of the respective SKG.

Currently, the BIP! Spaces support keyword-based search for research products (publications, datasets, software, and other products) combined with annotations coming directly from the determined SKG. Multiple annotation types are supported, while each of them corresponds to a particular graph query that can be served by the KG engine in which the SKG is loaded. As it is evident, the BIP! Space should be configured so that it is aware of a KG engine deployment to which it should be connected.

The BIP! Spaces also support topic evolution features (as it is elaborated in D3.1). The implementation is capable of supporting various types of topics and the intention is to make it possible to support this feature by consuming FoS topics that have been incorporated into the OpenAIRE Graph as a result of the SciNoBo FoS classifier. BIP! Spaces has integrated features from the SciTo tool so that it will be able to provide the respective functionalities.

In addition to BIP! Spaces, each domain specific environment was also covered with a dedicated OpenAIRE CONNECT dashboard in order to facilitate the communities with a common space which allows sharing, linking, disseminating and monitoring of various research products such as publications, research data, research software and methods.

Each one of four pilots has its own OpenAIRE Community Gateway created:

- Cancer Research (restricted access)
- Energy Planning (restricted access)
- Transport Research
- Neuroscience (restricted access)

Regarding the reproducibility assistance service, the implementation of the end-user features will be carried out in the form of badges of reproducibility and replicability integrated into the BIP! Spaces UI. The enrichments required to support these functionalities (e.g., the links between research products) are being produced from the respective components and being integrated into the OpenAIRE Graph or/and the domain-specific SKGs. Through the respective APIs they are being available to the respective value-added services.

By the end of the project, both value-added services are expected to provide not only useful user interfaces for experts involved in the project pilots but also API endpoints to facilitate their integration, whether complete or partial, into third-party services. The specifications for these API endpoints will be refined and documented later in the project.

6. Conclusions

At the current stage the integrated system is realised as a federation of independent nodes, each one with a different set of components involved. In particular domain agnostic OpenAIRE Graph, materialised also as an AvantGraph database, is already integrated with the set of components responsible for content delivery and text and data mining while the domain specific SKGs are at the stage of being materialised by feeding the dedicated AvantGraph instances with the data. The discoverability of all available SKGs and services is addressed by the Catalogue instance which was already deployed and is currently being finalised.

The specification of the OpenAPIs, which is currently under development, is expected to be concluded within the upcoming period and should result in a more standardised way of accessing SKG resources and services.

The plan for the final integrated system is to allow incorporation of more components into each of the SciLake nodes in order to widen the range of added value services tailored for a specific use case and to make the infrastructure highly customizable. The collaboration with the pilot representatives is meant to advance further in order to tailor already developed solutions to the specific pilot needs and to propose viable integration scenarios. At the later stage, when the most of the SciLake components are in matured state, researchers outside of the project consortium are meant to be engaged in an attempt to receive additional feedback.

7. References

Not Applicable