Promoting and Incentivising Federated, Trusted, and Fair Sharing and Trading of Interoperable Data ASsets

D2.1
Data Interoperability, Management and Protection Framework

Editor(s)	Yury Glikman
Lead Beneficiary	FHG
Status	Final
Version	1.0
Due Date	30/09/2023
Delivery Date	31/10/2023
Dissemination Level	PU

Funded by the European Union under Grant Agreement 101093016. 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 Commission. Neither the European Union nor the granting authority can be held responsible for them.

Project	PISTIS – 101093016
Work Package	WP2 - PISTIS Data Spaces Factory and Trusted Data Management Services
Deliverable	D2.1 - Data Interoperability, Management and Protection Framework
Contributor(s)	FHG, SUITE5, UBITECH, ATOS, EUT, ATHENA, ASSENTIAN, SPH, UBIMET, OAG
Reviewer(s)	ICCS
Abstract	The deliverable presents the landscape of the methods and relevant technologies, models and specifications to drive the design and implementation of services that deal with the management, quality improvement, assessment and protection of data in PISTIS.

Executive Summary

The deliverable comprises the first results of Task 2.1 " Interoperability & Semantics, Security, Privacy & Trust, and Quality Assessment Methods", Task 2.2 "Data Lineage, IPR Safekeeping, Usage Tracking and Enforcement Models", Task 2.3 "Data Ingestion, Transformation, Insights Generation and Quality Assessment services", Task 2.4 " XAI driven Data Lineage Marking, IPR and Sovereignty Reviewing and Analytics services", Task 2.5 "Distributed Data Space Exploration, Asset Discovery, and Access services" and Task 2.6 " Intelligent, Performant Data Privacy, Trust Generation and Content Protection services", documenting the relevant methodologies and technologies that are considered for the design and implementation of the Data Spaces Factory and Trusted Data Management services responsible for Data Interoperability, Management and Protection in PISTIS.

The deliverable comprises the stepping stone where the design and development activities of WP2 " PISTIS Data Spaces Factory and Trusted Data Management Services" will be based on, driving the process for the final selection of the methodologies and technologies that will be devised for the implementation of the PISTIS Data Interoperability, Management and Protection services. This deliverable has received input from the deliverable D1.1 "PISTIS Operation Principles and Context Detailing" regarding the high-level conceptualization of PISTIS Market Exchange Platform and is interconnected with the deliverable D3.1 "Data Valuation, Sharing and Trading Framework" for the interrelation among components of WP3 "PISTIS FAIR Data Trading and Value Exchange/Monetisation Platform Services" and it will provide feedback to the deliverable D4.1 "PISTIS Reference Architecture and API Documentation" towards the finalization of the overall PISTIS Architecture and the planning of integration processes. It will also constitute the basis for the deliverable D2.2 "Data Management and Protection services - Alpha version" for the design and development of the Alpha version of the PISTIS Market Exchange Platform.

Table of Contents

List of Figures

List of Tables

Terms and Abbreviations

AA	Attribute Authority
ABE	Attribute-based Encryption
AES	Advanced Encryption Standard
CP-ABE	Ciphertext Policy Attribute-Based Encryption
DCAT	Data Catalog Vocabulary
DCAT-AP	DCAT Application Profile for data portals in Europe
DES	Data Encryption Standard
DH	Diffie-Hellman
DLT	Distributed Ledger Technology
DQA	Data Quality Assessment
DQD	Data Quality Dimension
DQM	Data Quality Metric
DSSE	Dynamic Symmetric Searchable Encryption
DV	Data Valuation
ECC	Elliptic Curve Cryptography
FM	Factorisation Machine
GDPR	General Data Protection Regulation
GNN	Graph Neural Network
IBE	Identity-based Encryption
IRI	Internationalized Resource Identifier
KNN	K-Nearest Neighbours
KP-ABE	Key-Policy Attribute-Based Encryption
LDA	Latent Dirichlet Allocation
LIME	Local Interpretable Model-Agnostic Explanations
LSA	Latent Semantic Analysis
LSH	Locality Sensitive Hashing
MF	Matrix Factorisation
ML	Machine Learning
MQA	Metadata Quality Assessment
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMF	Non-Negative Matrix Factorization
PII	Personally Identifiable Information
PKG	Private Key Generator
RDF	Resource Description Framework
RSA	Rivest - Shamir - Adleman
SE	Searchable Encryption

SSE	Searchable Symmetric Encryption
TTP	Trusted Third Party
URI	Uniform Resource Identifier

1 INTRODUCTION

The Data Space Factory Environment is an important part of PISTIS. It is going to be deployed at the premises/private infrastructure by every organisation participating in the PISTIS ecosystem. It enables the organisation to import the data from its other data infrastructure and prepare it for the offering and publishing the data offer at the PISTIS Data Trading & Value Exchange/Monetisation Platform Market Exchange Environment. The Data Space Factory Environment will provide such functionalities as data import, transformation, data insights generation, enrichment, anonymisation, quality assessment and lineage tracking. It will empower the organisation to keep the control under own data and then supply it to the data buyer after a valid contract is made.

The next sections of the document present an overview of relevant methods, models and technologies to be considered for the implementation of the Data Space Factory Environment functionalities and for ensuring the data security and trust in PISTIS. It summarises the theoretical results from the WP2 “PISTIS Data Spaces Factory and Trusted Data Management Services” tasks:

- Task 2.1 " Interoperability & Semantics, Security, Privacy & Trust, and Quality Assessment Methods",
- Task 2.2 "Data Lineage, IPR Safekeeping, Usage Tracking and Enforcement Models",
- Task 2.3 "Data Ingestion, Transformation, Insights Generation and Quality Assessment services",
- Task 2.4 " XAI driven Data Lineage Marking, IPR and Sovereignty Reviewing and Analytics services",
- Task 2.5 "Distributed Data Space Exploration, Asset Discovery, and Access services",
- Task 2.6 " Intelligent, Performant Data Privacy, Trust Generation and Content Protection services".

1.1 DOCUMENT STRUCTURE

The document is structured into sections according to groups of functionalities:

Section 2 “Pistis (Meta-&)Data Models and Data Exploring” elaborates the role of data models, for both, actual data and metadata in PISTIS and introduces relevant general and proven concepts, methods and technologies for data models creation and management, metadata and data management, distributed querying and data matchmaking.

Section 3 “Data Ingestion & Transformation” elaborates on the role of Data Check-in, Data Processing Jobs Configuration, Data Enrichment, Data Transformation, Data Quality Assessment, Data Analytics and Insights Generation, Data Lineage and Usage Tracking and Enforcement in PISTIS and on the suitable methods and technologies for their implementation.

Section 4 “Data Peer-To-Peer Transfer” explains the role of Data Transfer in PISTIS and provides and overview of suitable methods and technologies.

Section 5 “Data Security and Trust” provides an overview of the main security and trust enabling functionalities together with methods and technologies to be considered for their implementation.

Section 6 concludes the document.

2 PISTIS (META-&)DATA MODELS AND DATA EXPLORING

This section elaborates the role of data models, for both, actual data and metadata in PISTIS and introduces relevant general and proven concepts, methods and technologies.

2.1 DATA MODEL MANAGEMENT

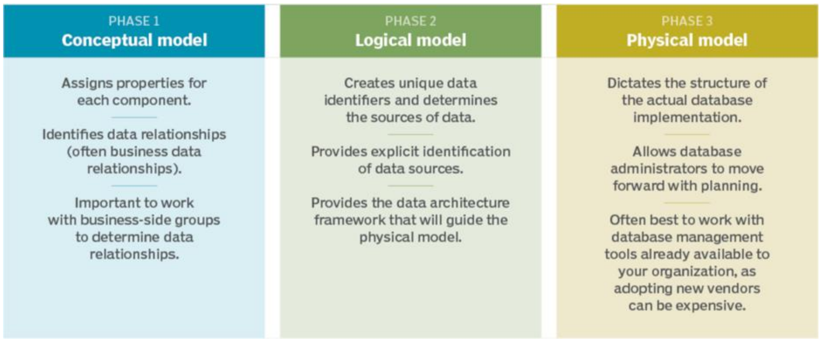
2.1.1 Data models, Data model management in PISTIS

Data modelling entails the creation of a conceptual/visual representation (i.e., a data model), either for an entire information system or specific components, to convey the connections between data points and structures. One of the most critical stages in the development of an information system is a data model creation for a given database[1]. Notably, alignment between data modelling strategies and database design schemas is crucial as these strategies are implemented in organisations according to their specific business requirements.

In general, a data model’s purpose is to organise and visually depict the different types of data utilised and stored in a system, standardize how they relate to one another (i.e., their nesting), describe how the data can be categorised and arranged (which is referred to a structured data), as well as the formats and characteristics of the data, for a domain of interest. As such, a data model’s main goal is to ensure that all data objects required by a database are completely and accurately represented. Entities, relationships, attributes, and cardinalities are the primary elements of a data model[2]; while the different phases of data modelling include the development of the conceptual, the logical and the physical data model; the latter being specific to the database management system or software that will be implemented, by defining the structures that the database or a file system will use to store and manage the data.

[1] G. Simsion and G. Witt, “Data Modeling Essentials.,” Elsevier, 2004

[2] H. e. a. Vera-Olivera, “‘Data Modeling and NoSQL databases - A systematic mapping review’,” ACM Computing Surveys, vol. 54, no. 6, p. pp. 1–26., 2021)



PHASE 1 Conceptual model	PHASE 2 Logical model	PHASE 3 Physical model
Assigns properties for each component.	Creates unique data identifiers and determines the sources of data.	Dictates the structure of the actual database implementation.
Identifies data relationships (often business data relationships).	Provides explicit identification of data sources.	Allows database administrators to move forward with planning.
Important to work with business-side groups to determine data relationships.	Provides the data architecture framework that will guide the physical model.	Often best to work with database management tools already available to your organization, as adopting new vendors can be expensive.

Figure 1: The three phases of data modelling[3]

Data modelling plays a pivotal role in two fundamental aspects of a digital enterprise; i) Software development projects performed by IT professionals, since data modelling guides software designers by outlining data structure and flow, ensuring accurate data if procedures are followed during development; ii) Analytics and visualisation – or business intelligence – a primary decision-making tool for users; where data modelling transforms raw data into actionable information for dynamic visualizations, while also preparing the data for analysis by cleansing, defining measures and dimensions, and enhancing it with hierarchies, units, currencies, and formulas[4].

A data model establishes a shared understanding and description of the information critical to the firm and the organization's larger data environment[5].

There are several types of data models used in various contexts, thus it is important to select an appropriate data model type that aligns with the requirements of a system/application, considering factors such as data complexity, interrelationships, performance needs, and scalability. Each model has its own strengths and weaknesses, and the choice depends on the specific use case and goals of the system. The most widely used types of data models include[6]:

- Hierarchical Model: Here data are organised in a tree-like structure with parent-child relationships. Each parent can have multiple children, but each child can have only one parent. The hierarchical model was widely used in early database systems but has become less popular in modern times.
- Network Model: It is an extension of the hierarchical model, allowing many-to-many relationships between records. It represents data as a collection of nodes connected by links, enabling more flexible relationships compared to the hierarchical model.
- Relational Model: It is the most widely used data model in modern database systems, where data are organized into tables with rows and columns. Each table represents an entity, and the relationships between entities are established through keys. The

[3] Nolle, T. (2020) *An enterprise architect's guide to the data modeling process: TechTarget*, *App Architecture*. Available at: https://www.techtarget.com/searchapparchitecture/tip/An-enterprise-architects-guide-to-the-data-modeling-process [Accessed: 04 September 2023].

[4] SAP, "what-is-data-modeling," SAP, [Online]. Available: https://www.sap.com/products/technology-platform/datasphere/what-is-data-modeling.html. [Accessed 26 june 2023].

[5] G. Li, X. Zhou and L. Cao, "AI meets database: AI4DB and DB4AI.," in *Proceedings of the 2021 International Conference on Management of Data*, 2021.

[6] https://en.wikipedia.org/wiki/Data_model

relational model emphasizes data integrity, and it supports powerful query capabilities through the use of Structured Query Language (SQL).

- Object-Oriented Model: The object-oriented model represents data as objects, similar to the concepts in object-oriented programming. Objects encapsulate both data and the operations that can be performed on that data. This model is useful for representing complex real-world entities with their attributes and behaviours.
- Entity-Relationship Model: The entity-relationship (ER) model is a conceptual model used to design databases, representing entities (objects or concepts) as tables, also defining the relationships between them (also referred to as ER diagrams or ERDs). The ER model focuses on the logical structure of the data and helps in understanding the relationships between different entities.
- NoSQL Models: NoSQL (Not Only SQL) refers to a category of database models that provide non-relational data storage options. These include a) Document data models where data is stored as self-contained documents, typically in JSON; b) Key-Value stores, where data is stored as pairs of keys (for fast and direct data retrieval)and associated values; c) Columnar data model where data is organised in columns storing data of same type, allowing for efficient compression and aggregation, and d) Graph data model, which represents data as nodes (entities) and edges (interrelationships), creating a network of interconnected entities. These models are designed to handle large amounts of unstructured or semi-structured data and offer high scalability and performance.

When it comes to the actual process of data modelling and the data model's lifecycle management, key activity is:

- the actual Data Model creation, entailing the gathering of requirements, analysis of business processes, and design of the conceptual, logical, and physical data models. Here collaboration between stakeholders, domain experts, and data professionals is imperative to ensure accurate representation of the data. In regard to the physical model creation, sub activities include:
- b) Data Model modification, where updates of the data model take place, to reflect possible changes to the business requirements, by adding or modifying entities, attributes, relationships, or constraints.
- Data Model documentation: essential for understanding and communicating the structure and semantics of the data; including creation of data dictionaries, metadata repositories, and model diagrams to serve as references for developers, administrators, etc.
- Data Model standardization, which involves defining naming conventions, data modelling guidelines, and best practices; towards improving the consistency, quality, and reusability of data models across projects and organizations.
- Data Model optimisation, which may involve denormalization, indexing strategies, partitioning, or other techniques to optimise performance and efficiency, thus enhance data retrieval and processing speed.

- Data Model governance, which involves establishing data model review processes, and compliance mechanisms, towards ensuring that data models align with organizational policies, data standards and regulatory requirements.
- Data Model versioning, which involves managing any changes to data models through versioning, tracking, and controlling model revisions, thus enabling traceability, rollback options, and collaboration among multiple stakeholders working on the same model.

In general, the importance of data modelling, lies in the fact that an all-encompassing (as practically as possible) and well-optimized data model, can facilitate the creation of a streamlined and organised database, effectively eliminating unnecessary repetition, minimising storage-resources needs, and enabling swift retrieval of information. Additionally, it provides all systems that incorporate it, with a unified and reliable data source, which is crucial for smooth operations and ensuring adherence to regulations and compliance to standards.

2.1.2 Methods

The field of data modelling is continuously evolving, and various data modelling techniques have been employed to represent and capture the structure and relationships of data. However, newer approaches such as ontological modelling, graph-based models, and NoSQL data models have gained prominence due to their ability to handle complex and diverse data structures. Widely used methods in the field of data modelling include[7]:

- Entity-Relationship (ER) Modelling: A traditional data modelling approach focusing on identifying and defining entities, attributes, and relationships between entities. ER diagrams are commonly used to visualise the relationships and structure of data.
- Dimensional Modelling: This technique is primarily used in data warehousing and business intelligence environments. It involves organising data into dimensions and facts, allowing for efficient analysis and reporting.
- Data Vault Modelling: This method emphasizes scalability and flexibility in data modelling. It involves breaking down data into hubs, links, and satellites, enabling easy integration and adaptability to changing business requirements.
- Ontology Modelling: This approach represents knowledge and concepts by defining classes, properties, and relationships. Ontology modelling is commonly used in semantic data integration and knowledge representation.
- NoSQL Modelling: These methods focus on schema flexibility, scalability, and performance optimization for distributed and document-oriented databases. With the rise of NoSQL databases, data modelling approaches specific to non-relational databases have gained importance.
- Data Lake Modelling: which involves storing and organising large volumes of raw and unstructured data for analysis. It emphasises data exploration and provides flexibility in schema design.

[7] https://en.wikipedia.org/wiki/Data_modeling

In the context of data models, an important component in the domain of data management and interoperability, is the Data Catalog Vocabulary (DCAT)[8]. It is an RDF vocabulary designed for describing datasets and data catalogues on the web. Developed by the W3C Government Linked Data Working Group (2014)[9] as a W3C Recommendation, DCAT provides a standard way to publish structured metadata about datasets, including their characteristics, access methods, and relationships with other datasets.

DCAT is very relevant in the context of data models, because it helps in describing and organising data models within a data catalogue. Data models define the structure, relationships, and constraints of data, and DCAT provides a standardized way to describe and publish metadata about these models. By utilising DCAT, data catalogue publishers can provide detailed information about their data models, making it easier for users to discover and understand the available data resources.

It's worth noting that DCAT is just one of the standards and vocabularies used in the field of data modelling and management; other specifications like from Dublin Core[10], Schema.org[11], and the DCAT Application Profile for data portals in Europe (DCAT-AP)[12] are also relevant in this domain.

2.1.3 Technologies

Regarding the technologies used for data modelling, these vary depending on the specific context and requirements, the type of selected data model (relational, NoSQL, etc.), and the level of complexity and scalability needed for an application/project. However, widely adopted technologies and tools commonly used for data modelling include:

- Entity-Relationship Diagrams (ERDs): which are graphical representations of entities, attributes, and relationships in a database. These are often created using tools such as Lucidchart[13], Visio[14], or draw.io.
- Unified Modelling Language (UML): UML is a standard modelling language that includes a set of diagrams, including class diagrams, which can be used for data modelling in object-oriented systems. UML modelling tools like Enterprise Architect, Visual Paradigm[15], and Sparx Systems[16] are commonly used for UML-based data modelling.

[8] The actual version: https://www.w3.org/TR/vocab-dcat-3/

[9] https://www.w3.org/2011/gld/wiki/Main_Page

[10] https://www.dublincore.org/

[11] https://schema.org/

[12] https://joinup.ec.europa.eu/collection/semic-support-centre/solution/dcat-application-profile-data-portals-europe

[13] https://www.lucidchart.com/

[14] https://www.microsoft.com/en-us/microsoft-365/visio/flowchart-software

[15] https://www.visual-paradigm.com/

[16] https://sparxsystems.com/

- Relational Database Management Systems (RDBMS): RDBMSs, such as Oracle Database[17], MySQL[18], PostgreSQL[19], and Microsoft SQL Server[20], provide built-in support for data modelling using SQL-based technologies. They often include tools and utilities for creating and managing database schemas, tables, and relationships.
- NoSQL Database-specific Modelling Tools: NoSQL databases, such as MongoDB[21], Cassandra[22], and Neo4j[23], often have their own modelling and visualization tools tailored to their specific data models. These tools assist in designing schemas and defining relationships for NoSQL databases.
- Data Modelling in Programming Languages: Some programming languages, such as Python, provide libraries and frameworks for data modelling. For example, the Django web framework[24] includes an Object-Relational Mapping (ORM) layer that allows developers to define data models using Python classes.
- Data Modelling Tools: These are dedicated tools offering advanced features and capabilities for designing, documenting, and managing data models. Examples include ER/Studio[25], ERwin Data Modeler[26], PowerDesigner,[27] and Toad Data Modeler[28].

2.2 Metadata and Metadata Models Management

This sub-section introduces established metadata models, standards, methodologies, repository solutions that are relevant to create and implement the PISTIS metadata models and metadata management.

2.2.1 Metadata Models and Metadata in PISTIS

Metadata models and metadata in PISTIS constitute the very foundation to enable interoperability and discoverability within the PISTIS ecosystem. In PISTIS the metadata serves multiple purposes across all use cases and functionalities:

- Data discovery
- Data processing and reuse
- Enable data interoperability
- Data quality tracking
- Usage tracking and lineage

[17] https://www.oracle.com/database

[18] https://www.mysql.com/

[19] https://www.postgresql.org/

[20] https://www.microsoft.com/en-us/sql-server

[21] https://www.mongodb.com

[22] https://cassandra.apache.org/doc/latest/cql/

[23] https://neo4j.com

[24] https://www.djangoproject.com/

[25] https://www.idera.com/er-studio-enterprise-data-modeling-and-architecture-tools

[26] https://erwin.com/products/erwin-data-modeler/

[27] https://www.sap.com/products/powerdesigner-data-modeling-tools.html

[28] https://www.quest.com/products/toad-data-modeler/

In principle, we can distinguish between four aspects regarding the metadata model and metadata management.

Metadata Models

Metadata models cover all blueprints and schemata that guide and enable the creation of PISTIS-compliant metadata. This may include simple data structure guidelines, complex ontologies, and controlled vocabularies.

Management of Metadata Models

To create harmonized metadata across all services in PISTIS it is paramount to establish a single point of truth for the metadata models. This can be achieved by offering a central repository to manage these models and making them globally accessible and reusable. This includes a machine-readable access, but also a well-documented human-readable version to support developers and data users. Metadata models will evolve and change over time to be adjusted to changing requirements and use cases. Hence, a clear governance and lifecycle process is required. This includes especially versioning and tracking of changes. In addition, a mechanism to synchronise with the upstream of existing models is required.

Metadata

Metadata constitutes a concrete instance of a metadata model. Hence, metadata is created in compliance with a given blueprint. In most cases, the metadata model standard and data structure match the metadata standard.

Metadata models for Data Value

Metadata for data value can help with the description of the various dimensions that compose data value – data quality, data utility, cost, legal and ethical etc. - and sets the basis for quantifying and aggregating these various dimensions, two crucial characteristics in the context of data markets. Without an agreed definition of data value, any vocabulary that describes it should be flexible enough to allow for the addition or removal of metadata. Finally, such a vocabulary should be able to function independently, as well as integrate smoothly within a more complex platform for data exchange, such as PISTIS.

Management and Distribution of Metadata

As the metadata models, the metadata needs to be managed, published and distributed. This can be achieved via various methods. We can distinguish between two general types of repositories: local metadata repositories for managing the metadata within a single organization and distributed repositories for offering a central and consolidated view on the local metadata.

2.2.2 Methods

This sub-section introduces general methods and specifications to represent, model, store and validate metadata.

2.2.2.1 General Methods

There exist multiple general methods to create metadata models, that are not bound to any specific domain or use case.

JSON Schema

JSON Schema is a specification that allows to describe and specify the structure of JSON data. This includes constraints and data types that should be applied to that data. This can be beneficial in a variety of applications - for instance for defining a contract for an API, validating user input in a web form, or ensuring that input data is correctly formatted before it is further processed. JSON Schema itself is expressed in the JSON data format and is created to be easy to read and write by humans and machines. It is an open standard that is widely supported by a plethora of programming languages and frameworks. JSON Schema is based on a set of keywords. E.g, the "type" keyword us used to define the data type of a certain property, such as "string", "number", or "boolean". In addition, keywords like "required" are used to indicate what properties are required, and the keyword "enum" can be applied to define a list of allowed values for a specific property.[29]

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://example.com/product.schema.json",
  "title": "Product",
  "description": "A product from Acme's catalog",
  "type": "object",
  "properties": {
    "productId": {
      "description": "The unique identifier for a product",
      "type": "integer"
    }
  }
}
```

Figure 2: Example of a JSON Shema

XML Schema

XML Schema, also known as XML Schema Definition (XSD) is a specification used to define the structure and constraints of XML documents. It is based on the set of recommendations[30] from W3C. XML Schema describe the elements, attributes and allowed data types in the XML document. It is specified in XML itself and can be used for validation of the XML files.

Knowledge Graphs

Metadata model typically helps people to understand the dependencies and relationships between all attributes in data, and it is possible for data to have multiple dimensions that enable the readers to form various information. Connecting metadata of a big dataset over a knowledge graph can become powerful to help people to gain more knowledge and obtain a more meaningful information, since it shows how data is interconnected.

[29] https://json-schema.org/learn/

[30] https://www.w3.org/XML/Schema#dev

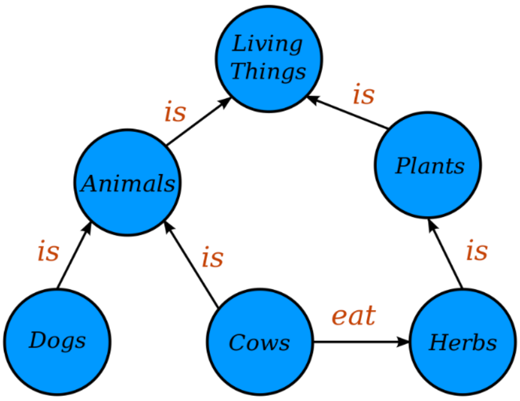


Figure 3: Conceptual diagram example of knowledge graph[31]

Resource Description Framework

The Resource Description Framework (RDF)[32] is a World Wide Web Consortium (W3C) standard originally designed for representing information on the Web in a graph-based format. The core structure of the syntax is an RDF statement, which is a triple consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph. The elements of the triples may be Internationalized Resource Identifiers (IRIs), blank nodes, or datatyped literals. An IRI or literal denote a resource in the world, where anything can be a resource including physical things, documents, abstract concepts, numbers, and strings. IRIs are generalizations of Uniform Resource Identifiers (URIs) that allow a wider range of Unicode characters. One application of blank nodes is when the relationship between a subject node and an object node is n-ary, with n>2. RDF is a base standard for the Semantic Web, which is an extension of the Web that allows data to be shared and linked across applications and organizations. RDF lets diverse data sources be integrated into a single graph, that can be queried and analyzed. RDF can be serialized in a variety of formats, for instance, XML, Turtle, and JSON-LD. It is widely adopted in the scientific and public domain.

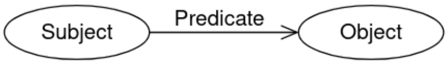


Figure 4: RDF Concept[33]

RDF Schema

RDF Schema constitutes a vocabulary and straightforward ontology language to define the structure of RDF data. RDF Schema allows to create classes, properties, and constraints that can be used to describe the meaning of RDF data. In addition, it provides a way to define hierarchies of classes and properties and constraints on the values that can be used for a certain property. RDF Schema is a crucial building block for many other Semantic Web technologies, such as OWL and SKOS.[34] An example could look like this:

[31] https://en.wikipedia.org/wiki/Knowledge_graph

[32] https://www.w3.org/RDF

[33] https://www.w3.org/TR/rdf12-concepts/

[34] https://www.w3.org/TR/rdf-schema/

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:ex="http://example.org/">

    <rdfs:Class rdf:about="http://example.org/Book"/>
    <rdfs:Class rdf:about="http://example.org/Author"/>
     <rdf:Property rdf:about="http://example.org/hasTitle">
        <rdfs:domain rdf:resource="http://example.org/Book"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </rdf:Property>
     <rdf:Property rdf:about="http://example.org/hasAuthor">
        <rdfs:domain rdf:resource="http://example.org/Book"/>
        <rdfs:range rdf:resource="http://example.org/Author"/>
    </rdf:Property>
 </rdf:RDF>
```

OWL

The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is an ontology language and defines terminology such as classes and properties to be used in RDF data. OWL describes two types of properties: object properties and datatype properties. Object properties define relationships between pairs of resources. Datatype properties specify a relation between a resource and a data type value. In addition, OWL can be used to connect instances. For example, the "sameAs" property is used to state that two instances are identical, which is useful in a distributed environment where multiple identifiers get assigned to the same logical object by different entities. Furthermore, OWL supports to set restrictions on properties and provides features for relating ontologies to each other in cases like importing an ontology or creating new versions of an ontology. A primary goal of the Semantic Web is to describe ontologies in a way that allows them to be reused. However, different applications have different needs even if they function in the same domain and as such might requires slightly different ontologies.[35]

SHACL

SHACL stands for Shape Constraint Language and supports the definition of precise constraints for RDF data. SHACL allows to define rules that particular RDF resources must conform to. These rules are called shapes and can include limitations on the types of values that can be used for a particular property, the cardinality of properties, and the relationships between resources. SHACL can be used to detect and report errors in RDF data that does not conform to the specified rules. SHACL is an essential technology for the Semantic Web, since it provides a method to ensure that RDF data is accurately formatted and conforms to a particular schema or ontology.[36]

```
ex:PersonShape
       a sh:NodeShape ;
       sh:targetClass ex:Person ;    # Applies to all persons
       sh:property [                 # _:b1
```

[35] https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/

[36] https://www.w3.org/TR/shacl/

```
        sh:path ex:ssn ;           # constrains the values of ex:ssn
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
        sh:pattern "^\\d{3}-\\d{2}-\\d{4}$" ;
    ] ;
    sh:property [                 # _:b2
        sh:path ex:worksFor ;
        sh:class ex:Company ;
        sh:nodeKind sh:IRI ;
    ] ;
    sh:closed true ;
    sh:ignoredProperties ( rdf:type ) .
```

SKOS

SKOS stands for Simple Knowledge Organization System and is a W3C recommendation for representing and sharing knowledge organization systems like thesauri, classification schemes, controlled vocabularies and subject heading lists. SKOS provides a standard model for representing concepts and their relationships in a machine-readable manner and is designed to make it easy to publish and share vocabularies for different domains. This helps to improve interoperability between different systems and applications and facilitates the discovery and use of information across different organizations. SKOS itself is based on the RDF data model.[37]

LinkML

LinkML is a versatile modelling language that allows you to create schemas aka models in YAML that describe the structure of data. It supports simple use cases to most complex modelling tasks and is suitable to be used by non-technical domain modelers. LinkML is based on the concepts of classes and slots. Classes represent the entities to be modelled, and slots are reusable properties. LinkML works well with other frameworks, such as RDF-based frameworks and frameworks for JSON modelling. LinkML models can be exported into multiple formats, such as JSON Schema, GraphQL, SHACL, OWL and even code like Java[38]. This makes it a great foundation in projects where a single modelling language is not sufficient enou ￼

```
id: https://w3id.org/linkml/examples/personinfo
name: personinfo
prefixes:
linkml: https://w3id.org/linkml/
  personinfo: https://w3id.org/linkml/examples/personinfo
imports:
  - linkml:types
default_range: string
default_prefix: personinfo

classes:
  Person:
    attributes:
      id:
      full_name:
      aliases:
```

[37] https://www.w3.org/TR/skos-reference/

[38] https://linkml.io

```
    phone:
    age:
```

DCAT

DCAT[39] is an RDF vocabulary designed to facilitate interoperability between data catalogues published on the Web. It enables a publisher to describe datasets and data services in a catalogue using a standard model and vocabulary that facilitates the consumption and aggregation of metadata from multiple catalogues, which can increase the discoverability of datasets and data services. DCAT also makes it possible to have a decentralized approach to publishing data catalogues and makes federated search for datasets across catalogues in multiple sites possible using the same query mechanism and structure. Aggregated DCAT metadata can serve as a manifest file as part of the digital preservation process.

2.2.2.2 Specific Metadata Models

In addition, to general metadata model approaches, more specific ones have evolved. They cover more narrow use cases and domains.

DCAT-AP

The DCAT Application Profile for data portals (DCAT-AP)[40] is a specification based on the DCAT for describing public sector datasets in Europe. Its basic use case is to enable cross-data portal search for data sets and make public sector data better searchable across borders and sectors. This can be achieved by the exchange of descriptions of datasets among data portals. Many data portals in the EU have implemented DCAT-AP for describing data sets in a common way.

GAIA-X Self Descriptions

One Gaia-X[41] added values is the creation of a FAIR (findable, accessible, interoperable, reusable) knowledge graph of verifiable and composable Self-Descriptions (SD)[42], which must parse as a well-formed JSON-LD, an RDF JSON-based serialization. And then the RDF graph defined by this serialization must validate against the SHACL shapes defined by Gaia-X.

Gaia-X Self-Descriptions (SD) describe Entities from the Gaia-X Conceptual Model in a machine interpretable format. This includes SDs for the Participants themselves, as well as the Resources and Service Offerings from the Providers. Well-defined SD Schemas enable ensuring a unified representation of the SDs. The SD allows to find and compare Entities inside Gaia-X. In combination with trustworthy verification mechanisms, SDs empower Participants in their decision-making processes.

[39] https://www.w3.org/TR/vocab-dcat-1/

[40] https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe

[41] https://docs.gaia-x.eu/

[42] https://docs.gaia-x.eu/technical-committee/architecture-document/22.04/self-description/

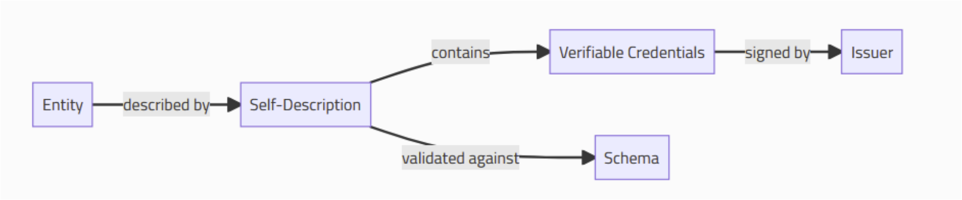


Figure 5: Overview on Self-Descriptions[42]

SDs are W3C Verifiable Presentations in the JSON-LD format. SD consist of a list of Verifiable Credentials, which contain a list of Claims: assertions about Entities expressed in the RDF data model. Both Verifiable Credentials and Verifiable Presentations come with cryptographic signatures to increase the level of trust. Note that the Verifiable Credentials inside a SD may be signed from different (trusted) parties. For example, a certification assessment body may assert a certification result in a Verifiable Credential. This can then be included in a SD for that service.

IDSA Information Model

Clarity about the meaning of data is essential for the International Data Spaces Association (IDSA) to ensure that data can be accurately and consistently interpreted and used by different people and systems [43] . When the meaning of shared data is unclear, it can lead to miscommunication and misinterpretation, resulting in errors and poor decisions. To achieve this clarity and semantic interoperability, IDSA uses semantic technologies such as RDF, SHACL, OWL and SKOS, so that the data can be enriched with meaning by creating links to other datasets and vocabularies.

Frictionless Data

Frictionless Data[44] is an open specification, including software artefacts to enable and simplify the publication and consumption of Open Data. The standard was initiated by the Open Knowledge Foundation and is partly based well-known standards. The main objective is a detailed, semantic and structured description of tabular data. It advocates the use of so-called “data packages”, which contain both, a payload of data and structural information about the data. In general, the specification is domain-agnostic, but a specialization for fiscal data is available. A rich selection of apps, libraries and platforms is on hand in order to create and process data packages, especially in combination with the Open Data platform CKAN[45].

Data Quality Vocabulary (DQV)

This W3C standard[46] provides a framework in which the quality of a dataset can be described. It aims to cover various aspects of data quality such as the quality of data, update frequency, user correction acceptance, and persistence commitments. By providing a vocabulary for expressing quality metadata, DQV allows various actors to evaluate dataset quality and foster trust in published data. DQV defines quality measurements as specific instances of Quality

[43] https://docs.internationaldataspaces.org
[44] https://frictionlessdata.io/
[45] https://ckan.org/
[46] https://www.w3.org/TR/vocab-dqv/

Measurements, adapting the daQ quality framework[47]. Quality dimensions and quality metrics are used to represent quality-related characteristics of a dataset relevant to the consumer and to provide a procedure for measuring data quality dimensions.

However, instead of defining a normative list of quality dimensions and metrics, DQV provides a set of examples of quality dimensions and metrics, allowing implementers to customize these dimensions and metrics to fit their needs, mix existing approaches, or develop their own metrics.

Open Digital Rights Language (ODRL)

The Open Digital Rights Language (ODRL) is a versatile language designed to specify rules and policies for content and service usage. It is based on an information model that covers core concepts and relationships that serve as the semantic foundation for these rules. Policies in ODRL define allowed and restricted actions on assets and stakeholder obligations. These policies can have conditions, such as time or location limits, and may require duties like payments. It is also possible to specify who is responsible for creating and following these policies, along with additional conditions, such as permissions and prohibitions. ODRL supports both content producers, by protecting them against misuse, and consumers, by specifying what they can or cannot do to avoid violations. The language leverages RDF to encode the information. A very simple example can look like this[48]:

```
{
    "@context": "http://www.w3.org/ns/odrl.jsonld",
    "@type": "Set",
    "uid": "http://example.com/policy:1010",
    "permission": [{
        "target": "http://example.com/asset:9898.movie",
        "action": "use"
    }]
}
```

CSV on the Web

CSV on the Web (CSVW) is a specification for creating metadata annotations for CSV files. The overall objective is to guide parsers (and humans) on a correct interpretation. This includes specifying a dialect, which dictates text character reading, and describing a table schema. The table schema ensures that data values are automatically interpreted, saving users the effort of manual data preparation. This results in syntactically correct variable names and data types for each cell[49]. CSVW is again expressed in RDF. An example could look like this, where the first snippet is the input CSV file and the second snippet a CSVW annotation for it.

```
"country","country group","name (en)","name (fr)","name
(de)","latitude","longitude"
"at","eu","Austria","Autriche","Österreich","47.6965545","13.34598005"
"be","eu","Belgium","Belgique","Belgien","50.501045","4.47667405"
"bg","eu","Bulgaria","Bulgarie","Bulgarien","42.72567375","25.4823218"
```

[47] Jeremy Debattista; Christoph Lange; Sören Auer. daQ, an Ontology for Dataset Quality Information. 2014. LDOW 2014. URL: http://ceur-ws.org/Vol-1184/ldow2014_paper_09.pdf

[48] https://www.w3.org/TR/odrl-model/

[49] https://csvw.org/guides/why-use-csvw.html

```
{
  "@context": "http://www.w3.org/ns/csvw",
  "url": "countries.csv"
  "tableSchema": {
    "columns": [{
      "titles": "country"
    },{
      "titles": "country group"
    },{
      "titles": "name (en)"
    },{
      "titles": "name (fr)"
    },{
      "titles": "name (de)"
    },{
      "titles": "latitude"
    },{
      "titles": "longitude"
    }]
  }
}
```

PDL Schema

PDL (Pegasus Data Language)[50] is a schema definition language used for data modelling in LinkedIn's Rest.li framework, an open-source REST framework that enables the creation of robust, scalable RESTful architectures using type-safe bindings and asynchronous, non-blocking IO. PDL schema provides a user-friendly and concise format that replaces the older JSON-based PDSC (Pegasus Data Schema)[51] schema format. Key features of PDL include Java-like syntax, support for import statements, shorthand for custom properties, and cleaner enum declarations compared to PDSC.

Dataset Usage Vocabulary (DUV)[52]

DUV is a vocabulary for describing different uses of data assets. This is useful in the specification and quantification of data usage, as well as in the development of search and matchmaking applications for data assets.

Metadata methods for data valuation contexts

Metadata for data valuation contexts describe the context for the valuation of a data asset, including the purpose of a data asset, possible uses, the industries, and business areas it can impact. Such metadata appear as part of Datasheets for datasets[53], taxonomy of business capabilities determining data value[54].

[50] https://linkedin.github.io/rest.li/pdl_schema

[51] https://engineering.linkedin.com/blog/2020/pegasus-data-language

[52] https://www.w3.org/TR/vocab-duv/

[53] Gebru, T., Morgenstern, J., Vecchione, B., Wortman Vaughan, J., Wallach, H., Daumé III, H., & Crawford, K. (2020). Datasheets for Datasets. *ArXiv:1803.09010 [Cs]*. http://arxiv.org/abs/1803.09010

[54] Hafner, M., & Silva, M. (2023). *Towards a taxonomy for business capabilities determining data value*. https://doi.org/10.21203/rs.3.rs-2609006/v1

Legal and ethical metadata describe the legal aspects surrounding a data asset, such as licensing, intellectual property, distribution rights, whether it contains personal data or other sensitive information, compliance with data protection frameworks (e.g., GDPR) etc. Unified metadata. Such models try to bring together various dimensions of data value under one vocabulary. Several such approaches exist in the form of taxonomies, however most of them cover only a subset of the previous ones: decision making data value (DMDV)[55] (unifies data quality and data utility), data sheets for data sets[56] (while not being a taxonomy, the metadata model it proposes covers all aspects above), EUT Data Valuation Process[57] (covers all these aspects), DataValue (DaVe)[58] (covers all aspects of data value, makes use of current standards (W3C daQ, DCAT), and appears to have the advantage of flexibility).

2.2.3 Technologies and Specifications

This sub-section lists relevant technologies to implement the methods from the previous Section.

2.2.3.1 Metadata Model Tools

Protégé

Protege[59] is a free and open-source ontology editor. It is widely employed in the Semantic Web community for creating and maintaining ontologies. Protege provides a user-friendly interface for creating and editing ontologies and tools for visualizing, querying, and analysing this data. Protege supports many ontology languages, including OWL, RDF, and RDFS. Furthermore, it includes support for reasoning, version control, and collaboration.

Big Data Valuation Platform Metadata

JSON-based vocabulary, covering all dimensions related to data valuation. Its simplicity also makes it easy to extend.

```
{
    "enterprise": {
      "previous_use": "true",
      "added_value": "true",
      "collection_cost": "10",
      "storage_cost": "2",
      "processing_cost": "4",
      "improvement_potential": "true",
      "customer_reach": "true",
      "dept_contribution": "marketing",
```

[55] Lega, M., Colot, C., Burnay, C., & Linden, I. (2022). *Supporting Data Selection for Decision Support Systems: Towards a Decision-Making Data Value Taxonomy*. 492. https://doi.org/10.18293/SEKE2022-104

[56] Gebru, T., Morgenstern, J., Vecchione, B., Wortman Vaughan, J., Wallach, H., Daumé III, H., & Crawford, K. (2020). Datasheets for Datasets. *ArXiv:1803.09010 [Cs]*. http://arxiv.org/abs/1803.09010

[57] Tufiş, M., & Boratto, L. (2021). Toward a Complete Data Valuation Process. Challenges of Personal Data. *Journal of Data and Information Quality*, *13*(4), 1–7. https://doi.org/10.1145/3447269

[58] Attard, J., & Brennan, R. (2019). DaVe: A Semantic Data Value Vocabulary to Enable Data Value Characterisation. In *Enterprise Information Systems* (pp. 239–261).

[59] https://protege.stanford.edu/

```
      "hierarchy_use": "executive"
    },
    "metadata": {
      "provenance": "survey",
      "description": "medical",
      "features_desc": "false",
      "aggregated": "false",
      "unique": "false",
      "transformed": "false",
      "anonymized": "true",
      "encrypted": "false"

    },
    "legal": {
      "contract_bound": "false",
      "user_consent": "true",
      "license": "open",
      "access_difficulty": "easy"
    },
    "applications": {
      "exploration": "true",
      "regression": "true",
      "classification": "false",
      "clustering": "false",
      "supervised": "true",
      "unsupervised": "true"
    }
}
```

Data Value Vocabulary (DaVe)[58]

Covers all dimensions related to data valuation. It is built using RDF schema and reuses the W3C Data Cube Vocabulary, as well as DCAT for identifying and describing datasets.

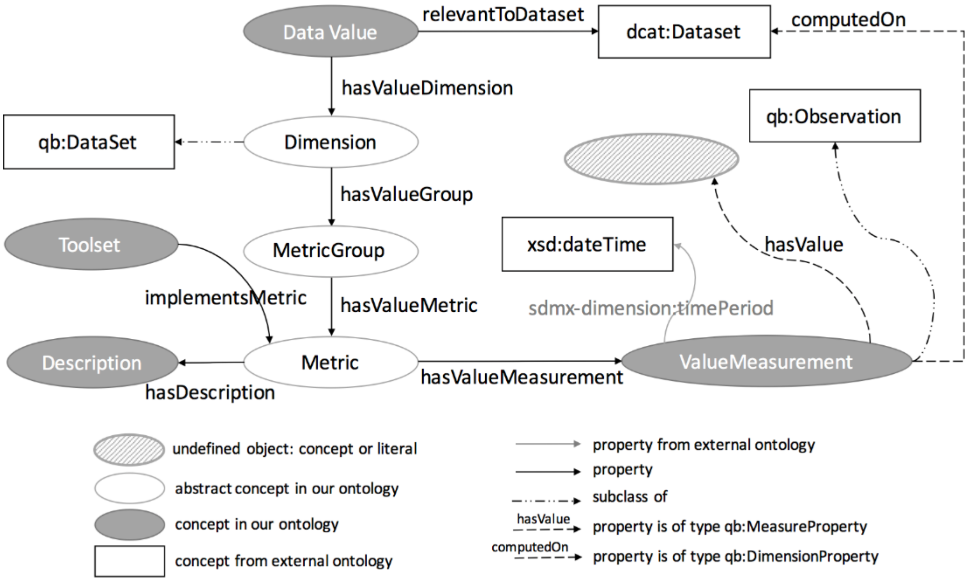


Figure 6: The Data Value (DaVe) Vocabulary[60]

[60] Attard, J., & Brennan, R. (2019). DaVe: A Semantic Data Value Vocabulary to Enable Data Value Characterisation. In Enterprise Information Systems (pp. 239–261).

2.2.3.2 Metadata Repositories

A metadata repository is a location in which metadata is stored and managed. Usually, it is used to help manage and govern data assets by providing a single source of truth for metadata.

piveau

Piveau[61] is an ecosystem for data management designed for the public sector, comprising a range of components. One of these components, Piveau Hub, is an open source metadata catalogue and responsible for metadata cataloguing and search. It was designed and developed using Semantic Web technologies, the W3C standard DCAT and the European standard for Open Data portals DCAT-AP. Piveau puts a strong emphasis on Open Data and is a leading solution for public administrations and non-profit organizations to provide interoperable and flexible Open Data catalogues.

Piveau can be leveraged as a building block for Data Spaces and Gaia-X projects, so that it serves as the Federated Catalogue, Piveau-X, in Gaia-X world. Piveau-X adopts Gaia-X specifications to extend specific features of Piveau and to integrate it into third-party components.

GXFS Federated Catalogue

The Gaia-X Federation Services (GXFS)[62] Federated Catalogue[63] is the first implementation of the GAIA-X federated Catalogue. It aims[64] to (1) enable Consumers to find best-matching offerings and to monitor for relevant changes of the offerings, (2) enable Producers to promote their offering while keeping full control of the level of visibility and privacy of their offerings, and (3) prevent deadlock scenarios from a handful of catalogues instances. The catalogue is used to store and index the data/service assets and resources, as well as to answer queries over this index. The Catalogue is not directly accessed by the end users, but it offers an API, which can be consumed by a portal that offers a graphical user interface for the end users.

Authentication within the Federation is realized by a dedicated component, which is not part of the Federated Catalogue. To demonstrate the integration, the Federated Catalogue offers an Authentication component, based on common, off the shelf software. The Federated Catalogue offers the possibility to verify the set of assets and resources. Therefore, signatures in this set must be verified, which requires trust anchors. According to the Gaia-X Trust Framework, those trust anchors are available in the Gaia-X Registry. When submitting a set of assets and resources, the public key is downloaded. The belonging X.509 certificate is then sent to the Gaia-X registry if it has a valid trust anchor.

[61] https://www.piveau.de/en/

[62] https://www.gxfs.eu

[63] https://gitlab.eclipse.org/eclipse/xfsc/cat

[64] https://gaia-x.gitlab.io/technical-committee/architecture-document/federated_catalogue/

In order to reuse existing technology and to allow scaling, the Federated Catalogue from GXFS consists of multiple components, which can be deployed individually. Figure 7 below shows the catalogue overview, and its components and Table 1 describes each component[65].

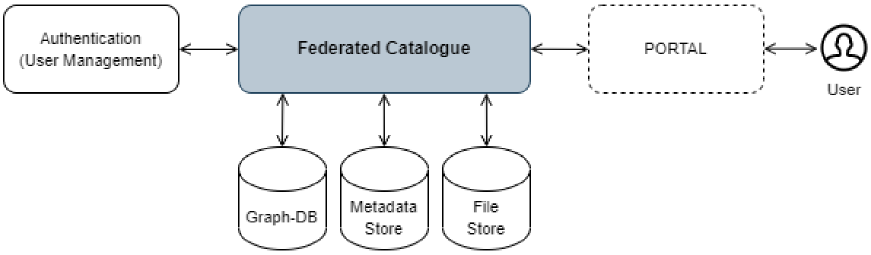


Figure 7: Components of the Catalogue

Table 1: Responsibility of the Catalogue Components

Name	Responsibility
Federated Catalogue	Main component, implementing the core catalogue functionality.
Authentication	External component implementing the authentication flow and user management. Responsibilities: storage of users; storage of user roles for a Participant.
Graph-DB	Graph database, holding all claims contained in active Self-Descriptions[42]. The Graph database is responsible for executing semantic search queries.
Metadata Store	Store for metadata on the Self-Descriptions, and Schemas stored in the File Store.
File Store	The File store is a blob storage. It is responsible to persist all file based content submitted to the catalogue. This includes historical versions of the Self-Descriptions and Schemas.

IDSA

IDSA[66] has a DCAT catalogue protocol that defines how a Catalogue is requested from a catalogue service by a consumer using an abstract message exchange format. All messages must be serialized in JSON-LD compact form. The catalogue protocol is designed to be used by federated services without the need for a replication protocol. Each consumer is responsible for issuing requests to as many catalogue services as they want and managing the results.

Ocean Protocol

Ocean Protocol[67] aims to unlock the value of data and promote data sovereignty by allowing individuals and organizations to have control over their data assets. It provides on-ramp and off-ramp services for data assets to enter crypto ecosystems by using data NFTs and

[65] https://gaia-x.gitlab.io/data-infrastructure-federation-services/cat/architecture-document/architecture/catalogue-architecture.html#_level_1_components_of_the_federated_catalogue
[66] https://internationaldataspaces.org
[67] https://oceanprotocol.com/

datatokens. A comparison can be made to music, where data NFTs are similar to master tapes, and datatokens resemble CDs, which represent base IP and licenses, respectively. A data NFT is a unique ERC721 token that represents the *"base IP"* or exclusive rights of a data service. This can be compared to a music *"master tape"*. A datatoken is a standardized ERC20 token. If you have 1.0 datatokens for a particular data service, it means you have the license to access that service as it provides access to the base IP. This is the equivalent to a music CD.

Each data service gets its own data NFT and zero or more datatokens against that data NFT. The Ocean smart contracts and libraries make it easy for data service providers to create these data NFTs and datatokens and publish their data services. Users can then spend the datatokens to access these data services. This setup keeps Ocean relatively simple and allows for flexibility in using Ocean's tools rather than being restricted to a single platform.

Relational Databases

Relational Databases is a well-known and popular family of databases organising data in tables. Relational Databases are well suited for storing data and metadata, which can be well represented in tabular structures. Storing metadata in a relational database offers several advantages, including data organization, query ability, and data integrity. Relational databases are widely used in various applications and domains. There are many well-known Relational Databases products on the market, including such well-known open source databases as MySQL[68] and PostgreSQL[69].

Document Databases

Document Databases (document store/document-oriented database) are a type of nonrelational database that stores and queries data in the form of a JavaScript Object Notation (JSON) among other data serialization formats. These databases do not require a database schema since the data is stored as structured documents instead of rows and columns. A document is a self-describing record and they make it easier for application developers to store and query data by using the same document-model format that is commonly used in building applications. These documents are also self-describing, which means they contain both data and information about what type of data is stored.

Document databases may not be an ideal choice for every use case, but they offer several benefits such as flexibility, adaptability and scalability by design. They are also quite efficient in managing structured and unstructured data. Some of the most commonly used document databases are, MongoDB[70], Couchbase[71] and Amazon DocumentDB[72].

Graph Databases

Graph Databases are NoSQL databases where data is stored as network graphs. Graphs are an ideal way to model complicated data structures and to represent connectivity among data in such structures. A graph is a collection of nodes, edges and relationships between them. The

[68] https://www.mysql.com

[69] https://www.postgresql.org/

[70] https://www.mongodb.com/

[71] https://www.couchbase.com/

[72] https://aws.amazon.com/de/documentdb/

nodes in a graph are called entities and any connection between these entities is called a relationship. Relationships are identified using unique keys and they contain properties that are used to connect one node to another. This facilitates flexible and efficient querying. Graph databases are the best way to deal with complex, semi-structured and densely connected data. They are quite fast in querying and responses can usually be received in milliseconds. Graph databases are categorized into two groups based on the storage and data model. Data model-based graph databases include property paragraphs, RDF graphs and hypergraphs. Storage based graph databases include native storage graphs, relational storage and key-value store graphs.

Neo4J

Neo4j[73] is a graph database management system that allows the users to store and query data as nodes, relationships, and properties. Nodes represent entities such as people, places, or things, relationships represent the connections between nodes, and properties represent additional information about nodes and relationships. For example, we might have a node representing a person named Jane Doe, with properties for her name and age. We could then have another node representing John Doe, and a relationship between them to represent that they are siblings.

Neo4j supports a query language called Cypher[74], which is specifically designed for working with graph data, and allows the users to perform advanced queries and analytics. It is particularly well-suited for managing data with complex relationships and dependencies, such as social networks, recommendation engines, and network topologies.

Apache TinkerPop

Apache TinkerPop[75] is a vendor agnostic and open-source computing framework for graph databases. It allows for a flexible creation and traversing of various kinds of graphs in the context of both OLAP and OLTP. Gremlin, the provided querying language, naturally supports imperative as well as declarative querying. The software thereby constitutes an alternative to RDF and SPARQL. It is published under Apache 2.0 license.

Triplestore

A triplestore is a type of database, created specifically for storing and querying triples. A triple consists of a subject, a predicate, and an object, and is a fundamental unit of storage in RDF. Triplestores enable sophisticated querying and reasoning about relationships between subjects, making them a foundation in linked data projects, semantic web applications, and ontology-based data storage[76]. In most cases triplestores offer a SPARL endpoint to query the data[77].

[73] https://neo4j.com/product/neo4j-graph-database/

[74] https://neo4j.com/developer/cypher

[75] https://github.com/apache/tinkerpop

[76] https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf-triplestore/

[77] https://www.w3.org/TR/sparql11-query/

DataHub by LinkedIn

DataHub is a metadata search and discovery tool developed by LinkedIn[78], which is described as *"an open-source metadata platform for the modern data stack"*. The tool was developed in response to the challenges faced by LinkedIn employees in discovering and utilizing the large amounts of data available to them. The DataHub system consists of two distinct components: a modular UI frontend and a generalized metadata architecture backend. Since its introduction in 2019, DataHub has been deployed in production at LinkedIn and has been adopted by other companies to manage their data. The current architecture of DataHub is designed to handle large volumes of data in the current era, which is often referred to as the *"golden age of data"*.

OpenMetadata

OpenMetadata[79] is a metadata store that includes functionalities to support data discovery, data lineage, data quality, observability, governance, and team collaboration. It is an evolving open source project with the support of a large community and is also based on Open Metadata Standards and APIs to enable an end to end metadata management. The Metadata Schemas attached to OpenMetadata is based on the foundation of the Open MetaData Standard that defines the abstractions and vocabulary for metadata with schemas for Types, Entities and Relationships between entities. Along with the metadata store to store the metadata graph that connects the data, user and metadata, it also contains an API for creating and retrieving metadata built on schemas. It has an ingestion framework that supports 55 connectors and many well-known data warehouses, databases and pipeline services. It also has a well-developed User interface to help users to discover and collaborate on all data.

Some of the additional features of OpenMetadata includes Data discovery, Data Collaboration, Standardized tests for Data Quality Assessment, Data Lineage, Data governance, Metadata Versioning and Integrations. The Data Discovery feature of OpenMetadata has a full-text search engine that can search among entity definitions, its descriptions, extended metadata and more. To enable data governance, OpenMetadata has a Role-based access control system and on top of it an ownership and importance layer implemented. For Data Lineage application, it has a query parser to collect lineage data along with dbt and data source query logs to create and enrich data lineage. To tackle the issue of data quality, OpenMetadata allows users to group different tests and create a test suite and run it on any selected data assets. Metadata versioning provides valuable information to developers and users when collaboration runs across different data sources.

CKAN

CKAN is an open source data management system mainly for publishing and managing Open Data. It is maintained and developed by the Open Knowledge Foundation The web application is developed in Python and offers a comprehensive frontend for creating, editing and searching a metadata. It employs a PostgreSQL database for storing the data and a Solr search

[78] https://engineering.linkedin.com/blog/2019/data-hub

[79] https://open-metadata.org/

server for efficiently searching the data. In addition, all functionalities are available via a JSON-based API. An extensive plug-in

interface allows the customization of built-in features and the extension with new functionalities. The underlying data structure consists of key-value pairs for representing the attributes of a dataset. CKAN is used for building Open Data platforms. The flat JSON-based data structure is tightly coupled to the technology stack employed. Therefore, an

adoption to different data, especially Linked Data formats and structures is only possible to a limited extent, since CKAN only allows to define custom extra data attributes within the limitations of the JSON standard.[80]

Fedora

Fedora is a versatile, open-source storage system designed for managing and sharing digital content. It is particularly well-suited for digital libraries and archives, facilitating both access to and preservation of materials. Additionally, it offers tailored access to extensive and intricate collections of historical, cultural, and scientific data. A global network of academic institutions, cultural heritage organizations, research centers, and government agencies utilize Fedora. Its ongoing development and community are overseen by the LYRASIS organization[81].

2.3 Distributed Querying

2.3.1 Distributed Querying in PISTIS

The main purpose of this component is to query directly the unstructured or semi-structured data to discover datasets that cannot be retrieved by querying their metadata on the Distributed Data Catalogue. However, the volume of the data stored in the Data Factories does not allow extensive search approaches to be used. Therefore, Locality Sensitive Hashing techniques will be employed to quickly obtain a list of matches. Subsequently, the list of potential matches yielded by the LSH methods will be further evaluated and combined with those returned by the Distributed Data Catalogue. Finally, the merged list will be given as input to a pretrained ML-model that will re-rank it in order to give prominence to the most relevant matches.

[80] https://ckan.org/

[81] https://fedora.lyrasis.org/

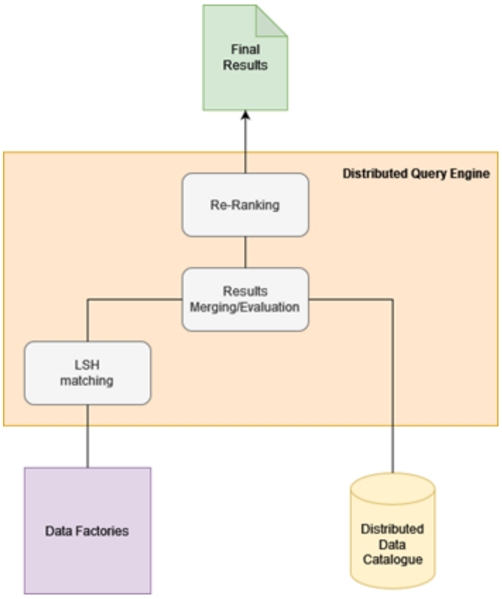


Figure 8: Conceptual architecture of distributed querying in PISTIS

2.3.2 Methods

2.3.2.1 Locality Sensitive Hashing

Locality Sensitive Hashing (LSH)[82] is a powerful technique that addresses the problem of approximate nearest neighbour search in high-dimensional spaces. LSH operates under the principle that similar items tend to hash to the same buckets with a high probability. The key idea behind LSH is to design hash functions that preserve the locality of data points. In other words, nearby points in the input space are more likely to collide in the hash space than distant points. This property enables LSH to efficiently identify approximate nearest neighbours by focusing on the potential candidate items residing in the same hash bucket.

Several LSH algorithms and techniques have been developed to handle different data types and similarity measures. One of the most widely used approaches is based on random projections. In this method, a set of random vectors acts as hash functions. These random vectors divide the input space into multiple regions, and points falling into the same region are mapped to the same hash bucket. By adjusting the number of hash functions and hash buckets, the trade-off between search accuracy and computational efficiency can be controlled.

Variants of LSH have also been introduced to address specific challenges. For example, Multi-Probe LSH improves search quality by exploring neighbouring buckets in addition to the primary bucket. Cascade LSH and Multi-Index Hashing utilize hierarchical structures to enhance the efficiency of similarity search. These advancements have further improved the performance and versatility of LSH in various applications.

Locality Sensitive Hashing finds applications in a wide range of domains where efficient similarity search is critical. Some notable applications include:

[82] https://en.wikipedia.org/wiki/Locality-sensitive_hashing

- Near Duplicate Document Detection: LSH is extensively employed for identifying near duplicate documents in large text collections. By hashing documents based on their textual features, LSH can quickly identify potential duplicates, facilitating content de-duplication, plagiarism detection, and information retrieval systems.
- Image and Video Retrieval: LSH is a popular technique for image and video similarity search. By representing images as high-dimensional feature vectors, LSH can efficiently identify visually similar images or videos. This enables applications such as reverse image search, recommendation systems, and content-based image retrieval.
- Collaborative Filtering: LSH can enhance recommendation systems by efficiently identifying similar users or items. By hashing user or item features, LSH can quickly retrieve nearest neighbours, enabling personalized recommendations and improving system performance.
- High-Dimensional Data Analysis: LSH is valuable in analysing high-dimensional data, such as gene expression data, sensor data, and social network data. By reducing the dimensionality and enabling fast approximate nearest neighbour search, LSH aids in clustering, outlier detection, and data exploration tasks.

Locality Sensitive Hashing offers several advantages that make it an appealing technique for approximate nearest neighbour search:

- Scalability: LSH scales efficiently to large datasets, enabling fast similarity search even in high-dimensional spaces. It reduces the computational complexity by focusing on potential candidates within the same hash bucket, making it suitable for big data applications.
- Approximation Guarantee: LSH provides probabilistic guarantees for approximate nearest neighbour search. By tuning the parameters of the LSH scheme, one can achieve the desired trade-off between recall (probability of finding true nearest neighbours) and precision (probability of avoiding false positives).
- Versatility: LSH can be adapted to various data types and similarity measures. Whether dealing with text, images, videos, or other forms of data, LSH can be customized to capture the inherent similarity structure and efficiently retrieve similar items.
- Robustness to Noise: LSH exhibits robustness to noise and outliers in the data. Since LSH focuses on the local properties of the data, it can still produce accurate results even in the presence of some level of noise.

2.3.2.2 Learn to Rank

In the realm of information retrieval, the ability to effectively rank search results based on their relevance to user queries is of paramount importance. Learn to Rank is a machine learning framework that addresses this challenge by training models to rank items according to their relevance.

The main idea behind Learn to Rank is that the relevance of search results can be learned from labelled training data. It leverages supervised learning algorithms to train models that can predict the relevance of items given a query. These models are trained using a combination of query features, document features, and relevance labels. By learning the underlying patterns and relationships between these features and the relevance judgments, Learn to Rank models can effectively rank search results.

Several algorithms have been developed for Learn to Rank, each with its own characteristics and applicability. Some popular algorithms include:

- Pointwise Methods: Pointwise methods treat ranking as a regression problem, where each item is treated as an independent instance. These methods use the relevance labels as target values and learn a regression function that maps the query and document features to the relevance score.
- Pairwise Methods: Pairwise methods consider pairs of items and learn a ranking function by comparing their relevance. The goal is to determine which item in each pair is more relevant. Pairwise methods learn to directly optimize the pairwise preference between items.
- Listwise Methods: Listwise methods treat the ranking task as an optimization problem, aiming to directly optimize the ranking of the entire list of items. These methods consider the complete list of search results and optimize a list-based loss function.

Learn to Rank offers several advantages that contribute to its popularity in information retrieval:

Relevance Optimization: Learn to Rank enables the optimization of relevance by leveraging machine learning algorithms. By learning from labelled data, these techniques can capture complex patterns and relevance signals that traditional ranking algorithms may overlook.

- Personalization: Learn to Rank algorithms can incorporate user preferences and historical data, allowing for personalized ranking of search results or recommendations. This personalization enhances the user experience and increases user engagement.
- Flexibility and Adaptability: Learn to Rank frameworks offer flexibility in incorporating various features, relevance labels, and learning algorithms. This adaptability allows the models to be tailored to specific domains and optimize for specific metrics or user preferences.
- Continual Improvement: Learn to Rank systems can continuously learn and adapt to changing user preferences and search trends. By incorporating feedback mechanisms and regularly updating the models, these systems can improve their ranking accuracy over time.

2.3.3 Technologies

2.3.3.1 Locality Sensitive Hashing

- **SimHash[83]:** Similarity Hashing or SimHash is a technique used for comparing the similarity between text documents or data points. It aims to produce a compact

[83] Monika Henzinger. 2006. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in

fingerprint, or hash, for each document, enabling efficient similarity calculations and duplicate detection. The core idea behind SimHash is to convert documents into fixed-length hash codes while preserving their similarity relationships. The algorithm has found great commercial success and is used by the Google Crawler to find near duplicate pages. Its main advantages are:

  - Efficiency: SimHash enables efficient similarity calculations, as the hash codes are compact and fixed-length. Comparing the hash codes using Hamming distance is computationally efficient, even for large document collections.
  - Robustness to Noise: SimHash is robust to minor variations or noise in the input documents. Small changes in the text content typically result in a small number of differing bits in the hash codes, thereby preserving the similarity relationships between documents.
  - Scalability: SimHash scales well with the size of the dataset. As the number of documents increases, the time complexity remains relatively low, as the size of the hash codes and the Hamming distance calculations remain constant.
  - Duplicate Detection: SimHash is particularly effective in duplicate detection tasks, where the goal is to identify near-duplicate or plagiarized content. The Hamming distance between hash codes provides a measure of document similarity, allowing for efficient identification of duplicates.

- **MinHash[84]:** short for "Minimum Hashing," is a technique used to estimate the similarity between two sets efficiently. It is commonly applied in data mining and information retrieval tasks, such as document similarity, recommendation systems, and deduplication. The underlying principle of MinHash is to transform sets into shorter signatures while preserving their similarity relationships. It relies on the observation that the probability of two sets having an overlapping element is related to the Jaccard similarity coefficient, which measures the intersection over the union of two sets. MinHash is a widespread technology thanks to its many advantages:
  a. Efficiency: MinHash allows for efficient estimation of set similarity, even for large sets or collections. The size of the signature matrix is typically much smaller than the original sets, making it computationally efficient to perform similarity calculations.
  b. Space Reduction: MinHash significantly reduces the space required to represent sets. The signature matrix is compact, enabling efficient storage and retrieval of set representations.
  c. Scalability: MinHash scales well with the size of the dataset. As the number of sets increases, the time complexity remains relatively low, as the number of hash functions and the size of the signature matrix remain constant.

information retrieval (SIGIR '06). Association for Computing Machinery, New York, NY, USA, 284–291. https://doi.org/10.1145/1148170.1148222

[84] Monika Henzinger. 2006. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06). Association for Computing Machinery, New York, NY, USA, 284–291. https://doi.org/10.1145/1148170.1148222

d. Versatility: MinHash can be applied to various types of data and set representations. It is not restricted to specific domains or data structures, making it a versatile technique for measuring similarity.

- **DB-LSH[85]:** Traditional LSH methods can generate a small number of candidates quickly from hash tables but suffer from large index sizes and hash boundary problems. Recent studies to address these issues often incur extra overhead to identify eligible candidates or remove false positives, making query time no longer sub-linear. To address this dilemma, a novel LSH scheme which supports efficient ANN search for large high-dimensional datasets was proposed in a 2022 paper by Y. Tian et al[85]. DB-LSH organizes the projected spaces with multi-dimensional indexes rather than using fixed-width hash buckets. This approach can significantly reduce the space cost as by avoiding the need to maintain many hash tables for different bucket sizes. During the query phase of DB-LSH, a small number of high-quality candidates can be generated efficiently by dynamically constructing query-based hypercubic buckets with the required widths through index-based window queries. For a dataset of n d-dimensional points with approximation ratio c, DB-LSH achieves a smaller query cost $O(n^{\rho^*}d \log n)$, where ρ^* is bounded by $1/c^{\alpha}$ while the previous bound was 1/c.
- **PM-LSH[86]:** locality-sensitive hashing (LSH) is able to answer c-approximate NN (c-ANN) queries in sublinear time with constant probability. LSH methods focus mainly on building hash bucket-based indexing such that the candidate points can be retrieved quickly. However, existing coarse-grained structures fail to offer accurate distance estimation for candidate points, which translates into additional computational overhead when having to examine unnecessary points. This in turn reduces the performance of query processing. In contrast, PM-LSH framework that was introduced in 2021 by Zheng, B. et al., computes the c-ANN query on large- scale, high-dimensional datasets. First, a simple yet effective PM-tree is used to index the data points. Second, a tunable confidence interval is developed in order to achieve accurate distance estimation and guarantee high result quality. Third, an algorithm on top of the PM-tree is tasked with improving the performance of computing c-ANN queries.

2.3.3.2 Learn to Rank

- **Pointwise Methods:** This family of techniques inherits its members directly from the Machine Learning field. Most notably the regression technologies that are used for dealing with Pointwise Learn to Rank problem are:
    - Support Vector Regression (SVR): Support Vector Regression (SVR) is a machine learning algorithm that applies the principles of Support Vector Machines (SVM) to the task of regression. SVR aims to find a regression function that best fits the training data while minimizing the prediction errors. It achieves this by identifying a hyperplane that maximizes the margin between the training data

[85] Y. Tian, X. Zhao and X. Zhou, "DB-LSH: Locality-Sensitive Hashing with Query-based Dynamic Bucketing," in 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 2022 pp. 2250-2262. doi: 10.1109/ICDE53745.2022.00214

[86] Zheng, B., Zhao, X., Weng, L. *et al.* PM-LSH: a fast and accurate in-memory framework for high-dimensional approximate NN and closest pair search. *The VLDB Journal* 31, 1339–1363 (2022). https://doi.org/10.1007/s00778-021-00680-7

    and the regression function. SVR is particularly effective in handling non-linear relationships and can handle high-dimensional data efficiently. By employing a kernel function, SVR can capture complex patterns and make accurate predictions in various regression tasks.
  - Deep Neural Networks: NNs have shown significant effectiveness in addressing ranking problems. With their ability to learn complex patterns and hierarchical representations from data, deep neural networks excel in capturing the intricate relationships and dependencies inherent in ranking tasks. In the context of learning to rank, deep neural networks can be employed to model the relevance between query-document pairs, item-item similarities, or user preferences. By leveraging multiple hidden layers, these networks can learn high-level features that encode the underlying relevance structure, leading to more accurate and nuanced ranking predictions. Their ability to handle large-scale data, process diverse features, and capture intricate relationships make them a powerful tool for tackling ranking problems and improving the overall quality and relevance of ranked results.
- **Pairwise Methods:**
  - RankNet[87]: was originally developed using neural nets, but the underlying model can be different and is not constrained to just neural nets. The cost function for RankNet aims to minimize the number of *inversions* in ranking. Here an inversion means an incorrect order among a pair of results, i.e. when we rank a lower rated result above a higher rated result in a ranked list. RankNet optimizes the cost function using Stochastic Gradient Descent.
  - RankBoost[88]: is an ensemble method that combines multiple weak rankers to form a strong ranker. RankBoost operates by iteratively training weak rankers on re-weighted training examples, where the weights are adjusted to emphasize difficult instances. Each weak ranker focuses on a specific feature or aspect of the ranking problem. The final ranking is obtained by combining the outputs of all weak rankers based on their individual importance weights
  - RankSVM[89]: is a variation of the traditional Support Vector Machine (SVM) algorithm adapted for ranking problems. RankSVM addresses the challenge of learning a ranking function by directly optimizing pairwise preferences between pairs of instances. It constructs a binary classification problem by considering pairs of instances and learning a ranking function that correctly orders them. RankSVM maximizes the margin between positive and negative

[87] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning (ICML '05). Association for Computing Machinery, New York, NY, USA, 89–96. https://doi.org/10.1145/1102351.1102363

[88] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. J. Mach. Learn. Res. 4, null (12/1/2003), 933–969.

[89] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02). Association for Computing Machinery, New York, NY, USA, 133–142. https://doi.org/10.1145/775047.775067

pairwise comparisons, effectively capturing the relative preferences. By modelling the ranking problem as a classification task, RankSVM is able to learn effective ranking models and has been widely utilized in information retrieval, recommender systems, and other ranking-related applications.

- **Listwise Methods:**
    - ListNet[90]: operates by training a neural network model that takes as input the features of a list of items and predicts the corresponding ranking probabilities. The training objective of ListNet is to minimize the Kullback-Leibler (KL) divergence between the predicted rankings and the true rankings. By modeling the entire list, ListNet captures the global structure of the ranking problem, considering the interactions and dependencies between items. ListNet has shown strong performance in information retrieval, recommendation systems, and other ranking applications, providing accurate and effective ranking results.
    - LambdaRank[91]: Burgess et. al. found that during RankNet training procedure, you don’t need the costs, only need the gradients (λ) of the cost with respect to the model score. You can think of these gradients as little arrows attached to each document in the ranked list, indicating the direction we’d like those documents to move. Further they found that scaling the gradients by the change in NDCG found by swapping each pair of documents gave good results. The core idea of LambdaRank is to use this new cost function for training a RankNet. On experimental datasets, this shows both speed and accuracy improvements over the original RankNet.
    - ListMLE[92]: tries to directly optimize the likelihood of observing the true ranking order of a list of items. ListMLE treats the ranking problem as a probabilistic model and seeks to maximize the probability of observing the true ranking given the input features. It learns a scoring function that assigns higher scores to more relevant items in the list. By considering the entire list and optimizing the likelihood, ListMLE captures the dependencies and interactions among items, leading to improved ranking performance. ListMLE has been widely used in information retrieval and recommendation systems, providing accurate and highly relevant ranking results.

[90] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning (ICML '07). Association for Computing Machinery, New York, NY, USA, 129–136. https://doi.org/10.1145/1273496.1273513

[91] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to rank with nonsmooth cost functions. In Proceedings of the 19th International Conference on Neural Information Processing Systems (NIPS'06). MIT Press, Cambridge, MA, USA, 193–200.

[92] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In Proceedings of the 25th international conference on Machine learning (ICML '08). Association for Computing Machinery, New York, NY, USA, 1192–1199. https://doi.org/10.1145/1390156.1390306

2.4 Data Matchmaking and Recommendation

2.4.1 Data Matchmaking and Recommendation in PISTIS

The PISTIS platform is expected to facilitate the exchange of a variety of data in terms of volume, velocity, type, application domains. Given the expected heterogeneity of the platform's elements (both user intentions and data assets), a user might have difficulties in finding assets according to their search preferences. Matchmaking services are employed to link, as proactively as possible, data providers to data consumers, based on the latter's interest in data assets (e.g., application domain), as well as complementary data assets, which can result in an increase of data value.

The problem can thus be modelled as a recommender system, matching users (participants to the platform) and items (data assets), based on similarity between attributes.

The first challenge is to model the embeddings representing data characteristics, on the one hand, and user preferences and intentions, on the other. For this, we turn to the information coming from two other components of the PISTIS platform:

1. FAIR Data Valuation Services (see Deliverable D3.1[93], Section 3.2) will facilitate multidimensional representations of data assets, including data quality, functional utility, privacy assessments, legal assessments etc. This quantitative information could easily constitute the basis for the embeddings representing the data.
2. Data Usage and Intentions Analytics (see Deliverable D3.1, Section 3.3) will provide statistics about how the data is used. These could be previous interactions between users and data assets, representations of user intentions (e.g., application domain, volume, velocity, GDPR compliance, minimum data quality requirements etc.). All this information could then be remodelled into both user filters and user embeddings.

A second important challenge is to try to model the two types of encodings (items and users), such that there is as much overlap as possible between their dimensions. For example, if data quality is an important dimension representing data assets, then data quality should also be represented within the user preferences (e.g., in the form of minimum data quality requirements). More complicated representations could be those connecting the need of a data consumer (e.g., generate analytics, train ML models etc.) to the properties of a data asset prepared by a different data provider.

Finally, we underline the importance of the interpretability and explainability of recommender systems (XRecSys). XRecSys are part of the field of explainable artificial intelligence (XAI), whose main goal is the development of methods for addressing the lack of interpretability of some of the machine- and deep learning solutions. Model interpretation helps to improve desiderata of ML models, such as transparency, persuasiveness, effectiveness, trustworthiness, and user satisfaction. It also helps system designers to diagnose, debug, and refine the algorithm[94]. Besides from providing personalized content to a set of users, XRecSys

[93] PISTIS Deliverable D3.1 „Data Valuation, Sharing and Trading Framework“, October 2023.

[94] Lipton, Zachary C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue 16.3 (2018): 31-57.

also provide human-understandable explanations of why such recommendations were made. These explanations depend on the kind of recommender system used.

2.4.2 Methods

2.4.2.1 Filtering approach

This is a straightforward application of user preferences to filter those data assets whose encodings match these preferences. The filters can be set on most data-relevant dimensions: volume (minimum or maximum amount of data), data quality (e.g., accepted ranges for completeness, accuracy, timeliness), functional data utility (e.g., accepted ranges for the training predefined ML models), legal compliance (e.g., desired compliance with a subset of requirements in GDPR).

2.4.2.2 RecSys approaches

KNN Item-Based Collaborative Filtering

This method learns from a database (of item features) to classify all data point (items) into several clusters to make inference for new samples. The inference is based on the distance between this new sample and all other items in the database.

Matrix Factorization (MF)

This technique learns a low-dimensional representation of users and items by mapping them into a joint latent space (consisting of latent factors). Recommendations are then generated based on the similarity of user and item factors.

Factorization Machines (FM)

This method is an improvement of the MF technique because it learns not only user and item latent factors, but also the relation between users and items with any auxiliary features.

Deep Neural Networks

The two previous approaches are linear methods. Using a multilayer perceptron (MLP) architecture, we can add non-linear transformations to capture the non-linear (i.e., non-trivial) user-item relationships. Examples are Neural Collaborative Filtering (NCF) and Deep Factorization Machines (DeepFM).

Graph Neural Networks (GNNs)

Graph learning-based methods model the information pertaining to the recommender systems through graph paradigm. Interaction data in a recommender system can be represented by a bipartite graph, in which the nodes are of two types – users and items, while the links are the observed interactions. Applications of link prediction can be used to infer new interactions.

2.4.2.3 Explainable Recommender Systems (XRecSys)

Explaining recommendations through local surrogate models

This is an adaptation to RecSys of the well-known algorithm for local interpretable model-agnostic explanations (LIME)[95]. As a local method, LIME is used to provide instance-based explanations, whereas being model-agnostic means that it can be used as an interpretability extension to any recommendation model. LIME sets to explain an individual prediction, by training an intrinsically interpretable model (e.g., a linear model) in a locality around the instance of interest. This neighbourhood is usually formed by generating perturbations around the instance of interest. However, since introducing unreal data can cause problems with the recommender model (inexistant interactions, fake ratings etc.), LIME for RecSys builds the locality by sampling real data around the point of interest.

Explanation mining

This post-hoc, model-agnostic approach[96] also leverages the interpretability skill of an existing approach – the well-known a-priori algorithm for generating association rules based on frequent itemsets[97]. In a RecSys setting, a transaction consists of a user's history (as input) and the recommended items (as output). The association rules are extracted and used to explain the recommendations of the underlying recommender system. An item is explainable if it gets recommended both by the recommender system, as well as through the association rules.

Interpretable recommendations via overlapping co-clustering

This is a model-intrinsic method[98] for generating interpretable recommendations for the collaborative filtering setting, without additional features. It is based on the detection of co-clusters between users and items. Co-clusters are groups of both users and items with similar patterns. Explanations are generated based on users' collaborative information. E.g., *Item A is recommended to User X with confidence c, because User X also purchased Items B, C, and D, while users with similar purchase history (Users Y and Z) also bought Item A*. If a user-item pair falls into multiple co-clusters, we can thus generate multiple user-based and item-based explanations from each of the co-clusters.

2.4.3 Technologies

scikit-learn (sklearn)

Scikit-learn[99] is a widely used, open-source machine learning library in Python. It serves as a foundational tool for implementing various machine learning tasks, such as preprocessing, feature engineering, model selection and evaluation and seamless integration with other Python libraries. It supports various supervised and unsupervised learning methods. In the

[95] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). 'Why Should I Trust You?': Explaining the Predictions of Any Classifier. ArXiv:1602.04938 [Cs, Stat]. http://arxiv.org/abs/1602.04938

[96] Peake, G., & Wang, J. (2018). Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2060–2069. https://doi.org/10.1145/3219819.3220072

[97] Agrawal, R., & Srikant, R. (1998). Fast Algorithms for Mining Association Rules.

[98] Heckel, R., Vlachos, M., Parnell, T., & Mendler-Dünner, C. (2016). Scalable and Interpretable Product Recommendations via Overlapping Co-Clustering. 2017 IEEE 33rd International Conference on Data Engineering (ICDE), 1033-1044.

[99] https://scikit-learn.org/stable/index.html

context of recommender systems, we can leverage its implementation of K-Nearest Neighbor (k-NN).

pyFM

pyFM[100] is an open-source Python library that provides an implementation of Factorization Machines (FM). The library allows users to build and train FM models efficiently, making it suitable for handling large-scale, sparse datasets commonly encountered in recommendation systems and related tasks.

The pyFM library offers several key features, including easy integration with other popular Python machine learning ecosystems, such as NumPy and sklearn, efficient and scalable handling of large datasets with sparse features, and customizable hyperparameters which can control the complexity and capacity of the FM model.

Microsoft Recommenders

The Microsoft Recommenders Python library[101] is a popular library that focuses on building and evaluating recommender systems. It provides a collection of algorithms and utilities, making it easy for researchers and developers to implement, benchmark, and compare various recommendation techniques. It includes a wide range of recommendation algorithms, such as collaborative filtering, matrix factorization, content-based filtering, and deep learning-based approaches, including state-of-the-art graph neural networks. This diverse set of algorithms allows users to explore and compare different recommendation strategies based on their specific data and use cases. The code is publicly released on GitHub under an MIT License.

Elliot

Elliot[102] is a comprehensive recommendation framework that analyses the recommendation problem from a researcher's perspective. The library aims to switch the focus from coding to designing and parametrising experimental recommendation pipelines, by means of YML configuration files. Elliot untangles the complexity of combining splitting strategies, hyperparameter model optimization, model training, and the generation of reports of the experimental results. The entire experiment is reproducible and puts the researcher in control of the framework. It integrates to date 50 recommendation models, including latent models, deep neural networks and graph-based neural networks. The code is publicly released on GitHub under an Apache2 License.

EUT XRecSys toolkit

Python-based toolkit, providing implementations for the three methods described in the previous sub-section: the LIME adaptation for RecSys, explanations mining and overlapping co-clustering. Even though the toolkit is still in experimental phase and has been tested on limited datasets, the methods it implements can provide valuable explainability extensions for most of the recommender systems discussed in this section. Such extensions can be used to

[100] https://github.com/coreylynch/pyFM

[101] https://github.com/microsoft/recommenders

[102] https://github.com/sisinflab/elliot

increase the transparency of matchmaking services and help developers debug and improve the underlying recommenders.

2.5 AI MODELS MANAGEMENT

2.5.1 AI Models Management in PISTIS

Different use cases are to be covered in the PISTIS project, with many of them working in the creation of predictive models. The creation of a predictive model is the result of a long process that, in short, starts with the raw data, then the data gets pre-processed and prepared for the model training and, finally, it is consumed by the model to get trained. Although this might sound trivial, each step can be carried out in several ways (with a lot of variables such as the data features that will be used, format changes, feature or element removal, model type, model architecture, etc.). As each use case might need to generate several models, it is needed a repository where all these models are stored in order to allow their future use when needed.

2.5.2 Methods

Regarding the methods to address this functionality in PISTIS, there are two main approaches that can be followed:

File repository

This approach relies on the storage of AI models as a file, leaving the responsibility of the maintenance, tracking, deployment and interoperability of the model to the communication between the model creator and the model consumer. This approach might offer a wide adaptability, but it lacks of many AI models-oriented functionalities that could be provided.

MLOPS based solution

By means of this approach, a MLOps layer is set over the file storage repository, providing a full set of functionalities that might enrich the component for its use in PISTIS. Some of these extra functionalities oriented to the management of AI models might include: integration on code when generating the model, ease the tracking of the models (including versioning with a proper labelling of each instance and its performance), mechanisms for model deployment, etc. The most related functionality to be taken into account for the AI Model Management would be the capability to manage artefacts related to the models, that allows embed the previous approach in combination with the aforementioned metadata related to the model creation process (also allowing to define which metadata link with each artefact).

With these approaches in mind, in order to allow the end users to interact with the AI model management Repository, the following functionalities have been identified as desirable in PISTIS:

- **Model registration**
  This method will allow the storage of a new model in the repository.
- **Model Deployment**
  This method will return a model that was previously stored (and has not been deleted).
- **Model update**

This method will overwrite an already existing model with another one.

- **Model deletion**
  This functionality will remove an existing model from the repository.

2.5.3 Technologies

In order to boost the AI model management in PISTIS, some tools have been spotted for their potential inclusion, adaptation and extension to allow the proper management of the predictive model to be used in PISTIS:

MLFlow

This tool[103] is a framework for MLOps, allowing to keep track of different experiments created in order to generate machine learning-based models. This framework also offers the possibility of store those artefacts that might be considered of interest in an experiment, as well as the models generated with their corresponding metrics.

MinIO

MinIO[104] is an open-source cloud-based high-performance S3-compatile object store to store and manage access to files into data lakes. This solution can be used standalone in order to store files or in combination with a MLOps solution (such as MLFlow) in order to be used as storage for the artefacts tracked during the MLOps experimentation process.

3 Data Ingestion & Transformation

3.1 Data Check-In

3.1.1 Data check-in in PISTIS

Data check-in is a key aspect in any technological project. As data is the main good to address in these projects, data consumption and its methods are a crucial point in order to ensure the proper working of any solution developed in this field. One of the key points in the design of a Data Check-in functionality is the support for data sources that will provide data for the solution workflow, as data sources can come in many different ways (repositories, data streaming flows, etc.). As data sharing is one of the key aspects to address in PISTIS, the Data check-in developed in the project needs to be aligned with the different solutions offered in the project, as well as the potential inclusion of other ways to introduce external data into the data processing workflow offered in PISTIS.

3.1.2 Methods

There are the following basic methods for data ingestion:

- File(s) Upload: This method enables the end user to provide a data file to be uploaded to a server to be consumed by the subsequent data processing. Some basic validations

[103] https://mlflow.org/

[104] https://min.io/

could be carried out in order to check some requirements regarding the data file provided (e.g. size limits, data formats, etc.).

- Data Push: This method enables 3rd party applications to proactively push data in the system via the provided by the system API.
- Import from FTP: In case the data to be ingested by the workflow is stored in an FTP server, a method will be provided to retrieve that data. This method should be called providing all the information needed to get access to the data (I.e. endpoint, path to the file, filename, required credentials, etc.).
- Import from API: It is more general version of the previous method where data are imported from an API.
- Import Data from Data Space: In this case, connectors compatible with GAIA-X and IDS could be implemented in order to retrieve data stored in data spaces, providing a similar functionality to the one of the data check in from an ftp server. It will be required, as in the previous case, to get all the details (as well as credentials when necessary) needed in order to get access to the required data source.
- Get Data from Subscription: The possibility to subscribe as a client to a topic is also being evaluated, allowing to get data from this kind of data source (I.e. Kafka or MQTT topics). By means of these, data can be consumed following a given criteria (e.g. defining a time window, a data limit, etc.) and then set for its processing.

3.1.3 Technologies

Due to the generic nature of the data check-in task, the process of data ingestion can be carried out in many different ways, going from the full implementation of the process to the use of technologies that already provide some of the required functionalities. As the data check-in is the starting point of a structured data workflow to be carried out in the PISTIS platform, it would make sense the inclusion of technologies capable to manage or orchestrate data workflows with some data ingestion mechanisms implemented, allowing, in this case, the development of a data check-in solution adapted to the needs of the project and also easing the integration of this process with the subsequent stages of the data workflow. Some technologies that would be integrated in the proposed solution might include:

Apache NiFi

Apache NiFi[105] is an open-source platform in order to manage data workflows orientated to real-time data integration. It allows the creation of module-based graphs to drive the different stages of the data in the ingestion and pre-processing stages. It includes a wide variety of nodes with different capabilities, from data connectors to ETL processing functionalities. Additionally, it can manage with several aspects regarding the characteristics of the data traffic to handle by the component such as latency and throughput, loss-tolerance, data provenance, etc.

Apache AirFlow

[105] https://nifi.apache.org/

Apache AirFlow[106] is another open-source platform to orchestrate data workflows and their scheduling. In this case, workflows are structured as tasks that define the different data processing works to carry out, being able to define dependencies between tasks.

Piveau Consus

Piveau Consus[107] is a component in the Piveau toolchain responsible for the data acquisition from various sources and data providers. It enables individually scheduled data collection/harvesting, transformation and harmonization of data according to rules defined in JavaScript or XSLT. The component is implemented using scalable microservice architecture enabling super scalable processing of up to hundreds of thousands of datasets per source. It supports numerous (meta-)data protocols, sources and formats like OAI-PMH, RDF, CKAN, uData, OwnCloud, JSON, SPARQL, Socrata, Drupal. A flexible configuration-based orchestration enables low-effort inclusion of custom processing steps and even third-party services.

3.2 Data Processing Jobs Configuration

3.2.1 Job Configuration Repository in PISTIS

Due to the wide range of the functionalities involved in the data workflow to implement in PISTIS, a modulable approach, based in different components is being proposed. In order to execute some of the tasks or jobs to be carried out in the aforementioned data processing workflow, it might be needed a component that allows, itself, the definition and automation of these jobs carried out by other components. To that extend, it is needed to define the different steps to follow in each job as well as the conditions required for their execution or the potential dependencies between them. In order to store all the definitions and configurations related to those jobs it is needed a component that checks the proper definition and functioning of the registered jobs, also triggering the execution of the jobs when it is needed.

3.2.2 Methods

For the job configuration repository functionality, it is needed to define the way the jobs will be configured as well as the operations that will be used with those job entities. In this sense, there are two main approaches to follow:

Programmatic job description

With this approach, jobs are to be described following a programmatic approach. This approach allows a wider range of job definitions as it could support any job that might be triggered in a virtual machine.

Workflow orchestration tool-based description

This approach relies on the usage of a main workflow orchestration tool that drives the execution of the whole workflow. This might limit the format of the job definitions allowed, only supporting those formats accepted by the workflow orchestration tool. Nevertheless, the

[106] https://airflow.apache.org/

[107] https://doc.piveau.eu/consus/

orchestration tool might support triggering external jobs, which might broaden the accepted formats depending on the capabilities of the proposed tool.

Additionally, it is important to consider the different functionalities that must be supported by this component, including:

- **Job definition**
  This method should allow the storage of a job definition in the component, checking that the job has been properly defined. This might depend on the approach proposed at the beginning of this section.
- **Job execution**
  This is one of the main functionalities of the proposed component, that would allow the manual (or automatic if the triggering criteria is provided) execution of a job already registered given its id.
- **Job status retrieval**
  The component should provide information regarding a given job execution.
- **Job stopping**
  This method should allow the stopping of a running job given its id.
- **Job listing**
  A list with all the registered jobs in the component should be provided when needed.
- **Running job listing**
  Additionally, a list of all the jobs running in the component in real time should be provided.
- **Job definition removal**
  The deletion of a registered job in the component should be supported as well.

3.2.3 Technologies

For Job configuration, there are several tools that can provide basic functionalities that might be deployed adapted and further developed in order to give response to the needs of the PISTIS project. In this sense, tools such as Apache NiFi or Apache AirFlow (previously described) can provide some functionalities like the ones here introduced. Nevertheless, in order to ease the integration of components developed in a heterogeneous environment like the one proposed in PISTIS, as well as for offer a wider range of technological definitions of the jobs to be managed in the project, some other tools can be taken into account:

NiFi Registry

NiFi registry[108] is an open-source component that allows the storage, governance and management of NiFi data workflows. It allows the definition of jobs as a whole workflow including integrations with external components

Ansible AWX

Ansible AWX[109] is an open-source tool to manage and monitor ansible playbooks, jobs, inventories and credentials. The job definition in this tool is done via YAML files, allowing the

[108] https://nifi.apache.org/registry.html

[109] https://www.ansible.com/community/awx-project

integration of many tools as well as the execution of almost any console-based command, which offers an enormous compatibility with almost any deployable component.

3.3 Data Enrichment

3.3.1 Data Enrichment in PISTIS

Data enrichment refines and enhances datasets to add more value and utilization to the existing data. Typically, data enrichment refers to data harmonization using additional data sources. It combines information from several data sources into a standardized format for further data analysis. The different source of data could be in different file formats, naming conventions and disparate data sources. The main steps included in data harmonization are data cleaning, appending, sorting and aggregating.

Enrichment of data can extend beyond data harmonization using data cleaning and aggregating. Metadata of the data can be involved in this process to extend the actual data. Metadata enrichment is about controlling the onboarding of new data into a standardized data landscape by using domain specific vocabularies. Metadata available in RDF format can be semantically enriched to align with the available semantic data models.

3.3.2 Methods

This section describes the methods that can be used to perform semantic enrichment of the metadata as well as the actual data.

Natural Language Processing

Datasets often resort to a range of controlled vocabularies in the data they contain, that means data values are entered or captured in a controlled way, i.e., for certain positions in a data graph the value used should come with a limited set of pre-existing resources. Such controlled vocabularies (for example EuroVoc[110]) are also available as URI sets on the web. It is important to align the metadata literal values with the controlled vocabularies to increase its interoperability. Values that are not belonging to a controlled vocabulary needs to be replaced by their unique identifiers from the domain reference vocabularies.

Natural Language Processing (NLP)[111] allows machines to learn and comprehend human language. It is a powerful algorithm when it comes to analysing large text or speech data. Among many other applications, NLP can be used to extract, classify and translate text data. It can be used to identify the values not belonging to the controlled vocabulary and replace them with their unique identifiers. A commonly used NLP technique to identify entities is Named Entity Recognition (NER)[112]. Named entity recognition is an NLP technique that can be used to identify entities and map them to the vocabularies. Entities are pre-defined categories, such as person names, organisations, locations, quantities, dates, currencies etc.

[110] https://op.europa.eu/en/web/eu-vocabularies

[111] https://www.ibm.com/topics/natural-language-processing

[112] https://www.turing.com/kb/a-comprehensive-guide-to-named-entity-recognition

When RDF data is represented as Knowledge graphs, entity extraction can be extended to add entity linking. Entity linking adds to NER by identifying which specific entity was recognized. There are several algorithms available to perform Entity Linking which includes both text and graph-based approaches. It is also referred to as Named Entity Disambiguation[113].

Along with entity extraction and linking, NLP can be used for content analysis of the actual data and identifying more metadata properties that would extend the existing metadata. For example, File-type detection, language detection and Natural Language Understanding (NLU)[114] algorithms can produce valuable information about the data. Another aspect of extending the metadata is by making them available in several EU languages. It can be performed using NLP translation libraries or with the EU Machine translation[115], which translates to and from any official EU languages, as well as Arabic, Chinese, Icelandic, Japanese, Norwegian, Russian, Turkish and Ukrainian.

Topic modelling algorithms are powerful content analysis algorithms for text data. It is different to the usual keyword extraction approaches in a way that it provides insights into the hidden context of a text. It is an unsupervised machine learning approach that can find word and phrase pattern in text data and cluster the topic groups that represents a set of data. Topic modelling algorithms can be used for sentiment analysis, to build chatbots and to identify spam emails. Some of the widely used topic modelling algorithms are Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA) and Non-Negative Matrix Factorization (NMF).

Ontology Matching

An ontology typically provides a vocabulary that describes a domain of interest and a specification of the meaning of terms used in the vocabulary. In order to ensure interoperability, it is important that any system that uses multiple ontologies must build semantic mappings of them. Ontology matching[116] or Ontology alignment is a promising solution to the semantic heterogeneity problem. It finds correspondences between semantically related entities or concepts of the ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, or for navigation on the semantic web. Thus, matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate. Several alignment tools exist for this purpose that finds classes of data that are semantically equivalent. This similarity can be expressed as syntactic, external and semantic.

Ontology matching solutions range from elementary matching techniques such as linguistic methods or string-based methods to automatic matching methods using machine learning algorithms. Considering the complexity of the task, it is important for the data users to identify

[113] https://blogs.oracle.com/ai-and-datascience/post/named-entity-disambiguation-with-knowledge-graphs

[114] https://www.sinequa.com/blog/natural-language-processing/natural-language-understanding-the-secret-to-content-analysis/

[115] https://commission.europa.eu/resources-partners/etranslation_en

[116] https://inria.hal.science/hal-00917910/document

what methods are suitable to ensure semantic interoperability while building an ontology matching solution.

Knowledge Graph Completion

Knowledge Graphs[117] are a way to describe entities, concepts and the relationships between the entities and concepts in a structured triple from. It is an ideal way to organize and standardize the vast amount of data available to us. Although it is a widely used approach in many applications, they are mostly a large amount of valuable knowledge missing. Knowledge Graph Completion (KGC)[118] is a widely researched topic which aims at completing the structure of the knowledge graphs by identifying missing entities or relationships. It is a popular technology that will help in inferring missing links and in extending the metadata. It infers new edges in an existing knowledge graph based on the existing data. The methods involving knowledge graph completion includes techniques that relay on structural information and any other additional information.

Important research when it comes to KGC, includes Knowledge Graph Embeddings[119] that embeds the Knowledge Graph components into continuous vector spaces which simplifies the manipulation while preserving the inherent structure of the graph. These embeddings are used in many tasks such as relation extraction, entity classification, entity resolution along with KGC. KGC methods range from conventional approaches where triples are processed independently to graph neural networks where the triples also consider the local neighbourhood[120].

Data Harmonization

Along with performing semantic enrichment of the metadata, data harmonization[121] aims at harmonizing the actual data to improve the data quality and utility. It could be considered similar to data integration, but it also focuses on bringing the data together into a single schema. Data harmonization is in many cases a challenging task. For processes using ETL[122] solutions, it is important to identify the volume, variety of the data source, the structural differences between data sources and the speed in which updates are performed to design an efficient data harmonization process. When it comes to big data sources, AI and machine learning play an integral part here. While performing data harmonization, it is important to understand technologies that perform data quality assessment, data integration, data modelling, data mapping, data governance and data visualization.

The process of data mapping[123] performs matching of fields from one database to another and it serves as an initial step to data harmonization. It is also an integral and crucial part of data management and errors in this step will result in corrupted data and that will ripple through the whole data management process. Before performing the actual mapping

[117] https://dl.acm.org/doi/abs/10.1145/3447772

[118] https://aclanthology.org/2020.acl-main.489.pdf

[119] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8047276

[120] https://arxiv.org/ftp/arxiv/papers/2208/2208.11652.pdf

[121] https://playbook.cd2h.org/en/latest/chapters/chapter_6.html

[122] https://learn.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl

[123] https://segment.com/blog/data-mapping/

procedure, it is important to define the data mapping process, match the source fields to destination fields, define any transformation process that is required and define the frequency of this data transfer. While identifying variables to perform the harmonization, a balance should be identified between finding information that is similar and finding information that corresponds exactly. Some of the example variables that are commonly found in data mapping are, data coming from the same source, geolocation data and time series data.

3.3.3 Technologies

3.3.3.1 NLP Tools and Libraries

Natural Language Processing (NLP) enables machines to interpret and generate human language. NLP is an ever-evolving field of AI thanks to the amount of data available and advances in deep learning algorithms. One of the noteworthy advancements in NLP is the development of large language models such as OpenAI's GPT-4. They are trained on vast amount of text data and are used for applications such as automatic content creation and building virtual assistants. Some of the NLP tools and models that can be used for data enrichment in PISITS are explained below.

Open Semantics Search[124] is an open-source research tool that can be used to search and analyse many documents using a Semantic Search Engine and Open-Source Text Mining and Text Analytics platform. It includes tools for text extraction, entity extraction and named entity recognition along with some specific data enrichment pipelines.

spaCy[125] is a state of the art, python-based, "industry-strength" library for NLP tasks, including NER. Some of the features that make it a good candidate for NER in PISTIS: supports 73+ languages, includes 80+ pre-trained models, as well as the possibility to retrain them or develop new ones, includes a visualiser for NER, easy to integrate and manage within data pipelines.

Apache OpenNLP[126] supports basic NLP tasks, including some relevant to data enrichment in PISTIS, such as NER, language detection and coreference resolution. It offers implementations of both rule-based (potentially more suitable for structured data) and statistical (useful for unstructured data) methods.

[124] https://github.com/opensemanticsearch

[125] https://spacy.io/

[126] https://opennlp.apache.org/

Stanford CoreNLP[127] is a toolkit which allows for the definition of a pipeline of NLP tasks, including NER, dependency parsers and coreferences. The solutions are based on implementations of conditional random fields[128,129].

Gensim[130] is a popular open-source python library used for unsupervised topic modelling, document indexing and similarity retrieval algorithms. It is memory efficient and fast due to its use of large matrix operations by means of its NumPy dependency. Memory efficiency is achieved by using Python's built in generators and iterators for streamed data processing.

3.3.3.2 Ontology Matching/Alignment Systems

Ontology matching[131] is the solution to the semantic heterogeneity problem of large amounts of data. For better understanding, Figure 9 shows two ontologies and an alignment. Rectangles with rounded corners are classes or subclasses. *Book* is a subclass of *Product* and *price* is an attribute which is defined in the *integer* domain and *creator* is a property. *AlbertCamus:La chute* is a shared instance. The thick arrows are correspondence that links an entity from *O1* to another entity from *O2*. The annotation is a relation that is expressed by the correspondence. For example, *Person* in *O1* is less general (⊑) than *Human* in *O2*.

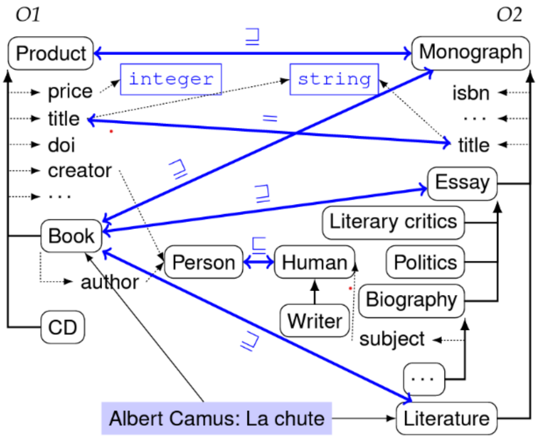


Figure 9: An example ontology alignment

If there is a case in which information sources represented as these ontologies need to be integrated, the first step is to identify the correspondences. Once the correspondences are identified, they can be used to create query expressions that translate instances of these ontologies to an integrated ontology. For example, attributes that are under the label *title* in *O1* and *O2* have the best chances to be merged where the class with label *Monograph* in *O2* will be subsumed by the class *Product* in *O1*.

[127] https://stanfordnlp.github.io/CoreNLP/

[128] Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the Eighteenth International Conference on Machine Learning, 282–289.

[129] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. ACL. https://doi.org/10.3115/v1/P14-5010

[130] https://github.com/RaRe-Technologies/gensim

[131] https://inria.hal.science/hal-00917910/document

To explain the process of the matching operation, consider the following Figure 10. The matching operation is expected to find an alignment *A'* for the pair of ontologies *O1* and *O2*. Some other parameters that can be used to extend the matching task along with using an input alignment *A'* are, the matching parameters, for example weights or thresholds, the external resources, for example domain specific thesauri and common knowledge.

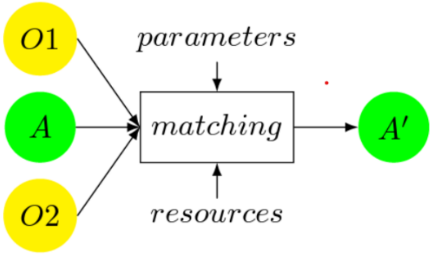


Figure 10: Ontology matching operation

BERTMap[132] is an ontology alignment system that uses the BERT fine-tuning for mapping predictions and uses graphical and logical information in ontologies for mapping refinement. The initial process of this system is a combination of *corpus construction*, where synonym and non-synonym pairs from various sources are extracted and *fine tuning* where a BERT model is selected and fine-tuned on the previously created corpora. It is followed by the process of *mapping prediction*, where the sub word inverted indices are used to extract the mapping candidates and predicted using the BERT classifier and *mapping refinement* where neighboring classes that have the highest scored mappings are used to recall additional mappings along with deleting some mappings that might result in logical inconsistency. This system is licensed under the Apache 2.0 license

LogMap[133] is a highly scalable ontology matching system that can work with semantically rich ontologies that contain tens of thousands of classes. It has reasoning and inconsistency repair capabilities built in its module and it extracts mappings between classes, properties and instances. The ontology obtained after integrating LogMap's output mappings with the input ontologies are clean, consistent and does not contain any unsatisfiable classes. It implements highly optimized data structures for lexically and structurally indexing the input ontologies. It also provides support for user intervention during the matching process using a web interface. It is licensed under the Apache 2.0 license.

3.3.3.3 Knowledge Graph Completion

As explained in the earlier section, Knowledge Graph Completion (KGC) methods predict missing links in knowledge graphs. Data is transformed in a graph representation by converting the knowledge in the data as triplets *(h,r,t*) which represents a relation *r* between head entity *h* and tail entity *t*. KGC methods aim at automatically predicting missing links between these entities. Either it predicts *r* between two already existing entities or predicts the tail entity *t* if the head entity and relation is made available. Traditional KGC methods

[132] https://github.com/KRR-Oxford/BERTMap

[133] https://github.com/ernestojimenezruiz/logmap-matcher

embed entities and relations into a vector space and use a score function to measure the plausibility of the triplet. Recent developments in this field include score functions which utilizes neural networks such as CNNs, RNNs and GNNs which produce competitive results to the previous embedding-based approaches.

Pykg2vec[134] is a python library for Knowledge Graph Embedding (KGE) methods. It is built on top of Pytorch[135] and it can be used for learning the representation of entities and relations in Knowledge Graphs. It combines state-of-the-art KGE algorithms and important building blocks in the pipeline of knowledge graph embedding into one library. It supports state-of-the art KGE model implementations, custom and benchmark datasets. It has tools for inspecting the learned embeddings and can export the embeddings in TSV or pandas supported format. It has an interactive result inspector and supports automatic discovery for hyperparameters. It is licensed under the MIT license.

KG-BERT[136] is a variation of the BERT[137] language representation model specifically for the purpose of KGC. It treats entities, relations and triples as textual sequences and turns the process of KGC into a sequence classification problem. These sequences are further fine-tuned using BERT and used for predicting the plausibility of a triple or a relation. It has the capability to make use of the rich language information present in large amount of text data and highlight the most important words connected to a triple. It is licensed under the Apache-2.0 license.

ANNIF

ANNIF is an open-source automated tool built with a combination of Natural Language Processing and Machine Learning to predict subject indexes for documents that use a controlled subject vocabulary. It learns to predict subject headings to new documents if trained with a subject vocabulary and existing metadata. ANNIF is an impressive solution since all the existing alternate solutions are either expensive or have limited vocabulary and language support. Some of the other open-source automated subject indexing solutions are also hard to integrate with other services as they are implementations of individual algorithms. ANNIF is also available as a web service and can be easily integrated with other systems and can be extended by adding further analysers and subject indexing algorithms.

It is language independent and can be used with any indexing subject vocabulary. It is aimed for improving subject indexing and classification of non-indexed electronic documents. The creators of ANNIF[138] have tested it on several document collections which includes scientific papers, scanned books, e-books, Finnish Wikipedia and archives of local newspapers. It uses machine learning libraries such as Maui, Omikuji, fastText and Gensim and includes a command line interface, web user interface and a Rest API.

[134] https://pykg2vec.readthedocs.io/en/latest/

[135] https://pytorch.org/

[136] https://github.com/yao8839836/kg-bert/tree/master

[137] https://arxiv.org/abs/1810.04805

[138] https://liberquarterly.eu/article/view/10732/11613

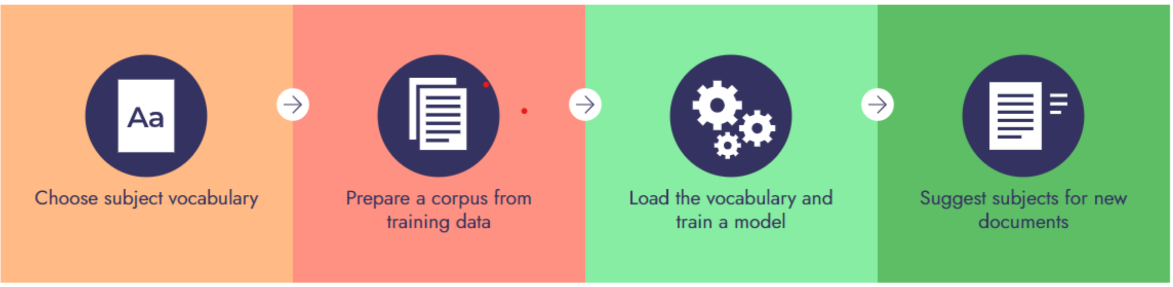

Aa
Choose subject vocabulary
Prepare a corpus from training data
Load the vocabulary and train a model
Suggest subjects for new documents

Figure 11 : How to use ANNIF[139]

3.4 Data Transformation

3.4.1 Data Transformation in PISTIS

The data Transformation Component will offer the capabilities of performing a set of transformation operations on a given dataset. These transformations have to be properly defined in order to be applied over the data, following an ETL approach for pre-processing. Nevertheless, one of the point keys to consider in this component is the application of the different transformations, that should be guided with the aim of improving the dataset quality in mind. To do so, the dependency with the Data Quality Assessment component has to be noted as that component could guide on the different steps to follow during the data transformation process.

3.4.2 Methods

Data transformation embodies a wide set of operations to carry out on a dataset. These operations can be structured into three main categories:

Data Normalization

This kind of transformations aims at the "Homogenization" of data, by using a criterion that should check all the data entries and processing those entries that do not fit the given criteria. Examples of data normalization include providing a similar format to a text (for instance, formatting dates or texts according to given regular expressions) or limiting a numeric field into a range of valid values.

Data Cleaning

The different processing mechanisms considered as data cleaning aim at the removal of "noise" or errors in the data as well as inconsistencies. These processings might include the addressing of missing values, duplicated data entries, typos or values that are not possible or valid.

The Data Quality Assessment (DQA) can inform data cleaning, particularly through the application of data quality dimensions (DQD) such as uniqueness, completeness, format validity and domain validity. As discussed in Section 3.5, this depends on the identification of

[139] https://annif.org/

data quality rules to formalise each DQD. Depending on the selected approach (automatic, manual, semi-automatic), the application of these rules could either be automatically learned from data, defined with the help of metadata or manually specified by a domain expert. While DQA would quantify a dataset's compliance with said rules, they could subsequently be used for data transformations resulting in a cleaner and eventually more valuable data set.

Data Enrichment

For this kind of transformations, it is needed the inclusion of complimentary data sources that, can provide further details on the current dataset. The resultant dataset can be a bigger, more valuable or "richer" dataset, as more information can be extracted from it. A basic requirement for the application of this kind of transformation techniques is that both the current dataset and the additional data sources are interoperable to ensure that the data to be added is good and generates value to the current dataset.

3.4.3 Technologies

Apache Nifi

As defined in section 3.1.3, Apache NiFi[140] provides the capability to generate ETL data workflows, allowing to include nodes for data transformation according to the input definition of this component.

Ansible AWX

This tool[141], also already presented, allows the automatic execution of jobs. A framework that triggers different data transformations based on the current needs of a given dataset could be also a potential solution to evaluate.

Piveau Consus

As presented in the section 3.1.3, Piveau Consus[142] is a component in the Piveau toolchain responsible for the data acquisition from various sources and data providers. It can be configured to perform the data transformation during the data acquisition from a source to a target data model. The transformation rules can be defined in form of XSLT or JavaScript.

3.5 Data Quality Assessment

3.5.1 TData Quality Assessment in PISTIS

The Data Quality Assessment (DQA) and Metadata Quality Assessment (MQA) processes evaluate and quantify the quality of data and metadata within a data asset. The DQA considers the content of the data with dimensions such as completeness, consistency, uniqueness, validity, accuracy and timeliness. The MQA on the other hand ensures efficient dissemination, aggregation and reuse of data within data collections with dimensions such as findability, accessibility, interoperability, reusability and contextuality. DQA and MQA generate

[140] https://nifi.apache.org/

[141] https://www.ansible.com/community/awx-project

[142] https://doc.piveau.eu/consus/

multidimensional scores which reflect the overall data quality. The reporting of this score should include clear explanations of the subsumed dimensions and the aggregation method employed.

DQA is an important part of data valuation. The quantitative outputs from DQA are used as inputs into Data Valuation (DV). Data quality metrics (DQM) can also be used as features for the matchmaking services, allowing users to uncover data assets with a certain data quality profile. The results of the DQA can also be used by the data transformation component: once DQA is initially performed according to a set of data quality rules, those same rules can be applied to transform those data entries which were initially deemed incorrect. Nevertheless, this kind of data quality-guided reconstruction might not be possible across all data entries, therefore DQA can be ran again to measure the quality difference between the 2 states (transformed vs. initial).

Metadata quality is crucial to make data useful in collections of data. A high standard of metadata quality enables easy data discovery and facilitates data governance. Moreover, data lineage can be integrated into metadata. The quality of metadata can also reflect the value of a data asset and should be integrated into DV since it can reveal its usage potential, context and other important information about the data. Hence, metadata quality can become an important component to the functioning of the matchmaking services.

3.5.2 Methods

DQA benefits from a rich body of work, with the first applications dating back to the beginning of the 20th century (in assembly-line production and manufacturing) and initial theoretical advancements between the 1950s – 1970s. The field later followed the evolution of digital technologies in 1990s and 2000s and is currently seeking adapt to the establishment of big data technologies and the increased use of AI-enabled solutions. The concept of DQA is built around several key areas: data quality methodologies, data quality dimensions (DQD) and underlying data quality metrics (DQM).

Quality is often described as the 'fitness for purpose' or in the words of DIN the 'totality of characteristics (and characteristic values) of a unit with regard to its suitability to fulfil specified and presupposed requirements' [143]. Hence, there is no objective, ideal definition of quality but it depends on the requirements of users. In the case of data valuation, proper definitions, and description of purpose (or context) are needed.

In DQA nevertheless, recent literature seems to converge towards six data quality dimensions:

- Accuracy – The degree of agreement between a set of values and another set of values which are assumed to be correct.
- Completeness – Information having all required parts of an entity's description. Typically measured as the proportion of missing values or equivalent.
- Credibility – the values that are different from pre-set default values.
- Uniqueness – the amount of non-duplicate values.

[143] DIN EN ISO 8402 (1995), p. 212

- Validity can have several interpretations. Domain validity requires that all values of an attribute be drawn from a specified domain. Format validity requires data values to respect a certain syntax (format, type, range).
- Timeliness – The time delay from data generation and acquisition to utilization. Timeliness is part of a family of time-related dimensions such as currency, volatility, freshness, and readiness.

In MQA, several frameworks exist that suggest metadata quality dimensions. The FAIR principles[144] describe guidelines for the quality assessment of research data. They comprise the aspects ’findability’, ’accessibility’, ’interoperability’ and ’reusability’. The scope of these principles is further specified in 15 sub-principles that include amongst others the assignment of unique and persistent identifiers, retrievability of metadata by using a standardized communication protocol and usage of usage of a formal, accessible, shared, and broadly applicable language for knowledge representation. Furthermore, they require the presence of detailed provenance and a usage license in metadata. Some frameworks such as the Metadata Quality Assessment Methodology of data.europa.eu extend these principles by ‘contextuality’. Other metrics found in literature are ‘completeness’, ‘accuracy’, ‘consistency’, ‘objectiveness’, ‘appropriateness’, ‘correctness’, ‘conformance’, ‘currency’, ‘intelligibility’, ‘presentation’, ‘provenance’, ‘relevancy’ and ‘timeliness’. Metadata quality dimensions can be grouped into intrinsic (innate correctness), contextual (value to the purpose), representational (ease of understandability) and accessibility (ease of obtainability) metadata quality.

How to perform DQA and MQA?

Quality assessments for Data and Metadata can be implemented in three ways: automatic, manual and semi-automatic.

The automatic approach means that rules and patterns are inferred from data and metadata. They are then applied to the dataset resulting in a quantification (e.g., between 0 and 1) of the dataset against the inferred rules.

For the manual approach a formal definition of each of the DQDs allows us to measure if the data complies to it. This results in a quantification (e.g., [0, 1]) of how much of the dataset is aligned with a certain DQD. Such an approach, while allowing for a very fine-grained definition of the DQM also requires that an exact description of the data quality rules, via a Data Quality Language, is provided.

The semi-automatic method uses ML to automatically learn quality rules and patterns, which can then be manually validated by the user, before DQA is performed.

Aggregation and reporting

These are sensitive aspects when dealing with a process with high-dimensional output. For instance, for a structured or semi-structured dataset, DQA can be performed for each field and then aggregated for the entire data set. However, DQA itself is multidimensional and can be performed in a heterogeneous way across the structure of the data asset (e.g., a user can

[144] https://www.go-fair.org/fair-principles/

decide to measure completeness for Field-1, timeliness and accuracy for Field-2 etc.). Aggregating and reporting all this in a big data context, such that the user gets the relevant insight is an important design challenge.

3.5.3 Technologies

Piveau Metrics[145]

This tool presents a practical and scalable solution to address the challenges of measuring and improving the quality of DCAT Application Profile (DCAT-AP) datasets, essential for Open Data publication and reuse in Europe. Based on the FAIR and 5-star principles, the methodology defines concrete metrics across dimensions like Findability, Accessibility, Interoperability, Reusability, and Contextuality, offering a quantitative representation of metadata quality. The microservice architecture ensures flexibility and extensibility, with a pipeline layer computing metrics and scores, a service layer for further data processing, and a UI layer providing interactive visualizations and detailed reports. Piveau Metrics aims to provide an automated and scalable approach for assessing the quality of DCAT-AP datasets, enabling data providers to improve the quality of their data.

Ydata-quality[146]

An open-source Python library that evaluates data quality throughout the multiple stages of a data pipeline development. The library includes a data quality engine that performs several tests on the input data, and warnings are raised depending on the data quality. It includes specific modules for each dimension, such as Bias and Fairness, Data Expectations, Data Relations, Drift Analysis, Duplicates, Labelings, Missings, and Erroneous Data. By providing prioritized warnings and informative reports, YData Quality enables researchers to proactively identify and address data quality issues, enhancing the reliability and impact of AI solutions.

Apache Griffin[147]

Apache Griffin is an open-source data quality tool that provides a unified platform to measure and monitor the quality of data. The tool supports various data sources and formats, including structured, semi-structured, and unstructured data. Apache Griffin uses a set of predefined data quality metrics that are extensible and customizable to assess and monitor data quality across various data sources and processing pipelines.

Great Expectations[148]

Fully-fledged python-based data validation framework, it functions around the notion of “expectations" with respect to a target dataset. "Expectations” are very similar to unit tests and allow developers to set up individual data checks with respect to various data dimensions, including DQDs: format and domain validity, consistency etc. When an “expectation” fails, it

[145] https://www.piveau.io/en/

[146] https://github.com/ydataai/ydata-quality

[147] https://griffin.apache.org

[148] https://greatexpectations.io/

returns a relevant sample from the database, thus helping with debugging. Besides its basic support for pandas' data frames, it comes with support for SQL databases and Spark data frames. It leverages Jupyter Notebook capabilities, which might impose development constraints.

Pandera[149],[150]

Provides a flexible and expressive python-based API for runtime validation of data frames. Validation rules for completeness, domain and format validity, as well as hypothesis tests are defined in an accompanying validation schema, which is applied on the input data frame. Pandera was initially intended for observation-wise checks on ”wide” data frames but has since been extended for column-wise checks, column-conditional checks and element-wise checks.

Deequ[151]**/PyDeequ**[152]

Deequ is a quality check framework developed in Scala and aimed at Spark data frames; PyDeequ is a python wrapper for PySpark. Deequ is used by AWS, for their own framework - Glue Data Quality[153]. Much like Great Expectations, Deequ is built as a unit test library for data. Its main elements are data quality metrics (e.g., completeness, max, correlation, entropy, uniqueness etc.), constraint checks (e.g., domain or format validity) and constraint suggestions, which provide the capability to infer constraints by analysing the input data. The library exhibits two particularly useful features for big data:

- Evolving datasets allows for the computation of the same set of metrics over various snapshots of a dataset that changes over time. This can prove very useful when working with stream data. However, instead of recalculating the metrics for the entire dataset, Deequ persists them in the form of states, in an HDFS-compatible file system and aggregates them with the metrics computed only on the new incoming data.
- Anomaly detection uses the metrics in previously stored states to analyse abnormal changes in data quality.

One possible drawback is its reliance on (py)Spark, which means that the use of the library is pre-conditioned by loading the input data into a (py)Spark data structure.

IBM Data Quality for AI (DQAI)

DQAI “is an integrated toolkit that provides various data profiling and quality estimation metrics to assess the quality of ingested data in a systematic and objective manner.”[154] It is mostly destined for assessing data prior to entering ML-specific pipelines (I.e., for training ML models). It currently functions for CSV-files, using commas as separators, but further development is planned.

[149] https://union.ai/pandera

[150] https://pandera.readthedocs.io/en/stable/

[151] https://github.com/awslabs/deequ

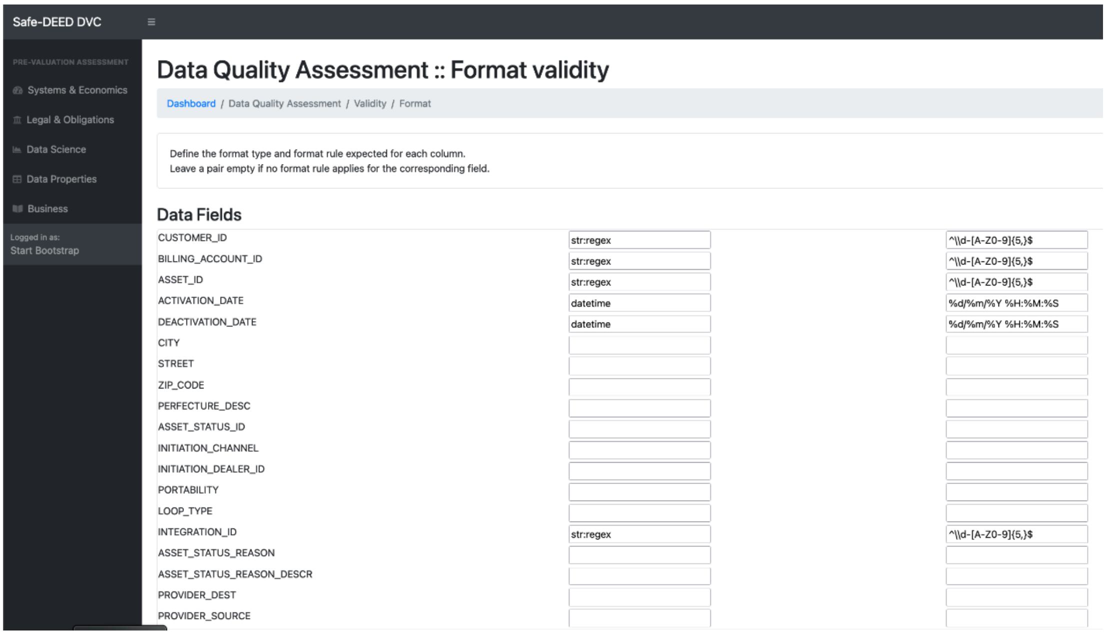
[152] https://pypi.org/project/pydeequ/

[153] https://aws.amazon.com/blogs/big-data/test-data-quality-at-scale-with-deequ/

[154] https://developer.ibm.com/apis/catalog/dataquality4ai--data-quality-for-ai/Introduction

Data Quality Assessment module

The Data Quality Assessment module is part of the EUT Data Valuation Platform and can be classified as a manual DQA tool. It implements metrics for the six DQDs mentioned before (completeness, consistency, uniqueness, validity, accuracy, timeliness). Some of these metrics require the user to define data quality rules to be applied to each field in a structured dataset (see Figure 12). The tool is then automatically applying the rules corresponding to each field and computes a score (between 0 and 1 for each metric and for each column). The score for each metric is the average across all columns. The final data quality score is the average of all six metrics. The scores are reported both metric-wise as well as an aggregate. Future developments of the platform include further formalising of the data quality language for rules, an extension to more data quality metrics, as well as more flexible aggregation metrics (allowing the user to define the relative importance of records or columns).



Data Quality Assessment :: Format validity

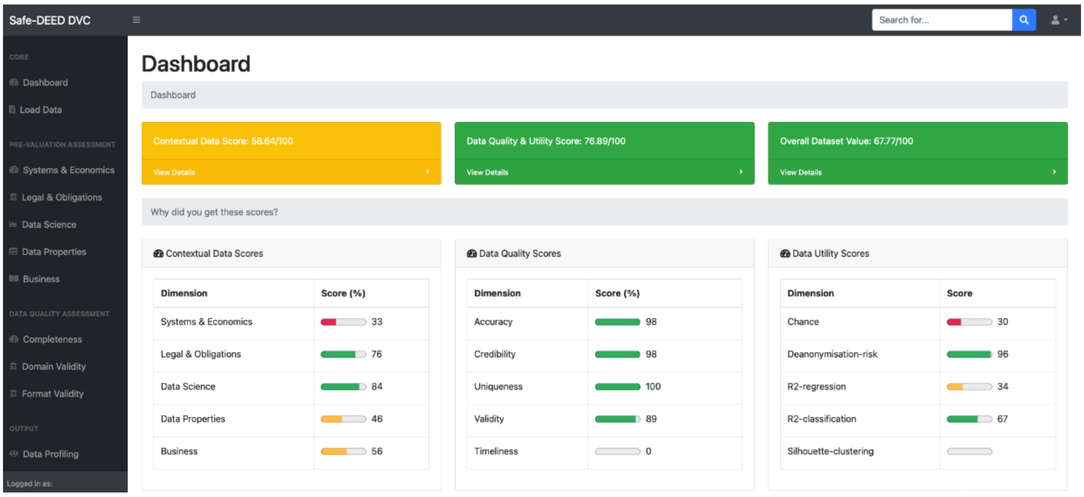
Dashboard / Data Quality Assessment / Validity / Format

Define the format type and format rule expected for each column.
Leave a pair empty if no format rule applies for the corresponding field.

Data Fields

CUSTOMER_ID	str:regex	^\\d-[A-Z0-9]{5,}$
BILLING_ACCOUNT_ID	str:regex	^\\d-[A-Z0-9]{5,}$
ASSET_ID	str:regex	^\\d-[A-Z0-9]{5,}$
ACTIVATION_DATE	datetime	%d/%m/%Y %H:%M:%S
DEACTIVATION_DATE	datetime	%d/%m/%Y %H:%M:%S
CITY		
STREET		
ZIP_CODE		
PERFECTURE_DESC		
ASSET_STATUS_ID		
INITIATION_CHANNEL		
INITIATION_DEALER_ID		
PORTABILITY		
LOOP_TYPE		
INTEGRATION_ID	str:regex	^\\d-[A-Z0-9]{5,}$
ASSET_STATUS_REASON		
ASSET_STATUS_REASON_DESCR		
PROVIDER_DEST		
PROVIDER_SOURCE		

Figure 12: DQA module, part of the EUT Data Valuation Platform.



Dashboard

Dashboard

Contextual Data Score: 58.64/100
View Details ›

Data Quality & Utility Score: 76.89/100
View Details ›

Overall Dataset Value: 67.77/100
View Details ›

Why did you get these scores?

Contextual Data Scores

Dimension	Score (%)
Systems & Economics	33
Legal & Obligations	76
Data Science	84
Data Properties	46
Business	56

Data Quality Scores

Dimension	Score (%)
Accuracy	98
Credibility	98
Uniqueness	100
Validity	89
Timeliness	0

Data Utility Scores

Dimension	Score
Chance	30
Deanonymisation-risk	96
R2-regression	34
R2-classification	67
Silhouette-clustering	

Figure 13: Data aggregation and reporting in the EUT Data Valuation Platform.

3.6 Data Analytics and Insights Generation

3.6.1 Data Analytics and Insights Generation in PISTIS

The Data Analytics and Insights Generation embodies a quite wide functionality to be provided in the PISTIS environment. In order to structure what is expected from this functionality, it has been split into two main results to be provided: on the one hand, the capability of performing analytics jobs to evaluate and assess the characteristics of a given dataset, offering new insights about the dataset being analysed, focusing on the characteristics of the different variables available in the dataset. Moreover, in the same lines, the presence of an analytics engine will allow other modules of the overall PISTIS environment to take advantage of its functionalities, for instance to enable automatic data transformations, to accommodate ML-based anonymisation activities, as well as to run analyses relevant to the PISTIS market, such as trends identifications, predictions, etc.

On the other hand, the same infrastructure could be used by the demonstrators to design models and execute them during the implementation of the pilot scenarios. These demonstrators will be able to use the tools as a data analytics playground to analyse a given dataset and generate results.

While the result of the first functionality of this component aims at offering a descriptive set of insights that will ease the comprehension of the current state of a given dataset, willing to impact on its value, and at the same time facilitate the needs of other modules that can run their models in the same infrastructure, the result of the second offering will provide a completely virtualized and functional environment that brings to the end user the possibility to perform any data processing or generate any predictive model by means of developing any python script in Jupyter notebooks.

3.6.2 Methods

For data analytics and insight generation, the main methods to consider are in line with methods that can offer descriptive characteristics of the datasets to be handled. In this sense, the initial proposed insights cover a set of traditional dataset characteristics including:

Data type description

A description of the data type of each variable analysed (e.g. string, numeric, boolean, etc.).

Variable covered range

In case of numeric variables, describe the maximum and minimum values in the variable.

Missing data presence

A numeric summary of instances in the dataset where each variable has no value given.

Outlier detection

In numeric variables, provide the number of instances in the dataset that are considered as outliers (i.e. data points that differs significantly from other observations) for each variable.

Language recognition

In textual variables, an analysis to identify the language of a given text can be carried out. There are several tools and methods to do that automatically, such as the langdetect library[155] or the langid library[156].

Regarding the AI model editor functionality, the method to be followed must trigger the creation of the virtualized playground according to the definition provided as input to this component, including information related to datasets to be processed in the playground, potential python libraries to include, etc.

3.6.3 Technologies

In order to provide the functionalities here described in the PISTIS platform, the usage of certain technologies has been proposed:

SEAS

SEAS is a customization and deployment tool that given an initial set of datasets returns a customized analytic engine supported mainly by Jupiter Notebook[157] as AI Model Editor, MLFow[158] as Analytics framework and Apache Superset[159] as Visualization framework incapsulating the set of datasets provided as part of the analytics playground. In this way the user has all the ingredients to start in a decoupled environment the execution of any analytic over the dataset provided to generated added value over them in terms of analytic results and metrics.

Jupyter Notebook

This framework allows the end user to create python scripts in a web-based interactive development environment.

MLFlow

MLFlow (presented in section 2.5.3) will allow the end user to keep track of the experiments and models generated while using the AI Model Editor.

MinIO

This tool[160] (also presented in section 2.5.3) is commonly used to support MLFlow by providing data persistence functionality for the artefacts and models generated during the experiment development using the AI Model Editor.

[155] Nakatami Shuyo, 2010. Language detection library for java. https://github.com/shuyo/language-detection/blob/wiki/ProjectHome.md

[156] Lui, Marco and Timothy Baldwin (2012) langid.py: An Off-the-shelf Language Identification Tool, In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), Demo Session, Jeju, Republic of Korea. Available from www.aclweb.org/anthology/P12-3005

[157] Jupiter Notebook: https://jupyter.org

[158] MLFLow: https://mlflow.org/

[159] Apache Superset: https://superset.apache.org/

[160] https://min.io/

3.7 DATA LINEAGE

3.7.1 Data Lineage in PISTIS

The Lineage of a data asset inherits information regarding all the processes it underwent throughout its life cycle starting at its origin. In the context of PISTIS, Lineage Tracking can be utilized to keep detailed track of data flows, including the changes, which the data has been subjected to along its way. The goal is to monitor the evolution of datasets within the PISTIS ecosystem. As the to be developed platform will offer data sharing and –trading services, involved datasets are to be treated as assets, whose quality and security have to be assured in order to gain the trust of users. Throughout their life cycle these assets are being processed and ingested into the ecosystem, frequently consumed and updated by users while also being transferred in between different data spaces. It therefore is essential to incorporate a tracking system into the platform, which will provide the necessary transparency in respect to the detailed changes and movements of the data. This way, errors can be tracked down to their root more easily, ultimately leading to a better data quality. At the same time, the data security can be improved by utilizing the Lineage Tracker as a monitoring tool regarding the data´s access policies. Keeping in mind the platform´s functionality as a data marketplace, the lineage of an asset plays a key role when it comes to dynamic pricing functionalities. And most importantly, Lineage Tracking will assist in keeping the platform compliant in an increasingly regulated world, where data security plays a more and more important role.

3.7.2 Methods

Pattern-Based Lineage Construction

This technique creates data lineages without analysing the code, based on which the data is being created and processed. Instead, it uses metadata as a foundation in order to infer transformation processes connecting different states of the same dataset or relationships between different datasets. E.g., if two different datasets both contain a column with the same name and values, Pattern-Based Lineage will draw a connection between those columns and embed this relationship in the respective Data Lineage Map.

The biggest pro of this approach is it being agnostic towards the implemented database technology, since the basis for its lineage tracking is metadata rather than the involved operations, which can be performed in various languages and dialects, such as PostgreSQL or Sparql. This makes it deployable more flexibly and less dependent on the tech stack of the database system.

On the contrary however, it fails to recognize any type of connection, which is not directly inferable from human readable metadata, making it less powerful than other approaches. In contexts, where more sophisticated data transformation algorithms lead to respectively complex dataset connections, Pattern-Based Lineage Tracking does not constitute a suitable solution.

In PISTIS, the operations, which the data assets are being subjected to may eventually not exceed a critical level complexity, in a sense that the recorded metadata is detailed enough for Pattern-Based Lineage Tracking to work. In order to finally evaluate this method, more details regarding the to be offered data ingestion and –transformation services are needed.

Lineage Construction by Data Tagging

This approach assumes the presence of one central data transformation tool, which is responsible for *all* data transformations being performed within the system and respectively assigns metadata tags to the involved datasets. These tags enable the drawing of data lineages by following them from the latest performed action all the way back to the dataset´s origin.

A big benefit of this approach is how easy it is to implement. Both manual and automatic data tagging are usually easy to integrate into data transformation workflows.

The downside however is that this technique inherits some architectural prerequisites as it assumes every dataset to be processed by one central engine. Every operation taking place outside of this system will not be tracked and hence also not be represented in the corresponding Data Lineage Map.

In PISTIS, each stakeholder hosts their own instance of the Data Space Factory, which is responsible for ingestion-, transformation-, enrichment- and lineage-tracking activities. The actual lineage itself however, will be stored on a publicly accessible ledger. Thus, the architecture requirements of this approach are fulfilled, making this technique a suitable Lineage Tracking method for PISTIS.

Lineage Construction by Parsing

Lineage by Parsing constitutes the most complex approach as it reverse engineers all operations, which the respective dataset has been subjected to in order to construct its Data Lineage Map.

Therefore, this technique, as opposed to the two previously introduced methods, requires a full understanding of the algorithms and programming languages used to process and transform the data. Hence, it is not technology-agnostic and by far less flexible to implement.

On the contrary, once implemented, this advanced method is able to produce the most detailed Data Lineage Maps.

Whether or not Lineage by Parsing constitutes a suitable Lineage Tracking approach in the context of PISTIS is yet to be determined and depends on the heterogeneity of the technological landscape used for data ingestion, -transformation and –enrichment purposes. The more different technologies are being used in this field, the more important a technology agnostic method for Lineage Tracking becomes.

Lineage Marking

Towards facilitating the traceability and provenance of data, lineage markers are essential for specifying and documenting all the different steps that a dataset passes through during its lifecycle, such as its origin, the repositories stored in and its various transformations. These markers constitute specific indicators or metadata that provide information about the complete journey of data within a system or an organization. The metadata, which is being collected by this method serves as a foundation to construct the actual lineage of the respective asset, using either tagging- or pattern-based lineage construction approaches (mentioned above).

Data lineage markers exist in various types based on the context and the technologies exploited. Such lineage markers include:

- Data tags/labels: These are metadata annotations assigned to specific data elements or datasets. They can contain information such as the source system, the date and time of creation, the owner, and any transformations or processes performed on data.
- Timestamps: Time-related information can be used to indicate when specific data were created, modified, or accessed. Timestamps can be used to facilitate a chronological sequence of data events.
- Unique identifiers: A unique identifier, such as a UUID (Universally Unique Identifier), may be assigned to each data element, enabling the tracking and linking of related data across different systems.
- Metadata repositories or catalogues: Centralized repositories or catalogues can store detailed metadata about datasets, including information about their origin, structure, transformations and usage, facilitating in this way the data lineage process.
- Change logs and versioning: A log of changes made to data may be maintained, containing, for example, updates, deletions, additions, etc. Versioning mechanisms can be also utilized to track the different editions of data during their lifecycle.

3.7.3 Technologies

W3C PROV-O[161]

PROV-O is an RDF-ontology published by the World Wide Web Consortium (W3C). It was designed to organize provenance related metadata in a structured way and allows for the creation and continuous extension of an RDF-graph, which consists of numerous interconnected triplets. Each of these triplets reflects the matter of an “agent” performing an "activity” and thereby changing an “entity”. The relationship between different entities can be drawn via the “wasDerivedFrom” attribute (Figure 14).

[161] https://www.w3.org/TR/2013/REC-prov-o-20130430/

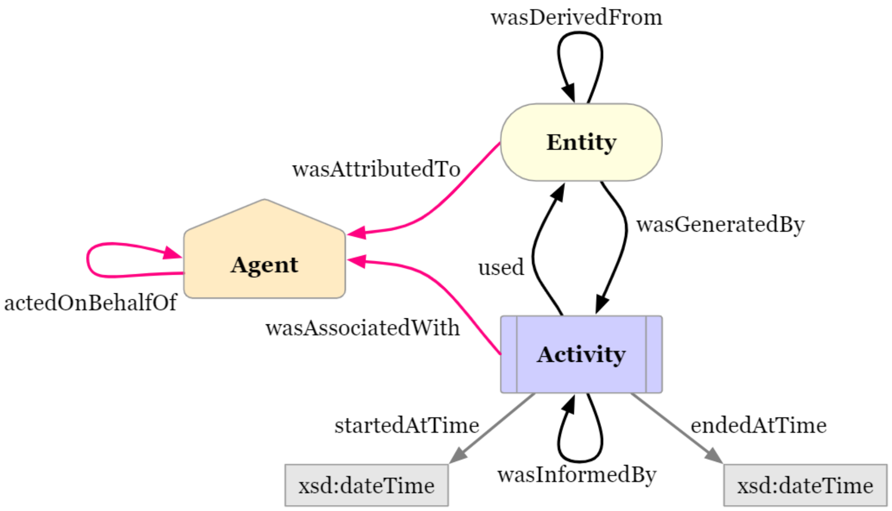


Figure 14 - Visualization of the W3C PROV-O Ontology[162]

Provenance can be referred to as the track keeper of the data´s historical metadata, from which the respective Data Lineage ultimately can be derived. "Agents" can represent users or companies, while activities can refer to all types of operations, which are being performed on different data assets, previously referred to as "entities". In other words: This framework can be used to implement an RDF based documentation mechanism, which keeps track of a dataset's life cycle and serves as a foundation for Lineage Trackers, operating on top of it. A possible storage solution for the generated metadata is the OpenLink Virtuoso triple store.[163]

Python prov[164]

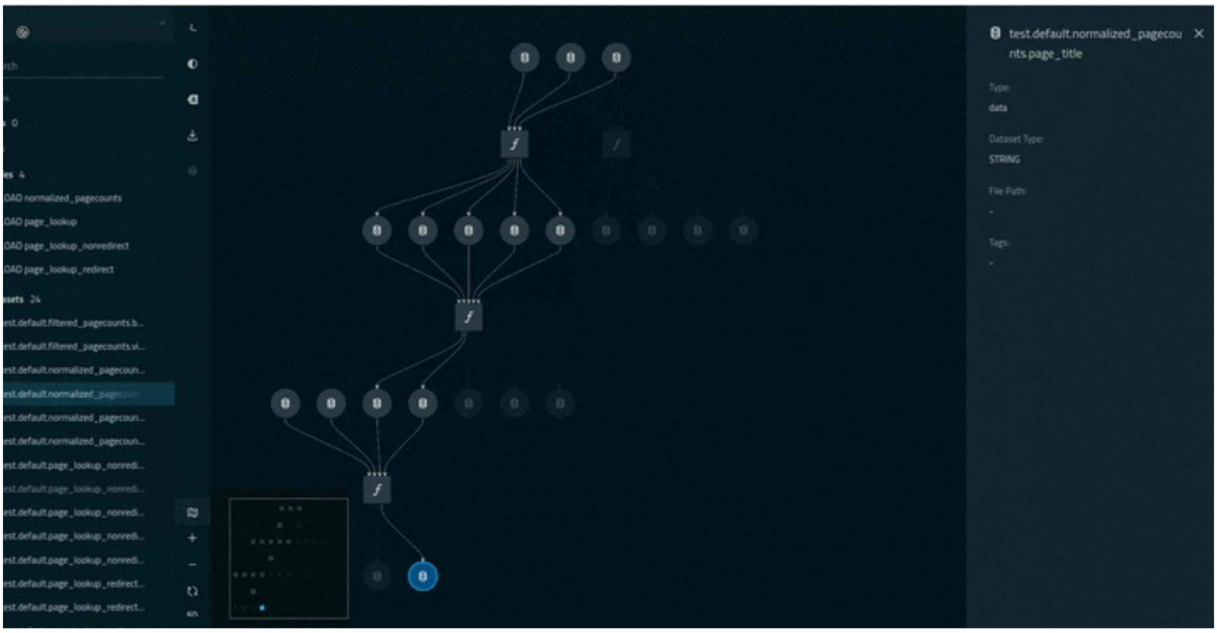
Python prov is a framework, which allows for a flexible creation and management of W3C PROV-O based RDF graphs. It offers an intuitive object-oriented interface with which the user can create and connect triplets based on this ontology. The generated Python code can be converted into TTL-format easily, sparing the user the process of creating these scripts himself. It thereby constitutes a toolbox, which allows for an easy integration of a metadata documentation mechanism directly into Python code. The software is open source and published under MIT license.

Tokern Lineage Engine[165]

Tokern Lineage Engine is a parsing based column-level Lineage Tracking tool, which is compatible with any ETL-framework, integrates with open source data catalogues and is furthermore capable of parsing SQL-query-histories in order to generate data lineages. It

[162] https://www.w3.org/TR/2013/REC-prov-o-20130430/
[163] https://github.com/openlink/virtuoso-opensource
[164] https://github.com/trungdong/prov
[165] https://github.com/tokern/data-lineage

offers its services via API and provides visualization functionalities, which allow for an intuitive and detailed analysis of a dataset´s history (Figure 15).



rch

s 0

es 4

OAD normalized_pagecounts

OAD page_lookup

OAD page_lookup_nonredirect

OAD page_lookup_redirect

sets 24

est.default.filtered_pagecounts.b...

est.default.filtered_pagecounts.vi...

est.default.normalized_pagecoun...

est.default.normalized_pagecoun

est.default.normalized_pagecoun...

est.default.normalized_pagecoun...

est.default.page_lookup_nonredi...

est.default.page_lookup_nonredi...

est.default.page_lookup_nonredi...

est.default.page_lookup_nonredi...

est.default.page_lookup_nonredi...

est.default.page_lookup_redirect...

est.default.page_lookup_redirect...

test.default.normalized_pagecounts.page_title

Type:
data

Dataset Type:
STRING

File Path:
-

Tags:
-

Figure 15 - Tokern Lineage Engine: Visualization of a Data Lineage[166]

The software is open source and published under MIT license.

Open Lineage[167]

Open Lineage is a Lineage Tracking Platform, which records metadata about datasets and executed jobs. It offers an open standard for metadata collection and provides an API through which the respective metadata can be submitted. The generated lineage can be queried via a GraphQL-API or visualized using Marquez[168]. It easily integrates with ETL-frameworks like Apache Airflow[169] or Apache Spark[170] by collecting their produced metadata and providing it in via the Marquez metadata repository (Figure 16).

[166] https://github.com/tokern/data-lineage

[167] https://github.com/OpenLineage/OpenLineage

[168] https://github.com/MarquezProject/marquez

[169] https://github.com/apache/airflow

[170] https://github.com/apache/spark

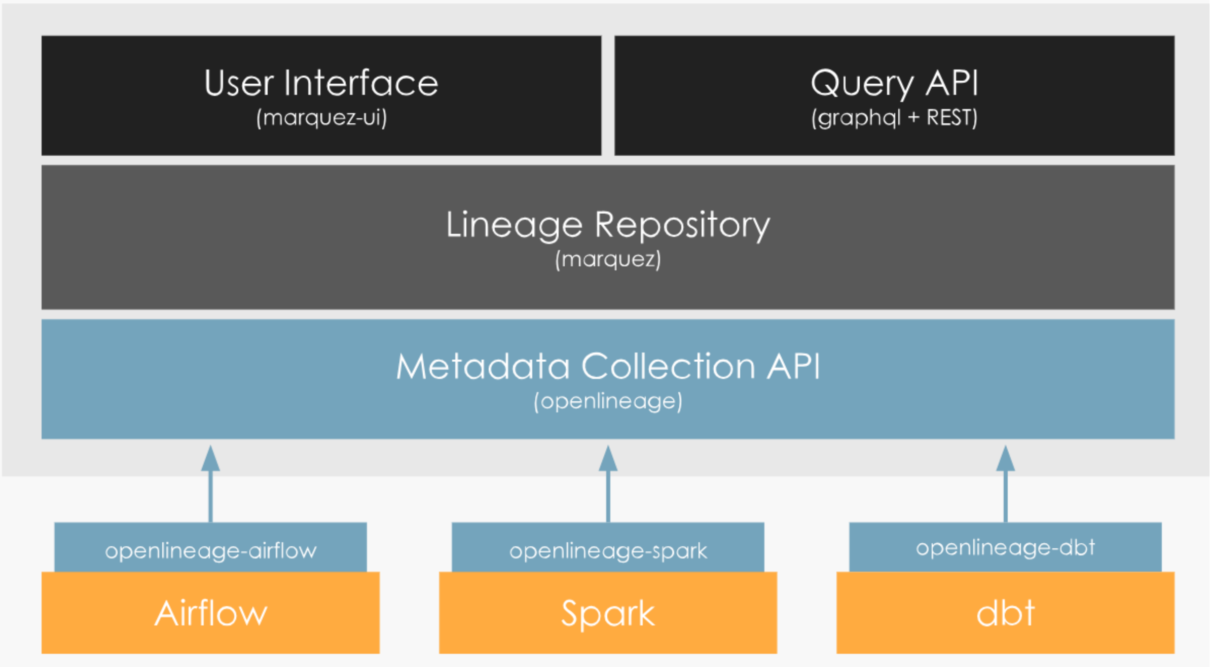


Figure 16 - OpenLineage integration with different ETL frameworks[171]

The related software project is open source and also published under MIT license.

Apache Atlas[172]

Apache Atlas is an open-source data governance and metadata management platform. It provides features for capturing and visualizing data lineage, enabling the tracking of the flow of data across various systems, processes and transformations. It provides an intuitive UI to view lineage of data and REST APIs to access and update lineage information. It also supports integration with various data platforms, including Hadoop, Hive, HBase, etc. Apache Atlas facilitates the definition and management of metadata entities, relationships and classifications, providing an efficient data lineage marking process.

DataHub[173]

DataHub is an open-source metadata platform developed by LinkedIn. It focuses on discovering, capturing and managing metadata related to data assets, providing capabilities for tracking data lineage through its metadata management features. It allows the documentation of data pipelines, datasets and related modifications, facilitating the marking and visualization of data lineage. DataHub enables the establishment of a metadata-driven approach for data lineage marking, by exploiting its metadata management and discovery offerings.

Egeria[174]

[171] https://openlineage.io/

[172] https://atlas.apache.org/#/

[173] https://github.com/datahub-project/datahub

[174] https://egeria-project.org/

Egeria is an open-source metadata and governance framework developed by the ODPi Foundation. It aims to provide a comprehensive solution for managing and understanding metadata across various data platforms, offering features for tracking data lineage, towards visualizing the movement and transformation of data assets. By leveraging its broad metadata management capabilities, coupled with its support for data lineage, Egeria enables an organized and insightful approach towards managing data lineage marking.

3.8 Usage Tracking and Enforcement

3.8.1 Usage Tracking and Enforcement in PISTIS

Usage tracking refers to the process of collecting and monitoring data related to the usage of a particular system, application, or service. Specifically, within PISTIS it involves the collection and analysis of various metrics and events to gain insights into how PISTIS assets are being used by its various stakeholders (users, internal and external systems).

Usage enforcement is a mechanism that is used to control and regulate the way users interact with a particular system, software, or service. Usage enforcement mechanisms are commonly employed in areas such as software licensing, digital rights management (DRM), cloud computing, and network management. Specifically, within PISTIS it implements various measures to ensure that users adhere to the predefined rules and restrictions governing their usage of PISTIS assets (i.e. datasets, real-time data APIs, etc.).

3.8.2 Methods

The typical methods that are used to manage usage tracking and provide valuable information about how PISTIS assets are being used and therefore enabling data-driven decisions regarding usage enforcement, include the following:

- **Event Logging**: The instrumented code generates events or logs that capture information about user interactions, system behaviour, or other relevant actions. These events can include user actions (e.g., clicks, form submissions), system events (e.g., errors, exceptions), or performance-related metrics (e.g., response time, CPU usage).
- **Data Collection**: The generated events or logs are collected and stored in a centralized location such as a database or log management system. This can be done through APIs, message queues, or direct write operations to a storage system.
- **Data Processing**: The collected data is processed to extract meaningful information and derive insights. This involves transforming the raw data into a more structured format, filtering or aggregating data, and applying algorithms or statistical techniques to perform calculations or analysis.
- **Metrics Calculation**: Metrics are calculated based on processed data to measure various aspects of system usage. These metrics can include user engagement metrics (e.g., active users, session duration), feature adoption metrics (e.g., frequency of feature usage), or performance metrics (e.g., average response time).

In lower level of abstraction, these methods can be employed by 4 approaches of usage analysis:

- Transaction Analysis: Transactions on the blockchain can be analysed to track the movement of assets across accounts. This can be done by observing the public ledger and analysing the transaction data. Researchers have developed various methods and tools for analysing blockchain transactions. One such tool is BlockSci[175], a tool for blockchain analysis that has been used in various academic studies.
- Smart Contract Analysis: Smart contracts are self-executing contracts with the terms of the agreement directly written into code. Analysing the execution of smart contracts can help in tracking their usage and understanding how they interact with other contracts and accounts on the network. Various tools and methods have been developed for smart contract analysis. One such tool is Mythril[176], a security analysis tool for Ethereum smart contracts.
- Network Analysis: Analysing the network structure and the connections between nodes can also provide insights into the usage of the blockchain network. Network analysis can help in identifying influential nodes, understanding the propagation of transactions, and detecting anomalies in the network. Several studies[177] have used network analysis methods for analysing blockchain networks.
- Data Analytics and Machine Learning: Data analytics and machine learning techniques can be applied to the data on the blockchain to track usage and detect patterns. For example, clustering algorithms can be used to group similar transactions or accounts together, and classification algorithms can be used to identify fraudulent activities. Several studies[178] have applied data analytics and machine learning techniques to blockchain data for usage tracking and other purposes.

The typical methods that are used to implement usage enforcement can vary depending on the specific system and requirements. Given that usage enforcement mechanisms are commonly employed in areas such as software licensing, digital rights management (DRM)[179], cloud computing, and network management and that usage enforcement in PISTIS aligns with aforementioned areas, the following aspects need to be considered in the usage enforcement methodology:

- **Access Control**: Usage enforcement begins with controlling access to the system or service. This may involve user authentication, authorization, and verification processes to ensure that only authorized individuals or entities can utilize the system.

[175] Kalodner, H., et al., 2017. Blocksci: Design and applications of a blockchain analysis platform. arXiv preprint arXiv:1709.02489

[176] Müller, Bernhard, et al. "Mythril: A framework for bug hunting on the Ethereum blockchain." Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2018

[177] Feng, W., et al. (2019). The future of bitcoin: a comprehensive analysis and research outlook. IEEE Access, 7, 37846-37859

[178] Alessandretti, L., et al. (2018). A first look at the crypto-market meltdown through data science lenses. arXiv preprint arXiv:1902.01941

[179] Cohen, Julie E. "DRM and Privacy." Communications of the ACM 46.4 (2003): 46-49.

- **License Management**[180]: In software licensing scenarios, usage enforcement often relies on license management mechanisms. Data licensing involves creating legal agreements that specify the terms and conditions under which data can be used by others. These mechanisms validate the licenses held by users and ensure that they are using the software within the terms and conditions defined in the license agreement. These licenses can be custom-made or based on existing open data licenses, such as the Creative Commons licenses. By clearly stating the allowed uses, restrictions, and obligations, data providers can enforce the proper usage of their data. This can include limitations on the number of installations, concurrent users, or the duration of usage.
- **Usage Monitoring**[181]: Usage enforcement often involves monitoring user activities to detect any violations or misuse. This can be accomplished through various means such as log analysis, tracking user interactions, or employing dedicated monitoring tools. Monitoring helps identify unauthorized usage, excessive resource consumption, or attempts to circumvent the enforcement mechanisms.
- **Compliance Checks**: Usage enforcement systems perform regular compliance checks to verify that users are adhering to the established rules. These checks may involve analysing usage patterns, comparing against predefined thresholds, or applying predefined policies. If violations are detected, appropriate actions can be taken, such as issuing warnings, imposing restrictions, or even terminating user access.
- **Enforcement Actions**: In response to detected violations or non-compliance, usage enforcement mechanisms can take proactive actions to enforce the rules. This can include restricting access, reducing functionality, or applying penalties. Enforcement actions can be automated or manual, depending on the severity of the violation and the system's capabilities.
- Secure Data Enclaves[182]: Secure data enclaves are secure computing environments where researchers can access and analyse sensitive data without the ability to download or remove the data from the secure environment. This approach allows data providers to control and monitor the usage of the data, ensuring that it is used appropriately and for the agreed-upon purposes.

3.8.3 Technologies

In web based distributed system usage tracking is realized with the methods already mentioned and implemented with various tools. These tools fall under several categories which differentiate on the objectives and the granularity of data collection. The most essential of the tools that are used for usage tracking are:

- Distributed Tracing:

[180] Ribeiro, A. M., & da Silva, J. M. N. (2020). Data sharing in academia: Benefits, barriers and the role of data management. Information Services & Use, 40(1-2), 29-43.

[181] Habermann, S., et al. (2019). Monitoring and assessing data quality in large scale data sharing. Biometrical Journal, 61(2), 375-389.

[182] van Panhuis, W. G., et al. (2020). A manifesto for the use of routinised data by researchers in low and middle-income settings. BMJ Global Health, 5(11), e003631.

  - Jaeger[183]: open-source, end-to-end distributed tracing system used for monitoring and troubleshooting the performance of microservices-based applications. Jaeger enables the collection of distributed traces, which are records of the paths that requests take as they traverse through various components of a distributed system. Each trace consists of one or more spans, representing individual operations within the system.
  - Zipkin[184]: Another open-source distributed tracing system. Zipkin provides a way to trace requests as they move through different services in a distributed architecture. Each trace consists of a collection of spans, which represent individual units of work in the system. Spans are linked together to form a complete trace, showing the journey of a request.
  - OpenTelemetry[185]: OpenTelemetry is an open-source project that provides a set of APIs, libraries, agents, and instrumentation to enable observability in cloud-native and microservices-based applications.
- Log Management and Analysis:
  - Elastic Stack (ELK: Elasticsearch, Logstash, Kibana)[186]: For collecting, storing, and visualizing logs. The ELK Stack is widely used for log and event management, allowing organizations to centralize their log data, analyse it, and gain valuable insights into their systems and applications.
  - Graylog[187]: Graylog is an open-source log management and log analysis platform designed to help organizations collect, store, and analyse log data from various sources. It provides a centralized solution for managing log files, monitoring system behaviour, and extracting valuable insights from log data.
  - Splunk[188]: Splunk is a powerful and widely used software platform designed for searching, analysing, and visualizing machine-generated data. It's often used for log and event data analysis, but it can also handle a wide range of data types, making it versatile for various use cases. Splunk is especially popular for its ability to help organizations gain insights from large volumes of data, monitor system performance, and detect security threats.
  - Fluentd[189]: Fluentd is an open-source data collection and log forwarding tool that simplifies the process of collecting, processing, and forwarding log and event data from various sources to different destinations. It is designed to be lightweight, flexible, and highly extensible, making it a popular choice for log management and data integration tasks in cloud-native and containerized environments.
  - Collectd[190]: Collectd is an open-source system statistics collection daemon that runs on Unix-like operating systems. It is designed to periodically collect various system

[183] https://www.jaegertracing.io/

[184] https://zipkin.io/

[185] https://opentelemetry.io/

[186] https://www.elastic.co/elastic-stack

[187] https://graylog.org/

[188] https://www.splunk.com/

[189] https://www.fluentd.org/

[190] https://collectd.org/

  - and performance-related metrics and statistics from the host it is installed on. Collectd is commonly used in system monitoring and performance analysis to gather data on system health and resource utilization.
  - Prometheus[191]: Prometheus is an open-source monitoring and alerting toolkit designed for collecting and analysing metrics from software systems and services. It is part of the Cloud Native Computing Foundation (CNCF) and is commonly used in cloud-native and containerized environments.
  - Grafana[192]: Grafana is an open-source, web-based platform for monitoring, visualization, and observability often used in tandem with Prometheus. It is commonly used by organizations and individuals to create interactive and customizable dashboards for visualizing and analysing data from various sources, including monitoring systems, databases, applications, and more. Grafana is highly extensible and supports a wide range of data sources and visualization options.
- Web Analytics:
  - Google Analytics[193]: Google Analytics is a web analytics service provided by Google that allows website owners and marketers to track and analyse the performance and user behaviour of websites and mobile apps. It provides valuable insights into how users interact with a website or app, helping businesses make data-driven decisions to improve their online presence and user experience.
  - Matomo[194] (formerly Piwik): Matomo is an open-source web analytics platform that provides website owners, businesses, and organizations with tools to track and analyse user behaviour on websites and applications. Matomo distinguishes itself from many other web analytics services by emphasizing user privacy and data ownership.
  - Mixpanel[195]: Mixpanel is a product analytics platform that helps companies track and analyse user behaviour within their web and mobile applications. It provides tools for measuring user engagement, conversion rates, and the effectiveness of features and marketing campaigns.
- Other: Additional tools are used for purposes like application performance and monitoring (APM like New Relic[196], Dynatrace[197], Datadog[198]), error reporting and exception tracking (e.g. Sentry[199], Rollbar[200]) and infrastructure monitoring (e.g. Splunk[201], Zabbix[202]) can be used to complement the usage tracking data collection.

[191] https://prometheus.io/

[192] https://grafana.com/

[193] https://analytics.google.com/analytics/web/?pli=1

[194] https://matomo.org/

[195] https://mixpanel.com/

[196] https://newrelic.com/

[197] https://www.dynatrace.com/

[198] https://www.datadoghq.com/

[199] https://sentry.io/

[200] https://rollbar.com/

[201] https://www.splunk.com/

[202] https://www.zabbix.com/

However, distributed systems nowadays are a very large ecosystem of architectures and technologies with different inherent characteristics. This variety of tools and technologies introduces several restrictions on what can be used in each case. PISTIS, explores and incorporates technologies that come from the distributed ledgers domains and particular blockchains. Event logging and usage tracking in blockchain ecosystems differ from traditional systems due to the inherent transparency of public blockchains, the pseudonymous nature of transactions, and the decentralized infrastructure.

Regarding the methods reported for usage tracking, blockchain technology required the development of tools that match the peculiarities of the technology. With that respect the main approaches, technologies and tools for event logging and data collection for usage tracking are:

- Blockchain Explorers: These are tools that provide a user-friendly interface to query blocks, transactions, and other blockchain-related data.
    - Etherscan[203]: Etherscan is a popular blockchain explorer and analytics platform specifically designed for the Ethereum blockchain. It provides users with tools to explore, search, and analyse the Ethereum blockchain and its associated data. Etherscan is widely used by Ethereum developers, traders, and enthusiasts to access real-time information and historical data about Ethereum transactions, smart contracts, addresses, and more.
    - Blockchair[204]: Blockchair is a blockchain explorer and analytics platform that provides users with tools to explore, search, and analyse data from multiple blockchain networks. It is designed to offer insights and information about various blockchain transactions, addresses, and blocks. Blockchair supports several prominent blockchain networks, including Bitcoin, Ethereum, Bitcoin Cash, and Litecoin.
    - Blockscout[205]: BlockScout is an open-source blockchain explorer and analytics platform designed specifically for the Ethereum ecosystem. It provides users with tools to explore, search, and analyse data from Ethereum-based blockchains. BlockScout offers detailed information about Ethereum transactions, blocks, smart contracts, tokens, and more.
- Node and Network Monitoring:
    - Run full nodes or specific tools to gather more granular data about the network, block propagation, orphan rates, etc.
    - Geth[206] and Parity[207] for Ethereum, Bitcoin Core[208] for Bitcoin.
- Smart Contract Events (for platforms like Ethereum)[209]:

[203] https://etherscan.io/

[204] https://blockchair.com/

[205] https://www.blockscout.com/

[206] https://geth.ethereum.org/

[207] https://github.com/openethereum/parity-ethereum

[208] https://bitcoincore.org/

[209] https://ethereum.org/uz/developers/tutorials/logging-events-smart-contracts/

  - Smart contracts can emit events that log specific actions or state changes. These events can be easily monitored and filtered using libraries like Web3.js [210] or ethers.js[211].
- API Services:
  a. Many blockchain analytics firms and explorers offer APIs to programmatically pull transaction data and integrate into custom tools. Examples: APIs from Etherscan, Blockchair, and Chainalysis[212].

Usage tracking is commonly paired to usage enforcement. Based on the principles already described in previous section, usage enforcement in distributed systems involves the implementation of various tools and technologies to manage, control, and monitor resource usage, access, and compliance with policies and regulations.

The most common tools and technologies used for usage enforcement in distributed systems are:

- Access Control Lists (ACLs): ACLs are commonly used to control access to resources within distributed systems. They specify which users or entities have permissions to access specific resources.
- Role-Based Access Control (RBAC): RBAC tools help enforce access control policies by defining roles and associating users or entities with these roles. Roles determine the level of access a user has within the system.
- Identity and Access Management (IAM) Services: IAM services, like AWS IAM[213] or Azure Active Directory[214], provide centralized user authentication and authorization management for distributed systems and cloud environments.
- API Gateways: API gateways, such as NGINX[215], Kong[216], or Apigee[217], enforce usage policies related to API access, rate limiting, and security. They can also provide authentication and authorization for API endpoints.
- Rate Limiting and Throttling: Rate limiting tools, like Redis[218] or custom middleware, can be used to enforce rate limits on API requests and prevent abuse or overuse of resources.
- Content Filtering and DLP Solutions[219]: Content filtering and Data Loss Prevention (DLP) solutions help enforce policies related to data security and content filtering in distributed systems.

[210] https://web3js.readthedocs.io/en/v1.10.0/#

[211] https://docs.ethers.org/v5/

[212] https://www.chainalysis.com/

[213] https://aws.amazon.com/iam/?nc=sn&loc=0

[214] https://learn.microsoft.com/en-us/azure/active-directory/

[215] https://www.nginx.com/

[216] https://konghq.com/

[217] https://docs.apigee.com/

[218] https://redis.io/

[219] Alneyadi, Sultan, Elankayer Sithirasenan, and Vallipuram Muthukkumarasamy. "A survey on data leakage prevention systems." Journal of Network and Computer Applications 62 (2016): 137-152.

- Digital Rights Management (DRM): DRM technologies, such as Widevine[220] or PlayReady[221], are used to enforce copyright and licensing policies for digital content distribution.
- Token-Based Authentication and Authorization: Tools like OAuth 2.0[222] or JWT[223] (JSON Web Tokens) are used to enforce secure authentication and authorization mechanisms in distributed systems.
- License Management Solutions: License management tools, such as FlexNet[224] or Keygen[225], help enforce software licensing policies and manage license compliance for distributed applications.
- Blockchain and Smart Contracts: Blockchain technology, along with smart contracts, can be used to enforce trust and usage policies in decentralized and distributed applications.
- Compliance and Governance Platforms: Compliance and governance platforms, like Chef Compliance or AWS Config, help enforce regulatory compliance and governance policies in distributed systems.

The specific tools and technologies used for usage enforcement in distributed systems will depend on the nature of the system, the resources involved, and the policies and regulations that need to be enforced. Often, a combination of these tools and technologies is used to provide comprehensive usage enforcement and security in distributed environments.

4 Data Peer-to-Peer Transfer

4.1.1 Data Transfer in PISTIS

Data Transfer between different stakeholders in PISTIS is the logical termination point of a monetary or otherwise exchange agreement flow, where following the establishment of an electronic contract (and the possible value exchange that is part of this agreement), the dataset that is part of the agreement has to reach the buyer.

In order to meet the promised privacy and data security guarantees set in the PISTIS concept, these data transfers between different stakeholders are following a peer-to-peer pattern, where no intermediary exist, so that in the whole process beneficiaries retain control of their data and the transfer route is direct between their premises (e.g. their PISTIS Data Factory Environments).

The latter is quite important, as the envisaged data transfer has to take place between two PISTIS Data Factory deployments, and thus it is not possible to get access to the data transfer methods outside of a PISTIS deployment, in order to enable some of the PISTIS global services that have to do with lineage tracking, transaction monitoring, etc.

[220] https://www.widevine.com/solutions/widevine-drm
[221] https://www.microsoft.com/playready/overview/
[222] https://oauth.net/2/
[223] https://auth0.com/docs/secure/tokens/json-web-tokens
[224] https://www.flexera.com/products/flexnet-manager
[225] https://keygen.sh/

4.1.2 Methods

Peer-to-peer data exchange methods are not something new and one might argue that these were the mainstream data transfer methods prior to the cloud era. With the wide penetration of cloud technologies, and the rise of centralised data repositories, many other data sharing methods have emerged, which has many benefits, but also carried also pain points, with the majority of the latter having to do with privacy and security of data, especially in the case third parties were involved for facilitating the data exchange (in the case of data brokers, or third-party operated data repositories, etc.).

As a result, in the last years, peer-to-peer data exchange is once again witnessing a bloom, as for certain operations, and especially those which have to do with business data exchange, call for more trusted and secure communication channels for the end-to-end transfer of the payload, without placing in the middle intermediaries that handle these transactions, and without using public or third party infrastructures where the data can reside until it is sent to, or collected by the recipients.

Such peer-to-peer exchanges can happen using different methods that allow both synchronous and asynchronous communication between the different parties and that could facilitate the exchange of bulk or of real-time data, using different protocols and tools.

4.1.2.1 Data Transfer through RESTfull APIs

One of the mostly used method for exchanging data is that of using REST APIs[226] which are specifically designed to exchange data between transaction parties, using either POST or GET methods[227], depending on the use case. In principle, using such a method allows the data consumer party to "request" for some data (GET methods) or the data providing party to "push" the data to an address provided by the recipient (POST method), using pre-defined and documented calls, which are known to both parties, alongside with the expected response messages and formats, in order to allow them to build applications that can integrate these methods. In essence, these APIs communicate with other backend entities within the system (such as a database server), to collect and wrap the data that needs to be transferred and in the whole process make use of various accompanying options, such as the usage of encrypted communication channels, the use of authentication tokens, etc.

4.1.2.2 Event & Data Streaming platforms

When it comes to the exchange of streaming data, REST APIs are not always the best option, and therefore other methods have been developed such as event streaming[228] platforms and pub/sub[229] servers. These allow data producers to constantly stream the data output and interested parties (e.g. data consumers) to "subscribe" to these streams and pull the data they need as it is coming into the feed. In the same sense, also message brokers can be used, however when it comes to large data volumes such brokers are not ideal as their main design

[226] https://www.ibm.com/topics/rest-apis

[227] https://www.w3schools.com/tags/ref_httpmethods.asp

[228] https://en.wikipedia.org/wiki/Stream_processing

[229] https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern

purpose is to facilitate messaging between different services, rather than transfer large amounts of data.

4.1.2.3 Data Exchange between Data Spaces participants

Another method of data exchange is that of Data Space Connectors. Following the rise of Data Spaces[230], other approaches came into surface to tackle data sovereignty and security of the different transactions within data space, a new way of data exchange has been proposed, that of the Data Space Connector[231], with the initial implementation (performed by Fraunhofer ISST) of this concept being the IDS Data Space connector[232] that included all elements found in the IDS RAM (Reference Architecture Model)[233]. In principle, the idea behind a dataspace connector is to act as a secure gateway between two different transaction parties that want to exchange data with each other. The connector provides services for the data plane (e.g. the actual transfer of the data) and the control plane (e.g. all the services that should precede and succeed a data transfer transaction in the frame of the IDS ecosystem, such as identify management, access policies, contract negotiations, logging, etc) and those two planes are coupled.

4.1.2.4 File Transfer Protocol

Finally, there exist also other methods that can be used to facilitate data transfer between transaction parties, such as the setup of FTP connections between the transaction parties to allow data consumers to directly download files from a data provider using their FTP servers. Nevertheless, although there are certain use cases that deem FTP file transfer necessary (for example in the case of extremely large files), such methods are not considered a modern way of data exchange in the frame of a holistic data sharing ecosystem, as they cannot be easily controlled nor coupled with other services that can manage transactions.

4.1.3 Technologies

RESTfull APIs

Data transfer via APIs can be based on REST-services that can be developed from scratch and operate over HTTP or HTTPS and would allow transaction parties to exchange data by performing such calls. Such solutions are usually written in the same language as the one used for the main application, as they are in most cases deeply integrated into the overall environment they need to serve.

Apache Kafka

[230] https://joinup.ec.europa.eu/collection/semic-support-centre/data-spaces#:~:text=A%20secure%20and%20privacy%2Dpreserving,and%20trustworthy%20data%20governance%20mechanisms.

[231] https://www.isst.fraunhofer.de/en/business-units/data-business/technologies/Dataspace-Connector.html#:~:text=The%20Dataspace%20Connector%20(DSC)%20is,what%20happens%20to%20the%20data.

[232] https://internationaldataspaces.org/offers/ids-components/

[233] https://docs.internationaldataspaces.org/knowledge-base/ids-ram-4.0

Regrading real-time data streaming and exchange, perhaps the most known solution is **Apache Kafka**[234] which is dominating the market. Kafka is a distributed event store and streaming processing platform, offered as an open source solution that is written in Java and Scala and is under the umbrella of the Apache Foundations. The overall infrastructure provides a highly scalable and performant system that is able to cope with high-throughput and low-latency streaming data, and is used by tech giants as the main medium for allowing event stream communications between different systems of any size.

RabbitMQ

RabbitMQ[235] is an open-source message broker service that can in many cases deliver the same functionality as Kafka (see above) and is implementing the AMQP protocol. Essentially, as a message broker RabbitMQ can be used by different parties to send and retrieve messages and build messaging queues between different parties.

Data Space Connectors

When it comes to Data Space Connectors, there are many approaches available, with most of them being evolutions and extensions to the **IDS connector** (which is still in use, as for example in the Mobility Data Space. This connector, and most of its successors are written in Java and Scala, but they can operate and be integrated in any environment. Possibly, the most promising evolution is that of the **Eclipse Data Space Connector**[236] (and is the successor of the original **Data Space Connector**[237]), which is back-up by the Eclipse foundation and is part of the Eclipse Data Space Components pack, which provides services to set-up a minimum viable data space. The EDC comes with a separation of the control and data planes to allow for more scalability and also to comply with the GAIA-X architecture requirements. Of course, as the IDS RAM is an open architecture and the original implementation of the IDS Connector (as of all others) is open source, there are also many other connectors available, such as the **True Connector**[238], etc. The Data Connector report published by IDSA in May 2023 provides a list of 23 available Data Space Connector implementations with 10 of them being open source at the moment.

[234] https://kafka.apache.org/

[235] https://www.rabbitmq.com/

[236] https://projects.eclipse.org/projects/technology.edc

[237] https://international-data-spaces-association.github.io/DataspaceConnector/

[238] https://github.com/Engineering-Research-and-Development/true-connector

5 DATA SECURITY AND TRUST

5.1 GDPR CONFORMANCE CHECKING

5.1.1 GDPR Conformance Checking in PISTIS

European General Data Protection Regulation (GDPR)[239] compliance is a functionality of paramount importance for data trading platforms such as PISTIS. For instance, in order for a data transaction to be GDPR compliant, the data subject and controller, the data processor, the type of data (e.g., personal data), the purpose of this transaction, the form of data (e.g., if are anonymized or not), the cryptographic algorithm used are some aspects that need to be checked by a legal entity (e.g., DPO) prior to the actual transaction.

5.1.2 Methods

The introduction of the GDPR enables users to control how their data is accessed and processed, requiring consent from users before any data manipulation is carried out on their data. Until now, several research works, and commercial tools have been proposed to provide GDPR compliance services. Among them blockchain technology could play an important role in helping organisations comply with GDPR rules. Even though blockchains are used to provide GDPR services, the enactment of GDPR might bring a tension between general data protection principles and the core features of blockchain technology[240]. Below we provide some methods described in the literature for checking the GDPR compliance.

Blockchain and Smart Contracts

Blockchain-based support and smart contracts is a common method for tracking data-related operations and incorporate user privacy and security in the development of applications. Several works in the literature are utilised these methods [241]. Also, research works have proposed the use of a blockchain-based approach to support data accountability and provenance tracking [242]. The nature of blockchain is suitable for tracking assets exchanged and/or changing ownership of an asset between different entities. The problem they try to address can be described as the loss of control over how personal data is handled when a person (subject) gives personal information to a company (controller). In addition, researchers proposed frameworks for supporting GDPR-compliant processing of user data

[239] REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL
of 27 April 2016, https://eur-lex.europa.eu/eli/reg/2016/679/oj

[240] M Berberich, M Steiner., "Blockchain Technology and the GDPR - How to Reconcile Privacy and Distributed Ledgers?", HeinOnline Eur. Data Prot. L., 2016.

[241] K. Gai, Y. Wu, L. Zhu, Z. Zhang, and M. Qiu, ''Differential privacy-based blockchain for industrial Internet of Things,'' IEEE Trans. Ind. Informat., vol. 16, no. 6, pp. 4156–4165, Jun. 2020.

[242] A Blockchain-based Approach for Data Accountability and Provenance Tracking https://arxiv.org/pdf/1706.04507.pdf

generated using IoT devices [243]. Lee et. al in their research paper[244] proposed an efficient method to obtain user consent during collection of personal data from IoT devices. Moreover, a blockchain-based method that facilitates secure data management in the healthcare domain is also proposed where smart contracts and private/public keys are used to improve user privacy and ide an access control mechanism [245]. A recent work proposes a blockchain-based architecture along with business processes to demonstrate how the integration of GDPR and blockchain can appear as design patterns enhancing user privacy [246]. In general, most implemented approaches using blockchains also combine encryption, anonymization, private contract, mixing, and differential privacy techniques.

Online form

Another method is the online form, asking the data controller a series of simple questions to establish the GDPR compliance of the data asset. The questions cover issues such as: definition of the purpose of the processing activity, the existence of personal or sensitive data, compliance with the EU Data Protection Principles, the rights of data subjects, appropriate technical and organisation security measures. The expected answers are concise (Yes/No, multiple selection) and no open questions are included. This allows for easy quantification of GDPR compliance.

5.1.3 Technologies

This section provides tools for GDPR Checking.

Data Protection Check in the Data Valuation Platform

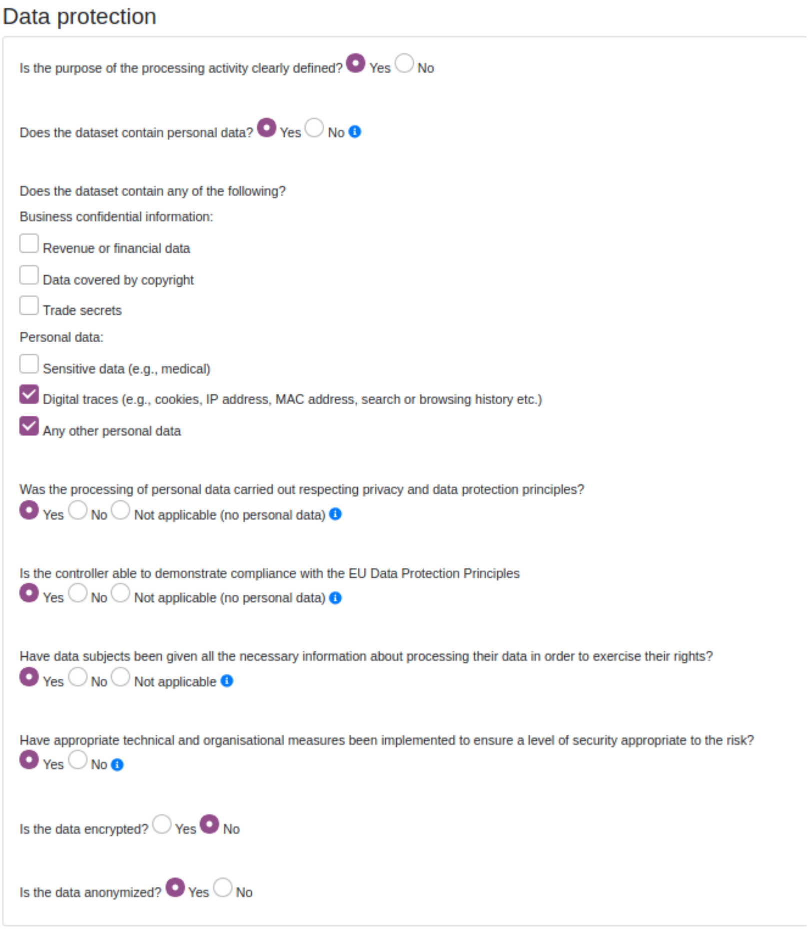
The Data protection form, part of the EUT Data Valuation Platform, functions according to the Online form method described in the subsection 5.1.2.

[243] K. Rantos, G. Drosatos, K. Demertzis, C. Ilioudis, and A. Papanikolaou, ''Blockchain-based consents management for personal data processing in the IoT ecosystem,'' in Proc. 15th Int. Joint Conf. e-Bus. Telecommun., 2018, pp. 572–577.

[244] G. Y. Lee, K. J. Cha, and H. J. Kim, ''Designing the GDPR compliant consent procedure for personal information collection in the IoT environment,'' in Proc. IEEE Int. Congr. Internet Things (ICIOT), Jul. 2019, pp. 79–81.

[245] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, ''A decentralized privacy-preserving healthcare blockchain for IoT,'' Sensors, vol. 19, no. 2, p. 326, Jan. 2019.

[246] M. Barati and O. Rana, ''Enhancing user privacy in IoT: Integration of GDPR and blockchain,'' in Blockchain Trustworthy Systems (Communications in Computer and Information Science), vol. 1156, Z. Zheng, H. N. Dai, M. Tang, and X. Chen, eds. Singapore: Springer, 2020, pp. 322–335.



Data protection

Is the purpose of the processing activity clearly defined? ● Yes ○ No

Does the dataset contain personal data? ● Yes ○ No

Does the dataset contain any of the following?

Business confidential information:

□ Revenue or financial data

□ Data covered by copyright

□ Trade secrets

Personal data:

□ Sensitive data (e.g., medical)

☑ Digital traces (e.g., cookies, IP address, MAC address, search or browsing history etc.)

☑ Any other personal data

Was the processing of personal data carried out respecting privacy and data protection principles?
● Yes ○ No ○ Not applicable (no personal data)

Is the controller able to demonstrate compliance with the EU Data Protection Principles
● Yes ○ No ○ Not applicable (no personal data)

Have data subjects been given all the necessary information about processing their data in order to exercise their rights?
● Yes ○ No ○ Not applicable

Have appropriate technical and organisational measures been implemented to ensure a level of security appropriate to the risk?
● Yes ○ No

Is the data encrypted? ○ Yes ● No

Is the data anonymized? ● Yes ○ No

Figure 17: Data protection form in the EUT Data Valuation Platform

GDPR Check

GDPR Check[247] supports organisation, mainly SMEs, to protect their data based on reputation of these data. Their approach is based on a questionnaire that includes various guidelines in order to be easier the identification of weak points of data processing. The whole procedure takes approximately one day.

Automated GDPR Compliance Checking

Automated GDPR Compliance Checking [248] is a rule-based machine learning tool to automate compliance checking of privacy policies. The tool has two modules one that relies on NLP to extract data practices from privacy policies and another one that encodes GDPR rules to check the presence of mandatory information.

GDPR Compliance Tool

[247] GDPR Check, Website: https://www.dekra.com/en/gdpr-check/

[248] Automated GDPR Compliance Checking, Website: https://github.com/smartlawhub/Automated-GDPR-Compliance-Checking

GDPR Compliance Tool[249] is scalable data protection by design tool for automated GDPR compliance verification based on semantically modelled informed consent. The tool supports also data interoperability through the use of semantic technology.

GDPR-Compliant Data Subjects' Personal Data Access Control System

GDPR-Compliant Data Subjects' Personal Data Access Control System [250] is a lightweight blockchain-based GDPR-compliant personal data management system, that provides public access immutable evidence showing the agreements between a data subject and a service provider about data subjects' personal data.

5.2 Anonymisation

5.2.1 Data Anonymisation in PISTIS

Within the context of PISTIS anonymisation will be used to preserve private or confidential information. It is done to protect the private activity of an individual or a corporation while preserving the credibility of the data collected and exchanged. Anonymisation is the technique that will be used to support adherence to strict data privacy regulations that require the security of personally identifiable information (PII), such as health reports, contact information, and financial details.

5.2.2 Methods of Data Anonymization

5.2.2.1 Data masking

Data masking refers to the disclosure of data with modified values. Data anonymization is done by creating a mirror image of a database and implementing alteration strategies, such as character shuffling, encryption, term, or character substitution. For example, a value character may be replaced by a symbol such as "*" or "x." It makes identification or reverse engineering difficult.

5.2.2.2 Pseudonymization

Pseudonymization is a data de-identification tool that substitutes private identifiers with false identifiers or pseudonyms, such as swapping the "John Smith" identifier with the "Mark Spencer" identifier. It maintains statistical precision and data confidentiality, allowing changed data to be used for creation, training, testing, and analysis, while at the same time maintaining data privacy.

5.2.2.3 Generalization

Generalization involves excluding some data purposely to make it less identifiable. Data may be modified into a series of ranges or a large region with reasonable boundaries. For example,

[249] GDPR Compliance Tool, Website: https://github.com/tekrajchhetri/GDPR_compliance_tool

[250] GDPR-Compliant Data Subjects' Personal Data Access Control System, Website: https://github.com/toful/BC_GDPR-Compliant_PDManagement_System

the house number at an address may be deleted, but make sure the name of the lane does not get deleted. The aim is to remove some of the identifiers while maintaining the accuracy of the data.

5.2.2.4 Data swapping

Data swapping – often known as permutation and shuffling – rearranges dataset attribute values so that they do not fit the original information. Switching attributes (columns) that include recognizable values, such as date of birth, can make a huge impact on anonymization.

5.2.2.5 Data perturbation

Data perturbation modifies the initial dataset marginally by applying round-numbering methods and adding random noise. The set of values must be proportional to the disturbance. A small base can contribute to poor anonymization, while a broad base can reduce a dataset's utility. For example, a base of 5 should be used for rounding values like age or house number.

5.2.2.6 Synthetic data

Synthetic data is algorithmically generated information with no relation to any actual case. The data is used to construct artificial datasets instead of modifying or utilizing the original dataset and compromising privacy and protection.

The synthetic data method includes the construction of mathematical models based on patterns contained in the original dataset. Standard deviations, linear regression, medians, or other statistical methods can be used to produce synthetic results.

5.2.2.7 K-Anonymity

***K*-anonymity**, is a privacy model commonly applied to protect the data subjects' privacy in data sharing scenarios, and the guarantees that *k*-anonymity can provide when used to anonymise data. In many privacy-preserving systems, the end goal is anonymity for the data subjects. Anonymity when taken at face value just means to be nameless, but a closer look makes it clear very quickly that only removing names from a dataset is not sufficient to achieve anonymisation. Anonymised data can be re-identified by linking data with another dataset. The data may include pieces of information that are not themselves unique identifiers, but can become identifying when combined with other datasets, these are known as quasi-identifiers.

K-anonymity is a property of a dataset that indicates the re-identifiability of its records. A dataset is *k*-anonymous if quasi-identifiers for each person in the dataset are identical to at least *k* – 1 other people also in the dataset.[251]

5.2.2.8 Differential Privacy (DP)

Differential privacy is a rigorous mathematical definition of privacy. In the simplest setting, consider an algorithm that analyses a dataset and computes statistics about it (such as the data's mean, variance, median, mode, etc.). Such an algorithm is said to be differentially

[251] https://epic.org/wp-content/uploads/privacy/reidentification/Sweeney_Article.pdf

private if by looking at the output, one cannot tell whether any individual's data was included in the original dataset or not. In other words, the guarantee of a differentially private algorithm is that its behaviour hardly changes when a single individual join or leaves the dataset -- anything the algorithm might output on a database containing some individual's information is almost as likely to have come from a database without that individual's information. Most notably, this guarantee holds for *any* individual and *any* dataset. Therefore, regardless of how eccentric any single individual's details are, and regardless of the details of anyone else in the database, the guarantee of differential privacy still holds. This gives a formal guarantee that individual-level information about participants in the database is not leaked.

Differential privacy mathematically guarantees that anyone seeing the result of a differentially private analysis will essentially make the same inference about any individual’s private information, whether or not that individual’s private information is included in the input to the analysis

DP provides a mathematically provable guarantee of privacy protection against a wide range of *privacy attacks (*include *differencing attack*, *linkage attacks,* and *reconstruction attacks)*[252]

5.2.2.9 *Gaussian Mixture Models*

Gaussian Mixture Models (GMMs) represent a dataset as a mixture of Gaussian distributions. In other words, GMMs assume that any observed data set is a combination of several Gaussian distributions. Consisting of two main components (mixture components and mixing weights) are the individual Gaussian distributions that build the model, while the mixing weights determine the contribution of each element to the overall distribution.

GMMs aim is to estimate the parameters that describe the mixture components and mixing weights from the given data. This estimation is typically performed using an algorithm called Expectation-Maximization (EM). The EM algorithm iteratively updates the parameters to maximize the likelihood of the observed data given the model.

A trained GMM can be used for various tasks, from clustering to data generation. Because of their generative nature, GMMs can generate synthetic data by sampling from the learned distribution. By randomly selecting a component according to the mixing weights and sampling from the corresponding Gaussian distribution, synthetic data points are drawn from the learned distribution, ensuring that the new data mimics the characteristics of the original data.[253]

5.2.3 Technologies

ARX

ARX is a scalable Data Anonymization Tool that supports multiple privacy models. It is open-source and can be used for anonymizing sensitive personal data. It supports a wide variety of

[252] https://towardsdatascience.com/understanding-differential-privacy-85ce191e198a

[253] https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

privacy and risk models, methods for transforming data and methods for analysing the usefulness of output data. It supports various anonymization techniques, methods for analysing data quality and re-identification risks and it supports well-known privacy models, such as k-anonymity, l-diversity, t-closeness and differential privacy. It supports the ability to transform structured (i.e. tabular) personal data using selected methods from the broad areas of data anonymization and statistical disclosure control. It supports transforming datasets in ways that make sure that they adhere to user-specified privacy models and risk thresholds that mitigate attacks that may lead to privacy breaches. ARX can be used to remove direct identifiers (e.g. names) from datasets and to enforce further constraints on indirect identifiers. Indirect identifiers (or quasi-identifiers, or keys) are attributes that do not directly identify an individual but may together with other indirect identifiers form an identifier that can be used for linkage attacks. It is typically assumed that information about indirect identifiers is available to the attacker (in some form of background knowledge) and that they cannot simply be removed from the dataset (e.g., because they are required later for analyses).[254]

Persona Generator[255]

This provides a capability to provide generalised insights from aggregated data without ever revealing personally identifiable information on any of the individuals to whom that aggregated data may relate to. The privacy risk is considerably reduced as long as visibility of the actual data is not required and demographic insights of aggregated data based on specific criteria e.g people living in Berlin OR people over 70 who are still in employment, has the required utility.

CTGAN

CTGAN is a data synthesizer that can generate synthetic tabular data with high fidelity. CTGAN uses more sophisticated Variational Gaussian Mixture Model to detect modes of continuous columns.

CTGAN expects the input data to be a table given as either a numpy.ndarray or a pandas. DataFrame object with two types of columns:

- Continuous Columns - columns that contain numerical values and which can take any value.
- Discrete columns - columns that only contain a finite number of possible values, whether these are string values or not.

This is an example of a table with 4 columns:

- A continuous column with float values
- A continuous column with integer values
- A discrete column with string values
- A discrete column with integer values

[254] https://arx.deidentifier.org/

[255] A proprietary tool developed by ASSENTIAN

Table 2: Example of input expected by CTGAN

	A	B	C	D
0	0.1	100	'a'	1
1	-1.3	28	'b'	2
2	0.3	14	'a'	2
3	1.4	87	'a'	3
4	-0.1	69	'b'	2

NOTE: CTGAN does not distinguish between float and integer columns, which means that it will sample float values in all cases. If integer values are required, the outputted float values must be rounded to integers in a later step, outside of CTGAN.

This has specific value when dealing with location data and more specifically trajectory data. Location data has proven to have significant monetary value in a host of application areas such as retail and transport for example. The privacy risks are significant however if for examples geo-locations are simply stripped out it no longer has the utility and monetary value.

CTGAN allows for location perturbation in this instance. The Synthetic Data Vaults library supports the generation of new Synthetic Data that has the same format and statistical properties as the original location dataset. The library makes use of probabilistic graphical modelling and deep learning-based techniques.[256]

DiffPrivLib

Diffprivlib is a general-purpose library for developing applications in, differential privacy. It is comprised of four major components:

- Mechanisms: These are the building blocks of differential privacy and are used in all models that implement differential privacy. Mechanisms have little or no default settings and are intended for use by experts implementing their own models. They can, however, be used outside models for separate investigations, etc.
- Models: This module includes machine learning models with differential privacy. Diffprivlib currently has models for clustering, classification, regression, dimensionality reduction and pre-processing.
- Tools: Diffprivlib comes with a number of generic tools for differentially private data analysis. This includes differentially private histograms, following the same format as Numpy's histogram function.
- Accountant: The BudgetAccountant class can be used to track privacy budget and calculate total privacy loss using advanced composition techniques.[257]

[256] https://github.com/sdv-dev/CTGAN

[257] https://github.com/IBM/differential-privacy-library

5.3 Access Policies Management

5.3.1 Identity Management and Access Control in PISTIS

Identity management and access control within the PISTIS platform will ensure security and controlled interactions within its ecosystem. The platform will integrate various methods and technologies to authenticate, authorize, and manage identities and resource access effectively.

One of the available features in PISTIS will be the alignment with eIDAS verifiable credentials, as they evolve through eIDAS 2.0[258] regulation, and the support of eIDAS certificates, aligning with European Union standards for electronic identification and transaction which embraces the concept of Self-Sovereign Identity (SSI) used in Single-Sign On scenarios, empowering individuals with permanent, self-owned digital identities. This approach offers control and autonomy to users over their identities and data sharing, enhancing privacy and security.

Various access control policies will be used, such as Attribute-Based Access Control (ABAC) and Role-Based Access Control (RBAC). Each type will be used to different scenarios, depending on the requirements of each module and functionality.

Also, authentication and authorization protocols like OpenIDC will be used, contributing to securing a user's identity validation and single sign-on capabilities.

5.3.2 Methods

5.3.2.1 eIDAS Certificates

eIDAS[259] , stands for "electronic Identification, Authentication and trust Services," is a regulation established by the European Union to create a standard for electronic identification and transaction across EU member states. sets guidelines and standards for electronic transactions, including aspects like electronic signatures, seals, time stamps, delivery services, and website authentication.

A critical aspect of eIDAS is its directive on electronic identification schemes. The objective of eIDAS[260] is to boost confidence in electronic transactions, establish a legal foundation for electronic interactions, and encourage cross-border digital commerce. *It facilitates secure and seamless electronic interactions among EU citizens, businesses, and public administrations.*

5.3.2.2 Identity and Access Management Frameworks

IAM ensures the consistent use and management of identities across all applications while preserving security. It authenticates users, devices, or services, and regulates access to resources and data. Instead of having unique identity storage or authentication for each application, IAM connects with a trusted identity provider to ease this process. IAM simplifies

[258] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0281&from=EN

[259] https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation

[260] Dr. Ignacio Alamillo Domingo April – 2020, EUROPEAN COMMISION, How eIDAS can legally support digital identity and trustworthy DLT-based transactions in the Digital Single Market.

the management of vast distributed systems and can be used both internally and externally. It identifies cloud objects, controls access to resources, and upholds pre-set policies[261,262].

5.3.2.3 Self-Sovereign Identity (SSI)

Self-Sovereign Identity in its simplest form as a digital representation of the individual's characteristics, description, and identifiers. SSI can be defined as a permanent identity owned and controlled by the per-son or entity to whom it belongs to without the need to rely on any external administrative authority and without the possibility that this identity can be taken away[263]. That requires not just the inter-operability of a user's identity across multiple locations, with the user's consent, but also, true user control of that digital identity, and full user autonomy.

5.3.2.4 Identity Management via Blockchain

Blockchain technology offers a secure, decentralized solution for identity management, reducing reliance on central authorities. It enables individuals to create a blockchain identity, offering more control over personal data accessibility[264]. This tech can create a digital ID as a unique watermark for online transactions, reducing fraud by enabling real-time identity checks. With blockchain, consumers can authenticate transactions through an app, storing their encrypted identity, and managing their data sharing on their terms.

5.3.2.5 Access Control Types

5.3.2.5.1 ABAC access control policies

Attribute-Based Access Control (ABAC) is a secure method for regulating access within computer systems, particularly useful in large or complex environments. It uses granular attributes tied to users, resources, or the environment for refined access control. User attributes could be job title or location, while resource attributes might be data type or owner, and environment attributes could relate to time or network[265]. ABAC enables intricate access policies suited for diverse scenarios, supporting the principle of least privilege. It can limit access to sensitive information accurately, enhance security, and aid compliance demonstration. Furthermore, its automation potential and integration ease with systems using the same attributes make it efficient for access rights management[266].

5.3.2.5.2 Role-based access control (RBAC)

Role-Based Access Control[267] (RBAC) manages access to computer resources based on users' roles in an organization rather than their individual identities. Users are assigned roles, like

[261] Annu Myllyniemi, Helsinki University of Technology, Identity Management Systems A Comparison of Current Solutions

[262] Deepak H. Sharm, Dr. C. A. Dhote, 7th International Conference on Communication, Computing and Virtualization 2016, Identity and Access Management as Security-as-a-Service from Clouds.

[263] Dirk van Bokkem, Rico Hageman, Gijs Koning, Luat Nguyen and Naqib Zarin, Computer Science and Engineering, Delft University of Technology, Self-Sovereign Identity Solutions: The Necessity of Blockchain Technology.

[264] Hamza ES-SAMAALI, Aissam OUTCHAKOUCHT and Jean Philippe LEROY, International Journal of Computer Networks and Communications Security, A Blockchain-based Access Control for Big Data.

[265] Lidong Chen, David McGrew, Chris Mitchell, Third International Conference, SSR 2016 Gaithersburg, MD, USA, December 5–6, 2016 Proceedings, Security Standardization Research.

[266] Li, F. (2015). Context-Aware Attribute-Based Techniques for Data Security and Access Control in Mobile Cloud Environment. (Unpublished Doctoral thesis, City University London)

[267] Ravi S. Sandhu, Volume 46, 2001, Pages 237 -286, Advances in Computers.

'accountant' or 'engineer', each with specific permissions[268]. New users receive access based on their role(s), preventing access to irrelevant sensitive data or operations. RBAC's efficiency in managing user permissions for large user groups offers simplicity, reduces errors, and improves compliance.

5.3.2.5.3 Mandatory Access Control (MAC)

Mandatory Access Control[269] (MAC) limits access to computer resources based on security clearances and data classification. It labels all system resources according to data sensitivity and assigns users security clearances that determine their access level. User clearance and resource classification are compared when a user tries to access a resource. MAC enhances security by strictly controlling access based on pre-set classifications and clearances, and preventing user-alterations to permissions, safeguarding access control integrity.

5.3.2.5.4 Discretionary Access Control (DAC)

Discretionary Access Control (DAC) is a policy where data owners determine who can access specific resources and how. DAC[270] permits owners to grant or deny different access rights, like read or write permissions, using Access Control Lists (ACLs). The system's advantage lies in its flexibility, allowing owners to adjust controls based on their needs or changes in requirements. However, this adaptability must be carefully managed to maintain system security.

5.3.2.5.5 Rule – Based Access Control (RBAC)

Rule-Based Access Control[271] (RBAC) is a system where access rights are managed based on pre-set rules. Each rule has a condition and an action, which triggers when the condition is met. RBAC's key advantage is its adaptability for complex conditions and intricate business scenarios. It offers customizable access control, easy-to-understand rules aiding in audits, compliance, and troubleshooting, and adaptability to changing conditions.

5.3.2.5.6 Risk-Based Access Control

Risk-Based Access Control[272] (RBAC) regulates access to computer resources considering the potential risk of a user's request, not just their identity and permissions. It uses risk scoring, evaluating factors like location, device type, data sensitivity, and access time. Benefits of RBAC include dynamic security, as it identifies unusual behaviour indicating threats, and improved user experience, as it requires additional authentication only in high-risk situations.

[268] Riccardo Focardi Roberto Gorrieri (Eds.), Foundations of Security Analysis and Design (Lecture Notes in Computer Science, 2171).

[269] Sylvia Osborn, Department of Computer Science, The University of Western Ontario, Mandatory Access Control and Role-Based Access Control Revisited.

[270] Hiroshi Yoshiura, Kouichi Sakurai, Kai Rannenberg, Yuko Murayama, First International Workshop on Security, IWSEC 2006, Japan, Advances in Information and Computer Security.

[271] Li, F. (2015). Context-Aware Attribute-Based Techniques for Data Security and Access Control in Mobile Cloud Environment. (Unpublished Doctoral thesis, City University London).

[272] Hany F. Atlam , Muhammad Ajmal Azad , Madini O. Alassafi, Abdulrahman A. Alshdadi and Ahmed Alenezi, Risk-Based Access Control Model: A Systematic Literature Review.

5.3.3 Technologies

5.3.3.1 Auth/auth/sso Protocols

- **Oauth 2.0**[273] is a standard for authentication on the Internet, used by major identity providers like Google, Twitter, GitHub. OAuth 2.0 is an authorization framework that enables a third-party application to obtain limited access to an HTTP service. It relies on HTTP's and generates tokens that allow it to protect resources. As the SAML standard that is explained below, provides Single Sign-On (SSO).
- **OpenID Connect**[274] protocol provides a delegated authentication, and it was released due to the adoption of OAuth 2.0 as a pseudo authentication mechanism. OIDC is built on top of OAuth 2.0 and provides a pure authentication mechanism, uses the authorization code and implicit grant types and even a hybrid of both. Returns Json Web Tokens instead of a simple token. A Json web token is composed of the header (including signing algorithm), the payload (user's information) and the signature (to verify the validity of the token).
- **Security Assertion Markup Language**[275] (SAML) is one of the standards for providing Identity Federation, is an open-source protocol released by OASIS (Organization for the Advancement of Structured Information Standards). SAML provides authentications and authorization mechanisms between participants using HTTP, XML, SOAP.

  The SAML authentication and authorization flow has three main roles: user, Identity Provider (IdP) and Service Provider (SP). IdP [276] provides the authentication mechanisms and stores the user's information while the SP could be a web application that wants to check the user's identity. However, before starting an authentication request, the SP and the IdP must ensure communication between them. Both generate a private and a public key to sign or encrypt the SAML messages, usually expose, through an endpoint a metadata file containing the public keys needed. IdP and SP must securely store the private keys on their servers.

5.3.3.2 Self-Sovereign Identity (SSI) implementations

Self-Sovereign Identity (SSI)[277] represents a decentralized and user-centric way for managing digital identities. Its primary objective is to empower individuals with greater authority over their personal information, thereby diminishing reliance on centralized identity authorities. Within the SSI framework, individuals possess the autonomy to independently initiate, own, and oversee their digital identities. SSI holds substantial promise in mitigating several issues endemic to conventional identity management systems, including identity theft, data breaches, and the inconvenience associated with managing numerous usernames and

[273] https://oauth.net/2/

[274] https://openid.net/developers/how-connect-works/

[275] https://www.oasis-open.org/standard/saml/

[276] Juraj Somorovsky, Andreas Mayer, Jorg Schwenk, Marco Kampmann and Meiko Jensen, Horst Gortz Institute for IT-Security, Ruhr-University Bochum, Germany, Adolf Wurth GmbH & Co. KG, K unzelsau-Gaisbach, Germany, On Breaking SAML: Be Whoever you want to Be.

[277] Michael Friedewald, Melek Önen, Eva Lievens, Stephan Krenn, Samuel Fricker (Eds.), Privacy and Identity Management Data for Better Living: AI and Privacy, 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2.2 International Summer School Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers

passwords. Furthermore, SSI finds versatile applications in various sectors, including finance, healthcare, government, and more, where the assurance of secure and user-centric identity verification remains paramount.

Current implementations of SSI[278] such as uPort, IDchainZ, EverID, Sorvin, LifeID, SelfKey and Sora focus on various aspects of digital identity and identity management, often utilizing blockchain technology and principles of self-sovereign identity. These projects aim to provide individuals with greater control, security, and privacy over their digital identities and personal information. While each of these projects may have its unique features and approaches, their overarching goal is to improve how individuals manage and secure their online identities in a more user-friendly and secure manner.

5.3.3.2.1 Use of SSI in Gaia-X Federation Services

Gaia-X uses Self-Sovereign Identity (SSI) technology across various facets of its ecosystem[279]. It enables decentralized identity management, allowing participants to create and manage their digital identities independently, reducing reliance on centralized identity providers. SSI facilitates decentralized authentication and authorization through technologies like Self-Issued OpenID Provider (SIOP)[280], enabling users to authenticate themselves without central authority and use verifiable credentials for fine-grained access control. Gaia-X incorporates secure wallet solutions for credential management, granting users control over disclosed credentials, enhancing privacy. Gaia-X's portal authentication integrates SSI, ensuring secure access to services.

5.3.3.3 Identity and access management (IAM) frameworks.

Keycloak[281,282] is an open-source Identity and Access Management (IAM) solution developed by Red Hat. It offers a unified platform for authentication, authorization, and user management across various applications and services. Key features include Single Sign-On (SSO) for streamlined user experiences, support for identity federation through protocols like SAML 2.0 and OpenID Connect, flexible authentication methods including multi-factor authentication (MFA), fine-grained authorization controls based on roles and permissions, user self-service capabilities for password resets and profile management, and the issuance of security tokens for securing API calls.

OneLogin[283] is an Identity and Access Management (IAM) solution designed to streamline and secure user authentication and authorization processes across applications and devices. OneLogin enables organizations to implement Single Sign-On (SSO), Multi-Factor Authentication (MFA), and comprehensive identity management. Its identity integration capabilities cover various systems, while its adaptive authentication ensures enhanced

[278] Dirk van Bokkem, Rico Hageman, Gijs Koning, Luat Nguyen and Naqib Zarin, Computer Science and Engineering, Delft University of Technology, Self-Sovereign Identity Solutions: The Necessity of Blockchain Technology.

[279] Gaia-X secure and trustworthy ecosystems with Self Sovereign Identity, Developing a Decentralised, User-Centric, and Secure Cloud Ecosystem. https://gaia-x.eu/wp-content/uploads/2022/09/SSI-White-Paper_Design_Final_ENG-V2_Updated-1-9-22.pdf

[280] https://openid.net/specs/openid-connect-self-issued-v2-1_0.html

[281] https://www.keycloak.org/

[282] Department of Information and Communication Technology, Centre for e-Health, University of Agder, 4630 Kristiansand, Norway, 2022, Applying Spring Security Framework with KeyCloak-Based OAuth2 to Protect Microservice Architecture APIs: A Case Study.

[283] https://www.onelogin.com/

security through risk-based analysis. OneLogin's centralized administration dashboard simplifies user provisioning, role-based access control, and compliance monitoring, making it a powerful tool for efficiently managing user identities and access to digital resources.

OPENAM[284] is a powerful Access Management solution that provides advanced capabilities for securing user identities and controlling access to applications and services. Has features like Single Sign-On (SSO), Multi-Factor Authentication (MFA), and fine-grained authorization controls, OpenAM empowers institutions to establish robust security protocols. Its integration flexibility spans various protocols and platforms, ensuring seamless identity management across systems.

Okta[285, 286] is a leading Identity and Access Management (IAM) platform that empowers organizations with seamless authentication and secure access management. Providing Single Sign-On (SSO), Multi-Factor Authentication (MFA), and user provisioning. Its cloud-based architecture ensures easy integration with various applications, allowing businesses to manage user identities across different platforms.

AWS Identity and Access Management (IAM)[287] is a fundamental component of Amazon Web Services that enables businesses to manage user identities and access to AWS resources securely. Provides centralized control over user permissions, allowing organizations to grant fine-grained access to specific resources and services. Users can be assigned roles, groups, and permissions, streamlining user management, and reducing security risks. Its integration with AWS services ensures a secure and scalable environment, and IAM's support for Multi-Factor Authentication (MFA) adds an extra layer of protection.

Google Cloud Platform Identity and Access Management (IAM)[288] is a core component of Google Cloud that enables organizations to manage user identities and access to cloud resources effectively. Businesses can define granular permissions and roles, granting users and service accounts precise access to specific resources. IAM supports fine-grained control over permissions, ensuring security while avoiding over-privileged access. Its integration with Google Cloud services facilitates secure collaboration and resource management.

Microsoft Azure Active Directory (Azure AD)[289] is a comprehensive cloud-based identity and access management solution within the Microsoft Azure ecosystem. It empowers organizations to manage user identities and access to various applications and services. Azure AD supports Single Sign-On (SSO), Multi-Factor Authentication (MFA), and role-based access controls, enhancing security and user convenience. Its integration with Microsoft services and third-party applications ensures seamless access across platforms.

[284] https://github.com/OpenIdentityPlatform/OpenAM

[285] https://www.okta.com/

[286] Anastasios Liveretos, Ivo Draganov, Technical University of Sofia, 8 Kliment Ohridski Blvd., Sofia, 1756 Bulgaria, Customer Identity and Access Management (CIAM): An overview of the main technology Vendors.

[287] https://aws.amazon.com/iam/

[288] https://cloud.google.com/iam/

[289] https://www.microsoft.com/en-us/security/business/identity-access/microsoft-entra-id

5.4 SEARCHABLE ENCRYPTION

5.4.1 Searchable Encryption in PISTIS

Searchable encryption (SE) not only protects the privacy of data owners but also enables data users to search over the encrypted data. SE assumes that the user that wants to search over the encrypted data, owns the decryption key, and the sender must know the identity of the user querying the data to encrypt using the corresponding encryption key. In parallel, the encrypted data are shared between several receivers and kept in remote shared storage that is not trusted for confidentiality. Also, dynamic symmetric searchable encryption (DSSE) meets the demand of rapidly locating desired data among the huge amount of data in the cloud storage, while simultaneously assuring privacy. Thus, such a scheme is highly desirable in the context of PISTIS.

5.4.2 Methods

A few schemes exist in the literature that implements SE by making the encrypted data searchable[290]. Most of them generate an encrypted searchable index, by extracting metadata items. These metadata, often called as keywords and are encrypted with a technique that enables search operation over this index, while the actual data are encrypted with symmetric encryption techniques. In parallel, the symmetric encryption key may be encrypted with the same encryption technique which we used for generating the index. However, this searchable symmetric encryption (SSE) approach supports equality queries and was developed to handle just one keyword. Also, this approach has a restriction that it needs to break the data into fixed-size words which are incompatible with the existing encryption standard. This fixed size restriction, relaxed by using a direct index approach where an index is built for every data file using Bloom filters and a dictionary of search words respectively[291, 292]. Another approach is the inverted index that achieved sublinear search time[293]. More recent research works are focused on extracting an auxiliary index, thus there is no need to scan the whole document, and the search is accelerated[294, 295]. However, an important challenge is the development of dynamic schemes. The first efficient (e.g., sublinear complexity), dynamic symmetric

[290] Jiyi Wu, Lingdi Ping, Xiaoping Ge, Ya Wang, and Jianqing Fu. Cloud storage as the infrastructure of cloud computing. In 2010 International Conference on Intelligent Computing and Cognitive Informatics, pages 380–383. IEEE, 2010.

[291] Eu-Jin Goh. Secure indexes. IACR Cryptol. ePrint Arch., page 216, 2003.

[292] Yan-Cheng Chang and Michael Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings, volume 3531 of LNCS, pages 442–455, 2005.

[293] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. J. Comput. Secur., 19(5):895– 934, 2011.

[294] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel Catalin Rosu, and Michael Steiner. Dynamic searchable encryption in very-large databases: Data structures and implementation. In 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014.

[295] Seny Kamara and Tarik Moataz. Boolean searchable symmetric encryption with worst-case sub-linear complexity. In Jean-Sebastien Coron and Jesper Buus Nielsen, editors, ´ EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III, volume 10212 of LNCS, pages 94–124, 2017

searchable encryption (DSSE) scheme is presented by Kamara[296]. As it can be easily understood, a DSSE always encounters more challenges than a static one (e.g., new means of attack).

5.4.3 Technologies

This section presents libraries for searchable encryption.

OpenSSE

OpenSSE[297] is an open-source library for single-keyword searchable encryption implemented in C/C++. OpenSSE combines secure cryptographic primitives such as AES, RSA, BLAKE2 and elliptic curves, and focuses on both efficiency and security.

Clusion

Clusion[298] is a java-based library for searchable symmetric encryption that provides modular implementations of various SSE schemes. Clusion includes constructions that handle single, disjunctive, conjunctive and boolean keyword searches.

Jsse

Jsse[299] is a java-based symmetric searchable encryption developed by Sashank Dara. This library provides a full text search.

5.5 Secure Cryptographic Operations and Key Management

5.5.1 Secure Cryptographic Operations and Key Management in PISTIS

In the context of PISTIS, secure cryptographic operations, and key management are crucial parts. Cryptography is one of the most used techniques to build security and is an indispensable tool for protecting data in computer systems[300], while key management entails securing all stages of a key's lifecycle, such as generation, storage, distribution, and deactivation. Cryptography is used to store and share data in such a form that only the sender and the receiver can understand or process it. However, cryptography depends on both the used algorithm and the key length. PISTIS decentralized data management and sharing mandates the need for encryption/decryption techniques to achieve confidentiality and privacy (i.e., unlikability of the trading actions being performed during data trading) of the involved entities, while a proper key management mechanism will serve as the basis to achieve robust crypto operations.

[296] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Dynamic searchable symmetric encryption. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012, pages 965–976. ACM, 2012.

[297] OpenSSE, Website: https://opensse.github.io/

[298] Clusion, Website: https://github.com/encryptedsystems/Clusion

[299] Jsse, Website: https://github.com/sashank/jsse

[300] Ghulam Mustafa, Rehan Ashraf, Muhammad Ayzed Mirza, Abid Jamil, Muhammad: A review of data security and cryptographic techniques in IoT based devices. ICFNDS 2018: 47:1-47:9

5.5.2 Methods

Two are the main cryptographic approaches. The symmetric key cryptography and the asymmetric key cryptography. In addition, some other asymmetric cryptography modes have been proposed in the literature the identity-based and the attribute-based encryption.

Symmetric Key Cryptography

Symmetric key cryptography utilises a shared secret key between the sender and receiver to encrypt/decrypt the data. Several algorithms exist in the literature for symmetric key cryptography such as caesar cipher, block ciphers, stream ciphers, DES (Data Encryption Standard), and AES (Advanced Encryption Standard). However, the main issue of symmetric key cryptography is the need to exchange the secret key between the two parties in a secure manner[301].

Asymmetric Key Cryptography

Asymmetric key cryptography or public key cryptography utilises two different keys one for encryption (e.g., public key) and one for decryption (e.g., private key). These two keys are known as a public key and private key, where one the former is used for encryption and the latter is used for decryption. Again, several algorithms exist in the literature for asymmetric key cryptography such as Diffie-Hellman (DH), RSA (Rivest - Shamir - Adleman) and Elliptic Curve Cryptography (ECC). This method eliminates the need for the existence of a unique shared key between the communicating partners but requires more computational power compared to the symmetric cryptographic techniques[302].

Identity-based Encryption

A type of asymmetric key encryption is the identity-based encryption (IBE). In this specific type, the user's public key is the user's identity, while the private key is obtained from a Private Key Generator (PKG)[303].

Attribute-based Encryption

A recent promising type of asymmetric key encryption is the attribute-based encryption (ABE)[304]. More specifically, the ABE extends the IBE by enabling expressive access policies and fine-grained access to encrypted data. Both IBE and ABE utilise a Trusted Third Party (TTP) for key management. In ABE, this TTP usually called Attribute Authority (AA). Encryption in ABE is achieved with an access structure which is the logical expression of the access policy. Decryption can be done by any user if his secret key has the attributes that satisfy the access policy. The two main ABE variants have been proposed in the literature the Key-Policy

[301] Lisonek, D. AND Drahansky, M. 2008. SMS Encryption for Mobile Communication. In SECTECH '08: Proceedings of the 2008 International Conference on Security Technology. IEEE Computer Society, Washington, DC, USA, pp. 198-201.

[302] Lisonek, D. AND Drahansky, M. 2008. SMS Encryption for Mobile Communication. In SECTECH '08: Proceedings of the 2008 International Conference on Security Technology. IEEE Computer Society, Washington, DC, USA, pp. 198-201.

[303] Dan Boneh and Matthew Franklin. Identity-based encryption from the weil pairing. SIAM J. Comput., 32(3) :586615, March 2003.

[304] Amit Sahai and Brent Waters. Fuzzy Identity-Based encryption. In Ronald Cramer, editor, EUROCRYPT, volume 3494 of Lecture Notes in Computer Science, pages 457473. Springer, 2005.

Attribute-Based Encryption (KP-ABE)[305] and the Ciphertext Policy Attribute-Based Encryption (CP-ABE)[306].

5.5.3 Technologies

Several tools exist for providing key management and encryption/decryption functionalities. In this section we will focus on key management technologies as they are more complex. In addition, UBITECH owns cryptographic libraries (as an artefact from other research project) that provide traditional encryption/decryption functionalities.

EJBCA

EJBCA is a free software PKI certificate authority software package maintained and sponsored by the Swedish for-profit company PrimeKey Solutions AB, which holds the copyright to most of the codebase[307]. EJBCA offers multipurpose PKI software that supports multiple CAs and levels of CAs to enable one to build a complete infrastructure for multiple use cases within one instance of the software.

OpenDSU

OpenDSU[308] stands for Open Data Sharing Units and is a technology to solve data self-sovereignty based on Distributed Ledger Technology (DLT) / blockchain. OpenDSU defines a standard of how to store data and code outside the Blockchain (off-chain). It provides a set of tools and services that can be used to create, store, and manage data on top of any blockchain technology. This includes among others access control and secure communication channels. In other words, OpenDSU is a prominent tool that manages SSI digital identities, keys and data stored on- and off-chain in seamless and agnostic to the underline blockchain technology way.

6 CONCLUSIONS

The document provides a short overview of methods, models and technologies, which can be used to drive the PISTIS Data Space Factory Environment services dealing with the data management, data interoperability, data quality improvement/assessment and data protection within the PISTIS ecosystem. The goal of this report is to create awareness in the PISTIS consortium about them. The authors of the report don’t aim to make any decisions regarding methods or technologies to be used in PISTIS at this stage. It will be done at the next step in the project and will be presented in the next deliverables D2.2 “Data Management and Protection services - Alpha version” and D4.1” PISTIS Reference Architecture and API Documentation”.

[305] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for ne-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security, CCS '06, pages 8998, New York, NY, USA, 2006

[306] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-Policy Attribute-Based encryption. In Proceedings of the IEEE Symposium on Security and Privacy, SP '07, pages 321334, Washington, DC, USA, 2007

[307] EJBCA Enterprise from PrimeKey, Website: https://www.primekey.com/products/software/ejbca-enterprise/

[308] OpenDSU, Website: https://opendsu.com/