Cyprus University of Technology
Faculty of Engineering and Technology

Doctoral Dissertation

Modern Data Storage Architectures for Managing Big Data: The Role of Semantically Enrichment Mechanisms in Data Management and Security

Michalis Pingos

Limassol, May 2025

CYPRUS UNIVERSITY OF TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING, COMPUTER ENGINEERING AND INFORMATICS

Doctoral Dissertation

Modern Data Storage Architectures for Managing Big Data: The Role of Semantically Enrichment Mechanisms in Data Management and Security

Michalis Pingos

Supervisor:

Dr Andreas S. Andreou

Professor

Limassol, May 2025

Approval Form

Doctoral Dissertation

Modern Data Storage Architectures for Managing Big Data: The Role of Semantically Enrichment Mechanisms in Data Management and Security

Presented by

Michalis Pingos

Supervisor: Dr. Andreas S. Andreou, Professor, Cyprus University of Technology

Signature: _____

Member of the committee: Dr. Herodotos Herodotou, Associate Professor, Cyprus University of Technology

Signature: _____

Member of the committee: Willem-Jan van den Heuvel, Professor, Tilburg University, Netherlands

Signature: _____

Cyprus University of Technology

Limassol, May 2025

Copyrights

Acknowledgements: I would like to express my special appreciation and say a big thank you to my advisor Professor Andreas S. Andreou. Without his guidance and constant support, this PhD would not have been achievable. I am grateful for giving me the opportunity to work with him and become a member of his team.

I would also like to thank my committee members, Associate Professor Herodotos Herodotou and Associate Professor Michalis Michaelides who devoted much of their precious time to read my thesis and provide me valuable comments and suggestions.

Furthermore, I would especially like to thank the members of the Software Engineering and Intelligent Information Systems Research Laboratory for their support and outstanding cooperation we had these years.

An exceptional thanks to my family, my parents Frixos Pingos and Niki Pingou, for their persistence, love and support.

Last, but most important, I would like to express my deep gratitude to my beloved wife, Emmanouela Manoli and my son Nicholas Pingos. Their endless love and support were the driving force behind what I have achieved so far.

ABSTRACT

This PhD thesis moves in the broader area of Smart Data Processing (SDP) and Systems of Deep Insights (SDI) and focuses on Big Data storage and management, addressing significant challenges such as optimizing data access, security, and retrieval. It explores current approaches for efficiently managing data sources, their organization, and storage for seamless access and retrieval while addressing challenges related to data integrity, privacy, and access control. A key contribution of this research is the development of a semantically enriched Data Lake framework, which enhances data structuring, accessibility, and governance by leveraging metadata-driven semantic data blueprints (SDB) supporting also process mining. Empirical findings demonstrate that Data Mesh architectures significantly outperform traditional Data Lakes, offering improved scalability, flexibility, and decision-making agility. The thesis demonstrates how transitioning from centralized Data Lakes to decentralized, semantically enriched Data Meshes enables enhanced data discoverability, real-time insights, and secure cross-organizational collaboration. The application of the aforementioned concepts in a smart manufacturing environment showcases how metadata-driven Data Meshes streamline operational efficiency, improve data traceability, and facilitate decentralized access control mechanisms. The integration of Blockchain technology and Non-Fungible Tokens (NFTs) further strengthens data ownership, integrity, and secures access management in Data Lakes and Data Meshes. Through experimental evaluation using real-world industrial data, research conducted highlights the effectiveness of the proposed framework in optimizing data workflows, reducing processing delays and enhancing security. This research provides valuable methodologies for enterprises seeking to harness the power of Big Data, fostering a more intelligent, secure, and adaptive data management paradigm.

Keywords: Big Data, Data Lakes, Data Meshes, Semantic Enrichment, Metadata, Data Blueprints, Blockchain Technology, Process Mining, Smart Data Processing, Systems of Deep Insight.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SDP:	Smart Data Processing
SDI:	Systems of Deep Insights
SDB:	Semantic Data Blueprint
RDF:	Resource Description Framework
XML:	Extensible Markup Language
AI:	Artificial Intelligence
SPARQL:	Protocol and Query Language
API:	Application Programming Interface
ML:	Machine Learning
RL:	Record Linkage
KG:	Knowledge Graph
DT:	Digital Twin
AR:	Augmented Reality
IoT:	Internet of Things
LLMs:	Large Language Models

LIST OF PUBLICATIONS

1) Pingos, M. and Andreou, A. (2022). A Data Lake Metadata Enrichment Mechanism via Semantic Blueprints. In Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE, ISBN 978-989-758-568-5; ISSN 2184-4895, pages 186-196.
(Candidate for Best Paper Award) DOI: 10.5220/0011080400003176

2) Pingos, M., & Andreou, A. S. (2022). Exploiting Metadata Semantics in Data Lakes Using Blueprints. In International Conference on Evaluation of Novel Approaches to Software Engineering (pp. 220-242). Cham: Springer Nature Switzerland. DOI: 10.1007/978-3-031-36597-3_11

3) Pingos, M., Christodoulou, P., & Andreou, A. (2022). DLMetaChain: An IoT Data Lake Architecture Based on the Blockchain. In 2022 13th International Conference on Information, Intelligence Systems & Applications (IISA) (pp. 1-8). IEEE.

4) Pingos, M. and Andreou, A. (2022). A Smart Manufacturing Data Lake Metadata Framework for Process Mining. In Proceedings of the Seventeenth International Conference on Software Engineering Advances ICSEA 2022 DOI: 10.5281/zenodo.7501059
(Best Paper Award) DOI: 10.1109/IISA56318.2022.9904404

5) Andreou, A. S., Firmani, D., Mathew, J. G., Mecella, M., & Pingos, M. (2023). Using Knowledge Graphs for Record Linkage: Challenges and Opportunities. In International Conference on Advanced Information Systems Engineering (pp. 145-151). Cham: Springer International Publishing. DOI:10.1007/978-3-031-34985-0_15

6) Pingos, M.; Mina, A. and Andreou, A. S. (2024). Transforming Data Lakes to Data Meshes Using Semantic Data Blueprints. In Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering, ISBN 978-989-758-696-5, ISSN 2184-4895, pages 344-352. DOI: 10.5220/0012620200003687

7) Loizou, S.; Pingos, M. and Andreou, A. S. (2024). Enhancing Interaction with Data Lakes Using Digital Twins and Semantic Blueprints. In Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering, ISBN 978-989-758-696-5, ISSN 2184-4895, pages 353-361. DOI: 10.5220/0012620600003687

8) Pingos, M.; Panayiotis Christodoulou, and Andreas S. Andreou. 2024. Security and Ownership in User-Defined Data Meshes Algorithms MDBI Journal 17, no. 4: 169. DOI: 10.3390/a17040169

9) Pingos, M.; Andreou, A.S. Discovering Data Domains and Products in Data Meshes Using Semantic Blueprints. Technologies 2024, 12, 105. https://doi.org/10.3390/technologies12070105

10) Pingos, M.; Loizou, S. and Andreou, A. S. (2025). Integrating Data Lakes with Self-Adaptive Serious Games. In Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering, ISBN 978-989-758-742-9, ISSN 2184-4895, pages 762-772. DOI: 10.5220/0013467700003928

CHAPTER 1 : INTRODUCTION

Big Data has been called “the new oil”, recognized as a valuable human asset that, with proper collation and analysis, can yield information that generates deep insights into various aspects of our everyday lives and enables us to predict future trends. Big Data is essentially a combination of structured, semi-structured, and unstructured data primarily originating from five sources: media, cloud, web, traditional business systems, and the Internet of Things (IoT) (Chen et al., 2014). The speed and frequency with which digital data is produced and collected from an increasing variety of sources are projected to increase exponentially. This surge in data volume, coupled with its immense social and economic value (Bertino, 2013; Günther et al., 2017), is driving a global data revolution.

The rapid growth of data across sectors, such as manufacturing and healthcare, has further propelled the concept of Big Data (Cristaldi et al., 2023), characterized by its volume, velocity, and variety often referred to as the 3Vs (Laney, 2001). In 2020, an astounding 1.7 megabytes of data were generated per person every second (Beckman, 2023), significantly transforming how organizations operate and presenting vast opportunities for innovation, efficiency, and competitive advantage. For instance, Netflix has effectively utilized data to implement customer retention strategies, saving approximately $1 billion annually (20 Data Hygiene Statistics, 2023). Conversely, poor data quality costs U.S. businesses over $3 trillion each year, accounting for roughly 12 percent of revenue for individual organizations. By 2023, the world is projected to generate 120 zettabytes of data, with forecasts suggesting this will reach 181 zettabytes by 2025 (Amount of Data Created Daily, 2024; Beckman, 2023). Furthermore, the market for data analytics is expected to grow to $103 billion. These statistics underscore the critical importance of effective Big Data management and analytics, highlighting their rapid growth and significant economic impact.

Consequently, traditional data management and processing approaches, such as Relational Database Management Systems (RDBMS), Data Warehouses (DW), and classic methods like Extract, Transform, Load (ETL), are increasingly inadequate for managing this deluge of data (Khan et al., 2018). In response, the concept of Data Lakes was introduced to provide a solution for storing fast vast amounts of raw data in its native format, enabling cost-effective storage and facilitating advanced analytics and machine

learning (Beheshti et al., 2018). However, while Data Lakes effectively store large volumes of raw data at high speeds, they often encounter challenges related to data governance, quality, and accessibility. To address these issues, Data Meshes have emerged as an evolution of Data Lakes, distributing data ownership across various data domains and product teams, thereby enhancing scalability and agility in data processing (Dolhopolov et al., 2023). The transformation of Data Lakes into Data Meshes has become an essential challenge for efficiently managing Big Data in a domain-based context.

This research explores SDP and SDI within the context of Big Data storage and management, addressing the increasing challenges in data organization, retrieval efficiency, and security. Traditional Data Lakes face critical limitations in governance, accessibility, and scalability, often resulting in inefficient data utilization. To address these issues, this research introduces a semantically enriched Data Mesh framework, which leverages metadata-driven blueprints and process mining techniques to enhance data structuring, governance, and interoperability. By decentralizing data management, the framework improves data discoverability, operational agility, and security. Additionally, it integrates Blockchain technology and Non-Fungible Tokens (NFTs) to fortify data ownership verification, access control, and privacy mechanisms, mitigating risks associated with unauthorized data access and integrity breaches.

Empirical evaluations using real-world industrial datasets—particularly in smart manufacturing environments demonstrate that Data Mesh architectures significantly enhance data retrieval efficiency, streamline workflows, and enable secure cross-organizational data exchange. The framework also improves decision-making agility by facilitating process mining and decentralized governance. Findings reveal that transitioning from centralized Data Lakes to decentralized Data Meshes not only enhances data quality and traceability but also optimizes resource allocation and reduces operational inefficiencies. This research contributes a robust and scalable methodology for modern data management, equipping enterprises with the tools necessary to navigate the complexities of Big Data, maximize security, and improve data-driven strategic outcomes

The investigation is guided by three primary research questions. RQ1 aims to define the scope of existing scientific research on SDP, SDI and metadata enrichment within modern

Big Data storage architectures, particularly focusing on Data Lakes and Data Meshes. This question also explores key security challenges, including data integrity, privacy, and access control. RQ2 seeks to synthesize recent findings on the interconnections between Smart Data Processing, System of Deep Insights, Data Lakes, and Data Meshes, emphasizing advancements in data management, organization, and retrieval. Additionally, it investigates how emerging security enhancements, such as Blockchain technologies and Non-Fungible Tokens (NFTs), contribute to data governance and protection. RQ3 examines the challenges associated with Big Data management, focusing on the evolution of security frameworks and assessing the role of visual querying and process mining in optimizing data structuring, access control, and security mechanisms. The answers to these research questions will inform future research directions and practical implementations, equipping organizations with innovative strategies to manage and secure complex data ecosystems.

Through a thorough examination of these topics, this research highlights several key challenges and opportunities, including the need for robust security measures that leverage Blockchain to ensure data integrity and privacy, the role of Non-Fungible Tokens (NFTs) in facilitating secure data transactions, and the necessity of developing effective strategies for transitioning Data Lakes to decentralized Data Mesh architectures.

Essentially, this PhD presents an extended framework described partially in each chapter aimed to enhance Data Lakes through a semantic enrichment mechanism utilizing metadata blueprints. By integrating the 5Vs of Big Data Volume, Velocity, Variety, Veracity, and Value at first, the framework addresses challenges in data processing and retrieval within a pond-structured Data Lake architecture, which accommodates diverse data types. A real-world case study conducted showcased the framework's effectiveness in identifying operational delays and bottlenecks through process mining. The insights derived from this analysis enabled the optimization of task sharing among personnel and machinery, thereby improving overall production efficiency. Additionally, the extended framework employs Blockchain technology and NFTs to enhance data security, ownership verification, and governance.

Furthermore, this work explores the transformation of Data Lakes into Data Meshes by establishing standardized approaches for the discovery and construction of Data Products. While the creation of a Data Mesh may require initial time investment, findings indicate

that Data Meshes facilitate rapid data retrieval and enhanced performance compared to traditional architectures. The integration of Data Meshes with software analytics provides granular insights and contextual awareness, promoting agile product enhancements. By leveraging Blockchain and NFT technologies, the framework ensures secure access and transfer of ownership within the Data Mesh, thereby addressing critical security concerns and fostering a robust governance model. This approach not only enhances decision-making capabilities but also supports efficient management of manufacturing data, ultimately contributing to improved operational outcomes.

The following sections of this thesis will delve into essential theoretical and technical backgrounds that underpin the study, providing first a comprehensive literature overview to contextualize the research. The exploration continues with an analysis of how to exploit metadata semantics in Data Lakes using blueprints, followed by the introduction of a scalable semantic framework utilizing ponds and puddles enhancing granularity. A significant focus will be placed on enhancing Data Lake security and metadata management through Blockchain integration. The discussion will then transition to the transformative process of converting Data Lakes into Data Meshes, utilizing semantic data blueprints to discover data domains and products within these decentralized structures. Furthermore, considerations of security and ownership in user-defined Data Meshes will be examined. Finally, the thesis concludes with reflections on the findings and outlines ongoing and potential future research avenues that could further advance this field.

By addressing the aforementioned critical areas, this thesis aspires to contribute significantly to the field of data management, offering insights that can inform both academic discourse and practical applications. The findings will serve as a resource for organizations seeking to navigate the complexities of Big Data, while maximizing the security and utility of their data assets in a rapidly evolving technological landscape.

CHAPTER 2 : LITERATURE OVERVIEW

This chapter firstly presents the approach adopted to search and classify relevant papers in literature that address the research questions posed in this study (see below), and secondly it outlines the main findings and results. Quantitative features are also provided in the form of bar charts to make it easier for the reader to identify the work performed categorized in areas, the venue, as well as the year of publication.

2.1 Material and Methods

This PhD study is motivated by three research questions:

- RQ1: What is the scope of scientific research focused on Smart Data Processing (SDP), Systems of Deep Insight (SDI) in terms of big data storage and management using specific data storage architectures.
- RQ2: What are the primary aspects and findings from recent studies exploring the current approaches for efficiently managing data sources and their organization for easy data access storing and retrieval.
- RQ3: What are the challenges of approaches for big data management considering also security aspects such as data integrity, privacy, and access control.

The two research pillars of the research study are:

- Smart Data Processing in Modern Data Storage Architectures: This research pillar encompasses various stages including data ingestion, aggregation of diverse datasets (structured, unstructured, semi-structured), utilization of knowledge-based metadata representation techniques to transform raw data into smart data, and considerations for data privacy and protection.
- Smart Data Management and its role for supporting Systems of Deep Insight: This research pillar focuses on new approaches to handle the complexity of the production and the handling of huge volume of data. It also emphasizes on how to efficiently utilizing this volume of data to support systems of deep insight that effectively transform data into actionable insights. These systems systematically test and contextualize insights, identifying critical data through advanced mechanisms such as Data Lakes and Data Meshes.

In order to provide answers to these research questions, a methodology was applied firstly to gather enough material and then assess the positions and statements made; this methodology is described in this section. As a first step, the guidelines for a systematic literature review (SLR) proposed by Kitchenham et al. (2010) were followed. Although an extended SLR is outside of the scope of this thesis, those guidelines assisted in organizing the process of finding and classifying relevant works.

The search process aimed at locating articles indexed in Scopus, Science Direct, IEEE Xplore, ACM Digital Library, SpringerLink, Google Scholar and Wiley Online. The general search strings used were "Smart Data Processing Systems", "Systems of Deep Insight" and "Big Data Management". Additional, more refined searches were conducted using the following strings: "data ingestion", "data aggregation", "structured datasets", "unstructured datasets", "semi-structured datasets", "knowledge-based metadata representation techniques", "conversion of raw data into smart data", "data privacy and protection", "Blockchain", "run-time software performance monitoring and dynamic configuration", "Big Data", "data lakes ", "data meshes ", "data warehouses", "NFTs" "optimization in data processing", "data analytics", "business intelligence", "turn data into insights", "contextual and actionable data", "Visual Queries", "SmartPM" , "process models". The search results consisted of articles published up to 2020.

As Smart Data Processing and Deep Insights is a very recent topic, both journal and conference articles were considered. Finally, duplicate papers were removed from the results, since the search engines and databases produced overlapping results to a certain extent. After these steps, the initial collection consisted of 93 potentially relevant works. Then, a detailed, qualitative analysis was performed by examining closely these papers in order to identify and merge different papers of the same authors/groups reporting their results incrementally and also works that used the term "Smart Data" with a different meaning compared to the target of this survey. In addition, the snowballing approach was used (see e.g., Wohlin 2014) based on a set of four different types of criteria applied on the initial list of papers. This set of criteria consisted of the type of each paper, publication year, publication venue and number of citations. Only scientific papers published in recognized venues with a significant number of citations were included in the final set of papers, which was organized into several categories presented in the following section.

2.2 Primary studies

As mentioned above, the final list of papers consisted of 93 studies, which were organized in several categories based on their content. Figure 1 (a) shows this list of papers by year of publication. It is worth noting that in each of the years 1996, 2004, 2005, 2011, 1998 and 2011 only one paper was published. In the subsequent years up until and including 2006 the number of publications was slightly increased, while from 2012 to 2020 a significant uptake is observed gradually doubling this figure. This may be attributed to the fact that SDP and SDI, as relatively emerging scientific fields, initially received limited research attention. However, as the challenges associated with Big Data management became more evident, interest in these fields gradually increased, leading to a growing body of research over time.

Figure 1 (b) shows a categorization of the material gathered in this survey by type: book chapters, conference papers, and journals papers and business reviews. In their majority, the articles were published in journals and then in conferences. More specifically, fifty-seven (57) of them are journal articles, twenty-eight (28) appear in conference proceedings, six (6) of them appear in workshops, one (1) is a book chapter and one (1) a business review.

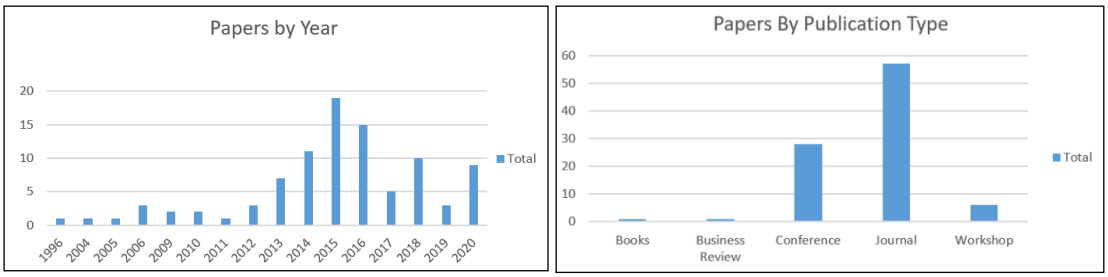

Figure 1. Distribution of papers by year of publication (a) and categorization by type (b).

Based on the content orientation, the papers were split into the two categories formed by the research questions: (i) Smart Data Processing Systems, (ii) Systems of Deep Insight, Papers belonging to the first category mainly revolve around data ingestion and data curation, data aggregations, stream-processing, knowledge-based meta data and scalable, reusable and secure processing frameworks. The second category consists of papers which are mainly involved with large-scale data analytics, cross-correlation and cross functional model, actionable, context-depend deep insights, decision monitoring and next

best action and descriptive, predictive cognitive analytics. The following section outlines the main findings of the articles selected in each of the aforementioned categories, focusing on the research challenges faced, their methodology used, their results and the open problems reported.

2.3 Literature Review

The area of SDP comprises the ability to define, interoperate, share, access, transform, and manage data efficiently. It leverages knowledge-based metadata representation techniques to structure diverse datasets, annotate them, establish links with associated processes and software services, and enable seamless data syndication and retrieval within modern Big Data architectures such as Data Lakes and Data Meshes. SDP integrates data ingestion and aggregation methodologies to manage a vast array of structured, semi-structured, and unstructured datasets, ensuring efficient organization and accessibility. Furthermore, process mining techniques and visual querying play a crucial role in optimizing data management and enhancing interoperability.

Security and governance are also critical aspects of SDP, incorporating mechanisms for data privacy, integrity, and access control through Blockchain technologies and Non-Fungible Tokens (NFTs). Additionally, automated deployment, run-time software performance monitoring, and dynamic configuration enhance adaptability and scalability in data-driven environments. To further optimize efficiency, SDP relies on adaptive frameworks and tool-suites that handle both data in motion (e.g., real-time sensor streams) and data at rest, employing advanced resource management techniques for workload partitioning across private and public cloud infrastructures. SDP also facilitates the integration of diverse data sources, including Hadoop, NoSQL databases, Data Lakes, Data Warehouses, Data Meshes, IoT devices, social platforms, and Software-as-a-Service (SaaS) applications. By creating a unified data ecosystem, SDP enhances Big Data analytics, decision-making, and security, supporting organizations in efficiently managing vast and complex data environments. (Yuhanna, 2017).

Most of the work on Big Data integration has been focused on the problem of processing very large sources, extracting information from multiple, possibly conflicting data sources, reconciling the values and providing unified access to data residing in multiple, autonomous data sources. Various studies mainly addressed isolated aspects of data

source management relying on schema mapping and semantic integration of different sources (Cafarella et al. (2009), Hassanzadeh et al. (2013) and Venetis et al. (2011)). Those studies focused mostly on the construction of a global schema or a knowledge base to describe the domain of the data sources. Web table search is also closely related to data source search. Most of the proposed techniques outlined in Cafarella et al. (2009), Limaye et al. (2010), Das Sarma et al. (2012), Yakout et al. (2012) and Fan et al. (2014), examine user queries and return tables related to specific keywords presented in the query; however, keyword-based techniques fail to capture the semantics of natural language, i.e., the intentions of the users, and thus they can only go as far as giving relevant hits.

Common fields of data processing systems are semantic models, structured data configurations, data lakes, data warehouses, data meshes and ontologies. One of the most significant findings in these studies are the importance of using data lakes architecture to store relational and non-relational large amount of data combining them with traditional data warehouses. Another notable finding is the exploitation of ontology frameworks in order to manage and make heterogenous data sources that produce large amounts of data meaningful. Finally, another major finding in these studies is the need for upgrading Data Lakes to a decentralized place via transforming to / or adopt Data Mesh concept.

Sawadogo and Darmont (2020) provide a comprehensive state of the art of the different approaches to data lakes design. They particularly focus on data lake architectures and metadata management, which are key issues in successful data lakes. The authors also discuss the pros and cons of data lakes and their design alternatives. Finally, they classify metadata and introduce the features that are necessary to achieve a full metadata system.

Lanzenberger et al. (2010) examined an enormous number of ontology visualization tools to identify solutions for dealing with the complexity of large ontologies. Their work was a starting point to demonstrate the usefulness of Information Visualization techniques, aimed to boost the adoption of ontologies in common Web applications.

Yang et al. (2008) adopted an expressive Web Ontology Language (OWL) and a Semantic Web Rule Language (SWRL) to model product configuration knowledge which have the advantage of reusing configuration models that is crucial, considering incremental changes and updates on products due to new technology advances.

Roda and Musulin (2014) propose an ontology-based framework for Intelligent Data Analysis (IDA) which is based on a knowledge model composed by existing ontologies, the Semantic Sensor Network ontology (SSN) and the SWRL Temporal Ontology (SWRLTO), and a new developed one, the Temporal Abstractions Ontology (TAO). They demonstrate their framework by using it in a chemical plant case study to show how complex temporal patterns that combine several variables and representation schemes can be used to infer process states and/or conditions.

The work of Petersen et al. (2017) mentions that the digitization of the industry requires information models that describe assets of companies to enable the semantic integration and interoperable exchange of data. Their proposed model is centered around machine data and describes all relevant assets, key terms and relations in a structured way. They evaluated their approach with stakeholders on two case studies. While the stakeholders find the advantages of semantic technologies appealing, the lack of ready-to-use business solutions, industrial ontologies and available IT personnel is halting their efforts to move forward.

Drabent et al. (2009) firstly outline the current state of the Semantic-Web stack and its components, and then discuss the open issues in combining rules and ontologies before defining a combined rule and ontology knowledge-base two-step redact, in which, as a first step, the ontology predicates are eliminated under the open-world assumption (OWA) and, as a second one, the negated logic-programming predicates under the closed-world assumption (CWA).

Mehdi et al. (2017) report that industrial rule-based diagnostic systems are often data-dependent in the sense that they rely on specific characteristics of individual pieces of equipment. This dependence poses significant challenges in rule authoring, reuse, and maintenance by engineers. That work addressed the aforementioned problems by proposing a semantic rule language, sigRL, where sensor signals are first class citizens. Their evaluation shows that up to 66% of the time is saved when employing ontologies and that execution of semantic rules is efficient and scales well to real-world complex diagnostic tasks.

Cuenca et al. (2016) describe the outcomes of an ongoing collaboration between Siemens and the University of Oxford, with the goal of facilitating the design of ontologies and

their deployment in applications. They present SOM, a tool that supports engineers in the creation of ontology-based models and in populating them with data.

Bock et al. (2010) show how to combine ontological and model-based techniques in languages that facilitate collaborative design exploration. The proposed approach uses ontology to capture alternative designs and incremental refinements that meet requirements and earlier design commitments. In this work, model-based techniques are applied to develop more powerful, engineering-friendly languages for using ontology.

Jørgensen (20010) introduced fundamental concepts of product configuration. The introduction and implementation of product configuration demand a systematic way of thinking in constructing, documenting, and maintaining the configurable products. This can be achieved by defining a product family model as a model of a set of possible products.

Lee et al. (2011) state that product knowledge has played an increasingly significant role in new product development process especially in the development of One-of-a-Kind products. Their paper provides a comprehensive review on the recent development of knowledge-based systems (KBS), methods and tools in supporting rapid product development.

The Internet of Things (IoT) is nowadays a vital source of data, both in terms of volume and frequency of production. Quite a few papers are devoted to the study of problems pertaining to the collection, structuring, processing and presentation of IoT data towards the development of new applications and services. Lee and Lee (2015) firstly identify the mostly widely used IoT technologies that are essential in the deployment of successful IoT-based products and services and then discuss the three IoT categories for enterprise applications to enhance customer value. Qin et al. (2016) review the main techniques and state-of-the-art research efforts in IoT from data-centric perspectives, including data stream processing, data storage models, and complex event processing. This paper covers investigations on data models, search and event processing, and present the potential of IoT applications in smart cities, environment monitoring, health and energy home.

Data structuring, organization and fast processing have also gained significant interest during the last decades, with studies investigating a rich number of relevant issues. Over the years a rich ecosystem emerged around Hadoop comprising tools for parallel, in-

memory and stream processing. Luckow et al. (2015) survey use cases and applications for deploying Hadoop in the automotive industry and argue about the need to develop automotive applications and requirements for data discovery, integration, exploration and analytics. Dean and Ghemawat (2008) outline the novel programming model MapReduce, which has been successfully used by Google for many different purposes. The authors attribute this success to several reasons: Firstly, the model is easy to use, even for programmers without experience with parallel and distributed systems. Secondly, a large variety of problems are easily expressible as MapReduce computations. Thirdly, an implementation of MapReduce has been developed those scales large clusters of machines.

Guerrero et al. (2017) propose a heterogeneous data source integration based on IEC (Electrotechnical Committee) standards and metadata mining. The system includes several data mining tools to model information for classification, outlier detection, pattern detection, forecasting, or information retrieval based on the level of importance established by metadata mining process. Erkin et al. (2013) present recent and ongoing research in the field of privacy protection for smart grids, where individual smart meter measurements are kept secret from outsiders, including the utility provider itself, while processing private measurements under encryption is still feasible. The authors focus particularly on data aggregation, which demonstrates the major research challenges in privacy protection for smart grids and conclude that researchers should invest more in cryptography.

Data Lakes related research is also rich. Miloslavskaya and Tolstoy (2016) firstly state that a data lake holds a vast amount of raw data in its native format and then define fast data as a time-sensitive structured and unstructured "in-flight" data that should be gathered and acted upon right away. The authors conclude that not all Big Data is fast, as well as not all fast data is big. Khine and Wang (2018) argue that a data lake is one of the arguable concepts appeared in the era of Big Data. The idea of a data lake originated from business field instead of the academic. As data lake is a newly conceived idea with revolutionized concepts, it brings many challenges for its adoption. However, the potential to change the data landscape makes the research on data lakes worthwhile. Fang (2015) discusses the concept of data lakes and shares the author's thoughts and practices on the subject. The main goal of the paper is to examine and provide answers to a series

of questions, such as what is a data lake, or how does it help with the challenges posed by Big Data. The author concludes that the data warehouse is a wise choice for a company dealing with Big Data challenge and outlines the best practices of data lake implementations.

The area of SDI focuses on analytic solutions that optimize asset performance in smart data processing systems, enabling data-driven decision-making. It leverages advanced data integration techniques to extract meaningful insights from vast and complex datasets, facilitating real-time analytics, pattern recognition, and predictive modeling. SDI enhances the ability to correlate, contextualize, and analyze diverse data sources, improving operational efficiency and strategic planning. Additionally, it explores methods for uncovering hidden relationships and trends within data, incorporating visual querying, process mining, and automated analytics to enhance data accessibility, security, and governance. By integrating semantic enrichment, metadata frameworks, and decentralized architectures, SDI may support intelligent, adaptive data ecosystems, ensuring scalability, efficiency, and enhanced decision-support capabilities.

Process mining is an emerging research discipline that helps organizations discover and analyze business processes based on raw event data. Basically, it sits between computational intelligence and data mining on one hand, and process modeling and analysis on the other (Van Der Aalst et al., 2007). Many researchers are developing new and more powerful process mining techniques and software vendors are incorporating these in their software and especially now to the world of Big Data (Van Der Aalst et al., 2012). Generally, process mining techniques based on the business log files produced. There are three types of process mining activities, discovery, conformance checking, and enhancement. These activities use an existing process model produced based on event logs. Companies and organizations tend to produce their log files according to their own data standards. Therefore, a standardization model is needed, to unify and formalize the description of all business entities in the enterprise under analysis, allowing to efficiently monitor and extract knowledge from event logs.

Various studies belong to this category, investigating topics which include Big Data, services, Cyber Physical Systems (CPS), business intelligence, machine learning techniques and algorithms, and various applications to real-world problems which involve models and systems that provide insights for decision support, optimization and

control. The most significant findings in this area's literature review is the importance of predictive and prescriptive analytics which improve and strengthen monitoring, reconfiguration, self-adaption, discovery, decision making, process effectiveness, actions and intelligence in many application areas such as Health Care, Smart Cities, Smart Manufacturing. In order to achieve the aforementioned results and according to the literature review, machine learning and algorithms play a ''vital'' role because they enrich and strengthen the ability to predict and prescript utilizing smart data processing systems (previous research pillar).

Barnaghi et al. (2013) describe the Big Data issues in the Web of Things (WoT), discuss the challenges of extracting actionable knowledge and insights from raw sensor data, and introduce the theme articles in this special issue. The authors demonstrate different steps that can be envisaged for efficient processing and for making use of WoT data.

Delen and Demirkan (2013) provide a conceptual framework for service oriented managerial decision-making process, and briefly explain the potential impact of service-oriented architecture (SOA) and cloud computing on data, information and analytics. The authors believe that their proposed approach to service-oriented data, information and analytics in the cloud will create great opportunities, as well as many challenges.

Stojmenovic (2014) explores Cyber Physical Systems (CPS) beyond the M2M (Machine to Machine) concept before describing a number of particular use cases that motivate the development of the M2M communication primitives tailored to large-scale CPS. The author argues that there is a need to design M2M communication primitives able to scale to thousands and trillions of M2M devices, without sacrificing solution quality.

Larson and Chang (2016) examine the application of Agile methodologies and principles to data-driven business intelligence delivery and discuss how these methodologies also changed with the evolution of business intelligence. In addition, the authors address how Agile principles and practices have evolved with business intelligence, as well as their challenges and future directions.

Wang, et al. (2019) present a new deep learning-based machine vision inspection method to identify and classify defective product without the loss of accuracy. More specifically, firstly a Gaussian filter is utilized on an acquired image to minimize the random noise and secondly, a region of interest (ROI) is conducted based on the Hough transformation

to remove the unrelated background, thereby offloading the computational burden of the subsequent identification process. The experimental study on defective bottles inspection demonstrates the usefulness of the proposed method.

Lee et al. (2014) discuss the trends of manufacturing service transformation in Big Data environments, as well as the readiness of smart predictive informatics tools to manage Big Data, thereby achieving transparency and productivity. The objective of the paper is to review how current manufacturing industries evolve for the upcoming industrial Big Data environment, and to propose the key technology for sustainable innovative service.

The work of Yan et al. (2011) sets the ground for research on home power management systems optimization as regards to the privacy of customer power usage behaviors. The performance of the reading data aggregation and dispatch has been analyzed subject to the HAN setting. The levels of security were discussed qualitatively, focusing on the secrecy of pseudo-random spreading codes and circuit shift. Simulation results demonstrated the advantage of the proposed scheme over the traditional BSS approach.

Kim (2017) presents a new transactional scheduler, called partial rollback-based transactional scheduler (or PTS), for a multi-versioned DTM (Distributed Transactional Memory) model. The model supports multiple object versions to exploit concurrency of read-only transactions and detects conflicts of write transactions at an object level. PTS's design shows that partial rollback-based scheduling is a viable strategy for transactional processing in in-memory data grids.

Mahdavinejad et al. (2018) assess various machine learning methods that deal with the IoT data challenges extracted from a smart city use case. The key contribution of this study is the presentation of a taxonomy of machine learning algorithms explaining how different techniques are applied to the data in order to extract higher level information.

Chen et al. (2012) initially argue that business intelligence and analytics (BI&A) has emerged as an important area of study for both practitioners and researchers, reflecting the magnitude and impact of data-related problems to be solved in contemporary business organizations. They continue their work by reporting a bibliometric study of critical BI&A publications, researchers, and research topics based on more than a decade of related academic and industry publications.

Farid et al. (2016) present CLAMS, a system to discover and enforce expressive integrity constraints from large amounts of lake data with very limited schema information (e.g., represented as RDF triples). CLAMS has been deployed in a real large-scale enterprise data lake compromising 1.2 billion triples and was able to spot multiple obscure data inconsistencies and errors early in the data processing stack, providing huge value to the enterprise. This paper shows how CLAMS holistically combines the signals from diverse constraints spanning over multiple datasets and utilize user feedback to obtain accurate repairs.

Bertsimas and Kallus (2020) combined ideas from Machine Learning (ML) and operations research and management systems (OR/MS) in developing a framework, along with specific methods, for using data to prescribe optimal decisions in OR/MS problems that leverage auxiliary observations. They motivate their methods based on existing predictive methodology from ML, but, in the OR/MS tradition, focus on the making of a decision and on the effect on costs, revenues, and risk. The authors support that their approach is generally applicable, tractable, asymptotically optimal, and leads to substantive and measurable improvements in a real-world context.

Aceto et al. (2013) provide a survey on Cloud monitoring. The authors start by analysing motivations for Cloud monitoring, providing also definitions and background for their following contributions. Then, they analyse and discuss the properties of a monitoring system for the Cloud, the issues arising from these properties and how such issues have been tackled in literature.

Charest and Delisle (2006) propose the realization of a hybrid intelligent data mining assistant, based on the synergistic combination of both declarative (Description Logic) and procedural (SWRL Rules) ontology knowledge in order to empower the non-specialist data miner throughout the key phases of the CRISP-DM data mining process. The authors successfully present some novelty features their intelligent DM assistant attempts to provide by combining both declarative and procedural ontology knowledge. Furthermore, the use of the DM ontology provides a natural extension to the existing CBR (Case-Based Reasoning) for addressing the need for “deeper” knowledge to empower the data miner.

Lee et al. (2013) discusses the principles of predictive manufacturing system as a strategy to allow the manufacturing industry to increase competitiveness through a highly

transparent and worry-free manufacturing process, as well as an analytic framework that can be implemented using a coupled model approach to unravel and measure uncertainties in certain industries.

Yu and Boyd (2016) outline a general-purpose flexible in-memory indexing technique based on multi-level key ranges, which can be easily adopted into existing systems with B+-tree, ISAM or data list of sortable keys to make the indexing smarter.

Saldivar et al. (2016) present a k-means cluster approach used to manage relevant Big Data. The identification of patterns from Big Data is achieved with a cluster method and with the selection of optimal attributes using genetic algorithms. The final outcomes of this work present that Big Data analytics (nodes) help to visualize the influence of product characteristics and to cluster customer needs and wants.

Wang et al. (2018) present a comprehensive survey of commonly used deep learning algorithms and discuss their applications toward making manufacturing "smart". Specifically, a deep learning enabled advanced analytics framework is proposed to meet the opportunistic need of smart manufacturing. Deep learning provides advanced analytics and offers great potentials to smart manufacturing in the age of Big Data. By transforming the unprecedented amount of data into actionable and insightful information, deep learning gives decision-makers new visibility into their operations, as well as real-time performance measures and costs.

Chungoora et al. (2013) propose a hybrid approach that combines federated and multi database techniques, which provide the most feasible avenue for large scale integration. Under the proposed architecture, the individual data site administrators provide an augmented export schema specifying knowledge about the sources of data, their structure, their content and their relationships. This knowledge is used to generate a partially integrated, global view of the data.

Azvine et al. (2006) firstly discuss the issues and problems of current business intelligence systems, and then outline their vision of real-time business intelligence. In addition, they present a list of emerging technologies that are being developed within the research program of British Telecommunications (BT) plc., which could contribute to the realisation of real-time business intelligence.

Ben et al. (2005) discuss issues and problems of current business intelligence systems, and then outline their vision of future real-time business intelligence. Moreover, they present a list of emerging technologies which could contribute to the realization of real-time business intelligence and some examples of applying them to improve BT's systems and services. Azvine et al.(2005) present the future RTBI infrastructure will include the following elements: (i) static data warehouses and dynamically user- configurable data shopping malls, (ii) meta-data information for the whole enterprise, (iii) taxonomies and ontologies for describing contents and providing semantic content information, (iv) information about the context of data sources, (v) advanced ETL tools for gathering and feeding data to analytical modules, (vi) feedback mechanisms to operational systems.

Polyvyanyy et al. (2017) propose the Process Querying Framework, which aims to guide development of process querying methods. The proposed framework is composed of generic components that can be configured to create a range of process querying methods. The motivation for the framework stems from use-cases in the field of Business Process Management.

Sahay and Ranjan (2008) examine the need for real-time business intelligence (BI) in supply chain analytics. The authors focus on the necessity to revisit the traditional BI concept that integrates and consolidates information in an organization to support firms that are service oriented and seeking customer loyalty and retention.

Tang et al. (2018) describe the vision of smart shop-floor based on the notion of Industry 4.0 that denotes technologies and concepts related to Cyber-Physical Production Systems (CPPS). The experimental results prove that intelligent manufacturing paradigms aligning with smart shop-floor enable agile reaction to disturbances and maintenance of high production performance.

Denno et al. (2018) present a methodology, called production system identification, to develop a model for a manufacturing system from system operation logs. The model produced is intended to aid in making production scheduling decisions. The proposed methodology is evaluated on an automotive assembly system concluding that it is possible to use log content to produce a model useful to production control tasks, such as line balancing and job sequencing.

Zhong et al. (2015) propose a holistic Big Data approach to excavate frequent trajectory from massive RFID-enabled shop-floor logistics data with several innovations highlighted in order to deal with existing methods which are not suitable for removing noises due to the highly complex and specific characteristics of RFID Big Data.

Borkar et al. (2012) propose the use of recursive queries to program a variety of machine learning algorithms instead of creating a new system for each specific flavor of machine learning task, or hard-coding new optimizations. By utilizing this approach, database query optimization techniques can be used to identify effective execution plans, which can be executed on a single unified data-parallel query processing engine. The authors demonstrated that their approach can offer a plan tailored to a given target task and data for a specific machine resource allocation.

Rusitschka et al. (2010) present a cloud computing model for managing the real-time streams of smart grid data using real-time information needs. The Smart Grid Data Cloud is suitable for liberalized energy markets with a data clearing house concept, large vertically integrated utilities, as well as associations of transmission system operators, such as in the ENTSO-E.

Lu and Wen (2014) propose a minimum-cost-forwarding-based asynchronous distributed algorithm to find the optimal placement for the data aggregation service tree with optimal cost of in-network processing. The authors demonstrate that the proposed algorithm has less message overheads than the synchronous algorithm (Sync).

Smart grid data analytics play a critical role in the business and physical operations of delivering electricity and managing consumption. Even though utilities start from a difficult position as there is need to integrate data analytics into the enterprise, data science is critical to modernize the grid. Stimmel (2016) demonstrates the critical role that smart grid data analytics bring to the electricity business.

Papazoglou et al. (2015) present a production architecture which lowers the barrier for entrepreneurs to design novel products and processes and develop manufacturing software that could be plugged into the Smart Manufacturing Networks (SMN) platform for easy access by multiple users to enable collaborative manufacturing of new products and response to product demand.

Roh et al. (2019) perform a comprehensive study of data collection from a data management point of view and discuss interesting data collection challenges that remain to be addressed by the research community.

O'Leary (2014) examines the notion of the Big Data Lake and contrasts it with existing solutions (data warehouses) to discuss the risks of the Emerging Lake concept and investigate the embedding of different artificial intelligence and crowdsourcing (human intelligence) applications into that lake. The final results present that the emerging conceptual vision of the lake is able to integrate and analyze multiple data sources in a single table captured as part of in-memory computing.

The emergence of Big Data, fueled by diverse software applications, has led to the establishment of Big Data Warehouses and Data Lakes as fundamental components for organizational decision-making. However, the limitations of these monolithic architectures have highlighted the necessity for a paradigm shift towards data-oriented organizations. Enter Data Mesh, a novel architectural concept that prioritizes data as the central organizational concern. In a Data Mesh architecture, data is intentionally distributed across multiple nodes, mitigating chaos and data silos through centralized governance strategies and shared core principles. The work in (Machado et al., 2022) elucidates the motivation behind the Data Mesh paradigm, its key features, and approaches for its practical implementation. Furthermore, Dehghani 2019 discuss the prevalent trend of enterprises in-vesting in next generation Data Lakes to democratize data access and drive business insights through automated decision-making. However, traditional Data Lake architectures often encounter failure modes that hinder scalability and fail to deliver on their promises. To overcome these challenges, the paradigm needs to shift away from the centralized model of a Data Lake or data warehouse towards a distributed architecture. This paradigm shift involves prioritizing domains as the primary concern, implementing platform thinking to establish self-serve data infrastructure, and treating data as a product. The reference to Dehghani's article 2019 highlights the importance of transitioning from a monolithic Data Lake to a distributed Data Mesh to address these issues effectively.

Coined by Zhamak Dehghani in 2019, Data Mesh is underpinned by four key principles: domain-oriented decentralized data ownership, self-service data infrastructure as a platform, data as a product, and federated computational governance. These principles

encourage domain teams to own and manage their data, treat data as a product for internal and external users, and maintain high-quality data governance. Wrembel (2023) investigated operational mechanics, benefits, architectural elements, limitations, and prospects of Data Meshes, offering a comprehensive guide for organizations seeking to enhance their data management strategies

Panigrahy et al. (2023) explored the concept of transitioning from "data mess" to "data mesh," a decentralized data architecture framework that addresses scalability and governance issues within organizations. This approach organized data by business domains, such as marketing, sales, and customer support, allowing domain-specific data producers to assume ownership and establish governance policies tailored to their expertise. As the paper presents, by decentralizing data management, data mesh promotes self-service data usage across the organization, avoiding inefficiencies of centralized systems while still utilizing traditional storage solutions like data lakes and warehouses.

Transforming Data Lake into Data Meshes leads to a significant development concerning how organizations manage their data. By decentralizing data ownership and management, Data Mesh architecture empowers domain-oriented teams to take ownership of their data domains, creating a culture collaboration in terms of data. This not only promotes agility and scalability but also equalizes access to data, enabling teams to make data-driven decisions independently. However, this transition requires significant organizational change, including restructuring teams and redefining roles and responsibilities Machado et al. (2021). It also introduces technical challenges, such as managing distributed systems and ensuring interoperability across different domains. Despite these difficulties, the potential benefits of improved agility, scalability and decentralization make the transformation to a Data Mesh architecture an important and vital transformation in the Big Data era.

The combination of Data Lakes, Data Meshes, and Blockchain-based technologies specifically, NFTs in the field of modern data management creates a dynamic synergy that is changing how businesses handle data ownership, accessibility and storage. Data Lakes function as large storage spaces for heterogeneous data, promoting a single repository that can handle a variety of data types. In addition, the Data Mesh paradigm supports distributed data processing and domain-oriented ownership using decentralized

data architectures. A new dimension is brought to data ownership and authenticity by the integration of NFTs on Blockchain platforms, which offer a safe and verifiable framework for identifying the provenance and ownership of individual pieces of data. The integration of Blockchain, Data Lakes and Data Meshes improves the scalability and flexibility of data ecosystems and lays the groundwork for more open, safe, and cooperative data management procedures as related work unveils.

A visionary approach to establish a distributed federated medical Data Lake and ecosystem was proposed by Phuc et al., (2023), involving hospitals and personal health data from wearable medical devices. It emphasized the creation of a Blockchain-based platform with commercial incentives, addressing data ownership, patient privacy, and controlled access. The platform facilitated owner-centric medical data exchange, securely aggregated data from various hospitals, and unlocked academic and business value by rep-resenting medical data as NFTs. The primary goal was to improve healthcare research while fostering a sustainable medical data ecosystem.

Finally, in order to manage data at scale, Dolhopolov et al. (2023) investigated how a Blockchain-powered metadata catalogue might be integrated into a Data Mesh architecture. The metadata catalogue improved governance, efficiency, access, and discovery. The catalogue managed metadata across a dispersed network of data domains with federated governance, immutability, and transparency thanks to the use of Blockchain technology. A proof-of-concept solution utilizing HyperLedger Fabric was presented, with advantages including increased reliability, efficiency, and transparency being high-lighted. It also discussed and suggested possible solutions for issues including governance, scalability, and interoperability.

The methodology used to search and locate these papers followed systematic survey principles, ensuring a comprehensive and objective review of the latest advancements in critical research pillars. This survey organizes the key findings and summarizes the most significant recent advances across eight core research areas. Each category highlights the challenging, unresolved issues identified in the reviewed literature, providing a structured view of the landscape and pinpointing areas that require further investigation. These open research problems will serve as a foundation to inform the specific research directions of this PhD thesis. The eight categories are described as follows:

I. **Advanced Optimization, Decision Support, and Predictive Modeling:** This category underscores the need for advanced methods and algorithms to streamline optimization, enhance decision-making, and improve predictive capabilities in complex environments. Key challenges include developing robust algorithms for dynamic systems, advancing predictive analytics for real-time insights, and creating automated decision-support systems. These solutions are essential for managing complex, data-rich environments where optimizing resource allocation and predicting outcomes can significantly impact operational efficiency and strategic planning.

II. **Innovative Applications of Artificial and Computational Intelligence:** Artificial intelligence and computational intelligence (AI/CI) offer transformative potential in solving complex, data-intensive problems, and process mining has become a critical tool for uncovering inefficiencies and bottlenecks within business processes. This category explores challenges such as scalable machine learning models, process mining methodologies, advanced pattern recognition, and real-time data analysis. Data lake and data mesh architectures also play a crucial role, providing structured and flexible environments for large-scale, diverse datasets used in training AI models. The development of intelligent, adaptable systems that leverage AI/CI is crucial for handling vast data volumes, automating decision processes, and supporting a variety of industrial and real-world applications where data-driven insights drive value creation.

III. **Scalability and Integration in Experimental Research and Prototyping:** To validate emerging models and approaches, experimentation must be robust, scalable, and integrative. This category addresses the need for empirical testing to assess model generalizability, scalability, and integration with real-world systems. Challenges include transitioning from prototype to stable systems, conducting large-scale benchmarking, and refining experimental designs to capture complex, multi-platform interactions. Enhanced experimentation methods ensure that innovative solutions meet the high demands of real-world applicability and reliability.

IV. **IoT Infrastructure and Application Innovations for Intelligent Environments:** As the Internet of Things (IoT) continues to grow, challenges related to infrastructure scalability, data security, and application reliability

become more pronounced. This category examines issues around IoT-enabled environments, such as secure data transmission, resource optimization, and real-time data processing. Effective IoT systems are pivotal for data-driven industries and applications that require seamless, continuous connectivity and intelligent automation across devices and platforms.

V. **Data Modeling, Integration, and Semantic Interoperability:** Data modeling is fundamental to understanding and utilizing complex data structures in today's interconnected systems. This category focuses on challenges such as data quality, semantic enrichment, and interoperability across different domains. Data lakes and data meshes are increasingly employed to enable scalable, flexible data structures that support cross-domain and cross-platform analytics, offering a more cohesive approach to data integration. Developing comprehensive data models that can handle high variability and support scalable, cross-platform analytics is essential to drive meaningful insights and enable cohesive data integration across industries.

VI. **Challenges and Opportunities in Cloud Environments:** Cloud technology offers immense potential for scalable, flexible computing resources, but managing cloud environments comes with its own set of challenges. This category highlights issues such as effective cloud monitoring, data security, and multi-tenant management. Research in this area is directed towards developing tools that enhance cloud performance, optimize resource allocation, and address the complexities of federated and hybrid cloud structures to provide reliable, on-demand services.

VII. **Secure, Distributed Smart Data Processing:** Smart data processing is integral to making sense of large, diverse data sets in real time. This category outlines challenges related to data analytics, real-time processing, and the development of intelligent, self-adapting systems. Blockchain technologies are becoming valuable for ensuring data integrity, privacy, and security, especially in applications where transparent, verifiable data management is essential. Data lakes and data meshes are also relevant here, facilitating the management of complex data ecosystems that support efficient and decentralized data access. Efficient smart data processing enables actionable insights and automated responses in dynamic

environments, where rapid data-driven decision-making is crucial, such as in smart manufacturing and autonomous systems.

VIII. **Global Standards for Interoperability and Quality Assurance:** Standardization ensures interoperability, quality, and consistency across diverse systems and applications. This category discusses the necessity of developing unified standards in data handling, modeling, and system integration. The goal is to facilitate seamless data exchange, reduce system complexities, and support global collaboration. Standardization in these areas will help pave the way for widespread adoption and interoperability of advanced technologies, especially in complex industrial and multi-stakeholder environments.

The above categories encapsulate the primary open challenges identified in the literature and represent strategic areas for continued research. This PhD thesis will target selected issues from these categories to contribute meaningful insights and solutions that advance knowledge and applications within these critical domains.

CHAPTER 3 : THEORETICAL AND TECHNICAL BACKGROUND

Smart Data Processing (SDP) and Smart Data Integration (SDI) are foundational for developing intelligent systems that interact seamlessly with human users, smart devices, extensive networks, and diverse environments - both natural and artificial. An intelligent system's ability to provide and utilize cognitive support within its environment is a necessary condition for its intelligence. This "smartness" is built upon various functionalities such as sensing, data processing, knowledge-based data communication, aggregation, interoperability, decision-making, and human-like perception and behavior. Together, these components enable systems to not only operate effectively but also to adapt and respond to changing contexts.

The term Smart Data highlights the latent value in widely dispersed, unconnected data sources. Its primary goal is to extract meaning and provide decision-makers with high-value information that enables optimal choices and addresses complex problems. The essential properties of Smart Data include:

(i) Normalized Data: This refers to conflict-free, homogenized data sourced from multiple representations, which is interpretable within a specific context. By normalizing data, organizations can ensure consistency and reliability in their analyses.

(ii) Contextualized Data: Building on normalization, contextualized data adds a layer of contextual awareness that improves decision-making (Khine & Wang, 2018). This layer helps organizations understand not just the data itself but also the conditions under which it was generated, leading to more informed decisions.

(iii) Orchestrated Data: This involves cross-correlating secure data within a specific domain - such as healthcare or smart cities—to transform it into actionable insights, often assisted by artificial intelligence (AI). Orchestrated data enables organizations to leverage diverse data sources to generate comprehensive insights.

A core contribution of this research is the innovative use of Data Lakes and Data Meshes to achieve the Big Data processing levels required for effective Smart Data Processing.

Data Lakes, which were proposed in 2010 as architectures suitable for managing Big Data, assist organizations in adopting data-driven approaches (Wieder & Nolte, 2022). They enable the storage of structured, semi-structured, and unstructured data at any scale, facilitating the selection and organization of data within a centralized repository. Significantly, data (both relational and non-relational) can be stored directly in a Data Lake in its current form, negating the need for conversion into a structured format. Furthermore, organizations can perform a wide range of analytical methods such as dashboards, visualizations, real-time analytics, and machine learning without needing to transition to another system for decision-making support through predictive and prescriptive analytics (Lepenioti et al., 2020). Overall, Data Lakes provide a cost-effective, flexible, and scalable way to host data in its raw format, empowering organizations to store large volumes of data without the necessity of conforming to a specific schema in advance. However, despite these advantages, Data Lakes present several adoption hurdles due to their relatively recent emergence and revolutionary concepts.

Khine and Wang 2018, outline the main challenges associated with Data Lakes: (i) the need for decentralization (both physically and virtually); (ii) the necessity for data product discovery and a domain-driven approach; (iii) the absorption of all types of data without proper monitoring or governance; and (iv) the lack of a descriptive metadata mechanism, which can lead to the creation of a data swamp. In Sawadogo and Darmont 2021, a comprehensive state-of-the-art overview of different approaches to Data Lake design highlights architectures and metadata management as key issues in successfully leveraging Data Lakes.

Enhancements to Data Lakes provide additional features and benefits, notably through the concept of data ponds, which constitutes a fraction of a Data Lake that is typically smaller in scale and used for specific purposes, such as testing or implementing dedicated functionality . A data pond is specifically designed and optimized for particular applications or use cases. Conversely, Data puddles refer to smaller, pre-built datasets created for particular purposes (e.g., offering information on a subset of the data), which further enhance data accessibility and usability.

The relevant literature indicates a growing trend toward systems for decentralized data interchange, such as Data Meshes (Van den Heuvel et al., 2023). The term Data Mesh

was first defined by Zhamak Dehghani in 2019, who later elaborated on its principles and logical architecture. A Data Mesh represents a next-generation data architecture that adopts a decentralized approach to data management by treating data as a product. This model defines ownership and accountability for data products while emphasizing data governance, quality, and operational excellence. Within a Data Mesh framework, data is managed as a product, with clear product owners, enabling data consumers to have self-service access to the data they require. This paradigm shift offers a more flexible and scalable solution compared to traditional Data Lakes, fostering multiple sources of truth, promoting data agility, and encouraging data literacy across the organization.

A Data Mesh is predicated on four core principles (Dehghani, 2020):

i. Decentralized Data Ownership: Each team or microservice within an organization should own and be responsible for the data they produce, thereby fostering accountability.
ii. Product-Centric Data: Data should be treated as a product rather than merely as a by-product of software development, emphasizing its strategic value.
iii. Automated Data Governance: This principle advocates automation to enforce data quality, security, and privacy policies, ensuring compliance and enhancing reliability.
iv. Shared Data Services: Involves the creation of shared data services and APIs, enabling teams to access and use data consistently and reliably, thus fostering collaboration and efficiency.

Furthermore, a Data Mesh can be viewed as a data architectural pattern that provides a means to manage and share data between microservices in a scalable and decentralized manner. This approach aims to create a “mesh” of data services that work in unison to deliver consistent and reliable data access to all services in need. By doing so, Data Meshes reduce the complexity and dependency on central databases, making it easier to manage data at scale within a microservice-based architecture. Additionally, Data Meshes address some of the shortcomings associated with monolithic data platforms, such as Data Lakes, by creating data products and domains. A data product is a tangible and valuable output of data that serves a specific business need, created and managed within a Data Mesh architecture. Data products are developed with a product mindset, meaning they

have clear ownership, defined users, and a lifecycle that includes continuous improvement.

In summary, Data Lakes and Data Meshes represent two distinct paradigms in data architecture, differing fundamentally in their conceptual and operational approaches. Data Lakes rely on a centralized repository model, where data is managed by a central team, primarily focusing on storage without necessarily treating data as a product (Vlasiuk & Onyshchenko, 2023). In contrast, Data Meshes, as an organization approach not a storage architecture, embrace a decentralized approach, distributing data ownership across different business domains and promoting the idea of "data as a product" with clearly defined responsibilities (Driessen et al., 2023). While Data Lakes scale horizontally by adding storage capacity, Data Meshes achieve scalability and flexibility by breaking down complex systems into smaller, manageable domains. Governance in Data Lakes is centrally controlled, whereas Data Meshes empower local autonomy within domains for governance and discoverability. Furthermore, Data Lakes expose data through a centralized repository, while Data Meshes allow access to individual domains based on user requests. These differences reflect a shift from monolithic data architectures toward more federated, domain-driven (Dehghani, 2020).

In this landscape, Blockchain technology serves a critical purpose by providing secure and transparent means for recording and transferring data. Notably, it addresses privacy concerns by anonymizing personal data, which contributes to its increasing popularity and integration into infrastructure, opening avenues for innovative applications (Alam, 2022). Functioning as a decentralized database on a peer-to-peer network, Blockchain establishes a distributed communication network that enables non-trusting nodes to interact without relying on a central authority. Its protocols ensure a verifiable and trustworthy system, offering traceability, transparency, and enhanced privacy and security features. In essence, Blockchain is evolving into a fundamental technology with wide-ranging applications, including IoT, Smart Contracts, NFTs, Cybersecurity, and Cryptocurrency, providing a robust foundation for secure and trustworthy data transactions (Di Angelo & Salzer, 2020).

Algorithmically, Blockchain encompasses a variety of essential elements, procedures, and guidelines that create a strong and feature-rich decentralized system. Initializing basic components, such as a consensus mechanism and cryptographic algorithms for secure key

management and hashing, represents the first steps in the process. Implementing token and smart contract standards like ERC-20 and ERC-721 enhances functionality by managing the creation, transfer, and ownership of assets (Yildiz et al., 2023). Furthermore, techniques such as zero-knowledge proofs smoothly incorporate Decentralized Identity Standards (DIDs) to guarantee secure identification and privacy standards, thus ensuring robust user data security. Interledger Protocol and other interoperability standards facilitate cross-chain communication (Rehman et al., 2021). The integration of decentralized storage protocols, such as IPFS, ensures that file storage is dispersed and resistant to censorship. Governance norms support secure and efficient decision-making, while security measures protect against vulnerabilities and compliance standards ensure adherence to legal requirements. This comprehensive strategy creates a conceptual framework for building a Blockchain that integrates fundamental criteria, promoting a safe, compatible, and considerate decentralized ecosystem.

Specifically, Non-Fungible Tokens (NFTs) are a ground-breaking innovation in the ownership and management of digital assets. Because every NFT is distinct and has a unique identifier, it cannot be copied or traded. Blockchain technology is used to accomplish this uniqueness. NFTs are used to verify ownership of a wide range of digital and physical goods (Rehman et al., 2021), including digital art, music videos, real estate, gaming avatars etc. NFTs are also crucial to Web 3.0, the next iteration of the Internet that many companies and analysts are pushing. Blockchain's decentralized structure guarantees the integrity and transparency of ownership data, and smart contracts streamline transactions by automating tasks like ownership transfers and royalty distribution.

Finally, the NFT process algorithm starts with the digital asset being initialized, having its nature defined, and being given a unique identification. The implementation of a smart contract that oversees the NFT requires integration with a Blockchain platform, such as Ethereum, via the ERC-721 standard (Phuc et al., 2023). The NFT is created during the minting process by adding ownership information and other pertinent metadata to the smart contract. Smart contract updates enable ownership transfers guaranteeing safe and transparent transactions documented on Blockchain. The NFT ecosystem is made more efficient by automating features in the smart contract, such as the distribution of royalties upon resale. NFTs are posted on NFT marketplaces such as OpenSea or Rarible, where

buyers and sellers can transact to make them more widely available (Rehman et al., 2021). Verifying the integrity of related metadata and examining ownership records on the Blockchain are two steps in the process of authenticating NFTs. The foundation of the NFT lifecycle is the aforementioned algorithmic procedure, which provides a methodical way to create, transfer, and confirm ownership of distinct digital assets on the Blockchain.

CHAPTER 4 : EXPLOITING METADATA SEMANTICS IN DATA LAKES USING BLUEPRINTS

4.1 Introduction

One of the greatest challenges in Smart Big Data Processing nowadays as aforementioned in the previous chapter revolves around handling multiple heterogeneous data sources that produce massive amounts of structured, semi-structured and unstructured data through Data Lakes. The latter requires a disciplined approach to collect, store and retrieve/ analyses data to enable efficient predictive and prescriptive modelling, as well as the development of other advanced analytics applications on top of it. The present chapter addresses this highly complex problem and proposes a novel standardization framework that combines mainly the 5Vs Big Data characteristics, blueprint ontologies and Data Lakes with ponds architecture, to offer a metadata semantic enrichment mechanism that enables fast storing to and efficient retrieval from a Data Lake.

The basic principles of manufacturing blueprints were adopted and their purpose and meaning were modified to reflect the description and characterization of data sources and the data they produce. Along these lines, a framework is proposed that builds upon the utilization of the five Big Data characteristics (5Vs) to describe Big Data sources. These characteristics guide the description of data sources by means of specific types of blueprints through an ontology-based approach. Big Data sources are accompanied by this blueprint description before they become part of a Data Lake. The proposed mechanism is compared qualitatively against existing metadata systems using a set of functional characteristics or properties, with the results indicating that it is indeed a promising approach.

The main research contributions of this chapter include the utilization of Data Lakes as a means to achieve the desired level of Big Data processing and ultimately lead to

developing Smart Big Data Processing. Along these lines, a standardization framework is proposed for storing data (and data sources) in a Data Lake and a metadata semantic enrichment mechanism is introduced that is able to handle effectively and efficiently Big Data coming from disparate and heterogeneous data sources. These sources produce different types of data at various frequencies and the mechanism is applied both when this data is ingested in a Data Lake and at the stage of extracting knowledge and information from the Data Lake.

4.2 A Semantic Enrichment Metadata Mechanism via Blueprints

As mentioned above, a new standardization framework is introduced in this first methodology chapter combining the 5Vs Big Data characteristics and blueprint ontologies to assist data processing (storing and retrieval) in Data Lakes organized with pond architecture. According to the pond architecture, a Data Lake consists of a set of data ponds and each pond hosts/refers to a specific data type. Each pond contains a specialized storage system and data processing depending on the data type (Sawadogo & Darmont, 2021).

A Data Lake with pond architecture is assumed to use a dedicated pond to store each data source with the same type of data, structured, unstructured, and semi-structured, as shown in Figure 2 and the following pseudocode.

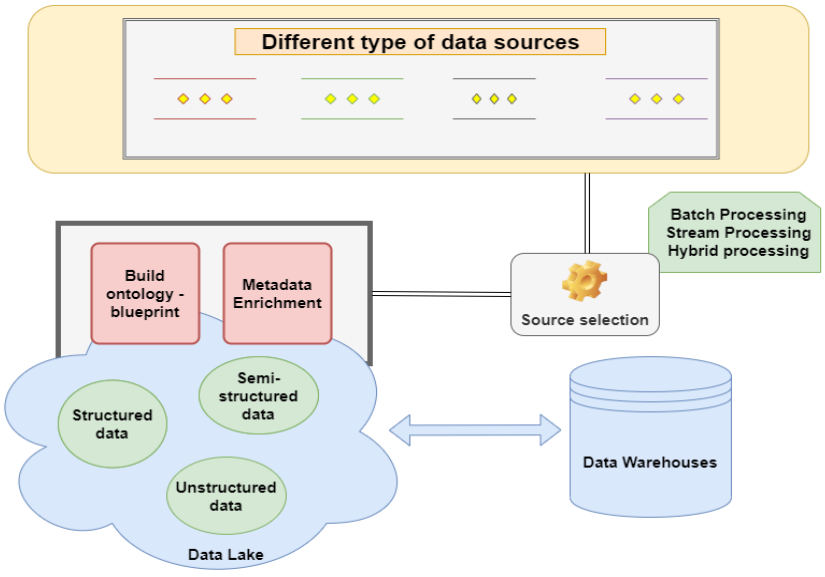

Figure 2. Data sources selection metadata enrichment mechanism using 5Vs

```
# Iterate over all candidate data sources
for source in candidate_sources:
    # Extract 5Vs metadata etc (volume, velocity, variety, veracity, value, keywords…)
    metadata = extract_5Vs(source)
    # Create RDF representation (semantic blueprint) from metadata
    rdf = create_rdf(metadata)

    # Check if RDF metadata matches user-defined criteria (e.g., value=High)
    if matches_query(rdf, user_criteria):
        # Insert RDF metadata into the triplestore (semantic metadata repository)
        triplestore.insert(rdf)
        # Classify source by structure to pond (structured, unstructured, etc.)
        pond = classify_by_type(source)
        # Add source to the appropriate pond in the Data Lake
        data_lake[pond].append(source)

# Function to retrieve relevant data based on SPARQL query criteria
def retrieve(criteria):
    # Execute SPARQL query on metadata to find matching sources
    matches = sparql_query (triplestore, criteria)
    # Retrieve data sources only from matching sources (efficient retrieval)
    return [load_data(s) for s in matches]
```

This innate pond architecture is particularly helpful when extracting information from the Data Lake as will be demonstrated in the next section. Big Data sources are filtered before they become part of the Data Lake as shown in Figure 2. Every data source, which is a candidate to be part of the Data Lake, will be characterized according to the blueprint values shown in Figure 3. Therefore, the selection of data sources is performed according to the blueprint of each different data source.

As previously mentioned, a dedicated blueprint is developed to describe each data source storing data in the Data Lake. Specifically, the blueprint of a source consists of two interconnected blueprints as shown in Figure 3.

The first one is static and records the name and type of the source, the type of data it produces, as well as the value, velocity, variety, and veracity of the data source. The second is dynamic as it changes values in the course of time and essentially characterizes the volume of data, the last source update, and the keywords of the source. The dynamic blueprint is updated every time data sources produce new data. Figure 4 presents the ontology graph of the data source created via Protégé, a free open-source ontology editor and framework for building knowledge-based solutions (http://protege.stanford.edu). This graphical representation tool produces an RDF ontology for the data sources.

RDF stands for Resource Description Framework and is a framework for describing resources usually found on the Web. RDF is designed to be read and understood by computers, is not designed for being displayed to people, and is written in XML. RDF is part of the W3C's Semantic Web Activity and is a standard for data interchange that is used for representing highly interconnected data. Each RDF statement is a three-part

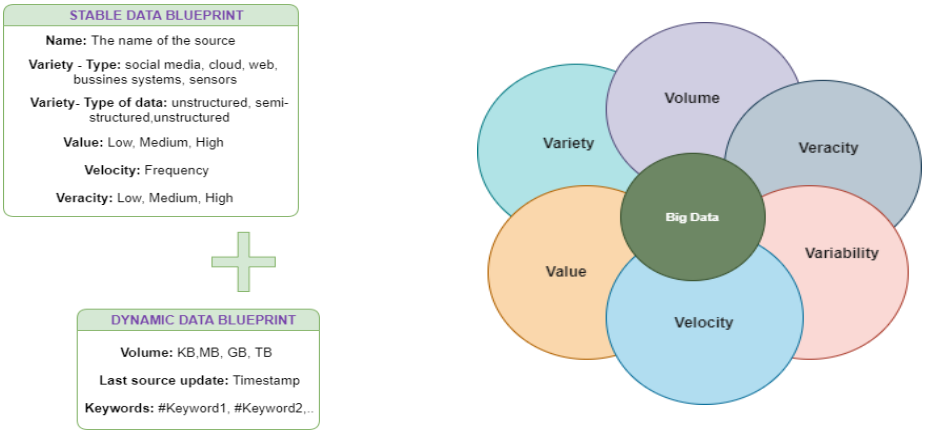

Figure 3. Data source blueprints description using 5Vs Big Data characteristics

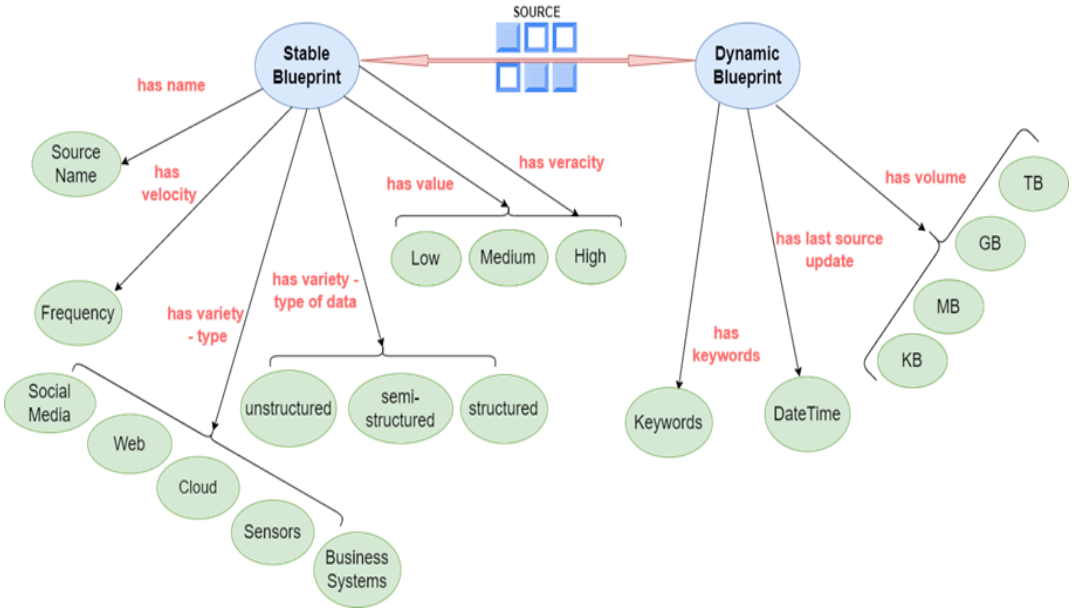

Figure 4. Stable and Dynamic data source blueprint ontology graph

structure consisting of resources where every resource is identified by a URI. Representing data in RDF allows information to be easily identified, disambiguated and interconnected by AI systems. RDF ultilzed to describe a source's stable and dynamic blueprint with the combination of the theory of triples (subject, predicate, object) (see Figure 5).

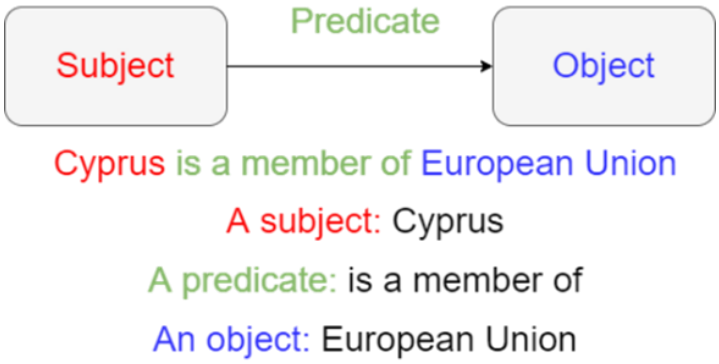

Figure 5. The basic semantic RDF triple model

For example, let us assume that we wish to store values in a Data Lake produced by four candidate sources and that these sources include the following characteristics – attributes in Table 1 according to the data source blueprints (see Figure 3):

Table 1. Attributes of Four Candidate Sources According to Data Source Blueprints

Attribute	**Source 1**	**Source 2**	**Source 3**	**Source 4**	**Type**
Name	Source 1	Source 2	Source 3	Source 4	Stable
Variety-Type	Sensor	Business Systems	Web	API	Stable
Variety-Type of data	Unstructured	Structured	Unstructured	Structured	Stable
Value	High	High	High	Medium	Stable
Velocity	1sec	1sec	12Hours	Real-Time	Stable
Veracity	Medium	Medium	High	Medium	Stable
Volume	KB	KB	KB	MB	Dynamic
Last Update	24/01/2024 15:22	24/09/2024 08:34	24/07/2024 06:50	20/08/2024 15:30	Dynamic
Keywords	# Products delivery	# Products delivery	# Production	# Production	Dynamic

By using SPARQL, or other methods, we may query all RDF resources before being ingested into the Data Lake or after their ingestion. This requires that each data source has its description in RDF form as mentioned before, which can be retrieved by a public API (RDF API) for the sources to be queried. Figure 6 shows the RDF files written in XML for Source 1 of the given example based on the blueprint ontology description created. The XML follows the same structure for every source according to its characteristics.

Let us now assume that all three sources described above will become members of our Data Lake. Therefore, we must first build a specific part of the Data Lake that consists of data sources with Value - High, Veracity – Medium OR High, AND Keywords - # Products delivery. A source selection middleware is fed with these preferred rules (see Figure 1) and executes the following SPARQL query:

SELECT ? sources

WHERE {

? source <has value> High &&

<has veracity> Medium &&

<has keyword> #Products delivery

}

Stable Blueprint

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.blueprints.com/">

<rdf:Description
rdf:about="https://www.blueprints.com/stable">
  <cd:name>Source 1</cd:name>
  <cd:varietytype>Sensor</cd:varietytype>
  <cd:varietypeof>Unstructured</cd:varietypeof>
  <cd:value>High</cd:value>
  <cd:velocity>1s</cd:velocity>
  <cd:veracity>Medium</cd:veracity>
</rdf:Description>

</rdf:RDF>
```

Dynamic Blueprint

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.blueprints.com/">

<rdf:Description
rdf:about="https://www.blueprints.com/dynamic">
  <cd:volume>KB</cd:volume>
  <cd:lastupdate>24/01/2022 08:34</cd:lastupdate>
  <cd:keywords>
    <li>#Products delivery</li>
  </cd:keywords>
</rdf:Description>

</rdf:RDF>
```

Figure 6. Stable and Dynamic Blueprint for Source 1

Once this selection process is completed, the RDF created earlier becomes now part of the Data Lake and contributes to the Data Lake's metadata semantic enrichment which is the cornerstone for addressing the challenge of easy storing and efficient retrieval of data. The result of the query consists of the stable and dynamic blueprints of Source 1 and Source 2 that satisfy the query parameters, and, thus, will be added to the Data Lake's RDF schema.

The selected data sources are then distributed to the specific Data Lake Pond for further processing according to the corresponding attribute values. Essentially, this process and the associated characterization help to handle and manage multiple and diverse types of

data sources, and to contribute to the Data Lake's metadata enrichment before and after these sources become members.

When a data source becomes part of the Data Lake, from that point forward the metadata schema is utilized for filtering and retrieving data based on the blueprints. The latter involves attributes such as the type of data produced by the sources, the size of the data they produce, the speed of production, the accuracy of the data, the importance of the source data, etc. Therefore, each action for retrieving data from the Data Lake is effectively guided by the information provided in the metadata mechanism, that is, in the blueprints. Especially in the case of the dynamic blueprint, this portion of the metadata will dynamically be updated each time new data is produced by the sources, or when deemed necessary (e.g., when changing the location of the associated pond).

To sum up, metadata-driven SPARQL could support the efficient ingestion of new data sources into the Data Lake. When a new source is introduced, its metadata such as its volume, variety, veracity, and contextual tags is first captured and mapped into a Semantic Data Blueprint (SDB). These attributes are encoded as RDF triples, and a SPARQL INSERT DATA query is generated to store the metadata within the triple store backing the Data Lake as an example the previous query. This ensures the new source is immediately discoverable and categorizable within the semantic framework. By aligning ingestion with semantic blueprints, the system supports consistent, metadata-enriched storage, improving governance, discoverability, and integration with downstream querying and mesh transformation processes.

To further demonstrate the applicability and the value that the proposed metadata mechanism brings to supporting the data actions in a Data Lake, and particularly the retrieval of data, let us use once again the example of the sources given earlier. As previously mentioned, the selected data sources are distributed to the specific Data Lake Pond for further processing according to the corresponding attribute values. After the completion of the selection process, the retrieval process is based on the metadata semantic enrichment – RDF schema of the Data Lake encoded in the blueprints. Let us now assume that Source 1 is a member of the pond hosting the structured data and that Source 2 is member of the pond with the unstructured data. If we wish to retrieve all the product delivery data from our Data Lake we may utilize a middleware residing between the Data Lake and the application layer of a system that uses the Data Lake. In the simplest case all that needs to be done is to execute the following SPARQL query:

```
SELECT ? Dlsources
    WHERE {
      ? source  <has keyword>   #Products delivery
        }
```

Essentially, this performs a classic retrieval action from the Data Lake, pushing all data sources with the "Product Delivery" keyword to the application layer. However, this approach results in large volumes of data being retrieved, and filtering the relevant information afterward becomes more complex and time-consuming.

By utilizing the metadata schema through semantic enrichment, the process becomes significantly faster and more efficient. Querying the metadata is, of course, much quicker than querying the actual data sources, as the metadata provides a lightweight structure that can be easily filtered and searched. This enables more precise queries that can target specific values or attributes, greatly reducing the amount of data retrieved. By refining the query at the metadata level, only the most relevant data is brought to the application layer, streamlining the retrieval process and minimizing the need for additional filtering. Therefore, the metadata schema not only improves the speed of retrieval but also enhances the overall efficiency of data handling in the Data Lake.

Specifically, if we utilize the metadata information offered by the semantic enrichment mechanism, then we can refine the type of information sought in the Data Lake and get the results we need focusing on specific values or attributes. For example, using the attributes shown in Figure 5 it is feasible to execute more guided queries such as:

```
SELECT ? DLsources
    WHERE {
        ? source <has value> High &&
        <has Variety-Type of data> Unstructured &&
         <has keyword> #Product delivery }
```

These guided queries can range from simple to more sophisticated by utilizing the full spectrum of the 5Vs data characteristics mentioned in Figure 3. Thus, they allow data scientists to derive more value from their data and to define custom levels of granularity and refined information in the data sought as required. Essentially, this SPARQL query

process and the associated characterization support the handling and management of multiple and diverse types of data sources residing in a Data Lake in a simple, yet efficient way.

To clarify further, the metadata-driven SPARQL construction process, consider the following illustrative example: a Data Lake user aims to retrieve sources characterized by the aforementioned user criteria. Each data source is represented using a Semantic Data Blueprint (SDB) in RDF format, encoding these metadata attributes. Based on the user's criteria, the system constructs the previous SPARQL query. This query is executed against the metadata repository, returning URIs of matching data sources. These results inform downstream processes such as domain-specific Data Product formation or pond allocation within the Data Lake. This step-by-step flow illustrates how semantic metadata directly informs query formulation, ensuring accurate and efficient source retrieval aligned with the framework's design principles.

4.3 Preliminary Validation

Sawadogo et al. (2019) identified six main functional characteristics that should ideally be provided by a Data Lake metadata system:

- Semantic Enrichment (SE)
- Data Indexing (DI)
- Link generation and conservation (LG)
- Data Polymorphism (DP)
- Data Versioning (DV)
- Usage Tracking (UT)

The proposed approach will now be evaluated using the above characteristics:

Semantic Enrichment consists in generating a description of the context of data (e.g., tags) using knowledge bases such as ontologies. It summarizes the datasets contained in the lake to make it understandable and to identify data links. For instance, data associated with the same tags can be considered linked. Our mechanism meets this characteristic since it utilizes both the dynamic and the stable blueprint based on the basic RDF triple model presented in figures 4 and 5.

The second main functionality identified by Sawadogo et al. (2019) is Data Indexing which includes setting up a data structure to retrieve datasets based on specific keywords

or patterns. This functionality provides optimization of querying the Data Lake through keywords filtering. This characteristic is offered in our metadata semantic enrichment mechanism via the attribute Variety - type of data, which is used to distribute data sources and data ponds according to their structure and the keyword attribute in the stable blueprint as presented in Figure 3.

Link generation and conservation is the process of detecting similarity relationships or integrating pre-existing links between datasets to identify data clusters, data groups where data are strongly linked to each other and significantly different from other data. Our mechanism provides this functionality via the keyword attributes in the dynamic blueprint which is updated every time a new data source or new data that is produced by a registered source are pushed to the Data Lake.

Data Polymorphism is defined as storing multiple representations of the same data and Data Versioning refers to the ability of the metadata system to support data changes during the processing in the Data Lake. These functional characteristics are provided by our metadata semantic enrichment mechanism via the process of storing the metadata description - blueprint every time sources in the Data Lake change or produce new data. When new data is pushed into the Data Lake a new timestamped representation of this data is created and stored in the Data Lake along with the existing representations-blueprints. During the data processing in the Data Lake, the proposed mechanism updates the Dynamic Blueprint, especially the keywords if deemed necessary.

Finally, the mechanism provides also the last referred functionality Usage Tracking which is the process of recording the interactions between users and the Data Lake. Essentially, when data is queried, a timestamp accompanied by the user details that executed the last query are recorded in the Dynamic Blueprint.

Additionally, Sawadogo et al. (2019) provide a synthetic comparison of 15 metadata systems. We selected the two most completed systems examined in that chapter in terms of functionality, that is, CoreKG (Beheshti et al., 2018) and MEDAL (Sawadogo et al., 2019), with the aim of comparing them with our metadata data mechanism using a set of new functional characteristics introduced in this chapter. These characteristics can add value to the synthetic examination of the quality and efficiency of metadata enrichment mechanisms for Data Lakes. The new characteristics are: Granularity, Ease of storing/retrieval, Size and type of metadata, Expandability.

We define *Granularity* as the ability to refine the type of information that needs to be retrieved using for example keywords. This ability is expressed by the number of fine-grained levels the metadata mechanism supports for defining the information sought. *Ease of storing/retrieval* refers to the ability of the metadata mechanism to store or retrieve data in the Data Lake in a simple and easy way. It is assumed here that the retrieval action is efficient enough to return the desired parts of the information sought. This characteristic is reflected on the number of steps that need to be executed for the process of storing and retrieving data items to be completed. The *Size and type of metadata* refers to the volume and the kind of metadata that are produced by the mechanism and which are necessary for the efficient and accurate retrieval of data. The larger the size and/or the higher the complexity of the type of data needed the lower the performance and suitability of the metadata mechanism. Finally, we define *Expandability* as the ability to expand the metadata mechanism with further functional characteristics or other supporting techniques and approaches, such as visual querying. Obviously, the more open the mechanism for expansion the better. These characteristics are evaluated using a Likert Linguistic scale, including the values Low, Medium, and High. Table 2 provides a definition of Low, Medium and High for each characteristic introduced. Nevertheless, although these values may be characterized as subjective, the selection was made logically, utilizing relevant literature. Moreover, a comparative evaluation showed that even if these values were to change, the results would remain similar. As previously mentioned, we used the suggested characteristics to provide a short comparison between our metadata enrichment mechanism and the top two existing metadata mechanisms suggested by Sawadogo et al. (2019), that is, MEDAL and CoreKG.

Table 2. Definition of Low, Medium, and High of each characteristic

Characteristic/ Mechanism	**Low**	**Medium**	**High**
Granularity	1 level	2 levels	3 or more levels
Ease of storing/retrieval	5 or more actions	3-4 actions	2 actions maximum
Size of metadata	KB	MB	GB
Expandability	No or limited	Normal	Unlimited

MEDAL adopts a graph-based organization based on the notion of object and a typology of metadata in three categories, intra-object, inter-object, and global metadata. A hypernode represents an object containing nodes that correspond to the versions and representations of an object. MEDAL is modeled also by oriented edges which link the nodes providing transformation and update operations. Hypernodes of the mechanism can be linked in several ways, such as edges to model similarity links and hyperarcs to translate parenthood relationships and object groupings. Finally, global resources are present in the form of knowledge bases, indexes, or event logs. This concept and operation of the framework provide SE, DI, LG, DP, DV, and UT. As a result, MEDAL can be characterized with High Granularity, Medium Ease of storing/retrieval using indexes and event logs, Medium Size and type of metadata, and an undefined Expandability since no reference is made on how it can be evolved in the future.

CoreKG is an open-source complete Data and Knowledge Lake Service which offers researchers and developers a single REST API to organize, curate, index and query their data and metadata over time. At the Data Lake layer, CoreKG powers multiple relational and NoSQL database-as-a-service for developing data-driven Web applications. This enables the creation of relational and/or NoSQL datasets within the Data Lake, create, read, update, and delete entities in those datasets, and apply federated search on top of various islands of data. It also provides a built-in design to enable top database security threats (Authentication, Access Control and Data Encryption), along with Tracing and Provenance support. On top of the Data Lake layer, CoreKG curates the raw data in the Data Lake and prepares them for analytics. This layer includes functions such as extraction, summarization, enrichment, linking and classification. Another part of the mechanism is the Notion of Knowledge Lake, a centralized repository containing virtually inexhaustible amounts of both raw data and contextualized data as a result to providing the foundation for Big Data analytics by automatically curating the raw data in data islands so as to provide insights from the vastly growing amounts of local, external and open data. This open-source service provides SE, DI, LG, DP, and UT. Based on the proposed properties scheme, CoreKG can be evaluated to have High Granularity, Medium Ease of storing/retrieval using the single API, with Medium Size and type of metadata, and High Expandability due to the use of the Hadoop ecosystem.

As described above, the proposed metadata enrichment mechanism provides DI, LG, DP, DV and UT. Furthermore, our mechanism presents High Granularity, High Ease of

storing/retrieval using the stable and dynamic data source blueprint descriptions, with a Medium Size and type of metadata, and High Expandability. These values are attributed as follows:

High Granularity is achieved by the use of keywords that describe the sources and the blueprint values. This enables the user to define details at the level of the properties offered by these keywords and the type of blueprint characteristics for which values are kept. This list of features may be considered quite rich to enable the retrieval of data based on fine-grained query-like information. The High Ease of storing/retrieval is achieved by the blueprint description of the Data Lake as each time data sources are pushed to the Data Lake a variety of types of data attributes is produced, which helps the mechanism place the sources to a specific pond according to the structure of the data involved (structured, semi-structured and unstructured). This source distribution in the Data Lake facilitates simple and easy storing and retrieval of the information stored. Our framework is characterized by a low number of actions to: (1) Select and query data sources according to their stable and dynamic blueprint; (2) Push data in specific Data Lake Ponds. The Size and type of metadata produced by the mechanism has the maximum value of High due to the creation of the metadata description of the Data Lake every time new sources or data are pushed to the Data Lake and the DV characteristic that our blueprint provides by using the 5Vs Big Data characteristics. This may be considered a small overhead as it introduces a considerable number of metadata features, but their complexity is very low and their interpretation according to the 5Vs quite straightforward. Finally, our Data Lake implementation is based on the Hadoop ecosystem and hence this provides High Expandability. It should be noted here that expandability of the proposed mechanism can be traced in two aspects: (i) By using this simple semantic enrichment and blueprint ontologies we can easily apply visual querying during the source selection or data extraction; (ii) We may improve Data Lakes' privacy, security, and data governance, and, therefore, address some of the main challenges met with Data Lakes, by storing the descriptive metadata information to the Blockchain. This enables storing of encrypted metadata information and may guarantee immutability of the metadata. Both aspects are currently under development as proof of concept, with very encouraging preliminary results thus far.

Table 3 sums up all the information of the short comparison between the three mechanisms made in this section. It is clear that the proposed mechanism seems to

perform at least equally well, while in some characteristics it seems to prevail, such as Ease of storing/retrieval and Expandability.

Table 3. Evaluation and comparison of the mechanisms

Characteristic/ Mechanism	**MEDAL**	**CoreKG**	**Proposed approach**
Granularity	High	High	High
Ease of storing/retrieval	Medium	Medium	High
Size of metadata	Medium	Medium	Medium
Expandability	Undefined	High	High

4.4 Summary

This chapter proposed a novel framework for standardizing the processes of storing/retrieving data generated by heterogeneous sources to/from a Data Lake organized with ponds architecture. The framework is based on a metadata semantic enrichment mechanism which uses the notion of blueprints to produce and organize meta-information related to each source that produces data to be hosted in a Data Lake. In this context, each data source is described via two types of blueprints which essentially utilize the 5Vs Big Data characteristics Volume, Velocity, Variety, Veracity and Value: The first includes information that is stable over time, such as, the name of the source and its velocity of data production. The second involves descriptors that vary as data is produced by the source in the course of time, such as the volume and date/time of production.

Every time data sources or data are pushed in and out of the Data Lake, the stable and dynamic blueprints are updated thus keeping a sort of history of these transactions. Essentially the description of the sources helps to treat and manage many, multiple, and different types of data sources and to contribute to Data Lakes' metadata enrichment before and after these sources become members of a Data Lake. When a data source

becomes part of the Data Lake the metadata schema is utilized, describing the whole Data Lake ontology. The filtering and retrieval of data is based on this metadata mechanism which involves attributes from the 5Vs, attributes such as last source updates and keywords.

A short comparison to other existing metadata systems revealed the high potential of our approach as it offers a more complete characterization of the data sources and covers a set of key features reported in literature and expanded in this work. Furthermore, it provides the means to perform efficient and fast retrieval of the required information.

Building on the framework presented in this chapter, the next chapter extends the concept of metadata semantic enrichment and blueprint-based data organization by incorporating the notions of ponds and puddles for finer granularity, specifically to support process mining activities. While this chapter focuses on organizing data within a Data Lake, the following chapter introduces ponds for larger-scale data storage and puddles for smaller, formatted portions of data, both tailored to facilitate process mining. These structures enable more efficient storage, retrieval, and analysis of data related to business processes, particularly in smart manufacturing environments. Moreover, the approach is applied to a real-world factory case study, showcasing the practical implementation of the Data Lake framework. By utilizing ponds and puddles in this context, the framework is able to support process mining, helping the factory gain real-time insights into operations, optimize decision-making, and enhance overall production efficiency.

CHAPTER 5 : A SCALABLE DATA LAKE SEMANTIC FRAMEWORK USING PONDS AND PUDDLES

5.1 Introduction

Nowadays, in the era of Big Data, with huge amounts of information produced and consumed, data is considered as the "power" of businesses only if it is properly processed to offer mainly decision support. Most companies have a lot of unused data that can be used for process mining. This is a side-effect of the widespread digitization and automation of business processes, which leaves digital traces of real process executions as a byproduct. To the best of our knowledge, the integration of Data Lakes with process mining activities has not received much attention in research literature yet. As Data Lakes appear to be a promising technique for temporal Big Data storing, the present work, apart from the goals described below, intends to cover this gap as well.

This chapter introduces a new approach to handle Big Data in terms of storing and retrieval, which intends to serve best process mining activities that use this data. More specifically, a novel metadata mechanism is proposed that provides the ability to characterize and describe data sources, data items and process related information which are stored in a Data Lake by means of blueprints as presented in the previous chapter. The proposed approach is an extension of the previous chapter, which builds upon the notions of Data Lakes and blueprints to add the following contributions: (a) a separate class of blueprints to account for the information related to process mining activities applied in a smart manufacturing environment (processes, events, machines); (b) the actions to store, retrieve and process data produced by various sources (e.g., sensors) and relate to workflows and mining activities (e.g., events, sequencing, dependencies, etc.); (c) an extension to the Data Lake architecture where the notion of data puddles is introduced to be used for storing smaller portions of data according to some formatting criterion; and (d) the successful application of the framework on a real-world industrial case study,

which, on one hand is quite rare in literature to report (i.e., a real business customer to study), and on the other, it has yielded some very interesting findings.

The target here is to build a unified Data Lake-based business information standardization model, which is tailored to the needs of manufacturing organizations and consists of a number of blueprint entities. These blueprints were essentially presented in the previous chapter and are extended in this chapter to describe an environment that produces large amounts of different types of data in a specific, disciplined form, including the data sources and their outputs, as well as the business processes. The entities in the business process related blueprints describe and correlate the process information stored in the Data Lake, offering also links to events interacting in chronological order and based on dependencies. These special blueprints codify, integrate, and contextualize business data and processes. They provide parameterized solution-aware patterns that represent operational processes and inter-relate a variety of data of diverse types, critical functional, sensor, and performance factors in business production. These blueprints are considered part of the smart manufacturing intelligence environment. They are built using ontologies and utilizing a dedicated blueprint processing data mechanism along with event logs to facilitate efficient execution of process discovery, conformance checking, and model enhancement.

5.2 A pond and Puddle Data Lake Architecture Supporting Process Mining

As mentioned above, an extended, unified standardization framework for smart manufacturing and business process related data residing in a Data Lake is introduced in this chapter, which utilizes a semantic metadata enrichment mechanism via blueprints. The latter utilizes the 5Vs Big Data characteristics and ontologies to assist data processing (storing and retrieval) in Data Lakes with pond architecture, with emphasis on organizing and preparing data to facilitate process mining.

According to the proposed pond architecture, a Data Lake consists of a set of data ponds, and each pond hosts / refers to a specific data type: structured, semi-structured and unstructured. Each pond contains a specialized storage and data processing system depending on the data type (Sawadogo et al., 2019) .

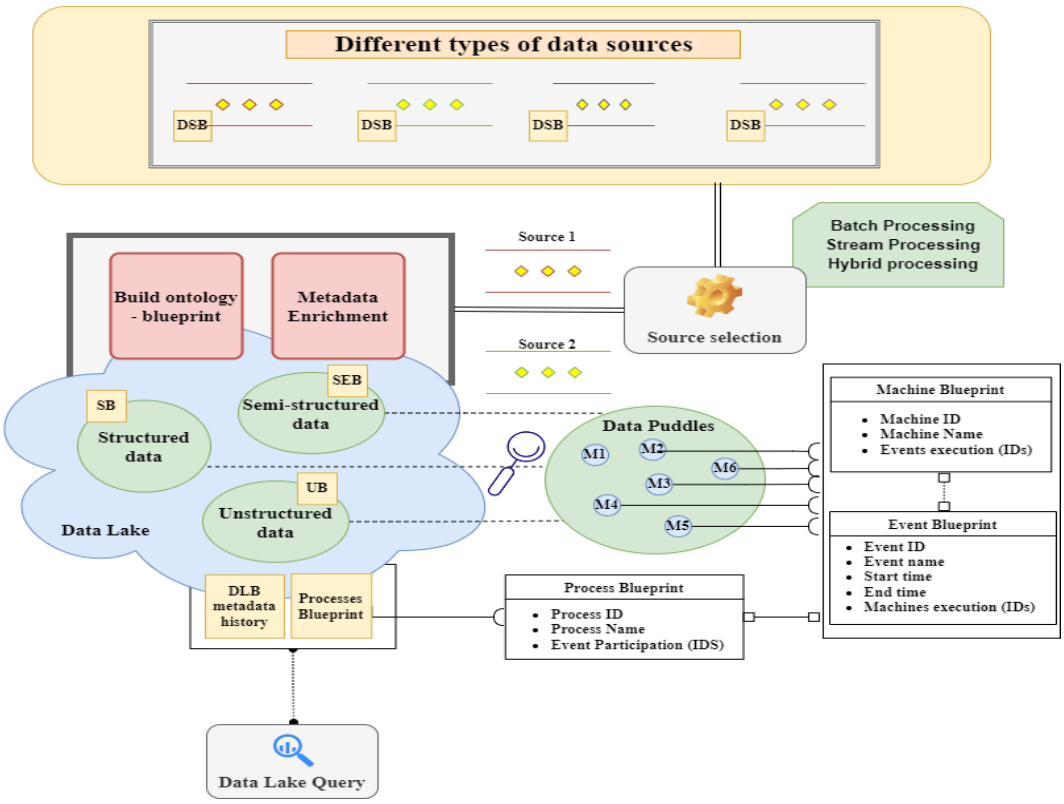

Figure 7. The architectural structure of the pond and puddle proposed approach

Data puddles contain smaller, pre-build datasets which store for example to our methodology data that machines produce in the production line (Machine and Event Blueprints). Each, ponds contains puddles where ponds divided according data types, while puddles are smallest data collections of each pond.

Process mining is performed mainly by using time-stamped data logs. In a Data Lake, however, there are various types of unstructured and semi-structured data, such as images, video, and sounds that may lack time information. Furthermore, structured data as well may not be ready to participate in process mining activities, mainly because it does not have timestamps. To achieve a uniform and constrained approach to the way related data is stored, we will adopt the data blueprint for the approach presented in the previous chapter (see Figure 7) and extend it by creating three new manufacturing blueprints that describe the data produced by machines and processes in a factory. This is performed by enriching the metadata manufacturing semantics of the Data Lake framework that will prepare the data to be used by process mining tasks.

As mentioned above, this chapter builds upon the framework of Chapter 4, which is based on a metadata semantic enrichment mechanism that uses the notion of blueprints

(Papazoglou & Elgammal, 2018) to produce and organize meta-information related to each source producing data to be hosted in a Data Lake. In this context, each data source is described via two types of blueprints as shown in Figure 8, which essentially utilize the 5Vs Big Data characteristics Volume, Velocity, Variety, Veracity and Value. The first includes information that is stable over time, such as the name of the source and its velocity of data production. The second involves descriptors that vary as data is produced by the source in the course of time, such as the volume and date/time of production. The combination of these two blueprints creates the Data Source Blueprint (DSB).

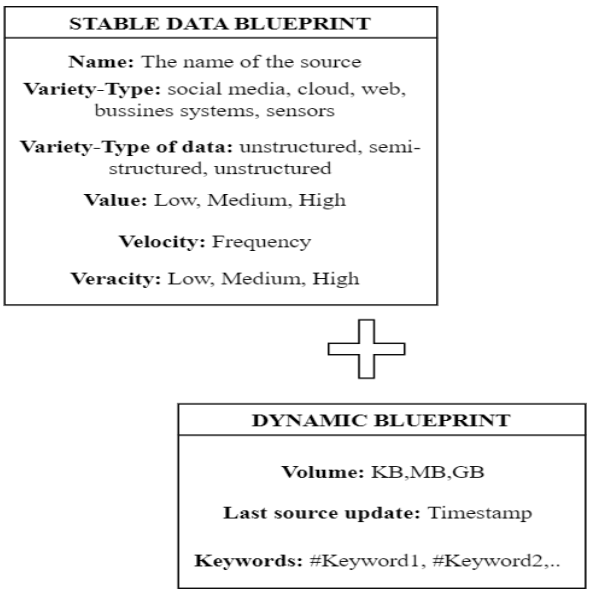

Figure 8. Stable and Dynamic Data Blueprint

As shown in Figure 7, every time data sources or data are pushed in and out of the Data Lake (for example, Source 1 and Source 2, which are accompanied by their DSBs), the stable and dynamic blueprints are updated thus keeping a sort of history of these transactions on the Data Lake Blueprint (DLB) metadata history, which include the Structured Data Blueprint (SB), the Semi-structured Data Blueprint (SEB), and the Unstructured Data Blueprint (UB) residing in the data ponds.

Essentially, the description of the sources helps to treat and manage many, multiple, and different types of data sources and to contribute to the Data Lakes' metadata enrichment before and after these sources become its members. When a data source becomes part of the Data Lake, the metadata schema is utilized, describing the whole Data Lake ontology. The filtering and retrieval of data is based on this metadata mechanism which involves attributes from the 5Vs, such as last source updates and keywords.

This chapter also extends the framework proposed in Chapter 4 by creating within the ponds the so-called data puddles, which are smaller, pre-build datasets, as shown in Figure 7, that store data machines produce in the production line (Machine and Event Blueprints). Furthermore, the existing framework is modified to include a process-related

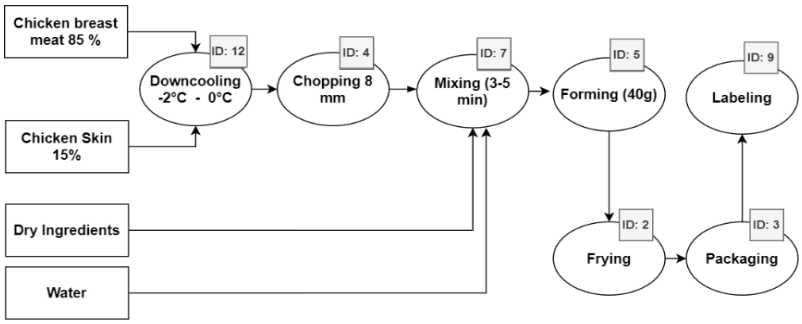

Figure 9. PARG chicken nuggets manufacturing process

blueprint which provides information about the participation of each machine in various processes during production, that is, which machine executes each event within a process cycle.

To test that the proposed framework works properly and that it meets the needs of a real factory, we investigated its applicability to a major local industrial player, namely Paradisiotis Group (PARG) (https://paradisiotis.com/). PARG is one of the most significant local companies and experts in the field of poultry farming and trading of poultry meat in Cyprus. It offers a wide selection of high-quality products that meet the needs of modern consumers for convenience in cooking and healthy eating. The business processes and the manufacturing data of the factory are, of course, confidential, and therefore, this chapter reports only a part of the processes with not so many details, associated with a masked and downgraded version of the data. Nevertheless, the case study is more than enough to demonstrate the basic principles of the proposed framework and prove its applicability and usefulness.

Every process in the manufacturing cycle consists of events and each event is executed by a machine that participates in a specific blueprint. Figure 9 describes an example of a process followed during the production of chicken nuggets at the PARG factory. First, the ingredients are prepared consisting of 85% chicken breast meat and 15% chicken skin. These ingredients are pushed into a flaker machine that cools down the raw material to a uniform temperature ranging between -2°C and 0°C and then shapes it to blocks. Subsequently, the blocks are chopped into smaller pieces of 8mm size by a mincing machine. Then, these minced pieces are mixed with dry ingredients (such as spices) and

water, for 3-5 minutes and the end-product created consists of chicken nuggets formed to have a net weight of 40g each. Finally, the chicken nuggets are deep fried and packaged with appropriate labeling.

The process analyzed above is practically followed for all pre-fried products, such as drumsticks and meatballs, with the only difference being in the forming, with size and shape changing accordingly depending on the product. In addition, for fresh products, such as burgers, the forming event is omitted and another event is added before down-cooling, namely the deboning of raw material, which is executed by the Tappler machine (out of scope of the present study). In all these processes, the material is pushed from machine to machine via conveyors.

If we analyze the process of producing chicken nuggets depicted in Figure 9, we may derive that the following seven events take place:

- Down-cooling (ID: 12)
- Chopping 8m (ID: 4)
- Mixing (ID: 7)
- Forming (ID: 5)
- Frying (ID: 2)
- Packaging (ID: 3)
- Labeling (ID: 9)

This process consists of events that are carried out by machines which produce data during execution. To prepare this data for process mining we use the process blueprint that provides information about production and is part of the manufacturing Data Lake, as shown in Figure 1, as well as a machine blueprint and an event blueprint, which describe the data that machines produce during the production cycle.

The process described previously involves 10 machines (3 of which perform the same task) in the chicken nuggets production as follows:

- Flaker (ID, Type: FL2, Down-cooling)
- Mincer (ID, Type: MC1, Chopping)
- Mixer (ID, Type: MX3, Mixing)
- Former (ID, Type: FRM1, Forming)
- Fryer (ID, Type: FR4, Frying)
- Labeler (ID, Type: LBI, Labeling)

- Packager (ID, Type: PC3, Labeling)
- Conveyors (x3) (ID, Type: CV1, CV2, CV3 Conveyor)

These machines execute multiple events in a specific order during the production of nuggets. The type of machine that executes a particular event is stored in the Manufacturing Blueprint thus being able to check the availability of machines of this type.

The Process Blueprint captures the Process ID, the Process name and Events participation. For example, the chicken nuggets process blueprint is described as follows:

- Process ID: 100
- Process Name: Nuggets production
- Events execution: 12, 4, 7, 5, 2, 3, 9

The Event Blueprint consists of Event ID, Event name, Start Time, End Time, Expected execution time, Executed by (Machine Type) and Dependencies, the latter describing what other event has to be executed before the current event may start. For example, the Mixing event is described by the following Event Blueprint:

- Event ID: 7
- Event name: Mixing nuggets ingredients
- Start time: Timestamp
- End time: Timestamp
- Expected execution time: 4 minutes
- Executed by (ID, Type): MX
- Dependencies: 12, 4

Furthermore, the Event Blueprint captures the Expected execution time of the event to be able to check for abnormalities in case the execution of the event is delayed for some reason. This is performed through the analysis of the start and end times, the resources utilized, the roles etc., so as to trace the causes of this delay. The Machine Blueprint captures Machine ID, Machine Type, Machine name and the IDs of the machines that may execute the specific event.

Essentially, the proposed information structure for the description of the data sources that exist in a smart factory efficiently supports the management of multiple data formats. It also allows data to be prepared for process mining through the metadata semantic enrichment that requires events to be timestamped and set in chronological order

according to the process executed. Finally, sources that produce unstructured and semi-structured data that are stored in the relevant pond of the proposed approach may also be linked with the rest of the event information and provide added value to the analysis of a certain process. For example, data from a sensor installed on some machine in the production line (e.g., counting packages in the case of chicken nuggets) is coupled with photos captured of a certain spot (e.g., when packaging or labeling) to allow for assessing productivity or the level of defects, offering a complete root-cause analysis.

5.3 Experimentation

To demonstrate the applicability and effectiveness of the proposed framework, we will use the chicken nuggets production process of the PARG factory described in the previous section. The target here is twofold: First, to demonstrate how the proposed approach was used in practice for the PARG case-study and highlight some interesting findings. Second, to make a short assessment of different Data Lake structures, including the proposed one, according to specific metrics and present the results that show the superiority of this approach.

Figure 10 presents an excerpt of the data produced by the Flaken and Mincing machines at PARG factory during the manufacturing process presented in Figure 9. This data is stored in the structured data pond of Figure 7. More specifically, each dataset produced by the two different machines is stored using a different puddle within the data pond. During this process other formats of data are generated as well, such as video and images (omitted due to size limitations) and, since these constitute unstructured and XML-based data (semi-structured), they are stored in the respective pond and distinct puddles, respectively.

As presented in the previous sections, according to the process blueprint describing process with ID100, the execution of events is in sequence 12, 4, 7, 5, 2, 3, and 9. In order to retrieve the data for this specific process, the following SPARQL query should be executed:

```
SELECT ? DLsources
WHERE {
? process ID <has ID> 100 }
```

Executing this query on the Data Lake blueprint first triggers the retrieval of information on event execution from the process blueprint. Using this information as the basis, all relevant data for this process is retrieved and mapped depending on the order in which events are executed by machines. As shown in Figure 7, the process blueprint is connected with the event blueprint, which provides the expected execution time of each event by a machine, as well as the type of machine that executes this event, while the machine blueprint describes the events that can be executed by each machine. This information was combined with the data retrieved from the appropriate puddles residing in the specific pond of the Data Lake and were made ready for use in the process mining activities that followed, the latter yielding some interesting results. For confidentiality purposes, we report here two of them very abstractly.

A few delays were encountered in some of the steps, which were revealed during this analysis by comparing the expected with the actual execution time. This led to further investigating these delays through the start and end times of the relevant events. It was noticed that optimization in the way the sequence of the execution of tasks (events) by the machines had ample room for improvement in terms of timing: There were delays in commencing operation from a machine after the previous task was finished. This was partly attributed to roles and resources within the production line, as various tasks were shared amongst employees and, in some cases, multitasking was an issue. All the above were communicated to the senior management of PARG who acknowledged their value for future actions.

The analysis was also supported by unstructured data (images of the production line) which was recognized by all parties involved (analysts, managers, workers) to have played a crucial role in identifying bottlenecks. Therefore, as the proposed framework allows for utilizing both unstructured and semi-structured data for process mining, it was considered a significant benefit.

The second experimental aim was to investigate in general the readiness of the manufacturing data residing in a Data Lake to participate in process mining tasks, when the Data Lake has the following structure:

- without the proposed metadata enrichment mechanism
- with the metadata mechanism, without a pond architecture
- with the metadata mechanism and a pond architecture

- with the metadata mechanism, and using pond and puddle architecture (the proposition of this chapter)

The following characteristics/metrics were utilized for each Data Lake structure:

- Granularity
- Ease of storing/retrieval
- Process mining readiness
- Expandability

Granularity, as defined in Chapter 4, refers to the ability to refine the type of information that needs to be retrieved, and specifically, it now applies to a particular factory process. This ability is expressed by the number of fine-grained levels the Data Lake mechanism supports for defining the information sought. Ease of storing/retrieval also defined in Chapter 4 refers to the ability of the Data Lake structure to store or retrieve data in the Data Lake in a simple and easy way. It is assumed here that the retrieval action is efficient enough to return the desired parts of the information sought. This characteristic is it now applies on the number of steps that need to be executed for the process of storing and retrieving data items to be completed. Moreover, we define Expandability as the ability to expand the structure of the Data Lake and the metadata mechanism with further functional characteristics or other supporting techniques and approaches, such as visual querying and Blockchain. Obviously, the more open the mechanism for expansion, the better. Finally, Process mining readiness is reflected in the number of steps that need to be executed after the query is executed for the data to be fed to process mining activities. The aforementioned characteristics are evaluated using a Likert linguistic scale, including the values Low, Medium, and High. Table 4 provides a definition of Low, Medium and High for each characteristic introduced.

Table 4. Definition of Low, Medium, and High of each assessment characteristics

Characteristic/ Level	**Low**	**Medium**	**High**
Granularity	1 level	2 levels	3 or more levels
Ease of storing /retrieval	5 or more actions	3-4 actions	2 actions maximum
Expandability	No or limited	Normal	Unlimited
Process mining readiness	4 – 5 actions	2 – 3 actions	1 action maximum

Table 5. Evaluation and comparison of Data Lake structures

Approach	**Granularity**	**Ease of storing /retrieval**	**Process mining readiness**	**Expandability**
Without metadata mechanism	Low	Low	Low	Low
With metadata mechanism without pond architecture	Medium	Medium	Low	Unlimited
With metadata mechanism with pond architecture	High	High	Medium	Unlimited
With metadata mechanism with pond and puddle architecture	High	High	High	Unlimited

As an example, let us now assume that the PARG factory's Data Lake owner wants to retrieve all data present in the Data Lake relevant to the process of chicken nuggets production presented in the previous section so as to feed it to the process mining stages of discovery, conformance checking, and model enhancement. Note that the PARG factory consists of hundreds of production processes. In order to retrieve the data, the following SPARQL query should be formed and executed:

```
SELECT ? Dlsources
        WHERE {
         process ID <has ID> 20
         }
```

The metadata mechanism with pond architecture (third from the top in Table 5) may be considered as the benchmark of our comparison. It presents High Granularity, High Ease of storing/retrieval using the stable and dynamic data source blueprint descriptions, with a Medium Process Mining Readiness, and High Expandability. These values are attributed as follows: High Granularity is achieved using keywords that essentially describe the sources and the blueprint values. This enables the user to define details at the level of the properties offered, enabling the retrieval of data based on fine-grained query-like information. The High Ease of storing/retrieval is achieved by the Data Lake metadata history which stores the blueprint description of the Data Lake as each time data

sources are pushed into it. This helps the mechanism to place the sources to a specific pond according to the structure of the data involved (structured, semi-structured and unstructured). This source distribution in the Data Lake also facilitates simple and Easy Storing and Retrieval of the information stored. In addition, the implementation of this Data Lake architecture is based on the Hadoop ecosystem and hence this provides High Expandability. Finally, the metadata mechanism provides Medium Level of Process Mining Readiness due to the lack of defining dependencies and describing the process, events and machines in the production line.

It is logical that, as we move to the upper structural forms of Table 5, the evaluation of the selected characteristics gets worse: Assuming that PARG's Data Lake has an architecture without metadata, a SPARQL query could not be executed at all. In addition, with this structure, all the data is pushed to the Data Lake without any management policy and as a result the Data Lake is highly likely to transform into a Data Swamp, while at the same time it would take quite a few actions (more than five) to retrieve data because all datasets will need to be visited and checked if they are related to the specific process. Process Mining Readiness is Low as no clear separation of events, type, dependencies etc., exists, let alone the fact that data needs to be timestamped. Finally, in the absence of a management mechanism, Expandability may be characterized as Low. Taking into consideration now that the PARG's Data Lake has a structure with the proposed metadata enrichment mechanism but without a pond architecture, the data is pushed to the Data Lake following a metadata policy. As a result, this Data Lake can be characterized with Medium Granularity due to the fact that the metadata mechanism provides 2 levels of Granularity, which are provided by the metadata mechanism and the pond the data is stored in, with Low to Medium Ease of storing/retrieval as the data is pushed to the Data Lake with its metadata. Therefore, in order to retrieve it, one needs to access and process the metadata to check if a certain piece of information is related to the specific process examined. This Data Lake structure could be characterized also with Low Process Mining Readiness because, after executing a query, the data is not separated according to its type and an additional task to separate and timestamp it should be performed, along with proper definition of any dependencies. Finally, it can be characterized with Unlimited Expandability as the metadata mechanism allows for practically any extension.

The proposed metadata mechanism with pond and puddle architecture can be characterized similarly to the previous benchmark (3rd row in Table 5), but with High

Process Mining Readiness. By extending the mechanism reported in Chapter 4 and introducing the process blueprint that captures the specific events triggered while a process is executed, the event blueprint that captures the type of machines that participate in a process and the machines blueprint that captures the events that specific machines can trigger, results in a data environment ready to perform process mining readiness. Furthermore, extending data ponds with data puddles where each puddle stores data from each machine on PARG factory enables a query to provide the requested data of a specific process to be separated according to its format types.

Table 5 sums up all the information of the short comparison between the Data Lake structures made in this section. It is evident that the proposed mechanism outperforms all alternatives in terms of the Process mining readiness characteristic as a result of the extensions made in the blueprints.

5.4 Summary

This chapter proposed a novel smart manufacturing Data Lake framework for process mining utilizing a semantic enrichment mechanism via metadata blueprints. The framework utilizes the 5Vs Big Data characteristics and blueprint ontologies to assist data processing (storing and retrieval) in Data Lakes, the latter being organized with a pond architecture that hosts different types of data, structured, semi-structured and unstructured, enhanced by data puddles. The puddles consist of data produced by machines in the production line and essentially prepare the data in the ponds for process mining activities.

The applicability of the framework was demonstrated and assessed through a real-world case-study on a local poultry meat production factory. The process of producing chicken nuggets was modeled with relevant data captured, stored and processed. Process mining revealed delays and bottlenecks in the sequencing of the execution of events by machines and personnel which may be avoided by optimizing task sharing amongst roles. The senior management of the factory greatly appreciated the support of the proposed approach for decision support with respect to production control.

Furthermore, a short comparison with different Data Lake structures was performed revealing the high potential of the proposed approach as it offers a more complete

characterization of the data sources and covers a set of key features reported in literature. Especially, the inclusion of data paddles can greatly enhance the management of manufacturing data that can later participate in process mining activities, such as discovery, conformance checking, and model enhancement utilizing all available data types.

To build upon the framework presented in this chapter, the next critical step in enhancing Data Lake management is addressing the security and integrity of the data and metadata. While Chapter 5 has focused on optimizing data organization and processing for process mining through semantic enrichment and the innovative pond and puddle architecture, it is essential to ensure that these data assets remain secure, accurate, and protected against unauthorized alterations. As Data Lakes evolve to accommodate increasingly complex and diverse data sources, the challenge of safeguarding metadata and ensuring compliance with privacy and regulatory standards becomes more pressing and challenging. Chapter 6 explores how integrating blockchain technology into the proposed framework can address these concerns by providing a decentralized, secure environment for managing metadata, protecting against fraud, and ensuring authorized access, thus completing the data management lifecycle from collection to secure utilization.

CHAPTER 6 : ENHANCING DATA LAKE SECURITY AND METADATA MANAGEMENT THROUGH BLOCKCHAIN INTEGRATION

6.1 Introduction

Despite the great and drastic solutions proposed in recent years in the area of Big Data Management and Smart Data, the treatment of Big Data produced by multiple heterogeneous data sources remains a challenging and unsolved problem. As previously mentioned, a Data Lake is a repository that can store a large amount of structured, semi-structured, and unstructured data. It is a place to store every type of data in its native format with no fixed limits on account size or file that offers high data quantity to increase analytic performance and native integration. As Data Lakes are a quite new data storage architecture linked with Big Data processing, it presents some unsolved challenging problems (Khine & Wang, 2018). Two of the major and challenging problems of Data Lake are the following: (i) there is no descriptive metadata or mechanism to maintain metadata leading to data swamp (Sawadogo & Darmont, 2020) , and, (ii) security (privacy and regulatory requirements) and access control as data in a lake can be replaced without the oversight of the contents. The former was addressed in the previous chapters where the semantic enrichment mechanism comprising metadata information via blueprints was presented and evaluated. The aim of this chapter is to create a Data Lake structure (ponds or zones) following the semantic annotation mechanism introduced earlier with emphasis on security, that is, the information included in the metadata is protected from malicious alteration so as to prevent fraud and unauthorized activity between Data Lake of different organizations. Blockchain technology is the means to achieve this data protection as it enhances privacy issues by anonymizing personal data and it is also used to grant authorized access to Data Lake owners.

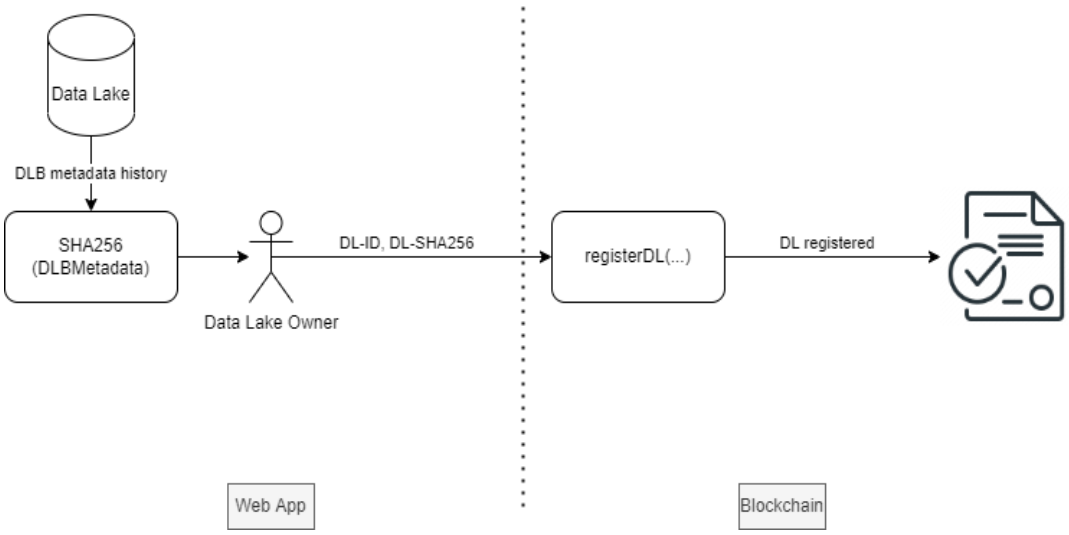

Figure 10. DLMetachain framework architecture DLB metadata history creation

6.2 The DLMetachain Framework Architecture

The goal of the proposed framework is to enrich the existing novel Data Lake pond metadata mechanism framework described in Chapter 4 with the Blockchain technology to provide advanced security and privacy, and to ensure that the data in the Data Lake has not been modified or altered. Blockchain anonymizes personal data by storing only encrypted or hashed references on-chain, not the raw data itself. Smart contracts enforce access control, allowing only authorized users to interact with data under predefined rules. All access actions are transparently logged on the blockchain, ensuring traceability without compromising user identity.

Each time a new source is pushed in the Data Lake, or each time a source generates new data that modifies the dynamic and stable blueprint, a new version of Blueprints is created, thus the DLB metadata history changes as presented in Figure 2 and Figure 7 in previous chapters. To ensure that the DLB metadata will not be altered or modified, a Blockchain smart contract is developed within the proposed framework with a twofold purpose: (i) Allow a Data Lake owner to register a Data Lake into the Blockchain based on its metadata, and, (ii) Allow end-users to verify the correctness of a Data Lake before conducting any actions on it. As outlined in Figure 10, each time a new version of a DLB metadata is created, a new SHA256 value for that specific DLB is generated automatically and is used along with the Data Lake's ID to register the specific Data Lake into the Blockchain via the proposed smart contract. Multiple versions of a Data Lake can be

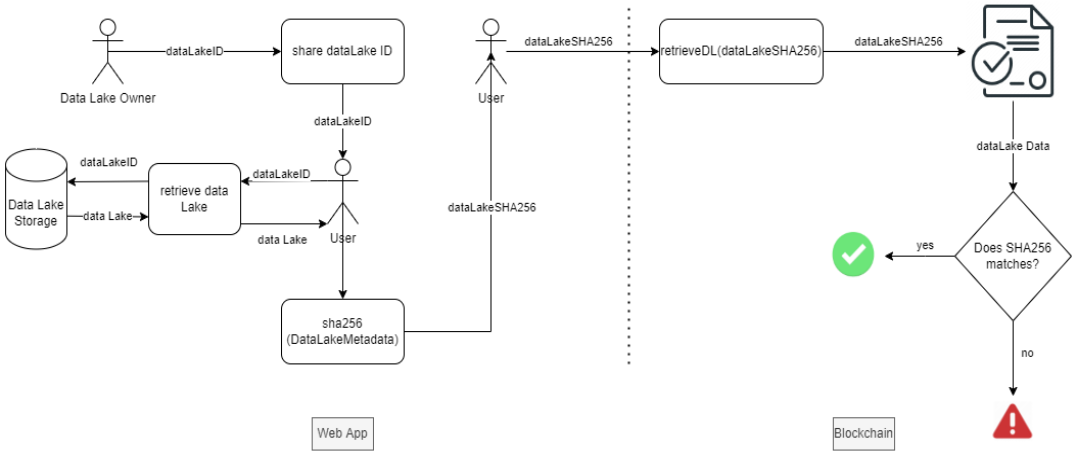

Figure 11. Overview of the system via the end-user's perspective

created for each Data Lake that results in the creation of a unique Data Lake Chain for each Data Lake and a whole new chain for the full system, namely DLMetaChain. In addition, mechanisms were developed that can be used by end-users to verify that a shared Data Lake has not been modified or altered.

Figure 11 presents an overview of the system from the end-user's perspective. Firstly, the Data Lake owner shares a Data Lake ID with an end-user who uses it to retrieve specific metadata. Once the metadata is retrieved, a mechanism that generates the SHA256 value for that specific source is executed and then the user uses that value to check if the Data Lake exists on the Blockchain or not. If the Data Lake was successfully retrieved by the smart contract, then this is a proof that the Data Lake has not been modified, thus the user can execute actions using the specific DLB. If the Data Lake cannot be found on the smart contract, then this means that the Data Lake was compromised, and the end-user is recommended not make use of the Data Lake.

6.3 IoT Data Lake Use case scenario

To demonstrate the effectiveness and usage of the proposed framework we have deployed the recommended smart contract on the Ethereum Rinkeby Test Network. The list of all executed transactions, as well as the source code of the smart contract can be found on the following smart contract address:

0x0E864521Ccf8BD65aBFcC920ec43d93fdf82D80a

The ETH public address *0x6c56618BCbF502b237369551cF2A f7317E 763eDb*, acts as the Administrator of the developed digital application (dApp), thus it can register Data Lakes into the smart contract.

Before evaluation, the *registerDL(…)* function of the smart contract first creates two different versions of a Data Lake that can be found on the GitHub repository, using IoT data and then the SHA256 value for each one of the versions was generated. The SHA256 values of the Data Lake versions are shown below:

DL0_v1:

5a221f47e54beac4c9548116bf9196a23b041c0c664280d018c96a1089643568

DL0_v2:

db9772aefdaa674d0c4b3ebc61a30d2409a8a07f41925a9b2b4192490ce98530

When the SHA256 values of the Data Lakes are generated, the Administrator of the proposed framework can register them to the smart contract by calling the *registerDataLake(…)* function and by providing the Data Lake ID along with its SHA256 value.

The

To test the efficiency and cost-effectiveness of the proposed blockchain-based approach were assessed through practical deployment and testing on the Ethereum blockchain. Specifically, both versions of the Data Lake were registered onto a smart contract deployed on the Ethereum mainnet. The registration process involved calling the *registerDataLake()* function within the smart contract, which securely logs each version of the Data Lake onto the Ethereum blockchain. Furthermore, deploying and interacting with smart contracts on Ethereum involve costs denominated in gas fees, which compensate miners for executing transactions and maintaining the integrity of the blockchain network. Gas fees vary based on network congestion, transaction complexity, and computational resources required. The detailed costs associated with these operations are provided below in Table 6:

Table 6. Cost of deploying and registering Data Lake

Deployment/Functions	Cost (ETH)	Cost (USD)
Contract Deployment	0.001822	6.19
registerDataLake()	0.000121	0.41

Furthermore, as soon as the Data Lake is successfully registered into the smart contract, the Administrator of the dApp can share the source code of the Data Lake with an end-user.

When the end-user wishes to check that the Data Lake was not compromised, he/she firstly uses the proposed dApp to find the SHA256 value of the shared Data Lake and then calls the *retrieveDataLakeUsingSHA256(...)* function to check whether the Data Lake exists or not. To demonstrate the effectiveness of the proposed approach, the specific function was called using as input the value `0x5a221f47e54beac4c9548116bf9196a23b041c0c664280d018c96a1089643568` and the smart contract returned back the results shown in Figure 12.

Figure 12. The SHA 256 matching

Table 7. Minimum time of calling the core functions

Core Functions	Time(s)
registerDataLake()	≈ 10
retrieveDataLakeUsingSHA256()	≈ 1ms

As depicted in Figure 12, the SHA256 value of the given Data Lake matches the one of the specific Data Lake that was successfully registered into the contract. In case the Data Lake was compromised, the SHA256 value would not match the one on the smart contract and, therefore, the end-user would be directed not to execute any action using the specific Data Lake. Table 7 presents the minimum time required to call the core functions of the smart contract. As can be observed in Tables 6 and 7, the cost and the time required to call the main functions of the smart contract are not prohibitive.

6.4 Summary

This chapter proposed a novel framework for standardizing the processes of storing/retrieving IoT data combined with data generated by heterogeneous sources to/from a Data Lake organized with ponds architecture and focusing on providing security and privacy. The framework is based on a metadata semantic enrichment mechanism which uses the notion of blueprints to produce and organize specific meta-information called (Data Lake Blueprint (DLB)), which is related to each source that produces data to be hosted in a Data Lake. In this context, each data source is described via two types of blueprints which essentially utilize the 5Vs Big Data characteristics Volume, Velocity, Variety, Veracity and Value: The first includes information that is stable over time, such as, the name of the source and its velocity of data production. The second involves descriptors that vary as data is produced by the source in the course of time, such as the volume and date/time of production.

The goal of the framework presented in this chapter was to ensure that the DLB metadata or Data Blueprint as presented in Chapter 4 will not be altered or modified. To this end, a Blockchain smart contract was developed aiming at providing the ability to a Data Lake owner to register the Data Lake into the Blockchain based on its metadata, and at allowing end-users to verify the correctness of a Data Lake before conducting any actions on it.

Each time a new version of a DLB metadata is created a new SHA256 value for that specific DLB is generated automatically and is used along with the Data Lake's ID to register the specific Data Lake into the Blockchain via the proposed smart contract. Multiple versions of a DLB can be created for each Data Lake that results in the creation of a unique Data Lake Chain for each Data Lake and a whole new chain for the full system.

CHAPTER 7 : INTEGRATING VISUAL QUERYING AND NFTS FOR SECURE AND EFFICIENT BIG DATA MANAGEMENT IN DATA LAKES

7.1 Introduction

This chapter extends and enhances previous work on Chapters 4, 5 and 6 by adopting the basic principles of manufacturing blueprints (Papazoglou & Elgammal, 2018) and modifying their purpose and meaning to reflect the description and characterization of sources and the data they produce via the utilization of the 5Vs Big Data characteristics. Recall that these characteristics describe data sources by means of specific types of blueprints through an ontology-based description representation. Big Data sources will thus be accompanied by a blueprint metadata description before they become part of a Data Lake. The Data Lake follows the pond architecture described in Chapters 4 and 5, while the data source selection process and retrieval is performed via a dedicated Visual Querying environment. Additional security characteristics are induced in this scheme by utilizing Blockchain as in the previous chapter but under a different perspective. Specifically, for each selected source, a mint function is executed through a smart contract which is responsible for creating Non-Fungible Tokens (NFTs) for that source and storing them in the Blockchain.

7.2 Methodology

As mentioned in the previous section, data processing (storing and retrieval) in Data Lakes organized with pond architecture will be facilitated by integrating the 5Vs Big Data characteristics and blueprint ontologies. Storing and retrieval of data sources will be supported by visual querying providing Digital Twin characteristics. The ownership of

the selected data sources will be recorded during Data Lake storing in the Blockchain as an NFT.

Using the pond architecture, a Data Lake consists of a set of data ponds each hosting / referring to a particular type of data. Based on the data type, each pond contains a specialized storage system and data processing. As presented in Figure 13, a Data Lake with pond architecture uses dedicated ponds to store structured, unstructured, and semi-structured data from each source. As will be demonstrated later, the innate pond architecture is particularly useful for extracting information from the Data Lake via Visual Query.

First, Big Data sources are filtered and selected by the Data Lake owner using a Visual Query environment before they become part of the Data Lake as shown in Figure 13. Since data sources that are candidates for inclusion in the Data Lake are characterized according to the blueprint attributes shown in Figure 14, their selection is performed by defining specific values for these attributes.

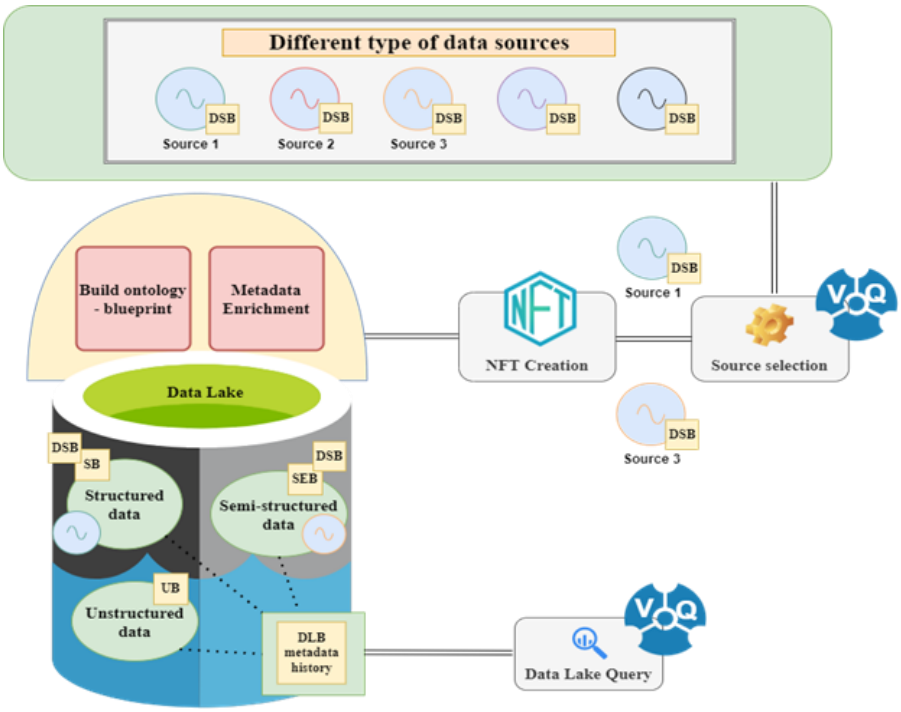

Figure 13. Visual Querying data sources selection metadata enrichment mechanism using 5Vs

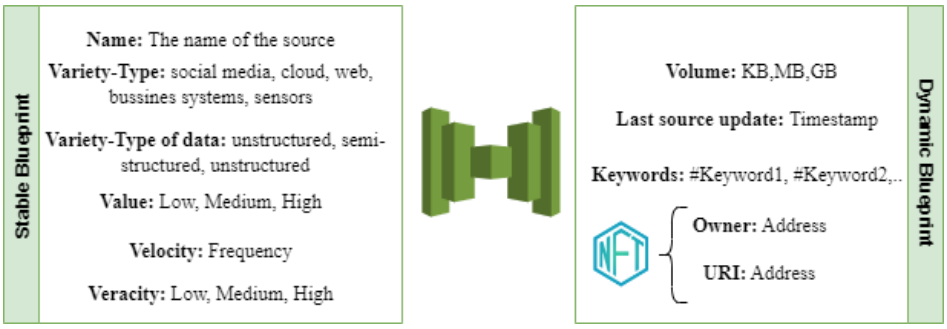

Figure 14. Data source blueprints description extended for NFT utilization

The Dynamic blueprint, as presented in Figure 14, now contains also attributes such as TokenID of the NFT created for the specific data source. Using the TokenID, the Owner and URI of the data source can be retrieved as presented also in Figure 14. The TokenID value is NULL until the data source is selected to be part of the Data Lake. When the data source is selected, the NFT is minted and the TokenID gets the value that is associated with the Owner and URI of the specific NFT. Every time the DSB of the specific data source is modified, a new URI is created pointing to the NFT. In essence, the dynamic and stable blueprints form the DSB are RDF (Resource Description Framework) files which follow the XML structure.

Let us assume that we have three data sources which produce energy data and are candidate to be part of a Data Lake, which bear the characteristics listed in Tables 8 to 10. SPARQL, as well as Visual Query that builds SPARQL via a visual interface, enables us to query all RDF resources prior to, or after their ingestion into the Data Lake. SPARQL queries can be executed without technical knowledge in a Visual Query environment. In order to query these sources from the visual environment, each data

```
Source 1 Stable Blueprint
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.blueprints.com/source1">

<rdf:Description
rdf:about="https://www.blueprints.com/stable">
  <cd:name>SourceE1</cd:name>
  <cd:varietytype>Business Systems</cd:varietytype>
  <cd:varietypeof>Structured</cd:varietypeof>
  <cd:value>High</cd:value>
  <cd:velocity>3600s</cd:velocity>
  <cd:veracity>High</cd:veracity>
</rdf:Description>

</rdf:RDF>
```

```
Source 1 Dynamic Blueprint
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.blueprints.com/source1">

<rdf:Description
rdf:about="https://www.blueprints.com/dynamic">
  <cd:volume>GB</cd:volume>
  <cd:lastupdate>11/08/2022 07:50</cd:lastupdate>
  <cd:keywords>
    <li>#Energy Prices</li>
    <li>#Energy Autonomy</li>
  </cd:keywords>
  <cd:owner>NULL</cd:owner>
  <cd:uri>NULL</cd:uri>
</rdf:Description>

</rdf:RDF>
```

Figure 15. SDB for Source 1 written in XML

source's description must be in RDF form as previously mentioned, which can be obtained by a public API (RDF API). Using the blueprint ontology description, Figure 15 illustrates the RDF files written in XML for Source 1. Based on the characteristics of each source, the XML follows the same structure. As presented in Figure 15, attributes Owner and URI are NULL, due to the fact that the specific data source has not been selected yet to be a member of the Data Lake. Once this selection is made, it will lead to creating an NFT for that particular source.

Table 8. Source 1 DSB attributes

Stable Blueprint	**Value**	**Dynamic Blueprint**	**Value**
Name	Source-E1	Volume	GB
Variety-Type	Bus. Systems	Last Update	11/08/2022 07:50
Variety-Type of data	Structured	Keywords	# Energy Prices # En. Autonomy
Value	High	TokenID	Null
Velocity	3600 sec		
Veracity	High		

Table 9. Source 2 DSB attributes

Stable Blueprint	**Value**	**Dynamic Blueprint**	**Value**
Name	Source-E2	Volume	MB
Variety-Type	Web	Last Update	11/08/2022 07:40
Variety-Type of data	Semi-Structured	Keywords	#Weather Conditions
Value	High	TokenID	Null
Velocity	1800 sec		
Veracity	Medium		

Table 10. Source 3 DSB attributes

Stable Blueprint	Value	Dynamic Blueprint	Value
Name	SourceE3	Volume	GB
Variety-Type	Sensors	Last Update	11/08/2022 07:30
Variety-Type of data	Semi-structured	Keywords	# Energy Production # En. Autonomy
Value	High	TokenID	Null
Velocity	10sec		
Veracity	Medium		

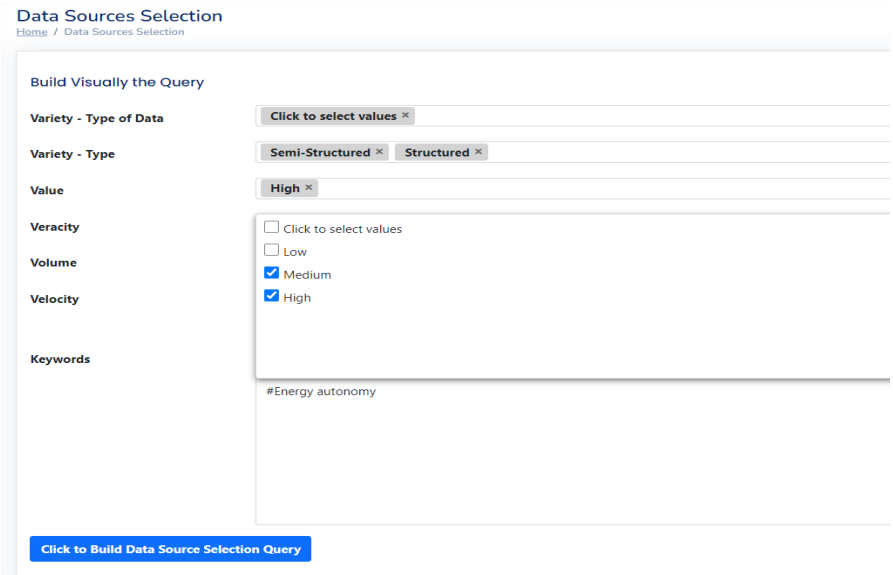

Figure 16. Visual Querying environment source selection (a)

For each selected source, a mint function is called through a smart contract which is responsible for creating NFTs for that source and storing it in the Blockchain. The function receives as parameters a Blockchain address representing the source, which is used for verifying the ownership of the source, and a URI that is automatically created from the system and used for storing the metadata of the source to IPFS (Inter Planetary File System). IPFS is a peer-to-peer distributed file system for storing and accessing files, and it is widely used for NFTs due to its high robustness, availability, and immutability. In order to verify the ownership of a specific data source, the NFTs of the data sources can be retrieved through the smart contract. The smart contract is developed in Solidity, which is a high-level object-oriented programming language for implementing mainly

Ethereum Virtual Machine (EVM)-compatible smart contracts. It is published in the Rinkeby Test Network, which is an EVM-compatible testing Blockchain network. Let us now assume that the three sources described in Tables 8-10 are candidate to become members of the Data Lake. Therefore, the Data Lake owner needs first to build a specific SPARQL, through the Visual Query environment, according to his/her preferences for the data sources characteristics. For example, if the Data Lake owner wants to insert into the Data Lake data sources with Variety type of data Structured OR Un-structured, Value High, Veracity Medium OR High, and with keywords #Energy Autonomy, a source selection middleware is fed with these preferred conditions and executes the query after the following SPARQL query is built by the Visual Querying environment presented in Figure 16.

The execution of the query will result in the creation of the NFTs for Sources 1 and 3, the stable and dynamic blueprints of which will be added to the Data Lake's RDF schema. Therefore**,** the DSBs of the selected sources become part of the Data Lake and contribute to the Data Lake's metadata semantic enrichment, something which is necessary for their efficient retrieval from the Data Lake. The selected data sources are then distributed to the specific Data Lake Pond according to the variety type attribute of the DSB. Their DSB is added to Structured Blueprint (SB), Semi-structured Blueprint (SEB) and Unstructured Blueprint (UB) according to the Data Lake Pond to which it is distributed. Essentially, SB, SEB and UB constitute the DLB metadata history as presented in Figure 13. In essence, this process and its associated characterization helps to manage and enrich the metadata of multiple and diverse sources before and after they become members of the Data Lake. The Data Lake uses the DLB metadata history for filtering and retrieving data based on blueprints and their metadata as soon as a data source is added. This involves characteristics like the type of data produced, the size, the speed, the accuracy, the significance of the data produced by the sources, etc. DSB, as well as the DLB metadata history, provide effective guidance for retrieving, via Visual Querying environment, data from the Data Lake. As presented in the example that follows, the Visual Querying environment is fed with DLB metadata history; as a result, the Data Lake owner is able to query the Data Lake according to the attribute(s) included to the DLB metadata history. It is important to note that in the case of the dynamic blueprint, this part of the metadata will be dynamically updated in the DLB metadata history whenever new data is produced

by the sources, or as deemed necessary (e.g., if the location of an associated pond is changed).

To further demonstrate the applicability and the value that the proposed metadata mechanism brings to supporting the data actions in a Data Lake, and in particular the retrieval of data, let us use once again use the example of the sources given earlier. As previously mentioned, the selected data sources are distributed to the specific Data Lake pond for further processing according to the corresponding at-tribute values. After the completion of the selection process, the retrieval process is based on the metadata DLB metadata history – RDF schema of the Data Lake encoded in the blueprints. Let us now assume that after the selection process Source 1 is a member of the pond with structured data and that Source 3 is also a member of the pond with semi-structured data as presented in Figure 13.

If the Data Lake owner wishes to retrieve all the energy production data from our Data Lake by the Visual Querying environment in the application layer of a system that uses the Data Lake, then in the simplest case all that needs to be done is to create the following SPARQL query through the Visual Querying environment (see Figure 17):

```
SELECT ? Dlsources
        WHERE {
? source <has keyword> #Energy production
        }
```

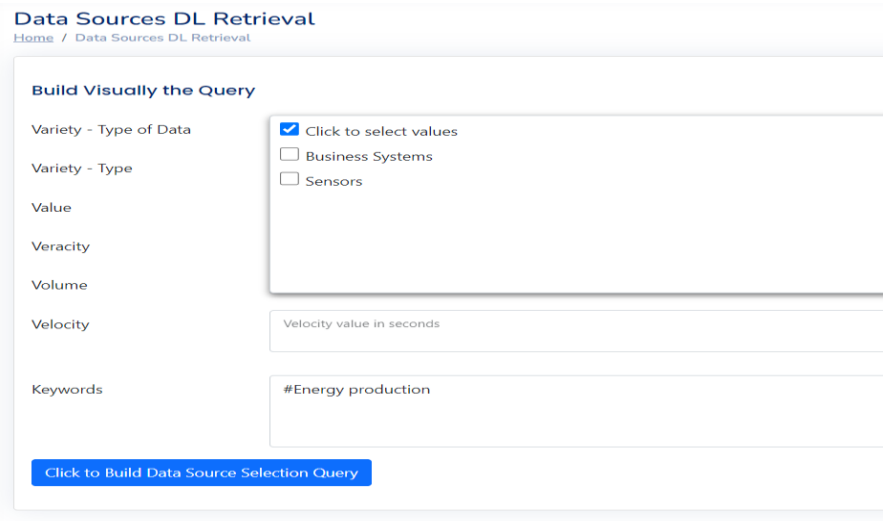

Figure 17. Visual Querying environment source selection (b)

As shown in Figure 13, the Visual Querying environment is fed with DLB metadata history, so Data Lake owners can query Data Lakes using only the attributes in the metadata history.

In essence, Visual Querying essentially retrieves data from the Data Lake and pushes all data sources with the Product Delivery keyword to the application layer. This results in a large volume of data and a greater complexity of filtering after retrieval. Using the metadata provided by the semantic enrichment mechanism, we can refine the type of information sought in the Data Lake and get the results we need through the Visual Querying environment. Of course, more guided queries can be built using such DLB metadata history existing attributes. These guided queries from the Visual Querying environment can range from simple to more sophisticated by utilizing partly or fully the spectrum of the 5Vs data characteristics. Thus, they allow not only data scientists but also Data Lake owner without IT knowledge to derive more value from their data and to define custom levels of granularity and refined information in the data sought as required. Essentially, this SPARQL query process and the associated characterization support the handling and management of multiple and diverse types of data sources residing in a Data Lake in a simple yet efficient way. Finally, this latter ensures and vests in the owner ownership of the source by creating NFT data sources after selected to be part of the Data Lake.

7.3 Preliminary Validation

As presented in Chapter 4, Data Lake metadata systems should provide the following characteristics as a minimum: Semantic Enrichment (SE), Data Indexing (DI), Link generation and conservation (LG), Data Polymorphism (DP), Data Versioning (DV) and Usage Tracking (UT) (Sawadogo & Darmont, 2029). Below we provide a short description for each characteristic:

A SE process uses knowledge bases, such as ontologies, to describe the context of data (e.g., tags). Using semantic enrichment, data from the Data Lake is summarized and linked so that it can be understood and identified. Linking data can be done, for instance, when the tags are the same. This characteristic is met by our mechanism similarly as presented in Chapter 4 , 5 and 6 using both the dynamic and stable blueprints provided in Figures 3, 6 and 12.

As described by Sawadogo and Darmont, 2019, the second major functionality is DI, which involves setting up a data structure based on keywords or patterns for retrieving datasets. Data Lake querying is optimized through keyword filtering using this feature. As illustrated, our metadata semantic enrichment mechanism offers this characteristic via the attribute Variety - type of data, which is used to distribute data sources and data ponds based on their structure and the keyword attribute within the stable blueprint.

The LG characteristic is defined as the procedure to identify data clusters, that is, groups of data strongly connected and substantially different from each other, by detecting similarity relationships or integrating pre-existing links. The keyword properties in the dynamic blueprint, which are modified each time a new data source or new data produced by a registered source are posted to the Data Lake, are the main instruments describing how our system implements this capability.

DV refers to the capability of the metadata system to handle data changes during processing in the Data Lake, while DP refers to the storage of various representations of the same data. When sources in the Data Lake alter or produce new data, our metadata se-mantic enrichment technique provides these functional qualities by preserving the metadata description - blueprint. A new time-stamped representation of the newly pushed data is built and stored in the Data Lake, alongside pre-existing representations and blueprints. The suggested approach changes the dynamic blueprint, particularly the keywords, as needed, during the data processing in the Data Lake.

Finally, UT, which is the process of documenting user interactions with the Data Lake, is the final functionality mentioned in Chapter 4 and Chapter 6 and is also provided by the mechanism and its extension reported in this chapter. In essence, the dynamic blueprint records the timestamp and user information associated with the most recent query whenever data is accessed, and this information is also added in the DLB metadata history.

In addition, Sawadogo & Darmont, 2019 compares 15 metadata systems in a synthetic way. Using the set of functional characteristics introduced in Chapter 4 and some new extra characteristics that were added in this chapter also reported in Chapter 5, we compare CoreKG and MEDAL as in Chapter 4, the two most fully developed systems examined in Sawadogo & Darmont, 2019 and in Chapter 6, with our extended metadata data mechanism. Data Lake metadata quality and efficiency can be assessed synthetically based on these extra characteristics, which are:

- Granularity (GR) [Chapter 4]
- Ease of storing/retrieval (ESR) [Chapter 4]
- Size and type of metadata (STM) [Chapter 4]
- Expandability (EX) [Chapter 4]
- Security and Ownership (SO)
- Process Mining readiness (PR) [Chapter 5]

Granularity is defined in Chapter 4 as the ability to refine what type of information is necessary to retrieve, for example, using keywords. Information sought can be defined at a variety of fine-grained levels using the metadata mechanism.

Ease of storing/retrieval describes the ease of storing or retrieving data in the Data Lake using the metadata mechanism. Retrieval action is assumed to be efficient enough to return the desired parts of the information. For the process of storing and retrieving data items, this characteristic is reflected in the number of steps required.

The Size and type of metadata are defined as the volume and type of metadata that the mechanism produces, needed for efficient and accurate retrieval. In general, the greater the size and/or the complexity of the type of data required, the less efficient and suitable the metadata mechanism is.

Expandability refers to the ability to add additional functional features and other approaches to the metadata mechanism. As expected, the better the expanding mechanism, the more open it must be.

Security and Ownership refers to the ability of the Data Lake mechanism to ensure that the data in the Data Lake has not been modified and to enable the sharing of data contained in Data Lake s with verified owners.

Finally, Process Mining Readiness as presented in Chapter 5 is measured by the number of steps that are needed after the query is executed to provide data suitable for process mining. Using a Likert linguistic scale with the values Low, Medium, and High, these characteristics are assessed in this chapter as defined in Table 11.

As previously mentioned, we use the suggested characteristics to provide a short comparison between our extended metadata enrichment mechanism proposed in this chapter with the two top existing metadata mechanisms suggested by Sawadogo &

Darmont, 2019, that is, MEDAL and CoreKG, as well as DLMetachain presented in Chapter 6.

Table 11. Definition of Low, Medium, High of each characteristic

Characteristic	Low	Medium	High
Granularity	1 Level	2 Levels	3 or more Levels
Ease of storing and retrieval	5 or more Actions	3-4 Actions	2 Actions max
Size of metadata	KB	MB	GB
Expandability	No	Normal	Limited
Security & Ownership	None	One of them	Both
Process Mining Readiness	5-4 Actions	3-2 Actions	1 Action max

MEDAL follows a graph-based organization based on the idea of an object and a typology of metadata falling in three categories: intra-object, inter-object, and global metadata. It corresponds to a representation and version of an object contained within the hypernode. The nodes in MEDAL are further connected by oriented edges that perform transformations and updates. Several ways can be used to link hypernodes of the mechanism, such as edges for modeling similarity relationships and hyperarcs for translating parenthood relationships. In addition, global resources are present in the form of knowledge bases, indexes, or event logs. SE, DI, LG, DP, DV, and UT are provided by this concept and operation of the framework. MEDAL can be described as having High GR, Medium ESR using Indexes and Event Logs, Medium STM, and an Undefined EX, SO & PR since no reference was made related to these characteristics.

A complete Data and Knowledge Lake Service is offered by CoreKG, which provides researchers and developers with a single REST API for organizing, curating, indexing and querying data and metadata. Using CoreKG's Data Lake service, Web developers can produce data-driven applications using relational and NoSQL databases. Within the Data Lake, these datasets can be created, read, updated, and deleted, and federated search can be applied on top of various islands of data. As well as providing built-in capabilities for authenticating, controlling, and encrypting data, it enhances traceability and provenance. Data Lake 's raw data is curated and prepared for analysis by CoreKG on top of the Data Lake layer. Extractions, summaries, enrichments, linking, and classifications are included

in this layer. The concept of Knowledge Lake is another component of this mechanism. It is a centralized repository of virtually inexhaustible amounts of raw data, as well as con-textualized data. As a result, it provides the foundation for Big Data analytics by automatically curating raw data into data islands, which can provide insights from vast amounts of local, external, and open data that are constantly growing. This service is open-source and offers SE, DI, LG, DP, and UT. According to the suggested property scheme, CoreKG can be assessed as having High GR, Medium ESR utilizing the single API, with Medium STM, and High EXP because it utilizes the Hadoop ecosystem, Medium SO provided by the aforementioned built-in capabilities, and Undefined PR.

The metadata enrichment mechanism of DLMetachain provides DI, LG, DP, DV and UT, which is an extension of the mechanism presented in Chapter 4. Both the proposed mechanism of this chapter and DLMetachain offer High GR and High ESR, both using the Stable and Dynamic DSB, with Medium STP, and High EXP. These values are attributed as follows: The use of keywords that specify the sources and the values of the blueprints provide High GR. This gives the user the ability to define specific levels of the properties these keywords offer and the type of blueprint characteristics for which values are preserved. The features set provided by both mechanisms may be regarded as being extremely comprehensive for enabling data retrieval based on precise query-like information. The blueprint description of the Data Lake achieves the High ESR because each time data sources are pushed to the Data Lake, a variety of types of data attributes are produced, aiding the mechanisms' placement of the sources to a particular pond based on the structure of the data involved (structured, semi-structured and unstructured). The Data Lake's source distribution by both mechanisms makes it straightforward and convenient to store and retrieve information. Both systems are distinguished by the Low number of actions to select and query data sources in accordance with Stable and Dynamic blueprint and push data into particular Data Lake Ponds. Due to the creation of the metadata description of the Data Lake each time new sources or data are pushed to the Data Lake, as well as the DV characteristic that the blueprint provides in both mechanisms, the STM produced by the mechanisms has the maxi-mum value. Additionally, because both of Data Lake implementation approaches are built on the Hadoop ecosystem, they offer High EXP. It should be noted here that EXP of the mechanism proposed in this chapter can be traced in the application of visual querying

using the simple semantic enrichment and blueprint ontologies during the source selection or data retrieval.

Table 12. Evaluation and comparison of the mechanisms

Mechanism /Characteristic	GR	ESR	STM	EXP	SO	PR
MEDAL	High	Medium	Medium	Medium	Undefined	Undefined
CoreKG	High	Medium	Medium	High	Medium	Undefined
DLMetachain	High	High	Medium	High	Medium	Undefined
Proposed approach	High	High	Medium	High	High	Undefined

Both mechanisms aim to enhance the privacy, security, and data governance of Data Lakes and, as a result, deal with some of the major challenges encountered in Data Lakes.

By storing the descriptive metadata information on the Blockchain, DLMetachain satisfies the SO requirement. This allows for the storage of encrypted metadata information and ensures the immutability of the metadata. This functionality provided by this mechanism offers Medium SO as Ownership is not guaranteed in contrast to the mechanism of this chapter that provides also this characteristic offering High SO. Finally, none of the mechanisms refers to the ability to prepare the data for process mining, but this can be provided by extending the data blueprints.

The information from the brief qualitative comparison between the four mechanisms made in this section is summarized in Table 12. It is evident that the proposed mechanism appears to function better than MEDAL and CoreKG in various characteristics, and that is equal in performance with DLMetachain, while outperforming the latter in terms of the SO feature.

7.4 Summary

This chapter was involved with a major challenge in Data Lakes, namely the standardization of the processes for storing/retrieving data generated by heterogeneous sources, In this context, it proposed a novel framework for managing data inputted to or outputted from a Data Lake organized with ponds architecture. The proposed framework relies on a semantic enrichment mechanism that utilizes blueprints comprising a set of properties in the form of metadata. This mechanism essentially produces and organizes meta-information describing a data source that will be included in a Data Lake. The meta-

information is divided into two categories, each being realized by a dedicated blueprint, which are structured around the 5Vs Big Data characteristics Volume, Velocity, Variety, Veracity and Value: The first blueprint includes static information, that is, information that does not change over time, such as, the name of the source and its velocity of data production. The second encloses descriptors that vary with time as data is produced by the source, such as the volume and date/time of production.

Each time a new data source or a new piece of data are pushed in or out of a Data Lake, the properties in the abovementioned stable and dynamic blueprints are updated and transactions are recorded for historical purposes. The description of the sources offered by the blueprints essentially supports the management of many, multiple, and different types of data sources by contributing to enriching a Data Lakes' metadata information before and after these sources become members. In case a new data source becomes part of the Data Lake, the metadata schema is used to update the description of the whole Data Lake ontology. Therefore, filtering and retrieving data relies solely on this metadata mechanism, mainly utilizing properties and descriptors based on the 5Vs, such as last source up-dates and keywords.

Two new significant features have been added in the proposed mechanism compared to previous work on the topic. The first revolves around making the approach more usable and easier to learn by using visual querying when selecting data sources. The second deals with security and integrates the mechanism with NFTs and Blockchain to claim and record ownership of the data sources.

A short evaluation cycle was performed by comparing qualitatively this approach to other existing metadata systems. The results indicated that there is high potential of our approach as it provides a complete and thorough way to characterize the data sources including a set of key properties usually met in literature which is further expanded in this chapter. Finally, the evaluation proved that the proposed approach offers the required tools for efficient and fast retrieval of the information sought as the evaluation also proved in Chapters 4, 5 and 6.

As presented in previous chapters, the proposed frameworks for managing data within Data Lakes through a semantic enrichment mechanism offers an innovative approach to organizing and retrieving data from diverse sources. These frameworks, through its use of blueprints and metadata characteristics, facilitates the efficient management and security of data, ensuring that metadata is consistently updated as new sources are

integrated. Such a structured system not only streamlines data processing within Data Lakes but also enhances the ability to track and retrieve specific datasets, irrespective of their origin or frequency of change.

However, the landscape of data management does not stop at Data Lakes. The rapid evolution of Big Data, driven by technological advancements like the Internet of Things (IoT), has catalysed the development of new storage and processing paradigms. As we step into the next phase of data architecture, the focus shifts towards more decentralized models, such as Data Meshes and Data Markets, which build upon the foundational principles laid by Data Lakes. This chapter delves into these next-generation architectures, exploring how they expand the possibilities for data storage and analysis while addressing the increasing complexity and volume of data.

In the evolving landscape of data management, while our blueprint-based metadata framework has significantly enhanced the organization and retrieval of data within both Data Lakes, challenges related to data consistency and integration persist. The increasing heterogeneity of data sources not only demands robust metadata systems but also calls for sophisticated techniques to reconcile discrepancies and duplicates across diverse datasets

By transforming Data Lakes into Data Meshes, could unlock the full potential of Big Data, enabling scalable and flexible systems that provide rapid access to valuable insights. The shift from monolithic platforms to decentralized models highlights the ongoing need for advanced metadata management systems, such as the blueprint-based approach discussed earlier, which continues to play a pivotal role in organizing, retrieving, and deriving insights from vast datasets. Chapter 8 and subsequent chapters investigate these aspects in greater depth.

CHAPTER 8 : TRANSFORMING DATA LAKES INTO DATA MESHES USING SEMANTIC DATA

8.1 Introduction

This thesis has already highlighted the significance of Big Data in our daily lives. In today's data-driven world, Big Data pervades every facet of our digital existence, while it is omnipresent and indispensable for producing insights that shape our world. It is the ubiquitous force driving innovation, analytics, and informed decision-making across diverse domains (Awan et al., 2021).

As the concept of Big Data has evolved so rapidly, there has been some confusion regarding how it should be explained; this has led to a divergence in terminology between “what Big Data is” and “what Big Data does”. This evolving landscape underscores the challenges associated with comprehensively defining and understanding the multifaceted roles and functionalities of Big Data (Machado et al., 2022).

As previously mentioned, the exponential growth in volume, variety, and complexity of data has necessitated the evolution of storage architectures to effectively manage and harness this wealth of information. Traditional storage solutions are often equipped to handle the diverse nature of modern data, which includes unstructured and semi-structured formats alongside conventional structured data. To address these challenges, innovative storage paradigms such as Data Lakes and Data Meshes were introduced and utilized in previous chapters, while Data Markets have emerged as indispensable components of data infrastructure. Recall that Data Lakes act as expansive repositories capable of accommodating vast amounts of raw, unprocessed data in its native form, while Data Meshes enable organizations to distribute and decentralize data processing, promoting scalability and flexibility. A Data Market is an organized and structured platform or ecosystem where data is treated as a tradable commodity, enabling the buying

and selling of datasets, information, or insights. These architectures empower businesses to derive insights from a broader spectrum of data types, fostering a more holistic and dynamic approach to data storage and analysis in the era of Big Data.

Data Lakes were first introduced as storage architectures well-suited for handling Big Data and guiding organizations towards a data-driven approach. Current research indicates a shift towards decentralized data exchange architectures, like Data Markets and Data Meshes (Driessen et al., 2021). Specifically, Data Meshes aim to overcome certain limitations associated with monolithic data platforms like Data Lakes (Dehghani, 2019). The development of effective data products of Data Meshes imposes demands on metadata templates, which are currently not adequately addressed by existing methodologies.

This chapter deals with transforming a Data Lake into a Data Mesh enjoying the benefits of rapidly storing high frequency data (Data Lake) and constructing on-demand portions of information in the form of a Data Mesh. The proposed approach builds on the notion of data blueprints that aim at semantically annotating data before storing it in the Data Lake. This metadata semantic enrichment guides the process for locating, retrieving and ultimately constructing the Data Mesh easily and quickly according to user needs. The approach is demonstrated using two case studies. The first is involved with real-world manufacturing data collected by a prominent local industrial entity in Cyprus in the area of poultry farming, meat production and trading, namely Paradisiotis Group (PARG). The second uses data obtained from the Europeana Digital Heritage Library (EDGL) and concerns cultural artifacts published and accessed by the public. The data from the two case studies is stored in a dedicated Data Lake utilizing the proposed semantic metadata enrichment mechanism. Subsequently, the corresponding Data Meshes are produced centered around diverse data products. Performance is then evaluated by varying the complexity of the Data Mesh levels constructed based on the granularity of information sought, and the number of data sources involved.

8.2 Methodology

A novel standardization framework was previously introduced in this thesis and is further extended here aiming at transforming a Data Lake into a Data Mesh using Semantic Data Blueprints as presented in Figure 18. The framework leverages standardized data descriptions in the form of blueprints, employing a domain-driven approach to generate Data Mesh products.

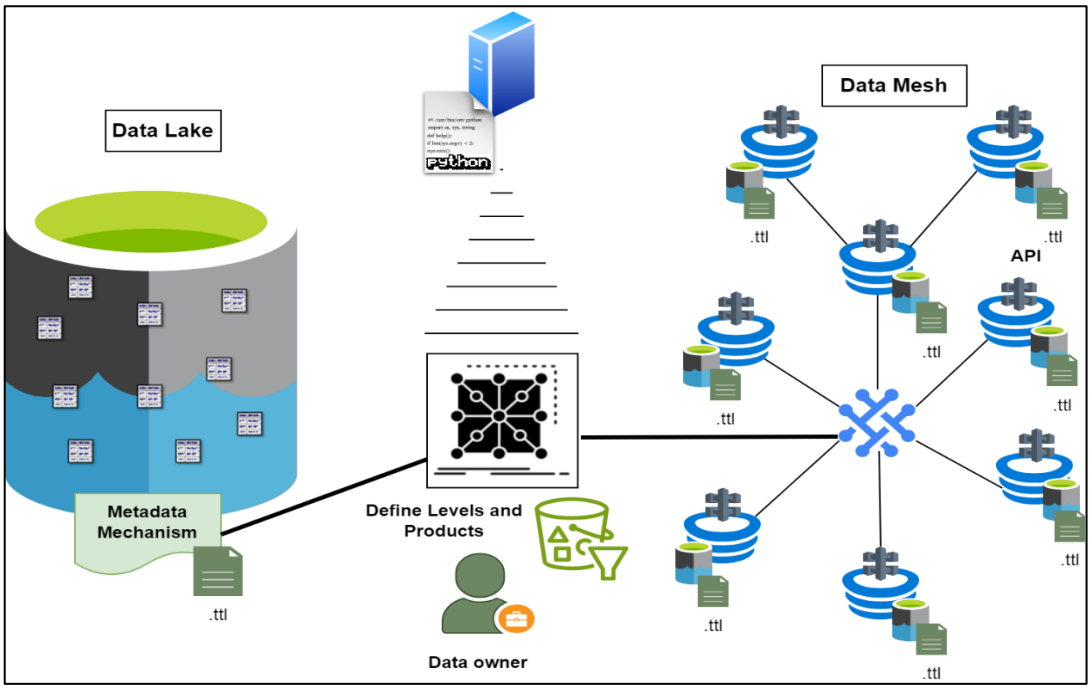

Figure 18. The architecture of proposed framework utilizing semantic data blueprints

A typical Data Lake is employed here which is further enhanced with a metadata mechanism that essentially describes the sources producing the data residing in the Data Lake. This metadata plays a crucial role in providing the transformation of the Data Lake to a Data Mesh and is constructed using TTL files, the latter referring to Turtle, a widely used serialization format for RDF data. The TTL files, through the use of the Turtle syntax, enable the creation of structured and semantically rich metadata within the Data Lake. This enhances comprehensibility and accessibility of the data by offering a standardized and machine-readable representation of the metadata, facilitating efficient data management and utilization within the Data Lake environment.

The ability to create Data Products and Data Domains while transforming a Data Lake into a Data Mesh is based on a dedicated form of blueprint as presented in the Data Lake

metadata description examples in GitHub link (https://github.com/mfpingos/ENASE2024), which provides examples of source descriptions within a TTL file that correspond to data produced from the PARG factory and EDGL.

PARG, as mentioned earlier, is a prominent local industrial entity, recognized as one of the key players and leading authorities in the domain of poultry farming and the trade of poultry meat in Cyprus. The company provides an extensive range of top-notch products designed to cater to the contemporary consumer's preferences for convenient cooking and healthy dietary choices. EDGL is a well-known cultural heritage website that provides access to cultural heritage materials, such as libraries, museums, archives, and other cultural institutions across the continent. The European Commission launched the Europeana platform in 2008 to increase public access to Europe's cultural heritage. Millions of digital artifacts, including books, artworks, pictures, manuscripts, maps, sound recordings, and archive documents, are available on this platform. The Europeana website offers users the ability to search for and view cultural goods having free access to them. Virtual exhibitions, educational resources, and APIs for developers are just a few of the additional tools and services that Europeana provides to assist users in exploring and interacting with the materials. The manufacturing data and business processes are confidential, and the digital heritage items are protected by intellectual property rights. Therefore, the present work made every effort to preserve data confidentiality where necessary. In the case of the PARG factory, the data underwent masking or downgrading to ensure anonymity and business confidentiality during demonstrations and when sharing descriptions. In the case of EDGL, synthetic data was generated using the existing metadata descriptions that are already available on the website. Despite applying the above measures, the provided case studies can sufficiently illustrate the fundamental principles of the proposed framework, demonstrating its applicability and usefulness as described above. It should also be noted that the cases were selected so as to demonstrate the wide range of application domains that the framework may support.

As mentioned above, a Data Lake was constructed using the metadata mechanism for semantic annotation of the sources. In order to transform the Data Lake into a Data Mesh as presented in Figure 18, a dedicated middleware was developed which has been installed on a server and is being fed with user/owner preferences. In essence, the owner defines the Data Meshes' products, and, hence, the levels of the Data Mesh according to business needs. For example, as can be observed in Europeana's website, each registered

digital item is characterized with a specific metadata structure (examples can be accessed on GitHub using link provided earlier). For demonstration purposes, we selected the following eight significant metadata characteristics:

- Century
- Providing Institution
- Type of object
- Subject
- Identifier
- Places
- Format
- Providing Country

Let us now assume that, based on the previously mentioned description, the data owner intends to implement a two-level hierarchical structure within the Data Mesh architecture. The two selected dimensions for this organization are Century and Providing Country, which are deemed the most relevant characteristics for indexing and accessing the data.

To illustrate this, we refer to a sample Data Lake represented in a TTL (Turtle) file, which contains metadata about a collection of items originating from the 19th and 20th centuries. These items have been contributed by two specific countries, Germany and France. These characteristics, temporal (Century) and geographical (Providing Country), form the basis of the two-level classification in the Data Mesh.

Accordingly, the metadata that describes this dataset is structured to align with the Data Mesh schema and is pushed to the middleware layer. The resulting metadata configuration, organized by Century and Providing Country, is depicted in Figure 18, which visualizes how the data is logically partitioned within the Mesh to reflect the owner's preferred access and governance model.

In essence, every part of the Data Mesh is a Data Lake portion described by the relevant metadata according to the levels defined by the user. The selected level attributes are sourced by the cultural heritage metadata characteristics of the Data Lake. These are treated as the components of the Data Mesh architecture providing the ability to create Domains according to selected attributes expressed via the data blueprint mechanism introduced.

The next section demonstrates the applicability and effectiveness of the proposed framework, which is also evaluated by converting the initial Data Lake into a PARG and EDGL metadata description. Note that the metadata mechanism describes the characteristics of the sources defining also the location of each source in the Data Lake. Finally, the framework is assessed by executing and comparing the performance of queries based on the Data Mesh level and using the metadata mechanism directly on the Data Lake.

8.3 Preliminary Evaluation

8.3.1 Design of experiments

The experiments conducted had a dual objective. Firstly, to assess the capability of the proposed approach in generating a Data Mesh and refined levels through the utilization of Semantic Data Blueprints. Secondly, the experiments aimed to evaluate the performance and effectiveness of the approach in terms of granularity. To fulfil these objectives, a series of experiments were carried out, and this section provides an explanation of the rationale behind their design.

Data from the two different application areas mentioned earlier, smart manufacturing (PARG) and cultural digital heritage (EDGL), was utilized for the execution of the experiments. As a starting point, a Data Lake metadata mechanism was built for each area (uploaded also on Github). The Data Lake metadata was described with a TTL file which contains the characteristics for each source that stored data in the Data Lake.

As mentioned in the previous sections, Python scripts automatically create the TTL files while also masking sensitive data. The growth in the number of sources directly impacts (increases proportionally to) the size of the respective TTL file, a crucial element parsed to extract sources that match a query. As an illustration, in the PARG case a TTL file describing 100 sources resulted in a size of 0.077 MB, 1000 sources produced 0.769 MB, 10000 sources yielded 7.5 MB, and 100000 sources led to a file size of 75.9 MB. In the case of EDGL, 100 sources in a file resulted in a size of 0.103 MB, 1000 sources amounted to 1 MB, 10000 sources equated to 10.1 MB, and 100000 sources reached 101.7 MB. However, it must be noted that despite the similar number of sources in the Data Lake example for each application area, there is a variance of the respective file sizes.

This difference arises because the EDGL sources' description includes more attributes. Specifically, EDGL sources are described with 20 attributes, whereas PARG sources are described with 15 attributes (also indicated when following the GitHub link). The size of the initial Data Lake metadata characteristics and the number of attributes represent another aspect explored in the experiments.

The experiments were conducted on a server computer comprising three Virtual Machines. The CPU configuration consisted of 4 dedicated cores, while the underlying server hosting these machines featured a total of 48 cores. The memory size allocated was 8192MB, and the hard disk capacity stood at 80GB. The software stack employed for the experiments included Hadoop (version 3.3.6) for distributed computing, Python (version 2.7.5) for scripting purposes, data generation was based on raw real-world data from PARG and EDGL, and the creation of data products was performed at the desired Data Mesh level. Additionally, Apache Jena was utilized for SPARQL query processing.

Two queries were constructed and executed using all Data Lake descriptions sizes and all Data Mesh levels produced. The first query (Query1.sparql) was executed on PARG and selected values for variables *flockid, source_name,* and *source_path,* where the RDF triples matched a set of conditions. The conditions included the accuracy of sensors being "Medium", the location "Limassol", the data variety "Structured", the velocity "Hourly", the flock size being "Low", and the year "2020". These criteria indicate a focus on data related to a specific context, pertaining to sensor information associated with a flock, with additional constraints on the geographical location, data characteristics, temporal aspects, and other specific attributes.

The second SPARQL query (Query2.sparql) executed on EDGL metadata was formulated also to extract specific information from the TTL file based on specified conditions. In essence, the query selected values for variables: *providing institution, source_path* and *provider collection name*. The conditions set for retrieval included the following criteria and values: Language "De" (German), data variety "Unstructured", format "audio/mp3", type of object "3D", rights associated with the "Creative Commons" license and a thematic association with "Manuscript".

The queries were structured to retrieve and display relevant data that met the aforementioned defined criteria, both with the same complexity in order to be comparable. Note that the queries were executed to the TTL file of the last (maximum) data product level provided by the corresponding Data Mesh structure.

Table 13. Experimentation Data Meshes levels for the PARG and EDGL

	PARG	**EDGL**
Level 2	Location, Variety	variety, theme
Level 3	Location, Variety, Velocity	variety, language, theme
Level 4	Location, Variety, Velocity, Flock_size	variety, language, format, theme
Level 5	Location, Variety, Velocity, Flock_size, Year	variety, language, format, rights, theme
Level 6	Location, Variety, Velocity, Flock_size, Year, Sensors_Accuracy	variety, language, format, type_of_object, rights, theme

Table 14. Average creation time (after 100 iterations) for each Data Lake level for each application area used for experimentation with varying number of sources

Levels Produced	**Number of Sources**	**PARG Data Meshes Average Creation Time (ms)**	**EDGL Data Meshes Average Creation Time (ms)**
DM Level 2	100	82.83	91.58
	1000	176.07	183.12
	10000	1073.80	1079.21
	100000	10390.86	10707.93
DM Level 3	100	108.89	102.53
	1000	238.55	267.30
	10000	1637.10	1633.11
	100000	15701.27	15286.41
DM Level 4	100	144.09	121.14
	1000	345.66	393.52
	10000	2217.61	2238.27
	100000	21604.65	21197.70
DM Level 5	100	155.38	185.63
	1000	504.86	632.46
	10000	2960.53	3653.30
	100000	27418.77	28000.31
DM Level 6	100	174.07	215.09
	1000	689.00	942.35
	10000	3882.33	5476.14
	100000	34536.87	40230.91

8.3.2 Experimental Results

Figure 19 illustrates the time required for constructing the Data Mesh levels in each application domain as presented in Tables 13 and 14. Data Lakes with varying metadata size and number of data sources were transformed into Data Meshes with diverse granularity levels. As indicated in the case studies, the transformation time evolved proportionally to both the number of sources and of the attributes characterizing those sources.

Figure 19. Time performance for constructing Data Mesh products with different numbers of data sources for the two use-cases

The traditional centralized Data Lake architecture presented in earlier chapters and the approaches in Chapters 4 and 5 served as the baseline for evaluating the proposed Data Mesh approach. Existing literature on transforming Data Lakes into Data Meshes has primarily focused on the conceptual and methodological aspects of the transformation, rather than performance metrics such as execution time. As this is an emerging field, there are still few published works offering quantitative comparisons or established benchmarks.

The creation time for Data Meshes with 2 levels was minimal, and this time steadily grew as more data products (granularity levels) were generated based on data owner requirements, as expected. The construction time for Data Meshes with the maximum level (6 data products) was significantly higher in the two examples compared to lower levels, exhibiting an average increase between 3 and 10 times as the number of data sources was increased for the same number of Data Mesh levels created in both case-studies.

It is noteworthy that the maximum construction time for Data Meshes was less than 0.6 minutes for PARG Data Lake metadata and less than 0.7 minutes for EDGL metadata. This can be regarded as a quite satisfactory performance, especially considering the extreme conditions tested with values reaching 100,000 for the sources and 6 for the granularity levels that are, in practice, very rare to encounter. This also indicates that the number of attributes describing the sources affects the construction time, as expected, because it increases the TTL file size.

The two benchmark SPARQL queries were executed 100 times each using PARG's and EDGL's metadata and different Data Lake and Data Mesh structures produced by varying the number of sources, and hence the metadata in the TTL files, to facilitate a comprehensive comparative analysis. Table 15 and Figure 20 present the query execution times in milliseconds, accompanied by the corresponding number of sources. To ensure a standardized comparison, the queries were configured to yield an identical number of sources at each level.

Notably, the observed trend reveals a direct correlation between query execution time and the aggregate number of sources returned. Specifically, as the granularity increases (i.e. the number of Data Meshes' levels), there is a discernible decrease in query execution time. This observation highlights a significant advantage inherent in employing a Data Mesh structure utilizing Semantic Data Blueprints, that is, the capacity to confine information within designated data product levels thereby facilitating immediate and efficient data retrieval.

Finally, it is evident that maximizing the granularity, if needed, in constructing Data Mesh proves beneficial. This becomes particularly apparent when comparing the execution time of a query on a Data Lake with 100,000 sources, as an extreme scenario, against a Data Mesh Level 6 with the same number of sources, with both returning 59 sources that satisfy the query for PARG and 8 for EDGL. The query execution time was observed to be 18 times faster in the latter case for PARG and 26,5 times faster for the EDGL.

8.4 Summary

This chapter explored the conversion of a Data Lake into a Data Mesh, leveraging the advantages of efficiently storing high-frequency data (Data Lake) and constructing

specific information segments as Data Mesh products. The proposed approach was based on the concept of data blueprints, which involve semantically annotating data before storing it in the Data Lake. This semantic enrichment of metadata guided the process of locating, retrieving, and swiftly constructing Data Mesh based on user requirements. The approach was exemplified through two case studies.

Table 15. SPARQL queries average execution time (after 100 iterations) and number of sources produced for each query in each application area

Levels	**Number of Sources**	**Average PARG Query Execution time in ms (Number of sources returned)**	**Average PARG Query Execution time in ms (Number of sources returned)**
Data Lake	100	975 (5)	973(4)
	1000	1221 (4)	1197(4)
	10000	2396 (7)	2833(5)
	100000	18122 (59)	26298(8)
DM Level 2	100	912 (5)	879(4)
	1000	951 (4)	910(4)
	10000	1148(7)	1010(5)
	100000	1975 (59)	1421(8)
DM Level 3	100	894 (5)	873(4)
	1000	918 (4)	889(4)
	10000	1007 (7)	923(5)
	100000	1360 (59)	1057(8)
DM Level 4	100	898 (5)	879(4)
	1000	903 (4)	874(4)
	10000	963 (7)	909(5)
	100000	1137 (59)	962(8)
DM Level 5	100	894 (5)	875(4)
	1000	896 (4)	874(4)
	10000	916 (7)	887(5)
	100000	1006 (59)	919(8)
DM Level 6	100	896 (5)	885(4)
	1000	888 (4)	878(4)
	10000	911 (7)	890(5)
	100000	991 (59)	990(8)

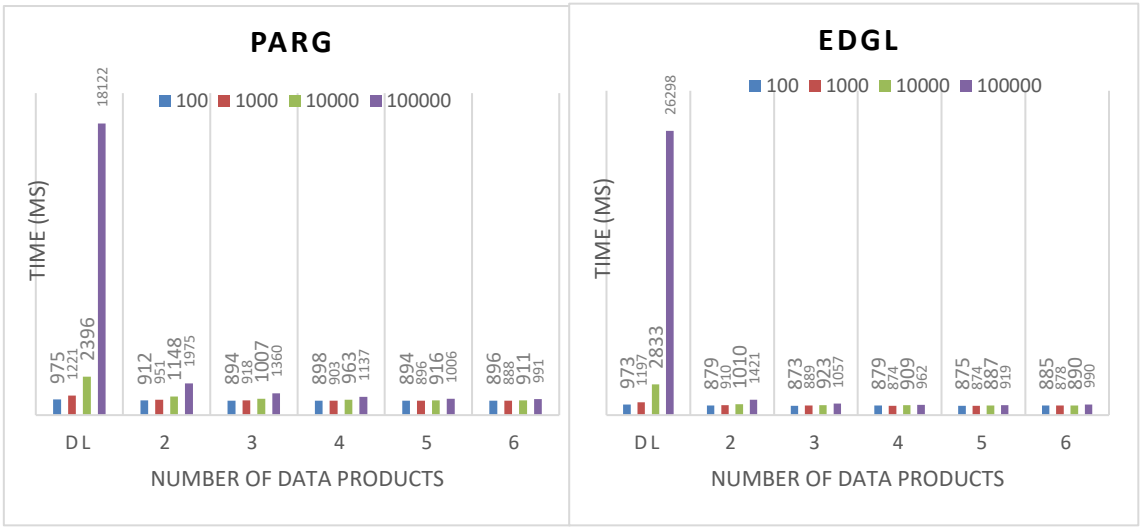

Figure 20. Time performance for executing queries on Data Meshes with varying number of levels and data sources for the two use-cases

The first employed real-world manufacturing data from the Paradisiotis Group of Companies (PARG), a prominent local industrial entity in Cyprus focusing on poultry farming and poultry meat product production and trading. The second case study utilized data from the Europeana Digital Heritage Library (EDGL), specifically cultural artifacts published by Europeana and accessed by the public.

The data from both case studies were stored in a dedicated Data Lake using the proposed semantic metadata enrichment mechanism. Subsequently, Data Meshes were generated, centered around various data products defined by the user. The performance was then assessed by varying the complexity of the constructed Data Mesh based on the granularity of information sought and the number of data sources involved.

The target of the conducted experiments was twofold: Firstly, they aimed to evaluate the capability of the proposed approach in generating Data Meshes and refined data products/levels through the application of Semantic Data Blueprints. Secondly, the experiments sought to assess the performance and effectiveness of this approach concerning granularity.

The results obtained were quite satisfactory indicating that transforming a Data Lake to a Data Mesh is fully supported under the proposed semantic enrichment mechanism with limited time requirements, as well as consistent behaviour over varying number of sources residing in the Data Lake and complexity of the queries executed to retrieve these sources.

CHAPTER 9 : DISCOVERING DATA DOMAINS AND PRODUCTS IN DATA MESHES USING SEMANTIC BLUEPRINTS

9.1 Introduction

As presented in the previous chapter, the current literature \ shows a trend towards more decentralized data exchange architectures/structures, such as Data Markets and Data Meshes (Machado & Santos, 2022). The latter two were key targets for many (large) companies and organizations to achieve, which adopted initiatives to facilitate transition from their existing, monolithic data platforms. One of the main challenges for this transition, in addition to the novelty of the concepts, is how to divide up the data landscape into domains and identify data assets that should be turned into data products. These organizational challenges are in fact often perceived to be more daunting than the technical challenges associated with Data Mesh design (Eichler et al., 2021).

The main research contribution of this chapter lies again with the utilization of Semantic Data Blueprints (SDB) for discovering Data Products and Domains in Data Meshes. In addition to the previous chapter, this work offers a dynamic way to transform a Data Lake into a Data Mesh. Unlike the previous chapter, which presented a predifined/static methodology for transforming a Data Lake into a Data Mesh, this work introduces a dynamic transformation approach by producing data products of Data Mesh. This dynamic approach ensures that Data Mesh structures are not only built upon initial metadata definitions but are continuously refined and optimized, supporting greater agility and contextual relevance in data discovery and product generation according business need.

The set of SDB essentially describes properties of data via stable attributes, such as variety, value, velocity, veracity, and attributes that are not stable over time, such as volume, last source update and keywords. The proposed approach as mentioned above builds upon previous work on the topic that introduced a semantic metadata enrichment mechanism for Data Lakes (see Chapters 4, 5 and partly 8) which allows for the efficient storing and retrieval of data belonging to dispersed and heterogenous data sources. The same concepts are extended, modified and adapted in this work to match the characteristics of Data Meshes. A Data Mesh is conceived here as the evolution of a Data Lake in terms of organizing huge volumes of information (i.e., Big Data) expressed in multiple data forms (structured, unstructured and semi-structured), but, most importantly, for tracing this information easily, quickly and efficiently. Although both Data Lakes and Data Meshes can offer the backbone for software analytics with useful insights, a Data Mesh provides a more narrowly focused and domain-centric approach. Users can have more control over their data and improve analytics skills, as well as provide more precise insights for software products and procedures by utilizing the Data Mesh principles (Dehghani, 2024). In this context, we propose a new set of semantic blueprints to facilitate the creation of Data Products through a domain-driven approach, which allows us to retrieve information directly from its stored location. The proposed approach is demonstrated using real-world manufacturing data collected from a major local industrial player in Cyprus, namely Paradisiotis Group (PARG). Performance is then assessed via the construction of Data Meshes based on various data products and the execution of SPARQL queries that vary in complexity, that is, granularity of information sought and number of data sources.

9.2 Methodology

A novel approach for storing and retrieving large amounts of data is proposed here that aims to best serve efficient storing to and retrieval from Data Lakes, while at the same time offering the means to transform dynamically a Data Lake into a Data Mesh when needed. More precisely, the unique metadata mechanism based on SDB established in chapters Chapter 5 and Chapter 6 enables blueprinting to characterize and describe the data sources and data items that are kept in a Data Lake. A novel standardization framework based on this mechanism was also introduced earlier to convert a Data Lake into Data Meshes by discovering Data Products and Domains using Semantic Data

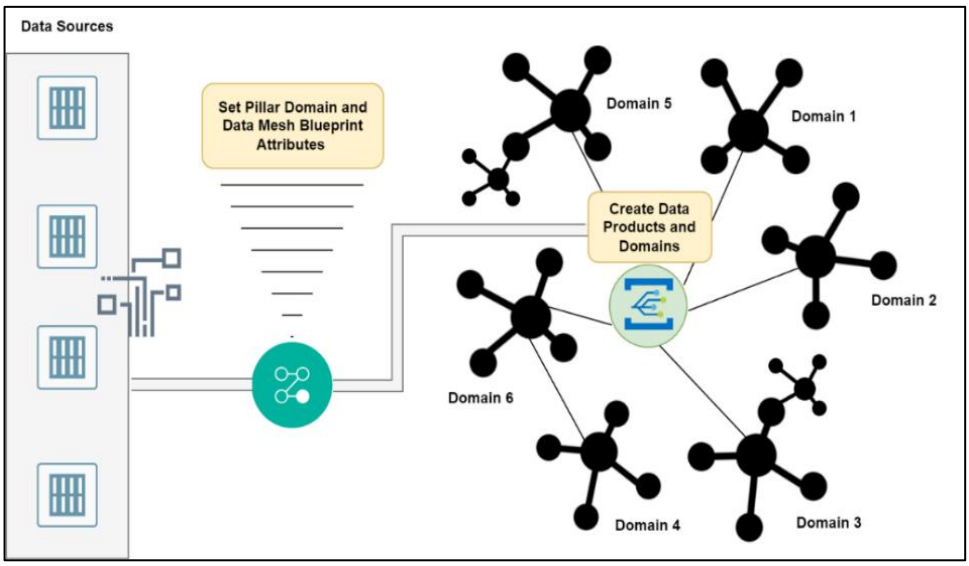

Figure 21. Summary of the proposed Data Mesh architecture

Blueprints (see Figure 21). The framework utilizes standardized descriptions in the form of blueprints to create data products using a domain driven approach. A real-world case-study from the domain of manufacturing is formed to demonstrate the proposed approach.

The data utilized is accessible via https://github.com/mfpingos/TechnologiesMDPI and was collected within the PARG factory. Consequently, this paper discloses only a portion of the processes, providing limited details, and utilizes a masked and downgraded version of the data. Nevertheless, the case study sufficiently illustrates the fundamental principles of the proposed framework, validating its applicability and effectiveness.

The ability to discover Data Products and Domains while creating Data Meshes is based on a dedicated form of blueprint depicted in Figure 22. Actually, this may be regarded as a global blueprint that can be applied to any application domain and type of data, not only in the manufacturing area. Specifically, the blueprint provides a standardized form of describing data constituents and contains as a starting point the Pillar Domain, followed by Subdomains. The Pillar Domain is the key operational attribute or category within the Data Mesh structure. It is the highest level of organization and acts as the starting point for structuring the data in the Data Mesh. Subdomains are more granular categories or attributes that refine and further detail the data organized under the Pillar Domain. They are secondary and tertiary levels of categorization within the Data Mesh architecture and are considered the more granular parts of the Data Mesh. As in Chapter 8, a Terse RDF

Triple Language (TTL) file is created for each level, which is written in XML format and describes the Data Mesh blueprint (see sample code provided in GitHub link https://github.com/mfpingos/TechnologiesMDPI). Using the manufacturing data the way a Data Mesh is constructed will be demonstrated by creating appropriate Data Products and Domains using the Data Mesh blueprint of Figure 22. A dedicate Python script (*finalpyoptfinal.py* file in GitHub) is developed which utilizes the Data Lake metadata enrichment mechanism to create semantic annotation and enrichment and produce data products according to owner/user needs. The sample data originates from PARG's systems operating in different locations of the factory and monitoring or facilitating chicken farming. These systems can be considered as data sources collecting data and managing measurements from various sensors within the facilities of the factory. For example, the Flock Daily files contain daily measurements of a specific poultry farming unit's cycle. A typical farming cycle usually spans from 1 to 60 days. These files include daily battery temperatures, minimum/maximum/required temperatures and humidity measurements, with specific timestamps indicating when the sensor readings were captured/sent. Moving to the Flock Hourly files, these consist of hourly measurements for a particular day of the facilities and provide data on hourly required temperature, temperatures of specific sensors, temperatures outside the facility, as well as measurements of humidity and carbon dioxide levels, all with corresponding timestamps for sensor data transmission. Examples of these data are also uploaded on GitHub (https://github.com/mfpingos/TechnologiesMDPI).

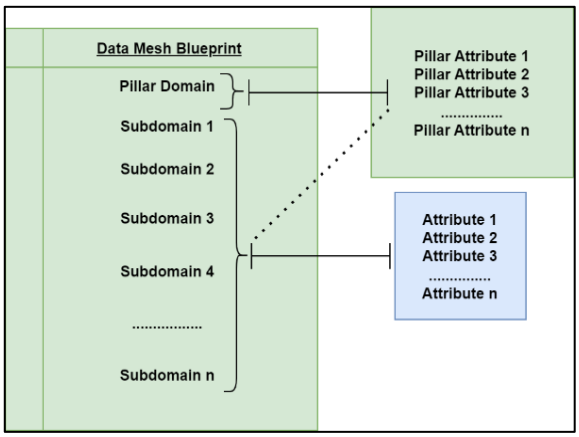

Figure 22. Data Mesh Blueprint

Each data source in the PARG environment is described via an RDF using TTL format. In order to demonstrate the proposed Data Mesh framework, we have selected the following metadata characteristics to describe a source: (i) Source Name; (ii) Location;

(iii) Feed cycle start; (iv) Feed cycle end; (v) Keywords; (vi) Variety; (vii) Velocity; (viii) Volume, and, (ix) Source Path. The corresponding description may be found at https://github.com/mfpingos/TechnologiesMDPI.

While a Data Mesh is a decentralized data architecture that treats data as a product according to business needs, it promotes domain-oriented ownership, with products and sub-products owners being responsible for the quality, discoverability, and usability of the data. Figure 22 shows an example of how the Data Mesh Blueprint and Data Mesh architecture are constructed taking into consideration the metadata characteristics listed above: The Pillar Domain attribute (Location as Level 1), as selected by the Data Mesh owner, constitutes the main part of the Data Mesh structure, while selected Subdomains (Velocity as Level 2 and Variety as Level 3) define the second and third level of refinement in the creation of the data products (structure also presented in Figure 21). The latter are treated as the next components of the Data Mesh architecture providing the ability to create domains according to selected attributes expressed via the blueprint mechanism introduced in Figure 23. Each Level of the Data Mesh consists of a TTL file that includes all the descriptions of the sources which are filtered according to the level. A sample TTL description for Source 1 is presented in Figure 24. Let us now assume that we want to retrieve all the sources in the Data Mesh for the Data Product << Limassol | Daily | Structured >>. The semantic Web framework Apache Jena is fed with the preferred characteristics of the attributes and executes the following SPARQL query:

```
SELECT ?flockid ?source_name ?source_path
    WHERE {?source rdf:type ex:Description ;
ex:flockid ?flockid ;
ex:source_name ?source_name ;
ex:source_path ?source_path ;
ex:location "Limassol" ;
ex:variety "Structured" ;
ex:velocity "Daily" . }
```

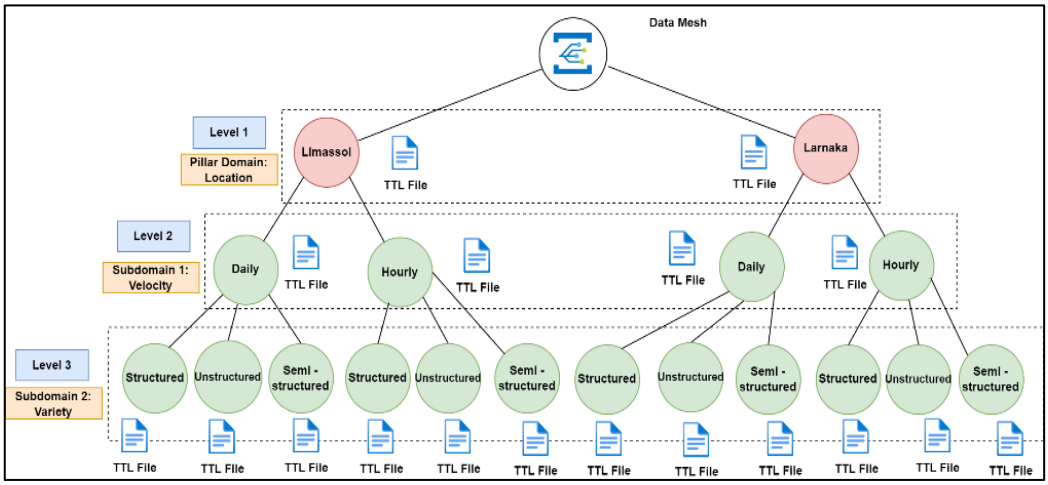

Figure 23. Creation of Data Mesh Domains with PARG data

```
ex:source1
  rdf:type ex:Description ;
  ex:source_name "1_FLOCK_Daily_EXPORT-07-05-2022" ;
  ex:flockid "1" ;
  ex:location "Limassol" ;
  ex:feedcycle_start "2022-05-07" ;
  ex:feedcycle_end "2022-05-20" ;
  ex:keywords "growDay, hour, requiredTemperature, coldTemperatureAlarm, hotTemperatureAlarm,
               sensor1, sensor2, sensor3, sensor4, sensor5, outsideTemp, currentAverageTemp,
               humidity, staticPressure, currentCO2, CO2HourMax, CO2HourMin" ;
  ex:variety "structured" ;
  ex:velocity "Daily" ;
  ex:source_path "hdfs://your-hadoop-namenode:9000/user/sources/daily_flock_1_data" ;
  ex:volume "76 KB" ;.
```

Part of TTL File

Figure 24. Creation of Data Mesh Domains with PARG data – Source 1 TTL description

Figure 25. Detailed workflow to identify Data Products and Domains using SDB

The result of the above query execution consists of the metadata (flock-ID, source name, source path), which satisfies the query parameters (Location: Limassol, Variety: Structured, Velocity: Daily). According to the level at which the SPARQL query is executed, the execution time differs as demonstrated in the experiments section. As we move to including more data products (levels) more fine-grained information is produced and the execution time of the query becomes shorter. Therefore, the proposed Data Mesh architecture offers the ability to treat data as a list of data products according to specific business needs, while the Pillar Domains and Subdomains are defined to reflect these needs via the Data Mesh Blueprint presented in Figure 22.

To sum up, Figure 25 presents how our framework identifies Data Products and Domains using SDB and a series of steps for transforming Data Lake into Data Mesh. The workflow begins by defining the metadata characteristics to describe each data source. In Step 2 the SDBs are developed which serve as a standardized form of describing data constituents and define Pillar Domain and Subdomains. In Step 3 the semantic annotation is created by using SDB to tag and enrich data sources with semantic metadata. This step involves creating RDF descriptions for each data source using a dedicated format. In Step 4, the SDB are utilized to create data products based on the owner's needs. The latter are defined by selecting the attributes that will represent the pillar domain and subdomains. Therefore, these needs are matched with metadata characteristics to return the desired pieces of information stored in the Data Lake in the form of data products. Finally, the Data Mesh architecture is constructed by organizing data products into domains and subdomains. The pillar domain acts as the highest level of information organization, while the subdomains provide secondary and tertiary levels of refinement.

The next section demonstrates the effectiveness of the proposed framework through its evaluation and comparison with other forms of a Data Lake architecture, which are considered rivals or predecessors of Data Meshes.

9.3 Qualitative validation

This section aims to investigate in general the ability of creating data products via comparison between the proposed Data Mesh architecture and the following data structures of storage architectures:

- Traditional Data Lake without the proposed metadata enrichment mechanism.
- Data Lake with a semantic metadata enrichment mechanism (see Chapter 4)

The selection of Data Lakes as the counterapproach serves two purposes: The first is to show the differences between the widely known and used architecture of Data Lakes and Data Meshes. This will provide some indications about whether Data Meshes can be regarded as the natural evolution of Data Lakes in Big Data management. The second, since there is limited work on the topic, is to provide a comparison with the closest approaches, that is, with similar studies that introduced the same concept of semantic enrichment and blueprints. This comparison will enable identifying potential pros and cons of the two approaches.

The following characteristics/metrics were selected to facilitate comparison be-tween the alternative architectures: (i) Data domain Readiness and Alignment; (ii) Granularity; (iii) Decentralization; (iv) Ease of Storing and Retrieval; (v) Agility.

Data Domain Readiness and Alignment refers to the level of preparation of a particular data domain or set of data for analysis or processing. It involves ensuring that the data is accurate, complete, consistent, properly formatted, and related to a specific domain. Once the data domain is deemed ready, a data product may be created and used for various purposes, such as building models, making predictions, generating reports, or creating visualizations. Overall, ensuring data domain readiness is crucial for achieving accurate and meaningful results from business data analysis or processing tasks. Without proper preparation, the data could lead to incorrect or misleading insights and decisions.

Granularity refers to the level of detail at which data is collected, processed and analyzed. Granularity can be defined at different levels depending on the specific use-case, business requirements and data sources. To support different levels of granularity in a Data Lake or Data Mesh, the data must be structured in a way that allows for easy querying, aggregation and analysis. This can be achieved through techniques such as data modeling, normalization and partitioning. By supporting different levels of granularity in a data storage architecture, organizations can ensure that each domain has access to the specific data they need to drive business outcomes. This can help to improve data quality, reduce data redundancy and promote collaboration across different teams and domains.

Decentralization in data architectures refers to the distribution of data across multiple nodes or storage systems instead of relying on a central data repository. This approach offers several advantages, including increased fault tolerance, improved scalability and greater flexibility in data management. In a decentralized storage architecture, data is distributed across multiple nodes or storage systems. Each node may contain a subset of the data or a complete copy. Nodes are connected to a network and can communicate with each other to exchange data and perform computations. This architecture can be organized in a variety of ways, such as peer-to-peer networks, distributed file systems or Blockchain-based systems. Decentralization can improve fault tolerance by reducing the risk of a single point of failure. In a centralized architecture, if the central repository goes down, all access to the data is lost. In a decentralized architecture the data is distributed across multiple nodes, so if one node goes down the others can continue to operate and serve data. Decentralization can also improve scalability by allowing data to be stored

and processed in parallel across multiple nodes. This can improve the performance of data-intensive applications and enable them to handle larger volumes of data. Finally, Decentralization can offer greater flexibility in data management by allowing data to be stored and processed closer to where it is being generated or used. This can reduce the latency and costs associated with transferring data to a central repository.

Agility in data storage architectures refers to the ability of an organization to quickly and easily adapt its infrastructure to meet changing business needs. This includes the ability to scale up or down, change data formats or structures, and integrate with new data sources or systems. Agility is important because it allows organizations to respond quickly to changes in their business environment, such as new regulations, new markets, or new opportunities. To achieve agility in data storage architectures, organizations must adopt flexible and scalable storage technologies and data management structures and practices that can be tailored to meet new business needs.

The characteristics described above are evaluated using a Likert linguistic scale including the values Low, Medium and High. Table 16 provides a definition of these linguistic values for each characteristic introduced. In the case of Data Domain Readiness and Alignment, the levels are defined by the number of actions required to prepare the data for analysis or processing. A low level indicates that more than five actions are needed, a medium level requires two to three actions, and a high level necessitates only one action. These actions include tasks such as data cleaning, formatting, and aligning data with specific domains.

Granularity bears levels that are determined based on the number of detail levels supported by the architecture: One level for low, two levels for medium, and three or more levels for high. This granularity ensures that data can be queried, aggregated, and analyzed at various levels of detail according to business needs.

Decentralization is categorized based on the extent to which data is distributed across multiple nodes or storage systems. A low level indicates none or limited decentralization, with data being largely centralized. A medium level represents normal de-centralization, with some distribution of data across nodes. A high level of decentralization means data is distributed in an unlimited manner, promoting fault tolerance and scalability.

Agility is assessed by evaluating the architecture's flexibility and ability to adapt to changing business needs. A low level signifies none or limited agility, where the system

is rigid and slow to adapt. A medium level represents normal agility, with some capacity for adaptation. A high level indicates unlimited agility, where the architecture can rapidly scale, integrate new data sources, and adjust data formats or structures as needed.

Table 16. Definition of Low, Medium, High values of each characteristic.

Characteristic	Low	Medium	High
Data Domain readiness and alignment	4-5 actions	2-3 actions	1 action maximum
Granularity	1 level	2 levels	3 or more levels
Decentralization	none or limited	normal	unlimited
Agility	none or limited	normal	unlimited

A traditional Data Lake without semantic metadata enrichment can be characterized with Low Data domain readiness and alignment as more than 5 actions are needed to prepare the data to create Data Domains and Data Products through existing data re-siding in the Data Lake. Naturally, this characteristic depends on whether semantic annotation is used in the Data Lake. If not, then the Data Lake is highly likely to become a Data Swamp where data domains are not distinct. A scheme with metadata enrichment, on the other hand, greatly benefits data domain readiness as it efficiently guides the retrieval process. Granularity also ranges according to the metadata semantic enrichment of the Data Lake. When a Data Lake does not follow any semantic enrichment policy it may be characterized with Low Granularity. Decentralization in Data Lakes can be provided somehow only through data ponds and data puddles (see Chapter 4). If a Data Lake follows a flat architecture, then it can be characterized with Low Decentralization and Low Agility as it is quite difficult to adjust quickly to changes of business needs. The traditional Data Lake without a metadata architecture was deliberately selected as an alternative approach for comparison purposes in order to demonstrate that without a metadata mechanism a Data Lake can indeed end up being a Data Swamp. Similarly, we

argue here that a Data Mesh may suffer from a similar weakness which may lead to becoming what we call here a Data Knot, that is, a route to a data product that is obstructed at some point before the full utilization of the relevant information is concluded due to the inability to combine semantics that lead to the product.

A Data Lake with semantic enrichment, such as the one relying on blueprint metadata proposed in Chapters 4 and 5 can be characterized with Medium Data domain readiness and alignment as 3 actions are needed to prepare the data in the Data Lake to create Data Domains as follows: (1) Set pillar domains and subdomains according to business needs; (2) Utilize ponds and puddles TTL metadata description; (3) Create the Data Mesh with a pillar domains matching ponds metadata attributes and subdomains according to puddle attributes. These actions are basically creating data ponds and data puddles inside the Data Lake using a domain driven approach with a maximum of 2 levels. The metadata mechanism in Chapter 4 also presents High Granularity because of the metadata enrichment included in the Data Lake, and specifically the Blueprint metadata history. High levels of Granularity are also achieved by using data puddles, which are smaller portions of organized data.

Decentralization as described above can somehow be provided in Data Lakes only through data ponds and data puddles as the framework in Chapter 5 suggests, and, of course, if distributed across multiple nodes or storage systems instead of relying on a central data repository as the original Data Lake concept dictates. Finally, a Data Lake enhanced with the blueprint semantic mechanism may be characterized with High Agility due to the fact that it can quickly adopt changes in business needs by utilizing the keywords attribute in the relevant blueprint mechanism. On the contrary, a flat Data Lake architecture does not offer such a flexibility and thus it is characterized with Low Agility.

The proposed Data Mesh architecture presented here achieves High Data domain readiness and alignment, Granularity and Agility due to the proposed Data Mesh Blueprint presented in Figure 22 and applied as demonstrated in Figure 23, which drives the creation of Data Domains and Data Products. Decentralization is one of the main characteristics of a Data Mesh architecture as presented in the Technical Background, while the proposed mechanism can be characterized with the value High for this feature.

Table 17 summarizes the points of the short comparison presented above between the Data Lakes and Data Meshes architectures and the utilization of the metadata enrichment mechanism proposed in this paper. It is evident that the use of the

mechanism offers significant benefits to the underlying data structures used in storage architectures which outperform their rivals (i.e., without the mechanism) in all characteristics used. What is most important, though, is that Data Meshes enhanced with the Data Blueprint mechanism improve their performance even further in terms of the Data Domain Readiness and alignment and Decentralization characteristics compared to the counter approach of a Data Lake with the same mechanism.

Table 17. Evaluation and comparison of the mechanism and data structures of architectures.

Approach	Data Domain Readiness and Alignment	Granularity	Decentralization	Agility
Traditional Data Lake without the proposed metadata enrichment mechanism	Low	Low	Low	Low
Data Lake with the proposed metadata enrichment mechanism [Chapter 4]	Medium	High	Medium	High
Data Mesh proposed architecture	High	High	High	High

9.4 Experimental Assessment

This section provides a short and concise description of the experiments conducted, starting with the design of the experiments and ending with discussing the results obtained.

9.4.1 Experimental Assessment

Experimentation here aims to investigate on one hand the ability of the proposed approach to create refined data products and on the other to assess its performance and effectiveness with the execution of queries. In this context a series of experiments were designed and executed to support the above targets. This sub-section describes the rationale behind their design.

Two alternative data structures of storage architectures were constructed to compare with the Data Mesh: The first one is a basic Data Lake enhanced with a similar semantic enrichment mechanism based on blueprints as the one reported in Chapter 4. The second one is an upgraded version of the first, that is, a Data Lake using the semantic enrichment

mechanism but also structured with Ponds and Puddles as presented in Chapter 5 and depicted in Figure 7. The selection of Data Lakes as the counterapproach serves two purposes: The first is to show the differences between the widely known and used architecture of Data Lakes and Data Meshes. This will provide some indications about whether Data Meshes can be regarded as the natural evolution of Data Lakes in Big Data management. Since there is limited work on the topic, the second purpose is to provide a comparison with the closest approaches, that is, with similar studies that introduced the same concept of semantic enrichment and blueprints. This comparison will enable identifying potential pros and cons of the two approaches.

Performance was assessed by varying the complexity of the experiments in terms of two factors, the number of sources producing data and the number of data products required. The former was set equal to three distinct levels, 100, 10000 and 100000, while the latter used five different values, that is, 2, 3, 4, 5 and 7. The value ranges of both factors were selected so that scaling up serves as a complexity rising factor, but at the same time the lower and upper boundaries are reasonable for addressing re-al-world needs, and even exceeding reality expectations (i.e. above 100 data sources) just to measure or compare performance. Data products for the PARG datasets were constructed at each level using the following characteristics: Level 2 – Location and Variety; Level 3 – Location, Variety and Velocity; Level 4 – Location, Variety, Velocity and Feed-cycle Start; Level 5 – Location, Variety, Velocity, Feed-cycle Start and Feed-cycle End; Level 7 – Location, Variety, Velocity, Feed-cycle Start, Feed-cycle End, Volume and Flock ID. The varying complexity targeted at investigating performance and efficiency of the proposed approach in terms of the time required for constructing the mesh (data products), as well as the ability and time for locating the appropriate sources to retrieve data from.

Description of the sources and their characteristics was performed using TTL files (uploaded on GitHub) and reflected the data characteristics provided by the PARG factory. The TTL files were created automatically by Python scripts that also masked confidential data. Increasing the number of sources directly affects (increases proportionally with) the size of the corresponding TTL file, which is the main element parsed to return sources matching a query. Indicatively, 100 sources described in a TTL file resulted in a size of 62KB, 10000 sources of 6.1MB and 100000 sources of 61.2MB. Additionally, the Data Lake architecture with ponds and puddles was created for the same datasets to facilitate direct comparison with the Data Meshes at the same level (level 2 of

data products). All Data Lake and Data Mesh constructs were implemented by splitting information into different layers of granularity using the characteristics of the TTL files described in the SDB.

Experiments were executed on a server computer with three virtual machines, a CPU with 4 x dedicated cores (the base server hosting the machines had 48 cores), memory size of 8192MB and hard disk capacity of 80GB. The software stack included Hadoop (version 3.3.6) for distributed computing, Python (version 2.7.5) for scripting, generation of data based on PARG's raw real-world data, and creation of the data products (Data Mesh level), and Apache Jena for SPARQL query processing.

Various queries were constructed and executed: (i) One reference query (Query#1), which requires all description data to be returned for each relevant source and its purpose is to measure response time (i.e., the time to locate the relevant sources), (ii) Three performance assessment queries: Query#2 retrieves source names, velocity, feed-cycle start, and feed-cycle end for all descriptions; Query#3 adds a filter to Query#2 to select only descriptions with a specific velocity (Monthly); Query#4 retrieves the source names, velocity (Monthly), and calculates the duration of each feed cycle in days for descriptions with a specific velocity.

9.4.2 Experimental Results

Table 18 presents the execution time required to construct a Data Lake with a metadata enrichment mechanism and ponds & puddles structure, and various forms of Data Meshes in terms of data products (granularity levels), while, at the same time, varying the number of data sources. The simple Data Lake structure (i.e., without ponds/paddles) was not included in Table 18 as the comparison with the other two alternatives would not be "fair" since it cannot semantically categorize information upfront and hence by default it would fall short. As can be observed in Table 18, creation time increases according to the number of sources and granularity: Data Lake and Data Mesh with 2 data products require the minimum time to create, with time steadily increasing as more data products are created, something which is expected. Construction time for Data Meshes with the maximum level used (7 data products) is substantially higher compared to lower levels, with an increase of 10 to 15 times more than the previous value of the number of sources for the same level. It is worth noting that the maximum Data Mesh construction time is less than 3 minutes, which may be considered a quite satisfactory performance taking into ac-count

the extreme conditions tested with values equal to 100000 for the sources and 7 for granularity level, which, in practice, are very rare to meet.

The reference SPARQL query (Query#1) was then executed using the various Data Mesh structures for comparison purposes. The execution times of the query are listed in Figure 26, along with the number of sources returned. As may be observed, query execution time is dependent on the overall number of sources used and is analogous to the number of sources returned when there exist various sources satisfying the query (levels 3 and 4). When granularity increases above level 4, only a limited number of resources is returned (1 in this case), which leads to executing the query rapidly and stably, irrespective of the number of underlying data sources level (see Figure 26). This is actually the most significant benefit of using the proposed Data Mesh structure, that is, to restrain the range of information categorized in the data product levels and retrieve data in an immediate and direct way.

Table 18. Creation time for each structure architecture used for experimentation with varying number of sources and data refinement levels.

Structure Architecture with Levels	**Number of Sources**	**Creation Time (s)**
DL with Ponds & Puddles / Data Mesh Level 2	100	0.0229
	10000	3.9879
	100000	29.9931
Data Mesh Level 3	100	0.0673
	10000	7.0856
	100000	72.3088
Data Mesh Level 4	100	0.1075
	10000	6.6337
	100000	70.0903
Data Mesh Level 5	100	0.0906
	10000	9.7921
	100000	93.5849
Data Mesh Level 7	100	0.2379
	10000	15.8697
	100000	166.9752

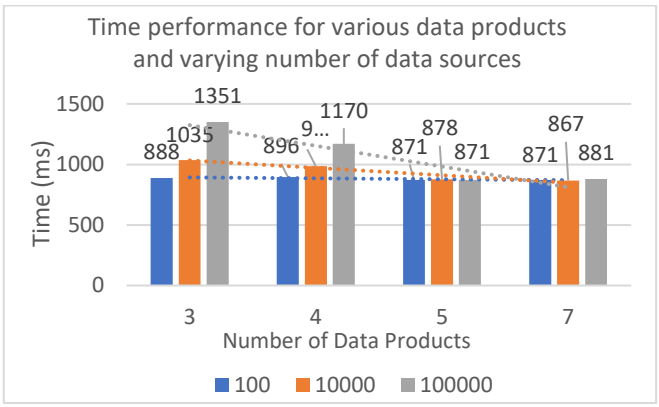

Figure 26. Execution of reference query on various Data Mesh architectures and varying data sources (10, 10000, 100000).

Finally, the same Data Mesh structures and data sources as above were utilized to execute the last experiments that used 3 SPARQL queries with varying complexity as previously described (uploaded also on GitHub). Figure 27 graphically depicts the results, which indicate consistent behavior across the queries: The average execution time after 100 iterations is quite low even with the maximum number of data sources tested, it increases proportionally to the number of available data sources, and it stabilizes as the number of sources returned saturates to 1 (data products equal to 5 and 7).

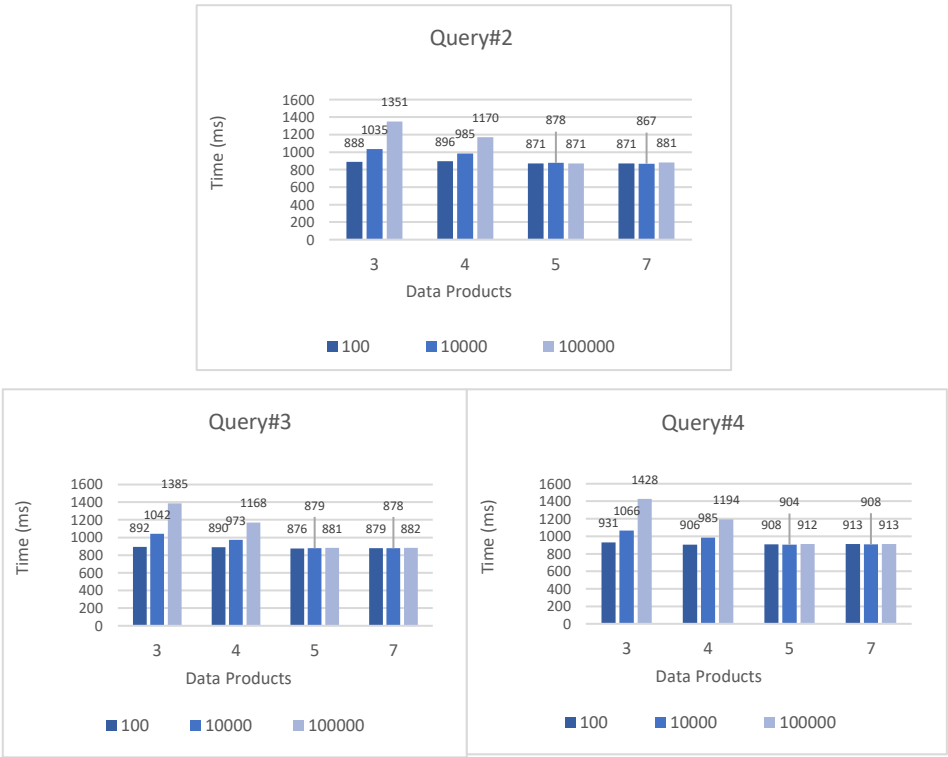

Figure 27. Execution of queries (#2, #3, and #4) on various Data Mesh architectures with increasing complexity and varying the number of data products and number of sources (10, 10000, 100000)

9.5 Summary

This chapter investigated the transformation of Data Lakes into Data Meshes by proposing a standardized approach to easily discover and construct Data Products. This approach modified and extended earlier work on Data Lakes and their metadata enrichment achieved through SDB. This was performed by following a domain-driven approach and providing a new set of blueprints able to identify and describe data products. The proposed approach was demonstrated and validated in two ways: The first involved comparison with alternative Data Lake-based structures which indicated superiority of Data Meshes over a set of qualitative features. The second involved using real-world data collected within the environment of a poultry meat factory. A set of experiments was designed and executed which revealed a successful performance both when compared to Data Lakes with similar semantic enrichment mechanisms and by varying complexity in terms of available data sources, number of data products created, and type of queries run.

One may argue that since a Data Mesh requires some time to create, which depends on the type and number of data products required, as well as the number of sources producing the data utilized, time performance may constitute a drawback hindering its wider adoption as the data products must be available before the execution of any queries. This could lead to characterizing the nature of Data Meshes as rather static in the sense that if data products need to change according to new business needs, then the Data Mesh must be recreated to accommodate changes. Nevertheless, as shown in the experiments conducted, the time it takes to create the mesh and the corresponding data products is very short. In addition, and more importantly, the execution of the queries once the data products are in place is far quicker than other similar storage architecture structures. Therefore, even with very large volumes of data, Data Meshes prove adequate to handle efficiently data retrieval. This advocates in favor of using Data Meshes as the underlying data management structure for practically any real-world application domain.

Combining a Data Mesh with software analytics may offer useful information on software processes and products, such as:

- Granular Insights: A Data Mesh enables individual teams to own their data, allowing them to use software analytics methods unique to their software systems. This strategy offers fine-grained insights into usage patterns, team-specific utilization performance, and other pertinent indicators;

- Contextual Awareness: By utilizing Data Mesh principles, teams increase their awareness of the context of the data they produce and how it relates to their software processes and products. This context gives a greater understanding of the variables affecting software performance and behavior, which improves the usefulness of software analytics;
- Rapid Iteration and Improvement: A Data Mesh provides teams with control and autonomy over their data, allowing them to iterate and enhance software products and procedures quickly using the knowledge gleaned from software analytics. Continuous improvement and agility are fostered by this iterative methodology;

Building on the principles of Data Lakes, where large volumes of structured, semi-structured, and unstructured data are stored and secured, the next evolution in data management seeks to make these vast repositories more accessible, dynamic, and governed by decentralized frameworks. As businesses increasingly demand real-time, actionable insights from their data, the limitations of traditional, centralized approaches become apparent. The need for flexibility, rapid traceability, and user-driven data products has led to the emergence of Data Meshes, a decentralized approach that embraces domain-oriented data ownership and management as presented in this chapter.

In the following chapter, an innovative framework introduced that addresses these challenges by combining SDB with Data Meshes and Blockchain technology. This framework enhances data ownership, security, and governance by leveraging NFTs to facilitate the transfer of ownership and ensure transparency. Through the application of this approach, we not only build upon existing research but also pave the way for the development of Data Markets, where data is not only accessible but securely governed, creating new opportunities for businesses to interact with and derive value from their data. The case study presented here illustrates the practical application of this framework, showcasing its effectiveness in the real-world setting of the PARG in Cyprus.

CHAPTER 10 : SECURITY AND OWNERSHIP IN USER-DEFINED DATA MESHES

10.1 Introduction

Nowadays, Big Data can be characterized as the “new currency” of the information age as it is recognized as a valuable human asset. Effective aggregation and analysis of this data may unearth information that provides insights into numerous facets of everyday activities and offers the ability to anticipate future occurrences. Big Data refers to the substantial volumes of digital information consistently produced by machine and global population from diverse sources such as social media, Internet of Things (IoT) devices, machines and sensors logs, public records and open data, online transactions, websites and applications, research and scientific instruments, etc. (Gupta et al., 2018). The vast majority of Big Data originates from heterogeneous data sources, yielding a variety of data types that include structured, unstructured, and semi-structured data. Encompassing a diverse range of content, Big Data spans from textual information to multimedia elements, such as images, videos, and audio (Blazquez & Domenech, 2017).

The three primary characteristics (3Vs) of Big Data, as presented by Dough Laney in 2001, form and define its fundamental framework (Al-Sai et al., 2022). Firstly, Volume represents the broad amount of data generated from data sources, often reaching high levels that challenge typical data processing methods. The second characteristic defining the tempo with which data is created, processed, and made available for analysis is denoted by Velocity. Fast processing speeds are required to keep up with the increasing rate of data creation due to the emergence of real-time data sources like social media and sensors. Thirdly, the term Variety highlights the variety of data kinds, encompassing organized, unstructured, and semi-structured information. By integrating a broad range of textual, visual, and audio information, this inclusivity recognizes that Big Data extends beyond traditional databases. Taken together, these three qualities create the foundation for realizing and capitalizing on the possibilities of Big Data in a data-driven modern

world. In addition, seven more characteristics were included in this list after 2001 leading to the 10Vs term for Big Data. The new properties are Value, Veracity, Volatility, Validity, Vulnerability, Variability, and Visualization (Khan et al., 2018) and offer additional descriptive assets of Big Data.

In the pre-Big Data era storage designs were mostly based on file systems and conventional relational databases. Relational databases with clear schemas, like MySQL and Oracle, were great at handling structured data. A lot of people used file-based storage systems, such us Network Attached Storage (NAS) and Storage Area Network (SAN), to store documents and other kinds of files. During the same period, the conventional approach to address escalating data requirements involved vertical scaling, which entailed augmenting resources on a single server (Khine & Wang, 2019). In the era of Big Data, which is characterized by immense data volumes, rapid data transfer rates and the diversity of weakly structured data from numerous heterogeneous sources, as declared also by the 10Vs characteristics, resulted in a fundamental transformation of storage architectures. NoSQL databases, such as MongoDB and Cassandra, as well as distributed storage systems like Hadoop Distributed File System (HDFS), have now become more popular (Shahid et al., 2021).

Data Lakes seamlessly integrate with tools and technologies that enable processing, querying, and analyzing the stored data. Properly configured Data Lakes can implement security measures and data governance policies to ensure privacy and compliance with regulations. While Data Lakes offer a high degree of flexibility, they require careful management to prevent them from becoming "data swamps", that is, hosting places where data is poorly organized, difficult to find, and hard to analyze (Derakhshannia et al., 2020). To address this concern, practices like metadata management, data cataloguing, and establishment of data governance policies are crucial as presented in Chapter 4, Chapter 5 and Chapter 6. Figure 28 presents the structure of a Data Lake and an algorithmic description of how the Data Lake concept works in practice, from collecting the data, annotating it using metadata, storing it and finally retrieving it based on the metadata tags.

The complex interactions amongst Data Lakes, Data Meshes, and Data Markets in the Big Data era create a dynamic ecosystem that transforms how businesses manage and extract value from heterogeneous data sources and Big Data (Driessen et al., 2023). These storage architectures and structures could be deployed with storage and processing

technologies, such as Apache Hadoop, Apache Spark, or cloud-based solutions like Amazon S3, Azure Data Lake Storage, or Google Cloud Storage (Kunigk et al., 2018). While these frameworks are linked with Big Data Processing, the primary unsolved challenging problems revolve around security, encompassing issues related to privacy, regulatory requirements, and access control. Notably, weaknesses in metadata management pose challenges, as data in lakes or meshes can be replaced without proper oversight of the contents (Derakhshannia et al., 2020).

The concept of Data Mesh, as introduced in Chapter 2, Chapter 3 and Chapter 8 in this thesis was introduced in 2019 (Dehghani, 2024), which essentially represents a novel approach to data management within large organizations. Unlike traditional methods, a Data Mesh emphasizes several key concepts to revolutionize data handling. Firstly, it advocates for Domain-oriented Ownership as mentioned in Chapter 2 and Chapter 9. Specifically, this means that data domains are entrusted to the teams or business units possessing the highest expertise in that specific domain. These teams bear the responsibility for ensuring the quality, accessibility, and privacy of their respective

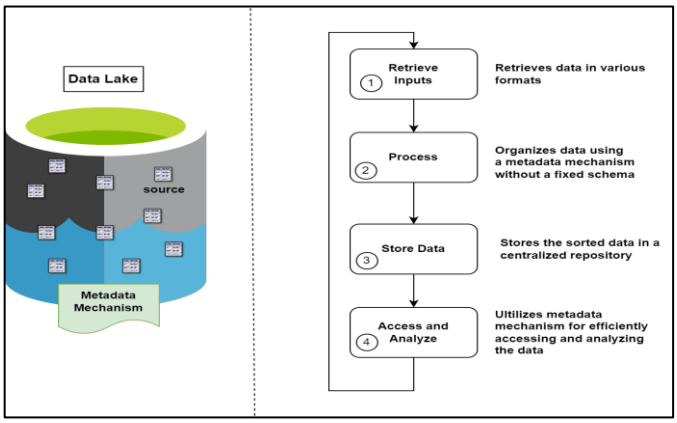

Figure 28. Data Lake architecture and the concept algorithmic approach

domain's data. Additionally, a Data Mesh promotes the idea of Decentralized Data Products. Here data is treated as a product and each domain team is accountable for the entire data lifecycle within their domain. This encompasses tasks such as production, consumption, quality assurance, privacy measures, and comprehensive documentation. Further-more, Data Meshes advocate for Federated Computational Governance, an approach where each domain team defines and enforces the computational logic specific to their do-main. This logic is then executed within the broader context of the mesh (Dehghani, 2024).

To facilitate autonomy and efficiency, Data Mesh incorporates a self-serve data infrastructure. This infrastructure is designed to empower domain teams with the necessary tools and resources to independently manage their data products, reducing reliance on centralized data engineering teams (Van den Heuvel et al., 2023). Embracing an API-first approach, Data Mesh encourages the utilization of Application Programming Interfaces (APIs) for seamless data exchange and communication between different components of the system. This promotes loose coupling and flexibility in how data is consumed and utilized. Figure 29 presents the structure of a Data Mesh and the algorithm that serves as the core of its operation.

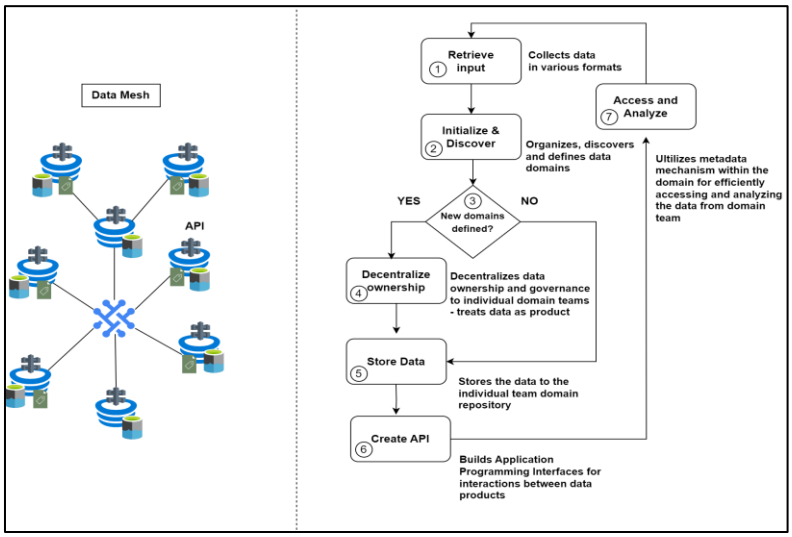

Figure 29. Data Mesh architecture and the concept algorithmic approach

The primary research contribution of this chapter lies in the introduction of an innovative framework that leverages the Semantic Data Blueprints (SDB) introduced in [Chapter 4] and [Chapter 5] or the dynamic assembly of Data Meshes and data products responding to user demands on one hand, and ensuring that stakeholders access specific areas of the Data Mesh as needed via transfer of ownership on the other. The integration of non-fungible tokens (NFTs) and Blockchain technology collaboratively establishes a novel approach to address data ownership and governance concerns. The core of the framework is a dedicated algorithm which involves the execution of specifics steps to facilitate secure and transparent data ownership transfers by incorporating the ability to mint time-based NFTs with extended functionality.

The proposed approach builds upon and expands earlier research on the subject that proposed SDB, a semantic metadata enrichment technique for Data Lakes that enables

the effective storing and retrieval of data from distributed and heterogeneous data sources and ensuring security in Data Lakes using Blockchain technology and NFTs (see Chapter 6). The same concepts are employed in this work but this time they align with the characteristics of Data Meshes, ensuring security and ownership through the integration of Blockchain and NFT technology, thereby paving the way for the development of Data Markets. In this context, a Data Mesh is thought of as the evolution of a Data Lake in terms of managing massive amounts of data (Big Data) expressed in a variety of formats (structured, unstructured, and semi-structured), but most crucially, for making it simple, rapid, and effective to trace. Users and their preferences or needs shape the form and variety of data products synthesized within the Data Meshes. Users initiate a series of functions within the system, including data retrieval from the Data Lake, data product construction, and ownership transfer. Through this iterative process, the framework generates a diverse array of outputs, encompassing both traditional data products and NFTs, something that underscores the collaborative nature of data management and emphasizes its adaptability to accommodate the evolving needs and preferences of its users. Again, real-world manufacturing data from Paradisiotis Group (PARG) is used to illustrate the proposed approach.

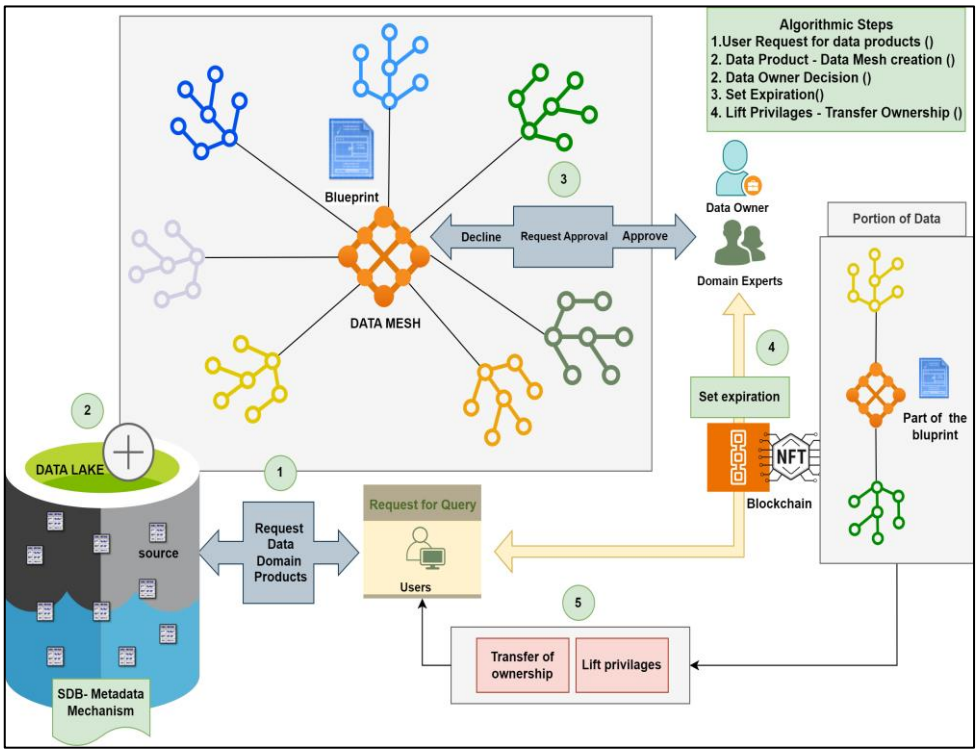

Figure 30. The algorithmic process of transferring ownership from the data owner to a user

10.2 A Supporting Framework for Transferring Ownership in Data Meshes

This section describes the proposed framework for transferring ownership of data products residing in Data Mesh (see Figure 30). The framework follows a series of algorithmic steps that include the creation of the Data Mesh through its transformation from a Data Lake that bears a specific architectural structure, and the development of the appropriate smart contracts the execution of which facilitates the transfer and proves ownership of a specific data product. In our case, fluidity of information in a Data Mesh depends on the type and diversity of the data stored in the Data Lake from which data products are created.

10.2.1 Semantically Enriched Data Lake Architecture and Data Mesh Products Creation

A metadata mechanism is of paramount importance for a Data Lake as it functions as its organizational backbone, offering a systematic and detailed catalog of the diverse datasets hosted within the Data Lake. Without such a metadata mechanism a Data Lake will gradually be transformed into a "Data Swamp". In essence, a metadata mechanism provides data owners with a vital insight into the type and context of the stored information by capturing important details about the origin, structure, relationships, and usage of data. By providing this information navigating and mapping the raw data becomes feasible, something that makes data search, retrieval, and management easier and efficient.

The SDB is a metadata enrichment mechanism that identifies and characterizes a candidate source before it becomes member of a Data Lake (see chapters 4 and 5. The framework described in integrates blueprint ontologies with the 5Vs Big Data features, namely Volume, Velocity, Variety, Veracity, and Value, to support data processing (storage and retrieval) in Data Lakes organized with a pond architecture. The latter structures a Data Lake in several distinct data ponds, each of which holds or refers to a certain type of data according to the pond design. Depending on the type of data (structured, semi-structured, unstructured), each pond has a unique data processing and

storage method. When extracting data from the Data Lake, this built-in pond architecture is quite useful as it supports quick and easy access to the storage space.

As previously mentioned, a dedicated blueprint is developed to describe each data source storing data in the Data Lake. Specifically, the blueprint of a source consists of two interconnected blueprints as shown in Figure 3, the stable and the dynamic blueprint (see Chapter 4).

The former is static and describes the name and type of the source, the type of data it produces, as well as the value, velocity, variety, and veracity of the data source pushed in the Data Lake. The latter is a dynamic blueprint which involves attributes that are not stable over time and essentially characterizes volatile properties such as the volume of data, the last source update, and keywords characterizing the source. The dynamic blueprint is updated every time data sources produce new real-time or batch data, or its description through keywords may be modified. In essence, the metadata description - SDB is provided in Terse Triple Language (TTL) using the Resource Description Framework (RDF). The metadata mechanism contains TTL descriptions for all the sources included in the Data Lake. In essence, TTL is a serialization format that provides a concise and hu-man-readable way to represent RDF data, making it easier for both machines and hu-mans to work with semantic information on the Web. RDF represents information as triples, which consist of subject-predicate-object statements. The resource being de-scribed is the subject, the property or attribute is the predicate, and another resource or value is the object. An example of a triple may be ex:variety “unstructured", which means that the subject is the source, the predicate is "variety" and the object is the value "unstructured".

Our framework supports the categorization of data into three pillars: structured, semi-structured, and unstructured. This classification allows for the flexible inclusion of diverse datasets transcending the constraints of data representation through meticulous metadata recording and semantic enrichment. By introducing the stable and dynamic blueprints, the framework captures the essence of data variety, thereby ensuring independence from specific representations and underscoring its adaptability and robustness across varying datasets, marking a significant stride in modern data analytics.

Let us assume that a user requests access to specific sources producing data and storing it to the Data Lake. In this case a dedicated SPARQL query is formed and executed on the Data Lake. When the query starts executing, it first asks the owner of the data for

her/his approval. If the owner approves the query, then the framework, and specifically the metadata mechanism of the Data Lake, is utilized to create the corresponding Data Mesh data product that satisfies the query as presented earlier in Figure 3. Figure 3 also shows that the user has access only to the sources requested through the corresponding APIs. Furthermore, this access is restricted to the specific person and is valid only within a specific period via Blockchain and NFT technologies as will be presented with details in the next subsection.

Analytically, the steps taken are as follows:

I. The owner of the contract can add an administrator on the contract by calling the addAmin() function inserting an EVM-compatible address. Once an administrator is created, (s)he gets access through her/his address to certain admin-only functions on the contract.

II. An administrator can mint an NFT by executing the safeMint() function providing the address of the recipient, an expiration date in UNIX epoch time, the query that is associated with the NFT and its access level. If the value of the access level is set to 1, then the NFT grants read-only access to its new owner and the NFT is non-transferable, while, if the value is set to 2 the owner of the NFT, besides the read access, gets also transfer access and therefore can transfer the ownership of the NFT to a different user. At any given time, the current owner of the NFT can access and read the data.

10.2.2 Smart Contract Architecture

The Blockchain-based architecture uses a specially designed ERC721 smart contract that was implemented to evaluate the use of the proposed framework. ERC721 is a standard that is used in EVM-compatible Blockchain networks to represent ownership of NFTs, where each token is unique and has its own metadata. In this work we decided to develop our smart contract based on the ERC721 standard for two main reasons: (i) with ERC721, users can securely own, transfer, and manage their digital assets with transparent and verifiable ownership records, and, (ii) the ERC721 standard ensures that NFTs can easily interact with several wallets and decentralized applications (dApps), enhancing their utility and accessibility. Additionally, as the smart contract was developed for deployment on EVM-compatible blockchain networks such as ETH, Matic, Avalanche

etc. that are using a Proof-of-Stake consensus mechanism, there is no need for large energy consumption for the calculation of blocks.

The purpose of the smart contract developed in this paper is threefold: (i) Allow data owners to mint time-based NFTs and transfer them to an address; (ii) Allow NFT owners to read specific portions of data for a certain period of time; and, (iii) Allow NFT owners to transfer ownership of the data to a different user. The proposed smart contract consists of three main actors: The contract owner, who is the deployer of the contract and is also responsible for registering administrators onto the contract; the contract administrators, who oversee the minting process; the authorized users who can view or transfer data. The administrator algorithmic workflow of the proposed framework is depicted in Figure 31 and summarized in pseudocode.

Figure 32 presents the algorithmic workflow followed for authorized users. Authorized users can view the assigned data based on two parameters, the expiration date and the query. A user holding a valid NFT can access a token-gated website to view the data. The website checks the eligibility of the connected address to allow or refuse access to the user. Finally, as depicted in Figure 33, NFTs are separated into two categories, transferable and non-transferable. When an NFT is minted, the admin specifies if the token has read-only or transfer access. When a user who holds a specific non-expired NFT initiates a transfer function, the contract checks whether the token can be transferred or not to a different address and proceeds to accepting or rejecting the request accordingly. If the NFT is successfully transferred, then the new owner of the NFT is automatically granted access to the token-gated website and can view the data.

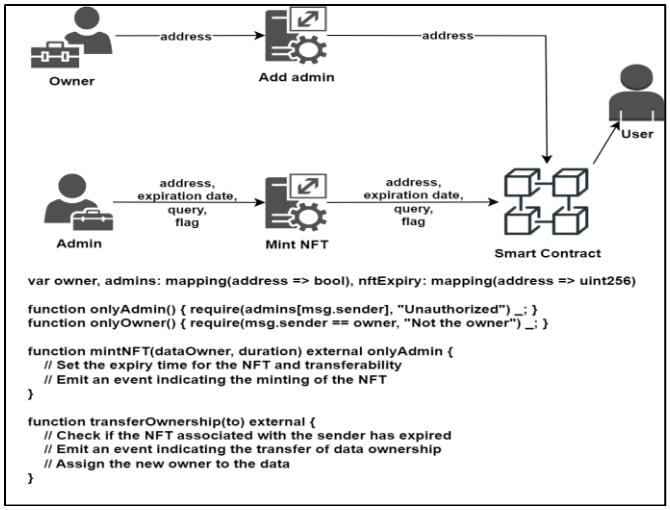

Figure 31. The admin algorithmic workflow and pseudocode

Two notes should be made here: (i) By using hash functions and asymmetric encryption we ensure alignment with the different data privacy regulations (e.g., GDPR and CCPA). Each NFT in the developed system is owned by an EVM-compatible address, thus only the address that owns the NFT can access the system and/or transfer the NFT. Thus, granting access to data can only be initiated by the owner of the NFT and in cases in which the administrator enabled transfer access to the specific NFT during its minting (creation). (ii) Delays and costs that may occur are normally associated with the number of transactions stored on the Blockchain as well as the data volume in each transaction.

In the proposed approach this volume is kept minimal as the data itself is not stored on the Blockchain, but rather a metadata description of this data. Moreover, as the transaction on the Blockchain does not concern the data but only the metadata, scaling of the Data Lake (i.e., its expansion) does not affect the description of an existing source. In general, the scalability of the Data Lake and the Data Mesh (i.e., the increase in their size) does

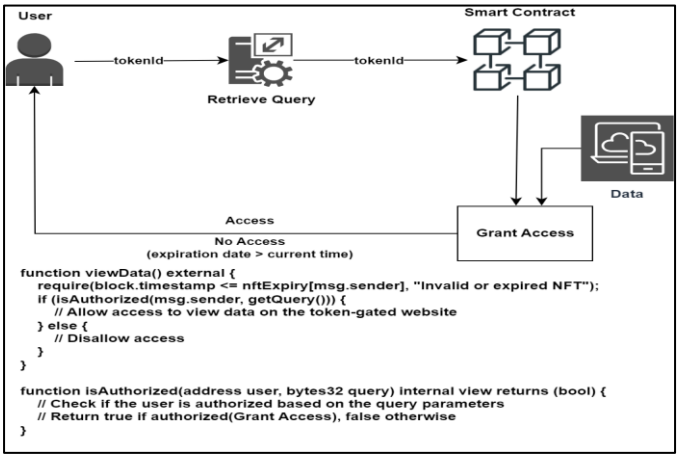

Figure 32. The authorized user algorithmic workflow and pseudocode

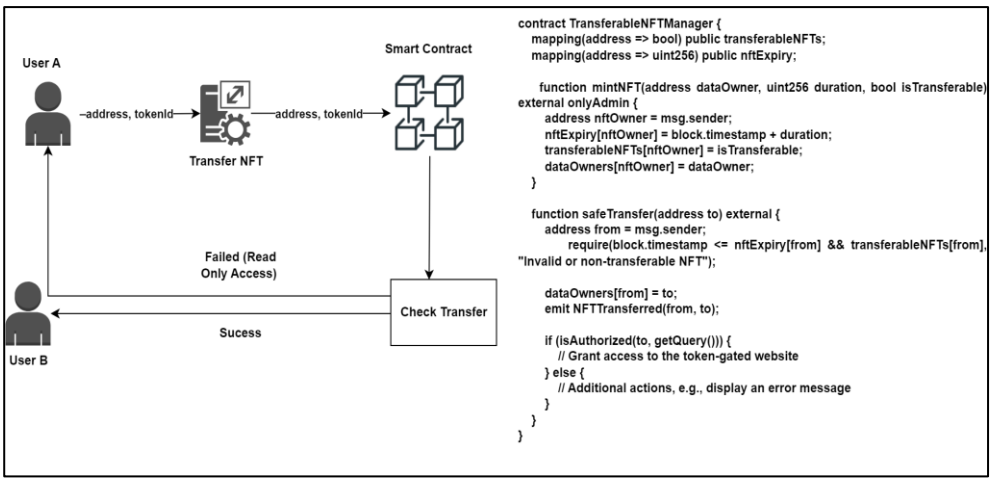

Figure 33. The algorithmic user workflow for transferable and non-transferable NFTs

not affect the metadata. The only case in which metadata is affected is when new sources of data are hosted in the Data Lake. But again, the information included in the metadata is very limited and it is produced once-off for each new source. Finally, it should be noted that this approach focuses on the transformation of the Data Lake into data products (Data Mesh), which again does not depend on the volume of the data kept in the Data Lake and, hence, not on scale.

10.3 Framework Demonstration Through a Real-World Case Study

10.3.1 The PARADISIOTIS Group (PARG) Factory Case-Study

As previously mentioned, this work utilizes a real-world case-study from the area of smart manufacturing to demonstrate the applicability of the framework. Specifically, it utilizes data recorded at the PARG factory, the main business line of which is chicken farming and poultry meat production and distribution. PARG is a continuously growing company that invested over the years in modern and technologically advanced equipment for the breeding processes (e.g., automatic ventilation system, technology assisted mill for mixing ingredients and preparing chick food, etc.) and the production line (cutting, mixing and packaging of poultry meat). The management of the factory constantly seeks to improve performance and quality levels by frequently adapting the production processes and adopting new technologies.

Data is produced within the factory mainly by two systems: (i) CUBORA is a fully operational heating control system designed to produce and monitor data related to poultry heating and emissions into the feeding atmosphere. This system is essential for ensuring the healthy growth and well-being of chicks on farms; and (ii) AGROLOGIC, which specializes in the field of automated climate controllers, feeding and weighting systems. AGROLOGIC is integrated with Chore Time controller and collects metrics from several remote sensors that are distributed into the farms, such as CO2, Temperature, Humidity, Air Static Pressure, and Light Intensity Level. All metrics are recorded in a database and are accessed through a Web application in real-time. Further-more, images of the farms and/or equipment may be recorded for shift managers to inspect visually when necessary. Finally, the system generates alerts if any of the metrics exceed pre-defined thresholds via an embedded GSM modem.

PARG case-study presents all characteristics of Big Data originating from heterogeneous sources with atypical patterns, which produce various kinds of structured, semi-structured, and unstructured data in high frequencies. This heterogeneous data needs to be treated differently than normal production speed data and be stored in more flexible and/or higher servicing speed data storage architectures or structures compared to classic Relational Databases and Data Warehouses, such as Big Data Warehouses, Data Lakes and Data Meshes. To this end, the current work developed a dedicated Data Lake for PARG in a controlled (lab) environment and applied the basic principles of SDB, Blockchain and NFT technologies for creating Data Products and Domains. The latter are produced based on a Data Mesh constructed through the Data Lake metadata mechanism. User requests for access to these data products are addressed to the Data Owner and then ownership may be granted through NFTs based on the relevant privileges, providing at the same time the ability to grant access and use the data only for a specific period of time.

10.3.2 Use-Case Scenarios

As previously mentioned, a request to access a Data Lake is supported by the utilization of the SDB semantic enrichment mechanism, which is the cornerstone for creating a data product as part of the Data Mesh according to user preferences and ownership granting. Access and ownership for a specific period of time is recorded on Blockchain using a dedicated NFT. The use-case of the PARG factory focuses on the department of poultry feeding where sources produce data during the feed-cycle of chicken within a specific farm. An excerpt of the structure of the corresponding SDB is depicted in Figure 8. We have selected the following metadata characteristics to describe a source which produces data for monitoring the chicken flock farming in different locations: (i) Source Name; (ii) Location; (iii) Feed cycle start; (iv) Feed cycle end; (v) Keywords; (vi) Varie-ty; (vii) Velocity; (viii) Volume, and, (ix) Source Path.

The use-case scenarios tested are based on user requests to access a specific portion of data. For example, PARG stakeholders (shift managers, farm carers, production workers) often need to consult data related to the number of chicks in a farm, the environmental conditions within a farm, electricity consumption, emissions in the atmosphere, biomass production, etc. Therefore, data products in the Data Mesh were constructed to reflect these pieces of available information. Furthermore, in the scenarios below we assume

also that the shift manager wishes to acquire access to all information related to the Limassol farm and that at some point she wishes to transfer this access to the head of production. Normally, access permissions are requested by sending a message to the owner of the data through a dedicated SPARQL. This query is essentially executed in all scenarios that follow. Essentially, a user requests access to specific portion (data sources) of a Data Lake through the Data Mesh data products that are constructed to provide information for location “Limassol” as presented also in Figure 34.

To evaluate the efficiency of the proposed framework we first developed a dedicated smart contract that was deployed on the Sepolia Test Network and then we executed a series of transactions. The smart contract’s address is 0x88790ed3407e3b395ab0276d5305a273a497612b and the contract owner is 0xfb43d1384FC250B59996933CA2D8C766722 7BE52. The reader can refer to the smart contract’s URL on etherscan.io for complete access to the source code of the contract. By using the smart contract, we have explored various scenarios to showcase its fundamental features that include NFT minting with additional on-chain information, data retrieval and transfer restrictions.

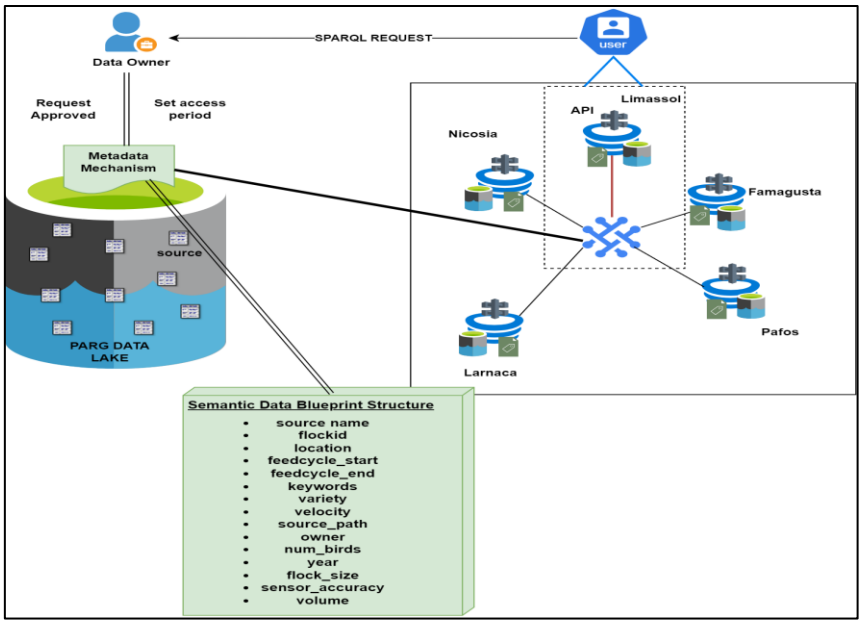

Figure 34. PARG’s Data Lake SDB structure for the use-case scenarios.

10.3.2.1 Scenario 1 - Minting

As previously mentioned, this scenario demonstrates how a user that wishes to access all sources of the factory that produce data during breeding with location the city of Limassol

is serviced by executing the SPARQL query listed below. This scenario illustrates the minting capability of the smart contract, as outlined in the proposed administrative algorithmic workflow framework. Initially, the admin of the smart contract with address 0xfb43d1384FC250 B59996933CA2D8C7667227BE52 executes sequentially two token processes for minting transactions to the address 0xcF1aB65AE4EFaA9BE8 cDB13078360B811D11616D, the first not allowing the token to be transferrable (i.e., the ownership of the data may not be passed on to another user) and the second allowing to do so. The processes are executed with the following parameters:

First token process parameters:

Date of Expiration: 1706094000 (Wednesday, 24 January 2024 11:00:00 UTC)

Query: PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX ex: <http://example.org/>

SELECT ?location ?sourcePath

WHERE { ?source rdf:type ex:Description; ex:location "Limassol"; }

Transferrable: NO (flag is set to 1)

Second token process parameters:

Date of Expiration: 1706095000 (Wednesday, 24 January 2024 11:16:40 UTC)

Query: PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX ex: <http://example.org/>

SELECT ?location ?sourcePath

WHERE { ?source rdf:type ex:Description; ex:location “Limassol”; }

Transferrable: YES (flag is set to 2)

Once the transactions are confirmed on the Blockchain network, the address 0xcF1a B65AE4EFaA9BE8cDB13078360B811D11616D becomes the owner of both token ids #0 and #1, as depicted in Figure 35. Essentially, the owner has access to the PARG sources for Limassol’s farm with either token#0 or token#1. The main difference between the two tokens is the ability to use them for transferring ownership to another user as will be demonstrated below.

10.3.2.2 Scenario 2 Retrieving Data

This scenario presents the retrieving capabilities of the smart contract which are based on certain requirements. Here we are using address 0xcF1aB65AE4EfaA9BE8cDB13078360B811D11616D that corresponds to the owner of both NFTs #1 and #2. This address is checked to comply with two restrictions: First, that it is the owner of the NFT, and second, that the NFT has not expired. These restrictions safeguard that the address has permissions to retrieve the data recorded on the smart contract for each token (see also Figure 36). Therefore, access to the data product constructed to include all information produced within the Limassol farm is now granted to the owner of the corresponding address. If any other address besides the owner of the NFT attempts to retrieve that data, it is automatically blocked by the smart contract and it is not allowed to enter the token-gated website (see Scenario 3).

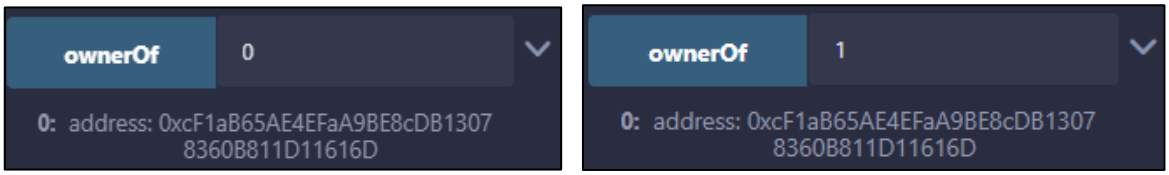

Figure 35. Owner for token ids #0 and #1

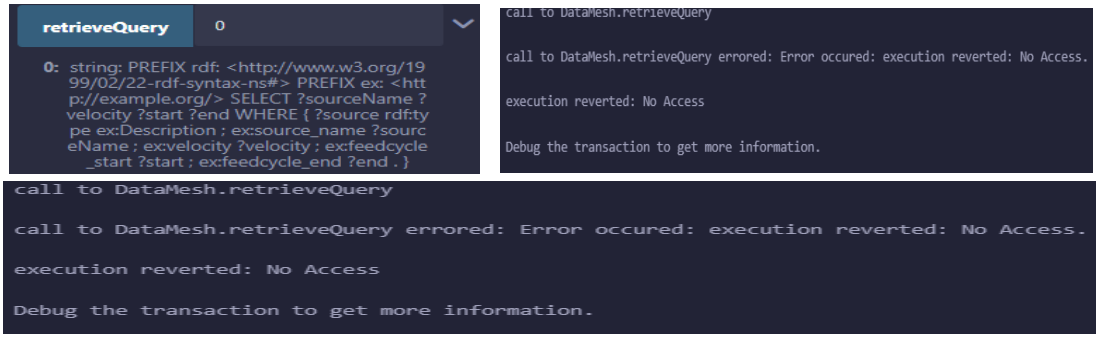

Figure 36. Query results directly from on-chain data and portal

10.3.2.3 Scenario 3 – Applying Transfer Restrictions

This scenario demonstrates the transfer restrictions that are set by the administrator when a token is minted. As described in Scenario 1, token#0 was minted as a non-transferable token, while token#1 was minted with transferrable properties. As outlined in Figure 37, when the owner attempts to transfer token#0 to a different ad-dress it is blocked by the

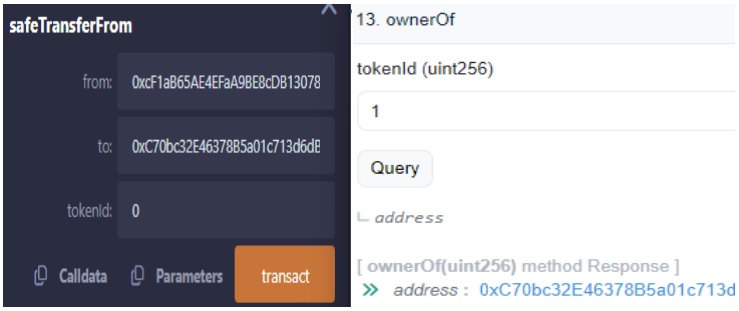

Figure 37. Result for token#0 and token#1

smart contract as this is not a valid action due to transfer re-strictions. Subsequently, when the owner of token#1 tries to transfer the token, this is carried out successfully as token#1 has the appropriate transfer rights and hence the permissions to do so. Here the owner of token#1 transfers the token to address 0xC70bc32E46378B5a01c713d6dB18 042Acd8F0200. Upon confirmation of the transaction on the Blockchain network, the previous owner of the token loses access to it as now the access rights are transferred to the new owner. Therefore, access to the data products is secured via Blockchain and single control of ownership is guaranteed by the NFT.

10.4 Summary

This chapter introduced an innovative framework for securing access and ownership in Data Meshes based on Blockchain and NFTs. The framework is applied on a Data Lake storage architecture, which may host Big Data at any scale, frequency and format, and utilizes Semantic Data Blueprints for dynamically constructing data products in Data Meshes. These products are designed to meet user demands and ensure that stakeholders access specific areas of the Data Mesh as needed through the transfer of ownership. The integration of NFTs and Blockchain technology offers a novel approach to address ownership and governance concerns. A dedicated algorithm was developed for incorporating the ability to mint time-based NFTs, thus facilitating secure and transparent data ownership transfers. The proposed framework was demonstrated using a real-world case study from the smart manufacturing area. Specifically, a Data Lake was built to host data produced at a poultry meat production factory by several sensors and automated systems during the breeding process followed in the farms. Specific portions of data were

selected to construct data products in a custom Data Mesh which were then used as key elements for granting access and transferring ownership to authorized users via the execution of smart contracts and NFTs. The scenarios tested suggested successful behavior in terms of ease of use, transparency, and correctness. It should also be noted that users in the factory (workers and managers in breeding sites and production line) were able to follow easily the algorithmic approach of the proposed framework and apply its steps efficiently, appreciating and greatly appreciated the ability to share data.

CHAPTER 11 : CONCLUSIONS, ONGOING AND FUTURE WORK

11.1 Conclusions

This PhD thesis has explored the evolution of modern data storage architectures in the context of Big Data management, emphasizing the role of semantically enriched mechanisms in enhancing data governance, security, and retrieval efficiency. Through theoretical analysis, empirical research, and real-world applications, this study has proposed and validated innovative approaches to improve data structuring, scalability, and security in decentralized environments.

The key contributions and findings of this thesis are summarized as follows:

1. **Advancements in Data Storage Architectures:**
    - The study demonstrated that traditional Data Lakes, while effective in storing large volumes of data, often struggle with governance, accessibility, and security challenges.
    - The transition from centralized Data Lakes to decentralized Data Meshes enhances data discoverability, real-time insights, and organizational agility.
2. **Semantic Enrichment through Metadata-Driven Frameworks:**
    - A metadata-driven semantic blueprint (SDB) was introduced to standardize data representation, improving retrieval efficiency and governance.
    - The integration of process mining techniques within Data Meshes enables more effective decision-making and operational efficiency.
3. **Blockchain and NFT Integration for Data Security:**

- Blockchain technology was successfully integrated into the proposed framework, ensuring data integrity, ownership verification, and secure access control.
- Non-Fungible Tokens (NFTs) were utilized to enhance data ownership verification, particularly in industrial and manufacturing environments.

4. **Empirical Validation and Real-World Applications:**
    - Experimental results from a smart manufacturing case study showcased the practical benefits of the proposed framework, including:
        - Reduction in data processing delays.
        - Improved traceability and security of data transactions.
        - More efficient decentralized access control mechanisms.
    - Comparative analysis highlighted the superior performance of Data Mesh architectures over traditional Data Lakes in terms of scalability, flexibility, and governance.

Furthermore, the research conducted contributes to the field of Big Data management by:

- Introducing a novel semantically enriched Data Lake framework that enhances metadata structuring and storing/retrieval efficiency providing granularity, process mining readiness and expandability such as synergies with Visual Querying and Knowledge Graphs.
- Transforming Data Lakes into Data Meshes, which improves governance, enhances data product discovery, and enables domain-driven decentralized architectures.
- Demonstrating the practical application of Blockchain and NFT technologies for securing data transactions and ownership in Data Lakes and within decentralized approaches such as Data Meshes.
- Providing a scalable and secure methodology that enterprises can adopt to optimize data management, security, and interoperability in large-scale environments.

11.2 Ongoing and Future work

11.2.1 Work in progress

Ongoing research in data ecosystems is rapidly evolving to address the increasing complexity and volume of heterogeneous data sources. As digital transformation continues to accelerate across industries, innovative solutions are needed to improve data quality, integration, and usability. This section presents key areas of current investigation that aim to enhance data management processes, particularly within Data Lakes and Data Meshes. Emphasis is placed on the application of emerging technologies such as Digital Twins, Knowledge Graphs, Recommended Systems and Large Language Models (LLMs) to address domain-specific challenges, support rat intelligent automation, and foster more scalable, explainable, and adaptive data systems. The following subsections outline selected research directions that reflect the ongoing efforts to tackle these challenges and contribute to the next generation of intelligent data infrastructure.

11.2.1.1 Using Knowledge Graphs for Record Linkage in Data Lakes

The exponential growth of data has driven advancements in information systems, enabling data-driven decision-making across multiple industries. However, the heterogeneity of data sources, formats, and domains presents significant challenges in ensuring data consistency, completeness, and accuracy. Record Linkage (RL), the process of identifying and resolving duplicate records across disparate data sources, plays a crucial role in mitigating these challenges. While RL has seen notable advancements with the adoption of machine learning (ML) techniques, domain-specific applications still require human intervention to incorporate expert knowledge, making RL expensive and time-consuming.

Knowledge Graphs (KGs) provide a structured and semantically rich representation of domain knowledge. By integrating KGs into RL methodologies, domain-specific challenges can be addressed more efficiently, reducing the need for extensive manual curation. This chapter explores how KGs can be utilized to enhance RL accuracy, reduce human intervention, and improve explainability in vertical applications such as healthcare, finance, and government.

RL involves comparing records from different datasets to determine whether they refer to the same entity. Traditional RL approaches include probabilistic models, rule-based systems, and ML classifiers. However, these methods face several limitations.

Data heterogeneity is a significant challenge, as different data formats (structured, semi-structured, and unstructured) complicate standard RL techniques. Additionally, domain-specific knowledge requirements pose a problem, as specialized domains, such as healthcare, require expert-driven rule sets to handle terminological variations and contextual dependencies. Scalability and cost further compound these issues, as RL processes are computationally expensive, especially when applied to large-scale datasets. Finally, explainability and interpretability remain hurdles, as many ML-based RL techniques function as black-box models, making it difficult to justify linkage decisions.

KGs offer a structured means to capture and store relationships between entities, enabling context-aware decision-making in RL tasks. KGs are particularly useful in RL due to their ability to improve accuracy, provide data augmentation, handle temporal variability, integrate multiple modalities, and enhance explainability.

KGs encode domain knowledge, helping RL models infer entity relationships that may not be directly evident in raw data, thereby improving accuracy. Additionally, KGs can be leveraged to generate synthetic labeled data, enhancing ML-based RL training. Temporal KGs allow RL models to track evolving entity relationships over time, improving longitudinal data linking. Furthermore, KGs support the incorporation of multiple data modalities, such as text, images, and structured records, which is crucial in applications like healthcare. Lastly, the structured representation of knowledge in KGs enables the development of interpretable RL models that provide context-aware explanations.

Data lakes can be strategically embedded into this framework to address the challenges of data heterogeneity and scalability in Record Linkage (RL). Data lakes are capable of storing vast amounts of heterogenous data from multiple sources in their native formats, making them ideal repositories for the diverse datasets involved in RL. By integrating Data Lakes with RL pipelines, organizations can centralize disparate data sources without imposing rigid schema requirements upfront. This flexibility will allows RL systems, especially those enhanced by Knowledge Graphs (KGs), to access a broad and diverse range of data types, facilitating more comprehensive and accurate linkage operations. Furthermore, the scalable nature of Data Lakes could support the processing demands of

ML-based RL techniques, which often require significant computational resources and access to large volumes of training data.

In addition to serving as a central data repository, Data Lakes could enhance KG construction and enrichment by acting as a source for continuous data ingestion and update. Through real-time or batch ingestion, Data Lakes can feed up-to-date records into KGs, allowing for dynamic KG updates that reflect evolving entity relationship - an essential capability for longitudinal RL tasks, such as tracking patient records over time in healthcare and other applications. Data lakes could also support the integration of multimodal data (e.g., clinical notes, images, sensor data), which, when structured through KGs, provides richer context for RL decisions. By embedding Data Lakes into the RL-KG ecosystem, organizations can create a more robust, scalable, and context-aware environment for entity resolution, driving more informed and accurate decision-making across domains.

The healthcare domain presents a complex RL challenge due to variations in patient records across hospitals, insurance providers, and research institutions. Consider an example where a patient, John Doe, appears in two different hospital databases with slight variations in name spelling, hospital name, and prescribed medications. A traditional RL system might misclassify these records as distinct entities due to these inconsistencies.

By integrating a healthcare-specific KG, the RL model can infer that Rifampin and Isoniazid are often prescribed together for tuberculosis treatment. The model can also recognize that Duke University Hospital and Durham Regional Hospital are part of the same healthcare network. This additional domain knowledge enables the RL model to correctly link the records, reducing false negatives and improving accuracy.

Automated knowledge integration presents an exciting opportunity, as advancements in KG construction can enable automated extraction and integration of domain knowledge, reducing the need for manual curation. Additionally, explainable AI in RL holds promise, as combining KGs with explainable AI techniques can provide intuitive justifications for RL decisions, improving trust and adoption in critical applications.

Cross-domain RL is another promising direction, as the use of interlinked KGs across multiple domains (e.g., healthcare and finance) can enhance RL effectiveness in multi-disciplinary applications. Real-time RL enhancements are also valuable, as implementing

real-time KG updates can support dynamic RL scenarios, where entity relationships evolve over time.

The integration of KGs and Data Lakes in RL presents a transformative opportunity to enhance entity resolution across various domains. By leveraging structured domain knowledge, KGs can improve RL accuracy, scalability, and explainability. Future research should focus on optimizing KG-driven RL methodologies, ensuring efficient and adaptable solutions for evolving data landscapes.

11.2.1.2 Advancing Data Lake and Data Mesh enhancing Interaction, Intelligence, and Governance

Another area of work in progress revolves around extending the proposed Data Lake and Data Mesh architectures by integrating advanced methodologies for enhanced data interaction, security, and process automation. The following outline key developments in progress, inspired by recent research efforts.

Enhancing interaction with Data Lakes is a key focus of ongoing research, particularly through the integration of Digital Twins (DTs) and semantic blueprints. Digital Twins provide a virtual representation of data sources and processes, enabling real-time simulations and interactive visualization. By embedding model-based simulations and graphical dashboards, this framework facilitates enhanced user interaction with Data Lakes, reducing the technical expertise required for data retrieval and decision support. The integration of semantic blueprints further improves the efficiency of data access and governance, ensuring structured metadata enrichment.

Our current work involves extending the DT framework to support real-time process monitoring and adaptive analytics. This development is particularly relevant in manufacturing and industrial IoT environments, where high-frequency data streams require efficient processing and visualization. The use of Digital Twins in these environments enables predictive maintenance, process optimization, and enhanced operational insights. Additionally, ongoing efforts are being directed towards integrating Augmented Reality (AR) interfaces with Data Lakes. This enhancement aims to allow immersive data interaction, offering dynamic exploration capabilities that improve the accessibility and usability of large-scale datasets.

Another significant area of research focuses on process mining in Large Language Model (LLM)-driven Data Meshes. Data Meshes represent a decentralized approach to data management, distributing ownership across various domains. Integrating LLMs with Data Meshes facilitates improved data domain discovery and process mining by enabling automatic classification and semantic metadata structuring. The application of clustering techniques within Data Meshes enhances process-to-domain mapping, ensuring efficient governance and compliance with regulatory frameworks.

Current research efforts are dedicated to optimizing the clustering techniques applied to Data Meshes, refining the automatic classification mechanisms to improve data usability. This work is particularly relevant for multi-tenant data environments, where the complexity of shared data structures necessitates advanced behavioral analytics. Future research aims to extend these capabilities to healthcare and automated workflow applications, leveraging LLM-driven insights to optimize data integration and process efficiency.

The integration of Data Lakes with self-adaptive serious games is another area of active exploration. Serious games, which are designed for educational, training, and therapeutic applications, generate vast amounts of heterogeneous data. Data Lakes provide an effective solution for managing this data, enabling real-time analytics to personalize user experiences and track learning outcomes. By leveraging semantic metadata, serious games can dynamically adapt to user interactions, optimizing engagement and effectiveness.

Ongoing research efforts are centered on the development of a metadata-driven recommendation engine for serious games. This engine adjusts game difficulty dynamically based on real-time user data, enhancing adaptability and personalization. Future work will focus on semantic-driven learning analytics to refine the adaptability of serious games, particularly in applications such as speech therapy and cognitive training. These advancements aim to improve rehabilitation outcomes and enhance the effectiveness of digital therapeutic interventions.

Our ongoing research focuses on the Data Domain Design, specifically an Interview Protocol prepared, which aims to ensure consistency across interviews and enable systematic insight gathering on how organizations define, structure, and govern their data domains. By employing a standardized set of questions, we seek to minimize interviewer bias and facilitate comparative analysis across different organizations.

Our methodology aims to integrate expert discussions, literature reviews, and structured interviews to identify best practices, challenges, and emerging trends in data domain design. Through this approach, we are working to enhance the understanding of decentralized data management, data domains within Data Mesh architectures, aligning technical, organizational, and business perspectives to support the development of scalable, efficient, and autonomous data ecosystems.

By advancing these research directions, this work seeks to create a next-generation data ecosystem that is intelligent, secure, and adaptable. The integration of Digital Twins, LLM-driven insights, and semantic metadata governance offers promising opportunities for enhancing data management and usability across various domains. Future efforts will focus on refining these frameworks, exploring federated learning for privacy-preserving analytics, and leveraging graph neural networks to enhance semantic data discovery and predictive insights. The ultimate goal is to establish a scalable, secure, and efficient data architecture that supports diverse applications, from industrial automation to healthcare and adaptive learning systems.

11.2.2 Future Research Steps

There is ample room for further research that stems from the outputs of the present thesis.

Building upon the advancements achieved thus far, future research will explore the role of LLMs in transforming Data Lakes into Data Meshes. Specifically, we will seek to investigate how LLMs can facilitate the automated enrichment of metadata within both Data Lakes and Data Meshes, thereby improving data structuring, retrieval efficiency, and governance mechanisms. The integration of LLMs for intelligent metadata management will enhance semantic organization, enabling more efficient data discovery and interoperability across decentralized systems.

A deeper examination of LLM-driven metadata enrichment will focus on understanding how machine learning models can contextualize raw data, automate schema evolution, and optimize data categorization across various domains. This approach will not only streamline metadata curation but will also introduce new methodologies for handling unstructured and semi-structured data within evolving data ecosystems. Moreover, the relevant research will investigate the potential of LLMs in dynamically adapting metadata taxonomies to support cross-domain data usability and governance.

Additionally, emphasis will be placed on the development of recommendation systems to optimize data insertion, retrieval, and transformation in Data Lakes and Data Meshes. By leveraging context-aware recommendation models, we aim to facilitate automated data source identification, classification, and integration, reducing the complexity of managing vast and heterogeneous datasets.

Further investigation will also explore intelligent recommendation algorithms that aid in the selection and prioritization of data sources, ensuring that Data Lakes remain efficiently organized and scalable. Additionally, recommender systems will be investigated for their ability to assist in the generation and selection of Data Products, enhancing the usability and accessibility of domain-specific datasets within Data Meshes. These algorithms will be tested in real-world scenarios to ensure they accurately align with user needs and data governance policies.

Future research directions will include studying how LLM-enhanced recommendation systems can facilitate automated decision-making in data ingestion and retrieval processes, effectively reducing human intervention while improving the accuracy and efficiency of data utilization. The adoption of adaptive machine learning models for predictive analytics in recommendation systems will also be explored, allowing for a more dynamic and responsive approach to data management within decentralized architectures.

By advancing the above research directions, this work aspires to establish a next-generation data ecosystem that is intelligent, secure, and adaptable. The convergence of Digital Twins, LLM-driven insights, and semantic metadata governance presents significant opportunities for the evolution of data management and usability across multiple domains. Future research will focus on the refinement of these frameworks, the exploration of federated learning for privacy-preserving analytics, and the implementation of graph neural networks to augment semantic data discovery and predictive insights. The overarching objective is the development of a scalable, secure, and efficient data architecture that supports a diverse array of applications, ranging from industrial automation to healthcare and adaptive learning systems.

REFERENCES

Aceto, G., Botta, A., De Donato, W., & Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks*, *57*(9), 2093-2115. 10.1016/j.comnet.2013.04.001.

Al-Sai, Z. A., Husin, M. H., Syed-Mohamad, S. M., Abdin, R. M. D. S., Damer, N., Abualigah, L., & Gandomi, A. H. (2022). Explore big data analytics applications and opportunities: A review. *Big Data and Cognitive Computing*, *6*(4),157. 10.3390/bdcc6040157.

Amount of Data Created Daily. (2024). Exploding Topics. https://explodingtopics.com/blog/data-generated-per-day [Accessed 30-07-2024].

Awan, U., Shamim, S., Khan, Z., Zia, N., Shariq, S., & Khan, M. (2021). Big data analytics capability and decision-making: The role of data-driven insight on circular economy performance. *Technological Forecasting and Social Change*. 10.1016/J.TECHFORE.2021.120766

Azvine, B., Cui, Z., Nauck, D. D., & Majeed, B. (2006, June). Real time business intelligence for the adaptive enterprise. In *The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)* (pp. 29-29). IEEE.

Azvine, B., Cui, Z., & Nauck, D. D. (2005). Towards real-time business intelligence. *BT Technology Journal*, *23*(3), 214-225.

Barnaghi, P., Sheth, A., & Henson, C. (2013). From data to actionable knowledge: Big data challenges in the web of things [Guest Editors' Introduction]. *IEEE Intelligent Systems*, *28*(6), 6-11.

Bashir, M. R., & Gill, A. Q. (2016, December). Towards an IoT big data analytics framework: smart buildings systems. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 1325-1332). IEEE.

Beckman, J. (2023). 30 impressive big data statistics for 2023. TechReport. https://techreport.com/statistics/science/big-data-statistics/ [Accessed 23-07-2024].

Beheshti, A., Benatallah, B., Nouri, R., & Tabebordbar, A. (2018). CoreKG: a knowledge lake service. *Proceedings of the VLDB Endowment*, *11*(12), 1942-1945.

Bertino, E. (2013, July). Big Data--Opportunities and Challenges Panel Position Paper. In *2013 IEEE 37th Annual Computer Software and Applications Conference* (pp. 479-480). IEEE Computer Society.

Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, *66*(3), 1025-1044.

Blazquez, D., & Domenech, J. (2018). Big Data sources and methods for social and economic analyses. *Technological Forecasting and Social Change*, *130*, 99-113. 10.1016/J.TECHFORE.2017.07.027.

Bock, C., Zha, X., Suh, H. W., & Lee, J. H. (2010). Ontological product modeling for collaborative design. *Advanced Engineering Informatics*, *24*(4), 510-524.

Borkar, V. R., Bu, Y., Carey, M. J., Rosen, J., Polyzotis, N., Condie, T., ... & Koenigstein, N. (2012). Declarative Systems for Large-Scale Machine Learning. *IEEE Data Eng. Bull.*, *35*(2), 24-32.

Cafarella, M. J. (2009). *Extracting and managing structured web data*. University of Washington.

Cafarella, M. J., Halevy, A., & Khoussainova, N. (2009). Data integration for the relational web. *Proceedings of the VLDB Endowment*, *2*(1), 1090-1101.

Charest, M., & Delisle, S. (2006, June). Ontology-guided intelligent data mining assistance: Combining declarative and procedural knowledge. In *Artificial Intelligence and Soft Computing* (Vol. 2006, pp. 9-14).

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 1165-1188.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, *19*, 171-209.

Chungoora, N., Young, R. I., Gunendran, G., Palmer, C., Usman, Z., Anjum, N. A., ... & Case, K. (2013). A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. *Computers in industry*, *64*(4), 392-401.

Cristaldi, L., Esmaili, P., Gruosso, G., La Bella, A., Mecella, M., Scattolini, R., ... & Tanca, L. (2023, October). The mics project: A data science pipeline for industry 4.0 applications. In *2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)* (pp. 427-431). IEEE.

Cuenca Grau, B., Jimenez-Ruiz, E., Kharlamov, E., & Nenov, Y. (2016). Capturing Industrial Information Models with Ontologies and Constraints.

Data Hygiene Statistics in 2023. (2023). Data Axle USA. https://www.dataaxleusa.com/blog/data-hygiene-statistics/ [Accessed 30-07-2024].

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.

Dehghani, Z. (2020). Data mesh principles and logical architecture. *martinfowler. com*.

Dehghani, Z. Data Mesh: Delivering Data-Driven Value at Scale. Available online: https://www.thoughtworks.com/insights/books/data-mesh (accessed on 12 February 2024).

Dehghani, Z. How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh Available online: https://martinfowler.com/articles/data-monolith-to-mesh.html (accessed on 10 January 2024).

Delen, D., & Demirkan, H. (2013). Data, information and analytics as services. *Decision support systems*, *55*(1), 359-363. http://dx.doi.org/10.1016/j.dss.2012.05.044.

Denno, P., Dickerson, C., & Harding, J. A. (2018). Dynamic production system identification for smart manufacturing systems. *Journal of manufacturing systems*, *48*, 192-203. https://doi.org/10.1016/j.jmsy.2018.04.006.

Derakhshannia, M., Gervet, C., Hajj-Hassan, H., Laurent, A., & Martin, A. (2020). Data lake governance: Towards a systemic and natural ecosystem analogy. *Future internet*, *12*(8), 126. doi:10.3390/FI12080126.

Di Angelo, M., & Salzer, G. (2020, August). Tokens, types, and standards: identification and utilization in Ethereum. In *2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)* (pp. 1-10). IEEE.

Dolhopolov, A., Castelltort, A., & Laurent, A. (2023, May). Trick or treat: Centralized data lake vs decentralized data mesh. In *International Conference on Management of Digital* (pp. 303-316). Cham: Springer Nature Switzerland.

Dong, X. L., Saha, B., & Srivastava, D. (2012). Less is more: Selecting sources wisely for integration. *Proceedings of the VLDB Endowment*, *6*(2), 37-48.

Drabent, W., Eiter, T., Ianni, G., Krennwallner, T., Lukasiewicz, T., & Małuszyński, J. (2009). Hybrid reasoning with rules and ontologies. In *Semantic Techniques for the Web: The REWERSE perspective* (pp. 1-49). Berlin, Heidelberg: Springer Berlin Heidelberg.

Driessen, S. W., Monsieur, G., & Van Den Heuvel, W. J. (2022). Data market design: A systematic literature review. *IEEE access*, *10*, 33123-33153.

Driessen, S., den Heuvel, W. J. V., & Monsieur, G. (2023, October). Promote: A data product model template for data meshes. In *International Conference on Conceptual Modeling* (pp. 125-142). Cham: Springer Nature Switzerland.

Eichler, R., Giebler, C., Gröger, C., Hoos, E., Schwarz, H., & Mitschang, B. (2021, July). Enterprise-wide metadata management: an industry case on the current state and challenges. In *Business Information Systems* (pp. 269-279). https://doi.org/10.52825/bis.v1i.47.

Erkin, Z., Troncoso-Pastoriza, J. R., Lagendijk, R. L., & Pérez-González, F. (2013). Privacy-preserving data aggregation in smart metering systems: An overview. *IEEE Signal Processing Magazine*, *30*(2), 75-86.

Fan, J., Han, F., & Liu, H. (2014). Challenges of big data analysis. *National science review*, *1*(2), 293-314.

Fang, H. (2015). Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* (pp. 820-824). IEEE.

Farid, M., Roatis, A., Ilyas, I. F., Hoffmann, H. F., & Chu, X. (2016). CLAMS: bringing quality to data lakes. In *Proceedings of the 2016 International Conference on Management of Data* (pp. 2089-2092).

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International journal of information management*, *35*(2), 137-144. http://dx.doi.org/10.1016/j.ijinfomgt.2014.10.007.

Garetti, M., Fumagalli, L., Lobov, A., & Lastra, J. M. (2013). Open automation of manufacturing systems through integration of ontology and web services. *IFAC Proceedings Volumes*, *46*(9), 198-203.

Gharaibeh, A., Salahuddin, M. A., Hussini, S. J., Khreishah, A., Khalil, I., Guizani, M., & Al-Fuqaha, A. (2017). Smart cities: A survey on data management, security, and enabling technologies. *IEEE Communications Surveys & Tutorials*, *19*(4), 2456-2501.

Giese, M., Soylu, A., Vega-Gorgojo, G., Waaler, A., Haase, P., Jiménez-Ruiz, E., ... & Rosati, R. (2015). Optique: Zooming in on big data. *Computer*, *48*(3), 60-67.

Giovannini, A., Aubry, A., Panetto, H., Dassisti, M., & El Haouzi, H. (2012). Ontology-based system for supporting manufacturing sustainability. *Annual Reviews in Control*, *36*(2), 309-317. http://dx.doi.org/10.1016/j.arcontrol.2012.09.012.

Guerrero, J. I., García, A., Personal, E., Luque, J., & León, C. (2017). Heterogeneous data source integration for smart grid ecosystems based on metadata mining. *Expert Systems with Applications*, *79*, 254-268. http://dx.doi.org/10.1016/j.eswa.2017.03.007.

Günther, W. A., Mehrizi, M. H. R., Huysman, M., & Feldberg, F. (2017). Debating big data: A literature review on realizing value from big data. *The Journal of Strategic Information Systems*, *26*(3), 191-209.

Gupta, S., Kar, A. K., Baabdullah, A., & Al-Khowaiter, W. A. (2018). Big data with cognitive computing: A review for the future. *International Journal of Information Management*, *42*, 78-89.

Harjunkoski, I., & Bauer, R. (2014). Sharing data for production scheduling using the ISA-95 standard. *Frontiers in Energy Research*, *2*, 44.

Hassanzadeh, R., & Nedovic-Budic, Z. (2012). Identification of earthquake disaster hot spots with crowd sourced data. In *Intelligent Systems for Crisis Management: Geo-information for Disaster Management (Gi4DM) 2012* (pp. 97-119). Berlin, Heidelberg: Springer Berlin Heidelberg.

Herschel, R., & Miori, V. M. (2017). Ethics & big data. *Technology in Society*, *49*, 31-36. https://doi.org/10.1016/j.techsoc.2017.03.003.

Howarth, J. 30+ Incredible Big Data Statistics. 2023. Available online: https://explodingtopics.com/blog/big-data-stats (accessed on 10 February 2024).

Jardim-Goncalves, R., Coutinho, C., Cretan, A., da Silva, C. F., & Ghodous, P. (2014). Collaborative negotiation for ontology-driven enterprise businesses. *Computers in Industry*, *65*(9), 1232-1241. http://dx.doi.org/10.1016/j.compind.2014.01.001.

Jardim-Goncalves, R., Coutinho, C., Cretan, A., da Silva, C. F., & Ghodous, P. (2014). Collaborative negotiation for ontology-driven enterprise businesses. *Computers in Industry*, *65*(9), 1232-1241.

Khan, N., Alsaqer, M., Shah, H., Badsha, G., Abbasi, A. A., & Salehian, S. (2018, March). The 10 Vs, issues and challenges of big data. In *Proceedings of the 2018 international conference on big data and education* (pp. 52-56).

Khine, P. P., & Wang, Z. (2019). A review of polyglot persistence in the big data world. *Information*, *10*(4), 141, 10.3390/INFO10040141.

Khine, P. P., & Wang, Z. S. (2018). Data lake: a new ideology in big data era. In *ITM web of conferences* (Vol. 17, p. 03025). EDP Sciences.

Kim, J. (2017). Partial rollback-based scheduling on in-memory transactional data grids. *Big data research*, *9*, 47-56. http://dx.doi.org/10.1016/j.bdr.2017.06.004.

Kim, Y., You, E., Kang, M., & Choi, J. (2012). Does big data matter to value creation?: Based on oracle solution case. *Journal of Information Technology Services*, *11*(3), 39-48. https://doi.org/10.9716/KITS.2012.11.3.039.

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering–a tertiary study. *Information and software technology*, *52*(8), 792-805.

Friedman, D. K. N. (2009). Probabilistic Graphical Models Principles and Techniques.

Kościelniak, H., & Puto, A. (2015). BIG DATA in decision making processes of enterprises. *Procedia Computer Science*, *65*, 1052-1058. https://doi.org/10.1016/j.procs.2015.09.053.

Kunigk, J., Buss, I., Wilkinson, P., & George, L. (2018). *Architecting modern data platforms: a guide to enterprise hadoop at scale*. " O'Reilly Media, Inc.".

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. *META group research note*, *6*(70), 1.

Lanzenberger, M., Sampson, J., & Rester, M. (2010). Ontology Visualization: Tools and Techniques for Visual Representation of Semi-Structured Meta-Data. *J. Univers. Comput. Sci.*, *16*(7), 1036-1054.

Larson, D. and Chang, V., 2016. A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, *36*(5), pp.700-710. http://dx.doi.org/10.1016/j.ijinfomgt.2016.04.013.

LeDell, E., & Poirier, S. (2020, July). H2o automl: Scalable automatic machine learning. In *Proceedings of the AutoML Workshop at ICML* (Vol. 2020, p. 24).

Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business horizons*, *58*(4), 431-440. http://dx.doi.org/10.1016/j.bushor.2015.03.008.

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters*, *3*, 18-23. http://dx.doi.org/10.1016/j.mfglet.2014.12.001.

Lee, J., Lapira, E., Yang, S., & Kao, A. (2013). Predictive manufacturing system-Trends of next-generation production systems. *Ifac proceedings volumes*, *46*(7), 150-156. http://dx.doi.org/10.3182/20130522-3-BR-4036.00107.

Lee, J., Ardakani, H. D., Yang, S., & Bagheri, B. (2015). Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. *Procedia cirp*, *38*, 3-7. http://dx.doi.org/10.1016/j.procir.2015.08.026.

Lee, J., Kao, H. A., & Yang, S. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia cirp*, *16*, 3-8. http://dx.doi.org/10.1016/j.procir.2014.02.001.

Lemaignan, S., Siadat, A., Dantan, J. Y., & Semenenko, A. (2006, June). MASON: A proposal for an ontology of manufacturing domain. In *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)* (pp. 195-200). IEEE.

Lepenioti, K., Bousdekis, A., Apostolou, D., & Mentzas, G. (2020). Prescriptive analytics: Literature review and research challenges. *International Journal of Information Management*, *50*, 57-70. https://doi.org/10.1016/j.ijinfomgt.2019.04.003.

Li, B. M., Xie, S. Q., & Xu, X. (2011). Recent development of knowledge-based systems, methods and tools for one-of-a-kind production. *Knowledge-Based Systems*, *24*(7), 1108-1119.

Limaye, G., Sarawagi, S., & Chakrabarti, S. (2010). Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, *3*(1-2), 1338-1347.

Lin, H. W., Nagalingam, S. V., Kuik, S. S., & Murata, T. (2012). Design of a global decision support system for a manufacturing SME: towards participating in collaborative manufacturing. *International Journal of Production Economics*, *136*(1), 1-12. http://dx.doi.org/10.1016/j.ijpe.2011.07.001.

Lopes, J., Guimarães, T., & Santos, M. F. (2020). Predictive and prescriptive analytics in healthcare: a survey. *Procedia Computer Science*, *170*, 1029-1034. https://doi.org/10.1016/j.procs.2020.03.078.

Lu, Y., & Xu, X. (2019). Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services. *Robotics and Computer-Integrated Manufacturing*, *57*, 92-102. https://doi.org/10.1016/j.rcim.2018.11.006.

Lu, Z., & Wen, Y. (2013). Distributed algorithm for tree-structured data aggregation service placement in smart grid. *IEEE Systems Journal*, *8*(2), 553-561.

Luckow, A., Kennedy, K., Manhardt, F., Djerekarov, E., Vorster, B., & Apon, A. (2015, October). Automotive big data: Applications, workloads and infrastructures. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 1201-1210). IEEE. https://doi.org/10.1109/BigData.2015.7363874.

Machado, I., Costa, C., & Santos, M. Y. (2021). Data-driven information systems: the data mesh paradigm shift.

Machado, I. A., Costa, C., & Santos, M. Y. (2022). Data mesh: concepts and principles of a paradigm shift in data architectures. *Procedia Computer Science*, *196*, 263-271. https://doi.org/10.1016/j.procs.2021.12.013.

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth, A. P. (2018). Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks*, *4*(3), 161-175. https://doi.org/10.1016/j.dcan.2017.10.002.

Mehboob, T., Ahmed, I. A., & Afzal, A. (2022). Big Data Issues, Challenges and Techniques: A Survey. *Pakistan Journal of Engineering and Technology*, *5*(2), 216-220.

Mehdi, G., Kharlamov, E., Savković, O., Xiao, G., Kalaycı, E. G., Brandt, S., ... & Runkler, T. (2017). Semantic rule-based equipment diagnostics. In *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II 16* (pp. 314-333). Springer International Publishing. http://dx.doi.org/10.1016/j.procir.2014.02.001.

Miloslavskaya, N., & Tolstoy, A. (2016). Big data, fast data and data lake concepts. *Procedia Computer Science*, *88*, 300-305.

O'Donovan, P., Leahy, K., Bruton, K., & O'Sullivan, D. T. (2015). An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. *Journal of big data*, *2*, 1-26.

O'Leary, D. E. (2014). Embedding AI and crowdsourcing in the big data lake. *IEEE Intelligent Systems*, *29*(5), 70-73.

Pang, L. Y., Zhong, R. Y., Fang, J., & Huang, G. Q. (2015). Data-source interoperability service for heterogeneous information integration in ubiquitous enterprises. *Advanced Engineering Informatics*, *29*(3), 549-561. http://dx.doi.org/10.1016/j.aei.2015.04.007.

Panigrahy, S., Dash, B., & Thatikonda, R. (2023). From data mess to data mesh: Solution for futuristic self-serve platforms. International Journal of Advanced Research in Computer and Communication Engineering, 12(4), 677-683.

Papazoglou, M. P., & Elgammal, A. (2017, June). The manufacturing blueprint environment: Bringing intelligence into manufacturing. In *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (pp. 750-759). IEEE.

Papazoglou, M. P., van den Heuvel, W. J., & Mascolo, J. E. (2015). A reference architecture and knowledge-based structures for smart manufacturing networks. IEEE Software, 32(3), 61-69.

Petersen, N., Halilaj, L., Grangel-González, I., Lohmann, S., Lange, C., & Auer, S. (2017). Realizing an RDF-based information model for a manufacturing company–a case study. In *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II 16* (pp. 350-366). Springer International Publishing.

Petersen, N., Galkin, M., Lange, C., Lohmann, S., & Auer, S. (2016). Monitoring and automating factories using semantic models. In *Semantic Technology: 6th Joint International Conference, JIST 2016, Singapore, Singapore, November 2-4, 2016, Revised Selected Papers 6* (pp. 315-330). Springer International Publishing.

Phuc, N. T., Khanh, H. V., Khoa, T. D., Khiem, H. G., Huong, H. L., Triet, N. M., ... & Quy, L. T. (2023). An Enhanced CoD System Leveraging Blockchain, Smart Contracts, and NFTs: A New Approach for Trustless Transactions. *International Journal of Advanced Computer Science and Applications*, *14*(10).

Polyvyanyy, A., Ouyang, C., Barros, A., & van der Aalst, W. M. (2017). Process querying: Enabling business intelligence through query-based process analytics. *Decision Support Systems*, *100*, 41-56. http://dx.doi.org/10.1016/j.dss.2017.04.011.

Qin, Y., Sheng, Q. Z., Falkner, N. J., Dustdar, S., Wang, H., & Vasilakos, A. V. (2016). When things matter: A survey on data-centric internet of things. *Journal of Network and Computer Applications*, *64*, 137-153. http://dx.doi.org/10.1016/j.jnca.2015.12.016.

Rehman, W., e Zainab, H., Imran, J., & Bawany, N. Z. (2021, December). NFTs: Applications and challenges. In *2021 22nd International Arab Conference on Information Technology (ACIT)* (pp. 1-7). IEEE.

Roda, F., & Musulin, E. (2014). An ontology-based framework to support intelligent data analysis of sensor measurements. *Expert Systems with Applications*, *41*(17), 7914-7926. http://dx.doi.org/10.1016/j.eswa.2014.06.033.

Rodríguez-Mazahua, L., Rodríguez-Enríquez, C. A., Sánchez-Cervantes, J. L., Cervantes, J., García-Alcaraz, J. L., & Alor-Hernández, G. (2016). A general perspective of Big Data: applications, tools, challenges and trends. *The Journal of Supercomputing*, *72*, 3073-3113. https://doi.org/10.1007/s11227-015-1501-1.

Roh, Y., Heo, G., & Whang, S. E. (2019). A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), 1328-1347.

Rusitschka, S., Eger, K., & Gerdes, C. (2010, October). Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain. In *2010 first IEEE international conference on smart grid communications* (pp. 483-488). IEEE.

Sahay, B. S., & Ranjan, J. (2008). Real time business intelligence in supply chain analytics. *Information Management & Computer Security*, *16*(1), 28-48.

Sakr, S., & Elgammal, A. (2016). Towards a comprehensive data analytics framework for smart healthcare services. *Big Data Research*, *4*, 44-58. http://dx.doi.org/10.1016/j.bdr.2016.05.002.

Saldivar, A. A. F., Goh, C., Li, Y., Chen, Y., & Yu, H. (2016, September). Identifying smart design attributes for Industry 4.0 customization using a clustering Genetic Algorithm. In *2016 22nd international conference on automation and computing (ICAC)* (pp. 408-414). IEEE.

Santos, M. Y., & Costa, C. (2020). 2 Big Data Concepts, Techniques, and Technologies.

Sarma, A. D., Fang, L., Gupta, N., Halevy, A. Y., Lee, H., Wu, F., & Yu, C. (2012). Finding related tables. In *SIGMOD Conference* (Vol. 10, pp. 2213836-2213962).

Sawadogo, P. N., Scholly, E., Favre, C., Ferey, E., Loudcher, S., & Darmont, J. (2019). Metadata systems for data lakes: models and features. In *New Trends in Databases and Information Systems: ADBIS 2019 Short Papers, Workshops BBIGAP, QAUCA, SemBDM, SIMPDA, M2P, MADEISD, and Doctoral Consortium, Bled, Slovenia, September 8–11, 2019, Proceedings 23* (pp. 440-451). Springer International Publishing.

Sawadogo, P., & Darmont, J. (2021). On data lake architectures and metadata management. *Journal of Intelligent Information Systems*, *56*(1), 97-120.

Sethi, P., & Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of electrical and computer engineering*, *2017*(1), 9324035.

Shahid, A., Nguyen, T. A. N., & Kechadi, M. T. (2021). Big data warehouse for healthcare-sensitive data applications. Sensors, 21(7), 2353,10.3390/S21072353.

Stimmel, C. L. (2015). *Big data analytics strategies for the smart grid* (pp. 155-169). Boca Raton: CRC press.

Stojmenovic, I. (2014). Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems. *IEEE Internet of Things Journal*, *1*(2), 122-128.

Sun, J. (2018). *Smart services for enhancing personal competence in Industrie 4.0 digital factory. LogForum 14 (1), 51-57.*

Tang, D., Zheng, K., Zhang, H., Zhang, Z., Sang, Z., Zhang, T. & Vargas-Solar, G. (2018). Using autonomous intelligence to build a smart shop floor. *The International Journal of Advanced Manufacturing Technology*, *94*, 1597-1606.

Tao, F., Cheng, Y., Da Xu, L., Zhang, L., & Li, B. H. (2014). CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system. *IEEE Transactions on industrial informatics*, *10*(2), 1435-1442.

Terkaj, W., & Urgo, M. (2015). A virtual factory data model as a support tool for the simulation of manufacturing systems. *Procedia CIRP*, *28*, 137-142. http://dx.doi.org/10.1016/j.procir.2015.04.023.

Tran, D., Hoffman, M. D., Saurous, R. A., Brevdo, E., Murphy, K., & Blei, D. M. (2017). Deep probabilistic programming. *arXiv preprint arXiv:1701.03757*.

Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2019). Implementing AutoML in educational data mining for prediction tasks. *Applied Sciences*, *10*(1), 90.

VAN DEN HEUVEL, J. A. N., MONSIEUR, G., TAMBURRI, D. A., & DI NUCCI, D. A. R. I. O. (2023). Data Mesh: a Systematic Gray Literature Review.

Van Der Aalst, W. M., & Dustdar, S. (2012). Process mining put into context. *IEEE Internet Computing*, *16*(1), 82-86.

Van Der Aalst, W. M., Reijers, H. A., Weijters, A. J., van Dongen, B. F., De Medeiros, A. A., Song, M., & Verbeek, H. M. (2007). Business process mining: An industrial application. *Information systems*, *32*(5), 713-732.

Venetis, P., Halevy, A., Madhavan, J., Pasca, M., Shen, W., Wu, F., & Wu, C. (2011). Recovering semantics of tables on the web.

Viriyasitavat, W., Da Xu, L., Bi, Z., & Hoonsopon, D. (2019). Blockchain technology for applications in internet of things—mapping from system design perspective. *IEEE Internet of Things Journal*, *6*(5), 8155-8168.

Vlasiuk, Y., & Onyshchenko, V. (2023, March). Data Mesh as Distributed Data Platform for Large Enterprise Companies. In International Conference on Computer Science, Engineering and Education Applications (pp. 183-192). Cham: Springer Nature Switzerland.

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of manufacturing systems*, *48*, 144-156. https://doi.org/10.1016/j.jmsy.2018.01.003.

Wang, J., Fu, P., & Gao, R. X. (2019). Machine vision intelligence for product defect inspection based on deep learning and Hough transform. *Journal of Manufacturing Systems*, *51*, 52-60. https://doi.org/10.1016/j.jmsy.2019.03.002.

Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer networks*, *101*, 158-168.

Wang, Y., Kung, L., & Byrd, T. A. (2018). Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological forecasting and social change*, *126*, 3-13. https://doi.org/10.1016/j.techfore.2015.12.019.

Wieder, P., & Nolte, H. (2022). Toward data lakes as central building blocks for data management and analysis. *Frontiers in big Data*, *5*, 945720.

Wohlin, C. (2014, May). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (pp. 1-10).

Wrembel, R. (2023, August). Data integration revitalized: From data warehouse through data lake to data mesh. In *International Conference on Database and Expert Systems Applications* (pp. 3-18). Cham: Springer Nature Switzerland.

Yakout, M., Ganjam, K., Chakrabarti, K., & Chaudhuri, S. (2012, May). Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (pp. 97-108).

Yan, Y., Qian, Y., & Sharif, H. (2011, December). A secure data aggregation and dispatch scheme for home area networks in smart grid. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011* (pp. 1-6). IEEE.

Yang, D., Dong, M., & Miao, R. (2008). Development of a product configuration system with an ontology-based approach. *Computer-Aided Design*, *40*(8), 863-878.

Yildiz, H., Küpper, A., Thatmann, D., Göndör, S., & Herbke, P. (2023). Toward interoperable self-sovereign identities. *IEEE Access*, *11*, 114080-114116.

Yu, C., & Boyd, J. (2016). FB+-tree for big data management. *Big Data Research*, *4*, 25-36. http://dx.doi.org/10.1016/j.bdr.2015.11.003.

Yuhanna, N., Leganza, G., & Lee, J. (2017). The Forrester Wave™: Big Data Warehouse, Q2 2017. *Adoption Grows As Enterprises Look To Revive Their EDW Strategy*, 17.

Zhang, Y., Yu, R., Nekovee, M., Liu, Y., Xie, S., & Gjessing, S. (2012). Cognitive machine-to-machine communications: Visions and potentials for the smart grid. *IEEE network*, *26*(3), 6-13.

Zhang, Y., Wang, W., Du, W., Qian, C., & Yang, H. (2018). Coloured Petri net-based active sensing system of real-time and multi-source manufacturing information for smart factory. *The International Journal of Advanced Manufacturing Technology*, *94*, 3427-3439.

Zhong, R. Y., Huang, G. Q., Lan, S., Dai, Q. Y., Chen, X., & Zhang, T. (2015). A big data approach for logistics trajectory discovery from RFID-enabled production data. *International Journal of Production Economics*, *165*, 260-272.