Journal of Theoretical and Applied Information Technology
15th November 2024. Vol.102. No. 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DLMF: AN INTEGRATED ARCHITECTURE FOR HEALTHCARE DATA MANAGEMENT AND ANALYSIS USING DATA LAKE, DATA MESH, AND DATA FABRIC

LAMYA OUKHOUYA[1] , ANASS EL HADDADI[2], BRAHIM ER-RAHA[1], ASMA SBAI[3]

[1]ESTIDMA team, National School of Applied Sciences, Agadir, Morocco

[2]SDIC team, National School of Applied Sciences, Al Hoceima, Morocco

[3]LBH Laboratory, Faculty of Medicine and Pharmacy, Marrakech, Morocco

E-mail: [1]l.oukhouya@uiz.ac.ma, [2]a.elhaddadi@uae.ac.ma, [1]b.raha@uiz.ac.ma , [3]asma.sbai@uca.ac.ma

ABSTRACT

The vast amount of data on healthcare, combined with the diversity of diseases, has led to a proliferation of work aimed at designing decision-making architectures capable of exploiting this information. These architectures are based on integrating heterogeneous data from different sources while ensuring that it is stored centrally. They also focus on data quality to guarantee the accuracy of analyses, and on reliable governance to ensure data compliance, security and traceability. These aspects are essential to enable optimal use of data for advanced analysis and informed decision-making in healthcare. Our objective in this article is to propose an architecture that ensures the 3 points: integration, storage and governance. The article proposes the DLMF architecture to ensure that these 3 points are adapted to good data analysis. This architecture uses the concept of a data lake for the consolidation and storage of data sources, a data mesh and a data fabric to ensure everything to do with data integration, quality and governance. The article also proposes a set of open-source technologies for its implementation. Finally, recommendations and future directions are suggested for a well-designed BI architecture capable of ensuring good data management, from data integration to analysis.

Keywords: *Data Lake, Data Mesh, Data Fabric, Healthcare, Decision Support Architecture.*

1. INTRODUCTION

Information systems play a crucial role in managing data from a variety of sources in all sectors[1]. The health sector is no exception, relying on a range of systems of resources, people, policies and technologies to deliver health services to the population[2]. These systems cover a range of activities, including disease prevention and public health management. Extracting value is based on exploiting this data to improve clinical outcomes, optimize processes and support decision-making[3]. The integration of big data technologies and artificial intelligence, together with the implementation of sophisticated and modern decision support systems, make it possible to transform raw data into actionable information, thereby improving the personalization of care, the prediction of public health needs, cost reduction and the patient experience[4]. The aim of this paper is to propose an architecture for a modern medical decision support system, capable of efficiently managing large amounts of data, providing in-depth analysis and supporting informed decision-making in medicine.

A medical decision support system is based on a complex, structured architecture made up of several interconnected layers or domains, each with a specific mission in the management of medical data[5]. In general, this architecture is based on four main missions, each represented by a layer. The first is to collect data from different sources within the medical information system, such as electronic medical records, medical images, connected devices and external databases. This data is then ingested by an integration layer, where it is harmonized and standardized to ensure consistency. Once normalized, the data is stored in a storage layer, which can be a data warehouse or a data lake, for later use in the decision-making process, quickly and securely. This use relies on advanced algorithms, such as artificial intelligence, to process the data, identify patterns, make predictions and generate clinical recommendations. The final mission of the BI architecture is to represent this data, providing interactive dashboards and visualizations, making it easier for healthcare professionals to make informed decisions[6].

The design of decision support architectures in scientific research, particularly in healthcare, presents a number of complex

challenges[7]. The integration of heterogeneous and voluminous data, often from different sources, poses difficulties in terms of standardization, which can lead to errors. Data quality is essential to the reliability of analyses and requires cleaning and validation efforts[8]. In addition, interoperability between existing systems is limited by the lack of unified standards. The protection of sensitive data, particularly in the healthcare field, requires robust security measures, while the scalability of DSS must allow for managing a growing volume of data without compromising performance[9]. Our mission is to propose an architecture that responds to these difficulties, integrating data lake as a centralized approach to data with basic concepts of a decentralization approach to data i.e. data mesh, data product, and data fabric.

A centralized data approach is used to consolidate data from different sources in one central point for easy management, analysis, and use[10]. The main tools associated with this approach include a data warehouse, data mart, data lake, and lakehouse, each offering specific benefits. A centralized approach optimizes analytics by consolidating data, reducing redundancy, improving security and compliance, providing greater flexibility and scalability, and maximizing the value of data while optimizing resources and reducing associated costs[11]. However, despite these advantages, a centralized approach may suffer from several disadvantages. Using a data lake can lead to complex management and performance issues, due to raw data storage and the difficulty of implementing robust security policies. A data warehouse, although optimized for analysis, can be costly and rigid to change. Data marts can create information silos and additional maintenance costs, while a lakehouse, albeit flexible, can present complex scalability and integration challenges[12].

In light of these limitations, decentralized approaches to data have emerged. They are characterized by the distribution of data management and access across several units or teams, rather than centralizing them in one place[13]. These approaches use various concepts; including data product, data mesh, and data fabric which offer several advantages such as the use of data products, hence improving the relevance and quality of the data by focusing on the needs of end users, and offering greater flexibility, team autonomy and increased ability to meet specific user needs, while reducing costs and information silos. Data mesh not only promotes team autonomy and scalability by distributing data management, while maintaining consistency across the organization and data fabric, but also facilitates unified access and data management in decentralized environments, and improves the flexibility and connectivity of data systems. The aim of this article is to propose a hybrid architecture that retains the advantages of existing approaches while addressing the aforementioned limitations. Specifically, the goal is to design an architecture combining centralized and decentralized data management models. We will integrate data lakes and address their shortcomings by adopting three decentralization approaches: Data Products, Data Mesh, and Data Fabric. It is worth noting that the concept of data products has been implicitly incorporated into the data mesh.

This article is organized as follows: Section 2 outlines the theoretical framework of our architecture, Section 3 presents related works, Section 4 details our proposed DLMF architecture, Section 5 discusses the implementation of this architecture, and finally, section 6 provides the conclusion.

2. THEORETICAL FRAMEWORK : DATA LAKE, DATA MESH, AND DATA FABRIC

As part of a BI architecture for health data management, the concepts of data lake, data mesh, and data fabric play essential and interdependent roles. Each of these concepts is involved in creating a robust and scalable data infrastructure that can meet the complex needs of healthcare organizations. This section presents these concepts in detail and describes how they can be integrated to create a highly effective data management architecture.

2.1 Data Lake: A Centralized Approach to Data Management

A data lake is a centralized repository that stores data from heterogeneous sources, whether structured, semi-structured, or unstructured[14]. Unlike a data warehouse, which uses ETL (Extraction, Transformation, Loading) processes to clean, transform, and organize data into a fixed model before storage, the data lake applies ELT processes (Extraction, Loading, Processing)[15]. In this model, data is extracted and loaded in its raw format, with transformations applied only when requested by an application for analytical purposes[16]. The design of a data lake is based on a functional architecture composed of several zones, each dedicated to specific tasks such as integration, storage, processing, and governance of data[17]. Although the data lake has overcome some limitations of data warehouses, such as rigid schemas, limited capacity for data analysis, and high storage costs, it also has a set of restrictions. These

include poor data quality, complex governance, low performance in accessing data, difficulties in ensuring data security, lack of structure that can slow down decision-making, and high cost of resources and technologies needed to meet these challenges[18].

2.2 Data Mesh and Data Product: A Decentralized Approach to Data Management

Data Mesh is a decentralized approach to data management, organized around specific areas, where each team is responsible for managing its data as a product. This approach is based on four key principles: distributed data architecture, domain-based self-management of data, self-service infrastructure, and federated governance[18].The concept of Data Product is at the heart of this philosophy[19]. In a Data Mesh, each domain treats its data as a product, which means that teams are responsible not only for the collection and management of data, but also for their quality, transformation, and the provision of other data domains or users. A Data Product is designed to be reusable, reliable, and interoperable, meeting the specific needs of end users within the organization. This allows for greater agility and adaptability to specific domain needs but also introduces new complexities in terms of governance and coordination between domains[20]. While the Data Mesh and Data Product concepts address some of the limitations of centralized approaches, such as over-centralization and bottlenecks in data processing, they also bring their challenges. Decentralization can lead to duplication of effort and inconsistent practices if the principles of federated governance are not rigorously applied. In addition, managing a self-service infrastructure and designing robust data products can require significant technological and human investments, and coordination between teams to ensure data quality and security, which remains a challenge[21]. Thus, the Data Mesh and the Data Product concept favor decentralization and autonomy of business teams, bringing their advantages and limitations according to the needs of the organization.

2.3 Data Fabric: A Centralized approach to data management

The Data Fabric is an integrated architecture that facilitates access, management, and processing of data across a distributed and heterogeneous environment. It creates a unified layer that connects disparate data sources, whether on-premise, in the cloud, or hybrid environments[22]. The Data Fabric is distinguished by its ability to provide centralized management of security, governance, and data quality while allowing seamless access to end users through analytics and data management tools[23]. The Data Fabric centralizes orchestration and governance while connecting data sources across a global infrastructure. This architecture ensures consistency and consistent governance while providing a unified view of data across the organization. The Data Fabric excels in system interoperability, allowing for seamless integration of data from different sources and formats without requiring significant redesign. However, this centralization can sometimes limit agility and flexibility, especially in environments where needs differ significantly between domains. In short, the Data Fabric centralizes data management while providing seamless connectivity and integration across multiple systems. It offers a powerful solution for data management, but its effectiveness will depend on the specific needs in terms of centralization, flexibility, and governance within the organization.

Table 1 presents a comparative summary of the characteristics of each concept .

Table1. A Comparison of The Characteristics of The Following Concepts: Data Lake, Data Mesh, Data Product and Data Fabric

Characteristics	Data Lake	Data Mesh	Data Fabric	Data Product
Approach	Centralized	Decentralized by domain	Centralized with distributed integration	Decentralized, end-user-centered
Governance	Centralized, often complex	Federated, decentralized by domain	Centralized, with global control	Governance by product, with defined standards

Data management	Storage of raw data	Data is managed as products by domain.	Unified management with transparent access	Data is managed as a product, ready for use.
Data Transformation	Post-storage Transformation (ELT)	Transformation managed by each domain	Integrated and accessible transformation	Integrated, product-specific transformation
Flexibility	Limited by centralization	Very flexible, each domain adapts its needs	Flexible, but with strict governance	Very flexible, user-centered
interoperability	Less interoperable between systems	Depends on coordination between the domains	Highly interoperable, connects various sources	High interoperability, designed for sharing and reuse
Agility	Less agile, slow transformation process	Very agile, each domain controls its processes	Moderate, fast agility for source integration	Very agile, allows fast iterations according to needs
Scalability	Highly scalable, suitable for large volumes	Domain-dependent scalability	Highly scalable, but with complex management	Scalable to products, adaptable to scale
Cost	Less expensive for raw storage	May be costly due to the complexity	Potentially high cost due to integrated technology	Variable cost depends on the sophistication of the product
Security	Centralized security but difficult to manage	Security managed by each domain, complex to federate	Centralized security with robust controls	Security integrated into each product, with standards

Based on this table, it can be concluded that the Data Lake is ideal for the centralized management of large amounts of raw data, offering scalable low-cost storage and easy integration with big data systems. Data Mesh, on the other hand, favors decentralization, allowing management of the data by domain, which promotes agility and personalization, while maintaining federated governance. Finally, the Data Fabric is distinguished by its interoperability and its ability to provide a unified view of data, transparently integrating various sources while ensuring centralized governance.

Therefore, we see that the integration of these three concepts in a complementary way offers us a hybrid architecture for data management to meet the different needs of organizations in terms of data management. In the following section, we will detail our proposal by describing the role that each concept plays in data management.

3. STATE OF ART

The design of decision support architectures is based on several essential components, capable of efficiently managing and storing data. In this section, we first review available works on data lake design, then present those integrating Data Mesh and Data Fabric into these architectures, which we will then compare to our approach.

In this context, the work [24] proposes a decision-making architecture for the medical domain called HEALLER. This architecture is composed of several layers: ingestion, storage, processing, and consumption. The ingestion area allows the extraction and integration of data in batch mode in the data lake, without transformation, using the first phase of the ELT (Extraction-Loading-Transformation) process. The data is then centralized in the storage area, composed of two modules: one for storing raw data and the other for refining it. The third area prepares and processes the

data stored in the lake according to the needs of users or applications. Finally, the fourth area is dedicated to the consumption of the stored and processed raw data.

In the same vein, article [25] proposes an architecture dedicated to the optimization of data management, taking agriculture as a case study. It presents a migration strategy for various data sources, mainly powered by IoT devices, focusing on storage and analytics aspects. The IISOBA (Smart Systems Oriented Big Data Architecture) addresses the challenges of managing massive data in a Big Data environment. It consists of three main layers: the batch layer, which centralizes raw data in an initial storage area; the structured layer, which organizes and cleans these data to integrate them into a data lake under a snowflake logic model, based on Hadoop and HDFS; and the refinement layer, which transforms data and stores it as Data Marts, accessible via visualization tools.

Regarding Lakehouse architectures, work [26] merges the concepts of a data lake and data warehouse. This architecture is multi-layered: the first uses Apache Avro and Parquet for storage; the second focuses on metadata management, including cataloging and indexing; the third optimizes queries through SQL engines such as Hive and Presto; finally, the fourth provides access management, security, and data encryption.

Furthermore, other works focus on the evolution of these architectures towards approaches integrating the concepts of Data Mesh and Data Fabric.

Article [27] proposes a distributed data platform architecture, highlighting the Data Mesh as the main approach. It first analyzes centralized data platforms and their limitations in terms of data management, in particular, due to the increase in complexity and costs. It then presents the advantages of distributed platforms, in particular the Data Mesh, highlighting aspects such as syndication and data quality, while emphasizing the need for additional research to ensure a successful adoption of this architecture.

Finally, article [28] proposes a combined approach of Data Fabric and Data Mesh to address the challenges of centralization and data management. Although these concepts may seem opposite, the authors suggest that they should be integrated to complement each other. The architecture presented includes data domains, such as research institutions or public and private organizations, that provide datasets in the form of data products. A semantic layer, implemented via the Data Fabric, provides a unified view of the data while allowing flexible access to source sources. A governance layer sets the standards for the entire Data Mesh and manages the ontology that feeds the semantic layer, ensuring smooth integration of data while maintaining the flexibility to meet specific access needs.

Based on these works, and to assess the advantages and disadvantages of its different approaches to data management and analysis in complex environments such as health care and other data-intensive areas, the following table 2 compares them according to certain characteristics.

Table 2. Comparison of Data Lake, Data Mesh, and Data Fabric Concepts

Characteristics	Data Lake	Data Mesh	Data Fabric
Architecture	Centralized with distinct layers: ingestion, storage, process, and consumption.	Decentralized, each domain is responsible for its data and governance.	Distributed with a semantic integration layer providing a unified view of the data.
Data model	Raw data is stored in dedicated areas (ex: raw and refined data).	Each domain produces consumable **data products** from other domains.	Data is integrated and enriched in a unified **semantic layer**.
Data management	Centralized data, with post-storage processing (ELT).	Each domain manages its data independently, with decentralized management.	Automated data management across distributed sources via rich metadata and centralized governance.
Data ingestion	Data ingested in **batch** without transformation, centralized in the data lake.	Decentralized ingestion with API interfaces for each domain.	Automatic ingestion with real-time integration via a semantic layer.
Data Transformation	Data is refined and prepared on demand in the processing area.	Each area is responsible for transforming the data to meet specific consumer needs.	Transformation of data through automatic processes within the semantic layer.

Storage	Separate storage for raw and refined data, with HDFS and Parquet.	Storage is managed by each domain, often with a technology adapted to its needs (e.g.: distributed systems).	Distributed and connected storage across multiple platforms, providing a unified view.
Governance	Centralized, requiring global processes for security and compliance.	Local governance by domain, with global standards to ensure interoperability.	Automated governance through centralized metadata and rules at the system level.
Data Quality	Data quality is ensured downstream during processing or consumption.	Each domain is responsible for the quality of its data products.	Quality is managed in real-time through automated data enrichment and validation processes.
Security and Compliance	Centralized management with global policies for access control and security.	Decentralized security, each domain applies its security measures, under common guidelines.	Centralized management of security and compliance across the governance layer, with real-time access controls.
Access to data	Centralized access, requires complex processes to extract specific information.	Decentralized access via APIs or interfaces specific to each domain.	Real-time access, unified by the semantic layer, offering flexibility and speed.
Scalability	Highly scalable for storage, but difficult to manage with increasing data volumes.	Highly scalable thanks to distributed and decentralized data management.	Scalable thanks to its distributed model and hybrid architecture (cloud and on-premises).
Data consumption	Data consumed after processing in the dedicated area (consumption area).	**Data products** are exposed for consumption by other domains or users.	Data is consumed via a unified layer with flexible access and advanced analytics tools.

Data ingestion in data lake works is done mostly in batch mode without initial transformation, which centralizes raw data in the data lake. In the Data Mesh, each domain is responsible for ingesting its data independently. The Data Fabric offers real-time, automated ingestion, providing seamless and continuous integration of data from various sources. Concerning data transformation, data lakes operate this step on demand, once the data is stored, according to the needs of users or applications. In the Data Mesh, transformation is delegated to each domain, allowing each unit to transform its data for specific uses. The Data Fabric enables real-time transformation, facilitated by automated processes that adapt to continuous integration and analysis needs. For storage, the data lake establishes a clear separation of data into separate zones dedicated to raw and refined data with technologies like HDFS and Parquet to optimize volume management. The Data Mesh takes a different approach by offering autonomous storage management per domain, allowing them to use solutions tailored to their needs. The Data Fabric offers distributed and integrated storage, providing a unified view of all data, whether centralized or distributed. In terms of governance and data quality, the Data Mesh advocates decentralized governance , where each area is responsible for ensuring the quality of the data it produces and exposes to others. The Data Fabric further centralizes this governance, automating metadata management and imposing global standards to ensure consistency of data across the organization. Finally, for security and compliance, traditional data lake architectures adopt a centralized management of these aspects, which can sometimes limit flexibility. The Data Mesh and Data Fabric offer more flexible and distributed approaches. In particular, the Data Fabric implements real-time access control, ensuring secure management that is adaptable to user needs while ensuring encryption and security compliance.

Based on the analysis of the different approaches (data lake, data mesh, and data fabric), it is quite possible to consider a complementary integration of these architectures to take advantage of their respective advantages. Data lake integration with Data Mesh and Data Fabric allows the strengths of each approach to be combined, thus optimizing data management, governance, and usage.

The table 3 summarizes the benefits of integrating the data lake with these concepts:

Table3. Aspect of Benefices Data Mesh and Data Fabric Integrated with Data Lake

Aspect	Data Lake	Data Mesh	Data Fabric	Integration (Data Lake + Data Mesh + Data Fabric)
Flexibility and Decentralization	Centralized, less flexible in terms of autonomous management of domains.	Increased autonomy for each domain.	Unifies access to distributed data.	Autonomy of domains with a centralized global view; increased flexibility and unified access.
Storage Optimization	Massive storage of raw and refined data.	Storage management by domain.	Distributed and integrated storage.	Modularity in the choice of storage technologies; scalability and optimization of resources.
Data Governance and Quality	Centralized governance and data quality are dependent on internal processes.	Decentralized governance, each domain manages data quality.	Centralized governance via metadata management and automation.	Combined governance: decentralized at the domain level and centralized for metadata management.
Security and Compliance	Centralized security management.	Security is applied at the domain level but is less centralized.	Real-time access controls, and integrated security.	Real-time access controls; fine-grained, contextual security at the domain and global levels.
Ingestion and Transformation	Batch ingestion without initial transformation.	Autonomous ingestion by domain.	Automated, real-time ingestion, integrated transformation.	Real-time ingestion; flexibility and automation in data management.

Data lake integration with Data Mesh and Data Fabric provides an optimal solution by combining the large volumes of raw data provided by the data lake, decentralization, and autonomy of the Data Mesh domains, as well as automation, scalability, and centralized governance of the Data Fabric. This synergy allows for the design of decision architectures that are both robust and flexible, adapted to the current challenges of data management. This is exactly what we have implemented in our approach. We will present in detail our DLMF architecture, which seamlessly integrates these tree concepts , data lake, data mesh and the data fabric.

4. DLMF ARCHITECTURE:A HYBRID DATA MANAGEMENT ARCHITECTURE

4.1. Presentation of the DLMF Architecture

The hybrid architecture combining Data Lake, Data Mesh, and Data Fabric is a modern and integrated approach to data management that enables organizations to leverage the strengths of each concept while mitigating their respective limitations. needs in data management.

Journal of Theoretical and Applied Information Technology
15th November 2024. Vol.102. No. 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

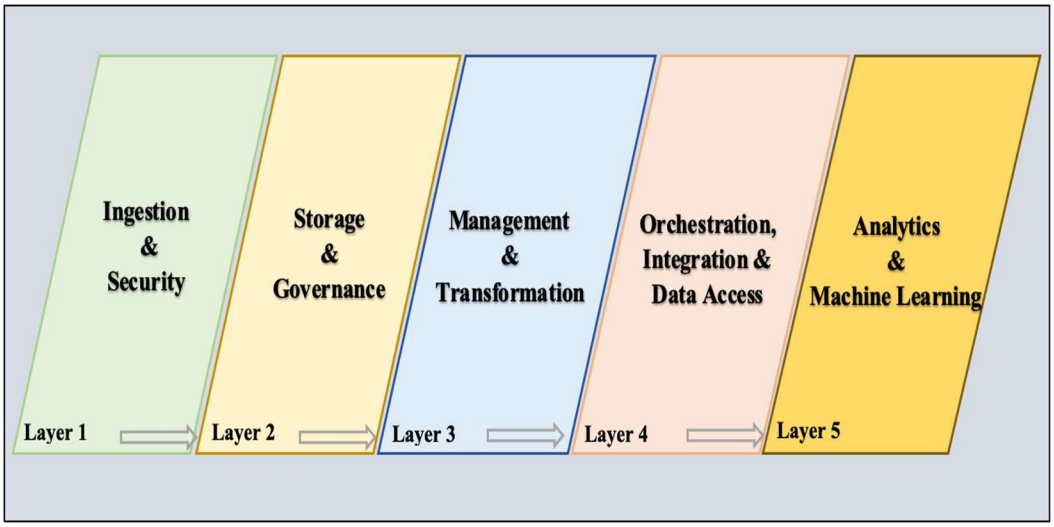

Figure 1. DLFM Architecture

- **Layer 1: Ingestion and Security**

This layer is responsible for integrating data from heterogeneous sources, whether structured, semi-structured, or unstructured, in native form. At the same time, it implements security and governance measures at entry, ensuring that data is protected and compliant with regulatory requirements from the moment it is ingested. Technologies such as Apache Kafka, and Talend and security protocols such as SSL/TLS ensure the smoothness and security of the incoming data flow.

- **Layer 2: Storage and Governance**

This layer, placed at the center of the architecture, stores data in its raw format, offering scalable and flexible storage capacity. The data lake allows for centralized data management while applying centralized governance rules. This approach ensures that data is accessible and usable without the need for a predefined structure while maintaining high-quality data. Technologies like Amazon S3 or Hadoop HDFS, together with data catalogs such as Apache Atlas, are essential to this layer.

- **Layer 3: Management and Transformation**

At the level of this layer, decentralized data management is used to allow specific teams or areas to take responsibility for their datasets. Each domain transforms its data into ready-to-use data products while ensuring local governance. This approach improves agility and efficiency, allowing faster decision-making and better adaptation to the specific needs of end users. Tools such as Snowflake or Databricks facilitate this management, while frameworks such as dbt (data build tool) support the transformation of data.

- **Layer 4 : Orchestration, Integration and Data Access**

This layer ensures seamless integration of data across layers and domains, orchestrating the flow of data consistently and providing unified and secure access to users. By seamlessly integrating data, the Data Fabric allows organizations to fully utilize their data resources for analysis and decision-making. Technologies such as Kubernetes for orchestration and Informatica for integration ensure that data flows efficiently through the architecture.

- **Layer 5: Analytics and Machine Learning**

By combining the features of Data Lake, Data Mesh, and Data Fabric, the DLMF hybrid architecture provides a complete and flexible solution for data management. It addresses today's challenges in data processing, storage, and analysis while ensuring the flexibility, governance, and scalability required to support business strategic objectives.

4.2. Features and Contributions of the Concepts

After having presented the layers of DLMF architecture, we will now detail the use of each concept (Data Lake, Data Mesh, and Data Fabric) in our proposal.

- **Using of the Data Lake :**

Massive and centralized storage: The Data Lake is used to store all data in its raw format, without the need for initial transformation. This allows for keeping a complete copy of the data, hence, ensuring its availability for different future use cases.
Scalability: Thanks to the almost unlimited expansion capacity of storage systems such as Hadoop HDFS or Amazon S3, the Data Lake ensures scalable management of data volumes, thus meeting the growing storage needs of companies.
Centralized Governance: Data Lake also plays a key role in managing data governance, with tools like Apache Atlas or Glue Data Catalog to ensure data quality, compliance, and security at a global level.

- **Using of Data Mesh**

Decentralization and Data Accountability : Data Mesh introduces a decentralized approach where each business area is responsible for managing its data. This allows each team to transform raw data into ready-to-use data products, tailored to their specific needs.
Data Products : Domains create data products, which are datasets that can be accessed and reused by other teams or applications. This encourages data reuse and accelerates the development cycle of analytics projects.
Domain-Level Governance : In addition to the centralized governance of the Data Lake, the Data Mesh allows for domain-specific governance rules to be applied, providing increased flexibility to meet local requirements while respecting global guidelines.

- **Use of Data Fabric**

Seamless Data Integration: The Data Fabric ensures the smooth and consistent integration of data from different domains (Data Mesh) and centralized storage (Data Lake). It allows to creation of an interconnected data fabric that makes data transparently accessible to all users.
Orchestration and Unified Access: Using orchestration tools like Kubernetes and Apache Airflow, the Data Fabric manages data flows across layers and domains, ensuring that data is accessible at the right time in the right format. This improves operational efficiency and ensures that users have the data needed for their analyses.
Strengthened Governance and Security: The Data Fabric strengthens governance by enforcing uniform rules across all domains and storage systems, ensuring global compliance. Additionally, it improves data security by providing centralized access control and ensuring that data is protected throughout its lifecycle.

In the DLMF hybrid architecture, the data lake plays an essential role in the massive, centralised storage of raw data, guaranteeing scalability and global governance. The Data Mesh offers the possibility of managing data in a decentralised way, where each business area converts data into specific data products, offering great flexibility and agility. Finally, the Data Fabric links these two worlds, providing seamless integration, improved governance and unified access to data, resulting in a robust and seamless data management ecosystem.

4.3. Technologies for implementing the DLMC architecture

To implement the DLMF's hybrid functional architecture, it is essential to rely on robust, scalable and proven technologies. Open-source software offers a wide range of solutions that not only meet the technical requirements, but also provide the flexibility and adaptability essential to modern data management. In this section, we propose a set of open-source and non-open-source technologies selected for their ability to effectively support the different layers of the DLMF architecture, ensuring seamless integration of the Data Lake, Data Mesh and Data Fabric concepts. These technologies are tailored to create a complete, integrated solution capable of meeting the current and future challenges of data management within organisations.

4.3.1. Open source technologies:

- **Ingestion and Safety Layer**

Apache Kafka: For real-time data ingestion, Kafka allows for managing massive data flows with low latency. It facilitates the collection and distribution of data from various sources.
Apache NiFi: Ideal for automating data flows, NiFi provides robust data management and traceability while offering built-in security features.
Apache Ranger: For security policy management and access auditing, Ranger provides granular, centralized access control over data.

- **Storage and Governance Layer**

Hadoop HDFS: HDFS is a proven solution for distributed storage of large amounts of data. It is designed to be scalable and fault-tolerant.
Apache Hive: For metadata management and querying of stored data, Hive allows SQL queries to be executed on the data in HDFS.

Apache Atlas: This governance tool helps manage metadata, ensure data traceability, and maintain regulatory compliance.

- **Transformation and Management Layer**

Apache Airflow: For orchestration of data workflows, Airflow allows to plan and automate transformation pipelines within each domain.
dbt (Data Build Tool): dbt is a powerful solution for transforming data into data products, with an emphasis on modularity and reproducibility.
Apache Kafka Streams: For real-time data stream processing, Kafka Streams offers native integration with Kafka, allowing complex transformations.

- **Orchestration, Integration, and Access Layer (Data Fabric)**

Kubernetes: Kubernetes orchestrates containers and provides efficient resource management, facilitating the integration and deployment of services in a distributed architecture.
Apache Camel: For data integration, Camel provides a lightweight routing library, which allows for connecting and transforming data between different applications and services.
PrestoSQL (Trino): For distributed SQL queries, PrestoSQL allows for running quick scans on data from various storage sources, whether structured or unstructured .

- **Analytics and Machine Learning Layer**

Apache Spark: Spark is a massive data processing solution that supports real-time analytics, machine learning, and large-scale graphs.
TensorFlow: For machine learning needs, TensorFlow allows you to create, train, and deploy machine learning models across large datasets.
Apache Superset: For data visualization, Superset offers a feature-rich open-source dashboard, allowing you to create interactive and intuitive visualizations.

This selection of open-source technologies provides a comprehensive and coherent solution for implementing the DLMF hybrid architecture. Each technology was chosen for its ability to integrate into a distributed, scalable, and flexible infrastructure while ensuring robust governance and increased security. Together, they provide a solid foundation for addressing diverse data management needs and take full advantage of the benefits of Data Lake, Data Mesh, and Data Fabric.

4.3.2. Non-Open Source Technologies

- **Ingestion, Security, and Governance layer**

Informatica PowerCenter: For data integration and transformation, Informatica PowerCenter provides a robust data flow management platform that also ensures data quality and security.
IBM DataStage: For large-scale data integration and transformation, IBM DataStage provides powerful ETL process management capabilities, making it easy to handle complex data.
Microsoft Azure Purview : For data governance, Azure Purview enables advanced metadata management and ensures compliance by providing traceability and security policy management capabilities.

- **Storage, Processing and Analysis Layer**

Amazon S3 : For scalable and durable storage, Amazon S3 provides a robust cloud solution that can handle large volumes of raw and transformed data with high availability.
Google BigQuery: For large-scale data analysis, Google BigQuery enables fast and interactive SQL queries on large datasets, making it easy to perform powerful analysis.
Snowflake : For data warehousing, Snowflake provides a high-performance, flexible scalability data storage and analytics platform that is tailored to complex analytical needs.
Matillion : For ETL data transformation, Matillion provides an intuitive interface for creating and managing data pipelines, and simplifying data transformation and integration.

- **Integration, Orchestration and Access Layer**

Red Hat OpenShift: For container orchestration, Red Hat OpenShift provides a platform for application and resource management, making it easy to deploy and scale services.
TIBCO MuleSoft: For application integration, MuleSoft offers tools for routing and transforming data between different systems, ensuring seamless connectivity.
Apache NiFi (Commercial version): For data flow management, the commercial version of Apache NiFi provides advanced features for data integration and orchestration with dedicated technical support.

- **Analytics and Machine Learning**

DataRobot: For automated machine learning, DataRobot enables the creation and deployment of

predictive models with an intuitive user interface and advanced automation capabilities.

H2O.ai: For machine learning and predictive analytics, H2O.ai offers powerful tools for model creation and deployment, making it easy to perform advanced data analysis.

Tableau: For advanced visualization, Tableau allows the creation of interactive dashboards and dynamic visualizations, making it easy to analyze and present data.

These non-open-source technologies complement the DLMF architecture by providing additional functionality and advanced capabilities for data management, processing, and analysis.

Table 4 presents a summary of the features of the different technologies presented according to each layer.

By combining open-source and non-open-source technologies, the DLMF architecture benefits from increased flexibility and a full range of features tailored to the diverse needs of organizations. Open-source solutions allow for flexible customization and integration, while commercial options offer advanced capabilities and strong technical support. Together, they create a robust and scalable architecture that is well-suited to modern data management challenges. In the following section, we will present a case study to design our architecture for analytical purposes.

Table 4. Summary Of The Characteristics Of The Technologies In The Different Layers Of The DLMF Architecture

Layer	Open-Source Technologies	Features/functions	Non-Open Source Technologies	Features/Functions
Ingestion, Security, and Governance	**Apache NiFi**	Data ingestion, flow management, data security	**Informatica PowerCenter**	Data integration, transformation, flow management
	Apache Ranger	Security policy management, audit, and compliance	**IBM DataStage**	Data integration and transformation, flow management
	Apache Atlas	Metadata management, traceability, classification	**Microsoft Azure Purview**	Data governance, metadata management, compliance
Storage, transformation, and Analysis	**Apache Hadoop (HDFS)**	Distributed storage on a large scale	**Amazon S3**	Scalable and sustainable storage
	Apache Spark	Massive data processing, machine learning, transformation	**Google BigQuery**	Large-scale data analysis, interactive SQL queries
	PrestoSQL (Trino)	Distributed SQL queries, interactive analysis	**Snowflake**	Data warehousing, high-performance SQL queries
	dbt (Data Build Tool)	Data transformation, creation of data products	**Matillion**	ETL data transformation, workflow management
Integration, Orchestration, and Access	**Kubernetes**	Container orchestration, resource management	**Red Hat OpenShift**	Container orchestration, application management
	Apache Camel	Integration and routing of data between systems	**TIBCO MuleSoft**	Application integration, routing, and data transformation
	Apache Airflow	Workflow orchestration, task planning	**Apache NiFi (En version commerciale)**	Advanced data flow management, application integration

Analytique et Machine Learning	**Apache Spark**	Big data processing, machine learning, analytics	**DataRobot**	Automated machine learning, model creation, and deployment
	TensorFlow	Creation and deployment of machine learning models	**H2O.ai**	Machine learning and predictive analytics, model deployment
	Apache Superset	interactive data visualization	**Tableau**	Advanced visualization, creation of interactive dashboards

After presenting our DLMF architecture, we can say that it stands out from other architectures presented in the literature due to its complete integration of advanced data management concepts such as Data Mesh and Data Fabric. Unlike architectures like HEALLER[24] or IISOBA[25], which focus on specific domains such as healthcare or agriculture, DLMF proposes a five-layer modular architecture that covers ingestion, storage, governance, transformation, and analysis. It relies on open-source technologies like Apache NiFi, Kafka, and TensorFlow to ensure flexibility, scalability, and secure data access. The use of Data Mesh decentralizes data management, while Data Fabric ensures unified access and governance, providing a solid foundation for managing complex and heterogeneous data sources.

What further differentiates DLMF is the emphasis on continuous data governance and metadata management through tools such as Apache Atlas and Amundsen. These components ensure end-to-end traceability and compliance, a feature less emphasized in other architectures, such as HEALLER or IISOBA, which focus more on storage and consumption layers without a strong governance framework. Moreover, the integration of Data Mesh across multiple layers of DLMF allows for better handling of evolving data needs, particularly in predictive analytics, compared to traditional approaches. Overall, the DLMF architecture offers a more adaptable, secure, and scalable solution for addressing modern data management challenges across various domains, including healthcare.

5. Implementation of DLMF Architecture

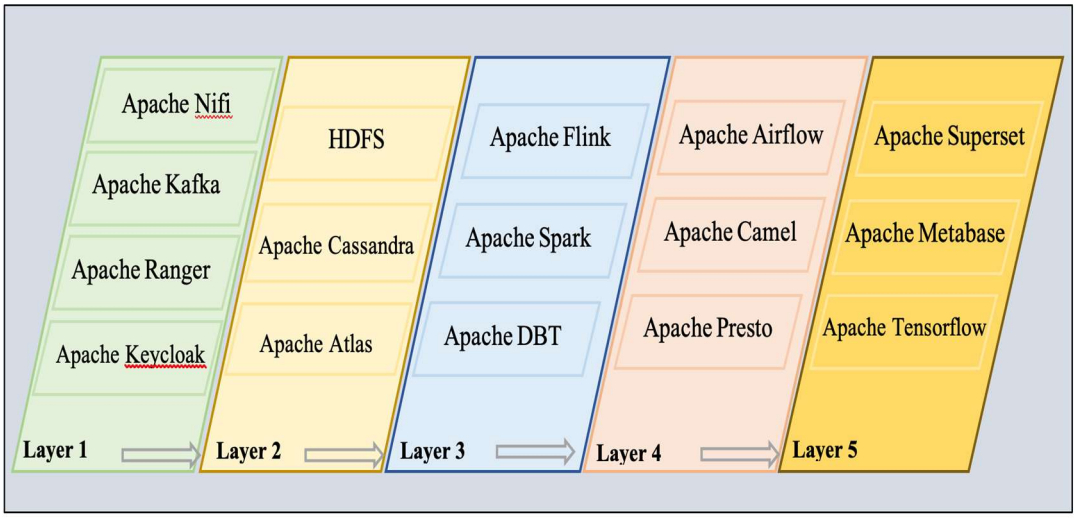

Figure 2. Implementation of DLMF Architecture

In this section, we will implement a simple technological solution for our DLMF medical architecture described in the previous section and illustrated in Figure 1. Figure 2 shows the DLMF architecture in technological form.

This figure illustrates the five-layer architecture for data management, integrating open-source technologies. Layer 1, dedicated to Ingestion and Security, uses Apache NiFi and Apache Kafka to manage data flows, while Apache Ranger and Keycloak provide security and access management. Layer 2 is focused on Storage and Governance, with Apache Hadoop HDFS and Apache Cassandra for data storage, and Apache Atlas as well as Amundsen for metadata governance. Layer 3 is for Management and Transformation, using Apache Flink and Presto for data management and querying, and Apache Spark and dbt for transformation. Layer 4 encompasses Orchestration, Integration, and Data Access, with Apache Airflow to orchestrate workflows, Apache Camel for systems integration, and Presto for data access. Finally, Layer 5 focuses on analytics and machine learning, with Apache Superset and Metabase for data visualization, and TensorFlow for predictive analytics and machine learning. This architecture provides robust, scalable, and flexible data management by integrating open-source solutions at every stage of the data processing.

In comparison to existing works, presented in section 3,our DLMF architecture presents unique added value through its comprehensive, layered structure, tailored specifically for healthcare data management. While architectures like HEALLER[24] and IISOBA[25]primarily focus on batch data processing and storage, our DLMF architecture encompasses six distinct layers: ingestion & security, storage & governance, domain management & transformation, orchestration, data integration & access, and analysis & machine learning .

The Ingestion and Security layer in DLMF utilizes Apache NiFi and Kafka for real-time data ingestion, unlike HEALLER's batch-only approach, enhancing data immediacy and responsiveness. Additionally, this layer integrates robust security with Apache Ranger and Keycloak, offering more sophisticated access control than what is seen in HEALLER and IISOBA.

In the Storage and Governance layer, DLMF combines HDFS and Cassandra for scalable storage, while tools like Apache Atlas and Amundsen strengthen metadata governance. This governance structure is more comprehensive than in IISOBA, which lacks advanced metadata handling.

The Domain Management and Transformation layer in DLMF uses Apache Flink, Presto, and dbt to support dynamic transformations and structured data management, distinguishing it from existing works that do not emphasize real-time transformations or domain-specific data structuring.

In the Orchestration, Data Integration, and Access layer, DLMF incorporates Apache Airflow and Camel, providing orchestrated workflows and seamless system integration. This level of orchestration and flexible data access, particularly through Presto for querying, goes beyond what is seen in the reviewed works, which often treat data access as a secondary feature.

Finally, the Analysis and Machine Learning layer of DLMF uses Apache Superset, Metabase, and TensorFlow for advanced analytics and predictive modeling, which is not addressed in articles [24] to [28]. This layer enables actionable insights, providing a practical application of machine learning specifically for healthcare.

Compared to works like article [26]—which merges data lake and data warehouse concepts without a healthcare focus—and article [27] on Data Mesh, DLMF provides a concrete technological implementation of these concepts tailored for healthcare. By leveraging the Data Fabric and Data Mesh principles from article [28], DLMF addresses both data integration and governance within a specific medical context. Therefore, DLMF offers an end-to-end, open-source, healthcare-specific architecture with robust governance, security, and advanced analytical capabilities not found in existing approaches.

The DLMF architecture, while robust, faces several limitations. Its multi-layer structure adds complexity to implementation and maintenance, requiring specialized expertise and significant resources. Scalability can be challenging, particularly for handling real-time healthcare data, which may introduce latency issues. Security and regulatory compliance, though addressed with tools like Apache Ranger, remain complex to manage across distributed layers. Integration with legacy healthcare systems can also be difficult, limiting data fluidity. Moreover, relying solely on open-source tools may restrict advanced customizations, while data fragmentation risks could affect data quality and consistency if strict governance isn't maintained.

6. CURRENT,PICTURE, RECOMMENDATION, AND FUTURE DIRECTIONS

Traditional data warehouse architectures, while effective for processing and analyzing

structured data, are often limited by their rigidity and centralization. These systems require complex ETL (Extract, Transform, Load) processes, resulting in long delays between data collection and availability for analysis. In addition, the rapid evolution of business needs and the diversity of data sources make these architectures difficult to adapt and expensive to maintain. With the emergence of data lakes, more flexible solutions have emerged, allowing large amounts of raw, structured, and unstructured data to be stored. However, data lakes often suffer from governance issues, data quality, and information silos which led to the emergence of the Lakehouse concept, which attempts to combine the benefits of data warehouses and data lakes. Despite this, the challenges of centralized management and scalability remain, limiting the effectiveness of these architectures in complex and ever-changing data environments. These are precisely the issues that our DLMF architecture, presented earlier, seeks to address.

Given the limitations of centralized architectures, decentralized approaches such as Data Product, Data Mesh, and Data Fabric are emerging as powerful and innovative solutions. Data Mesh, for example, is based on the idea that data should be managed as products by dedicated teams, with each business area being responsible for the management, quality, and accessibility of its data. This approach promotes agility, scalability, and flexibility, allowing teams to respond quickly to changing market needs. In addition, the Data Fabric provides unified and transparent management of data across hybrid environments, allowing for seamless integration between disparate data sources. By integrating federated governance principles and self-service infrastructure, these decentralized approaches allow for better interoperability, greater resilience, and a faster response to end-user requirements, while avoiding the bottlenecks associated with centralized systems. These benefits are implemented in our DLMF architecture, which leverages these principles to create a more efficient and modern-day data management solution.

To maximize the benefits of data management architectures, it is recommended to consider a hybridization of centralized and decentralized approaches, like the one implemented in our DLMF architecture. By combining the strengths of data warehouses, data lakes, and lakehouse systems with the decentralized principles of Data Mesh and Data Fabric, organizations can create more robust and flexible solutions. It is advisable to maintain centralized governance to ensure data compliance and security while allowing domain teams to manage their data with utmost autonomy and flexibility. Additionally, integrating a self-service infrastructure can facilitate data access and transformation, while respecting the specific needs of each business domain. By adopting this hybrid approach, organizations will be better equipped to meet current and future data management challenges, while optimizing their performance and adaptability.

Moreover, to ensure the continued relevance and effectiveness of the DLMF architecture, two future directions are essential. First, it is crucial to continue the in-depth exploration of the Data Mesh and Data Fabric concepts. This involves not only strengthening the understanding of these concepts to optimize their integration into the architecture but also defining and implementing robust policies for data security and data quality. Strict and well-defined governance will ensure that data is secured and maintained with a high level of quality while allowing for efficient decentralized management.

Second, it is recommended to use this architecture in predictive analytics projects to evaluate its performance in real-world conditions. Applying the DLMF architecture to predictive analytics will test its ability to manage and analyze large volumes of data, providing valuable insights to improve forecast-based decision-making processes. This practical use will help validate the effectiveness of the architecture and adjust data management strategies to better meet future needs.

7. CONCLUSION

In this article, we have proposed a DLMF architecture adapted to the medical domain, integrating the advantages of centralized and decentralized data management systems to offer a solution that is both flexible and scalable. We have detailed the layers that make up the architecture, namely ingestion and security, storage and governance, domain management and transformation, orchestration, data integration and access, and analysis and machine learning. We have also presented an open-source technology implementation for each layer of the DLMF architecture. In terms of future work, there are two main areas in which we hope to enhance the effectiveness of the DLMF architecture. Firstly, it will be essential to continue the in-depth exploration of the Data Mesh and Data Fabric concepts in order to optimize their integration into the architecture. Secondly, the application of the DLMF architecture in predictive analysis projects will make it possible

to test its performance under real conditions in the medical field. This approach will make it easier to assess the effectiveness of the architecture and adjust data management strategies to meet the specific needs and challenges identified.

REFERENCES:

[1] PEARLSON, Keri E., SAUNDERS, Carol S., et GALLETTA, Dennis F. Managing and using information systems: A strategic approach. John Wiley & Sons, 2024.

[2] SPENCER, Stephen A., ADIPA, Faustina Excel, BAKER, Tim, et al. A health systems approach to critical care delivery in low-resource settings: a narrative review. Intensive care medicine, 2023, vol. 49, no 7, p. 772-784.

[3] DIXON, Brian E. et CUSACK, Caitlin M. Measuring the value of health information exchange. In : Health Information Exchange. Academic Press, 2023. p. 379-398.

[4] HOSEN, Mohammed Shahadat, ISLAM, Raisul, NAEEM, Zain, et al. Data-Driven Decision Making: Advanced Database Systems for Business Intelligence. Nanotechnology Perceptions, 2024, p. 687–704-687–704.

[5] MAHIDDIN, Normadiah Binti, OTHMAN, Zulaiha Ali, BAKAR, Azuraliza Abu, et al. An interrelated decision-making model for an intelligent decision support system in healthcare. IEEE Access, 2022, vol. 10, p. 31660-31676.

[6] LÓPEZ-MARTÍNEZ, Fernando, NÚÑEZ-VALDEZ, Edward Rolando, GARCÍA-DÍAZ, Vicente, et al. A case study for a big data and machine learning platform to improve medical decision support in population health management. Algorithms, 2020, vol. 13, no 4, p. 102.

[7] HALAWA, Farouq, MADATHIL, Sreenath Chalil, GITTLER, Alice, et al. Advancing evidence-based healthcare facility design: a systematic literature review. Health Care Management Science, 2020, vol. 23, p. 453-480.

[8]FAN, Wenfei et GEERTS, Floris. Foundations of data quality management. Springer Nature, 2022.

[9] MRABET, Hichem, BELGUITH, Sana, ALHOMOUD, Adeeb, et al. A survey of IoT security based on a layered architecture of sensing and data analysis. Sensors, 2020, vol. 20, no 13, p. 3625.

[10] RANJAN, Jayanthi et FOROPON, Cyril. Big data analytics in building the competitive intelligence of organizations. International Journal of Information Management, 2021, vol. 56, p. 102231.

[11] ABUGHAZALAH, Mona, ALSAGGAF, Wafaa, SAIFUDDIN, Shireen, et al. Centralized vs. Decentralized Cloud Computing in Healthcare. Applied Sciences, 2024, vol. 14, no 17, p. 7765.

[12] ARMBRUST, Michael, GHODSI, Ali, XIN, Reynold, et al.Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In : Proceedings of CIDR. 2021. p. 28.

[13] VERGNE, Jean-Philippe. Decentralized vs. distributed organization: Blockchain, machine learning and the future of the digital platform. Organization Theory, 2020, vol. 1, no 4, p. 2631787720977052.

[14] ZAGAN, Elisabeta et DANUBIANU, Mirela. Data lake approaches: A survey. In : 2020 International Conference on Development and Application Systems (DAS). IEEE, 2020. p. 189-193.

[15] KHAN, Bilal, JAN, Saifullah, KHAN, Wahab, et al. An Overview of ETL Techniques, Tools, Processes and Evaluations in Data Warehousing. Journal on Big Data, 2024, vol. 6.

[16] SYED, Dabeeruddin, ZAINAB, Ameema, GHRAYEB, Ali, et al.Smart grid big data analytics: Survey of technologies, techniques, and applications. IEEE Access, 2020, vol. 9, p. 59564-59585.

[17] AZZABI, Sarah, ALFUGHI, Zakiya, et OUDA, Abdelkader. Data Lakes: A Survey of Concepts and Architectures. Computers, 2024, vol. 13, no 7, p. 183.

[18] GOEDEGEBUURE, Abel, KUMARA, Indika, DRIESSEN, Stefan, et al. Data mesh: a systematic gray literature review. ACM Computing Surveys, 2023.

[19] MAJCHRZAK, Jacek, BALNOJAN, Sven, et SIWIAK, Marian. Data Mesh in Action. Simon and Schuster, 2023.

[20] EDUCATION, Certybox. Data science. Certybox Education, 2023.

[21] VLASIUK, Y. et ONYSHCHENKO, V. Automated detection of the data product consumers in data mesh. 2023.

[22] CASTELLUCCIO, Michael. Data fabric architecture. Strategic Finance, 2021, vol. 103, no 4, p. 57-58.

[23] KUMAR, Abhishek, LOVÉN, Lauri, PIRTTIKANGAS, Susanna, et al. Data Fabric for Industrial Metaverse. In : 2024 IEEE 44th

International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, 2024. p. 113-121

[24] MANCO, Carlo, DOLCI, Tommaso, AZZALINI, Fabio, et al.HEALER: A Data Lake Architecture for Healthcare. In: EDBT/ICDT Workshops. 2023.

[25] Ouafiq, E.M., Saadane, R., Chehri, A., Wahbi, M. (2022). Data Lake Conception for Smart Farming: A Data Migration Strategy for Big Data Analytics. In: Zimmermann, A., Howlett, R.J., Jain, L.C. (eds)

[26] TEMIDAYO, Folalu et DAMOLA, Precious. STRUCTURE OF BIG DATA LAKE HOUSES. 2024.

[27] Vlasiuk, Y., Onyshchenko, V. (2023). Data Mesh as Distributed Data Platform for Large Enterprise Companies. In: Hu, Z., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education VI. ICCSEEA 2023. Lecture Notes on Data Engineering and Communications Technologies, vol 181. Springer, Cham. https://doi.org/10.1007/978-3-031-36118-0_17

[28]PAKRASHI, Arjun, WALLACE, Duncan, MAC NAMEE, Brian, et al. CowMesh: a data-mesh architecture to unify dairy industry data for prediction and monitoring. Frontiers in Artificial Intelligence, 2023, vol. 6, p. 1209507 Human Centred Intelligent Systems. Smart Innovation, Systems, and Technologies, vol 310. Springer, Singapore. https://doi.org/10.1007/978-981-19-3455-1_15