Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 196 (2022) 263–271

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Data Mesh: Concepts and Principles of a Paradigm Shift in Data Architectures

Inês Araújo Machado*, Carlos Costa, Maribel Yasmina Santos

University of Minho, ALGORITMI Research Centre, Guimarães, Portugal

Abstract

Inherent to the growing use of the most varied forms of software (e.g., social applications), there is the creation and storage of data that, due to its characteristics (volume, variety, and velocity), make the concept of Big Data emerge. Big Data Warehouses and Data Lakes are concepts already well established and implemented by several organizations, to serve their decision-making needs. After analyzing the various problems demonstrated by those monolithic architectures, it is possible to conclude about the need for a paradigm shift that will make organizations truly data-oriented. In this new paradigm, data is seen as the main concern of the organization, and the pipelining tools and the Data Lake itself are seen as a secondary concern. Thus, the Data Mesh consists in the implementation of an architecture where data is intentionally distributed among several Mesh nodes, in such a way that there is no chaos or data silos, since there are centralized governance strategies and the guarantee that the core principles are shared throughout the Mesh nodes. This paper presents the motivation for the appearance of the Data Mesh paradigm, its features, and approaches for its implementation.

Keywords: Big Data; Data Mesh; Data Architectures; Data Lake.

* Corresponding author.
E-mail address: inesamachado98@gmail.com

10.1016/j.procs.2021.12.013

1. Introduction

The Data Mesh emerges as a necessary paradigm shift that will enable companies to become truly data-oriented, implementing an architecture that brings the opposite of the current models for efficient data product cooperation [1]. From a more structural perspective, data is organized into domains and data teams manage themselves and carry out their own work in an agile and product-oriented way. However, this paradigm shift does not occur only on a structural level, but also on an organizational level - as the way data teams are organized will become focused on a specific domain and therefore decentralized [2]. The Data Mesh allows for the provision of complex management, access, and support components through the connectivity layer it implements - data from different locations will now be connected in the Mesh [3]. Recently, Zhamak Dehghani began taking the first steps in consolidating what might be the core principles and logical architecture of a Data Mesh [4]. Although this initial consolidation is relevant, the topic still lacks scientific contributions and research artefacts to support it. Therefore, this paper aims to consolidate the background knowledge regarding the concept of Data Mesh and the current known approaches and features, thus serving as a state of the art review for this topic. The contribution of this paper is twofold: i) present a comprehensive overview of the Data Mesh concepts, its main guiding principles and how they advance the state of the art in data systems; ii) describe two recent implementations, identifying their main differences in terms of followed approaches. This knowledge is crucial for a conceptual formalization that serves as basis for future contributions in this field.

In this paper, section 2 presents the literature review process and the background knowledge. Section 3 describes the features and approaches followed in the design and implementation of the Data Mesh. Section 4 presents two current approaches. Finally, section 5 discusses these approaches and concludes with some remarks.

2. Background Knowledge

To clarify the literature review process, this section starts by giving an overview of the steps and criteria that were followed. Due to the novelty of the Data Mesh topic, currently, there are no scientific contributions specifically related to the topic. Therefore, other reliable sources of contributions were analysed. For the remaining concepts (e.g.: Data Warehouse), priority was given to papers published from 2015 onwards - the exception was papers with a relevant number of citations. Some keywords used in reference databases such as "Scopus", "Research Gate" and "Mendeley", were "Data Lake", “Data Mesh”, “Domain Driven Approach”, among others. YouTube videos and blogs were used to collect information about the Data Mesh concept. The most relevant information was selected, analysed and compose the background knowledge discussed in this section.

The production and consumption of data is a constant in today’s world [5]. Clearly, the way data is produced and consumed nowadays is nowhere near the way this phenomenon occurred a few decades ago. Some state that a decade ago, what was considered "a great dataset", would nowadays be probably considered absurd [6]. The rapid evolution of Big Data [5] has led to some confusion on how to explain it, thus diverging between "what Big Data is" and "what Big Data does" [7]. With the observation of the different definitions [8–10],[11],[12], it is possible to conclude that to define Big Data, it is necessary to go into detail about its characteristics. Doug Laney formulated the 3Vs model (Volume, Variety, and Velocity) in 2001, and this model served as the basis for defining Big Data for a decade [13], considering that these would be the three main challenges when dealing with Big Data, also forming its main characteristics. With the continuous work in Big Data, it was possible to see that the 3Vs model was incomplete, and that two more Vs could be added to it: veracity and value [5]. Although these five main characteristics have been defined, other authors identify even more characteristics such as variability, complexity, ambiguity, viscosity, and virality [7]. In an organization, the operational data alone does not allow to support decision-making processes. For this to happen, there must be mechanisms that extract perceptible analytical value from this operational data, so that it can reach the respective stakeholders [14]. The phenomenon of Data Warehousing emerged as a response to this imminent need to use the data produced by organizations to generate value [15]. According to Kimball, Data Warehouses can be summarized as systems that ingest operational data (over which they have no quality control) and hold as output the analytical value for decision-making [14]. The purpose of this collection is to conduct data analysis to support decision-making [15]. Inmon also defines a Data Warehouse as being a data repository that supports decision-making. Kimball & Ross define the structure of a Data Warehouse as the integration of four distinct components: source transactions, Extract Transform Load (ETL) System, data presentation area, and Business

Intelligence (BI) application [14]. In short, the Data Warehouse is a repository that stores the organization's data, and that enhances its analysis [5]. The Data Warehouse is widely accepted and implemented, and its role has changed over the years. These changes are related to the characteristics of Big Data and the need for advanced analytics – without these changes, Data Warehouses would not be adequate in the support of Big Data contexts [5]. Therefore, the scientific community began to study the modernization of the Data Warehouse, in order to accommodate these several changes [16]. The concept of Big Data Warehouse emerges as the way to overcome the difficulties experienced by Data Warehouses when processing Big Data [8]. It is possible to define a Big Data Warehouse as a system presenting flexible storage, accompanied by adequate scalability and performance. These systems also focus on low latency when it comes to data ingestion and analytical workloads of complex nature [17]. It is possible to conclude that, considering the challenging Big Data characteristics, there was an evolution from a Data Warehouse to a Big Data Warehouse. The concept of Data Lake dates back to a decade ago, through James Dixon, and was partially devalued at the time because it was believed to be a Hadoop marketing label [18]. However, the concept has remained and has grown over time [19]. Laskowski describes a Data Lake as a largely scalable repository denoted as a significant mass of data, where it is stored in its "As-Is" form, remaining in this state until the need for processing emerges. This addition of new raw data does not interfere with the data structures already present in the Lake, which allows to continuously inject data into it [20]. Data Lake and Data Warehouse are both data repositories that differ from each other, in terms of structure and implementation [19]. Summarizing the comparison between the Data Lake and the Data Warehouse, it is possible to conclude that the former deals with three types of raw data (unstructured, semi-structured and unstructured), while the latter mainly deals with processed and structured data. Storage costs are much higher in Data Warehouse environments, and this type of repository is less agile compared to the Data Lake [19]. There is also a difference regarding their users, as the Data Warehouse mainly targets professional business users (e.g., managers and directors), while the Data Lake mainly targets data scientists [19]. In short, the Data Lake corresponds to a pool that accepts structured, semi-structured and unstructured data, and theoretically scales to an infinite amount of raw data. The Data Lake can be considered more effective and efficient to deal with heavy workloads, compared to the already established Data Warehouses. It should be highlighted that in a Data Lake governance must always exist, avoiding it to become a Data Swamp [19].

3. The Data Mesh Paradigm: Main Concepts and Constructs

Zhamak Dehghani argues that the current data architectures are in a state of crisis, and therefore the expression "paradigm shift" show up, being associated with Data Mesh [2]. In the last three years, there has been a significant increase of interest and investment by companies in areas such as Big Data and Artificial Intelligence. Between 2018 and 2019 there was an investment of 66% compared to the previous year. However, contrary to the expectations, the satisfaction of the companies has decreased (regarding the recognition of the importance of these technologies to increase competitiveness and value). Only between 2018 and 2019, there was a decrease of 19% in this area [2]. According to Zhamak Dehghani, this fact shows how there are still serious gaps in the adopted architectures [2], although there was an evolution (from Data Warehouses) to the present day (Data Lakes in the cloud). There is currently an overload in the data teams in response to the growing needs of the organization, ranging from ad-hoc exploration to central ETL pipeline management. There is an unsatisfactory alignment between the organizational needs and the architectures instituted [21]. These two facts lead to this overload felt by the data teams and the discontent of various investors. Although in software engineering there has been an evolution from monolithic architectures to microservice architectures, in data engineering this change has not happened yet [1]. The Data Mesh arises as a paradigm shift that occurs both at the technological and organizational levels. This change has the purpose of solving the problems such as loss of the nature of the data itself, high costs related to the management of monolithic architectures (e.g., Data Lakes), significant pressure on data teams, among others.

3.1. Features of a Data Mesh

The main purpose of the Data Mesh is to create a decentralized data architecture that enables the extraction of large-scale analytical data. In this context, scale can be understood as the adaptation to the proliferation of data sources [4]. In this sense, Zhamak Deghani argues that a Data Mesh should be based on four core principles: *domain-oriented decentralized data ownership and architecture, data as a product, self-serve data platform,* and *federated*

computational governance [1,4]. The first concept of the Data Mesh relates to the organization of the data in line with the business itself. When looking at the organizational structure, it is realized that, divisions are defined by areas of operation (e.g., logistics and customer support), also known as business domains [2]. The Data Mesh postulates the existence of a distributed responsibility, by the organization's teams, that can better understand and produce the data of their specific business domain [21]. In this sense, the ingestion of data obeys the nature of data and decentralize ownership. Therefore, providing the analytical data must always be aligned with the established domains [1]. To distribute responsibility and to decentralize the already known monolithic architectures, it is necessary to model the current data architecture based on the organization of analytical data by domains [4]. To better deal with change, domains will both store and serve their data [1]. The cost of discovering quality data is pointed out as one of the difficulties experienced in current monolithic data architectures (mainly Data Lake) [4]. Part of the problem comes from the fact that, even organizations that consider themselves data-oriented, do not treat the data with the proper democratization [21]. In this sense, the second concept of the Data Mesh emerges, being data as a product [1]. The Mesh applies the already known concept of "Product Thinking" to the data, so that it becomes the organization's top priority, and leaves data pipelining and storage concerns in the background [1]. A simple concept is applied here: analytical data is now seen as a product (and therefore its quality has to be ensured), and consumers of this data are now seen as customers, and their needs must be fulfilled [1]. This fact implies the existence of an infrastructure that allows the teams to produce and maintain their data products. For this, it is necessary that the teams have access to a high-level infrastructure, capable of encapsulating all the complexity inherent to it. Therefore, the third concept emerges - self-serve data platform - that empowers teams with the autonomy needed to manage their domains [1]. From domain to domain, there is a diverse set of technologies to meet the goals of each data product [1]. For the Data Mesh to work as expected, the notion of interoperability and connectivity must be present and well defined [4]. Thus, the self-serve platform must be able to provide the tools and interfaces necessary for the creation and maintenance of data products, without the need for highly specialized knowledge (such as the one that is currently seen in Data Lakes) [4]. Finally, there must be a mechanism that allows interoperability between different domains - the governance model (fourth concept). This governance model must be able to carry out an automated execution of decisions, as well as allow the decentralization and independence of each domain in the Mesh. For this, global normalization is necessary, which Zhamak Dehghani denominates as "federated computational governance" [4]. This concept aims to apply a set of rules to all interfaces of the various data products. The federated computational governance is a complex model, which does not reject change [4]. In general, and taking into account the above-mentioned concepts and key points, the main features that the Data Mesh provide are [4]: i) decentralized team composed of domain representatives, and a clear ownership and responsibility for each data product; ii) use of a self-serve platform to support the development of data products, maintaining the infrastructure and managing the Data Mesh itself; iii) definition of how to model the quality, requirements, and security of data; iv) dealing with the various technologies used in the data products, to ensure interoperability.

3.2. Dehghani's Approach for the Design and Implementation of a Data Mesh

It is important to understand how domains arise and are organized. There are two types of domains: source and consumer domains [2]. Source domains consist of the producers of the data in its raw state at the creation point, not modeled for any consumer. Consumer domains are data domains that may or may not be aligned with source domains. They are different in nature from source domain data, as they undergo significant structural changes. The transformed data is often presented in aggregated views (derived from the source domains) and also include models that allow access to them [1]. Therefore, the data pipelines are made internally within each domain [2]. Each domain has to establish its service quality level, making available to its consumers the quality that its data holds, as well as its lineage, error rate, and schema [1]. Fig. 1 illustrates de concepts described above.

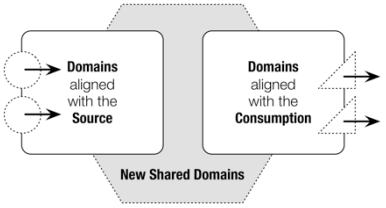

Fig. 1. Structure and Interaction of Domains. Adapted from [1].

Data as a Product implies that there is a set of characteristics that are held by the data [21], since this concept intends to maximize the quality of data [4]. Zhamak Dehghani defends that there are six principles that must be fulfilled to maintain the data quality and efficiency of the Data Mesh [1]. Those principles (DATSIS principles) are the following [22]: Discoverable, Addressable, Trustworthy, Self-describing, Interoperable, and Secure. This orientation towards data as a product results in the appearance of new roles, such as domain data product owner and data product developer [1]. The domain data product owner is concerned with the satisfaction of the consumers of their data products and manage their lifecycle. Consequently, they are responsible for the decision-making around their data products. The domain data product owners must make their work measurable (making use of key performance indicators such as lead time for data availability) [1]. Data product developers are responsible for building, maintaining, and serving the data product domains [4]. Data product developers are not exclusive to a specific domain, and given the organizational needs, these teams can change between domains. When compared to current and past paradigms, the responsibility model is thus reversed, since the responsibility over the data is now close to the source [4]. Considering the definition of this concept, it is possible to define the *architectural quantum* (Fig. 2) as a data product - it consists of the smallest architectural unit that can be deployed, including all the structural components involved [4].

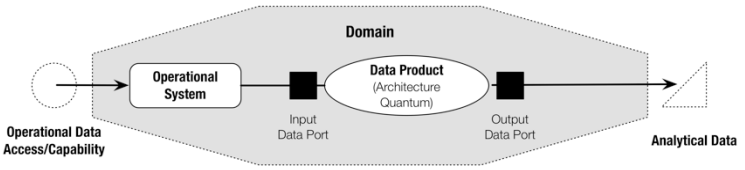

Fig. 2. Architecture Quantum in a Domain. Adapted from [4].

A Mesh Node consists of a data product, which includes three main components: code, data and metadata, and infrastructure [4]. The code component encompasses three distinct segments: data pipeline, applications that allow access to the data and metadata, and the code used for access policies. The data within a data product can be from different types (e.g., batch files and events), but for it to be used, there must be an association between the data and the respective metadata. The infrastructure component allows access to the data and metadata, as well as running the code related to the data product in question [4]. One of the major concerns with the self-serve data platform is the duplication of efforts in the setup of the pipeline engine by the domain teams [2]. To avoid this inefficient duplication of efforts, it is necessary that, when building the platform, business domain concepts are not considered. This abstraction will allow the use of the same infrastructural capabilities across different domains. In this sense, the author points out some capabilities of this platform, such as scalable polyglot big data storage, unified data access control and logging, and data governance and standardization [1]. This architecture can thus be divided into planes, being them remade into levels, which serve different user profiles and not architectural layers [4]. The governance model applied in the Data Mesh must consider two relevant dimensions: achievement of the measures imposed at a global level and respect for the autonomy of the various data domains that compose the Mesh [4]. The architectural result of the combination of the above-mentioned concepts, in Zhamak Dehghani's approach, can be visualized in Fig. 3.

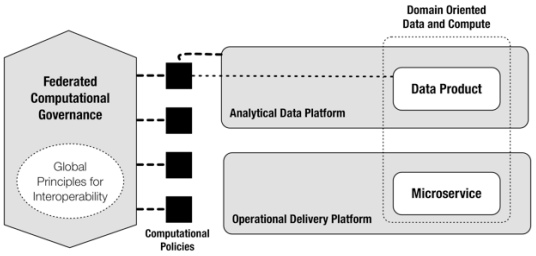

Fig. 3. Data Mesh Architecture. Adapted from [4].

4. Practical Data Mesh Implementations

The present section of the paper is intended to look at some implementations of Data Mesh. It is expected that with the analysis of these implementations, it will be possible to create space for a discussion of future work on the topic.

4.1. Zalando's Data Mesh Implementation

Zalando is a leading fashion platform in Europe [23]. Naturally, and given the nature of the E-Commerce business, there is a need to store, process and use significant amounts of data per day – to do so, the company used a Data Lake. With the use of this Data Lake, some challenges began to arise: the lack of ownership over the data, the poor quality of the data after its processing, and the organizational scalability (with the increasing number of data sources and consumers, the data team became a bottleneck) [23]. Faced with these difficulties, and as a way of trying to overcome them, Zalando decided to build its own Data Mesh. There were some relevant changes: i) evolution towards decentralized data ownership; ii) prioritization of data domains, in detriment of pipelines; iii) vision of data as a product and not as by-product; iv) institution of multifunctionality teams organized by domain; v) abandoning a centralized data environment. These changes led Zalando to overcome the bottleneck at the data team level (decentralizing this infrastructure responsibility to a data infrastructure as a platform) and migrate from a monolithic data architecture (the Data Lake) to an interoperable services environment (the Data Mesh) [23]. Fig. 4 shows the Data Mesh architecture implemented by Zalando. The initial central service (Data Lake Storage) was maintained and the metadata layer and governance that holds information about it were implemented. Zalando then created a concept of "bring your own bucket", which allows users to integrate their S3 (Simple Storage Service) buckets with their data into the common infrastructure [23]. Zalando has retained the central processing platform which uses technologies such as *Databricks* and *Presto.* Clusters (Spark clusters) are available on the processing platform and the users make use of these technologies, without the team responsible for the infrastructure having to know what the users do and without these users having to configure the clusters or to understand their complexity [23]. The main goal was to achieve data sharing among the organization, something that was possible with the use of this architecture, according to the authors [23]. In short, there are two key behavioral changes: treating the data as a primary concern and devoting resources to data quality assurance and understanding of its use [23].

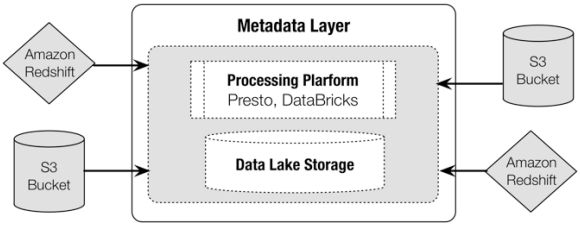

Fig. 4. Zalando's Data Mesh Architecture. Adapted from [23].

4.2. Netflix's Data Mesh Implementation

Netflix is a company that provides a streaming service (e.g., movies), which currently serves about 150 million

global users. Due to the number of users worldwide, Netflix generates trillions of events and, with this, petabytes of data per day [24]. The main objective linked to this topic (Data Mesh) is to make all the studios that work with Netflix (and its productions) into one system, capable of dealing with this large volume of data in an integrated and sustainable way [24]. Therefore, at the beginning of this process, Netflix made a survey about the problems of data transportation that they felt in their scope. They found five major problems: i) duplication of efforts regarding the data pipelines and the teams; ii) unnecessary overload in the maintenance of the pipelines; iii) the lack of implementation of good practices throughout the various processes; iv) the need for lower latency; v) problems in the correction of errors (due to the poor implementation and lack of knowledge by users) [24]. Currently, in its Data Mesh (still in a pre-alpha state as described by Cunningham), the team provides an infrastructure for the various users to develop pipelines [24]. This infrastructure abstracts the user from the complexities of the configurations. The user can take advantage of technologies such as *GraphQL* and *Apache Iceberg* (an open-source project developed by Netflix), access to a metadata catalog (which is seen as a list of sources) from which it is possible to choose to build their pipelines and, get access to a list of pipelining process standards. Currently, Netflix is prioritizing the decrease of operational complexity, not being focused, for now, on obtaining the best results in terms of cost and performance [24]. Fig. 5 summarizes the architecture currently proposed by Netflix.

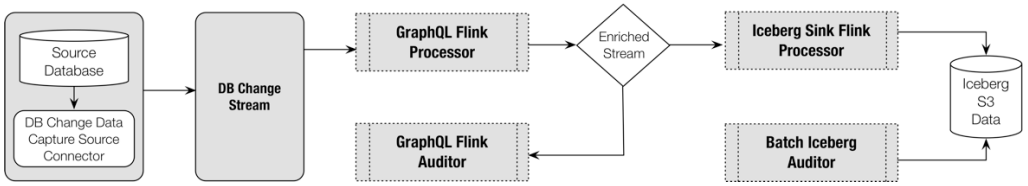

Fig. 5. Netflix's Data Mesh Architecture. Adapted from [24].

The changes that occur in the database serve as a trigger for the entire process. Thus, when they are registered, it triggers the action of the *GraphQL* enrichment processor. This, in turn, resorts to a service of the same nature (*GraphQL*), where it fetches those entities. Having these entities in their possession, these entities transfer them to the data repository, persisting them in an *Apache Iceberg* table [24]. It should be noted that Netflix implements this topology without the user having to know about it (abstracting complexity). This topology also implements an audit mechanism that is present in three different components: in the data source (to verify its accuracy), in the processor's output, and in the batches of data stored in the *Apache Iceberg* format [24].

5. Discussion and Conclusions

Over time, there has been an evolution of data architectures – for example, from Data Warehouses to Data Lakes. Naturally, and as explained in section 2 of this paper, each of these data architectures have been demonstrating their limitations (e.g., bottlenecks associated with having a central development team). It is within these limitations that the various evolutions have emerged until today. As far as Data Lakes are concerned, being these currently the most common type of data architectures adopted by companies [2], it was possible to infer that despite all the advantages associated with them (e.g., the possibility of a large data ingestion without worrying about rigid data schemes), they have not come to efficiently satisfy the needs of the organizations. Consequently, there is a problem that must be addressed. This work allowed perception that the Data Mesh is not just about inserting new technologies into existing architectures, or adding new capabilities, or reorganizing only their components. The Data Mesh brings with it the need to change the current paradigm since the platform's infrastructure itself to the reorganization of the data teams. The fact that organizations do not consider the nature of the data when ingested, raises the problem of the lack of ownership and understanding. Now, in an organization where data is needed and produced by the majority, but owned by no one in practice, there are, as highlighted in this work, problems of data quality - which end up affecting the potentiality of their analytical value. Therefore, the Data Mesh tries to solve this problem by making the data the real organizational concern. Taking this into consideration, it is possible to synthesize that the Data Mesh ensures compliance with DATSIS principles, turns data into a product, organizes the various teams and data products according to the organizational domains (from which they emerge) and decentralizes the whole process (relieving the current central teams of the pressure from the requests they constantly receive). The Data Mesh also provides the

notion of a self-serve infrastructure, so that the various teams can create and maintain their data products. The capabilities associated with the data of a specific domain are now made available through a Mesh Node, known by all users, who can access it accordingly to the federated governance policy present in that node (e.g., node A can read the data, but Node B cannot). In this way, it is possible to overcome the bottleneck felt in the central data platform teams and ensure the quality of the data within the organization in their Data Mesh, as demonstrated by Zalando's work [23]. The analysis of the works from Netflix and Zalando [23,24] allow us to conclude that the migration process from a Data Lake architecture to a Data Mesh architecture is possible. Furthermore, it is possible to realize that the architecture previously instituted (the Data Lake) can be reused and may be part of the self-serve platform supporting the Data Mesh (e.g., to support the storage needs of each Mesh Node). However, when analysed the practical work of both contributions, it is possible to infer that there is no technical and technological consensus in the way the Data Mesh is designed and implemented, because as much as they respect most of its core concepts (domain-driven approach, data as a product, self-serve data infrastructure and federated computational governance), both present significantly different ways of tackling the problem from a technical and technological perspective. However, despite the architectural and technological disparity felt between the presented approaches, it can be inferred that they are close in terms of the core concepts provided in Zhamak Dehghani's approach. For example, although Netflix does not present in its work the Data Mesh concepts in such an explicit way like Zalando, (e.g., Data as a Product), the core concepts and design decisions being shown follow a Data Mesh architecture - a fact evidenced, for example, by the presentation of a metadata catalog, usage of standard processes catalog, quality assurance of the various data products, availability of a self-serve infrastructure, among others. To conclude, it can be assumed that there are still many open challenges that need to be studied and tackled to answer the question "What are the rigorous, concrete, and well-evaluated steps that can be followed to design and implement a Data Mesh?", to ultimately achieve a detailed approach for the design and implementation of a Data Mesh, focused on both logical and technological concerns.

Acknowledgements

This work has been supported by FCT – *Fundação para a Ciência e Tecnologia* within the R&D Units Project Scope: UIDB/00319/2020.

References

[1] Dehghani Z. How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh. 2019;1–20. Available from: https://martinfowler.com/articles/data-monolith-to-mesh.html

[2] Dehghani Z. Data Mesh Paradigm Shift in Data Platform Architecture. San Francisco, USA: InfoQ; 2020. Available from: https://www.youtube.com/watch?v=52MCFe4v0UU

[3] Lance Johnson. What is a Data Mesh? [Internet]. 2020 [cited 2020 Sep 30]. Available from: https://trustgrid.io/what-is-a-data-mesh/

[4] Dehghani Z. Data Mesh Principles and Logical Architecture [Internet]. 2020 [cited 2020 Dec 7]. Available from: https://martinfowler.com/articles/data-mesh-principles.html

[5] Santos MY, Costa C. Big Data concepts, warehousing, and analytics. River Publishing; 2020.

[6] Diebold FX. A Personal Perspective on the Origin(s) and Development of "Big Data": The Phenomenon, the Term, and the Discipline, Second Version. SSRN Electron J. 2013;

[7] Gandomi A, Haider M. Beyond the hype: Big data concepts, methods, and analytics. Int J Inf Manage [Internet]. Elsevier Ltd; 2015;35:137–44. Available from: http://dx.doi.org/10.1016/j.ijinfomgt.2014.10.007

[8] Krishnan K. Data Warehousing in the Age of Big Data. Data Warehous. Age Big Data. Elsevier; 2013.

[9] Adam Barker, Jonathan Stuart Ward. Undefined By Data: A Survey of Big Data Definitions. 2013;

[10] Santos MY, Oliveira e Sá J, Andrade C, Vale Lima F, Costa E, Costa C, et al. A Big Data system supporting Bosch Braga Industry 4.0 strategy. Int J Inf Manage [Internet]. Elsevier; 2017;37:750–60. Available from: http://dx.doi.org/10.1016/j.ijinfomgt.2017.07.012

[11] Sam Madden. From databases to big data. IEEE Internet Comput. 2012;16:4–6.

[12] Chen H, H.L.Chiang R, C. Storey V. Business Intelligence and Analytics: From Big Data To Big Impact. MIS Q [Internet]. 2018;36:1165–88. Available from: http://www.jstor.org/stable/41703503

[13] Laney D. 3D Data Management: Controlling Data Volume, Velocity, and Variety. Appl Deliv Strateg. 2001;

[14] Kimball R, Ross M. The Data Warehouse Toolkit, The Definitive Guide to Dimensional Modeling. Wiley. 2013.

[15] Golfarelli, M., & Rizzi S. Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill, Inc.; 2009.

[16] Russom P. Data Warehouse Modernization. TDWI Best Pract Rep. 2016;

[17] Costa C, Andrade C, Santos MY. Big Data Warehouses for Smart Industries. Encycl Big Data Technol. 2019;341–51.

[18] Miloslavskaya N, Tolstoy A. Big Data, Fast Data and Data Lake Concepts. Procedia Comput Sci [Internet]. The Author(s); 2016;88:300–5. Available from: http://dx.doi.org/10.1016/j.procs.2016.07.439

[19] Khine PP, Wang ZS. Data lake: a new ideology in big data era. ITM Web Conf. 2018;17:03025.

[20] Nicole Laskowski. Data lake governance: A big data do or die [Internet]. 2016 [cited 2020 Dec 15]. Available from: https://searchcio.techtarget.com/feature/Data-lake-governance-A-big-data-do-or-die

[21] Barr M. What is a Data Mesh — and How Not to Mesh it Up [Internet]. 2020 [cited 2020 Oct 29]. Available from: https://towardsdatascience.com/what-is-a-data-mesh-and-how-not-to-mesh-it-up-210710bb41e0

[22] Sven Balnojan. Data Mesh Applied [Internet]. towardsdatascience.com. 2019 [cited 2021 Mar 10]. Available from: https://towardsdatascience.com/data-mesh-applied-21bed87876f2

[23] Max Schultze & Arif Wider. Data Mesh in Practice: How Europe's Leading Online Platform for Fashion Goes Beyond the Data Lake [Internet]. 2020 [cited 2020 Dec 15]. Available from: https://www.youtube.com/watch?v=eiUhV56uVUc

[24] Justin Cunningham. Netflix Data Mesh: Composable Data Processing - Justin Cunningham [Internet]. 2020 [cited 2020 Sep 25]. Available from: https://www.youtube.com/watch?v=TO_IiN06jJ4