Data Virtualization Layer Key Role in Recent Analytical Data Architectures

Montasser Akermi(✉), Mohamed Ali Hadj Taieb, and Mohamed Ben Aouicha

Data Engineering and Semantics Research Unit, Faculty of Sciences of Sfax,
University of Sfax, Sfax, Tunisia
montaccep@gmail.com, {mohamedali.hajtaieb,mohamed.benaouicha}@fss.usf.tn

Abstract. The amount of data, its heterogeneity and the speed at which it is generated are increasingly diverse and the current systems are not able to handle on-demand real-time data access. In traditional data integration approaches such as ETL, physically loading the data into data stores that use different technologies is becoming costly, time-consuming, inefficient, and a bottleneck. Recently, data virtualization has been used to accelerate the data integration process and provides a solution to previous challenges by delivering a unified, integrated, and holistic view of trusted data, on-demand and in real-time. This paper provides an overview of traditional data integration, in addition to its limits. We discuss data virtualization, its core capabilities and features, how it can complement other data integration approaches, and how it improves traditional data architecture paradigms.

Keywords: Data virtualization · Data integration · Data architecture · Big Data

1 Introduction

New emerging technologies such as smart devices, sensors, augmented and virtual reality, robotics, biometrics, 5G, and blockchain have led to generating a huge amount of real-time; both structured and unstructured; data that flows across different environments and infrastructures. It is estimated that the global data created by 2025 will reach 175 zettabytes [30].

Often, the overwhelming of information and distributed data streams surpass the current technological capacity to manage and analyse data [27]. However, this enormous amount of data creates more opportunities for modern organizations to reduce overhead and gain a competitive advantage [29].

Before it is analyzed by data scientists and data analysts, data has to be integrated. But with the volume, variety, and velocity of data, organizations ended up creating data silos and data swamps [34]. They assumed that data lakes would solve their problems, but now these data lakes are becoming data swamps with so much data that is impossible to analyse and understand [9,18], this data is referred to as dark data [10]. This occurred because a data lake;

A. Abraham et al. (Eds.): ISDA 2022, LNNS 716, pp. 415–426, 2023.
https://doi.org/10.1007/978-3-031-35501-1_42

just like a traditional data warehouse and data lakehouse [2]; involves physically extracting and loading the data.

Even with the most recent generation of data architecture paradigm; data lakehouse; which implements data warehouse features such as ACID transactions and SQL support directly over the data lake, relying on ETL presents new challenges, particularly challenges related to query execution engines [3,20].

Data virtualization enables disparate data sources to appear as a single data store. This allows faster data integration and processing for different practitioners. Instead of moving data to a new location, data is left in its original place, while data quality and ownership are managed by data virtualization [8].

In this year's report; 2022 Gartner Magic Quadrant for Data Integration Tools, Gartner analysts once again included data virtualization as a key criterion for evaluating data integration vendors. Data virtualization is no longer a differentiating but a "must have" feature [36].

This paper aims to focus on data virtualization and its approaches to creating modern data architectures. Section 2 addresses the challenges of traditional data integration. Section 3 discusses the concept and features of data virtualization. Section 4 provides an overview of how data virtualization can be used to enhance traditional data architecture paradigms i.e. data warehouse and data lake. Finally, the last section includes the conclusion and future work.

2 Traditional Data Integration Methods

There are many possible approaches for data integration, but they can be classified into physical integration and virtual integration. The goal of data integration is to offer unified access to a set of disparate data sources [14].

2.1 Extract, Transform, Load (ETL)

The extract, transform, load process is a data integration pattern first coined in the 1970 s. This process involves: i) extracting the data from the source, ii) transforming the extracted data into the format required by the final data repository, and iii) persisting the transformed data in the final data repository. This process is the main integration pattern used in data warehouses [25].

Over the years other processes emerged from ETL, such as ELT (Extract, Load, Transform), which loads the data before making any transformation. It is especially used in data lakes. ELT improves the up-front transformation that data warehouses demand. This up-front transformation is a blocker. Not only does it lead to slower iterations of data modelling, but also it alters the nature of the original data and mutates it. Both ETL and ELT involve copying the data, thus creating more replications.

When moving data in bulk, ETL is very effective, since it is easy to understand and widely supported. Most organizations have in-house ETL. However, moving data is not always the best strategy, since data in the new location has to be maintained as well. Another disadvantage is dealing with thousands of ETL processes synchronized by some scripts, it becomes very complex to modify. And with todays data volume and variety, the ETL process is struggling.

Organizations started looking for an alternative data integration approach that supports real-time capability.

2.2 Enterprise Service Bus (ESB)

Enterprise Service Bus was originally coined by analysts from Gartner [21]. It does not involve moving the data, instead, it uses a message bus to facilitate the interactions of applications and services. Applications are connected to the message bus. This allows them to communicate and exchange messages in real-time. Applications in ESB are decoupled, therefore, no need for one application to know about, or depend on other applications [21].

Just like ETL processes, ESB is supported by most organizations. It is fitted for operational scenarios but not for analytical use cases because it cannot integrate data. ETL processes run in batches. They are often scripted, and hard to maintain over time. ESB was introduced to move away from point-to-point integration, like ETL scripts, and offer real-time interaction between applications. However, it can not integrate application data to deliver analytical use cases. In the next section, we introduce the concept of data virtualization which overcome the challenge to offer real-time data integration.

3 Data Virtualization

Data virtualization is an abstraction layer that integrates data from a wide variety of structured, semi-structured and unstructured sources; whatever the environment; to create a virtual data layer that delivers unified data services in real-time [23] to disparate data consumptions e.g. applications, processes, and users (see Fig. 1).

Data virtualization is a modern data integration solution. It does not copy the data as most data integration patterns do. Instead, it takes a different approach by providing a view of the integrated data. It keeps the source data exactly where it is [32]. Resulting in lowered costs, fewer replications, and minimum data latency.

Data virtualization can replace traditional data integration to reduce the number of replicated data marts and data warehouses [24]. But it is also highly complementary since it is a data services layer, which means it can be used between ETL, ESB, applications, whatever the environment e.g. public cloud and on-premise.

Data consumers query the data virtualization layer which gets data from various data sources. The Data virtualization layer hides where and how data is stored, and whether data needs to be aggregated, joined, or filtered before it is delivered to data consumers. Thus, the location and the implementation of the physical data, and the complexity of accessing the data are hidden from

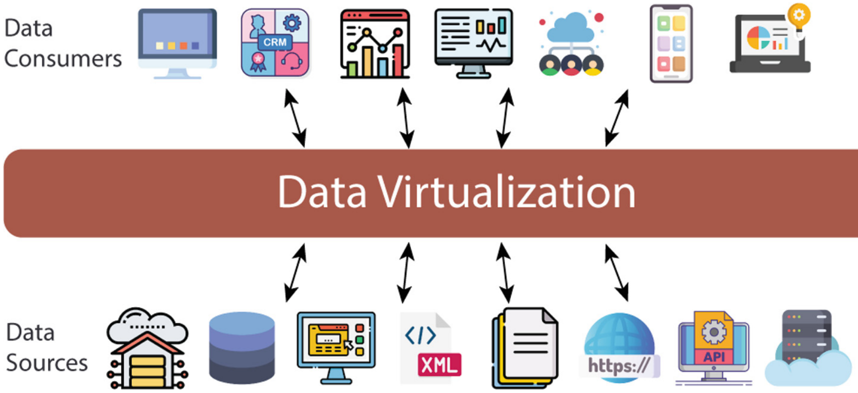

Fig. 1. Data virtualization integrates data from multiple sources and delivers it to different data consumers

data consumers [4]. The data virtualization layer takes the shape of a single data repository.

Because data is not replicated, the data virtualization layer contains only the metadata of each data source, as well as any global instruction e.g. data governance policies and data security rules.

Data virtualization is an abstraction layer. Therefore, it is highly complementary to use ETL, ESB, and other data integration patterns. In the following few sections, we discuss how is data virtualization complementing ESB and ETL processes. Furthermore, an overview of data virtualization core capabilities and features was provided.

3.1 Data Virtualization Complements ETL to Support Cloud-Based Sources

ETL was created to move data to other repositories e.g. data warehouses. However, it is not easy to move data from cloud-based sources with ETL [22]. Data virtualization can complement ETL processes to i) enable real-time data integration of multiple data sources. ii) connect on-premise with cloud-based data sources without the need to move all the data and put it in a single repository. iii) unify data across data warehouse and new on-premise or cloud-based data store. iv) use the data virtualization layer to access data faster than using ETL processes [13].

3.2 Data Virtualization Complements ESB to Add More Sources

Adding new sources to ESB can be a complex task, especially unstructured data sources e.g. web pages, flat files, email messages and cloud-based sources. To facilitate this process, data virtualization can unify these disparate sources and provide a single data source that is supported by ESB [22].

Table 1. Data integration use cases and different patterns

Use case	Patterns		
	Data virtualization	ETL	ESB
Moving data between data repositories		X	
Data unification	X		
Real-time reports and insights	X		X
Migrating data to the cloud	X	X	
Self-service analytics	X		
360 customer view	X		
Data warehouses and data marts		X	
Logical data warehouses and virtual data marts	X		
Data warehouse offloading	X	X	
Logical data lakes	X		

Table 1 shows data integration approaches that can be applied to different use cases. Data virtualization is not always the best data integration approach for a specific problem. Shraideh et al. [33] developed a structured and systematic decision support that considers fifteen critical success factors [12] to decide upon ETL, data virtualization, or a hybrid solution of both patterns as a suitable data integration approach.

3.3 Faster Data Access and Delivery

Traditional data integration patterns; such as ETL; involve physically moving the data to multiple locations e.g. data stores, databases, data warehouses, data lakes, data lakehouses, and cloud-based repositories. This process is usually done manually which creates many replications across the network. This makes the data architecture slower, costlier, and more complex [12].

Adding a data virtualization layer to the data architecture enables fast, easy, and agile solutions [17], therefore helping organizations become data-driven. In the case of self-service business intelligence [19], no need to physically move the data and aggregate it locally, business users can add multiple data sources, these sources are then connected to the data virtualization layer through pre-built data connectors, also called adapters.

The data is unified and rapidly delivered to the business intelligence system for future reports and insights because physically moving data is one of the main reasons for the high latency in traditional data architectures.

The development of data services is faster because of the unified data layer. Developers do not need to connect to all data sources that have different formats residing in different data repositories.

3.4 Self-service Analytics

With self-service analytics, business users do not need to ask the data engineering team to assist when doing analytics. The data engineering team can then focus on and improve other architectural matters. However, it can not be easily achieved because i) data is everywhere, in databases, data warehouses, cloud and big data architectures, ii) low data integrity because of no single source of truth, iii) high data latency, and iv) there is no data lineage which affects data quality and makes it questionable.

Data virtualization can overcome these challenges and makes self-service available to business users [1,7]. By this, cost and complexity are reduced, and replications are created only when it is necessary.

3.5 Data Virtualization Core Capabilities

A data virtualization layer should be at least capable of the following core principles [5,6]:

Pre-Built Connectors to quickly connect, explore, and extract the data from any on-premise or cloud-based sources and any data type: structured, semi-structured, and unstructured.

Self-Service Data Services where complexity is hidden from data consumers, data sources and data consumers are decoupled, which enables data services to be easily created without the interference of the data engineering team.

Single Logical Data Model by running automatic processing to maintain data catalogs that contain metadata, data classes, data clusters, etc.

Unified Data Governance to enable a single entry point for data, metadata management, audit, logging, security, and monitoring. External data management tools are also integrated into the system.

A Unified Data Layer is the main component of a virtual data layer. This layer harmonizes, transforms, improves quality, and connects data across different data types.

Universal Data Publishing Mechanism through unified connected services to provide users with the requested results of processed data.

Agile and High Performance where real-time optimizations are performed repeatedly to create a flexible workload [8,12].

3.6 Privacy and Data Protection

Data virtualization enables data privacy by default. It helps organizations comply with the protected by design requirement of GDPR[1]. Data sources are not required to be predefined in a particular format or accessed in a certain method. Data virtualization supports advanced data protection mechanisms e.g. anonymizing data, the immutability of data (refusal of signature), and end-to-end encryption of transmitted transactions [6].

[1] The European Unions General Data Protection Regulation.

3.7 Data Services

These services are critical to the data virtualization layer. Example of services: 360 customer view such as querying all customers who exist in all repositories, or getting the last 5 years' revenue, from all stores etc. These services are usually business-oriented, which helps business users create reports and insights easily and without needing to wait for a data engineer to build a pipeline to get the result needed. Operational-oriented services also exist, e.g. changing the address of a customer or updating his email address.

A service can be formed of three components. An interface that is responsible for handling incoming parameters and outgoing results. The logic forms the body of the service that deals with data preparation specifications. The source abstraction makes the service independent of the data source system.

These services may sound easy to implement, but that is not true. They are responsible for most of the data preparation work, such as data transformation, enrichment, joining, federation, synchronization, historicization, etc.

Some of the services might be extremely complex and need access to multiple systems, some might need to move the data, and others might need to run some machine learning algorithms. The fact that this complexity is common in ETL processes, however, with data virtualization, it is completely hidden.

The performance of a data service depends on the performance of the systems where data is stored and other services it runs. However, data services must know the best-optimized way to access these systems. It could be set-oriented by sending a request for a set of data (e.g. get a customer's info and all his invoices). Or a record-oriented approach by sending a request for one record which is relatively simple to process (e.g. get a customer's info).

The biggest performance challenge, however, is when a service joins data from multiple systems. Extracting the data and letting the service execute the joins leads to poor performance. This is because so much data is being extracted from the systems and so much data is being moved across the network. To overcome this challenge data virtualization layer needs to push-down queries as much as possible to use the power of connected data sources and get back only the requested records; or the result of the request; instead of the complete dataset.

Services must be served to data consumers as one integrated system. For example, developing a 360 customer view involves access to data from disparate repositories, however, this is hidden from the consumers. Abstraction is achieved through data services provided by the data virtualization layer.

3.8 Virtual Tables

Data virtualization helps in making the development of service logic easier. The main component of data virtualization is a virtual table.

A virtual table describes how the data sources need to be transformed. It can be used to define how data, which can be from other virtual tables, need to be prepared. A virtual table can be accessed through different interfaces, it can be published as REST services, OData[2] services, JDBC services, etc (see Fig. 2).

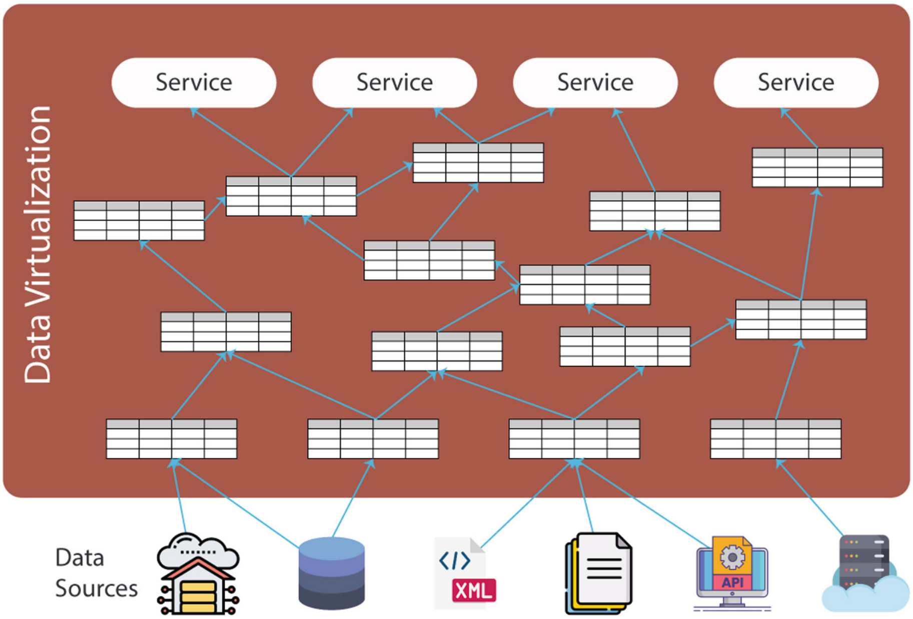

Fig. 2. Data virtualization and virtual tables

Data virtualization hides from virtual tables developers the location and the implementation of the real data. To them, it seems as if they are accessing a single logical database instead of many data sources and services, a single interface instead of all the different technologies and interfaces. This makes development faster.

3.9 Query Pushdown

Query pushdown is an optimization technique used by data virtualization. This technique consists of pushing down data processing as much as possible to the data source system e.g. data lake, database, flat file, etc. Data virtualization minimizes network traffic and uses the maximum potential of the connected data sources. For example, a NoSQL database is a scalable and highly optimized database engine [1]. Running aggregations on the database is much more efficient than running it on the data virtualization layer.

[2] Open Data Protocol.

4 Data Virtualization Architectural Approaches

In a monolithic data architecture e.g. data lake and data warehouse, data is located in a single repository. In contrast, in a distributed data architecture, data is distributed across multiple locations.

4.1 Monolithic Data Architecture Challenges

Monolithic data architectures often fail because of many challenges [5]. These challenges can be classified into three categories:

Architectural challenges such as source and use case proliferation, the continuous change of the data landscape, the non-scalable approach, and the data landscape outgrowing the data management architecture.

Technological challenges such as the tightly coupled pipelines, the inconsistent tools and various specialized skills, the complexity debt of the data pipeline, and the difficulty of sharing data between public clouds and on-premise.

Organizational challenges include the absence of data culture in the organization, the data engineers lacking domain expertise and being siloed from the operational units.

Overcoming many of these challenges is possible by integrating a data virtualization layer to enable distributed data architectures.

4.2 Logical Data Warehouses

Business analysts are experiencing new challenges with the technological revolution such as big data and cloud-based analytics [28]. New data sources e.g. data from robots and intelligent devices, social media, raw data, etc., are not structured to be suited for traditional data warehouses. It could be converted but it would increase the volume of data, therefore increasing the cost to maintain this data inside data warehouses. Organizations tend to use alternative solutions e.g. data lakes because storing data there is way cheaper than storing it in data warehouses [35]. However, since not all the data is in the data warehouse, analysts can not generate reports of all the relevant data [16].

The logical data warehouse is a data architecture that extends the traditional or enterprise data warehouse concept by adding an abstraction layer where external sources are integrated without the heavy ETL workload. The core components of a logical data warehouse architecture contain i) a layer of real-time connectors to data sources. ii) unified data layer. iii) normalized views [6,26].

Some common use cases for logical data warehouses include virtual data marts, integration of multiple data warehouses, and data warehouse offloading.

4.3 Logical Data Lakes - Big Data Virtualization

Just like data lakes, big data virtualization applies the schema-on-read principle. This approach can handle the massive volume and variety of data [11]. As mentioned before, data virtualization allows the data to stay in its original data storage, while being available to data consumers such as application and business users. To find datasets in the systems, a data catalog that presents all the data; including physical data lakes; is needed.

Data catalog stores metadata of all the data inside the different systems connected to the data lake. Thus, the search functionality runs rather quickly. The data catalog is the main interface of logical data lakes. When a user searches for data, it does not matter where it has physically located. The search process is always processed the same way [11].

5 Conclusion and Future Work

Data virtualization is a decoupling technology, that offers many features which overcome traditional data integration patterns and enhance analytical data architectures. It increases the efficiency of data operations, minimizes replications, and reduces complexity and cost. Data virtualization enables agile data services, real-time data delivery, and self-service for different users without the intervention of the data engineering team.

This shifts the paradigm from data storage to data usage, and from moving data to connecting data. Therefore, creating Data as a Service (DaaS) [31] where data is not moved from the source system to the target system. Instead, the target system can ask for data on demand. It can use services offered by the source system to manipulate data.

Data virtualization changed the way data is looked at and how data services are developed. The next information revolution will give more attention to semantics and the meaning of the distributed data through data virtualisation.

Data virtualization offers more opportunities for data management solutions, thus creating new data architectures which will stand out when more heterogeneous data comes in.

For future work, we recommend evaluating the performance of the logical data warehouse and the data lakehouse using the same data.

We also recommend creating a conceptual model or a reference architecture of the data virtualization. Another future work would be creating a new generation of data virtualization that supports many data management capabilities.

Lastly, we have not seen any implementation of data virtualization in the lakehouse architecture. Therefore, future work should focus on the data lakehouse and the newly emerging architectures in general and the role of data virtualization in these architectures.

References

1. Alagiannis, I., Borovica, R., Branco, M., Idreos, S., Ailamaki, A.: NoDB: efficient query execution on raw data files. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 241–252 (2012)
2. Armbrust, M., Ghodsi, A., Xin, R., Zaharia, M.: Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In: Proceedings of CIDR (2021)
3. Behm, A., et al.: Photon: a fast query engine for Lakehouse systems. In: Proceedings of the 2022 International Conference on Management of Data, pp. 2326–2339 (2022)
4. Bogdanov, A., Degtyarev, A., Shchegoleva, N., Khvatov, V.: On the way from virtual computing to virtual data processing. In: CEUR Workshop Proceedings, pp. 25–30 (2020)
5. Bogdanov, A., Degtyarev, A., Shchegoleva, N., Khvatov, V., Korkhov, V.: Evolving principles of big data virtualization. In: Gervasi, O., et al. (eds.) ICCSA 2020. LNCS, vol. 12254, pp. 67–81. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58817-5_6
6. Bogdanov, A., Degtyarev, A., Shchegoleva, N., Korkhov, V., Khvatov, V.: Big data virtualization: why and how? In: CEUR Workshop Proceedings (2679), pp. 11–21 (2020)
7. Chatziantoniou, D., Kantere, V.: Datamingler: a novel approach to data virtualization. In: Proceedings of the 2021 International Conference on Management of Data, pp. 2681–2685 (2021)
8. Earley, S.: Data virtualization and digital agility. IT Professional **18**(5), 70–72 (2016)
9. Eryurek, E., Gilad, U., Lakshmanan, V., Kibunguchy-Grant, A., Ashdown, J.: Data governance: the definitive guide. “O' Reilly Media, Inc.” (2021)
10. Gartner: Definition of dark data - it glossary. https://www.gartner.com/en/information-technology/glossary/dark-data. Accessed 14 Apr 2022
11. Gorelik, A.: The enterprise big data lake: delivering the promise of big data and data science. O'Reilly Media (2019)
12. Gottlieb, M., Shraideh, M., Fuhrmann, I., Böhm, M., Krcmar, H.: Critical success factors for data virtualization: a literature review. ISC Int. J. Inf. Secur. **11**(3), 131–137 (2019)
13. Guo, S.S., Yuan, Z.M., Sun, A.B., Yue, Q.: A new ETL approach based on data virtualization. J. Comput. Sci. Technol. **30**(2), 311–323 (2015)
14. Halevy, A., Doan, A.: Zgi (autor). Principles of data integration (2012)
15. Hilger, J., Wahl, Z.: Graph databases. In: Making Knowledge Management Clickable, pp. 199–208. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-92385-3_13
16. Kukreja, M.: Data engineering with apache spark, delta lake, and Lakehouse. “Packt Publishing Ltd.” (2021)
17. Van der Lans, R.F.: Creating an agile data integration platform using data virtualization. R20/Consultancy technical white paper (2014)
18. Van der Lans, R.F.: Architecting the multi-purpose data lake with data virtualization. Denodo (2018)
19. Lennerholt, C., van Laere, J., Söderström, E.: Implementation challenges of self service business intelligence: a literature review. In: 51st Hawaii International Conference on System Sciences, Hilton Waikoloa Village, Hawaii, USA, 3-6 Jan 2018, vol. 51, pp. 5055–5063. IEEE Computer Society (2018)

20. LEsteve, R.: Adaptive query execution. In: The Azure Data Lakehouse Toolkit, pp. 327–338. Springer (2022). https://doi.org/10.1007/978-1-4842-8233-5_14
21. Menge, F.: Enterprise service bus. In: Free and open source software conference, vol. 2, pp. 1–6 (2007)
22. Miller, L.C.: Data Virtualization For Dummies, Denodo Special Edition. "John Wiley & Sons, Ltd." (2018)
23. Mousa, A.H., Shiratuddin, N.: Data warehouse and data virtualization comparative study. In: 2015 International Conference on Developments of E-Systems Engineering (DeSE), pp. 369–372. IEEE (2015)
24. Mousa, A.H., Shiratuddin, N., Bakar, M.S.A.: Virtual data mart for measuring organizational achievement using data virtualization technique (KPIVDM). J. Teknologi **68**(3), 2932 (2014)
25. Muniswamaiah, M., Agerwala, T., Tappert, C.: Data virtualization for analytics and business intelligence in big data. In: CS & IT Conference Proceedings. CS & IT Conference Proceedings (2019)
26. Offia, C.E.: Using logical data warehouse in the process of big data integration and big data analytics in organisational sector, Ph. D. thesis, University of the West of Scotland (2021)
27. Oussous, A., Benjelloun, F.Z., Lahcen, A.A., Belfkih, S.: Big data technologies: a survey. J. King Saud Univ.-Comput. Inf. Sci. **30**(4), 431–448 (2018)
28. Papadopoulos, T., Balta, M.E.: Climate change and big data analytics: challenges and opportunities. Int. J. Inf. Manage. **63**, 102448 (2022)
29. Raguseo, E.: Big data technologies: an empirical investigation on their adoption, benefits and risks for companies. Int. J. Inf. Manage. **38**(1), 187–195 (2018)
30. Reinsel, D., Gantz, J., Rydning, J.: The digitization of the world from edge to core. Framingham: International Data Corporation, p. 16 (2018)
31. Sarkar, P.: Data as a service: a framework for providing reusable enterprise data services. John Wiley & Sons (2015)
32. Satio, K., Maita, N., Watanabe, Y., Kobayashi, A.: Data virtualization for data source integration. IEICE Technical Report; IEICE Tech. Rep. **116**(137), 37–41 (2016)
33. Shraideh, M., Gottlieb, M., Kienegger, H., Böhm, M., Krcmar, H., et al.: Decision support for data virtualization based on fifteen critical success factors: a methodology. In: MWAIS 2019 Proceedings (2019)
34. Skluzacek, T.J.: Automated metadata extraction can make data swamps more navigable, Ph. D. thesis, The University of Chicago (2022)
35. Stein, B., Morrison, A.: The enterprise data lake: better integration and deeper analytics. PwC Technol. Forecast: Rethinking Integr. **1**(1–9), 18 (2014)
36. Zaidi, E., Menon, S., Thanaraj, R., Showell, N.: Magic quadrant for data integration tools. Technical report G00758102, Gartner, Inc. (2022)