Data Fabric Technologies, Modeling and Applications – A Review

Radha Krishna Pisipati[1,*,†], Kamalakar Karlapalem[2,†] and Satyanarayana R Valluri[3,†]

[1]Department of Computer Science and Engineering, National Institute of Technology Warangal, India
[2]Data Science and Analytics Centre, IIIT-Hyderabad, India
[3]Databricks Inc., California, USA

Abstract

Data Fabric is an amalgamation of various database system technologies, offering extensive research opportunities for deploying end-to-end data management platform-based solutions. These platforms have seen advancements in middleware, advanced and powerful ETL pipelines, and generative AI-supported data pipelines with unified storage and compute to establish compliance and governance and reduce latency. Deployed systems using technologies such as data mesh, data lakes, data warehouses, and cloud databases serve as data sources, and the data fabric solution manages data, query, and analytics pipelines by leveraging distributed computing capabilities and dynamically routing queries for optimal performance without centralizing data storage. Understanding the interconnections (technology and applications) among source systems, data fabric, domain, and application is crucial for establishing correct and complete data fabric solutions for user applications. This paper presents a holistic view of data fabric technologies and addresses the importance of understanding the interconnections among source data systems, data fabric, domain, and application, focusing on metadata and application development. For metadata, we envisage an ER model solution to provide an overall conceptual data landscape for the underlying data systems for a data fabric.

Keywords

Data management, distributed data sources, data fabric technologies

1. Introduction

Researchers and practitioners developed various data platforms and storage technologies [1], such as data mesh, data lakes, data warehouses, and cloud and federated databases to manage flows of large amounts of data for queries and analytics. Each technology serves distinct but complementary roles within an organization's data ecosystem. Though one considers these technologies as silos but the data that they manage are interrelated and encompass the same organization's multiple needs. Thus, from the query and data analytics point of view, it is important to have a seamless and unified view of data and the technologies (different perspectives) to query and manage the data. Data fabric, a recent development in data management, offers a comprehensive architecture to integrate disparate data pipelines [2] seamlessly, reducing latency while ensuring robust data governance and compliance.

The database became necessary because traditional file systems with applications created data redundancy problems, no visible schema, and repeated functionality implementation. A distributed database is an integrated, interrelated, multiple databases in different systems. The key idea behind the technology is the complexity of the technology to support various levels of transparency in accessing the distributed database. A distributed database system can have homogeneous or heterogeneous database systems, an integrated system, or multi-database system solutions executed as per application. A data warehouse is a specialized database that provides a subject-oriented, integrated, time-variant, non-volatile database for online analytical processing (mostly business intelligence reports and data mining). The core technology for a data warehouse is a data cube, a multidimensional relational table that efficiently supports aggregates (data cube operations) across trillions of rows. A data lake is a central repository of data that allows ingesting, storing, processing, and analyzing large volumes of multi-modal data in real-time. This includes structured data (such as relational databases),

ER2025: Companion Proceedings of the 44th International Conference on Conceptual Modeling: Industrial Track, ER Forum, 8th SCME, Doctoral Consortium, Tutorials, Project Exhibitions, Posters and Demos, October 20-23, 2025, Poitiers, France
*Corresponding author.
prkrishna@nitw.ac.in (R. K. Pisipati); kamal@iiit.ac.in (K. Karlapalem); satya.valluri@databrics.com (S. R. Valluri)
0000-0002-2764-2818 (R. K. Pisipati); 0000-0003-2528-7979 (K. Karlapalem); 0000-0003-2314-2928 (S. R. Valluri)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-11-17

CEUR-WS.org/Vol-4099/ER25_tutorial_PRKrishna.pdf

semi-structured data (like log files, CSV, XML, JSON), and unstructured data (such as text, images, audio, and video). Unlike traditional data storage systems, data lakes store raw data in its native format and structure and thus avoid (expensive) preprocessing or modeling before storing or analyzing. Data lakes are often considered architecture-less, as they do not impose a specific structure or schema on stored data. Creating and maintaining a functional data lake requires integrating multiple technologies for data ingestion, processing, storage, and exploration tasks. A data Lakehouse, a large-scale data storage management system, blends the flexibility of data lakes with the structured data management capabilities of data warehouses. Data mesh emphasizes decentralization of data ownership and governance, advocating for domain-oriented teams responsible for managing data within their respective domains. Data lakes provide flexibility and agility for storing diverse data types and formats. Data mesh and data lake are decentralized approaches to managing and organizing data within an organization. Data warehouse adopts a centralized and schema-on-write approach. A data fabric abstracts the intricate technical processes involved in data movement, transformation, and integration, ensuring universal data accessibility throughout the enterprise. The key idea of Data fabric is to design data pipelines with the principle of loosely coupling data across platforms and applications, facilitating seamless access to data present across distributed and heterogeneous environments, including on-premises and cloud-based systems. A data pipeline is supported by metadata to route the query to the appropriate data sources. Once the metadata is in place, this routing can be done automatically without the application programmer explicitly coding it. Data fabric architectures manage query and analytics pipelines by leveraging distributed computing capabilities without moving data to a centralized location.

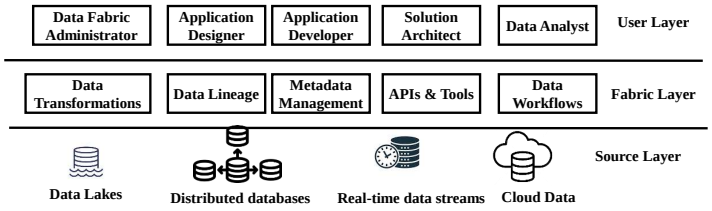

Figure 1: A three-layer view of Data Fabric

Figure 1 shows a three-layered view of a data fabric. At its base lies the source layer, composed of data lakes, distributed databases, real-time data feeds, cloud data, and other data sources. The User layer consists of various stakeholders, including application designers, developers, solution architects, and data analysts. The job of users includes: (i) understanding the data required by the application using domain knowledge, (ii) creating a schema that developers can use to write queries against, and (iii) defining ETL (Extract, Transform, Load) pipelines to extract the necessary data and populate the tables defined by the schema. All these tasks necessitate metadata information, which enables them to comprehend how to connect the actual data sources with the schema's tables and write efficient ETL pipelines to extract the required data.

2. Data Modeling

Modeling approaches can be broadly categorized into data-driven and query-driven methodologies [3]. Data-driven approaches initiate from a detailed analysis of the data sources, whereas query-driven methodologies start from users' requirements. Schema-on-read (e.g., data lakes) and schema-on-write (e.g., data warehouse) concepts are used in handling data schema in data storage and management systems. In a data lake, data is stored in its raw form without any predefined schema or structure imposed upon it at the time of ingestion. Instead, the data is ingested into the data lake in its native format, whether structured, semi-structured, or unstructured. The schema-on-read paradigm means that the data schema is applied when it is accessed or queried, rather than when the data is initially stored. This allows for flexibility in handling diverse data types and formats, as the data can be interpreted and processed according to the requirements of specific analytical or processing tasks. The schema-on-read approach enables organizations to store large volumes of raw data without needing upfront schema design, promoting agility and adaptability in data analytics and exploration. Data modeling in a data fabric involves creating a unified, flexible representation of diverse and distributed data sources. It must support varying data types, evolving schemas, and real-time updates while ensuring governance

and integration across platforms. Further, the modeling must support analytics while maintaining scalability and adaptability within the data fabric architecture.

3. Technologies and Applications

Data fabric technologies facilitate the acquisition, access, transformation, management, intelligence, orchestration, discovery, and governance of data without requiring explicit knowledge of data format. Categorized into technical, operational, business, social, passive, and active types, Data Fabric metadata plays a crucial role in understanding and profiling an organization's data assets and facilitating seamless integration across various platforms and technologies [4]. Thus, metadata management is essential for providing context and insights into the data stored within the fabric. It involves processes and tools for capturing, storing, and managing metadata, including information about data lineage, quality, usage, and relationships. Scalability and resilience are essential characteristics of data fabric that enable it to adapt to evolving application needs. Further, understanding and applying conceptual relationships such as keys, composite attributes, nary relationships, generalization, aggregation, and constraints are essential for designing robust, efficient, and effective data fabric that aligns with the underlying business logic and meets the desired functionality requirements of various applications over data fabric.

Data fabric components are built using various technologies designed to handle the complexity and scale of modern data environments. The widely used file formats for storing large data sources include CSV, Parquet, ORC, Avro, and Feather. Many of these formats include metadata within their storage [3]. Parquet and ORC are columnar storage formats optimized for read-heavy workloads, offering efficient compression, and encoding schemes. Avro is a row-based format that supports schema evolution, making it suitable for data serialization. Feather, developed by Apache Arrow, provides fast read and write operations, ideal for data exchange across platforms. Metadata file formats (such as Hive, Iceberg, Delta, and Hudi) manage large-scale data on distributed storage systems [5]. Hive provides a data warehousing solution with SQL-like querying. Iceberg offers table format capabilities for handling high-performance data lakes with ACID transactions. Delta Lake provides similar functionality with additional support for scalable metadata handling and schema enforcement. Apache Hudi supports incremental data processing and stream ingestion, making it suitable for real-time data lake architectures.

Data integration tools use mechanisms such as connectors, data ingestion pipelines, and ETL processes to ensure data quality and consistency during integration. Schema matching [6] algorithms identify semantic correspondences between the elements of two schemas using their structural and syntactic pattern. Metadata extraction, classification, and tagging mechanisms are used to automate the creation and updating of metadata [7]. Data governance and security mechanisms provide access control policies (e.g., Role-based Access Control, encryption techniques (e.g., Advanced Encryption Standard, and compliance management systems to secure and comply with regulations. MapReduce, machine learning algorithms, and real-time stream processing algorithms are used to process and analyze data by leveraging distributed computing frameworks such as Apache Hadoop and Spark for efficient parallel processing of large datasets. Data orchestration tools (e.g., [8]) use workflows and event-driven architectures to automate data movement through the pipeline, ensuring the data flows well and efficiently. Architecting a data fabric involves designing data workflows and potentially restructuring data to cater to varied user needs [9, 2]. Data fabric architectural frameworks integrate domain models and metadata structures tailored for diverse applications, including additive manufacturing [10] and road transport [11] systems. A data fabric needs a complex landscape characterized by numerous interconnected technologies, solutions, and diverse data and control flows for robust data management. Within this environment, varying levels of abstraction often leave end users struggling to identify the pertinent data and formulate relevant queries to meet application requirements—especially as these needs evolve dynamically over time. Automated data management trends in the industry are moving towards zero-ETL, No-Code data orchestration, Machine Learning pipelines, metadata discovery, cross-domain schema matching, etc., to reduce latency and improve governance in the data fabric environment.

4. Open Problems

Incorporating data governance, lineage, security, and privacy within a data fabric framework remains an ongoing research area, particularly in developing standardized metadata management and data

cataloging practices. This is because comprehensively understanding the metadata for one or more data lakes and their interrelationships is challenging. Achieving seamless consolidation across heterogeneous and distributed data sources requires advanced modeling techniques to handle varying data types, schemas, and formats and establish connections among them. Further, the dynamic nature of data sources and pipelines necessitates models that can adapt in real time to changes in data structure and volume while maintaining performance, scalability, and adaptability.

Ensuring secure and non-harmful operations within a data fabric is crucial, as it involves managing complex data pipelines, which in turn access multiple data sources. Moreover, it is essential to guarantee that the data fabric is always performing its intended functions, for instance, ensuring users access the right and complete data without getting lost among multiple data pipelines, which could obscure end-user abstraction. Metadata reasoning is an open problem to support correct and on-the-fly data integration and governance across varied platforms and applications and executing multiple data pipelines in parallel. These challenges hinder fully realizing the data fabric's potential, especially in large-scale, enterprise-level applications where performance, reliability, and resilience are critical.

Though LLMs are used to generate data pipelines and determine the metadata relevant to user queries, to comprehend the metadata, that needs to be mapped to a conceptual ER model using LLMs. Techniques to validate data pipelines across conceptual data models need to be developed. A toolkit to support conceptual model-driven data pipeline establishment is required. The evaluation of whether the data pipeline provides the complete result for a user query needs to be formulated along with techniques to establish the completeness of the data pipeline result.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase, and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] E. Hechler, M. Weihrauch, Y. Wu, Data fabric and data mesh approaches with AI, Berkeley, CA, USA: Apress Berkeley (2023).

[2] K. Liu, M. Yang, X. Li, K. Zhang, X. Xia, H. Yan, Data-fabric: A data fabric system based on metadata, in: 2022 IEEE 5th International Conference on BDAI, 2022, pp. 57–62.

[3] R. Hai, C. Koutras, C. Quix, M. Jarke, Data lakes: A survey of functions and systems, IEEE Transactions on Knowledge and Data Engineering 35 (2023) 12571–12590.

[4] R. Barik, Data fabric primer, 2022. URL: https://www.globallogic.com/in/wp-content/uploads/sites/21/2023/12/Paper-Data-Fabric-primer.pdf.

[5] M. Stonebraker, A. Pavlo, What goes around comes around... and around..., ACM Sigmod Record 53 (2024) 21–37.

[6] E. Rahm, E. Peukert, Large-scale schema matching, in: Encyclopedia of Big Data Technologies, Springer, 2019, pp. 1105–1110.

[7] M. Cherradi, A. El Haddadi, Ememodl: Extensible metadata model for big data lakes, International Journal of Intelligent Engineering and Systems 16 (2023) 231–243.

[8] K2View, Data transf. & orchestration, 2023. URL: https://www.k2view.com/platform/data-orchestration-tools/, accessed: 2025-09-02.

[9] A. McSweeney, Designing an enterprise data fabric, 2019. URL: https://www.researchgate.net/publication/333485699_Designing_An_Enterprise_Data_Fabric.

[10] P.-O. Östberg, E. Vyhmeister, G. G. Castañé, B. Meyers, J. Van Noten, Domain models and data modeling as drivers for data management: The assistant data fabric approach, IFAC-PapersOnLine 55 (2022) 19–24.

[11] S. A. Rieyan, M. R. K. News, A. M. Rahman, S. A. Khan, S. T. J. Zaarif, M. G. R. Alam, M. M. Hassan, M. Ianni, G. Fortino, An advanced data fabric architecture leveraging homomorphic encryption and federated learning, Information Fusion 102 (2024) 102004.