Exploring the Benefits of Blockchain-Powered Metadata Catalogs in Data Mesh Architecture

Anton Dolhopolov Arnaud Castelltort Anne Laurent

LIRMM, Univ. Montpellier, CNRS, Montpellier, France
{firstname.lastname}@lirmm.fr

Abstract. The Data Mesh architecture is gaining popularity as a new approach to data management within modern organizations. A key component of this architecture is the metadata catalog, which provides a centralized repository for documenting and discovering data products across different teams. However, maintaining an accurate and secure catalog in the context of this architecture is challenging. In this paper, we explore the benefits of using blockchain technology to power the metadata catalog in a Data Mesh architecture. Indeed, blockchain provides a decentralized and immutable ledger that can help to ensure the accuracy and consistency of metadata across different teams. It provides a secure and transparent way of tracking changes and access to the metadata. We discuss the potential advantages of using blockchain for metadata catalog, including data security, data provenance, and data ownership. We also examine the potential challenges and limitations of using blockchain technology in the context of Data Mesh architecture. Overall, we argue that blockchain-powered metadata catalogs have the potential to enhance the efficiency and reliability of data management in a Data Mesh.

Keywords: Metadata · Blockchain · Data Mesh · Data Governance

1 Introduction

Over the last 30 years research and industry came up with different approaches of building analytical systems. *Data warehouse* [8] was the first paradigm of its kind, tailored to analyzing mainly the structural data: relational databases, XML files, tabular data. It focuses on calculating statistical information over large amounts of data and then constructing multidimensional cubes. Then *slice-and-dice* or *drill-down* operations are used for fast navigation over computed values.

Data lake [11] was the second paradigm that went further into analyzing all kinds of data: structured, semi-structured, and unstructured. But the most significant change was collecting raw data, without preliminary schema enforcement. It is known as *schema-on-read*: when the final shape of data is determined only at the analysis phase, not before storing it. It gives a way to apply various types of analyses at different times, potentially extracting distinct insights.

Data mesh [3] is the most recent paradigm, which goes further into changing the way modern data platforms are built. Previous generations focused on creating big data monoliths, where separated teams work on disjoint parts of a global

IT system: operational applications on one side, and analytical data platform on another. The mesh stipulates the organizational change towards decentralization in the first place and adapts bounded business contexts for building both operational and analytical products by joint teams of engineers.

The key element of building any kind of data platform is a metadata management system [14]. It is necessary for preventing the formation of *data swamps*, which can easily happen when collecting large amounts of data. Most of the time, metadata management involves dealing with data cataloging (necessary for querying and discovery), data schemes, lineage, quality rules definition, technical and semantical information, etc. It can also integrate user activity tracking, security and privacy rules, and other policies.

Blockchains are well suited for developing solutions aimed at privacy, audibility, tamper-proof storage, and trust verification. It makes the technology applicable in the industries like finance, healthcare, or Internet-of-Things (IoT) [1]. Some recent research also attempted to implement metadata management by using the blockchain. In [4, 9] authors use it for provenance metadata registration, sensitive data access verification, and tracking. [12] proposes methods for preventing information leakage through metadata sharing. [10] considers the Hadoop Distributed File System (HDFS) platform environment and moves the central metadata node to a distributed network, while [5] offers a general model for deploying metadata using the blockchain.

In this paper, we review the Data Mesh architecture and point out the problems of centralized metadata systems. We compare the decentralization options to the properties of a blockchain solution and show, that the latter offers a range of benefits for implementing the catalog, but brings its own set of constraints.

2 Data Mesh Application for Inter-Organizational Work

2.1 Data Mesh Definition

Data Mesh helps to develop and deploy new services and products without facing administrative and communication delays. It proclaims the idea of data platform decentralization, just like microservice architecture is used for the application platform. The core principles adapt a domain-driven design around data and code, not just a single functional code part. In [3] authors define the following principles: 1) distributed data domains; 2) data-as-a-product; 3) self-serve infrastructure platform; 4) federated computational governance.

2.2 Running Example

To illustrate the mesh, we show a running example of a video-service company that comprises Human Resources, Sales, and Studio Production departments which are equivalent to different domains. Figure 1 shows its main components. For simplification, we assume that each team manages a single data domain.

In this example, the Studio team is occupied with both transactional and analytical online processing (OLTP and OLAP). The domain output can be one

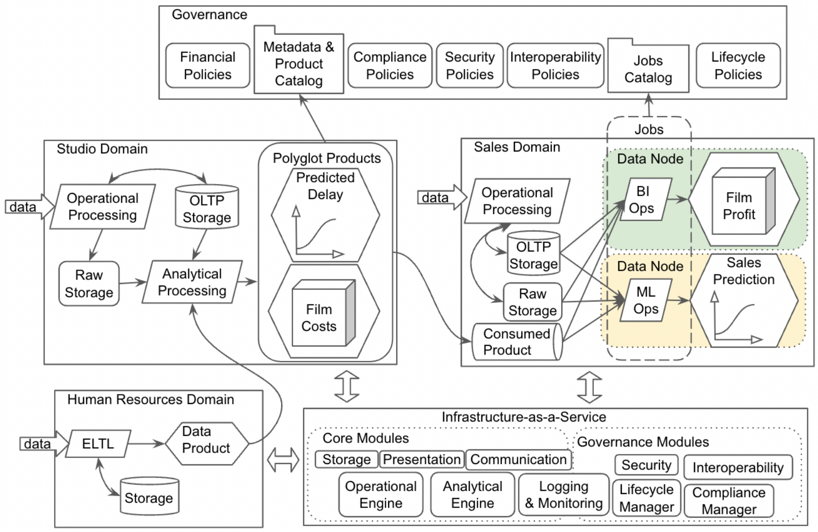

Fig. 1. Data mesh running example architecture overview.

or more polyglot data products, that is actual or derived data (e.g. IoT events, business intelligence cubes, machine learning models) published for use by any other domain or the end user.

Since there are a lot of products produced by a number of teams, it is a good practice to register them in the Product (data) Catalog alongside metadata describing their owner, purpose, lineage, mode of use, etc. In our case, the product is consumed by the Sales team, which also operates its own OLTP and OLAP.

By consuming Studio reports, Sales will improve the delivered value from data and potentially enhance the number of published products. Because different teams may apply similar and configurable computation over data, we could also have a Jobs (processing) Catalog that assembles common data operations and helps to create new products.

The common infrastructure platform provides another abstraction layer over the computing platform. It helps to put in place monitoring and automatic control execution on all domains at once, to provide service communication interoperability and avoid resource provisioning overhead when working with different technological stacks. In modern days, cloud providers (e.g. Microsoft Azure, Amazon Web Services) can be seen as good candidates for such a platform.

The federated computational governance is necessary for well-coordinated work and automated enforcement of mesh policies. Here governance is a collection of policies, standards, objectives, metrics, and roles aimed to maximize the value of data. Metadata management is one of its key elements. Product Catalog and Jobs Catalog can be seen as management system replacements. The policies can state the requirement of registering all the necessary information about the code and data products to the catalogs for improving the platform.

2.3 Implementing Inter-Organizational Collaboration

Distributed domains architecture opens a way for cross-organizational interaction when data from a company can be used by another one. Data mesh supports the implementation of business processes for storing and exchanging data in a decentralized environment with loosely connected collaborating domains or even organizations. In our scenario, we could have dozens of Studio domains (e.g. fiction, documentary, educational) that are dedicated to the production of different video content. In fact, each Studio could operate independently. But on the scale of large enterprises, connecting a big number of such independent domains created around the bounded business context will form a global data mesh.

Next, we review the challenges of implementing metadata systems.

3 Challenges of Metadata Management in Data Mesh

Previously we saw the core principles of the data mesh architecture. Although it promises to improve the performance of large enterprises, it also brings a set of challenges necessary to resolve.

3.1 Metadata Catalog as a Governance Machinery

The main aspect of coordinating the distributed environment is well-established governance. Generally, governance has a lot of important elements, but we focus on metadata management further.

Metadata systems help users to discover, understand, and use data produced by the organization. In our previous example, Sales and Studio teams are responsible for creating and managing the data products, including their metadata. But in the literature [2,6,7,15,16], such systems are built as a centralized repository that contains information on all the data available in the platform. A single metadata repository implies governance centralization with issues like a single point of failure (SPoF) [10], single source of truth (SSoT) [9], or product upgrade delays [13]. It becomes even more difficult to operate in a cross-organization environment since it is not clear who and how owns and operates such catalogs.

3.2 Metadata Catalog Decentralization Requirements

Metadata catalog decentralization is a solution that removes the SPoF by tolerating domain failures. But it also must conform to a list of requirements.

First, the different domains must be interoperable and implement common taxonomies, definitions, ontologies, etc. Otherwise, it will add unnecessary complexity and cause operating miscomprehension.

Second, we should have ways to verify, audit, and define the rules for sensitive (meta)data access. Immutable data structures are essential in these scenarios. In highly regulated fields (e.g. finance) it is of great importance to have security methods for preventing and prosecuting illegal activities. In an inter-organization collaboration, the verification process also helps to handle issues related to SSoT.

Third, as with the evolutionary nature of enterprise processes, we should be able to upgrade the active policies in place without interrupting the whole system. The situation when all domain teams declare and start using the new system rules is highly unlikely and error-prone. Thus, the system upgrade to a new set of rules should be automatic and without human intervention.

Next, we describe our research work for implementing these requirements.

4 Research Contribution

In Section 2, we saw how the data mesh could be used for cross-organization projects, and in Section 3 we saw some challenges associated with it. In this section, we provide our research contribution to solving these issues.

First, we make a renewed view of data mesh governance options by defining 3 formal types of metadata catalogs: centralized, distributed, and decentralized.

Second, we show how blockchain technology fulfills the requirements for building a distributed Type II metadata catalog and describe its application with our updated, blockchain-based running example.

The section concludes with some new challenges and potential solutions.

4.1 Defining the Data Mesh Governance Types

On one side, centralized governance dictates how each domain should function, which defeats the goals of data mesh in the first place. On the other side, interoperability is a big challenge of complete decentralization. In Figure 2, we define the data mesh governance scale that helps to understand better how we can configure and build our metadata catalog. The horizontal axis shows the proposed deployment modes ranging from centralization (left) to decentralization (right).

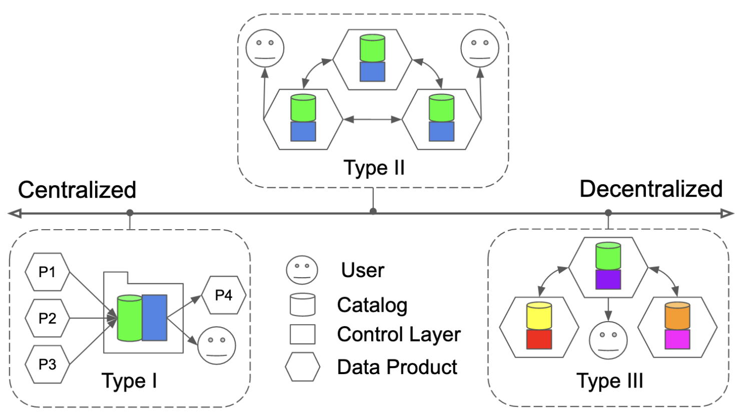

Fig. 2. Data Mesh Governance Scale

Type I - Centralized Metadata Catalog. The centralized catalog implies that a set of all metadata records $m_1, ...m_n = M$ is stored in a single repository C. Upon the release of a data product p_x, the new metadata record m_x is pushed to the repository in order to make the product discoverable and accessible by other domains. To have control over the records, a repository operator can define the visibility map function $v : (u, m) \rightarrow \{1, 0\}$, and the access function $a : (u, m) \rightarrow \{Read, Write, Delete\}$ that are associated with user profiles U.

In total, metadata catalog is defined as $C = \{P, M, U, d_m, v_m, a_m\}$, where:

- P is a set of data products
- M is a set of metadata records
- U is a set of catalog users
- $d_m : P \rightarrow M$ is a function returning the metadata of a product (description)
- $v^u_m : U \times M \rightarrow \{1, 0\}$ is a function returning a visibility map for pairs (u, m)
- $a^u_m : U \times M \rightarrow \{Read, Write, Delete\}$ is a function returning a permissions map for pairs (u, m).

Type II - Distributed Metadata Catalog. The middle state between Type I and Type III is a distributed catalog. There is no central repository and each domain d hosts a complete copy r of the metadata catalog on a dedicated metadata node n which resolves the SPoF problem. Meanwhile, there must be a peer-to-peer (P2P) system that keeps the data in sync across all nodes and the contracts or techniques T defined for enforcing the unified behavior B to be followed by all participants.

The shared catalog is $C = \{P, M, U, B, D, R, N, T, d_m, v_m, a_m, t_b, s_r\}$, where:

- D is a set of data domains
- B is a set of unified governing policies (behaviors)
- R is a set of metadata catalog replicas
- N is a set of metadata nodes that host the replicas
- T is a set of techniques (contracts) that enforce the policies
- $t_b : D \times N \times B \rightarrow T$ is a function returning a contract map for triples (d, n, b)
- $s_r : R \times R \rightarrow \{1, 0\}$ is a function returning the consistency state map for any two given replicas (synchronization state).

Type III - Decentralized Metadata Catalog. The decentralized catalog is the case where each domain (or a subset of domains) uses different technologies and policies for managing its catalogs. Each team independently owns and serves the (meta)data products, and in order to provide discoverability and querying one must interlink its metadata with other domain's metadata. It can be the case that n given domains can configure a shared distributed catalog, but also have their own private catalogs in parallel.

Therefore, the metadata catalog $\mathcal{C}$ is defined as:

- $\mathcal{C} = \bigcup_{i=1,n_d} C_i$ with C_i being the catalog associated with $D_i \subset \mathcal{D}$ and $D_i \neq \emptyset$
- $\mathcal{D} = \bigcup_{j=1,n_d} d_j$ is a set of all data domains
- $l : M \times M \rightarrow \{1, 0\}$ is a function establishing the link presence or absence between a pair of metadata records.

4.2 Blockchain-Powered Type II Metadata Catalog

Metadata catalog distribution comes with requirements such as consistency, immutability, auditing, versioning, and access control. Distributed databases have been a subject of research and industrial use for a long time. But their main drawback is a lack of trust verification methods in a multi-party environment.

At the same time, blockchain research demonstrates that we can use it within a virtual organization (VO) for implementing a joint project [4] or data exchange process with traceability and accountability [9]. For instance, VO can be formed by distributed data domains for inter-organizational collaboration.

Benefits of Using Blockchain. Blockchains possess a number of characteristics that fit the needs for implementing the Type II metadata catalog.

Blockchain benefits from the immutable append-only ledger that can store the metadata records M permanently and gives records version history. The blocks hashing function adds a trust component that is essential for verification and auditing. The network nodes can hold a ledger copy with all the records. It can be seen as equal to metadata nodes N that store the replicas R.

The implementation of smart contracts enforces the automatic execution of the globally defined, unified governing policies (a set of techniques T).

The distributed consensus algorithm that guarantees ledger synchronization is a form of consistency function s_r. Upon the business process evolution, it also provides a way to make an automatic upgrade to the new set of policies B'.

Blockchains can also be classified as public or private. Public ones make no assumption about the identities of the network users meaning that all participants are anonymous and equal: they have the same rights and can perform the same functions. It makes the network more vulnerable to malicious activities.

By contrast, private blockchain gives more control over the users. The bad actors can be identified and access can be denied since the identities are known in advance. Therefore, it is more favorable compared to a public one as it supports the implementation of visibility and access functions (v_m^u and a_m^u).

Blockchain Catalog Running Example. Our new diagram of the distributed metadata catalog is shown in Figure 3. It runs on a permission-based (private) blockchain network of independent domain nodes that form a shared metadata mesh. A (domain) metadata node can be seen as the most granular unit of catalog deployment. This unit is defined as a part of the data infrastructure code. In some sense, metadata is equivalent to the special form of a data product. In our running example, a Studio domain can have multiple data nodes where each node can serve one or more polyglot data products. The Studio team can run multiple data nodes and metadata nodes simultaneously.

When a new product is released or updated, its metadata has to be registered in the catalog. Upon a data node start-up, a special middleware can verify whether the hosted product metadata is present and run the corresponding smart contract if necessary. If a new record is added, it will be replicated in the network following the distributed consensus algorithm.

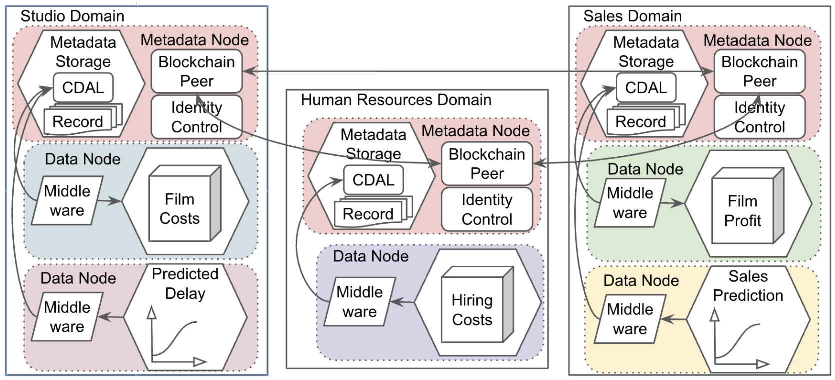

Fig. 3. Blockchain-based Type II metadata catalog running example overview.

Blockchain Peer is an application abstraction responsible for peer-to-peer communication, data synchronization, and consensus. Identity Control abstract defines the participants, roles, and permissions used to manage access to the chain data. It might be useful when access to the published metadata has to be limited, for instance, inter-organization access would be more restrictive rather than intra-organization. Metadata Storage is the medium of storage, the actual blockchain, replicated across all the network nodes. It has a Chain Data Access Layer (CDAL) that is responsible for writing and reading the data to the chain. It is based on a smart contract code. Records in the chain represent the data products metadata: product name, location, owner, format, access flow, etc.

Challenges of Using Blockchain. Although the blockchain unlocks a number of benefits, it comes with challenges. The append-only ledger keeps the whole history of each metadata record. Therefore, it raises two obvious concerns: querying performance and storage capacity.

First, in a catalog with a lot of similar or nearly identical records, it can result in bad querying performance since the system has to process a larger amount of data compared to the case of storing only the latest record versions.

Second, big number of data products and the duplication of metadata records can result in the exponential growth of the catalog size. It may become especially problematic when a new node joins the mesh and has to download all records.

Some potential solutions for these issues are to use the pruned nodes for reducing the storage by keeping the latest ledger segments and to build the "world state" database, as in Hyperledger Fabric, for improving querying performance.

5 Conclusions and Further Research

In our proposal, we demonstrated the main elements of data mesh architecture and distributed metadata catalog. We explored the theoretical benefits of doing the distribution through a private permission-based blockchain network.

Our further work will focus on implementing the proposed Type II blockchain-based architecture and its comparison to other solutions for the outlined challenges. Particularly, we will investigate a construction process of a Type III metadata catalog on top of property graph technologies by estimating the discoverability and querying issues caused by partitioning the underlying graph.

References

1. Casino, F., Dasaklis, T.K., Patsakis, C.: A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and informatics (2019)
2. Chessell, M., Jones, N.L., Limburn, J., Radley, D., Shan, K.: Designing and operating a data reservoir. In: IBM Redbooks (2015)
3. Dehghani, Z.: Data Mesh: Delivering Data-Driven Value at Scale. O'Reilly (2022)
4. Demichev, A., Kryukov, A., Prikhodko, N.: The approach to managing provenance metadata and data access rights in distributed storage using the hyperledger blockchain platform. In: Ivannikov Ispras Open Conference. IEEE (2018)
5. García-Barriocanal, E., Sánchez-Alonso, S., Sicilia, M.A.: Deploying metadata on blockchain technologies. In: Metadata and Semantic Research: 11th International Conference, Proceedings 11. Springer (2017)
6. Hai, R., Geisler, S., Quix, C.: Constance: An intelligent data lake system. In: Proceedings of the 2016 international conference on management of data (2016)
7. Halevy, A.Y., Korn, F., Noy, N.F., Olston, C., Polyzotis, N., Roy, S., Whang, S.E.: Managing google's data lake: an overview of the goods system. IEEE Data Eng. Bull. **39**(3), 5–14 (2016)
8. Inmon, W., Strauss, D., Neushloss, G.: DW 2.0: The architecture for the next generation of data warehousing. Elsevier (2010)
9. Koscina, M., Manset, D., Negri-Ribalta, C., Perez, O.: Enabling trust in healthcare data exchange with a federated blockchain-based architecture. In: International Conference on Web Intelligence-Companion Volume (2019)
10. Kumar, D.S., Rahman, M.A.: Simplified hdfs architecture with blockchain distribution of metadata. International Journal of Applied Engineering Research (2017)
11. Laurent, A., Laurent, D., Madera, C.: Data Lakes, Chapter 1, Introduction to Data Lakes: Definitions and Discussions, vol. 2. John Wiley & Sons, Inc (2020)
12. Liu, L., Li, X., Au, M.H., Fan, Z., Meng, X.: Metadata privacy preservation for blockchain-based healthcare systems. In: Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part I. pp. 404–412. Springer (2022)
13. Majchrzak, J., Balnojan, S., Siwiak, M., Sieraczkiewicz, M.: Data Mesh in Action. Manning Publishing (2022)
14. Sawadogo, P., Darmont, J.: On data lake architectures and metadata management. Journal of Intelligent Information Systems **56**(1), 97–120 (2021)
15. Sawadogo, P., Darmont, J., Noûs, C.: Joint management and analysis of textual documents and tabular data within the audal data lake. In: European Conference on Advances in Databases and Information Systems. pp. 88–101. Springer (2021)
16. Zhao, Y.: Metadata Management for Data Lake Governance. Ph.D. thesis, Univ. Toulouse 1 (2021)