Application of Parallel and Distributed Systems
Institute for Parallel and Distributed Systems
University of Stuttgart

Unleashing the Value of Metadata: An Extensible Architecture for Holistic Metadata Management in Enterprises

Jan Schneider, Christoph Gröger, Arnold Lutsch,
Holger Schwarz and Bernhard Mitschang

In: Proceedings of the 28th International Conference on Enterprise Information Systems (ICEIS) – Volume 2, pp. 1222–1229, SciTePress, 2026.

BIBTEX:

```
@inproceedings{Schneider2026,
    author={Schneider, Jan and Gröger, Christoph and Lutsch, Arnold and Schwarz, Holger and
    Mitschang, Bernhard},
    title={Unleashing the Value of Metadata: An Extensible Architecture for Holistic Metadata
    Management in Enterprises},
    booktitle={Proceedings of the 28th International Conference on Enterprise Information Systems
    (ICEIS)},
    year={2026},
    pages={1222-1229},
    publisher={SciTePress},
    organization={INSTICC},
    doi={XXX},
    isbn={978-989-758-834-1},
    issn={2184-4992}
}
```

Unleashing the Value of Metadata: An Extensible Architecture for Holistic Metadata Management in Enterprises

Jan Schneider[1][a], Christoph Gröger[2][b], Arnold Lutsch[2], Holger Schwarz[1][c] and Bernhard Mitschang[1][d]

[1]*University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany*
{Firstname}.{Lastname}@ipvs.uni-stuttgart.de
[2]*Robert Bosch GmbH, Borsigstraße 4, 70469 Stuttgart, Germany*
{Firstname}.{Lastname}@de.bosch.com

Keywords: Metadata Management, Data Catalog, Data Platform, Knowledge Graph, Data Analytics.

Abstract: In the digital age, the analysis of data is crucial for enterprises of almost all industries. To support the collection, management, and analysis of data, modern data platforms such as data lakehouses have become prevalent in recent years. Despite these technological advances, metadata management in today's enterprise data architectures still shows serious shortcomings, as metadata is only inadequately captured and locked up in silos. As a result, the knowledge transported by this metadata cannot be fully exploited, which complicates and slows down all metadata-based applications. In this paper, we first investigate common technical challenges of metadata management in enterprise data architectures. Secondly, we derive requirements that enterprise data architectures must meet to address these challenges. By considering these requirements, we then present a conceptual architecture as solution, which utilizes a knowledge graph and a microservice architecture to enable holistic management of metadata and to provide it to different applications in a flexible manner. Finally, we evaluate and discuss this architecture with the help of an early prototype.

1 INTRODUCTION

With the help of different analysis techniques, enterprises can derive insights and knowledge from data to support the optimization of business processes and products. *Data platforms*, such as traditional *data warehouses* (Inmon, 2005) and the more flexible *data lakes* (Hai et al., 2023), provide the technical foundation for the collection, processing and management of data. In industrial practice, the scope of data platforms and their architectures are becoming increasingly complex, as the amount of data is continuously growing, while more data sources, analytics applications, technologies and user groups get involved (Greco et al., 2025; Bianchini et al., 2024).

Besides the individual data platforms, also the complexity of the overarching *enterprise data architectures* is increasing, since organizational units in an enterprise operate their data architectures independently of each other. Supported by the lack of enterprise-wide guidelines (Gröger, 2021), this leads to an organic and uncontrolled growth.

To deal with this complexity and to allow the exploitation of the data's business value, extensive metadata management is necessary (Sawadogo and Darmont, 2021). For this purpose, various *metadata platforms* are utilized, like data marketplaces (Eichler et al., 2022) and data catalogs (Zaidi et al., 2017).

However, there are still serious shortcomings for metadata management in enterprises, as metadata is only inadequately captured and tracked and typically spread across several metadata platforms, which become silos and thus hinder the integration and flexible provisioning of metadata for various applications. As a result, no holistic metadata management can be established and the metadata not be fully exploited.

With the goal of addressing these challenges, this paper a) investigates the current state of metadata management in enterprises and its challenges (cf. Section 2), derives requirements that enterprise data architectures need to fulfill to address these challenges (cf. Section 3), proposes our Knowledge-graph-driven Enterprise Data Architecture (KEDA) as solution (cf. Section 3) and evaluates this architecture with the help of an early prototype (cf. Section 4).

[a] https://orcid.org/0009-0008-9740-9203
[b] https://orcid.org/0000-0001-6615-4772
[c] https://orcid.org/0000-0001-7085-2813
[d] https://orcid.org/0000-0003-0809-9159

2 CURRENT STATE AND TECHNICAL CHALLENGES OF METADATA MANAGEMENT

This section discusses the current state of metadata management in enterprise data architectures, based on a) long-term experiences from industrial practice, b) insights collected in four case studies (Schneider et al., 2025) and c) input from a literature review by applying the snowballing technique (Wohlin, 2014).

2.1 Enterprise Data Architectures

Enterprises generate many different types of data, such as IoT data and user-generated documents. This data is stored across different operational source systems, such as Manufacturing Execution Systems (MES) and Document Management Systems (DMS). To consolidate the data and to be able to perform complex analyses without impacting operational workloads, the data needs to transferred to data platforms. There, the data is then integrated, harmonized, processed, and analyzed with various techniques, such as reporting, Online Analytical Processing (OLAP), data mining, machine learning and streaming analytics.

Data warehouses constitute the most traditional type of data platforms. They are primarily designed for OLAP and reporting and their rigid data models render them less suitable for exploratory analyses such as data mining. To address this shortcoming, data lakes have emerged over the past decade, where unprocessed raw data can be ingested in arbitrary formats and then gradually transformed and prepared for analysis. Additionally, there have been efforts to combine advantages of data warehouses and data lakes into so-called data lakehouses.

This variety of data platforms already poses great diversity and complexity in enterprise data architectures, as data is often stored redundantly in different processing stages on several data platforms. Furthermore, enterprises usually consist of multiple, independent organizational units, which have different responsibilities and thus operate their own data architecture. This leads to a highly interconnected and heterogeneous enterprise data architecture (Gröger, 2021), which is illustrated in Fig. 1: Here, data from the source systems is ingested into different data platforms, which are operated by various organizational units. The data is then exchanged between the data platforms and exploited for analytical use cases. This process involves various different types of users as well as automated software agents.

Metadata platforms, such as data catalogs and enterprise data marketplaces, do not store actual data, but instead allow to manage different types of metadata. However, metadata platforms are typically operated by their organizational units and limited to specific types of metadata and applications (Kropshofer et al., 2025), which is why several metadata platforms exist within an enterprise (Eichler et al., 2022).

In summary, it can be observed that enterprise data architectures typically bear a high degree of decentralization, heterogeneity and high complexity.

2.2 Technical Challenges of Metadata Management in Enterprises

Despite the existence of numerous offerings for metadata platforms, significant shortcomings regarding metadata management can still be observed in enterprises. In this paper, we focus on technical challenges.

2.2.1 C1: Inflexible Metadata Modeling

In enterprises, typically only few types of metadata are actually captured and stored in corresponding metadata platforms. This usually includes static information that remains stable over longer periods of time, such as the storage locations of data assets and details about responsible data owners (Tonnarelli et al., 2025). Other information, such as details about how data assets have been processed and analyzed in the past, is often not captured at all. However, this knowledge can still support data scientists in finding relevant datasets (Gunklach et al., 2023). A root cause for this shortcoming is that many metadata platforms rely on static modeling techniques with limited expressiveness, such as tags and table-like data structures. In contrast, graphs and ontologies, which represent the most expressive modeling techniques (Feilmayr and Wöß, 2016), are barely utilized (Kropshofer et al., 2025). Furthermore, even graph-based metadata platforms, such as COLID[1], Zeenea Data Catalog[2] and data.world[3], define most types of metadata that can be modeled and cannot be arbitrarily customized. Since a high proportion of the available data catalogs represent proprietary offerings (Kropshofer et al., 2025; Tonnarelli et al., 2025), the flexibility for metadata modeling is further limited.

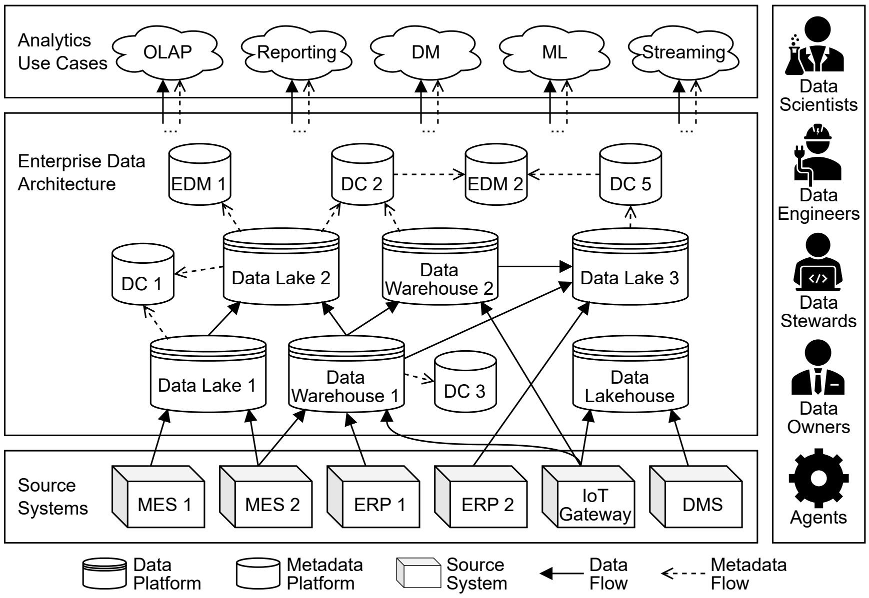
2.2.2 C2: Distributed Metadata Management

As discussed in Section 2.1, metadata in enterprises is typically spread across different metadata platforms,

[1] https://github.com/Bayer-Group/COLID-Documentation

[2] https://zeenea.com/data-catalog/

[3] https://data.world



DM: Data Mining; ML: Machine Learning; DC: Data Catalog; EDM: Enterprise Data Marketplace;
MES: Machine Execution System; ERP: Enterprise Resource Planning; DMS: Document Management System

Figure 1: Example of a typical enterprise data architecture, modified from (Gröger, 2021).

which are possible maintained by various organizational units. Inevitably, this leads to a decentralized silo landscape, which makes it difficult for users and agents to find relevant metadata (Jahnke and Otto, 2023), as they do not have a single point of access and likely do not even know which metadata platforms exist within the enterprise. Furthermore, the metadata silos prevent the integration of metadata, as related metadata from different metadata platforms cannot simply be connected. As a result, for example, data scientists cannot immediately tell whether two columns in different data assets refer to the same real-world concept. Moreover, changes to the enterprise data architecture or policies, such as increasing requirements with regard to privacy, must be reflected in all metadata platforms. Existing approaches that pursue to centralize metadata, such as Enterprise Data Marketplaces (Eichler et al., 2022), are limited to specific types of metadata and applications.

2.2.3 C3: Heterogeneity of Metadata Platforms

As metadata platforms are possibly operated by different organizational units, these platforms typically differ greatly in terms of their purpose, domain-specificity, functional scope, interfaces for accessing metadata, data formats and the types of metadata for which they are designed (Kropshofer et al., 2025). In addition, many data catalogs are tightly coupled to a specific ecosystem, such as AWS Glue Data Catalog[4], Snowflake Horizon[5], Unity Data Catalog[6], and hence vary in their support for different technology stacks. This heterogeneity and the resulting lack of interoperability between different metadata platforms complicates the uniform use of metadata and constitutes one reason why AI agents have not become widely established yet (Yang et al., 2025).

3 SOLUTION APPROACH

In this section, our solution approach is developed. Section 3.1 first derives several requirements for enterprise data architectures from the technical challenges, while Section 3.2 proposes KEDA as solution.

3.1 Requirements

Based on the challenges from Section 2.2, we derive six requirements for enterprise data architectures:

R1: Single Access Point for Metadata To address challenge C2, a single, central access point for

[4]https://docs.aws.amazon.com/glue/latest/dg/catalog-and-crawler.html

[5]https://www.snowflake.com/en/product/features/horizon/

[6]https://www.databricks.com/product/unity-catalog

metadata should be established within enterprise data architectures, so that users and agents know where to look for metadata. This access point does not necessarily have to manage the metadata itself, but must at least refer to locations where the required metadata can be found.

R2: Uniform Metadata Access From challenge C3, it can be concluded that metadata should be accessible in an uniform manner, i.e., via uniform interfaces and in uniform formats. This means, in particular, that the same instances of metadata should always be written, updated, deleted and queried in the same way, regardless of where they are stored.

R3: Integrated Metadata View To further address the issues of challenge C2, an integrated view on the available metadata must be created. This means that it must be possible to link different instances of metadata to each other if they are related, regardless of their type and origin.

R4: Flexible Metadata Modeling In accordance with challenge C1, flexible and expressive modeling techniques must be used to adequately capture and describe any type of metadata, as well as relationships between them.

R5: High extensibility Due to the challenges C2 and C3, enterprise data architectures severely limit the extensibility for new types of metadata and metadata-based applications, which make it necessary to consider and adjust multiple metadata platforms individually. Hence, an improved architecture should inherently provide high extensibility and flexibility.

R6: Gradual migration Since enterprises already employ enterprise data architectures that may have have grown organically, it must be possible to gradually migrate these architectures into an improved one in a non-disruptive way.

3.2 KEDA: Knowledge-graph-driven Enterprise Data Architecture

To address the problems of metadata management, we propose the Knowledge-graph-driven Enterprise Data Architecture (KEDA), which is shown in Fig. 2.

In KEDA, the actual metadata is managed in the *Metadata Knowledge Graph*, which consolidates all metadata of interest that is generated within the enterprise. Knowledge graphs are graphs that reflect knowledge of the real world, where nodes represent relevant entities and edges relationships between these entities (Hogan et al., 2022). This modeling technique provides a high degree of expressiveness (Feilmayr and Wöß, 2016) and flexibility for integrating data (Simsek et al., 2023). Consequently, knowledge graphs are particularly suitable for consolidating metadata from different sources into an integrated view (Kropshofer et al., 2025; Dibowski et al., 2020) and thus contribute to the fulfillment of the requirements R3, R4 and R5. The metadata knowledge graph serves as a central point of access where all metadata can be found that may be relevant to users or agents in the scope of metadata-based applications, such as data exploration, data quality control and maintenance. In contrast, technical metadata like transaction logs (Armbrust et al., 2020), which are solely required for the internal operation of individual data platforms and not of interest to users and agents, are not added to the knowledge graph.

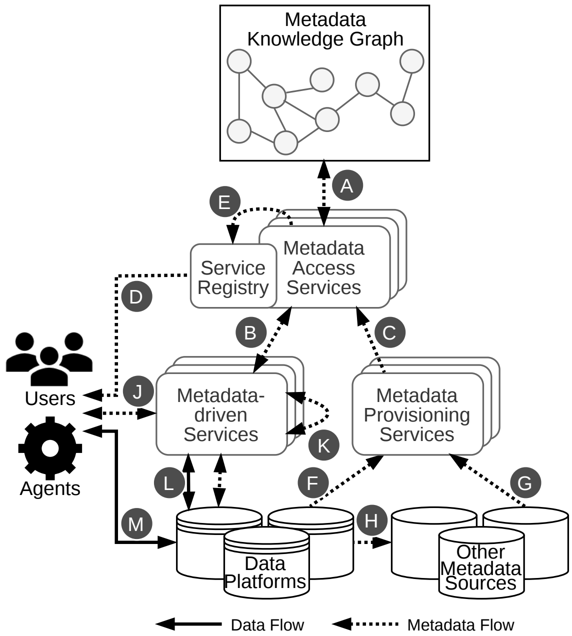


Figure 2: Conceptual architecture KEDA for metadata management in enterprises.

To enable uniform access to metadata and high expandability for arbitrary types of metadata and metadata-based applications, most components in the enterprise data architecture should not interact directly with the metadata knowledge graph. Instead, they invoke metadata services for this purpose, which encapsulate the access to metadata. To avoid redundancy in the implementation, metadata services can also invoke each other, forming a flexible and extensible microservice architecture (Newman, 2021). There are three different types of metadata services in KEDA: *Metadata Access Services* encapsulate the read and write access to the metadata knowledge graph (Ⓐ) by providing interfaces that allow other services to read and edit specific types of metadata.

For example, one metadata access service may be responsible for data lineage information and thus provides corresponding interfaces for retrieving, adding, updating and removing data lineage information in the metadata knowledge graph. This ensures that all other components that work with data lineage do not need to know how this information is modeled, mapped and queried in the knowledge graph, but can simply use the interfaces provided by the metadata access service (Ⓑ, Ⓒ). A special instance of a metadata access service, the *Service Registry*, serves as a central point of access for users, agents and other services for discovering and utilizing relevant services (Ⓓ). The service registry stores the directory of available metadata services in the metadata knowledge graph (Ⓐ). To populate the directory, all metadata services first have to register themselves (Ⓑ, Ⓒ, Ⓔ). This approach resembles the service registry pattern (Taibi et al., 2018), which is applied to support the dynamic discovery of services and to achieve loose coupling and load-balancing.

The *Metadata Provisioning Services* are responsible for capturing and extracting metadata from sources that regularly generate relevant metadata. These sources can be data platforms (Ⓕ), but also other sources of metadata (Ⓖ), such as already existing metadata platforms, which can be coupled with data platforms (Ⓗ). The metadata provisioning services invoke metadata access services in order to update the metadata in the metadata knowledge graph.

Finally, the *Metadata-driven Services* exploit metadata in order to provide users and agents with valuable application logic that support them in metadata-based applications (Ⓙ). For example, metadata-driven services can assist data scientists in exploring data assets via a suitable user interface and help data engineers to maintain data pipelines. Metadata-driven services can invoke themselves (Ⓚ) to implement functions of higher order and can have access to data and metadata of data platforms (Ⓛ). However, for reasons of scalability, users and agents should still access data on data platforms directly (Ⓜ) rather than via metadata services.

To ensure extensibility and flexibility, loose coupling between all metadata services must be established. This can be achieved by leveraging the service registry and using a message bus for the communication between the services (Hohpe and Woolf, 2004).

4 PROTOTYPE AND ASSESSMENT

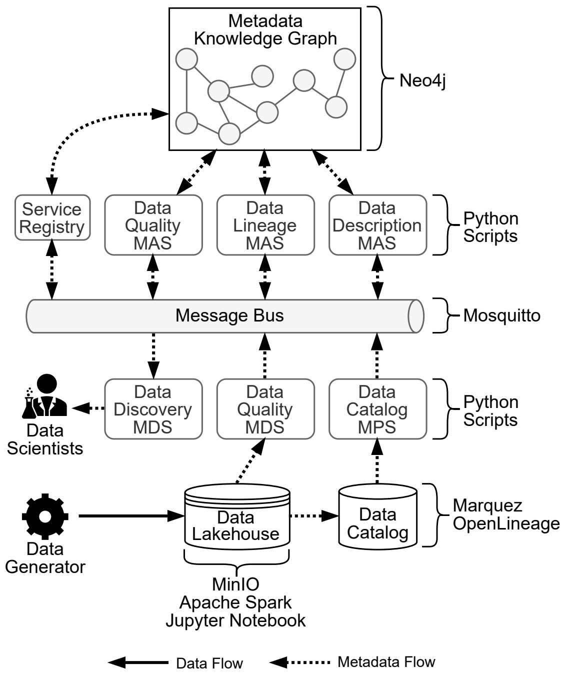
To evaluate KEDA, a prototype was developed that serves as a proof of concept. Section 4.1 first presents the prototype and its architecture, while Section 4.2 discusses how KEDA fulfills the requirements as derived in Section 3.1.

4.1 Prototype

The prototype relies on Docker[7] for the execution of the individual components in containers and facilitating the communication between them. Its technology architecture is depicted in Fig. 3. For the metadata knowledge graph, Neo4j[8] is used as graph database. In the scope of the prototype, it is supposed to manage three different types of metadata: a) descriptive information about data assets, which include technical metadata, b) data lineage information and c) data quality measurements. For each type of metadata, a Metadata Access Service (MAS) is available, consisting of a Python script that wraps the read and write access to this type of metadata in the metadata knowledge graph with a dedicated API. Additionally, another metadata access service is deployed, which acts as the service registry. All metadata services in the architecture can communicate with each other via a shared message bus that is provided by an instance of Mosquitto[9], a lightweight MQTT broker. The prototype includes a data lakehouse as data platform, which consists of MinIO[10] as storage system, an Apache Spark[11] cluster as processing engine and a Jupyter Notebook[12] for implementing data pipelines and analytics applications. An agent allows the generation of new data assets and their ingestion into the data lakehouse. The data is stored in the Delta Lake[13] format. With the goal of taking already existing data catalogs from the enterprise data architecture into account, the prototype also includes an instance of Marquez[14]. This data catalog is intertwined with the data lakehouse by utilizing OpenLineage[15] within Apache Spark. This way, technical metadata and data lineage information can be automatically captured and

[7] `https://www.docker.com`
[8] `https://neo4j.com`
[9] `https://mosquitto.org`
[10] `https://www.min.io`
[11] `https://spark.apache.org`
[12] `https://jupyter.org`
[13] `https://delta.io`
[14] `https://marquezproject.ai`
[15] `https://openlineage.io`

stored in the data catalog. A dedicated Metadata Provisioning Service (MPS) is responsible for extracting the descriptive information about the data assets and their data lineage from the data catalog and notifies the corresponding metadata access services about changes via the message bus. The data quality measurements are carried out by a dedicated Metadata-driven Service (MDS), which analyzes the data assets in the data lakehouse with respect to their completeness. For this purpose, it periodically requests technical metadata about the available data assets from the metadata knowledge graph via the message bus, locates the data assets on the data lakehouse and then measures the percentage of null values in each column of the associated tables. The results are subsequently stored in the metadata knowledge graph by utilizing the corresponding metadata access service. Another metadata-driven service provides data scientists with an overview of the data assets that are available on the data lakehouse, descriptive information about them, their data lineage and the corresponding data quality measurements via a web-based user interface. For this, it requests the necessary information periodically from the metadata knowledge graph via the three metadata access services.



MAS: Metadata Access Service
MDS: Metadata-driven Service
MPS: Metadata Provisioning Service

Figure 3: Architecture of the KEDA prototype.

The presented prototype is functionally complete and provides data scientists with different types of metadata about data assets that are managed in a data lakehouse and an associated data catalog. Hence, the prototype demonstrates as proof of concept that KEDA can be used as conceptual architecture for holistic metadata management.

4.2 Assessment

In the following, it is discussed whether KEDA fulfills the requirements for enterprise data architectures as derived in Section 3.1.

R1 is fulfilled by the microservice architecture and the Service Registry, since they provide a single access point that allows users and agents to find services for metadata-based applications. All they need to know is where to find the service registry; besides that, no further knowledge about the data architecture or the metadata knowledge graph is required.

Similarly, *R2* is fulfilled by the microservice architecture and the metadata access services. By encapsulating the query logic for retrieving, inserting, updating and deleting metadata in the metadata knowledge graph, they hide the graph-specific modeling details. This way, they can provide uniform interfaces to the other metadata services, while ensuring data integrity within the metadata knowledge graph.

In terms of *R3*, the metadata knowledge graph logically consolidates all metadata of interest from the enterprise data architecture at a central point. Hence, the metadata access services can establish connections between instances of metadata, contributing to an integrated, interconnected view of the metadata.

By using a knowledge graph to manage the metadata, also *R4* can be fulfilled, since knowledge graphs represent an expressive data modeling technique and provide a higher degree of flexibility compared to other storage systems, such as relational databases.

The architecture also fulfills *R5*, since the microservice architecture provides loose coupling and only requires little effort to add new metadata services. With the help of new metadata access services and metadata provisioning services, different types of metadata can be added to the knowledge graph, while new metadata-driven services allow to utilize the available metadata in different ways.

Regarding *R6*, the architecture provides several benefits that facilitate the migration of existing enterprise data architectures: In general, the metadata knowledge graph and the microservice architecture can be established in parallel to an already existing enterprise data architecture, where metadata is replicated to the knowledge graph with the help of metadata access services and metadata provisioning services. Existing components can either take the role of “Other Metadata Sources” or be wrapped as metadata-driven services. Step by step, new metadata-driven services can then be developed and existing applications gradually be migrated to KEDA.

Finally, it can be concluded that KEDA fulfills the six requirements for enterprise data architectures.

5 RELATED WORK

In literature, several related proposals can be found for improving metadata management with the help of semantic approaches: (Dibowski et al., 2020) refer to a scenario from the automotive sector in order to demonstrate how knowledge graphs can be used to manage metadata for a data lake. They propose a semantic layer consisting of a knowledge graph and a corresponding ontology, as well as a data lake architecture that utilizes the semantic layer. However, their approach solely focuses on semantic search for data assets, data provenance and access control and does not consider further metadata-based applications. Furthermore, they only discuss metadata management for a single data lake and their architecture does not generalize for holistic metadata management. Similarly, this also applies to the work of (Kasrin et al., 2018), who propose a framework for semantic data management systems and a corresponding implementation that allows to link data from a data lake to semantic metadata. While their architecture likewise employ components called ”Metadata Services”, these are limited to extracting technical metadata from documents in the data lake and storing it in a knowledge graph, whereas metadata services in KEDA perform a much wider range of tasks to support any metadata-based applications. (Schrott et al., 2023) present an ontology layer that allows to link metadata, which is collected in an existing data catalog, to the semantics of data. This way, a domain-specific description of data sets, schemas, tables, and attributes is created that is also machine-readable. However, this ontology layer is limited to connecting technical metadata of data assets with semantical information and not designed for holistic metadata management. (Bianchini et al., 2024) suggest a system for personalized data exploration on a data lake, which leverages a semantic metadata catalog and a corresponding ontology for this purpose. However, this concept is limited to data exploration on a single data lake with a specific architecture and does not provide extension points for other metadata-based applications or data platforms. Other works that leverage semantic metadata management are primarily limited to searching and querying data with the help of knowledge graphs (Hoseini et al., 2024; Hoseini et al., 2023; Diamantini et al., 2022; Schmid et al., 2019; Mami et al., 2019; Beheshti et al., 2018), such as by applying ontology-based data access (Xiao et al., 2018), but again do neither generalize to arbitrary metadata-based applications, nor to the level of complex enterprise data architectures.

In summary, we are not aware of any other comparable architectural approaches or technologies that provide company-wide, holistic and extendable metadata management for metadata-based applications.

6 CONCLUSION

Metadata management still poses severe challenges in enterprises. By leveraging a knowledge graph as single point of access for metadata and a microservice architecture, KEDA provides a flexible and extensible approach for capturing, managing and exploiting various types of metadata. In addition, KEDA allows the gradual migration of existing data architectures.

For the metadata knowledge graph, different implementation options exist, whose advantages and disadvantages have yet to be compared. This includes decisions regarding whether triple-based or property graphs should be used, whether a centralized or federated graph should be leveraged and how the metadata in the knowledge graph should actually be modeled. Finally, the overhead introduced by KEDA has to be evaluated and assessed. We plan to investigate these open questions as part of future work.

REFERENCES

Armbrust, M., Das, T., Sun, L., et al. (2020). Delta Lake: High Performance ACID Table Storage over Cloud Object Stores. *Proceedings of the VLDB Endowment*, 13(12):3411–3424.

Beheshti, A., Benatallah, B., Nouri, R., and Tabebordbar, A. (2018). CoreKG. *Proceedings of the VLDB Endowment*, 11(12):1942–1945.

Bianchini, D., De Antonellis, V., and Garda, M. (2024). A semantics-enabled approach for personalised Data Lake exploration. *Knowledge and Information Systems*, 66(2):1469–1502.

Diamantini, C., Potena, D., and Storti, E. (2022). A Knowledge-Based Approach to Support Analytic Query Answering in Semantic Data Lakes. In *Advances in Databases and Information Systems*, volume 13389 of *Lecture Notes in Computer Science*, pages 179–192. Springer International Publishing.

Dibowski, H., Schmid, S., Svetashova, Y., Henson, C., and Tran, T. (2020). Using Semantic Technologies to Manage a Data Lake: Data Catalog, Provenance and

Access Control. *13th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2020)*.

Eichler, R., Gröger, C., Hoos, E., et al. (2022). Data Shopping — How an Enterprise Data Marketplace Supports Data Democratization in Companies. In *Intelligent Information Systems*, volume 452 of *Lecture Notes in Business Information Processing*, pages 19–26. Springer International Publishing.

Feilmayr, C. and Wöß, W. (2016). An analysis of ontologies and their success factors for application to business. *Data & Knowledge Engineering*, 101:1–23.

Greco, D., Osborne, F., Pusceddu, S., et al. (2025). Modelling big data platforms as knowledge graphs: the data platform shaper. *Journal of Big Data*, 12(1).

Gröger, C. (2021). There is no AI without data. *Communications of the ACM*, 64(11):98–108.

Gunklach, J., Michalczyk, S., Nadj, M., and Maedche, A. (2023). Metadata Extraction from User Queries for Self-Service Data Lake Exploration. *Datenbank-Spektrum*, 23(2):97–105.

Hai, R., Koutras, C., Quix, C., and Jarke, M. (2023). Data Lakes: A Survey of Functions and Systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12571–12590.

Hogan, A., Blomqvist, E., Cochez, M., et al. (2022). Knowledge Graphs. *ACM Computing Surveys*, 54(4):1–37.

Hohpe, G. and Woolf, B. (2004). *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. The Addison-Wesley signature series. Addison-Wesley, Boston, 4. printing edition.

Hoseini, S., Ali, A., Shaker, H., and Quix, C. (2023). SEDAR: A Semantic Data Reservoir for Heterogeneous Datasets. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5056–5060, New York, NY, USA. ACM.

Hoseini, S., Theissen-Lipp, J., and Quix, C. (2024). A survey on semantic data management as intersection of ontology-based data access, semantic modeling and data lakes. *Journal of Web Semantics*, 81:100819.

Inmon, W. H. (2005). *Building the Data Warehouse*. Wiley technology publishing Timely, practical, reliable. Wiley, Indianapolis, Ind., 4th edition edition.

Jahnke, N. and Otto, B. (2023). Data Catalogs in the Enterprise: Applications and Integration. *Datenbank-Spektrum*, 23(2):89–96.

Kasrin, N., Qureshi, M., Steuer, S., and Nicklas, D. (2018). Semantic Data Management for Experimental Manufacturing Technologies. *Datenbank-Spektrum*, 18(1):27–37.

Kropshofer, J., Schrott, J., Wöß, W., and Ehrlinger, L. (2025). A Survey on the Functionalities of Data Catalog Tools. *IEEE Access*, 13:83297–83319.

Mami, M. N., Graux, D., and Scerri, S. e. a. (2019). Squerall: Virtual Ontology-Based Access to Heterogeneous and Large Data Sources. In *The Semantic Web – ISWC 2019*, volume 11779 of *Lecture Notes in Computer Science*, pages 229–245. Springer International Publishing.

Newman, S. (2021). *Building Microservices: Designing Fine-grained Systems*. O'Reilly, Beijing and Boston and Farnham and Sebastopol and Tokyo, second edition edition.

Sawadogo, P. and Darmont, J. (2021). On data lake architectures and metadata management. *Journal of Intelligent Information Systems*, 56(1):97–120.

Schmid, S., Henson, C., and Tran, T. (2019). Using Knowledge Graphs to Search an Enterprise Data Lake. In *The Semantic Web: ESWC 2019 Satellite Events*, volume 11762 of *Lecture Notes in Computer Science*, pages 262–266. Springer International Publishing.

Schneider, J., Gröger, C., Lutsch, A., et al. (2025). Architectures and Implementations of Data Lakehouses: Case Studies from Industrial Practice. In *Business Information Systems*, volume 554 of *Lecture Notes in Business Information Processing*, pages 241–255. Springer Nature Switzerland.

Schrott, J., Weidinger, S., Tiefengrabner, M., et al. (2023). GOLDCASE: A Generic Ontology Layer for Data Catalog Semantics. In *Metadata and Semantic Research*, volume 1789 of *Communications in Computer and Information Science*, pages 26–38. Springer Nature Switzerland.

Simsek, U., Kärle, E., Angele, K., Huaman, E., Opdenplatz, J., Sommer, D., Umbrich, J., and Fensel, D. (2023). A Knowledge Graph Perspective on Knowledge Engineering. *SN Computer Science*, 4(1).

Taibi, D., Lenarduzzi, V., and Pahl, C. (2018). Architectural Patterns for Microservices: A Systematic Mapping Study. In *Proceedings of the 8th International Conference on Cloud Computing and Services Science*, pages 221–232. SCITEPRESS - Science and Technology Publications.

Tonnarelli, M., Kumara, I., Driessen, S., et al. (2025). Data catalog tools: A systematic multivocal literature review. *Journal of Systems and Software*, 230:112584.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, pages 1–10. ACM.

Xiao, G., Calvanese, D., Kontchakov, R., et al. (2018). Ontology-Based Data Access: A Survey. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 5511–5519, California. International Joint Conferences on Artificial Intelligence Organization.

Yang, W., Fu, R., Amin, M. B., and Kang, B. (2025). The Impact of Modern AI in Metadata Management. *Human-Centric Intelligent Systems*, 5(3):323–350.

Zaidi, E., de Simoni, G., Edjlali, R., and Duncan, A. D. (2017). Data Catalogs are the New Black in Data Management and Analytics. *Gartner, Consultancy Report*.