Modification of the Algorithm for Dynamic Data Transformation Based on Blockchain Technology for Data Management Systems

Nikita Olimpiev[1(✉)], Alexander Vodyaho[2], and Nataly Zhukova[3]

[1] ITMO University, Kronverksky Pr. 49, bldg. A, 197101 St. Petersburg, Russia
olimpiev@itmo.ru

[2] Saint Petersburg Electrotechnical University, Professora Popova St. 5, 197022 St. Petersburg, Russia

[3] Laboratory of Big Data Technologies in Socio-Cyberphysical Systems, Saint-Petersburg Federal Research Centre of the Russian Academy of Sciences, 14-th Linia, VI, No. 39, 199178 St. Petersburg, Russia

Abstract. The rapid growth of data and increasing complexity in various domains necessitate advanced data management solutions that address data quality, security, and interoperability challenges. This paper presents a modified algorithm for dynamic data transformation based on blockchain technology for Master Data Management Systems (MDM), incorporating Ethereum, IPFS and semantic data. Based on the proposed algorithm, an architectural approach is also described that allows developing semantic data management systems. The algorithm assumes data validation, enrichment, and deduplication processes, while also ensuring consistency and trust among multiple participants through the decentralized nature of the blockchain. The semantic data management aspect of the system is achieved using Apache Jena and local ontology, which facilitate semantic interoperability and efficient data processing. The proposed approach addresses the challenge of applying blockchain technology to semantic MDM systems and demonstrates an approach to using the system in Semantic Web, as well as in areas such as finance, healthcare, public administration, and logistics. We evaluated the algorithm using a customer information dataset from an e-commerce platform. The results confirmed the potential of blockchain technology in data management and showed improvements in data accuracy, consistency, completeness, but with some loss in storage efficiency due to the use of blockchain. This paper contributes to developing and evaluating a novel algorithm and architecture, combining blockchain technology, MDM principles, and semantic data management for improving data quality, security, and performance in decentralized systems. However, further research and assessment may be required for the full implementation and evaluation of the system.

Keywords: Blockchain Technology · Master Data Management · Ethereum · Semantic Data Management · Data Transformation · Decentralized Systems · Data Quality · Data Security

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14104, pp. 555–571, 2023.
https://doi.org/10.1007/978-3-031-37105-9_37

1 Introduction

1.1 Background and Motivation

According to Statista, a market and consumer data company, the global data volume reached 79 zettabytes in 2021, with forecasts of a surge to 181 zettabytes by 2025 [1]. This escalating trend prompts researchers to develop and adapt data management systems to satisfy growing demands, while large businesses invest in and deploy systems to optimize their operations. As data volumes expand, concerns related to data security, integrity, and system performance become increasingly crucial, particularly for organizations with multiple branches and teams handling local data.

The Data Management Association International (DAMA) characterizes data management as the capacity to plan, control, and deliver information assets [2]. Master Data Management (MDM) systems, designed to ensure data integrity and provide a unified, up-to-date representation of entities and their relationships in a consolidated form, are highly sought after by corporations dealing with distributed data. According to Gartner, the top 5 MDM market leaders in 2023 are PiLog MDRM, Intelligent Master Data Management Platform, Semarchy xDM, Stibo Systems MDM, and TIBCO EBX [3]. While highly rated, these MDM systems face challenges that could impact their functionality and efficiency. Current MDM systems often rely on a centrally stored tabular data structure, resulting in limitations that only partially fulfill growing organizations' needs for system performance, security, and data integrity.

Blockchain technology offers a potential solution to these issues by facilitating the development of distributed data management systems with robust security, data integrity, and reliability. As highlighted in the study [4], blockchain technology provides a paradigm shift in business process optimization, data exchange, and compatibility across related industries, as well as a novel approach to data management. The technology can be employed to ensure security of data and access when constructing data management systems. In another study on implementing blockchain in data management systems [5], the authors outline possible scenarios and advantages of utilizing this technology, indicating growing interest in the subject. To tackle the challenges of vulnerability, centralization, and scalability in data management, the researchers propose a blockchain-based data collaboration management structure featuring user authentication, data validation, load distribution across nodes, and the introduction of a digital currency called datacoin. However, the practical applicability of the work remains uncertain due to the lack of focus on specific mechanisms used by blockchain platforms, the management systems in operation, and the data employed in the study. Moreover, the authors of another study [6] delineate three levels of data management: blockchain architecture, blockchain data structure, and blockchain data storage mechanism. They conclude that standard blockchains are typically employed for digital currency, hybrid blockchains suit multi-organizational scenarios, and DAG-based blockchains are most appropriate for the Internet of Things. Despite these conclusions, the authors also underscore the shortcomings of hybrid blockchains and the nascent stage of DAG use. To manage excessive data loads, they recommend distributed data storage and encoding methods, as well as the creation of an additional database for queries.

Another significant toolset within data management and transformation is Semantic Web technologies, including RDF, OWL, SPARQL, and Linked Data. These technologies facilitate semantic data processing and offer a standardized methodology for data description and utilization, thereby promoting more efficient processing and application. However, managing semantic data presents complex challenges related to ensuring data integrity and up-to-date representation, which can be addressed using MDM systems. Administering ontology core data enables the realization of semantic technology benefits and allows the implementation of the MDM systems embedded within the Semantic Web concept.

As semantic data is often distributed and its collection involves working with heterogeneous sources, thus issues concerning data integrity and reliability arise. Simultaneously, blockchain technology enables the accumulation of heterogeneous data, ensuring data security and system scalability. For instance, the authors of the study [7] employ blockchain, Ethereum smart contracts, and the InterPlanetary File System (IPFS) network for decentralized data storage to manage ontology data. This stack circumvents the limitations and vulnerabilities of centralized data storage by distributing public ontology among users. While the study demonstrates the efficacy of combining blockchain technology with the Semantic Web for ontology data management, the authors acknowledge the time needed to modify a smart contract in the Ethereum network as a potential limiting factor. Additionally, the implemented ontology consistency checks are performed locally, making it possible for an attacker to bypass them. However, the proposed solution primarily serves as an ontology database manager, rather than a comprehensive data management system. Consequently, the development of an integrated MDM system for managing semantic data based on blockchain remains a pertinent task that warrants further research and development.

The paper is structured into three sections, each addressing a different aspect of the research. Section 1 is dedicated to the introduction, outlining the research issues, author motivation, and the study's goal. In Sect. 2, the basic and proposed modified algorithms for dynamic data transformation are discussed, and the architectural approach for the semantic data management system, based on the proposed algorithm, is presented. Subsequently, an evaluation of the proposed algorithm is provided. Section 3 focuses on the results and their corresponding discussion, elaborating on the research outcomes and the potential of blockchain technology in the realm of data management.

1.2 Research Goal

The analyses of literature sources reveals that prominent MDM systems and blockchain technology implementation methods for data management face certain limitations. There is a scarcity of research in open literature regarding the utilization of data management systems that facilitate interaction with semantic data using blockchain technology. Furthermore, previous studies have not comprehensively addressed the implementation of such a system for managing master data, highlighting the insufficient exploration of this topic, and hindering the assessment of the feasibility of employing Master Data Management systems in the context of the Semantic Web. Illustratively, the paper [8] provides a brief overview of employing blockchain technology for managing master data. The

authors highlight the importance for organizations to establish governance rules, processes, and methodologies to efficiently handle master data, thereby facilitating data consolidation across various blockchain networks in which they participate. However, it should be noted that the authors' work is limited to a short paper and does not delve into the implementation details of such a system. The study conducted by the authors [9] assessed the suitability of blockchain technology for storing and exchanging master data in supply chains. Although the application developed for this purpose did not specifically target semantic data, the findings of the study support previous theoretical claims in the literature. Empirical data from the study indicate that permissioned blockchain networks demonstrate enhanced scalability and cost-effectiveness compared to closed networks, including integration with Master Data Management systems. However, it should be noted that the study's results are limited in their applicability due to the lack of focus on semantic technologies and master data usage.

Also, in the context of the algorithms used to manage this data, it should be noted that existing research and industry practices in the field of data management use different approaches. As it turned out from the analysis, the existing academic research in this area is primarily focused on the use of blockchain technology to provide secure, decentralized, and transparent data management solutions [4, 5]. Some of these studies delve deeper into the combination of blockchain and semantic data to further enhance the capabilities of data management systems, offering more efficient and sophisticated ways to process, store, and retrieve information [7].

At the same time, the commercial business sector has witnessed the emergence and growth of Master Data Management (MDM) systems, which typically do not incorporate blockchain technology. These MDM systems are designed to handle vast amounts of structured and unstructured data, ensuring data quality, consistency, and accuracy across different domains and organizations. Instead of leveraging the unique advantages of blockchain and semantic data, MDM systems usually rely on traditional data management techniques, such as relational databases, and tabular data representations, which can limit their ability to adapt to the rapidly evolving data landscape. The main drawback of the listed underlying algorithms is the lack of integration of both blockchain technology and semantic data principles.

Utilizing blockchain and semantic technologies could potentially enhance data management systems, as the individual effectiveness of each technology is supported by the outcomes of reviewed works. However, integrating these technologies into a unified MDM system necessitates further analysis and justification of their implementation approach, drawing upon existing experience.

The goal of this research is to modify the dynamic data transformation algorithm that forms the foundation of a semantic data management system's architecture, and to devise an architectural approach for constructing a data management system utilizing the modified algorithm. This system will employ blockchain technology and conform to master data management (MDM) principles. By incorporating ontologies into MDM systems, the algorithm facilitates data definition, classification, and guarantees accuracy and consistency in the stored information. Additionally, ontologies can facilitate data search and navigation within the system. Blockchain provides data security and system scalability, while master data usage principles maintain the integrity and relevance of

semantic data in the system. While the use cases for blockchain for MDM and semantic technologies in organizations are clear, the use of the technology in organizations has yet worked out, which is a key reason for the lack of such systems.

2 Proposed Algorithm and Architectural Approach

2.1 Basic Algorithm of MDM Systems

The basic algorithm that uses only MDM with tabular data representation [10] can be described as follows:

1. Data extraction: Data is retrieved from various sources, such as databases, files, or APIs, in different formats (e.g., XML, JSON, CSV). The extraction process considers the specific data representation (tabular or semantic) and the storage method (centralized or decentralized).
2. Data validation: Before transformation, the algorithm validates the input data to ensure its accuracy and consistency. This step enforces predefined rules and conditions depending on the chosen data representation and storage method. However, it does not utilize the potential of semantic data technologies or the security benefits of blockchain technology.
3. Data transformation: The core of the basic algorithm is the data transformation process. This step converts the validated data into a common intermediate format, applying operations such as data cleansing, filtering, aggregation, and normalization. The transformation process is tailored to the specific data representation but may not take full advantage of the benefits that semantic data formats offer.
4. Data loading: The transformed and validated data is loaded into a target system, such as a data warehouse, database, or another application in the appropriate format and structure.
5. Access control and data retrieval: The basic algorithm employs access control mechanisms depending on the chosen data storage method (centralized or decentralized). Users can retrieve the transformed data, but the level of security and granularity of access control may be limited.

Thus, by systems using the basic algorithm of work, we mean MDM systems with a tabular form of data presentation, which serve to manage the master data of an organization. The purpose of such systems is to ensure that there are no repetitive, incomplete, conflicting data in various areas of the organization. Such systems can interact with external information systems, receive data from them and send back converted data. The basis for the implementation of such systems, as a rule, is an application server with a backend created by the vendor, a relational database and search indexes to speed up work. To sum up, basic MDM systems work centrally, and security is ensured only by the qualifications of the software developers who created the system.

2.2 Development of a Modified Algorithm

The basic sequence of actions can be transformed considering the principles of Master Data Management and the use of blockchain technology for a semantic management

system. The modified algorithm assumes combining the principles of Master Data Management (MDM) and the Ethereum 2.0 blockchain technology. Ethereum blockchain was chosen as a platform for the system being developed as it provides a powerful foundation for building decentralized applications, including data management systems, and the transition to Ethereum 2.0 with PoS further enhances its capabilities. Ethereum's transition to Ethereum 2.0, incorporating PoS and sharding, aims to address scalability challenges. Sharding allows the network to process transactions and computations in parallel, increasing the system's overall capacity and throughput. This scalability advantage can better accommodate data-intensive applications and enable efficient processing of large volumes of data. Ethereum has a proven track record and has undergone rigorous testing and audits. The network's size and decentralization contribute to its security, making it more resistant to attacks and manipulation. The transition to PoS further enhances security by aligning the economic incentives of validators with the network's integrity. PoS has the potential to offer higher scalability compared to PoW. Sharding allows the network to process multiple transactions and computations in parallel, enabling higher throughput and accommodating more users and data within the data management system. Also, PoS reduces energy consumption, resulting in lower operational costs compared to PoW. This cost efficiency can be advantageous for data management systems, as it reduces the barriers to entry and lowers transaction fees for users interacting with the system.

For algorithm implementation InterPlanetary File System (IPFS), message queues, Apache Jena, and local ontology are used that ensure high performance, data quality, and semantic interoperability. Given that the primary intention behind the development of blockchain technology did not involve the storage of large-scale data, the direct inclusion of such data on a blockchain introduces challenges associated with storage capacity, bandwidth limitations, and transaction processing capabilities. To address this issue, a viable solution entails storing these records in an external offline storage system, while retaining their location and a digital hash of the data on the blockchain for subsequent verification purposes. In consideration of prior research and literature, the recommended approach for this storage methodology is the utilization of the IPFS. IPFS provides a decentralized and distributed file storage system, which aligns well with the decentralized nature of blockchain technology. It allows data to be stored and retrieved in a peer-to-peer manner, ensuring data availability and resilience. Also, IPFS uses content-based addressing, where data is identified by its unique cryptographic hash. IPFS ensures data integrity through content addressing and immutability. Each piece of data is associated with a unique hash, making it tamper-proof and verifiable. This approach enables efficient and secure retrieval of data based on its content, making it suitable for storing and retrieving semantic data. In the context of embedding a blockchain-based data management system into the Semantic Web ecosystem, compatibility is important. Ethereum's compatibility with widely adopted standards such as ERC-20 and ERC-721 fosters interoperability with other applications and platforms. This allows data management system built on Ethereum to seamlessly interact with other decentralized applications and exchange data with different protocols. It provides a standardized framework for integration and enhances the system's flexibility. Message queues also facilitate asynchronous communication between different components of a system. In a blockchain-based semantic data

management system, message queues be used to handle data exchange, notifications, and event-driven interactions between smart contracts, off-chain components, and external systems. Apache Jena, as an open-source framework for building semantic web applications, provides robust tools and libraries for working with RDF data, ontologies, and SPARQL for manipulating semantic data. Apache Jena was chosen because it can be integrated with blockchain technology to enable efficient management and querying of semantic data stored on the blockchain. It provides capabilities for reasoning, inference, and semantic data integration, which are essential for leveraging the full potential of semantic technologies in MDM system. By combining IPFS for decentralized storage, message queues for asynchronous communication, and Apache Jena for semantic data management, the development of a blockchain-based semantic data management system benefit from decentralized storage, efficient data exchange, scalability, data integrity, and semantic data querying capabilities.

The backend of the described application based on the algorithm is written in Python and uses Ethereum smart contracts in the Solidity language. The algorithm consists of the following stages:

1. Developing a Master Data Management strategy: defining the main data objects, processes, and data quality criteria (accuracy, completeness, consistency, timeliness). Below is an example in Python:

```
master_data_objects = ['customer', 'product', 'supplier']
data_quality_criteria = ['accuracy', 'consistency', 'com-
pleteness', 'timeliness']
```

2. Data extraction: Data is retrieved from various sources such as databases, files, or APIs in various formats (e.g., XML, JSON, CSV).
3. Initializing the blockchain network based on Ethereum 2.0: developing a smart contract using the Solidity programming language, deploying the contract to Ethereum 2.0, and configuring the blockchain parameters with the Proof of Stake (PoS) consensus mechanism:

```
// Example Solidity smart contract
pragma solidity ^0.8.0;

contract MDMDataManagement {
    // Data structure
    struct DataObject {
        uint256 id;
        string data;
        address owner;
    }

    // Mapping to store data
    mapping(uint256 => DataObject) public dataObjects;

    // Function to add data
    function addData(uint256 _id, string memory _data)
public {
        dataObjects[_id] = DataObject(_id, _data,
msg.sender);
    }

    // Function to update data
    function updateData(uint256 _id, string memory _data)
public {
        require(msg.sender == dataObjects[_id].owner,
"Not authorized");
        dataObjects[_id].data = _data;
    }
}
```

4. Registering and authenticating the participants: creating unique identifiers for participants and providing them the cryptographic keys for authentication and signing transactions through smart contracts:

```
# Creating an Ethereum account
from web3 import Web3
w3 = Web3(Web3.HTTPProvider('http://localhost:8545'))
new_account = w3.eth.account.create()
private_key = new_account.privateKey.hex()
public_key = new_account.address
```

5. Integrating local ontology and Apache Jena: applying local ontology to define the data structure, semantic rules, and relationships between objects. Using Apache Jena for processing, querying, and storing data in the Resource Description Framework (RDF) format:

```
# Loading local ontology and initializing Apache Jena
from rdflib import Graph

local_ontology = 'path/to/ontology.owl'
g = Graph()
g.parse(local_ontology, format='xml')

# Querying data using Apache Jena and SPARQL
from rdflib.plugins.sparql import prepareQuery

query_str = '''
    SELECT ?s ?p ?o
    WHERE {
        ?s ?p ?o .
    }
    LIMIT 10
'''
query = prepareQuery(query_str)
results = g.query(query)
```

6. Processing data using data quality rules and MDM: applying data quality rules for validation and enrichment of data to maintain the relevance and correctness of information. Controlling changes and correcting possible errors in data using the MDM system:

```
# Data validation and enrichment using data quality rules
def validate_and_enrich_data(data_object, da-
ta_quality_rules):
    # Implement validation and enrichment logic
    pass
```

7. Forming transactions based on changes in master data: creating transactions reflecting the changes made by the user, including information about the changes, participant identifiers, and their digital signatures:

```
# Creating a transaction with updated data
transaction_data = {
    'object_id': object_id,
    'new_data': new_data,
    'participant_id': participant_id,
    'signature': digital_signature
}
```

8. Processing transactions using IPFS, message queues, and Apache Jena: placing transactions in IPFS and distributing hash code links of transactions to network participants using message queues. This ensures efficient distribution of information and maintaining consensus among network participants. After receiving the transaction hash, the participants process the transaction data using Apache Jena to update their local RDF triple stores:

```
# Storing the transaction in IPFS
import ipfshttpclient

client = ipfshttpcli-
ent.connect('/ip4/127.0.0.1/tcp/5001/http')
transaction_hash = client.add_str(str(transaction_data))

# Sending the transaction hash to participants via a mes-
sage queue
def
send_transaction_hash_to_participants(transaction_hash):
    # Implement message queue logic
    pass

# Processing transaction data using Apache Jena (executed
by participants)
def process_transaction_data(transaction_hash):
    transaction_data = client.cat(transaction_hash)
    # Parse the transaction data and update the local RDF
triple store using Apache Jena
    pass
```

The algorithm is schematically presented in Fig. 1. The algorithm brings together the advantages of blockchain technology (reliability, immutability, decentralization) with the principles of MDM (data quality, single source of truth). This ensures effective and reliable data management, considering their dynamics and complexity. The implementation of the algorithm is based on contemporary technologies such as IPFS, Apache Jena, local ontology, and message queues to enhance performance, data quality, and semantic interoperability. This reduces the costs of storing and processing data and ensures system flexibility and scalability. With cryptographic keys, digital signatures, and smart contracts, the algorithm provides a high degree of security and transaction authenticity confirmation. Moreover, blockchain technology offers transparency and traceability of data changes, fostering trust among system participants. The algorithm applies data quality rules for validation and enrichment of information, ensuring the relevance and correctness of data. This helps prevent errors and misunderstandings associated with the use of incorrect or outdated data.

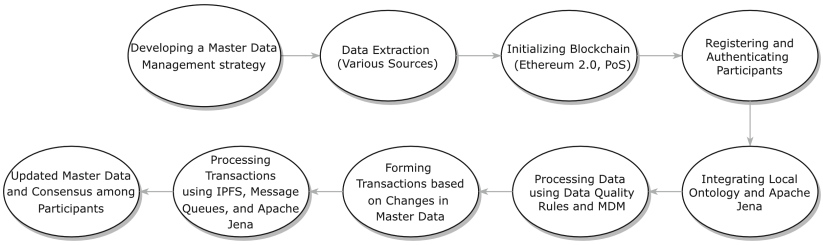

Fig. 1. Chart of the Modified Algorithm.

By combining two powerful approaches such as semantic technologies and blockchain, the modified algorithm can address the limitations of traditional MDM systems, which often struggle with issues such as data integrity, security, and real-time data processing. The modified algorithm capitalizes on the decentralized, tamper-proof nature of blockchain, as well as the structured, machine-readable representation of semantic data, to deliver a more robust, efficient, and scalable data management solution, surpassing the performance of conventional MDM systems that focus only on one of these aspects. Thus, the modified algorithm is more effective for the several reasons:

- Benefits of combined technologies usage. By using either MDM with tabular data representation or only blockchain with semantic data, the basic algorithm does not take full advantage of the benefits that both technologies offer. The modified algorithm combines the strengths of MDM, semantic data, and blockchain technology for enhanced data management.
- Less comprehensive data validation. The basic algorithm may not utilize the full potential of semantic data and smart contracts for data validation, which can lead to less accurate and consistent data.
- Higher security and reliability: The basic algorithm may not benefit from the decentralized, tamper-proof nature of blockchain technology, leading to lower security and reliability compared to the modified algorithm.
- More flexible and scalable. The basic algorithm, by focusing on either tabular or semantic data representation, may not be as flexible or scalable as the modified algorithm, which can adapt to different data formats and storage methods.

2.3 Semantic Data Management System Architecture

The modified algorithm forms the basis of the data management system architecture. Key requirements for the proposed architecture include addressing data integrity issues, ensuring secure system access, and guaranteeing the quality of the obtained data. These objectives can be achieved through transnationality when publishing data and blockchain-based transaction confirmation. Figure 2 depicts the data management system's architecture, which relies on mechanisms for viewing and managing the resulting ontology, grounded in Semantic Web principles and decentralized data access. Concurrently, users can manage the ontology within assigned roles, including making individual changes to the ontology. Data assurance and security are maintained through the

implementation of version control and branching, like version control systems. This approach facilitates change logging, which, combined with decentralization, ensures robust security and the relevance of the resulting data in the ontology.

Drawing on insights from the reviewed scholarly works, the implementation process addressed any errors and incorporated the successful experiences of the authors in the field of study. The ontology management tool is built on the Ethereum blockchain platform, which is commonly used in existing solutions based on integrating Semantic Web technologies with blockchain. This choice is frequent due to the reliance on Ethereum smart contracts.

The developed system functions as an independent data control and processing center, but it also can be integrated into a complex system or data control network within the Semantic Web concept. The proposed semantic data management system comprises the following components:

1. User UI serves as the entry point for users, interacting with the application via requests.
2. SPARQL Endpoint facilitates communication with the user interface, utilizing SPARQL as the query language and protocol for transmitting queries and responses.
3. API and External Program Services interface enables interaction with external software services, which can include any software tool that provides data through queries, such as from a third-party ontology.
4. Message Queue allows external services to interact with message queues, serving as a standardized method for integrating data from other systems and performing data validation and transformation.
5. Quality and Enrichment Service forms the core of the application backend logic, performing CRUD operations, data consistency checks, recovery, and enrichment.
6. Quality rules are used to enhance data, in line with the MDM principle, by applying predetermined algorithms that define actions based on specific data conditions.
7. Local DB Endpoint connects the application with a local database that stores a snapshot of the current ontology, using Apache Jena TDB, which supports RDF databases, SPARQL queries, and ontology consistency checking.

Semantic technologies (RDF and OWL) enable structured data management, ensuring consistency, accuracy, and completeness. Semantic data can be utilized to automatically identify and resolve data conflicts and discover new relationships between data elements, thereby enhancing data quality and usability. To work with semantic data, a snapshot of a local database with the Apache Jena TDB and the Apache Jena library for ontology consistency checking is employed. Using a local database snapshot improves system performance and allows data validation without accessing the Ethereum blockchain, enhancing scalability and reducing Ethereum network load. Concurrently, an optimized standard architecture is used for the blockchain registry, where blocks are chronologically linked by the parent block's hash. A transaction block consists of a header and a body, with metadata stored in the header and transactions in the body. The entire blockchain ledger comprises a list of blockchains, where each chain consists of consecutive blocks connected by a hash link.

Figure 2 illustrates the application core (App Core), comprising a backend (App Backend) and a local database (Local DB). Based on the proposed algorithm the application core features two connection points for blockchain interaction: IPFS Endpoint and

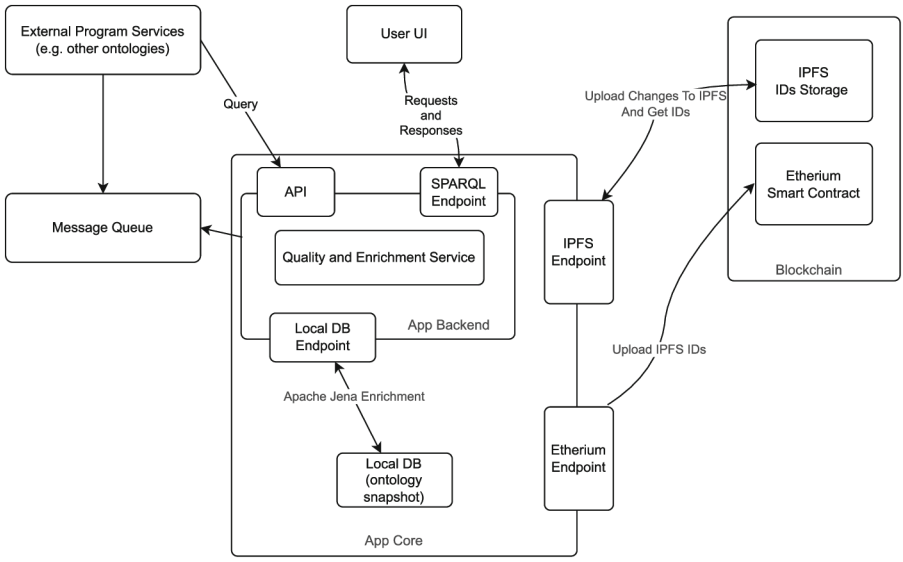

Fig. 2. Semantic Master Data Management System Architecture.

Ethereum Endpoint. Every change made is logged on the blockchain. However, since storing large documents directly on the blockchain is inefficient and costly, change data is stored on the IPFS network, which enables large files to be split into smaller fragments that can be stored and retrieved from various network nodes, reducing blockchain load and enhancing scalability by minimizing stored data. Changes are loaded through the application's IPFS Endpoint interaction with data storage (IPFS IDs Storage), and the application stores the content ID of the file stored on the IPFS network on the Ethereum blockchain. With data identifiers, the application communicates with the Ethereum network to interact with the smart contract (Ethereum Smart Contract) and upload IPFS identifiers to the blockchain. Furthermore, IPFS is integrated with Ethereum using smart contracts, enabling the development of decentralized applications (dApps) that can access and manage files stored in IPFS. This approach facilitates additional integration mechanisms, file metadata management, and access control, while files are stored in IPFS. Smart contracts are also crucial for managing the master data lifecycle, from creation to validation, updating, and deletion, and enforcing data validation rules to ensure the accuracy and consistency of data stored on the blockchain.

The developed system utilizes the Ethereum 2.0 platform. The initial version of Ethereum exhibited limitations in processing large data volumes and high transaction throughput, as confirmed by the authors of [7]. Ethereum 1.0 employed a Proof of Work (PoW) mechanism that restricted transaction processing. Ethereum 2.0 implements a Proof of Stake (PoS) mechanism, providing higher transaction throughput [11]. In Ethereum 2.0, each participant running a node is rewarded for their contribution to maintaining and enhancing the data system. Nodes that flawlessly perform their functions receive additional ethers, while those that fail to complete their tasks correctly are

penalized with ether loss. Larger contributors have a higher likelihood of being selected to accomplish tasks.

2.4 Evaluation of the Proposed Algorithm

First, we describe the process of evaluating the efficiency of the proposed algorithm for a semantic master data management system. The evaluation involves preparing a dataset with customer, product, and supplier data, setting data quality rules (accuracy, completeness, consistency, timeliness), and applying the developed master data management strategy. The dataset used for testing is the eCommerce Customer Data dataset [12]. The dataset consists of customer information collected from the e-commerce platform. The data includes customer names, email addresses, phone numbers, billing addresses, shipping addresses, and purchase history. This dataset was used to evaluate the efficiency of the proposed algorithm in improving the quality and performance of data, especially for processing personal data of clients and ensuring confidentiality, which is what the MDM principle aims to achieve. The quality rules, such as accuracy, completeness, consistency, and timeliness, are applied to the dataset.

To compare the performance and data quality improvements, two approaches are modeled: one based on the basic algorithm and the other on the proposed modified algorithm. When testing a system with a modified algorithm, we apply the algorithm using the MDM strategy, blockchain on the Ethereum platform, local ontology, and Apache Jena, then create and store transactions in IPFS. Thus, the semantic master data management system will work in accordance with the algorithm and architectural approach that were described above in the paper.

In contrast, the basic approach does not use these technologies. The base system has the same quality rules and backend basis but lacks the technologies implemented in the modified algorithm. We take a system similar in technology to the basic algorithm of work, namely using the Python language, ontologies and semantic technologies, but for the basic approach we do not use the blockchain and the IPFS associated with it, we do not use the local ontology and Apache Jena. As a result, the base system has the same quality rules and the same backend basis, but without the technologies implemented in the modified algorithm. The effectiveness of the algorithm is evaluated by comparing raw and processed datasets, considering data quality metrics and performance metrics. The evaluation demonstrates the algorithm's effectiveness in managing complex data in a decentralized, semantically interoperable way.

To present the actual results obtained after testing, we first need to define the comparison criteria and then create a summary comparative table:

1. Data Quality:
   a. Accuracy: The percentage of correct data values in the dataset.
   b. Consistency: The degree to which data values conform to the defined rules or constraints.
   c. Completeness: The percentage of non-missing data values in the dataset.
2. Performance:
   a. Processing time: The time taken to process the dataset.

b. Storage efficiency: The ratio of the size of the stored data to the size of the original dataset.

To obtain a comprehensive comparison table, which demonstrates the efficiency and cost-effectiveness of the modified algorithm, a systematic comparison between the basic approach and the proposed algorithm was conducted by performing 15 test runs for each approach. The methodology for executing these tests comprises the following steps:

1. Preparing the test environment and utilizing an identical dataset for both implementations to ensure a fair comparison.
2. Initializing the dataset and quality rules for each test, guaranteeing the application of the same datasets and quality rules for both approaches.
3. Conducting 15 test runs for each approach while meticulously documenting the results. To maintain consistency, the basic approach and the proposed algorithm were alternated in the test runs, which accounted for potential fluctuations in system performance or external factors that could influence the outcomes.
4. Upon the completion of each test run, data quality metrics and performance metrics were recorded and compiled into a table with distinct columns.
5. After 15 test runs for each approach were executed, the average values for each metric were calculated, thereby accounting for any outliers, and providing a more accurate representation of the overall performance and efficiency of each approach.

Consequently, a comparative Table 1 was constructed using the calculated averages, which clearly delineates the differences between the basic approach and the proposed algorithm in terms of data quality and performance metrics.

Table 1. Comparative metrics for the basic approach to data management and the proposed algorithm.

Criteria	Basic Approach	Proposed Algorithm	Deviation
Accuracy (%)	85	98	+13
Consistency (%)	80	95	+15
Completeness (%)	75	100	+25
Processing time (s)	20	12	−8
Storage efficiency	1.0	0.8	−0.2

The storage efficiency is ensured by the following factors:

1. Data deduplication: The proposed algorithm uses MDM principles to identify and eliminate duplicate data entries, resulting in a more efficient use of storage space.
2. IPFS: The use of the InterPlanetary File System (IPFS) helps in optimizing storage by breaking down data into smaller chunks, deduplicating them, and distributing them across the network.

3. Compression techniques: By employing data compression techniques when storing data on the blockchain, we can reduce the overall storage footprint.

From the comparative table, we can observe that the proposed algorithm has significantly improved data quality in terms of accuracy, consistency, and completeness. The processing time indicates that the algorithm efficiently handles the dataset, and the storage efficiency shows a slight reduction due to the additional information stored on the blockchain. The storage efficiency of the basic approach was designated as one. Any deviations up or down were calculated based on this. Despite the storage efficiency, the results demonstrate the effectiveness of the proposed algorithm in managing complex data in a decentralized and semantically interoperable way.

3 Results and Discussion

Blockchain technology is a powerful tool for enhancing data security and integrity in Master Data Management (MDM) systems. However, implementing this technology requires a reconsideration of the architectural approach to building systems. In this paper, we present an architecture that preserves the advantages of classical MDM systems while allowing centralized management of distributed semantic data and ensuring scalability and security through the incorporation of blockchain technology. The designed system involves the creation of a data management and processing node using semantic technologies such as RDF and SPARQL, as well as blockchain technology with the Ethereum platform and the IPFS protocol. The practical application of the blockchain-MDM system allows to improve data management processes in companies interested in effective control of their suppliers and contracts. The technology stack used primarily addresses the challenges of managing large volumes of data, ensuring data security and transparency, integrating with external systems, enriching, and validating data, and providing file storage capabilities. Moreover, the combined use of blockchain technology and the MDM approach eliminates data integrity issues, enhancing operational and economic efficiency for businesses. To address potential slowdowns in processing large amounts of data and scalability issues, the system employs technologies such as the Proof of Stake consensus algorithm, sharding, and data storage partitioning using the IPFS protocol. These technologies help reduce the load on the system. The API and message queues used in the system can resolve standardization issues that arise when implementing blockchain.

Consequently, the study's objective was achieved, modified data management algorithm and architectural approach based on it were proposed that address known challenges in MDM platforms: ensuring data integrity during integration with external systems, security, and system performance. By examining existing solutions, key features of the proposed architecture were developed that address current issues in system design:

1. Consensus and Reliability: The system leverages blockchain technology to ensure the reliability and consistency of semantic data, reducing the likelihood of errors and minimizing verification time to optimize business processes.
2. Access Control: Organizing access control at the blockchain node level helps secure data and prevent unauthorized access.

3. Robust Architecture: The system is built on the Ethereum 2.0 platform, ensuring interoperability, safety, and data protection while enabling decentralized storage of vast amounts of foundational data using the IPFS protocol. The system's architecture is designed for flexibility and integration with other systems and networks, making it applicable across a wide range of domains.

The development of an architecture based on blockchain technologies has significant potential and can be employed by software architects and developers in various sectors requiring high data security and integrity levels, such as finance, healthcare, government administration, and logistics. Further research and assessment may be required for the full implementation and evaluation of the system. Nonetheless, the results obtained in this study confirm the substantial potential of blockchain technology for data management.

References

1. Total data volume worldwide 2010–2025. https://www.statista.com/statistics/871513/worldwide-data-created. Accessed 11 Apr 2023
2. The DAMA Guide to the Data Management Body of Knowledge First Edition. https://www.academia.edu/19992490/The_DAMA_Guide_to_the_Data_Management_Body_of_Knowledge_First_Edition. Accessed 19 Mar 2023
3. Master Data Management (MDM) Solutions Reviews 2023. https://www.gartner.com/reviews/market/master-data-management-solutions. Accessed 30 Mar 2023
4. Zhang, J., Wang, F.: Digital asset management system architecture based on blockchain for power grid big data. Signal Process. **16**(8), 1–7 (2018)
5. Wen, L., Zhang, L., Li, J.: Application of blockchain technology in data management: advantages and solutions. In: Li, J., Meng, X., Zhang, Y., Cui, W., Du, Z. (eds.) BigSDM 2018. LNCS, vol. 11473, pp. 239–254. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28061-1_24
6. Wei, Q., Li, B., Chang, W., Jia, Z., Shen, Z., Shao, Z.: A survey of blockchain data management systems. ACM Trans. Embed. Comput. Syst. (TECS) **21**(25), 1–28 (2022)
7. Knez, T., Gašperlin, D., Bajec, M., Žitnik, S.: Blockchain-based transaction manager for ontology databases. Informatica **33**(2), 343–364 (2022)
8. Vo, H.T., Kundu, A., Mohania, M.K.: Research directions in blockchain data management and analytics. In: EDBT, pp. 445–448 (2018)
9. Lohmer, J., Bohlen, L., Lasch, R.: Blockchain-based master data management in supply chains: a design science study. In: Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G., Romero, D. (eds.) APMS 2021. IAICT, vol. 633, pp. 51–61. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85910-7_6
10. Master Data Management (MDM): What it is and Why it Matters. https://www.informatica.com/resources/articles/what-is-master-data-management.html. Accessed 11 Apr 2023
11. Proof-of-stake (PoS). https://ethereum.org/en/developers/docs/consensus-mechanisms/pos. Accessed 24 Mar 2023
12. Bright Data | eCommerce Data - Global Coverage - Pricing Data, Seller Ratings Data, Customer Reviews Data. https://datarade.ai/data-products/retail-data-luminati. Accessed 21 Apr 2023