Blockchain for Public Transportation: Digital Identity and Transaction Verification Architecture

Hidayet Burak Saritas [1 2][0000-0002-0425-3051] and Geylani Kardas [1][0000-0001-6975-305X]

[1] Ege University International Computer Institute, 35100, Bornova – Izmir, Türkiye
[2] Kentkart Teknoloji AŞ, Ege Teknopark, Ege University, 35100, Bornova – Izmir, Türkiye
burak.saritas@kentkart.com
geylani.kardas@ege.edu.tr

Abstract. Blockchain technology has emerged as a promising solution to address key challenges in public transport, such as interoperability, privacy and transaction transparency. Despite these advances, the lack of a unified platform for integrating different transport modes and the diversity of ticketing solutions across operators remain significant barriers to seamless collaboration and interoperability. This paper proposes a blockchain-based architecture for transaction verification and digital identity management in multimodal public transport systems. The approach ensures secure co-payment, reliable data validation and trust among diverse operators by leveraging private blockchain networks, where operators act as verification nodes through consensus-based transaction approval without a central authority. The system incorporates standardized data formats, advanced algorithms for operator data integration, and a comprehensive model for managing operators, assets, and transactions. To protect user privacy, Zero-Knowledge Proof (ZKP) techniques enable secure authentication without revealing sensitive information. This study shows how blockchain technology can improve interoperability, ensure fair revenue sharing, and provide a secure, decentralized infrastructure for efficient and privacy-preserving collaboration in public transport networks.

Keywords: Blockchain Technology, Public Transportation, Co-payment Systems, Data Verification, Interoperability, Transaction Security, Decentralized Networks, Zero-Knowledge Proofs, Standardized Data Format, Privacy Preservation.

1 Introduction

With advancements in technology, the public transportation sector is expanding and offering a variety of options to a wider audience. In addition to the traditional methods, new alternatives like shared vehicles, scooters, and app-controlled taxis are becoming more popular [1]. Regulations in this field set limits to prevent unfair competition and encourage cooperation. Mobility partners have the freedom to make their own decisions. Various public transport providers need to work together on a common platform to manage this diversity and deliver an enhanced user experience [2]. This can improve

users' transportation experiences and reduce private vehicle usage. However, actors who offer different transportation methods have their own ticketing solutions. This may pose a significant challenge in the development process of a unified platform [3]. These actors include ABT Kentkart [4], STIB-MIVB [5], MVV [6], and Whim [7]. Users need to adapt to various ticketing solutions, and each actor needs to offer features such as payment, personalization, and usability [8-9]. Therefore, creating a unified and accessible platform within the public transportation ecosystem is considered a significant innovation and challenge for the sector [10].

It is not easy to combine different actors of the public transportation industry. But it can be made simpler and efficient with following solutions like a single application, account management and a single card. To combine services and share profits, researchers are constantly exploring easy integration methods to create a common language that all transportation solution providers can use [11]. They are also working to establish a method for verifying every user transaction [10]. To solve these problems, this study proposes developing a blockchain-based solution that processes and verifies data produced by different transportation actors. Blockchain is a distributed ledger system that operates without central authority, ensuring data security and transparency [12]. In essence, Distributed Ledger Technology (DLT) is a secure and decentralized database that records transactions in an immutable manner, providing a trusted environment for data exchange [13]. It enables trust in transactions between two parties and eliminates the need for a central intermediary to provide this service. This allows for the secure transfer of unique assets, such as money, title deeds, and identification information, without intermediaries. Two users can conduct a financial transaction without the need for an intermediary institution [14]. They can communicate directly without any trust issues [15].

This study extends our previous conference paper on a blockchain-based transaction verification infrastructure [16], by focusing on the detailed architecture of digital identity and transaction verification in public transportation systems. The earlier work primarily focused on a standardized message format, transaction verification on Hyperledger Fabric among transportation companies, and privacy through Zero-Knowledge Proofs (ZKPs). In contrast, the current study extends adoption of these technologies significantly by providing a detailed technical architecture with explicit roles and interactions of components, implementing smart contracts and an API gateway, as well as enhancing digital wallet functionality for managing Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs). Additionally, a dual-chain architecture is introduced, integrating both private and public blockchains to achieve greater scalability, interoperability, and privacy preservation. These advancements collectively contribute to a more robust and practical blockchain architecture for public transportation.

The standard message package format was developed on a private blockchain network. This format creates an environment where transportation operators can add data. The study considered parameters such as speed and assets that validating operators must have. This way, different operators in the public transportation sector can use blockchain technology to integrate with each other, trust each other, and query all transactions created with standard message format. Creating a trustworthy environment is crucial and can be achieved through a consensus mechanism and blockchain structure to

confirm transactions. All businesses can join the blockchain network as validators, and data produced by any business can be added to the network after being approved by all nodes. The data added to the network in standard message format is trusted by everyone. To achieve this, a common data format and extensible approval mechanism have been created for the public transportation sector.

The rest of the paper is organized as follows: Section 2 reviews related work on blockchain applications in similar transportation sectors. In Section 3, we introduce the foundational elements and components of a blockchain environment optimized for public transportation. This section covers the architecture, roles, and interactions of transactions, as well as detailed descriptions of nodes, assets, and standardized message formats. In Section 4, we look at how to use decentralized technologies to keep people's information safe in public transportation. These technologies let people manage their digital identities without revealing sensitive information, which helps enhance people's privacy. We also look at how these technologies can be used with a single blockchain-based transaction verification system to verify user credentials. In Section 5, we summarize the contributions and implications of our blockchain-based architecture tailored for the public transportation sector.

2 Related Work

To address the outlined challenges, several studies have explored blockchain applications in public transportation. This section reviews key works in this area and highlights their relevance to our study. Some notable research has been done on using blockchain for public transportation. Jayalath et al. [17] propose a micro-transaction model based on blockchain to improve service in Sri Lanka's public transportation sector. They focus on a ticketing system using an Ethereum-based blockchain to reduce transaction fees and improve service quality. This approach creates QR-based tickets for users to make micro-payments without third-party intermediaries.

Wang et al. [18] introduce "InterTrust," an interoperable blockchain architecture to enhance interoperability and reliability across various blockchain systems. The InterTrust model is for communication and interoperability among existing blockchain systems, which is a broader scope. However, our study aims at providing a specialized blockchain network for the public transportation domain.

Yang et al. [19] suggest a blockchain and Edge Computing-based communication system for maritime transportation. Their work uses blockchain and Edge Computing to improve Internet of Things (IoT) device performance and security in maritime environments. This is different from our work, which focuses on public transportation.

Enescu et al. [20] discuss a blockchain application to promote ecological transportation and reduce traffic congestion. They imagine a system where blockchain records transactions and gives users digital currencies, which helps the environment and makes public transportation more popular.

Jabbar et al. [21] review blockchain applications in Intelligent Transportation Systems (ITS). They show how blockchain can improve transactional trust and efficiency. This supports various functionalities including automatic parking and fee payments.

Ganzha et al. [22] introduces a framework combining Blockchain-based Self-Sovereign Identity (SSI) and registry proof smart contracts to provide privacy-preserving solutions for Inter-Organizational Business Processes (IOBP). This approach effectively addresses the challenges of data sharing across organizations due to regulatory restrictions like the General Data Protection Regulation (GDPR) and demonstrates its applicability through a pharmaceutical supply chain case study on the Ethereum Blockchain.

Karataş et al. [23] propose a Self-Sovereign Identity (SSI)-based e-petition scheme utilizing the Sovrin blockchain. Their approach employs decentralized identifiers (DIDs) and verifiable credentials to ensure user privacy and anonymity while participating in e-petitions. This method addresses privacy concerns by enabling secure citizen engagement without revealing sensitive personal information.

Lastly, Chen et al. [24] describe a "Full-Spectrum Blockchain as a Service" (FSBaaS) approach with "Blockchain Lite" and "Hyperledger Fabric," focusing on providing blockchain services that cater to both centralized and decentralized frameworks. This study shows the flexibility needed in blockchain adoption.

Our study is different from the above-mentioned noteworthy efforts by providing a mechanism for public transportation operators to establish mutual trust. Each operator can independently verify and validate data transactions without relying on a central authority or intermediary. This ensures the accuracy of data and consistency of messages across the network, fostering seamless collaboration among operators. It facilitates collaboration among operators without dependence on a single authority. Additionally, this study improves privacy and security in public transportation systems by using DIDs, VCs, ZKPs explained in later sections. It supports user privacy by authenticating users and transactions without exposing sensitive personal information. This approach not only safeguards digital identities and transactions but also sets a precedent for incorporating privacy-preserving technologies in public transportation. The proposed framework introduces a standard message format and private blockchain network tailored for the public transportation sector. By utilizing consensus mechanisms and decentralized identifiers, the system ensures accurate transaction validation, secure data sharing, and enhanced privacy for users. This research aims to enable seamless integration among operators, establish trust, and set a precedent for privacy-preserving technologies in public transportation.

3 Key Elements and Core Components of the Blockchain Model

In this section, we present a detailed overview of the main model elements and foundational components of a blockchain-based transaction verification system specifically designed for public transportation. This includes an exploration of the core building blocks, the interactions between various system elements, and the overall architecture that facilitates secure, efficient, and transparent transaction processing across multiple transportation operators.

3.1 High-Level Overview of the Model in a Public Transportation Network

This section provides a conceptual view of the proposed blockchain-based transaction verification system for public transportation. The model integrates various transportation actors, enabling secure and transparent transactions while ensuring data privacy. Figure 1 illustrates the high-level design of the system, highlighting key components and their interactions within the network.

Core Concept of the Model. At the heart of the system, there exist the transactions, representing user actions such as ticket purchases or payments made across multiple transportation modes, including buses, scooters, and cars. These transactions are routed through the blockchain network, where a consensus mechanism ensures their validity and integrity. Participating businesses, acting as network nodes, collaboratively verify transactions without relying on a central authority. This decentralized approach fosters trust and reduces reliance on intermediaries.

Blockchain Environment and Consensus Mechanism. The system is built on the Hyperledger Fabric blockchain environment, which is tailored to enterprise-level applications [25]. Hyperledger Fabric provides:

Enhanced Transaction Control. Organizations can define private channels to restrict the visibility of specific transactions and data, ensuring privacy among participants.

Scalable Consensus. The system utilizes the Raft consensus algorithm, known for its ability to efficiently handle high transaction volumes with minimal latency [26]. Raft's practical fault tolerance makes it ideal for the dynamic and high-demand operations in public transportation [27]. Organizations (businesses) on the network verify transaction accuracy through a consensus mechanism [28].

Ensuring User Data Privacy. To safeguard user data, the system leverages advanced cryptographic techniques such as ZKPs:

1. Sensitive information like user identity or payment details is never directly revealed on the blockchain.
2. Instead, cryptographic proofs are included in transactions, allowing validators to confirm the authenticity of the transaction without accessing private data.
3. This mechanism ensures that only the essential details required for verification are shared, maintaining the integrity of the system while protecting user privacy.

Role of Smart Contracts. Hyperledger Fabric smart contracts (chaincodes) define the business logic of the system and automate transaction processing. These chaincodes:

1. Validate and enforce transaction rules.
2. Enable operations such as ticket issuance, membership validation, and fee payments.

3. Once validated, transactions are added to the blockchain ledger, which serves as a secure, immutable, and transparent record of all operations.

Visualization and Monitoring with Blockchain Explorer. To enhance transparency and monitoring:

1. A blockchain explorer is incorporated, allowing stakeholders to visualize, query, and track transactions and blocks.
2. This tool provides critical functionalities for auditing, reporting, and system monitoring, ensuring accountability across all participants.

Role of Network Participants.

1. Businesses (Organizations): Act as nodes within the blockchain network, validating transactions and managing data related to their services.
2. Users: Initiate transactions (e.g., purchasing a ticket) via client applications that interact with the blockchain network.
3. Interconnected Nodes: Collaborate to validate transactions, ensuring decentralized governance and efficient data management.

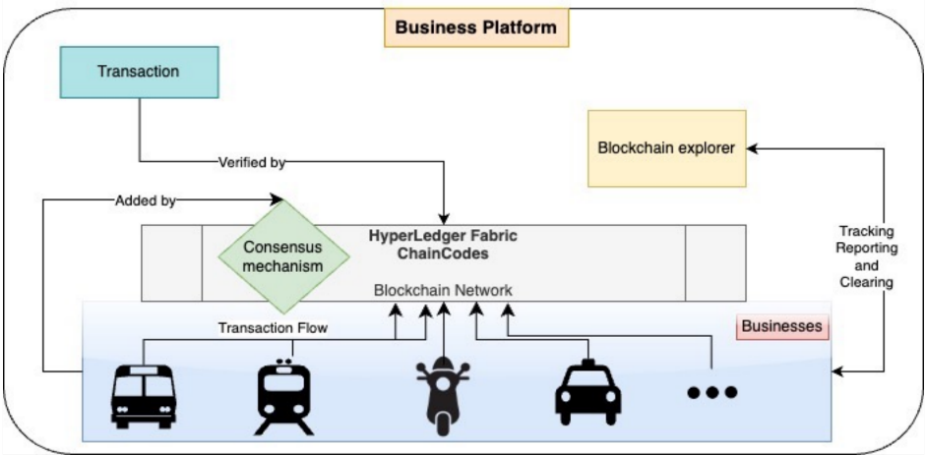

Fig. 1. High Level System Design

This high-level architecture provides the conceptual framework for secure and efficient transaction verification, while the underlying infrastructure ensures its seamless operation. It integrates privacy-preserving technologies, decentralized collaboration, and transparent data management, enabling seamless operations across diverse transportation operators while safeguarding user information.

3.2 Detailed Technical Architecture of the Blockchain Model

This section dives into the technical details of the proposed blockchain-based system, focusing on the roles, functionalities, and interactions of individual components. The system leverages Hyperledger Fabric's modular architecture and privacy features to meet the specific needs of public transportation. Figure 2 depicts the technical architecture, which ensures secure, decentralized, and efficient transaction processing.

Key Technical Components.

User. This is the end-user of the public transportation network. Users interact with the system via client applications to undertake actions such as purchasing tickets, validating them, and making enquiries regarding their balance.

Client Applications. These comprise mobile and web applications that serve as the principal interface through which users interact with the blockchain network. The applications process user requests and relay them to the blockchain infrastructure via the client gateway.

Client gateway. Serves as an intermediary between client applications and the blockchain network. The gateway validates incoming requests and routes them to the appropriate network services, thereby ensuring secure communication.

Businesses. Refers to the various transportation operators or stakeholders that are participants in the blockchain network. Each organization has its own set of peers, the responsibility of which is to validate transactions and maintain the ledger.

Peers. Nodes within the blockchain network that validate transactions, execute smart contracts, and maintain a copy of the ledger. Each organization is represented by multiple peers to ensure redundancy and reliability.

REST APIs. Facilitate interaction between external systems and applications and the blockchain network. The APIs facilitate integration with third-party systems, including those providing payment services or reporting tools.

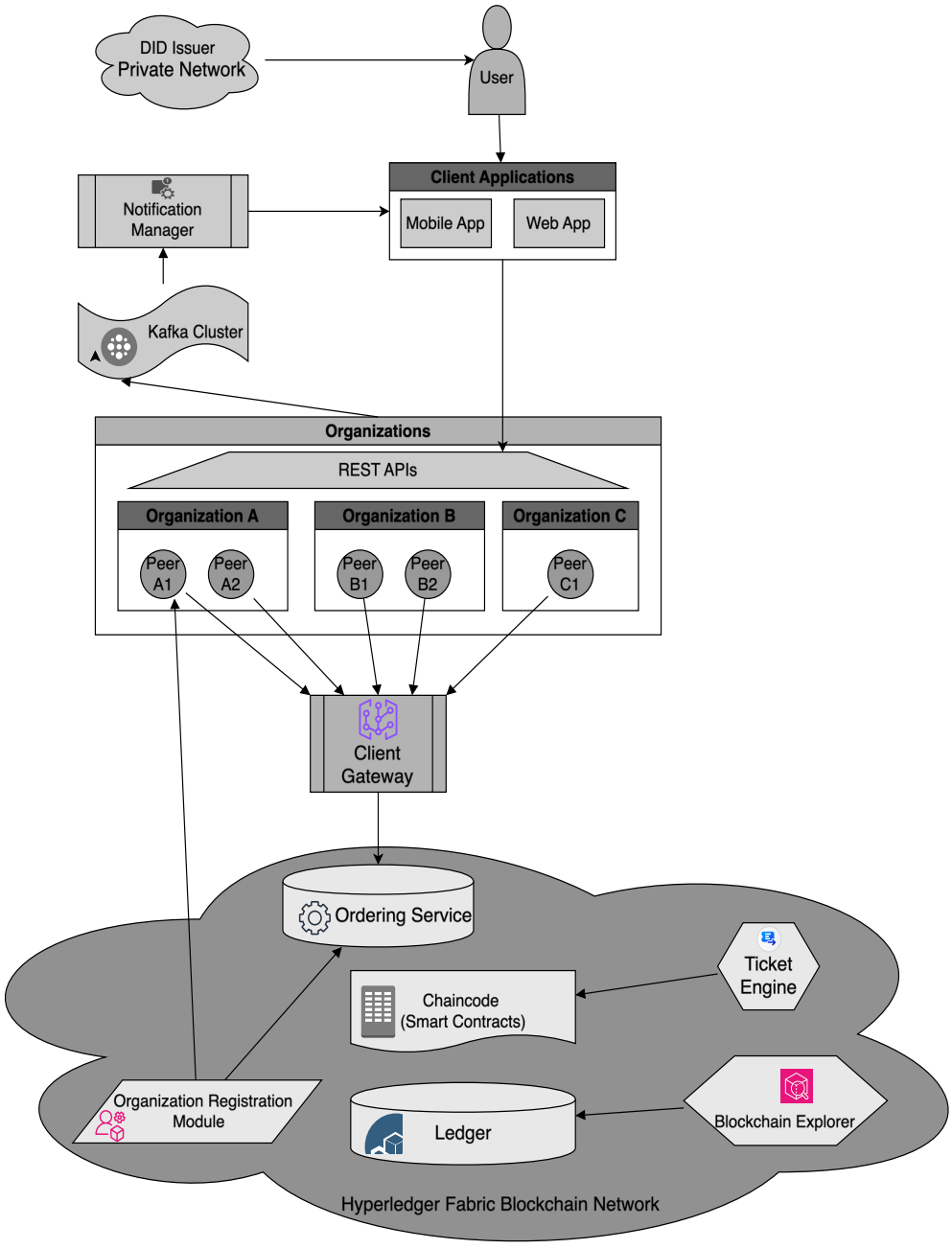

Fig. 2. Technical Architecture of the Blockchain-Based Public Transportation System

The Ordering Service. It is responsible for ensuring the proper sequencing of transactions across the network. It aggregates transactions into blocks and transmits them to peers for validation and incorporation into the ledger.

Ledger. An immutable, distributed database that records all transactions and state changes within the network. It functions as the definitive source of information for all participants.

Chaincode (Smart Contracts). Serves to define the business logic of the network. It processes incoming transactions, enforces the established rules, and executes the required operations in a secure and automatic manner.

Ticket Engine. A specialized module for the administration of ticket-related functionalities, including the issuance, validation, and revocation of tickets. It interacts with the ledger and chain code to guarantee accurate and secure processing.

Kafka Cluster. Provides messaging and event streaming capabilities for the network, ensuring efficient communication between components and handling event-driven processes, such as notifications.

Notification Manager. Manages alerts and updates for users and system operators, ensuring timely communication of critical information, such as transaction statuses.

Organization Registration Module (ORM). It functions as a network manager designed to streamline the process of adding organizations and peers to the blockchain network. The network manager leverages automated services and scripts to manage network expansion securely and efficiently.

Transaction Flow in the Architecture.

1. DID Retrieval (Optional): A user retrieves their DID and corresponding VCs from the issuer network. This process ensures that the user possesses a cryptographically secure and self-sovereign identity that can be used for transactions within the public transportation ecosystem.
2. Membership Creation: During membership creation, one of the verifiers (e.g., a transportation operator) adds the user's transaction information (e.g., membership ID, validity, or privileges) to the blockchain network. This information is linked to the user's DID but does not expose sensitive personal data, ensuring privacy.
3. A user initiates a transaction (e.g., ticket purchase) through a client application.
4. The client gateway receives the request, validates it, and forwards it to the blockchain network.

5. The transaction is processed by the ordering service, which organizes it into a block and distributes it to peers for validation.
6. Peers execute the relevant chaincode (smart contract) to validate the transaction and apply the business rules.
7. Once validated, the transaction is added to the ledger and becomes part of the blockchain's immutable history.
8. The blockchain explorer allows stakeholders to query and visualize the transaction, ensuring transparency and accountability.

By combining these components, the proposed system provides a scalable, secure, and privacy-preserving solution for managing transactions in public transportation networks.

3.3 Definitions and Contents of Model Elements

This section describes the model elements and their roles that form the basis of a private blockchain environment customized for public transportation. The structures used clearly demonstrate how nodes, entities, and transactions on the blockchain are identified, processed, verified, and integrated into the public transportation system.

The diagram in Figure 3 shows the defined classes for the fundamental components of the blockchain-based transaction verification system. The diagram demonstrates how a blockchain network is structured and how different components interact with each other. For example, a transaction executed by a node can trigger a smart contract, resulting in the addition of a block to the blockchain. Nodes communicates using different data formats and messaging protocols. This plays a crucial role in maintaining the security and integrity of the blockchain network.

Nodes (Businesses). Each node in the blockchain network represents a business or organization. These nodes manage various transportation services, such as buses, trams, and scooters, and perform critical roles in the system, including:

Transaction Management. Nodes initiate, process, and approve transactions on the network.

Consensus Participation. Nodes contribute to the consensus mechanism, ensuring the integrity and security of the blockchain

Data Sharing. Nodes communicate using standardized data formats and messaging protocols, enabling seamless collaboration between businesses.

For instance, a transportation operator (e.g., a bus company) acts as a node, issuing tickets and validating user transactions. When a user purchases a ticket, the node records the transaction, which is verified by other nodes before being added to the blockchain.

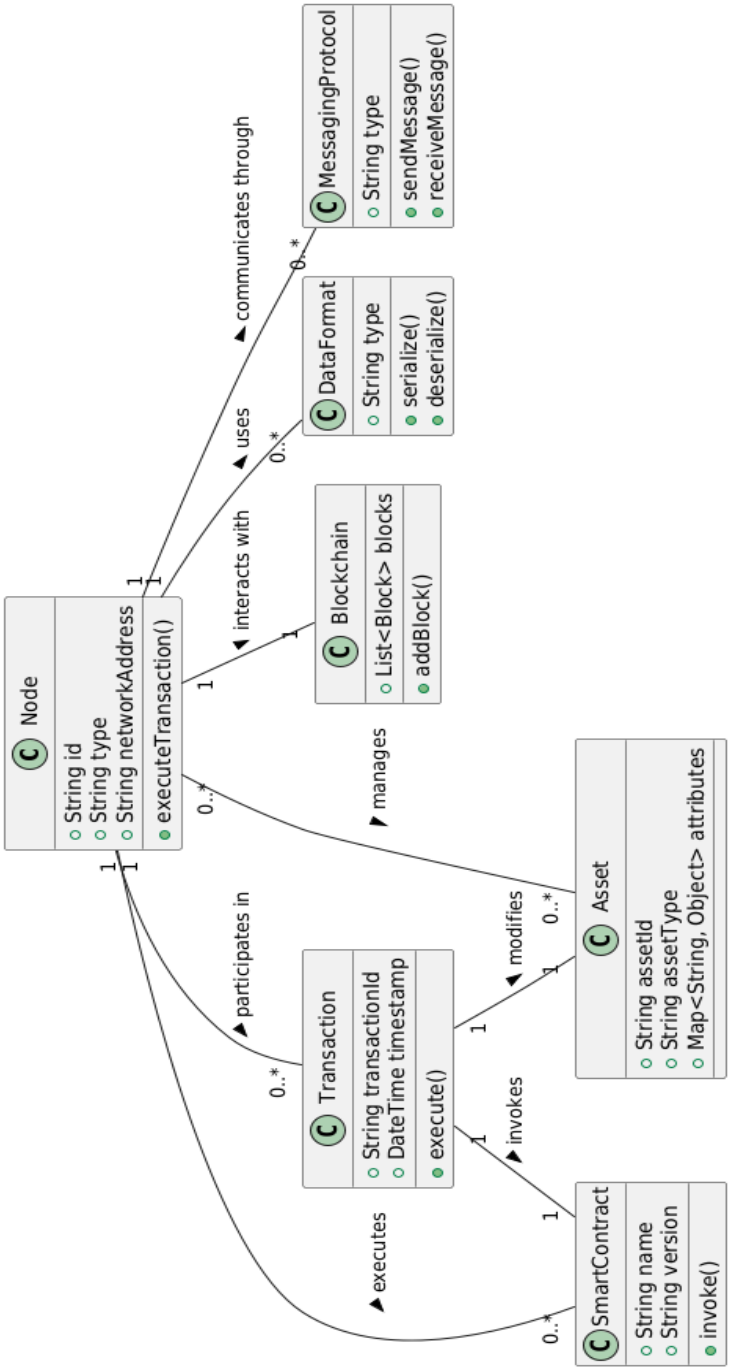

Fig. 3. Interactions between Components of Blockchain-based Transaction Verification System

Assets. Assets represent physical or digital items such as tickets, memberships, or payment records. For example, a ticket sales transaction creates an asset, while a boarding transaction represents the use of a ticket asset. An asset can be any item defined on the blockchain. Each entity has its own attributes, such as those given in the Table 1 defining the attributes of the Assets model element. Each asset has an Asset ID and Asset Type, along with other predefined or optional attributes. Additional attributes may include time information, journey details, or any definition attributes necessary for verifying operations.

Table 1. Message Structure of Assets Model Element

Attribute Name	Attribute Detail
Asset ID (assetId)	Each asset has a unique identifier
Asset Type (asset-Type)	The category into which the asset is classified (for example, 'ticket', 'membership')
Other Attributes	Other information that defines the properties of the asset (for example, validity period, price)

Assets are stored on the blockchain in a JSON-like structure, allowing for extensibility. For example, a ticket asset can include details such as route, validity, and pricing, as shown in Listing 1.

Listing 1. Message Structure of Assets Model Element

```
{
  "assetId": "123456",
  "assetType": "ticket",
  "attributes": {
    "issueDate": "2024-11-05 12:00:00",
    "expiryDate": "2024-12-05 12:00:00",
    "passengerId": "21412",
    "journeyDetails": {
      "origin": "Station A",
      "destination": "Station B",
      "departureTime": "2024-11-05 15:00:00"
    }
  }
}
```

Example Scenario. When a user purchases a monthly pass, the system creates an asset with attributes such as the pass ID, validity dates, and user association. This ensures that the pass can be validated without exposing sensitive user data.

Transactions: Transactions are defined as actions that create a change in the state of assets. Transaction types can include creating, updating, and deleting assets. Attributes of the Transactions model element are defined in Table 2.
Transactions record all movements that users and businesses make on the network. Every transaction must be verified by other nodes in the network. Any unique transaction must contain the Transaction ID, which is a unique identifier, the timestamp of the transaction, and the type of transaction.

Table 2. Message Structure of Transactions Model Element

Attribute Name	Attribute Detail
Transaction ID (transactionId)	Each transaction has a unique identifier
Timestamp	Indicates the time when the transaction took place
Operation Type	Indicates what type of action the operation is (for example, 'create', 'update', 'delete')

With the transaction examples given in Listing 2, it can be explained in detail how a transaction is initiated, how it progresses on the network and how it is concluded. For example, scenarios such as purchasing a ticket and boarding can be handled this way.

Listing 2. Message Structure of Transactions Model Element

```
{
  "transactionId": "tx123456789",
  "timestamp": "2024-11-05 12:00:00",
  "operation": "create",
  "transactionDetails": {
       "assetId": "123456",
       "assetType": "ticket",
       "attributes": {
              // Detailed information about the asset
       }
  },
  "signature": "DigitalSignatureOfTheUser"
}
```

Examples of transaction-specific data (payload) to be added to the message for different transactions in the standard message format to be sent over the network are given below. Listing 3 shows the ticket creation process, Listing 4 shows the membership update process, and Listing 5 shows sample "payload" information for fee payment. These payload samples represent the information required for various public transportation

operations and can be customized according to the needs of the transaction. Payload content may vary depending on the transaction type and the characteristics of the asset being processed.

Payload Examples.

Listing 3. Payload of Ticket Creation Process

```
"payload": {
  "ticketNumber": "1234567890",
  "issueDate": "2024-11-01 10:00:00",
  "expiryDate": "2024-11-02 10:00:00",
  "passengerName": "Hakan Demir",
  "journeyDetails": {
    "origin": "Station A",
    "destination": "Station B",
    "departureTime": "2024-11-01 11:00:00"
   }
}
```

Listing 4. Payload of Membership Update Process

```
"payload": {
  "membershipId": "MEMB1234567",
  "memberName": "Hakan Demir",
  "validFrom": "2024-11-01",
  "validTo": "2024-11-01",
  "membershipType": "Gold",
  "additionalBenefits": ["Extra Luggage", "Priority Boarding"]
}
```

Listing 5. Payload of Fee Payment

```
"payload": {
  "fareId": "FARE12345",
  "amountPaid": "15.00",
  "currency": "USD",
  "paymentMethod": "Credit Card",
  "transactionDate": "2024-11-01 12:30:00",
  "serviceType": "Tram"
}
```

These payloads define the necessary data for various public transportation operations. The content can be customized based on the type of transaction and the asset being processed.

3.4 Standardized General Message Format

It is necessary to determine a general message format to use all the message contents that we defined specifically for assets and transactions in the previous headings on the network. In this way, messages sent on the network can be standardized by using a common message format to produce transaction-specific data. Sample JSON structure and descriptions for the standard message format that can be used among all models in public transportation is defined as in Listing 6.

This format is designed to encapsulate all necessary details for executing and verifying transactions, such as payments or asset transfers. It standardizes data for all parties involved in the transaction and maintains the system's integrity and reliability. Moreover, it supports various transportation modes and is versatile in different scenarios. The message's key components are detailed below.

Transaction Metadata.

- transactionType: Specifies the type of transaction (e.g., Payment, Asset Transfer).
- transactionId: A unique identifier for the transaction, ensuring traceability.
- invokedBy: Identifies the initiator of the transaction, which can be a UserID (for individual transactions) or BusinessID (for operator-initiated actions).

Transportation Details.
This section captures all transport-related information:

- modeOfTransport: Specifies the transport type (e.g., bus, tram, metro).
- startLocation and endLocation: Define the journey’s starting and ending points.

Membership Details.
This section securely encodes membership-specific information for users who are eligible for benefits such as discounts:

- membershipId: A unique identifier for the user’s membership (e.g., student, elder, or disabled).
- proofOfDID: An optional field containing the ZKP generated from the user’s DID.

Transaction Details.
This section details the specific asset being acted upon and the nature of the transaction:

- assetId: The unique identifier of the asset (e.g., a ticket ID, membership ID).
- assetType: The type of asset (e.g., ticket, membership).
- journeyDetails: Includes origin, destination, and departure time.

Listing 6. Standardized General Message Format

```
{
"transactionType": "PaymentorAssetTransfer",
   "transactionId": "UniqueTransactionID",
   "timestamp": "2024-12-01 12:00:00",
   "invokedBy": "UserIDorBusinessID",
   "transportationDetails": {
     "modeOfTransport": "bus/tram/scooter/minibus/metro",
     "routeId": "RouteID",
     "startLocation": "StartingLocation",
     "endLocation": "EndLocation",
     "fare": {
       "amount": "Amount",
       "currency": "Currency"
     },
     "membershipDetails": {
       "membershipId": "MembershipID",
       "validity": "MembershipValidity"
       "proofOfDID": "ZKPProofofUser" (Optional)
       "proofOwnerDID": "DIDofProofOwner" (Optional)
       "verifierDID": "DIDOfVerifier" (Optional)
     }
   },
   "transactionDetails": {
     "assetId": "AssetID",
     "assetType": "AssetType",
     "operation": "create/update/delete",
     "payload": {
       // Customized data fields
       "journeyDetails": {
         "origin": "Station A",
         "destination": "Station B",
         "departureTime": "2024-05-05 15:00:00"
       },
       "fareDetails": {},
       "seatAllocation": {},
     }
   },
   "signature": "DigitalSignatureOfTheUser",
   "consensusDetails": {
     "endorsedBy": ["NodeID1", "NodeID2"],
     "consensusTimestamp": "2024-01-01 12:00:10",
     "consensusAlgorithm": "ConsensusAlgorithmUsed"
   }
`
```

3.5 Practical Implementation

This section outlines the development details of the blockchain-based transaction verification system, including implemented smart contracts and API gateways. It provides process of adding organizations and peers.

Adding New Nodes and Peers. Organization Registration Module (ORM) is responsible for adding new peers and organizations (nodes) to the network. The ORM includes a network manager that processes new requests using dedicated services and scripts. Organization and Peer Controllers validate certificates and execute the necessary code to integrate the new entities into the blockchain channel seamlessly.

Application Gateway (API Gateway). The Application Gateway bridges external client applications and the blockchain network by providing RESTful APIs. The gateway supports operations such as initializing the ledger, submitting transactions, and retrieving data securely. In Listing 7 is an example of how the gateway initializes services and connects to the blockchain. This code snippet is written in JavaScript and runs in a Node.js environment. The gateway uses the fabric-gateway library to interact with the blockchain network securely and efficiently.

Listing 7. API Gateway Services

```
async function startServer() {
  try {
    // Establish connection to the blockchain network
    await blockchainService.connectGateway();

    // Inject the blockchain contract into the ticketService
    ticketService.setContract(blockchainService.con-
tract);

    // Initialize the ledger (e.g., preload data)
    await ticketService.initLedger();

    // Start the Express server
    app.listen(PORT, () => {
      console.log(`Server is running on port ${PORT}`);
    });
  } catch (error) {
    console.error(`Failed to start the application: ${er-
ror}`);
    process.exit(1);
  }
}
```

Key functionalities provided with the implementation given in Listing 7 can be summarized as below:

1. Blockchain Connection:
   a. Establishes a secure connection to the Hyperledger Fabric blockchain network using the blockchainService.connectGateway() method.
2. Service Initialization:
   a. Injects the blockchain contract into a ticketService for managing blockchain transactions.
   b. Calls the ticketService.initLedger() method to initialize the blockchain ledger, potentially preloading essential data.
3. Server Setup:
   a. Starts an Express.js server to provide a RESTful API for external applications. The server listens on the specified port (PORT) and serves as the entry point for client requests.

Ticket Creation Using Standard Message Format. This process involves generating a new ticket asset on the blockchain network using a standardized message format. The example provided in Listing 8 highlights the practical use of these standardized methods to generate new ticket and membership assets securely, ensuring the blockchain ledger's integrity and immutability. This code snippet is written in TypeScript and is designed to run on the Hyperledger Fabric blockchain platform. It demonstrates the implementation of ticket creation and membership management using a standardized message format within a smart contract. The standardized format ensures consistent communication and interoperability between blockchain nodes, service providers, and client applications in the public transportation ecosystem.

Key functionalities provided with the implementation given in Listing 8 can be summarized as below:

1. Transaction Creation:
   a. The createTransaction function accepts transaction data, parses it into a TransactionRecord object, and validates its uniqueness by checking the blockchain ledger for existing transaction IDs.
   b. If the transaction ID is unique, it stores the transaction data on the blockchain ledger using ctx.stub.putState.
2. Membership Creation:
   a. The createMembership function allows the creation of a new membership for a user.
   b. It checks if a membership already exists for the provided user ID using the membershipExists method. If it does, an error is thrown to prevent duplication.
   c. The new membership data is saved to the blockchain ledger for future validation and use.

Importance of Standardized Message Format. The use of a standardized message format simplifies communication between various components of the system by:

- Ensuring uniform data structures for transactions and memberships.
- Reducing the complexity of data exchange between operators, service providers, and client applications.
- Supporting seamless integration and scalability across diverse public transportation scenarios.

Listing 8. Example Smart Contract Code for Transaction and Membership Creation

```
  @Transaction()public async createTransaction(ctx: Con-
text, transactionData: string): Promise<void> {
        // Parse transaction data from input
        const transaction: TransactionRecord =
JSON.parse(transactionData);
        // Ensure transaction ID is unique before storing
        const exists = await ctx.stub.getState(transac-
tion.transactionId);
        if (exists && exists.length > 0) {
            throw new Error(`Transaction with ID ${trans-
action.transactionId} already exists`);
        }
        // Save transaction data to the ledger
        await ctx.stub.putState(transaction.transac-
tionId, Buffer.from(stringify(transaction)));
    }

    @Transaction()public async createMembership(ctx: Con-
text, userID: string, membershipStatus: string): Prom-
ise<void> {
        // Create a new membership entry for a user
        const membership = new Membership(userID, member-
shipStatus);
        // Verify membership uniqueness
        const exists = await this.membershipExists(ctx,
userID);
        if (exists) {
            throw new Error(`Membership for user
${userID} already exists`);
        }
        // Store membership in the ledger
        await ctx.stub.putState(userID,
Buffer.from(JSON.stringify(membership)));
    }
```

4 Privacy-Preserving Digital Identities for Public Transportation

This section explains how decentralized technologies can help protect user privacy in public transportation. It also looks at the challenges in digital systems that affect personal data security and how blockchain technology can help protect user privacy. This study provides how these technologies can be used to improve privacy in public transportation.

4.1 Privacy Challenges in the Digital Era

In today's world, it is crucial to securely store and process personal data due to the significant impact of social media and digitalization. Despite existing laws and regulations to protect user data, data breaches still occur, highlighting significant vulnerabilities. Storing personal data on company or institution servers is often the main cause of security breaches [22]. This is because central storage of personal data can be vulnerable to attacks and single point of failure can make it susceptible to cyber-attacks and unauthorized access. Personal data must be stored securely, and access should be restricted to authorized personnel only.

4.2 Emerging Solutions for Data Protection

To tackle the challenges addressed in Sect. 4.1, digital identity solutions have been developed. These include the DID protocol [29], VC [30], and ZKP [31]. These technologies promote secure and digital storage of user data on individuals' devices, departing from traditional centralized systems. The DID protocol enables users to create and manage their digital identities without relying on central authorities. Verifiable Credentials enhance the dependability of digital identities and claims presented by users. They only include necessary data for a transaction. ZKP enables mathematical verification of information while maintaining privacy by not revealing personal details. The system is designed to utilize Polygon public chain nodes for issuing credentials and decentralized verification [32], integrating user wallet applications and verifier nodes to enable seamless, privacy-centric interactions.

4.3 Application in Public Transportation Systems

A unified architecture allows users to securely access various public transportation services. The proposed study is different from conventional systems because users do not have to repeatedly share sensitive personal information with each service provider. Instead, it uses Verifiable Credentials stored in digital wallets, backed by the DID protocol. The claim information is stored in a data structure called the Sparse Merkle Tree [33]. This ensures data integrity by updating the root hash value, which is critical for verifying proofs. The root hash value is stored immutably on the blockchain.

4.4 Implementing a Privacy-Preserving Approach

Users prove their eligibility to service providers by presenting ZKP-generated proofs alongside their credential information. These proofs can demonstrate eligibility as a student, teacher, elderly person, or person with a disability. Personal information is not required to be disclosed. Service providers verify these claims by referencing the proof and the root hash value on the blockchain. This model promotes a secure environment that minimizes personal data exposure and prioritizes privacy. Figure 4 shows how users can manage their digital identities securely and provide verification to service providers without revealing personal information. This is done by using technologies such as DID, VC, ZKP, and blockchain and explained step-by-step below

Step-by-Step Process.

1. DID Creation:

- Users generate a unique DID through their wallet application.

2. Credential Issuance:

- The user requests a credential from an issuer (e.g., proof of student status).
- The issuer validates the claim, generates a VC, and signs it.

3. Membership Creation:

- A transport network node verifies the user's eligibility (e.g., student or elder discount) during membership registration.
- The user's digital wallet generates a ZKP for the credential and shares it with the verifier, who is typically a transportation operator.
- Proof of credential is stored in membership creation transaction in Hyperledger blockchain

4. Proof Verification:

- The operator (verifier) requests proof of claim, DID and retrieves the current state from the blockchain to validate the transaction.
- It cross-checks the root hash stored on the blockchain to validate the proof.
- If valid, the transaction proceeds and is recorded on the blockchain.

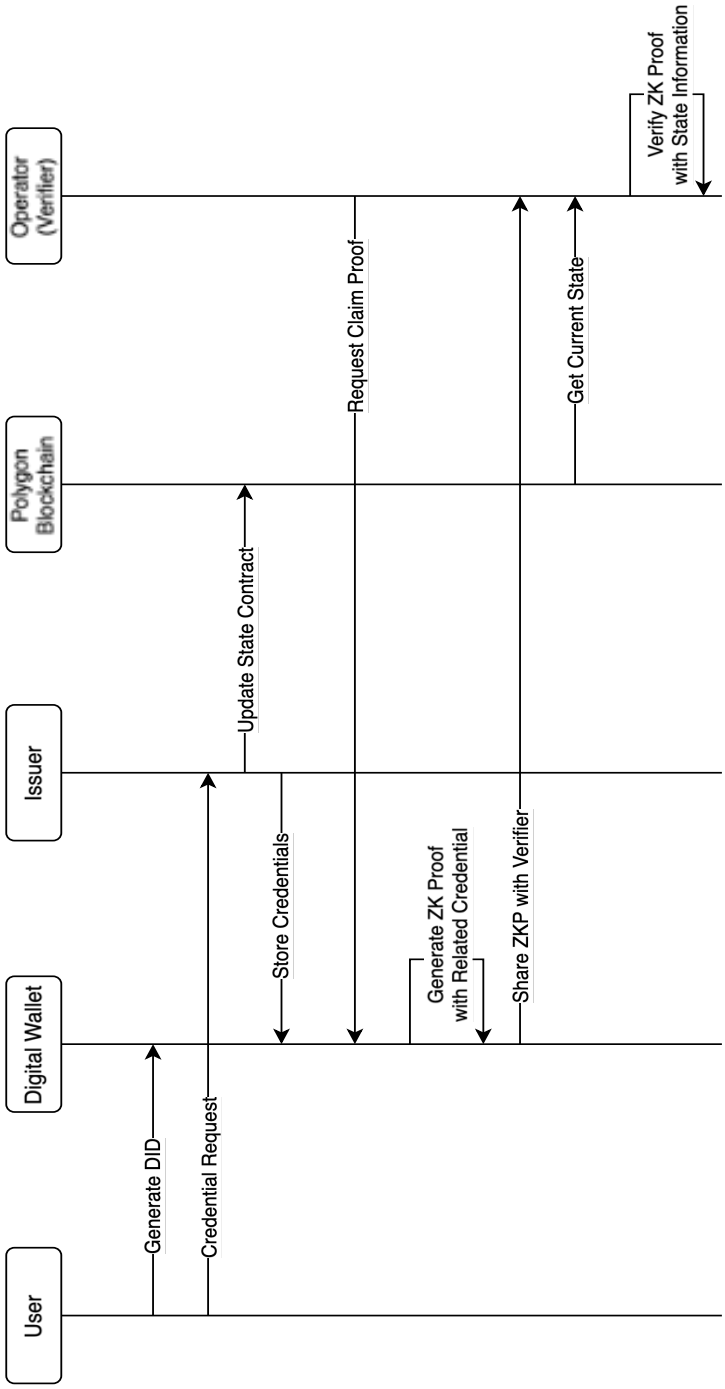

Fig. 4. User Privacy Protection in a Blockchain-Enabled Public Transportation System

4.5 A Sample Implementation: Proof Generation and Membership Creation

This section demonstrates a practical implementation of proof generation and membership creation processes within a blockchain-enabled public transportation network. The examples are implemented using TypeScript and leverage Hyperledger Fabric and Polygon for decentralized identity verification and transaction management.

Proof Generation on Digital Wallet. In the first step, a user's digital wallet generates a ZKP based on their credentials. This proof is used to validate claims (e.g., student or elder status) without exposing sensitive personal information. The implementation of proof generation is shown in Listing 9.

Key Highlights.

- The generateProof function:
  - Encodes credential data (cred_data) using Base64 for secure transport.
  - Retrieves the user's stored credentials and parses their DID.
  - Authenticates the user and generates a ZKP.
- The generated proof is returned as a secure object for use in the next steps.

Listing 9. Identity Controller Sample Code for Proof Generation on Digital Wallet

```
import {Base64} from 'js-base64';
import { DID } from '@iden3/js-iden3-core';

export default class IdentityController {
generateProof = async (cred_data) => {
  try {
    const url_param = Base64.en-
code(JSON.stringify(cred_data))
    const credentials = await this.credentialWal-
let.list();
    const _did = DID.parse(this.identityDID);
    const msgBytes = base64ToBytes(url_param);
    const authRes = await this.AuthHandler.handleAuthori-
zationRequest(_did, msgBytes);
    return new ReturnObj(ReturnObj.OK, 'Proof Generated
Successfully', authRes);
  } catch (error) {
    return new ReturnObj(ReturnObj.ERR, error, null);
  }
}
```

Membership Creation with User Proof.

Once the proof is generated, it is used during the membership creation process. The transportation network node verifies the user's eligibility using their ZKP and stores the validated membership details in the blockchain ledger. The membership record is structured using a standardized message format, as shown in Listing 10.

Key Highlights.

- The membershipDetails object contains:
  - A unique membershipId.
  - The validity period (validity).
  - Proof of Decentralized Identity (proofOfDID) from the user.
  - The DIDs of both the proof owner and the verifying node.
- The record is securely stored on the Hyperledger Fabric ledger for use in future verifications.

Listing 10. Membership Creation with User Proof

```
"membershipDetails": {
  "membershipId": "MembershipID",
  "validity": "MembershipValidity"
  "proofOfDID": "eyJhbGciOiJncm90aDE2IiwiY2lyY3VpdElkI-
joiYXV0aFYyIiwiY3JpdCI6WyJjaXJjdWl0SWQiXSwidHl-
wIjoiYXBwbGljYXRpb2…"
  "proofOwnerDID": "did:iden3:poly-
gon:amoy:x6x5sor7zpyAJu49Nch4Lixzawx1qd95sjtwnb2Uf"
  "verifierDID": "did:iden3:poly-
gon:amoy:2qa3Ni16rkJ4BSvK4M3u8FMmjcMyKbqn7kNNKSjeMs"
}
```

4.6 Outlook for Privacy Enhancement

The adoption of DIDs, VCs, and ZKPs represents a significant step forward in personal data protection and privacy. Self-Sovereign Identity (SSI) frameworks powered by blockchain have the potential to decentralize control of digital identities, empowering individuals to manage their data securely and privately without reliance on centralized authorities [34].

This study highlights the importance of these technologies within the system's architecture, supported by an infrastructure that ensures secure verification of credentials without exposing sensitive user information. By adopting open standards such as W3C's Verifiable Credentials and Decentralized Identifiers, the system ensures interoperability across networks while maintaining user privacy. The development of SSI systems will require collaboration between technologists, policymakers, and industry stakeholders. This collaboration is essential to address challenges such as governance

models, usability, and wider adoption. The integration of privacy-preserving technologies like ZKPs into the public transportation infrastructure lays the groundwork for a user-centric ecosystem

5 Conclusion

5.1 Summary of Contributions

This paper presents a blockchain-based infrastructure designed to address key challenges in the public transportation sector, including interoperability, privacy, and transactional transparency. By leveraging a private blockchain, the system facilitates seamless integration of diverse transportation providers, ensuring secure and efficient transaction processing through consensus mechanisms. Transportation providers participate as network nodes, enabling decentralized collaboration and trust.

Key aspects of the proposed infrastructure include:

1. Interoperability: A standardized message format ensures that diverse providers can operate on a unified platform, reducing complexity for both users and operators.

2. Privacy Protection: The integration of ZKPs and DIDs protects sensitive user information while enabling secure and verifiable transactions.

3. Smart Contract Automation: APIs and smart contracts enable autonomous agreements, extending functionality to include real-time fare adjustments, ticket management, and membership validations across the network.

Use of Dual-Chain Architecture. The system employs a hybrid dual-chain architecture by integrating a private Hyperledger Fabric blockchain with a public blockchain. The private blockchain enhances transaction processing speed and efficiency, making it suitable for the operational needs of transportation networks. Meanwhile, the public blockchain ensures privacy-preserving functionalities, such as credential verification across multiple issuers, enabling secure and decentralized identity management. This combination leverages the strengths of both blockchain types to create a robust and efficient system.

5.2 Planned Improvements.

Future development efforts will focus on real-world testing by utilizing large datasets that simulate various transportation scenarios, such as buses, trams, and scooters, to evaluate system performance under increased transaction volumes and diverse network conditions. A dedicated revenue distribution report tool will be developed to facilitate fair profit-sharing among operators based on blockchain-stored transaction data. Additionally, the user wallet will be enhanced to improve usability and security, integrating DIDs, VCs, and ZKP generation for privacy-preserving transactions. To support

broader interoperability, the development of issuer nodes on a public blockchain, such as Polygon, will be prioritized, enabling distributed credential issuance and efficient verification. Furthermore, we aim at benefitting from the results of our previous research (e.g. [35, 36]) on model-driven engineering of public transportation systems to supporting the design and implementation of the public transportation applications considering the digital identity and transaction verification architecture introduced in this paper.

Acknowledgments. Special acknowledgment is due to Kentkart A.Ş. Company for their generous support and guidance during the preparation of this paper.

References

1. Oeschger, G., Carroll, P., Caulfield, B.: Micromobility and public transport integration: The current state of knowledge. In: Transportation Research Part D: Transport and Environment (2020). doi: 10.1016/j.trd.2020.102628
2. Arslan, S. and Kardas, G. (2023) "Modeling Internet of Things software for public transformation", Journal of Intelligent Transportation Systems and Applications, vol. 6, no. 2, pp. 425-445, DOI: 10.51513/jitsa.1328020.
3. Aydin, M. B., Oz, C., Cetin Tulazoglu, D. and Kardas, G. (2019) "Development of an ITxPT compliant information system for public transportation vehicles", Journal of Intelligent Transportation Systems and Applications, vol. 2, no. 2, pp. 1-13.
4. ABT Kentkart: Automated fare collection system. In: Kentkart (2022). https://www.kentkart.com/fare-collection-system (Accessed: December 14, 2024)
5. STIB-MIVB: Ticket information. In: STIB-MIVB Ticket (2022). https://www.stib-mivb.be/article.html?l=en&_guid=80bb5be7-429c-3810-a795-dfe836d62585 (Accessed: December 14, 2024)
6. MVV: Online and handy ticket. In: MVV Ticketing (2022). https://www.mvv-muenchen.de/en/tickets-and-fares/online-und-handyticket/index.html (Accessed: December 14, 2024)
7. Whim: MaaS Global, Whim (2019). https://maasification.com/applications/by-application/whim-maas-global/ (Accessed: December 14, 2024)
8. Kazi, S., Bagasrawala, M., Shaikh, F., Sayyed, A.: Smart e-ticketing system for public transport bus. In: Proceedings of the 2018 International Conference on Smart City and Emerging Technology (ICSCET), pp. 1-7 (2018). doi: 10.1109/ICSCET.2018.8537302
9. Khedekar, T., Jamdar, V., Waghmare, S., Dhore, M.L.: FID automatic bus ticketing system. In: Proceedings of the 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), pp. 1-6 (2021). doi: 10.1109/AIMV53313.2021.9670957
10. Pasquale, G.D., Bie, J.D., Singh, J.: Ticketing in Mobility as a Service. In: International Association of Public Transport (UITP) (2022). https://cms.uitp.org/wp/wp-content/uploads/2022/07/Report-Ticketing-MaaS-JULY2022-web.pdf (Accessed: December 14, 2024)
11. Arslan, S., Kardas, G. and Alfraihi, H. (2024) "On the usability of a modeling language for IoT-based public transportation systems", Applied Sciences, vol. 14, no. 13, 5619, pp. 1-30, DOI: 10.3390/app14135619.
12. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. In: Technical Report (2008). https://bitcoin.org/bitcoin.pdf (Accessed: December 14, 2024)

13. Kakavand, H., Kost De Sevres, N., Chilton, B.: The blockchain revolution: An analysis of regulation and technology related to distributed ledger technologies. In: SSRN Electronic Journal (2016). doi: 10.2139/ssrn.2849251
14. Nath, I.: Data exchange platform to fight insurance fraud on blockchain. In: Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pp. 821–825 (2016).
15. Gupta, S., Sinha, S., Bhushan, B.: Emergence of blockchain technology: Fundamentals, working and its various implementations. In: Proceedings of the International Conference on Innovative Computing & Communications (ICICC) (2020). doi: 10.2139/ssrn.3569577
16. Saritas, H.B., Kardas, G., "A Blockchain-based Transaction Verification Infrastructure in Public Transportation," 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS), Belgrade, Serbia, 2024, pp. 169-176, doi: 10.15439/2024F5274.
17. Jayalath, S.A., Rajapakse, C., Senanayake, J.M.D.: A micro-transaction model based on blockchain technology to improve service levels in the public transport sector in Sri Lanka. In: Proceedings of the 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE), pp. 82-89 (2020). doi: 10.1109/SCSE49731.2020.9313037
18. Wang, G., Nixon, M.: InterTrust: Towards an efficient blockchain interoperability architecture with trusted services. In: Proceedings of the 2021 IEEE International Conference on Blockchain, Melbourne, Australia, pp. 150-159 (2021). doi: 10.1109/Blockchain53845.2021.00029
19. Yang, T., Cui, Z., Alshehri, A.H., Wang, M., Gao, K., Yu, K.: Distributed maritime transport communication system with reliability and safety based on blockchain and edge computing. In: IEEE Transactions on Intelligent Transportation Systems (2022). doi: 10.1109/TITS.2022.3157858
20. Enescu, F.M., Bizon, N., Serban, G., Hoarcă, I.C.: Environmental protection - Blockchain solutions for intelligent passenger transportation of persons. In: Proceedings of the 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1-6 (2021). doi: 10.1109/ECAI52376.2021.9515026
21. Jabbar, Y., Dhib, E., Said, A.B., Krichen, M., Fetais, N., Zaidan, E., Barkaoui, K.: Blockchain technology for intelligent transportation systems: A systematic literature review. In: IEEE Access, vol. 10, pp. 20995-21031 (2022). doi: 10.1109/ACCESS.2022.3149958
22. Ganzha, M., Maciaszek, L., Paprzycki, M., Ślęzak, D. (eds.): Proceedings of the 17th Conference on Computer Science and Intelligence Systems, ACSIS, vol. 30, pp. 685–694 (2022)
23. Karatas, R., Sertkaya, I., (2020). Self Sovereign Identity based E-petition Scheme (Vol. 9). Vol. 9. International Journal of Information Security Science.
24. Chen, Y., Gu, J., Chen, S., Huang, S., Wang, X.S.: A full-spectrum blockchain-as-a-service for business collaboration. In: Proceedings of the 2019 IEEE International Conference on Web Services (ICWS), pp. 219-223 (2019). doi: 10.1109/ICWS.2019.00045
25. Hyperledger Foundation: Hyperledger Fabric Documentation. In: Hyperledger Wiki (2023). https://hyperledger-fabric.readthedocs.io/ (Accessed: December 14, 2024)
26. Reddy, B., Aithal, P.S.: Blockchain based service: A case study on IBM blockchain services & Hyperledger Fabric. In: International Journal of Case Studies in Business, IT, and Education (IJCSBE), vol. 4, no. 1, pp. 94-102 (2020). doi: 10.2139/ssrn.3611876
27. Ongaro, D.: In search of an understandable consensus algorithm (Extended Version). In: Stanford University (2014). https://raft.github.io/raft.pdf (Accessed: December 14, 2024)
28. Awati, R.: Consensus algorithm. In: TechTarget. https://www.techtarget.com/whatis/definition/consensus-algorithm (Accessed: December 14, 2024)
29. W3C: Decentralized Identifiers (DIDs) v1.0. In: W3C Recommendation (2022). https://www.w3.org/TR/did-core/ (Accessed: December 14, 2024)

30. W3C: Verifiable Credentials Data Model v2.0. In: World Wide Web Consortium. https://www.w3.org/TR/vc-data-model/ (Accessed: December 14, 2024)
31. Hyperledger Foundation: Hyperledger AnonCreds: Anonymous Credentials with Zero-Knowledge Proofs. In: Hyperledger Wiki. https://wiki.hyperledger.org/display/anoncreds (Accessed: December 14, 2024)
32. Polygon ID: Zero Knowledge Identity for Web3: Polygon (2022). https://polygon.technology/blog/introducing-polygon-id-zero-knowledge-own-your-identity-for-web3 (Accessed: December 14, 2024)
33. Haider, F.: Compact Sparse Merkle Trees. (2018). doi: 10.31219/osf.io/8mcnh
34. Sherriff, A., Young, K., Shea, M.: Editorial: Establishing Self Sovereign Identity with Blockchain. In: Front. Blockchain, vol. 5, Art. no. 955868 (2022). doi: 10.3389/fbloc.2022.955868
35. Saritas, H. B. and Kardas, G. (2014) “A model driven architecture for the development of smart card software”, Computer Languages, Systems & Structures, vol. 40, no. 2, pp. 53-72, DOI: 10.1016/j.cl.2014.02.001.
36. Arslan, S. and Kardas, G. (2020) “DSML4DT: A domain-specific modeling language for device tree software”, Computers in Industry, vol. 115, 103179, pp. 1-13, DOI: 10.1016/j.compind.2019.103179