IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 December 2023, accepted 6 January 2024, date of publication 11 January 2024, date of current version 22 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3352738

RESEARCH ARTICLE

Blockchain-Based Framework for Traffic Event Verification in Smart Vehicles

FRANCISCO A. PUJOL[1], HIGINIO MORA[1], TAMAI RAMÍREZ[1], CARLOS ROCAMORA[1], AND ARTURO BEDÓN[2]

[1]Department of Computer Technology and Computation, University of Alicante, 03690 Alicante, Spain
[2]Facultad de Ingeniería y Ciencias Aplicadas, Escuela de Sistemas de Información, Universidad Central del Ecuador, Quito 170129, Ecuador

Corresponding author: Francisco A. Pujol (fpujol@ua.es)

This work was supported by the Spanish Research Agency (AEI) (DOI: 10.13039/501100011033) under Project High Performance Computing (HPC)4Industry under Grant PID2020-120213RB-I00.

ABSTRACT The development of smart vehicles has been a major focus of the automotive industry in recent years. Smart vehicles, equipped with advanced sensors and communication technologies, represent a transformative paradigm in modern transportation systems. Some of the challenges associated with the introduction of smart vehicles include developing reliable sensors, creating robust communication networks, and ensuring the security of vehicle systems. This paper proposes a Blockchain-based framework for accident prevention on the Internet of Vehicles, where vehicles monitor the state of the route and transmit information about hazardous situations to the Blockchain network. Once the event is confirmed, warnings are sent to all vehicles, and their speed is automatically reduced to avoid accidents. The tests in a 3D graphical simulator, combined with Hyperledger Besu technology for the creation of the Blockchain network, demonstrated both horizontal and vertical scalability, validating the potential of this framework for real-world integration. Moreover, it presents a fast reaction to anomalous situations on a route compared to human reactions under similar circumstances.

INDEX TERMS Blockchain, Hyperledger Besu, IBFT 2.0, security, smart contracts, smart vehicles.

I. INTRODUCTION

Smart vehicles represent a rapidly evolving technology that is ready to transform the transportation sector. These vehicles are equipped with state-of-the-art sensors, cameras, and other cutting-edge technologies, allowing fluid communication between them and the infrastructure, resulting in safer and more efficient operations. Beyond their safety improvements, smart vehicles are believed to be capable of reducing traffic congestion, having lower emissions, and optimizing fuel efficiency [1]. Furthermore, they are expected to provide a more comfortable and convenient driving experience for passengers. Despite these promising features, the integration of smart vehicles into mainstream use faces several challenges. These challenges include cybersecurity, data privacy concerns, and the need for comprehensive infrastructure development. Despite these challenges, the future of smart vehicles appears promising and is expected to play an important role in shaping the future of transportation [2].

The rise of smart vehicles and the growing complexity of modern traffic systems have required the development of reliable and secure solutions to monitor and verify critical incident information. Advanced traffic control systems improve the quality and efficiency of road services by providing accurate and current information from a variety of sources, including sensors, smart cameras, warning messages, traffic signals, and road meteorological systems. Without this traffic intelligence, it would be impossible to implement the necessary modifications to the network, incorporate new modes of transportation, and promote infrastructure that addresses current and future transport requirements [3].

To this end, decentralized technologies, such as Blockchain, have been identified as a potential framework to improve the reliability, security, and transparency of data exchange in smart vehicle environments. Blockchain is a technology

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro R. M. Inácio.

that enables the creation of secure decentralized networks for the exchange of information [4]. However, it is still difficult to use this technology for vehicle-to-vehicle communication due to the demanding resource needs and the complex integration with existing systems. Current smart vehicles use other communication technologies, such as 5G or 802.11p, which offer increased speed, latency, and reliability [5]. Blockchain may have other applications in the field of electric mobility, such as payment for recharging, buying, and selling vehicles, or generating synthetic data, but it has not yet been integrated into real vehicle-to-vehicle communications [6].

This paper presents a new approach that utilizes decentralized Blockchain technology to address the difficulties related to the transmission and authentication of anomaly events on roads (accidents, road works, bad weather, etc.) in the context of smart vehicles. In comparison to traditional distributed systems, Blockchain provides enhanced security, transparency, and decentralized consensus, crucial for managing such critical information in a vehicular network. Despite the acknowledged complexities in implementing Blockchain in a vehicular network, Blockchain brings unique advantages to the IoV, including increased trust, transparency, and a decentralized approach.

A private Blockchain network will be used to validate the information provided by vehicles in the event of the detection of an anomaly with their sensors. The vehicles will be the nodes of this network and will be responsible for sending information to the network about any incident. Once the information has been validated and verified correctly, a warning will be sent to the rest of the vehicles on the network so that drivers can take the actions they consider appropriate, such as slowing down or performing emergency braking. This Blockchain-based system will perform a secure evaluation of the information provided by the vehicles, ensuring that no action can be executed unless the information has been validated. This precaution is crucial to mitigate the risk of errors in the vehicle's system or potential network attacks. By decentralizing the infrastructure and incorporating Blockchain principles, our approach enables a secure and efficient framework for smart vehicles to interact with each other. This paper outlines the key components of our decentralized Blockchain-based approach, emphasizing its potential to manage event information in smart vehicle networks.

The main contributions of this work are:

- The use of the Hyperledger Besu framework for developing a private Blockchain network within the domain of smart vehicles.
- A scalability analysis of the proposed private framework for integration into a real-world scenario.
- The integration of the framework into a 3D simulator to test its performance.

To our knowledge, there are no previous works that have explored the intersection of smart vehicle technology, Blockchain-based traffic monitoring in the Internet of Vehicles (IoV), and comprehensive scalability testing conducted in a 3D graphical simulator, making this research a novel and significant contribution to this field.

This research is motivated by the need for a reliable framework for the verification of traffic events in smart vehicles, highlighting the lack of experimental studies to integrate Blockchain-based approaches into traffic monitoring. The proposed framework demonstrates both horizontal and vertical scalability, offering a rapid response to anomalous situations on a route compared to human reactions. These findings indicate the potential for real-world integration and suggest the viability of incorporating the system into practical scenarios, marking a novel and substantial contribution to the field.

The remainder of this paper is organized as follows. Section II presents an overview of Blockchain technology and the review of the literature on smart vehicles. Then the development of the Blockchain-based approach is analyzed in Section III. In Section IV, the smart contracts employed in the Blockchain framework are explained. The results obtained from the experiments and a comparison with other works are shown in Section V, and some concluding remarks are outlined in Section VI.

II. OVERVIEW OF BLOCKCHAIN TECHNOLOGY AND ITS APPLICATION TO SMART VEHICLES

A. BLOCKCHAIN: FUNDAMENTALS AND APPLICATIONS

Blockchain is a paradigm for maintaining information in a distributed system characterized by several key properties such as decentralization, immutability, and transparency. This technology is designed for secure and transparent storage and transfer of digital assets. The system consists of multiple layers, such as hardware, data storage, and communication layers. It offers decentralization, resistance to tampering, and uses cryptography for data integrity [7].

A Blockchain network is a distributed ledger system (DLS) in which user-represented computer processes collaborate on distributed ledger data structures. This network consists of nodes responsible for maintaining and validating the Blockchain, and they communicate to achieve consensus. The immutability in Blockchain is an intrinsic property of the technology. Once a block is added to the Blockchain, it cannot be modified or deleted without invalidating all subsequent blocks. This property can be challenged by agents with sufficient computing power, known as Mutable-By-Hashing-Power. Integrity ensures robustness, but not immunity to modification. The achievement of the immutability of the ledger is a desired but evolving goal, which warrants further research into specific conditions and implementation parameters [8]. Transactions on Blockchain are grouped into blocks, each linked to the previous one through a unique hash. Consensus mechanisms, such as Proof of Work (PoW) or Proof of Stake (PoS), validate and agree on the transaction validity. Once added, the blocks become immutable, preserving data integrity. Smart contracts, running on Blockchain, automate and enforce agreements without intermediaries, following predefined rules and conditions. They have the

potential to revolutionize industries by automating processes, reducing costs, and increasing efficiency [9].

Blockchain technology has found applications in many different areas, including transportation, commerce, privacy, finance, government, education, healthcare, and the Internet of Things (IoT), among others. Research in recent years has focused mainly on financial management and security, but new developments can be found in educational, healthcare, IoT and government applications [10]. The utilization of Blockchain technology, especially in the form of digital currency such as Bitcoin, has shown its capacity to have a significant influence on various domains. The initial implementation of Blockchain through Bitcoin serves as a prime example, effectively showcasing the extensive advantages and implications associated with this innovative technology, which have been thoroughly documented [11].

Blockchain's application to the IoV faces significant challenges [12]. First, achieving true decentralization is complicated due to the centralized nature and management of existing IoV infrastructure. Cybersecurity is of utmost importance to protect sensitive data exchanged in the IoV, and while Blockchain offers potential solutions, robust security mechanisms and defense against emerging threats are necessary. Data privacy, a critical concern, involves managing vast amounts of data generated by the IoV while complying with regulations like the General Data Protection Regulation (GDPR). Building trust among IoV ecosystem participants is vital, and Blockchain's transparent and immutable records can help, but challenges exist in ensuring data integrity, verifying participant identities, and addressing trust-related issues in smart contracts and consensus mechanisms. Scalability problems arise due to the IoV's large vehicle volume and real-time data needs, making it necessary to explore solutions like off-chain transactions and sharding to handle high transaction volumes and minimize latency [13]. Some of the literature related to the application of Blockchain to smart vehicles will be discussed next.

B. RELATED WORK

Blockchain technology can improve the security and safety of smart or autonomous vehicles (AVs) by preventing cyber-attacks and ensuring trust, data integrity, and quick access through a distributed system. It eliminates single points of failure, allows activity tracking, and enforces accountability in AV systems. By integrating Blockchain, AVs can achieve higher security, privacy, and service availability, addressing the limitations of centralized architectures for a decentralized and resilient solution [14]. The Toyota Research Institute is actively working to integrate Blockchain with autonomous vehicles with the aim of enhancing transparency, traceability, and trust in various applications, including data sharing, ride-sharing, transactions, and insurance. The immutability and transparency of the Blockchain offer benefits to the automotive industry by improving stakeholder interaction and securing connected and autonomous vehicles through activity tracking and record keeping [15]. Blockchain also improves security and transparency for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, improving road safety and efficiency. It also plays a crucial role in securing software updates and over-the-air diagnostics for autonomous vehicles, ensuring the integrity and authenticity of these updates [16], [17], [18].

In [19], there is a Blockchain-enabled IoV system that provides secure, lightweight, and decentralized peer-to-peer vehicular communication. The authors demonstrated how Blockchain technology can be used to ensure the authenticity of messages in a decentralized way and how it can help to solve issues related to security and privacy in IoV. They tested the proposed system by conducting simulations and experiments to evaluate its performance in terms of message dissemination, security, and privacy.

Singh et al. [20] discussed a Blockchain-based Secure Storage Architecture for the Industrial Internet of Vehicles Things (IIoVT), which integrates Blockchain technology into various layers of the system. It used Blockchain for secure communication at the coordination layer, introduced a Distributed Hash Table (DHT) for decentralized storage at the cloud layer, and provided intelligent services at the application layer. This integration improved security by offering secure communication, decentralized storage, and intelligent services. The proposed scheme was found to be superior to existing studies based on security and data storage cost evaluations.

In [21], the authors proposed a Blockchain-based strategy to improve the security of autonomous vehicles and reduce the risk of accidents. The focus is on validating data from accident-detecting nodes, which is essential for accurate accident detection and prevention. Using Blockchain, the proposed method increases the reliability of accident data, which is essential for making informed decisions and taking preventive measures. Furthermore, the paper suggests integrating artificial intelligence and machine learning to further enhance the accuracy of accident detection and prevention within the Blockchain-based system.

On the other hand, Blockchain technology provides a secure way for vehicles to communicate with each other, ensuring data integrity, authentication, decentralization, smart contract automation, privacy protection, traceability, and consensus mechanisms for reliability. This ensures secure communication and data transmission in intelligent vehicle networks [22]. This is achieved by ensuring trust, data accuracy, security, and privacy. Trust is established through transaction validation and chaining, preserving data integrity. Decentralization and cryptographic techniques reinforce security, while consensus mechanisms prevent unauthorized changes. Smart contracts and Merkle trees contribute to controlled access and efficient data sharing. Consequently, it improves security and privacy in the context of intelligent transportation systems and the IoV [23], [24], [25], [26].

In addition, a Blockchain-based framework for securing smart vehicles (B-FERL) was proposed in [24] to ensure secure data transmission and validation in connected vehicle systems. B-FERL used permissioned Blockchain technology, challenge-response data exchange, and authentication to improve communication and data security. The authors validated its effectiveness through qualitative and quantitative evaluations in an simulated scenario and comparative evaluations against other proposals for vehicular network security. B-FERL demonstrated suitable response times and storage efficiency, outperforming existing solutions by introducing the appendable block concept and improving data transmission and validation scalability in connected autonomous vehicles.

Researchers are also working on combining Artificial Intelligence and Blockchain [27]. In [28], authors presented the integration of Federated Learning (FL) and Blockchain in vehicular networks to address challenges in privacy, big data handling, computational cost and data integrity within the Smart Transport Infrastructure (STI). The applications of FL and Blockchain in Vehicular Ad Hoc Networks (VANETs) were examined, emphasizing privacy preservation and big data management. It was discussed using Blockchain as a distributed ledger to improve security and trust while storing data in the STI. This integration aims to improve privacy, security, and data handling in the STI era. Finally, the main novelty of the article in [29] is the exploration of the role of Blockchain in improving Federated Learning by addressing coordination challenges in a decentralized way. The study focused on using Blockchain to ensure the verifiable integrity of client data and the correctness of training data in a FL system for fault detection in IIoT. To secure data transmission and validation, the work proposed client-level differential privacy, which involved adding noise to client updates both locally and on the Blockchain. Additionally, the global model summary on the Blockchain can be encrypted, with decryption keys held only by participating clients, protecting it from public attacks.

Based on previous works, Blockchain technology can provide a secure and transparent framework for storing and accessing vehicle data, ensuring its integrity and preventing tampering or manipulation. The use of Blockchain can also address the issue of data authenticity, as the decentralized nature of the technology ensures that data are verified and validated by multiple nodes in the network. In this paper, the proposed approach aims to validate the data acquired from vehicles that detect anomalies in traffic conditions on a road, which is crucial for the accurate detection of dangerous situations on the roads and the prevention of their consequences. Using Blockchain, the proposed approach can improve the reliability and trustworthiness of the data, which is essential for making informed decisions and taking appropriate preventive measures. The Table 1 summarizes the key contributions of each reviewed survey.

III. BLOCKCHAIN-BASED FRAMEWORK FOR MONITORING HAZARDOUS SITUATIONS ON ROADS

A. DEFINITION OF THE BLOCKCHAIN NETWORK

In the context of our work, consider a set of R smart vehicles operating on a specific route. The main objective is to establish a framework for modeling potentially dangerous situations that can occur on the road. In this framework, smart vehicles themselves play an active role in detecting and reporting such events, thus becoming integral components of the definition of the state of the road. It is important to note that this set of vehicles will operate within a private Blockchain network. Access to this network is restricted to previously authorized nodes and only these authorized nodes possess the ability to modify the state of the road, making it a permissioned private network. Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ be the set of N vehicles in this private Blockchain network. These N vehicles will be those that can potentially cause a change in the state of the road section being considered. In the case that one of the vehicles v_i, identifies a new event, it will transmit the relevant data to the Blockchain network. Subsequently, the remaining vehicles within the set $\mathcal{V}$ assume the responsibility of validating this information.

Consider the implementation of an Application Programming Interface (API) that controls requests that come exclusively from authorized network nodes. The API works as an intermediate layer, enabling communication between the Blockchain network and the smart contracts deployed within it, for the N vehicles that make up the system to be controlled. Additionally, it interfaces with additional components, including a real-time database that contains essential network maintenance data and a data manager. The data manager is responsible for conducting real-time data validation tasks within the Blockchain network and also provides the capability to monitor the states of all routes. A visual representation of the developed system is shown in Fig. 1.

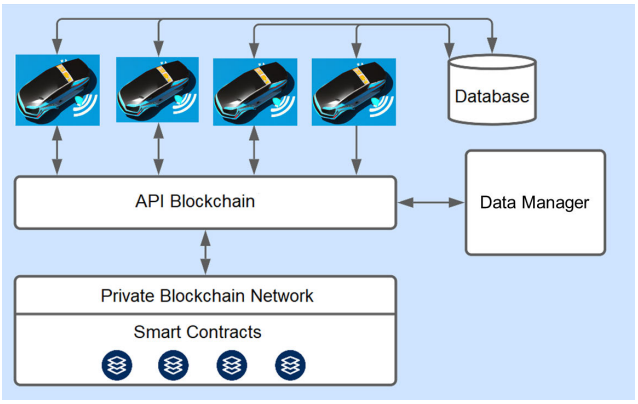

FIGURE 1. Overview of the proposed Blockchain Framework.

For the creation of the Blockchain network, Hyperledger Besu technology has been used, which allows it to be

TABLE 1. Overview of related works.

Title	Contributions
Blockchain-based security attack resilience schemes for autonomous vehicles in industry 4.0: A systematic review [14].	The authors systematically addressed security issues in traditional AV systems, explored Blockchain technology for enhanced security, categorized potential attacks based on objectives and cyber-security aspects, provided insights into AVs, Blockchain technology, and market size, and delved into the specifics of authentication attacks on AVs.
Integration of Blockchain with Connected and Autonomous Vehicles: Vision and Challenge [15].	Connected and Autonomous Vehicle (CAV) technology significantly improves the transportation system, promoting economic growth. Privacy challenges associated with transportation data, particularly issues resulting from centralized storage and a lack of access control, can be effectively addressed through the integration of Blockchain technology into transportation systems.
Investigating the Prospect of Leveraging Blockchain and Machine Learning to Secure Vehicular Networks: A Survey [16].	Proposed BFT based consensus protocol for reaching agreement in faulty systems.
Blockchain for the Internet of Vehicles Towards Intelligent Transportation Systems: A Survey [17]	This paper presented a comprehensive survey of techniques for integrating Blockchain and the IoV. It analyzed the challenges and motivations behind the integration of Blockchain with IoV, conducting an in-depth analysis of Blockchain adoption to improve vehicular data security, vehicle management, and on-demand transportation services.
Securing Vehicular Network Using AI and Blockchain-Based Approaches [18]	This work focused on the security and data-related issues within vehicular networks, exploring AI and Blockchain-based solutions to enhance vehicular network security.
Blockchain-Enabled Security and Privacy for Internet-of-Vehicles [19].	The authors conducted a comparison of Blockchain types and consensus algorithms applicable to vehicular applications. They introduced a novel voting-based consensus algorithm designed to incentivize vehicles for efficient message dissemination.
BIIoVT: Blockchain-Based Secure Storage Architecture for Intelligent Internet of Vehicular Things [20]	This paper introduced BIIoVT, a novel Blockchain-based Secure Storage Architecture for the IIoVT. The proposed architecture leveraged Blockchain for distributed secure communication at the coordination layer across various vehicular networks.
Lightweight-BIoV: Blockchain Distributed Ledger Technology (BDLT) for IoVs [21]	This research project focused on creating a custom consensus protocol for Hyperledger that could be used for proof-of-work investigations. The architecture was tested by simulating the exchange of data between connected devices in the IIoV. The results showed a 37.21% decrease in computing power costs and a 56.33% increase in node production and transmission effectiveness.
Introduce Reward–Based Intelligent Vehicles Communication Using Blockchain [23]	The authors introduced the Intelligent Vehicle-Trust Point (IV-TP) mechanism as a solution for ensuring security, reliability, and trustworthiness in Intelligent Vehicle (IV) communication. IV-TP was designed to validate vehicle behavior, distinguish between legal and illegal actions, and implement a reward-based system to incentivize successful communication among Intelligent Vehicles (IVs).
B-FERL: Blockchain based framework for securing smart vehicles [24]	This work proposes the Blockchain-based Framework for securing smart vehicLes (B-FERL), which adapted information access, utilized challenge-response data exchange for internal state monitoring, enabled authentic communication in vehicular networks, ensured resilience against identified attacks, and achieved trust management, vehicular forensics, and secure vehicular networks with suitable response time and storage size.
Blockchain Technology for Intelligent Transportation Systems: A Systematic Literature Review [25]	The purpose of Intelligent Transportation Systems (ITS) is to mitigate traffic problems and enhance traffic efficiency by ensuring compatibility and interoperability among IoV entities using various service providers. The integration of Blockchain technology with IoV introduces significant benefits and opportunities, which contribute to the development of IoV and the realization of the full potentials of ITS.
Autonomous Vehicles Security: Challenges and Solutions Using Blockchain and Artificial Intelligence [27]	The review explains existing security and privacy solutions for AVs that use AI and blockchain, analyzing the feasibility of their combined application. Additionally, the paper addressed open issues and challenges associated with the integration of Blockchain and AI into AVs, concluding with an outline of future research directions in this field.
Integration of Blockchain Technology and Federated Learning in Vehicular (IoT) Networks: A Comprehensive Survey [28]	A comprehensive survey on integrating Blockchain and FL in vehicular networks.
Blockchain-empowered Federated Learning: Challenges, Solutions, and Future Directions [29]	This comprehensive survey analyzed the challenges, solutions and future directions of BlockFed, examining critical issues within Federated Learning and illustrating how Blockchain can address these challenges.

developed over the Ethereum network through the so-called Ethereum Virtual Machine (EVM). Hyperledger Besu is an Ethereum client suitable for both public and private permissioned networks. It can be used on test networks such as Rinkeby, Ropsten, and Gorli [30]. This client offers a variety of consensus algorithms, including Proof-of-Work (PoW) and Proof-of-Assignment (PoA). It also has comprehensive permissioning systems that are designed for consortiums [31]. It has become increasingly popular in recent times due to its flexibility, enhanced privacy, and superior performance when constructing enterprise-level decentralized applications [32].

To configure a private Blockchain network using Hyperledger Besu, a configuration file, commonly referred to as the *genesis file* [33], is required. It is developed in JavaScript Object Notation (JSON) format. The genesis file assumes a pivotal role by defining the initial block of the Blockchain, which, in turn, determines the chain to which it is to be linked. Essentially, this file encapsulates the essential instructions required to build the network. Within the genesis file, the necessary parameters are configured to ensure that all network blocks contain a synchronized copy of the Blockchain, and that the data written by the validator nodes can be properly authenticated.

The parameters configured in this file include, among others:

- Consensus Algorithm: This parameter defines which consensus algorithm is used by the nodes to validate the information.
- Gas Limit: It is the total gas limit for all transactions included in a block. It defines how large the block size can be for the block, and is represented by a hexadecimal string.
- Epoch length: The number of blocks required for the reset of all votes. Votes refer to validators who vote to add or remove validators from the network.
- Timestamp: Date and time of block creation.

Network nodes, which are the vehicles of the system, will be able to perform both private and free transactions. In other words, even if a vehicle is part of the network, it will only be able to execute actions on a transaction if the transaction has been specified to be visible to that particular node. However, this does not imply that the vehicle loses its authority to validate transactions on the network. These features are achieved through the use of the Tessera privacy manager. Tessera is a stateless Java system utilized to enable encryption, decryption, and distribution of private transactions for Hyperledger Besu [34]. Tessera consists of two components: the transaction manager and the security enclave.

The transaction manager ensures the privacy and security of transactions through the use of data encryption and access control mechanisms. It manages the interactions between participants in private transactions, including the encryption and decryption of transaction data, as well as the communication between different nodes involved in the private transaction process. It is a key component that helps maintain the confidentiality and integrity of transactions in a Blockchain network, making it suitable for applications that prioritize privacy.

This last feature is of particular importance, since it allows the issuance of private transactions to certain users on the network. The steps involved in a transaction are as follows: first, the vehicle v_i sends a private transaction to the Besu node i; then, Besu transmits the information to Tessera, where it is encrypted and distributed to the N participating vehicles. Once encrypted, Besu signs the transaction as private and distributes it throughout the network. Although the R nodes can see the transaction, only these N vehicles that are actively involved in the transaction can decrypt the data and execute the transaction. This method effectively privatizes transactions within an already private network, so only authorized participants can access the information of the transactions that take place on the network. This fact increases security considerably for several reasons:

- The risk of external attacks is reduced, as hackers cannot access transaction data without proper permissions. Furthermore, because the data are encrypted and encoded, they are difficult to modify or falsify.
- Trust increases among participants, as they can verify the authenticity and history of transactions without the need for intermediaries or third parties. This also reduces transaction costs and time.
- Privacy and data protection are enhanced as participants can decide the information to be shared and specify their preferred receivers. This enables compliance with legal and ethical regulations on the treatment of personal and sensitive data.

On the other hand, the security enclave is a secure processing environment that establishes communication with the transaction manager, effectively protecting the information it contains from potential malicious attacks. Among its critical functions, the security enclave is tasked with performing encryption and decryption operations required by the transaction manager, as well as managing node keys. Furthermore, it generates and manages cryptographic keys directly within the hardware, preventing any attempts at unauthorized extraction or duplication. Additionally, the security enclave is responsible for verifying the identity and permissions of the nodes involved in the exchange of sensitive information, thus ensuring robust access control and strong authentication.

To complete the network configuration, it is necessary to define the following parameters in Hyperledger Besu:

- Consensus algorithm.
- Block reference of the Ethereum main block from which it is created.
- Network ID.
- Number of nodes in the network, R
- Number of validator nodes, N.
- Other parameters: gas limit, timestamp, etc.

B. CONSENSUS ALGORITHMS

A consensus algorithm is a critical mechanism in decentralized networks, responsible for establishing agreement among nodes. It ensures consensus on transaction validity and order within the Blockchain, maintaining system safety and efficiency. Consensus prevents issues like double spending and defends Blockchain integrity, enhancing network security and reliability. Consensus algorithms enable decentralized decision-making, preventing manipulation by malicious actors. They also significantly impact Blockchain scalability and transaction processing efficiency. Commonly used algorithms, such as Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS), maintain decentralization, stability, security, and immutability, ensuring the trustworthiness and functionality of Blockchain networks [35].

PoW involves miners competing to solve complex mathematical puzzles to validate transactions and create new blocks, enhancing network security through computational effort and making it challenging for malicious actors to manipulate the Blockchain [36]. However, it has limitations, such as high energy consumption and scalability concerns because of its time-intensive nature. On the other hand,

PoS operates differently by selecting validators to create new blocks based on their staked cryptocurrency holdings. Unlike PoW, it does not involve extensive computational work, addressing energy and scalability concerns. PoS motivates participants to hold and stake cryptocurrency, promoting network security and stability [37]. However, it has limitations, including the potential concentration of power among wealthier participants and the problem nothing at stake, where validators could theoretically validate multiple conflicting blocks. Therefore, PoW emphasizes computational effort, while PoS prioritizes cryptocurrency ownership, offering alternative approaches to achieving consensus in the Blockchain space [38].

Proof of Authority (PoA) is a Blockchain consensus algorithm that relies on trusted validators, known as sealers, for transaction validation and block creation. Prioritizes network throughput and scalability, but sacrifices decentralization [39]. Although PoA exhibits better Byzantine Fault Tolerance (BFT) compared to centralized systems and improves transaction efficiency, it requires a minimum number of honest sealers to reach consensus on a unique block, which can introduce the risk of censorship, blacklisting, and potential validator manipulation. Despite these limitations, PoA's BFT properties make it suitable for decentralized applications (DApps) [40].

The latest version of Hyperleger Besu (v23.10.1, October 2023) supports all previous consensus protocols. For the developed Blockchain network, the chosen consensus algorithm is PoA, specifically the Istanbul Byzantine Fault Tolerance (IBFT) version 2.0 variant. Unlike version 1.0, which was deprecated in version v23.4 (May 2023) in Hyperledger Besu, version 2.0 ensures that a proposed block in any subsequent validation round[1] after the transaction's publication is valid for the block to be added to the network. IBFT 2.0 is a PoA BFT Blockchain consensus protocol that extends the original IBFT with immediate finality and a dynamic validator set. Unlike traditional consensus protocols such as PBFT (Practical Byzantine Fault Tolerance), IBFT 2.0 allows nodes to adjust their validator set through a voting mechanism, offering flexibility. Immediate finality ensures that once consensus is reached on a block, it becomes irreversible. Additionally, IBFT 2.0 guarantees optimal Byzantine fault-tolerant safety and persistence, supported by the IBFT-2.0-block-finalization-protocol within an eventually synchronous network model. These features make IBFT 2.0 an adaptable consensus algorithm for Blockchain networks.

The decision to implement this consensus algorithm is based on its ability to take advantage of the more popular consensus algorithms (POW and POS) and its suitability for private or hybrid networks, which aligns with the objectives of this paper. This choice is further justified by its capacity to deliver considerably faster block validation, a critical factor that has been carefully considered in order to create the Blockchain network. Moreover, for the framework developed, scalability and performance are needed, with PoA an algorithm that prioritizes both.

[1]Time it takes for the network to validate a block.

IV. SMART CONTRACTS FOR MONITORING ROAD TRAFFIC EVENTS

The subsequent step in our approach involved the development of smart contracts. In the context of an application designed for smart vehicles, the development of these smart contracts determines the interactions among vehicles within the network and assesses the implications of these interactions on other network participants.

In general, roads are typically designated with a unique alphanumeric code. Consequently, it would be important to maintain up-to-date and comprehensive information for the R vehicles in the set when they are on a route that requires monitoring. As a result, the development of a dedicated smart contract for each route becomes necessary, which will be called the *Route Contract*.

This contract will serve as the repository for route-specific information, keeping a list of the vehicles on the route, and providing write permissions. Importantly, these permissions are valid only for the duration of a vehicle's presence on the route, allowing it to modify the data. Any vehicle lacking the requisite permission to modify the route's state will be restricted to accessing the route's data. Given the necessity of both having a *Route Contract* for each monitored route and allowing the participation of the set of vehicles in these contracts, a smart contract factory named the *Route Factory* has been established. This *Route Factory* is implemented as another smart contract tasked with storing all contracts of type *Route Contract*, thereby providing the ability to create, delete, or interact with specific routes through it. The general structure of smart contracts is illustrated in Fig. 2.

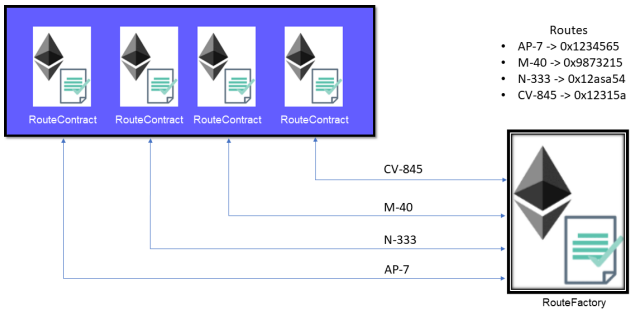

FIGURE 2. General structure of smart contracts.

This approach offers the advantage of scalability and security by categorizing vehicles based on the level of interaction on a particular route. This segmentation enables all participants to access and exchange data independently, regardless of their individual routes, potentially reducing traffic congestion by allowing drivers to make real-time routing decisions.

Regarding the coordinated behavior of the set of vehicles, the *Route Contracts* are responsible for the real-time storage

of route information. Vehicles that travel the same route can modify this information if an event is detected. Subsequently, once this information is validated, it will determine the actions to be taken by the rest of the vehicles within the system. This leads to a synchronized exchange of information between the vehicles. The key features of both Smart Contracts will be discussed next.

A. ROUTE CONTRACTS

As mentioned above, the development of *Route Contracts* is motivated by the objective of controlling the different situations that can occur on a road. Although the spectrum of potential events is extensive, for the purpose of validating our proposal, these events have been simplified into two possibilities: Clean route (i.e., a smooth flow of traffic without any incidents) or Congested route (where any event that disrupts the flow of traffic can be identified).

Smart vehicles traveling along the route will classify these situations. This information will then be sent to the Blockchain network for verification. After that, all vehicles on the route can take a synchronized action, such as simultaneous stopping or slowing down. In addition, this information will be shared with other vehicles that are not currently on that specific route. Contracts of type *Route Contract* are structured as shown in Fig. 3.

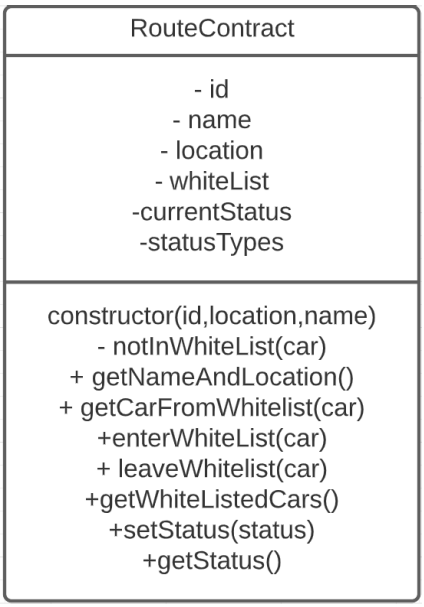

FIGURE 3. Attributes and functions of the smart contract *Route Contract*.

Each of the attributes has the following main features:

- **Id**: unique route identifier.
- **Name**: route name.
- **Location**: geographical area in which the route is located. In this way, the information to be collected can be segmented more precisely. This attribute also prevents the duplication of existing routes.
- **WhiteList**: this attribute consists of a vector where the hash addresses of the vehicles that are present on the route will be stored. These vehicles will have the permissions to interact with the corresponding *Route Contract*.
- **CurrentStatus**: current state of the route, which can be modified by any of the vehicles on the route existing on the whitelist.
- **StatusTypes**: it specifies the potential states found in the route. The following two states have been defined, as mentioned above:
  - CLEAN
  - CONGESTED

The main features of the functions developed (see Fig. 3) are described below:

- **constructor(id, location, name)**: this function initializes the route attributes mentioned above. The initial state will be CLEAN. The creation of the new route is made by a request of a *Route Contract*.
- **notInWhiteList(car)**: this function is used to ensure that transactions occur within the smart contract and is applied to the functions **enterWhiteList(car)** and **setStatus(status)**. Thus, it verifies that the sender of the transaction is in the route, i.e., it is in the whitelist. If this is true, the node is allowed to proceed with the transaction. Otherwise, the transaction is reversed.
- **getNameAndLocation()**: it returns a string containing the name and location of the route.
- **getCarFromWhitelist(car)**: in this function, the hash address of the vehicle is obtained if the vehicle is on the route.
- **enterWhiteList(car)**: this function first checks if the address of the vehicle exists on the route. If not, it will be added to the route, i.e. to the whitelist. This function modifies the state of the smart contract, so it must be indicated that it is a *payable* function, i.e. it will have a cost.
- **leaveWhitelist(car)**: unlike the previous one, this function will modify the state of the contract, but this time it will remove the vehicle that has left the route from the list and update the list.
- **getWhiteListedCars(car)**: it retrieves the hash addresses of all vehicles on the route.
- **setStatus(status)**: it modifies the state of the route based on vehicle observations. As it has the ability to modify the contract, it must contain the attribute *payable*. The function checks if the parameter status is validated. If this occurs, it updates the currentStatus attribute and triggers a Blockchain network event. Otherwise, it results in an error and reverses the transaction.
- **getStatus(car)**: it returns the current state of the route, that is, the information contained in the variable currentStatus.

B. ROUTE FACTORY

This type of contract is a factory contract that stores all existing routes. It allows the creation of new routes, the deletion of existing ones, and the interaction with them. Smart

contracts of the *Route Factory* type are structured as shown in Fig. 4.

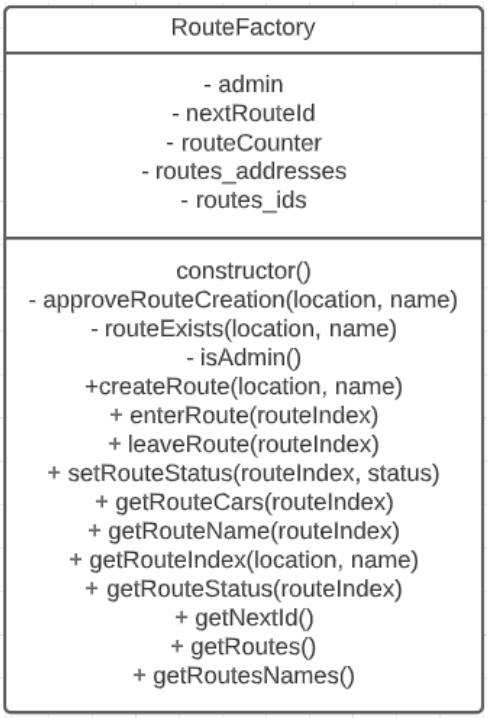

FIGURE 4. Attributes and functions of the smart contract *Route Factory*.

With respect to the attributes, these are their main features:

- **Admin**: this attribute is of type address and will be assigned only in the constructor to the address that deploys this smart contract.
- **NextRouteId**: it corresponds to the Id that will be assigned to the next route to be created.
- **RouteCounter**: counter indicating the existing routes.
- **Routes_addresses**: vector of type *Route Contract*, where the hash addresses of the route contracts deployed by the *Route Factory* contract are stored for subsequent interaction with them.
- **Routes_ids**: it stores routes and controls that they are not duplicated.

The main features of the functions developed are the following:

- **constructor()**: initialize the attributes.
- **approveRouteCreation(location, name)**: approves that a new route can be created, checking the Routes_ids attribute.
- **routeExists(location, name)**: checks if a route exists.
- **isAdmin()**: it is responsible for verifying that the node who wants to interact with a smart contract function is the administrator, i.e. that its hash address is the same as the one set by the constructor when the contract was deployed.
- **createRoute(location, name)**: it will create a new route by generating an instance of the *Route Contract* type. This involves calling its constructor, providing the required variables, deploying it on the network, and storing it in the Routes_addresses variable. Before initiating this process, a verification step is performed to confirm that the requester has the states of system administrator.
- **enterRoute(routeIndex)**: it will access a particular route by calling the enterWhitelist function of the *Route Contract*, allowing the sender of the transaction to gain access to the designated route.
- **leaveRoute(routeIndex)**: the previous process is repeated, but interacting with the leaveWhitelist function of the *Route Contract*.
- **setRouteStatus(routeIndex, status)**: it changes the state of the route corresponding to the index using the setStatus(status) function within the contract *Route Contract*.
- **getRouteCars(routeIndex)**: the getWhiteListedCars() function is called for the route stored at the index corresponding to routeIndex.
- **getRouteName(routeIndex)**: the route name is obtained by accessing the contract through its index in the routes vector and calling the getNameAndLocation function.
- **getRouteIndex(location, name)**: it provides, based on the route name and location, the corresponding index in the route vector.
- **getRouteStatus(routeIndex)**: it returns the value of the state variable stored in the contract corresponding to the index in the contract vector *Route Contract*.
- **getNextId()**: it provides the value corresponding to the next state of the route.
- **getRoutes()**: it gives the hash addresses of all the routes stored in the vector corresponding to the deployed contracts.
- **getRoutesNames()**: it returns all the names and locations of the existing routes.

In conclusion, adopting the described approach to develop smart contracts provides the ability to include sets of N vehicles as validators in a deployed Blockchain network. Meanwhile, vehicles from the global set of R vehicles that are not part of the specific route or have joined after the validation of an anomalous event can access the network information but do not have the authority to modify or validate it. This approach effectively isolates vehicles within a route from potentially malicious transactions, thus improving security.

C. DESIGN OF AN APPLICATION INTERFACE TO CONTROL THE BLOCKCHAIN NETWORK

As mentioned above, an application interface has been developed to serve as an intermediary layer between the network, the smart contracts deployed, and the vehicles. It also interfaces with other elements, such as the data manager, offering a means to monitor the states of all routes.

To implement the model, a two-layer architecture has been employed. The lower layer corresponds to a *Route*

Controller, responsible for controlling how vehicles interact with the network. The upper layer corresponds to an Application Programming Interface (API) that utilizes the methods declared in the controller to execute the actions desired by the vehicles in the set. This API enables vehicles to interact directly with the contracts deployed on the network.

The *Route Controller* will determine how vehicles interact with the Blockchain network. It will be responsible for deciding whether these interactions are secure and if the information to be introduced is reliable. This fact is crucial because it affects the potential modification of vehicle behavior. All forms of interaction by vehicles on the network, such as data modification or retrieval, will be conducted through a private transaction executed by Tessera. Fig. 5 shows a flow chart of a private transaction, using the creation of a route as an example.

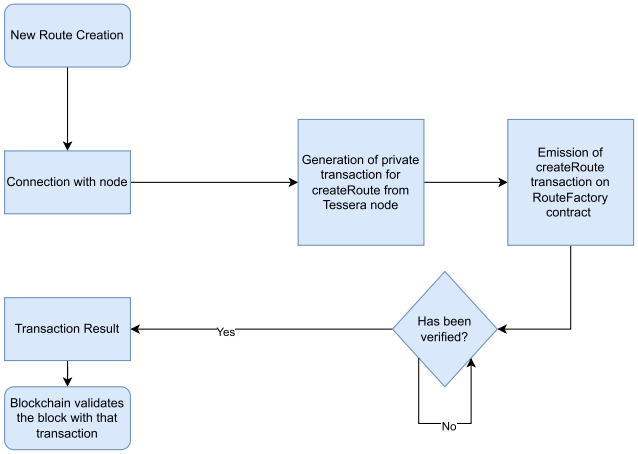

FIGURE 5. Flowchart for a private route creation transaction.

On the other hand, when the API is started, a contract of type *Route Factory* is deployed and stored as a constant value. This value will be used by all functions to subsequently access the contracts of type *Route Contract* stored within it, as explained above. This approach makes possible both the implementation of the main contract by the first node included in the private network and the creation of a new route. Therefore, the first node will be the only one capable of creating new routes, acting as the system administrator. Most of the requests made by the nodes follow a similar process, varying mainly in the subsequent steps after obtaining the index. Fig. 6 shows an example for the case of a change in the route state.

As shown, the private network provides a verification process when initiating a transaction to ensure that the addresses used are valid and reliable within the network (through Tessera and Hyperledger Besu). Furthermore, the security of network interactions is significantly improved by the security filters implemented in the smart contracts.

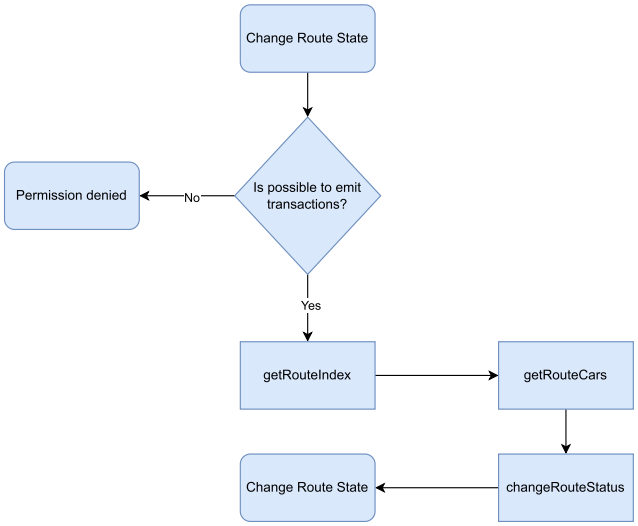

FIGURE 6. API flowchart for changing the state of a route.

V. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

The tests have been conducted through a 3D graphical simulation, using Python and the CoppeliaSim simulator [41] to generate simulated data of the vehicle on the road in different situations [42], [43]. All the materials related to this article (including source codes, configuration files, logs, and experimental results) can be found in our repository [44]. The smart vehicles in the set will use the same program to ensure a coordinated execution. The program is designed to perform three types of tasks:

- Interaction with the CoppeliaSim simulator, which contains the sensors necessary to provide information about the potential events to monitor on the route.
- Interaction with the real-time database of the system.
- Interaction with the Blockchain network deployed.

By implementing the system in this way, the objective is to overcome one of the major problems of using Blockchain technology: latency [45], a particularly relevant issue in a system in which the speed of response to a critical traffic event can be lifesaving. Therefore, it is decided not to establish a direct connection between vehicles, i.e., no message passing between vehicles is implemented. Instead, the focus is on the information that is validated and added as a block to the chain to improve this latency. This process can be summarized as follows:

1) The vehicle monitors with its sensors the states of the route, which initially is always CLEAN.
2) If a situation considered hazardous is detected, such as an accident, the vehicle that identifies this event transmits the information to the Blockchain network.
3) Once it is confirmed that the emitted event aligns with the information in the Blockchain, that is, the rest of the validator vehicles verify it is correct, a warning is instantly sent to the all of the vehicles.

4) When an event is confirmed, all vehicles will be alerted and automatically reduce their speed to avoid any accidents. If the car is driven autonomously, the driver must take control when the warning is given. The driver will then decide whether to slow down further or stop the vehicle. If the driver does not take any action, the car will automatically reduce its speed and come to a stop if necessary to avoid collisions with other vehicles.
5) Once the issue has been resolved and verified, the vehicles will return to the traffic circulation.

Thus, the interaction over the network to validate the data is crucial. This interaction occurs three times, involving the following processes:

- **Route selection**: the first step involves choosing the route to be acted on. The network is consulted to identify the available routes that can be taken action. Once selected, the corresponding values are stored for future interactions and the required keys are obtained for transaction execution.
- **Change in the state of the selected route**: If an event has occurred, this information needs to be updated in the network data. This is achieved by sending a transaction to the contract associated with the relevant data of the route.
- **Checking the state of the route**: Once the transaction has been sent, if a vehicle has not been the issuer, it must verify that the state of the route data is consistent with the event that was communicated. his verification guarantees the integrity of the data, enabling the vehicle to take the necessary actions.

Fig. 7 illustrates the global interactions of our system for the set of smart vehicles, based on the proposed model.

In summary, the created components work together to allow each smart vehicle to obtain an accurate and immutable piece of data on a private Blockchain. Requests are handled through the API of each vehicle, which interacts with the deployed smart contracts. These contracts store the data on the Blockchain, making it available for the set of vehicles to act in synchrony.

B. RESULTS

In this section, the description of the tested application environments and the results of the proposed system are presented. The process begins with a vehicle detecting an event in the Blockchain network, followed by validation from the validator nodes. After successful validation, an alert is broadcast to all network nodes, triggering a coordinated speed reduction to prevent collisions. Subsequently, the driver is asked to take control, deciding whether to slow down or stop. If the driver does not respond, the system automatically initiates a speed reduction or brings the vehicle to a complete stop. Obtaining low response times is crucial in real-world scenarios, where intelligent vehicles must react quickly to critical events, such as accidents. According to some studies, the human response time ranges from 0.5 to 1 s [46].

Scalability provided by Hyperledger Besu has been explored from two perspectives: horizontal scalability (or scale-out) and vertical scalability (or scale-up). Vertical scalability involves enhancing the capacity of a single node. Instead of adding more nodes, this approach focuses on improving the capabilities of existing hardware, such as adding more CPU, RAM, or storage. Various configurations can be tested by varying these hardware characteristics. For example, evaluating a network with nodes having 2 CPU cores and 16 GB of RAM versus a network with nodes having 8 CPU cores and 64 GB of RAM. In our case, it has been decided that the storage memory of the nodes remains fixed at 64GB SSD, since it is the minimum amount necessary to implement the proposed system. On the other hand, horizontal scalability involves adding more nodes to a distributed system, in this case, a Blockchain network, with fixed hardware capabilities for each node. Its goal is to increase the capacity of the system by distributing the workload among several nodes.

In order to evaluate these approaches, the average transaction latency of the network has been taken into account. As vehicle response time is essential to avoid accidents, a maximum latency of 1 second has been established, which is in line with the maximum response time of a human, as previously mentioned. This would be the maximum time it would take to complete the process of sending a transaction by a node and validating that transaction by the rest of the nodes. In the experiments conducted, node communication occurred through a virtual network with a bandwidth of 1000 Mb/s and network latency below 1 ms. The consensus algorithm chosen for this study has been PoA, IBFT 2.0. Additionally, for the analysis of the performance, the request sending rate to the Blockchain network has been varied, ranging from 500 req/s to 1000 req/s. Finally, tests have been conducted to observe the latency evolution for two types of transactions: those involving write operations to the Blockchain (open transactions) and those involving read operations from the Blockchain (query transactions). This aims to simplify the evaluation process.

A network of 18 nodes was established for the purpose of vertical scalability, and various node configurations were tested to simulate a system with multiple vehicles on the road in a realistic manner. To encode the node configurations, the notation aCbG has been utilized, where the nodes are configured with hardware having a CPU cores and b GB of RAM. The different node configurations and the evaluation results are shown in Fig. 8.

The optimal configuration for the network nodes is determined to be 4C16G. This configuration is chosen because it is the point at which the latency value falls below 1 second for any of the proposed sending ratios. However, it is possible to use nodes with higher capacity, but the performance improvement is not significant. Therefore, this configuration has been used in the subsequent experiments.

Subsequently, the horizontal scalability of the system was tested, maintaining the aforementioned hardware

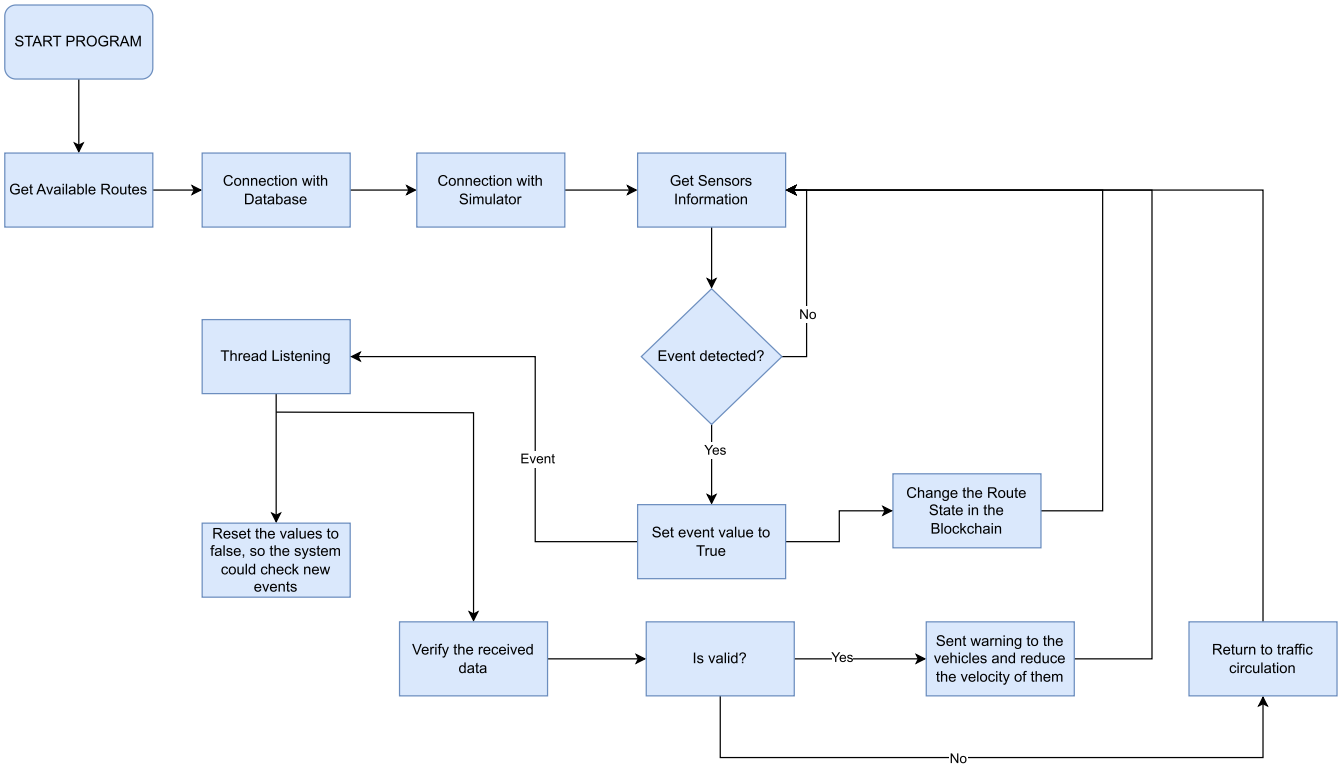

FIGURE 7. Global interactions of our framework.

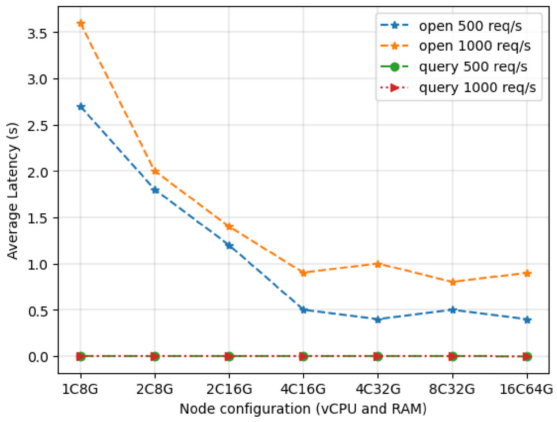

FIGURE 8. Vertical Scalability of IBFT 2.0 with different send rates and 18 nodes.

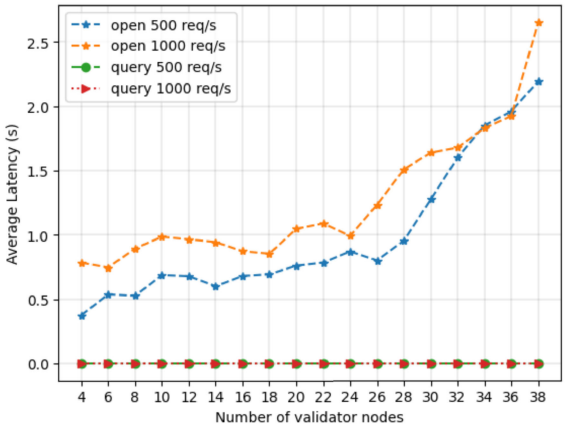

FIGURE 9. Horizontal Scalability of IBFT 2.0 with different send rates.

configuration. The objective is to determine how many validating nodes or vehicles the Blockchain network would be able to support and verify its applicability to a real system. The results are presented in Fig. 9. In this context, for 500 req/s, it is found that the maximum number of validating nodes the network can handle is 28, while for 1000 req/s, this value decreases to 24 without exceeding the proposed latency value.

As a result of the two scalability tests, it can be concluded that, for querying information on the Blockchain network, there is no variation in latency, remaining consistently at a minimal value. On the other hand, performing a write operation on the Blockchain for subsequent validation consumes, as expected, bandwidth and increases latency depending on the number of validating nodes configured in the network. Therefore, it can be concluded that the Blockchain network developed with Hyperledger Besu is suitable for testing in real-world applications to confirm its effectiveness.

Next, a simulation environment has been configured consisting of a maximum of $N = 28$ validating vehicles with a sending rate of 500 req/s. These vehicles will be able to observe the two possible states of the route: CONGESTED when the smart vehicle sensors detect congestion, and it is

labeled as CLEAN when there are no issues with the route. To simplify the graphical simulation, the CONGESTED state was modeled with a red block and the CLEAN state has been represented with a white block. In a real-world scenario, these events will be detected by appropriate intelligent sensors and a suitable Machine Learning algorithm, a question that is under development in future work. In addition, for clarity in the graphical representation, Fig. 10 illustrates an example with $N = 3$ vehicles on the route.

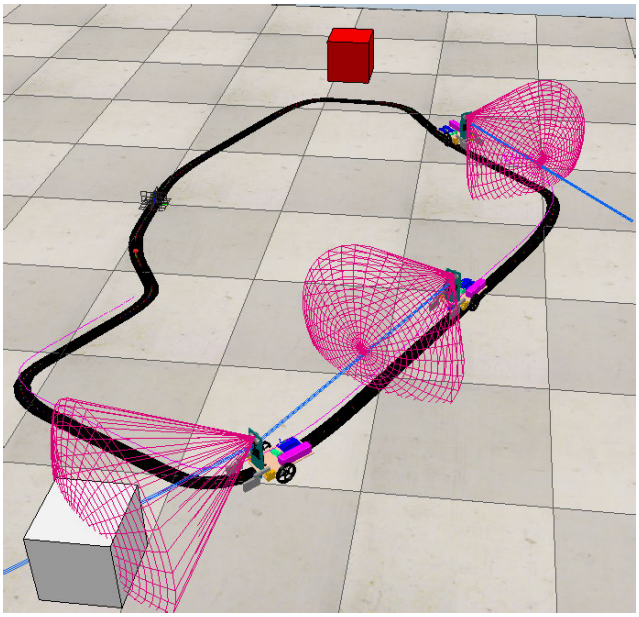

FIGURE 10. Simulation environment with $N = 3$ vehicles.

```
1. ROUTE AP-7
Select a route: 1
Connected to remote API server port 20000
12:16:09: ROUTE STATE CLEAN
12:16:16: ROUTE STATE CLEAN
12:12:22: ROUTE STATE CLEAN
12:16:22.164548: Vehicle 2 Emitting event...
12:16:22.481293: ROUTE STATE CONGESTED
-------------------------------------------
AP-7
12:16:22.518908: ROUTE STATE CONGESTED
-------------------------------------------
12:16:22.826329: Reducing speed of all members,
dangerous situation detected and verified by
blockchain system!
```

FIGURE 11. Blockchain system messages with $N = 3$ vehicles.

When an event is detected, a warning is sent out to all vehicles on the route where the event happened, alerting them to the potential for congestion. This notification serves as an initial indication of an anomalous event. The notification is carried out by broadcasting the event through the system database. Google Firestore has been selected as the database due to its additional layer of information verification and its fast response time [47]. Eventually, the verification of the event alert occurs at a different time for each vehicle. This can be due to multiple factors, including potential asynchrony in the software flow of each vehicle or varying response times in the transaction that requests data from the network, among other possibilities. Nevertheless, these potential scenarios represent what might naturally occur in a real-world implementation of our proposal.

The duration from emitting the event to sending the congestion alert to vehicles and coordinating the speed reduction falls within the designated maximum target latency. In the case of Fig. 11, the response time is approximately **0.32** seconds for the scenario with $N = 3$. This time achieved is faster than a driver's mean reaction time.

Next a series of tests were conducted to determine the average response time for the initial speed reduction moment, with $N = 28$ validating nodes, to evaluate the system's performance in a real-world scenario. To do this, the experiment was repeated six times to obtain multiple performance samples and calculate the time mean. Table 2 shows the times obtained.

TABLE 2. Average response time (in seconds) for each vehicle.

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Vehicle 1	0.84	0.81	0.95	0.82	0.95	0.78
Vehicle 2	0.82	0.82	0.79	0.75	0.94	0.91
Vehicle 3	0.98	0.92	0.89	0.74	0.85	0.76
Vehicle 4	0.78	0.73	0.97	0.8	0.93	0.88
Vehicle 5	0.89	0.9	0.79	0.94	0.85	0.96
...	...	...	...	...	...	...
Vehicle 28	0.88	0.92	0.81	0.77	0.75	0.76
Average Time	**0.82**	**0.84**	**0.79**	**0.77**	**0.75**	**0.81**

As it can be seen, although the number of nodes increases up to the maximum number considered, the latency remains below the average reaction time of a driver to a road event. Therefore, the results obtained in this research demonstrate the practical integration of the proposed system in a real operational scenario. However, it is important to note that such an implementation may involve adjustments and modifications due to the challenges associated with the complexity of integrating the system into a real context.

Finally, let us summarize the main results obtained from some of the related works:

- In [19] a Blockchain-enabled solution was proposed. It had a slightly higher latency in generating blocks compared to reputation-based consensus and conventional PBFT-based consensus algorithms.
- The authors in [20] showed that the gas consumption of all networks was low with the Proof of Authority (PoA) algorithm, and the proposed architecture had very low gas consumption for the Blockchain networks.
- The experimental evaluations in [31], provided a comprehensive understanding of the performance characteristics of Hyperledger Besu in a private Blockchain setting, highlighting the impact of load balancing, consensus algorithms, scalability, and resource utilization on the overall system performance.

- In [48] Hyperledger Besu was used. The authors showed that some Blockchain parameters, such as block time and block size, were the most significant factors in determining the performance of Hyperledger Besu. Besu performance was bottlenecked by transaction execution and Blockchain state updates, which were influenced by parameters such as node computation power, transaction complexity, and load balancing.
- The research in [49] introduced a combination of edge computing and consortium Blockchains to support efficient computation and storage capabilities with low communication delay while providing data auditability. Edge computing allows computing tasks to be offloaded locally, resulting in reduced latency for data processing and analysis.
- In [50], the results demonstrated that the proposed scheme achieves efficient and secure Federated Learning in the Internet of Vehicles, while preserving the privacy of vehicle data.

After performing a comparison between the results obtained by our model with previous research, it can be confirmed the validity of our proposal, providing an additional perspective on the effectiveness and consistency of the proposed system in relation to other existing approaches.

Therefore, this paper is unique in its focus on the use of Blockchain-based approaches to improve safety in smart vehicles. It proposes an innovative approach where advanced sensors are used to monitor route conditions and transmit data securely to a Blockchain network for real-time risk assessment. In addition, it stands out from prior research by responding quickly to anomalous situations, which has not been commonly explored in other works within the context of the IoV. Finally, this paper contributes to academia by addressing a critical aspect of smart vehicles technology through practical experimentation and scalability validation.

VI. CONCLUSION

This paper presents an introduction to a Blockchain-based framework that was specifically created to improve the security and verification of data among smart vehicles. This framework aims to contribute to improving communication, data integrity, authentication, and decentralization within smart and connected vehicle systems. The analysis of the scalability of the suggested private Blockchain network highlights its suitability for real-world integration, offering a strong and transparent foundation for managing event information within the connected vehicle ecosystem. The practical implementation of the system in the simulated scenarios highlights its potential and effectiveness. As smart vehicles become essential in the future of transportation, the framework presented in this paper sets the foundation for a more secure, reliable, and transparent framework in handling event information within the connected vehicle ecosystem.

This study has demonstrated the potential of a Blockchain-based framework for event detection in smart vehicles. The successful implementation of the framework in a 3D graphical simulator, along with the utilization of Hyperledger Besu technology for the creation of the Blockchain network, showed both horizontal and vertical scalability. The framework was able to respond quickly and effectively to anomalous situations on a route, improving human reaction times in similar circumstances. These findings suggest that the proposed solution could be applied in the real world. The integration of advanced sensors and communication technologies in smart vehicles, combined with the decentralized and secure nature of Blockchain, makes the framework a robust and reliable approach to improving road safety. However, further validation is needed through field trials and collaboration with industry stakeholders before the framework can be implemented in real-world scenarios.

Despite the challenges ahead, the comprehensive simulations conducted in this study encourage optimism about the tangible benefits that the proposed framework could bring to the realm of smart transportation systems, creating a safer and more efficient future for vehicular communication and accident prevention. It is important to note that the implementation of the proposed system in actual circumstances may require precise modifications to tackle the difficulties that come with real-world applications.

Future research will study a proposal for a Machine Learning model that categorizes different route states to provide more information to smart vehicles and give them the ability to make sound decisions. Furthermore, testing the integration of this system in a real-world environment will be done to prove its usefulness in practical circumstances.

REFERENCES

[1] F. Aznar, M. Pujol, R. Rizo, F. Pujol, and C. Rizo, "Energy-efficient swarm behavior for indoor UAV ad-hoc network deployment," *Symmetry*, vol. 10, no. 11, p. 632, Nov. 2018.
[2] K. Shah, C. Sheth, and N. Doshi, "A survey on IoT-based smart cars, their functionalities and challenges," *Proc. Comput. Sci.*, vol. 210, pp. 295–300, Jan. 2022.
[3] M. Saleem, S. Abbas, T. M. Ghazal, M. A. Khan, N. Sahawneh, and M. Ahmad, "Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques," *Egyptian Informat. J.*, vol. 23, no. 3, pp. 417–426, Sep. 2022.
[4] A. Hammoud, H. Sami, A. Mourad, H. Otrok, R. Mizouni, and J. Bentahar, "AI, blockchain, and vehicular edge computing for smart and secure IoV: Challenges and directions," *IEEE Internet Things Mag.*, vol. 3, no. 2, pp. 68–73, Jun. 2020.
[5] L. Montero, C. Ballesteros, C. de Marco, and L. Jofre, "Beam management for vehicle-to-vehicle (V2V) communications in millimeter wave 5G," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100424.
[6] F. Jamil, O. Cheikhrouhou, H. Jamil, A. Koubaa, A. Derhab, and M. A. Ferrag, "PetroBlock: A blockchain-based payment mechanism for fueling smart vehicles," *Appl. Sci.*, vol. 11, no. 7, p. 3055, Mar. 2021.
[7] M. H. Tabatabaei, R. Vitenberg, and N. R. Veeraragavan, "Understanding blockchain: Definitions, architecture, design, and system comparison," *Comput. Sci. Rev.*, vol. 50, Nov. 2023, Art. no. 100575.
[8] D. Conte de Leon, A. Q. Stalick, A. A. Jillepalli, M. A. Haney, and F. T. Sheldon, "Blockchain: Properties and misconceptions," *Asia Pacific J. Innov. Entrepreneurship*, vol. 11, no. 3, pp. 286–300, Dec. 2017.
[9] S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," *Peer-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2901–2925, Sep. 2021.

[10] J. C. Mendoza-Tello, H. Mora, F. A. Pujol-López, and M. D. Lytras, "Social commerce as a driver to enhance trust and intention to use cryptocurrencies for electronic payments," *IEEE Access*, vol. 6, pp. 50737–50751, 2018.
[11] F. A. Sunny, P. Hajek, M. Munk, M. Z. Abedin, M. S. Satu, M. I. A. Efat, and M. J. Islam, "A systematic review of blockchain applications," *IEEE Access*, vol. 10, pp. 59155–59177, 2022.
[12] H. Mora, F. A. Pujol, T. Ramírez, A. Jimeno-Morenilla, and J. Szymanski, "Network-assisted processing of advanced IoT applications: Challenges and proof-of-concept application," *Cluster Comput.*, pp. 1–17, Jun. 2023.
[13] P. Tasatanattakool and C. Techapanupreeda, "Blockchain: Challenges and applications," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 473–475.
[14] R. Gupta, S. Tanwar, N. Kumar, and S. Tyagi, "Blockchain-based security attack resilience schemes for autonomous vehicles in industry 4.0: A systematic review," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106717.
[15] T. Dargahi, H. Ahmadvand, M. N. Alraja, and C.-M. Yu, "Integration of blockchain with connected and autonomous vehicles: Vision and challenge," *J. Data Inf. Qual.*, vol. 14, no. 1, pp. 1–10, Mar. 2022.
[16] M. Dibaei, X. Zheng, Y. Xia, X. Xu, A. Jolfaei, A. K. Bashir, U. Tariq, D. Yu, and A. V. Vasilakos, "Investigating the prospect of leveraging blockchain and machine learning to secure vehicular networks: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 683–700, Feb. 2022.
[17] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.
[18] F. T. Progga, H. Shahriar, C. Zhang, and M. Valero, *Securing Vehicular Network Using AI and Blockchain-Based Approaches*. Cham, Switzerland: Springer, 2021, pp. 31–44.
[19] F. Ayaz, Z. Sheng, D. Tian, and V. C. M. Leung, *Blockchain-Enabled Security and Privacy for Internet-of-Vehicles*. Cham, Switzerland: Springer, 2021, pp. 123–148.
[20] S. K. Singh, J. H. Park, P. K. Sharma, and Y. Pan, "BIIoVT: Blockchain-based secure storage architecture for intelligent Internet of Vehicular Things," *IEEE Consum. Electron. Mag.*, vol. 11, no. 6, pp. 75–82, Nov. 2022.
[21] A. A. Laghari, A. A. Khan, R. Alkanhel, H. Elmannai, and S. Bourouis, "Lightweight-BIoV: Blockchain distributed ledger technology (BDLT) for Internet of Vehicles (IoVs)," *Electronics*, vol. 12, no. 3, p. 677, Jan. 2023.
[22] A. A. Laghari, A. K. Jumani, R. A. Laghari, and H. Nawaz, "Unmanned aerial vehicles: A review," *Cognit. Robot.*, vol. 3, pp. 8–22, Dec. 2023.
[23] M. Singh and S. Kim, "Introduce reward-based intelligent vehicles communication using blockchain," in *Proc. Int. SoC Design Conf. (ISOCC)*, Nov. 2017, pp. 15–16.
[24] C. Oham, R. A. Michelin, R. Jurdak, S. S. Kanhere, and S. Jha, "B-FERL: Blockchain based framework for securing smart vehicles," *Inf. Process. Manage.*, vol. 58, no. 1, Jan. 2021, Art. no. 102426.
[25] R. Jabbar, E. Dhib, A. B. Said, M. Krichen, N. Fetais, E. Zaidan, and K. Barkaoui, "Blockchain technology for intelligent transportation systems: A systematic literature review," *IEEE Access*, vol. 10, pp. 20995–21031, 2022.
[26] M. G. M. M. Hasan, A. Datta, M. A. Rahman, and H. Shahriar, "Chained of things: A secure and dependable design of autonomous vehicle services," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2018, pp. 498–503.
[27] G. Bendiab, A. Hameurlaine, G. Germanos, N. Kolokotronis, and S. Shiaeles, "Autonomous vehicles security: Challenges and solutions using blockchain and artificial intelligence," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 3614–3637, Apr. 2023.
[28] A. R. Javed, M. A. Hassan, F. Shahzad, W. Ahmed, S. Singh, T. Baker, and T. R. Gadekallu, "Integration of blockchain technology and federated learning in vehicular (IoT) networks: A comprehensive survey," *Sensors*, vol. 22, no. 12, p. 4394, Jun. 2022.
[29] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered federated learning: Challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–31, Nov. 2023.
[30] S. D. Palma, R. Pareschi, and F. Zappone, "What is your distributed (hyper)ledger?" in *Proc. IEEE/ACM 4th Int. Workshop Emerg. Trends Softw. Eng. Blockchain (WETSEB)*, May 2021, pp. 27–33.
[31] C. Fan, C. Lin, H. Khazaei, and P. Musilek, "Performance analysis of hyperledger besu in private blockchain," in *Proc. IEEE Int. Conf. Decentralized Appl. Infrastruct. (DAPPS)*, Aug. 2022, pp. 64–73.
[32] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, and F. Han, "An architecture and performance evaluation of blockchain-based peer-to-peer energy trading," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3364–3378, Jul. 2021.
[33] *Hyperledger Besu: How to Create an Ethereum Genesis File*. Accessed: Oct. 13, 2023. [Online]. Available: https://consensys.io/blog/hyperledger-besu-how-to-create-an-ethereum-genesis-file
[34] *Tessera—Enterprise Implementation of Hyperledger Besu Transaction Manager*. Accessed: Sep. 15, 2023. [Online]. Available: https://github.com/Consensys/tessera
[35] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2567–2572.
[36] P. R. Nair and D. R. Dorai, "Evaluation of performance and security of proof of work and proof of stake using blockchain," in *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV)*, Feb. 2021, pp. 279–283.
[37] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85727–85745, 2019.
[38] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 1545–1550.
[39] B. Lashkari and P. Musilek, "A comprehensive review of blockchain consensus mechanisms," *IEEE Access*, vol. 9, pp. 43620–43652, 2021.
[40] S. Joshi, "Feasibility of proof of authority as a consensus protocol model," 2021, *arXiv:2109.02480*.
[41] *Robot Simulator CoppeliaSim: Create, Compose, Simulate, Any Robot—Coppelia Robotics*. Accessed: Sep. 1, 2023. [Online]. Available: https://www.coppeliarobotics.com/
[42] I. A. Ribeiro, T. Ribeiro, G. Lopes, and A. F. Ribeiro, "End-to-end approach for autonomous driving: A supervised learning method using computer vision algorithms for dataset creation," *Algorithms*, vol. 16, no. 9, p. 411, Aug. 2023.
[43] C. Allione, S. Pincin, and S. Zudaire, "Development platform for autonomous driving in real and simulated environments," in *Proc. 22nd Simposio Argentino de Inteligencia Artificial (ASSAI)-JAIIO (Modalidad Virtual)*, 2021, pp. 77–90.
[44] *Repository of the Proposed Blockchain Framework*. Accessed: Nov. 17, 2023. [Online]. Available: https://github.com/doctorcako/swarmRobotic Blockchain
[45] L. Wan, D. Eyers, and H. Zhang, "Evaluating the impact of network latency on the safety of blockchain transactions," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 194–201.
[46] R. Jurecki, "Driver response time in different traffic situations for using in accident analysis," *Zeszyty Naukowe Instytutu Pojazdów/Politechnika Warszawska*, vol. 2, p. 106, Jan. 2016.
[47] *Firestore*. Accessed: Oct. 24, 2023. [Online]. Available: https://firebase.google.com/docs/firestore?hl=es-419
[48] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.
[49] Q. Mei, H. Xiong, Y. Zhao, and K.-H. Yeh, "Toward blockchain-enabled IoV with edge computing: Efficient and privacy-preserving vehicular communication and dynamic updating," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Jan. 2021, pp. 1–8.
[50] N. Wang, W. Yang, X. Wang, L. Wu, Z. Guan, X. Du, and M. Guizani, "A blockchain based privacy-preserving federated learning scheme for Internet of Vehicles," *Digit. Commun. Netw.*, May 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864822001134

FRANCISCO A. PUJOL received the B.S. degree in telecommunications engineering from the Polytechnic University of Valencia, Spain, in 1998, and the Ph.D. degree in computer science from the University of Alicante, Spain, in 2001. He was a Visiting Lecturer with Cardiff University, in 2004. He is currently an Associate Professor with the Department of Computer Technology and Computation, University of Alicante. His research interests focus on robotics, machine learning, face recognition, computer vision, and computer parallel architectures, on which he has published more than 100 technical journals and conference papers.

HIGINIO MORA received the first B.S. degree in computer science engineering, the second B.S. degree in business studies, and the Ph.D. degree in computer science from the University of Alicante, Spain, in 1996, 1997, and 2003, respectively. Since 2002, he has been a member of the Faculty of the Computer Technology and Computation Department, University of Alicante, where he is currently an Associate Professor and a Researcher with the Specialized Processors Architecture Laboratory. He has participated in many conferences and most of his work has published in international journals and conferences, with more than 50 published articles. His research interests include computer modeling, computer architectures, high-performance computing, embedded systems, the Internet of Things, and cloud computing paradigm.

TAMAI RAMÍREZ received the B.S. degree in robotics engineering from the University of Alicante, Spain, in 2022, and the M.S. degree in artificial intelligence from Valencia International University, Spain, in July 2023. He is currently pursuing the Ph.D. degree with the Department of Computer Technology and Computation, University of Alicante. His research interests include cloud computing, artificial intelligence, the Internet of Things, machine learning, and computer vision.

CARLOS ROCAMORA received the B.S. degree (Hons.) in computer science from the University of Alicante, Spain, in 2023. After graduation, he has demonstrated exceptional aptitude and commitment to advancing the frontiers of computer science. He is currently a Senior Developer of blockchain applications with the University of Alicante. His research interests include blockchain, robotics, artificial intelligence, and distributed systems.

ARTURO BEDÓN received the B.S. degree in informatic systems engineering from the Polytechnic School of the Army, in 2002, and the M.Sc. degree in computer science and electronic commerce and the Ph.D. degree in computer science from the University of Alicante, in 2014 and 2023, respectively. He is currently an Associate Professor with Universidad Central del Ecuador, Quito, Ecuador. His research interests include software engineering, cryptocurrencies, blockchain, and cloud computing.