JCST Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777652

MANAGEMENT OF TRUST IN MULTI-AGENT ENVIRONMENTS USING THE BLOCKCHAIN MODEL ON THE INTERNET OF THINGS FOR SMART CITY

Pankaj Jagtap and Dr. Sandeep Singh Rajpoot
Department of Computer Application, Dr. A. P. J. Abdul Kalam University, Indore, (M.P.),
India, Corresponding Author Email: Jagtap03@gmail.com

Abstract :
The proper management of trust in multi-agent settings is a vital component of guaranteeing the security and dependability of systems, particularly in the context of the Internet of Things (IoT) for smart cities. This is especially true in light of the fact that smart cities are becoming more interconnected. In order to overcome the issues that are brought about by the heterogeneous and linked nature of IoT devices inside smart cities, this article makes a proposal for a trust management mechanism that is based on the model of Blockchain. The trust management system that is based on blockchain technology provides a method that is safe and decentralized, as well as increased transparency and responsibility for transactions that are connected to trust. The suggested approach makes it easier for agents participating in the Internet of Things ecosystem to engage in trustworthy interactions by drawing on the immutability and consensus processes that are built into Blockchain technology. This study illustrates the usefulness of the suggested strategy in assuring the secure authorization of smart city resources via actual implementation, which is shown throughout the paper. In addition, a hybrid application that is both user-friendly and able to provide seamless interactions and control over smart city equipment is built. This further enhances the entire user experience.

Keywords: Internet of Things, Blockchain, smart city, Trust Execution , Smart contracts, Decentralized.

1. INTRODUCTION

The fast development of technology has resulted in the birth of the Internet of Things (IoT) and Smart Cities, which have revolutionized the manner in which people interact with urban surroundings. Specifically, the IoT connects everyday objects to the internet [1]. The Internet of Things (IoT) and other networked technologies are being used by Smart Cities to improve urban services, increase sustainability, and maximize resource management. However, there has been a significant proliferation of connected devices, and the Internet of Things networks themselves are decentralized, which presents a new set of issues, especially when it comes to maintaining trust and safety among a large number of interacting agents [2].

When it comes to multi-agent systems like smart cities, trust is one of the most important factors that enables cooperation and communication to occur without friction. It's possible that traditional trust management systems will have difficulty dealing with the scope and complexity of interactions among autonomous entities in these kinds of dynamic settings. In this context, the blockchain technology presents itself as a potentially useful option.

Blockchain technology [3] was first developed to serve as the foundation of cryptocurrency, but it has since developed into a sophisticated platform that can handle data in a way that is safe, transparent, and decentralized. The inherent difficulties with trust that are presented by Internet of Things (IoT) systems in smart cities are addressed by the key qualities of blockchain technology, such as immutability, transparency, and consensus. By using the characteristics of blockchain technology, many agents within the IoT ecosystem are able to develop and retain confidence in one another, which in turn helps to promote interoperability, security, and accountability.

In this work, we investigate the feasibility of applying blockchain technology [4] in multi-agent scenarios inside smart cities as a trust management model using a blockchain trust management model. We examine the ways in which the decentralized design of blockchains might help to build trust amongst a variety of Internet of Things devices, smart sensors, autonomous cars, and other stakeholders. In this section, we will focus on the following important aspects:
The use of Smart Contracts for Automated Trust Execution Smart contracts allow agreements to self-execute according to predetermined terms and circumstances. We investigate the ways in which these programmable contracts might automate the execution of trust among agents, therefore assuring compliance with preset norms and fostering interactions that are frictionless. This article seeks to give significant insights into the creation of safe, trustworthy, and efficient multi-agent settings by evaluating the possible integration of blockchain technology into the Internet of Things landscape of Smart Cities. We foresee that Smart Cities will be able to become more resilient, sustainable, and adaptable to the ever-changing demands of their citizens and stakeholders if effective trust management is implemented.

2. BACKGROUND STUDY

2.1 Smart City

The term "Smart City authentication" refers to the processes and procedures that are used to validate the identities of persons, devices, and organizations that access services, data, and resources included inside a Smart City's physical infrastructure. In a Smart City, [5] the many devices, sensors, networks, and apps that are networked to one another gather and share data in order to increase the quality of life for people and improve the urban services available to them. However, [6] this interconnectedness also presents concerns to security and privacy, making sophisticated authentication procedures absolutely necessary to guarantee that only authorized organizations are able to access and exploit the services provided by Smart Cities [7].

Authentication in Smart Cities is mostly comprised on the following key aspects:

User Authentication is the process of ensuring that people who use Smart City services like public transit, healthcare systems, or online government portals are authenticated in a safe manner. This may include conventional techniques such as usernames and passwords, but it should also include more robust mechanisms such as multi-factor authentication (MFA), biometric authentication (fingerprint, face recognition, etc.), or even sophisticated behavioral analytics. Traditional methods such as usernames and passwords may be used.

Authentication of Devices refers to the process of verifying the identities of devices and sensors that have been installed throughout a Smart City's infrastructure. To guarantee that only authorized devices are able to connect with the central systems, it is necessary for each device to have a distinctive identity as well as credentials. In order to accomplish this goal, one can make use of device certificates, cryptographic keys, or secure authentication mechanisms.

Application Authentication is the process of ensuring that all of the apps and services that are part of the Smart City ecosystem are authenticated whenever they interact with one another. It is possible to leverage secure Application Programming Interfaces (APIs) as well as authentication tokens in order to allow secure communication across a variety of apps.

Implementing secure communication protocols, such as TLS/SSL, to encrypt data that is being communicated between devices, sensors, and backend systems in order to avoid eavesdropping and data tampering. Secure communication.

The implementation of RBAC methods to restrict and manage access privileges based on the roles and responsibilities of persons or devices is referred to as role-based access control, or RBAC for short. This guarantees that only the resources to which users and devices have been granted access may be accessed by those users and devices.

Concerns Regarding Privacy It is vital, in the context of a Smart City, to strike a balance between the necessity for identification and concerns regarding privacy. The authentication procedure should include the collection of data, all of which should be handled in a responsible manner and in accordance with any applicable privacy legislation.

Continuous Monitoring Authentication in smart cities should be supplemented by continuous monitoring and real-time analytics so that possible security risks and abnormalities may be quickly identified and dealt with.

Integration with Pre-Existing Systems: Authentication solutions for smart cities should be integrated without causing any disruptions to the pre-existing infrastructure in order to ensure that users have a positive experience without compromising safety.

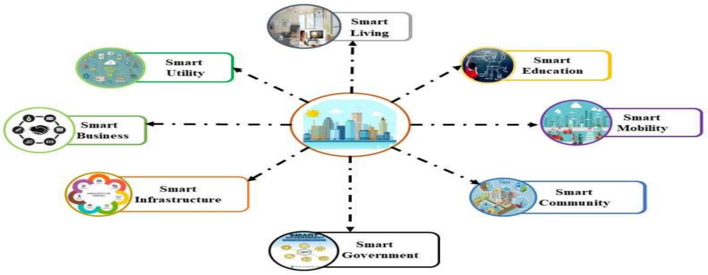


Figure 1. Principal defining features of so-called "smart cities".

Smart cities are urban settings that make use of technology, data, and creative solutions to improve the quality of life for people, boost sustainability, and optimize many elements of urban living. Smart cities are also known as connected cities. Although the meaning and use of the term "smart city" might shift from one region to another, there are a few key features that are almost always connected with the concept. These features are as follows:

Integration of Technology Smart cities are able to receive and analyze data from a wide variety of sources because they make use of cutting-edge technology such as the Internet of Things (IoT), sensors, artificial intelligence (AI), data analytics, and cloud computing. This method, which is driven by data, enables communities to make educated choices and maximize the use of available resources.

Infrastructure that is Sustainable Smart cities place an emphasis on being sustainable and environmentally friendly. They put in place environmentally friendly practices in order to cut down on their energy use, carbon emissions, and trash generation. This may include solutions for waste management, sustainable transportation systems, energy-efficient structures, and renewable energy sources.

Transportation Systems That Are Efficient Smart cities place a priority on having transportation systems that are both efficient and accessible. This may incorporate sophisticated traffic management, real-time monitoring of public transit, bike-sharing programs, infrastructure for charging electric vehicles, and support for alternate modes of mobility.

Smart Governance: Smart cities enhance both their governance and the delivery of services by using technology. These include data-driven decision-making procedures, online government services, e-governance initiatives, and digital platforms for public involvement.

The provision of high-quality public services to the citizens of smart cities is one of the primary goals of these types of communities. This entails the use of efficient emergency response systems, educational technologies, intelligent management of utility systems, and intelligent healthcare systems.

Making Decisions Based on Data: In order to make educated choices about the distribution of resources, urban planning, and the improvement of services, smart cities depend on data analytics and insights. This method, which places a strong emphasis on data, contributes to more successfully tackling urban concerns.

Safety and Security Smart cities make it a priority to invest in cutting-edge security technologies to protect both their citizens and their guests. This may include emergency response networks, smart surveillance systems, and cybersecurity procedures designed to guard against cyber attacks.

Participation of Citizens: Participation of residents is essential to the development of smart cities. They include inhabitants in decision-making processes and urban planning by encouraging active engagement from residents as well as soliciting input from them.

Innovation Ecosystem Smart cities stimulate innovation by providing assistance for technological companies, research institutions, and public-private collaborations. This is an example of an innovation ecosystem. They provide a climate that fosters the creation and use of innovative technologies and solutions by providing the necessary conditions.

Resilience and Adaptability: Smart cities are meant to be robust in the face of difficulties such as those posed by natural catastrophes, changes in climate, and social disturbances. They use flexible infrastructure as well as several backup plans in order to react quickly and efficiently to shifting conditions.

Inclusion of people in Digital Technologies and Services Smart cities work to eliminate digital exclusion by ensuring that all people have access to the various digital technologies and services available. People of varying abilities and socioeconomic backgrounds are targeted in the accessibility and usability research and development efforts that are now underway.

2.2 Overview of Blockchain Technology

The blockchain is a decentralized and distributed ledger system that allows safe and transparent record-keeping of transactions across numerous parties. Blockchain technology was developed by cryptocurrency pioneer Satoshi Nakamoto [8]. It was first presented as the core technology that supported the digital currency Bitcoin, but since then, its potential uses have expanded much beyond the realm of cryptocurrencies. Creating a chronological chain of blocks, each of which contains a list of transactions or data, and linking them together via the use of cryptographic methods is the fundamental idea behind blockchain technology [9].

The following is an explanation of how the blockchain technology operates as well as its primary characteristics:

Decentralization: Unlike conventional centralized systems, which allow a single authority to keep control over all of the data and transactions, blockchain functions on a network of nodes that is not governed by a single authority. Consensus procedures are used to verify and reach an agreement on the state of the distributed ledger while a copy of the whole blockchain is stored on each node.

Immutable and transparent: once data has been recorded on a blockchain, it is very difficult, if not impossible, to change or remove the information. Because it cannot be changed, the information can always be relied on for its honesty and reliability. In addition, the openness of blockchain technology makes it possible for anybody who has access to the network to observe the whole transaction history, which encourages accountability.

Security Provided by Cryptography Each block in the blockchain is connected to the next using cryptographic hashes. This ensures the data's authenticity and prevents unauthorized changes from being made. The transactions are digitally signed, which provides a high level of security and ensures their legitimacy.

Proof of Work (PoW) and Proof of Stake (PoS) are examples of consensus procedures that are used in order to verify and reach an agreement on the current state of the blockchain. These methods prevent malevolent actors from obtaining control over the network and guarantee that all nodes achieve a consensus on the legitimacy of transactions. Additionally, they ensure that all nodes communicate with one another.

Smart Contracts Blockchain systems like as Ethereum were the first to propose the notion of smart contracts, which are contracts that can execute themselves and come with predetermined conditions and restrictions. When certain criteria are satisfied, these contracts will automatically carry out their terms, doing away with the need for middlemen while also expediting a variety of corporate procedures.

Privacy Options Despite the fact that the immutability and transparency of blockchain technology are advantageous for public ledgers such as cryptocurrency, privacy options are necessary for certain business applications. There are a variety of privacy choices available throughout the different blockchain networks. These include private blockchains, which limit access to just those members who have been allowed, and permissioned blockchains, which have regulated access.

Interoperability: Currently, there are active efforts being made to facilitate interoperability across various blockchain networks. Interoperability would make it possible for data and transactions to be transferred across different blockchain systems in a frictionless manner, which would drive wider adoption and integration.

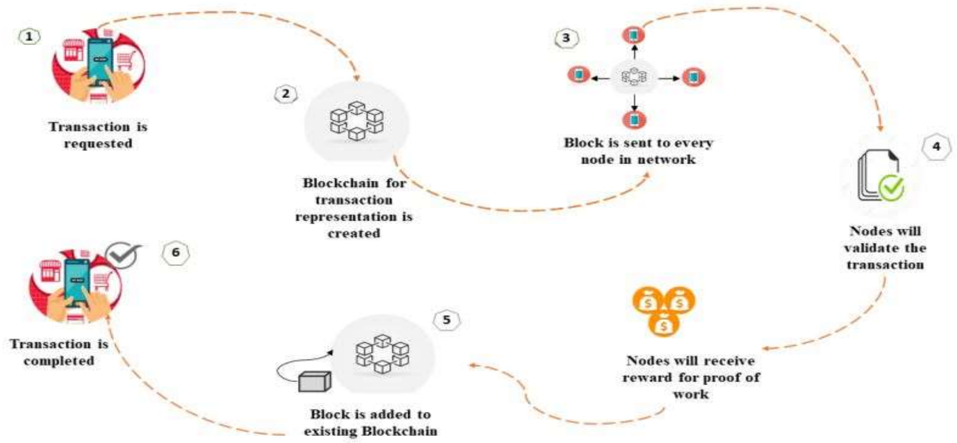

1
Transaction is requested
2
Blockchain for transaction representation is created
3
Block is sent to every node in network
4
Nodes will validate the transaction
Nodes will receive reward for proof of work
5
Block is added to existing Blockchain
6
Transaction is completed

Figure 2. The transaction procedure using blockchain.

The recording and verification of transactions on a blockchain network are required steps in the process of conducting blockchain transactions. In most cases, it consists of a set of processes that, when followed in order, guarantee the data's integrity, security, and consensus status. The following is an explanation of how transactions are processed using blockchain technology:

Initiation: When a user starts a transaction, a digital record of the transaction data is first created on their device. This might entail the transfer of bitcoin, an update to the ownership of an asset, the execution of a smart contract, or any other activity that changes data on a blockchain.

Verification and Digital Signature: It is necessary for a transaction to be validated by the nodes in the network before it can be added to the blockchain. For the purpose of establishing ownership and establishing authenticity, the transaction has been digitally signed by the sender using their private key.

Broadcasting: After the transaction has been signed, the information about it is sent out to the whole network. When the transaction is received by other nodes on the network, those nodes add it to a pool of unconfirmed transactions that is sometimes referred to as the "mempool."

Confirmation and General Consensus: The process of verifying the legitimacy of the transaction is kicked off by miners (in networks that are based on PoW) or validators (in networks that are based on PoS). They begin by selecting a group of transactions that have not yet been verified from the mempool and then proceed to compile them into a block.

The next step is for miners and validators to compete against one another to solve a difficult mathematical problem, known as the Proof of Work in Proof of Work consensus mechanisms or another consensus mechanism in Proof of Stake consensus mechanisms. Because this procedure uses a lot of resources and demands a lot of computer effort, it is secure and resistant to being manipulated [10].

After a miner or validator has completed the challenge and successfully solved it, the new block is broadcast to the network, where other nodes then check that it is legitimate.

Addition of Blocks: Following the completion of the verification process, the new block will be added to the blockchain. It is then added to the ledger that cannot be altered and all of the nodes in the network update their local versions of the blockchain so that they reflect the newly added transaction.

Confirmation: In the majority of blockchain networks, a transaction is regarded as validated after it has been included in a certain minimum number of following blocks. The minimum number of confirmations that a transaction must have might vary widely depending on the consensus rules of the blockchain and the desired degree of protection for that particular transaction[11].

Finalization: Once the transaction is validated and uploaded to the blockchain, it is regarded as complete and cannot be changed after it has been added. Anyone who has access to the blockchain can now see the specifics of the transaction, including the sender, the receiver, and the amount. This makes the transaction completely open and transparent.

Execution After the Transaction Has Taken Place (Smart Contracts): When it comes to smart contracts, the execution of the conditions that have been established inside the contract takes place automatically after the transaction that is linked with the contract has been verified. When the predetermined criteria are satisfied, the smart contracts automatically carry out the activities that were agreed upon in the contract [12].

2.2.1 Ethereum Blockchain Architecture

Ethereum is a blockchain platform that is decentralized and open-source. It gives developers the ability to construct and deploy smart contracts and decentralized apps (also known as DApps). Late in 2013, Vitalik Buterin made the first suggestion, and on July 30, 2015, it was introduced to the public. The architecture of Ethereum's blockchain was developed to be adaptable, programmable, and versatile. This makes it possible to use the blockchain for a broad variety of purposes in addition to the traditional use of digital currency transactions. The architecture of the Ethereum blockchain may be broken down into these essential components:

Blockchain: The blockchain is the underlying technology that powers Ethereum. It is a distributed, public ledger that records all transactions and the execution of smart contracts. It functions in a manner similar to that of the blockchain that underpins Bitcoin, in which data blocks are linked together by cryptographic hashes to create a chain of blocks. The blockchain used by Ethereum, on the other hand, is not only meant to record transactions, but also incorporates capabilities for the execution and storing of code [13].

Contracts that Execute Themselves Smart contracts are contracts that are self-executing and are authored in computer languages such as Solidity. These contracts include preset rules and circumstances. They are at the core of Ethereum's adaptability and make it possible to build complicated apps that are decentralized. When certain criteria are satisfied, smart contracts will automatically carry out its terms without the need for any middlemen. They make it easier for users to connect with decentralized applications (DApps) and enable a broad variety of features, including as the generation of tokens, protocols for decentralized financing (DeFi), governance systems, and many more.

The Ethereum Virtual Machine (EVM) is a runtime environment that runs the bytecode of smart contracts. It is also known as the Ethereum Virtual Machine. It is a piece of software that is deployed on every node in the Ethereum network and is responsible for ensuring uniformity in the execution of smart contracts throughout the whole system. For Ethereum applications to function in a decentralized manner, the Ethereum Virtual Machine (EVM) is an essential component.

Gas and Ether (ETH): Gas is a unit of measurement that represents the processing power that is necessary to carry out operations or smart contracts on the Ethereum network. Ether (ETH) is the cryptocurrency that powers the Ethereum network. Each action carried out inside a smart contract uses up a certain quantity of gas. To perform transactions or engage with smart contracts on the network, users are required to pay gas costs in the cryptocurrency Ether (ETH). The payment of gas prices discourages hostile players from flooding the network with loops that are inefficient or limitless [14].

Ethereum is now running on a Proof of Stake (PoS) consensus mechanism that is being referred to as Ethereum 2.0. In the past, it reached consensus by a process known as Proof of Work (PoW), which is quite similar to Bitcoin. In PoS, validators are selected to propose and verify new blocks based on the amount of ether they "stake" as collateral. This amount is determined by the total quantity of ether in circulation. When compared to PoW, PoS is more energy-efficient and offers better transaction throughput than the latter.

Clients: Ethereum is comprised of a wide variety of client implementations (pieces of software) that are responsible for running the Ethereum protocol and keeping the blockchain updated. Examples of this include Geth (Go-Ethereum), which was written in Go, and Parity, which was developed in Rust. These clients verify transactions and smart contracts in addition to ensuring that the network is in consensus.

Interoperability and Standards Ethereum is compliant with a number of standards and protocols, including ERC-20 (for fungible tokens), ERC-721 (for non-fungible tokens), and ERC-1155 (for multi-fungible tokens). These standards and protocols allow Ethereum to interact with other blockchains and cryptocurrencies. On the Ethereum network, the many smart contracts and decentralized applications may interact with one another more easily as a result of these standards.

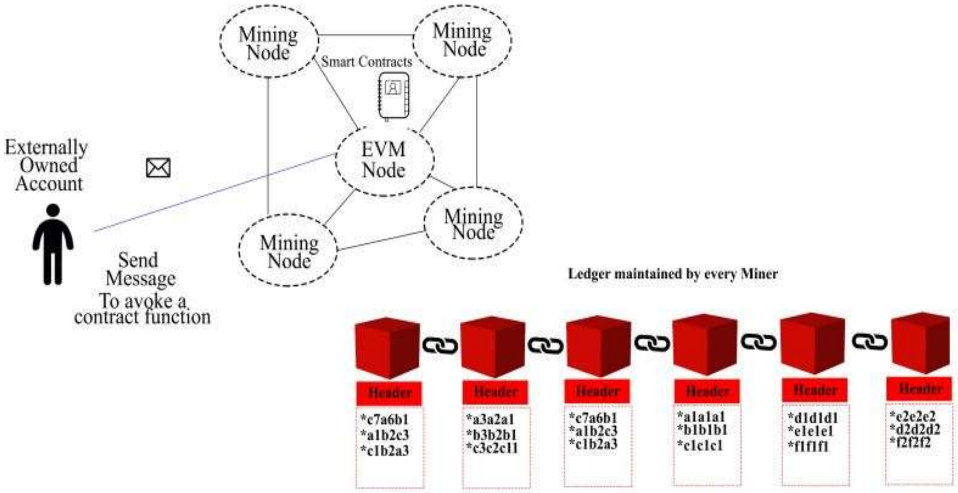


Figure 3. Ethereum Blockchain technology.

The term "parent-child relationship" is used to describe the way that blocks are connected together to create a chain in the context of blockchain technology. A linear and chronological order of blocks is created as a result of the fact that each block in the blockchain includes a reference to the block that came before it. This connectivity guarantees the completeness as well as the safety of the blockchain as a whole. Let's go into more detail about the parent-child connection that exists between the blocks that make up a blockchain:

Block Structure:A header and a body make up each individual block that makes up the blockchain. The header stores information, which includes the hash, or unique identity, of the block, as well as the date, the hash of the block that came before it (the hash of the parent block), and a nonce, which is a random integer that is utilized in the mining process. The real data, such as those pertaining to transactions or smart contracts, are included inside the body.

The parent-child linkage and hashing of data:The header of a new block must first go through a cryptographic hashing procedure before the block can be added to the blockchain. This process results in a string of characters with a predetermined length that is most often expressed as a hexadecimal number. The block may be identified by using this hash, which is a one-of-a-kind identifier [15].

The hash value of the previous block, known as the parent block, is provided in the header of the new block. This helps to ensure that the parent-child connection is preserved. This establishes a connection between the current block and the block that it is a child of.

Genesis Block:The first block that is added to a blockchain is referred to as the "genesis block." Since it is the first block in the chain, it does not have a parent block since it is intended to be the basis for the whole blockchain. In most cases, the hash of the genesis block is hard-coded into the software that runs the blockchain in order to start the chain.

Blocks strung together in a chain:The hash value of a newly mined block or one that has been added to the blockchain is computed and stored in the header of the block that comes after it. This technique is repeated with each successive block, thereby linking the blocks together in the order that they occurred in time.

Unchangeable and resistant to manipulation:The immutability and tamper-resistance of the blockchain is due to the parent-child connection as well as the cryptographic hashing process. If the data in any block were changed, the hash of that block would be invalidated, which would mean the hashes of all following blocks would also be invalid. As a consequence of this, the whole network would identify the altered blockchain as invalid, so preserving the authenticity of the initial blockchain and ensuring its continued use.

Longest Chain Rule:In the event that competing chains are created as a result of simultaneous block production, the network will adhere to the "longest chain rule." The nodes in the network choose the chain that has gathered the greatest computational effort (also known as "Proof of Work") or stake (also known as "Proof of Stake"). This guarantees that all nodes establish an agreement on the status of the blockchain and agree on the version of the blockchain that is considered genuine.

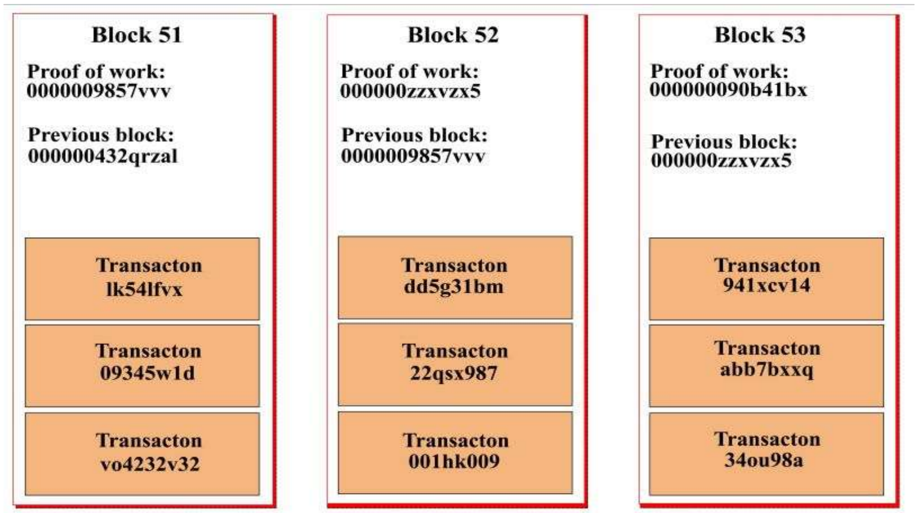


Figure 4. Blocks in Blockchain have a connection similar to that of a parent and a kid.

In a blockchain, transactions and blocks are intricately linked to one another, serving as the core support structure for the distributed ledger system as a whole. The immutability, integrity, and security of the blockchain are directly correlated to the link that exists between transactions and blocks. First, let's investigate the relationship between blocks and transactions on a blockchain:

Transactions: Transactions are the individual pieces of data that reflect activities or operations on a blockchain. A blockchain is a distributed ledger that stores data in a distributed ledger. For instance, transactions on a blockchain for a cryptocurrency like Bitcoin or Ethereum include the movement of digital assets (coins or tokens) from one address to another. Additionally, extra data or instructions for the execution of smart contracts may be included in transactions.

Structure of the Block: A block in a blockchain is a collection of several transactions that have been grouped together. Transactions that have been verified and authorized by the network are normally included in each block once they have been processed [16].

Block Creation: Transactions that are started by users are broadcast to the network and are initially unconfirmed. This is how blocks are created. These unconfirmed transactions are

gathered together by miners (in blockchains that are based on Proof of Work) or validators (in blockchains that are based on Proof of Stake), who then bundle them into a candidate block.
Confirmation of Transactions In order for a block to be added to a blockchain, the transactions that it contains first need to be confirmed. In the Proof of Work model, the process of confirmation is called mining. In the Proof of Stake model, the procedure is called validation. Miners and validators compete against one another to solve difficult cryptographic riddles or achieve a consensus on whether or not transactions are genuine, depending on which role they play.

The process of adding a block to the blockchain occurs once a miner or validator has successfully confirmed the contents of a block. The block stores the transactions that have been verified in addition to a reference to the hash of the prior block. This creates a connection between the new block and the block that was most recently added to the chain.

Blocks strung together in a chain:The act of adding new blocks to the blockchain is an ongoing operation that takes place over the course of time. Each new block is connected to the block that came before it, creating a chain that is ordered chronologically. The order and integrity of the transactions that take place inside the blockchain are protected by this chain of blocks.

Block Size and Frequency: The amount of transactions that may be included inside a block is determined by the block size, which can change amongst blockchain networks. The frequency at which blocks are generated is also variable. For instance, the size of a Bitcoin block is capped, but an Ethereum block's capacity might range anywhere from 8 to 32 megabytes.

Completeness of the Transaction: When a transaction is uploaded to the blockchain once it has been included in a block, it is regarded as confirmed and becomes unchangeable at that point. It is then included in the immutable and open-source record of previous transactions that the blockchain keeps.

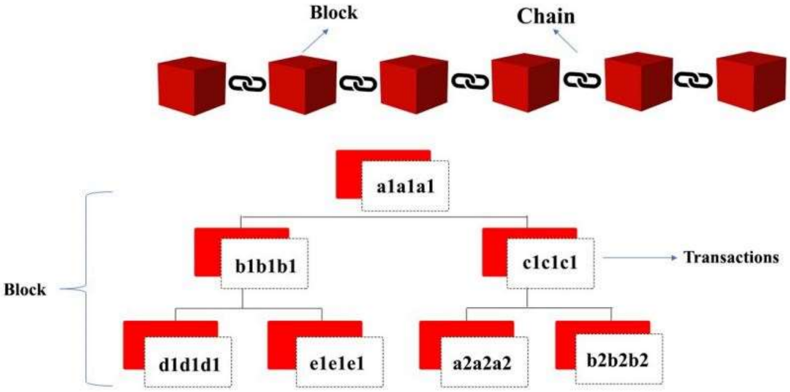


Figure 5. The connection between individual transactions and the blockchain's blocks.

The block header is an important part of each block in a blockchain, and it is where critical information is stored. This metadata contributes to the blockchain's continued authenticity and safety by keeping the blockchain's integrity intact. The actual data (transactions) that make up a block are preceded by a data structure with a predetermined size known as the block header. It is vital for joining blocks together in a chronological and tamper-resistant chain, which is why it is so important. The following is an example of a typical block header's contents:

Block Number: Each block in the blockchain is given its own unique identity, which is referred to as the block number. It is used to determine the sequence in which the blocks occurred in time and serves as a representation of the block's location within the whole chain.

Previous Hash of the Block The hash of the block that came before it in the blockchain is the cryptographic hash of the header of the block that came before it. It creates the connection between the current block and the one before it, so establishing the order of the blocks in chronological order.

Merkle Root The Merkle root is a hash that is generated cryptographically from all of the transactions that are contained in the current block. It is a summary of the data that was associated with the transactions and enables fast verification of the transactions that took place inside the block.

Timestamp: The timestamp provides information about the precise moment that the block was added to the blockchain or mined. It is of assistance in calculating the amount of time that has gone between the production of successive blocks.

In blockchains that employ the Proof of Work (PoW) consensus algorithm, the act of mining a block generates a random number called the nonce. Miners make an effort to locate a nonce value that, when paired with the other data in the block header, results in a hash that has certain qualities (for example, a hash that begins with a predetermined amount of leading zeros).

Target Difficulty The difficulty target is a number that defines the amount of complexity necessary for miners to obtain a valid block hash. Miners need to find a valid block hash in order to earn cryptocurrency. In addition to ensuring that new blocks are added to the blockchain at a pace that is reasonably consistent, it also controls the average amount of time that it takes to mine a new block.

Chainwork: Chainwork is a measurement of the overall computational work that has been completed on the blockchain. This metric is especially important for proof-of-effort (PoW) blockchains. It is a way of verifying the blockchain's overall security and is a representation of the total of the challenges faced by each of the blocks that came before it on the chain.

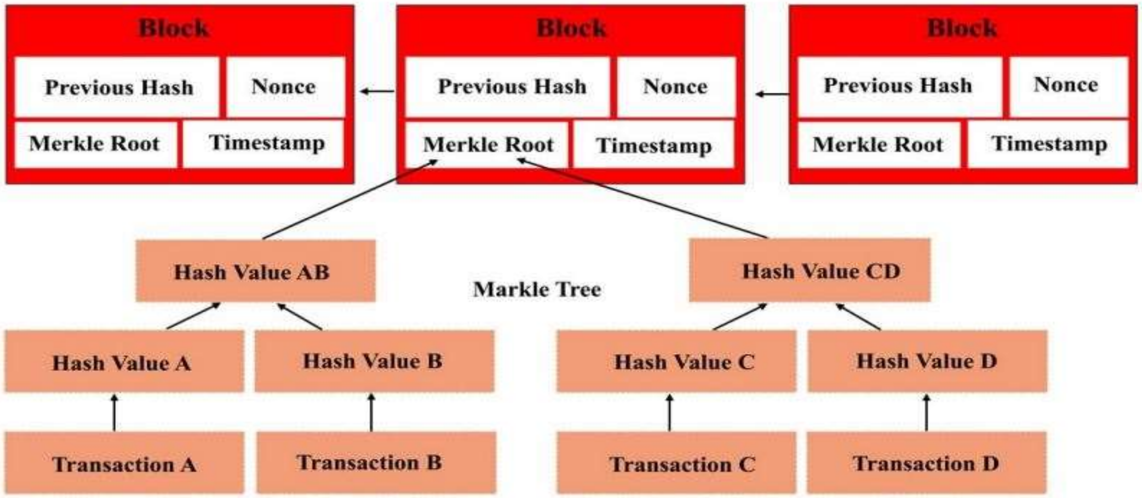


Figure 6. Block header and its contents.

3. RELATED WORK

This research study puts up an original multi-agent system as a solution for the control of traffic in smart cities. In order to gather information on the flow of traffic in real time, the system employs edge-of-things (EoT) devices such as smart sensors and cameras. In order to maintain trustworthiness, security, and transparency throughout the processing and management of the acquired data, blockchain technology is used. Through the use of automated decision-making among autonomous agents, the MA-AIM system intends to achieve the goals of increasing the effectiveness of traffic flow and decreasing congestion at crossings. The strategy that is based on blockchain technology ensures that there will be trust and accountability within the system, while also improving the coordination and collaboration between the many agents who are participating in the process of traffic management [17].This research investigates the use of blockchain technology in collaborative emergency management settings to investigate its possible benefits. The authors suggest a structure that employs blockchain to enhance the performance of emergency response systems by allowing multi-agent involvement. This would be accomplished by improving the systems' ability to track many participants. The solution that is based on blockchain technology guarantees that all of the various agents that are engaged in the emergency response process may communicate safely and openly with one another. It improves confidence and the integrity of data, making it possible for multiple agencies and stakeholders to collaborate and coordinate their efforts more effectively during times of disaster. The study emphasizes the advantages of using blockchain technology to maximize the effectiveness of emergency response operations and reduce the amount of time needed to respond [18].The purpose of this review article is to provide a detailed examination of the incorporation of blockchain technology into applications for the Internet of Things (IoT). The authors address the potential advantages of using blockchain technology in IoT contexts for ensuring the integrity of data, managing trust, and conducting secure transactions. They investigate many Internet of Things applications that are based on blockchain technology, such as supply chain management, smart energy systems, and healthcare, and they highlight open research issues in the implementation of blockchain for IoT, including scalability, interoperability, and energy efficiency [19].This research article presents a solution for job offloading in multi-Unmanned Aerial Vehicle (UAV)-assisted Internet of Things (IoT) networks that is enabled by blockchain technology. The authors make use of multi-agent deep reinforcement learning (DRL) in order to improve the choices about job offloading while taking into account the energy harvesting capabilities of UAVs. Utilizing blockchain technology allows for the establishment of trust and accountability between UAVs and other Internet of Things devices. The technique that has been suggested improves the overall performance as well as the energy efficiency of multi-UAV systems, all while guaranteeing the safety and dependability of job offloading in situations that are dynamic [20].

The authors of this conference paper offer a trust management system that may be used to multi-agent systems that are used in smart grids. The system makes use of blockchain technology to enable safe and reliable interactions between the many parties engaged in the administration and distribution of energy. The trust model that is based on blockchain technology improves the dependability and accountability of energy transactions, which ultimately results in a smart grid infrastructure that is more robust and secure. The issues that arise from a lack of confidence in complicated energy management situations are the focus of the system that is being suggested [21].In this chapter of the book, we will talk about how blockchain technology, the Internet of Things (IoT), electronic governance, and electronic democracy are converging to create smart cities. The authors investigate how blockchain technology may be used to make e-governance and e-democracy systems more trustworthy and open to public scrutiny by providing immutable records of transactions and votes. The combination of blockchain technology with internet of things further increases the openness and effectiveness of the operations of smart cities [22].In this research article, the authors suggest an access control system for Internet of Things settings that is based on blockchain technology. The authors make use of blockchain technology in order to provide a safe and decentralized access control system for Internet of Things devices and apps. [23] The technology offers efficient authentication and authorisation, which enhances the safety and privacy of Internet of Things (IoT) devices.This conference article provides a safe and reliable multi-agent system for the management of autonomous intersections. Within the context of smart cities, the authors make use of blockchain technology to improve trust, security, and collaboration in the management of crossings. Through the use of autonomous and transparent decision-making among agents, the MA-AIM system that has been developed intends to both maximize the flow of traffic and reduce the amount of congestion that occurs [24].This research study presents a novel methodology for securing Internet of Things-based smart home mobile agents by using blockchain technology. The authors suggest a strategy that makes use of blockchain technology as a means of improving the level of security and privacy offered by smart home devices and apps. The model makes sure that interactions between Internet of Things devices in a smart home setting are safe and trustworthy, hence reducing the risk of possible security breaches [25].

In this research article, a trust-aware energy trading system for smart grids is presented. It is supported by blockchain technology. The authors maximize the energy trading choices made by various agents in the smart grid by combining game theory, multi-agent systems, and blockchain technology. This technique, which is based on blockchain technology, improves trust and fairness in energy trade, which in turn encourages the effective allocation and usage of resources [26].The Internet of Things (IoT) is the topic of this research study, which presents a new demand-side management system as a potential solution. The demand-side management of energy systems may be made more safe and trust-distributed when using this system, which makes use of the blockchain technology. The objective of the proposed framework is to minimize energy usage and meet the highest possible demand, all while maintaining openness and safety in the interactions between different Internet of Things devices and interested parties [27].

In this piece of research, the authors introduce DITrust chain, which is a trust model for sustainable healthcare IoT systems that is built on blockchain technology. In Internet of Things-based healthcare applications, the approach assures that patient data and transactions are secure, private, and accountable. The DITrust chain that has been developed improves the reliability of healthcare IoT devices, which in turn facilitates improved patient care and data management [28].This research study explores how blockchain technology and artificial intelligence (AI) are converging in Internet of Things (IoT) networks to create sustainable smart cities. The authors investigate how the use of blockchain technology in conjunction with artificial intelligence might improve the safety, effectiveness, and long-term viability of smart city applications. The combination of these technologies enables safe and insightful administration of Internet of Things devices and data in urban settings [29].This study paper examines the difficulties associated with establishing connections between residents and smart cities. The authors examine the role that information and communication technology (ICT), electronic governance, and blockchain play in making it possible for smart cities to provide people with services that are both transparent and efficient. The purpose of this study is to investigate the use of blockchain technology in smart city applications from the perspective of increasing trust and safety [30].The writers of this article address the applications, possibilities, and problems that blockchain presents for edge of things (EoT) in the context of the article. [31] The purpose of this article is to investigate how blockchain technology might improve trust, security, and decentralization in edge computing settings, hence making data management and sharing more effective and dependable.

This research paper outlines an approach to the administration of trust for the Internet of Things (IoT) that is based on blockchain technology. The authors provide an innovative strategy that makes use of blockchain technology to assure safe and reliable interactions between IoT devices and apps. [32] The strategy improves the dependability and accountability of IoT systems, as well as facilitates seamless communication and cooperation across IoT devices.In this survey article, we offer an overview of how blockchain technology, Software-Defined Networking (SDN), and Network Function Virtualization (NFV) might be integrated for smart-home security. [33] The authors investigate the ways in which these technologies might be merged to improve the safety and privacy of smart homes and to ensure the safe and effective administration of Internet of Things devices and apps.

A distributed trust model for smart vehicle networks is presented in this conference paper, which takes its inspiration from blockchain technology. The authors suggest a unique solution that makes use of blockchain technology in order to create trust and ensure the security of communication between infrastructure and cars in intelligent transportation systems. The purpose of the model is to improve the dependability and security of communication when in motion [34].The purpose of this research paper is to provide a solution to resource allocation that is enabled by blockchain technology and is suitable for multi-UAV-enabled 5G Radio Access Networks (RAN). The authors improve resource allocation choices in dynamic wireless networks by combining multi-agent deep reinforcement learning (DRL) with blockchain technology. The solution that has been suggested attempts to increase the performance of the network as a whole, as well as its energy efficiency and general dependability [35].The authors investigate intelligent communication and service applications for the Internet of Things (IoT) that are enabled by blockchain technology in this paper. They examine the ways in which blockchain technology might improve trust, security, and privacy in IoT networks, hence making it possible for IoT devices to have communication that is both smooth and efficient [36].This research paper presents a framework for the trade of energy that is trust-aware and is enabled by blockchain technology. The authors maximize the energy trading choices made by various agents in the smart grid by combining game theory, multi-agent systems, and blockchain technology. This technique, which is based on blockchain technology, improves trust and fairness in energy trade, which in turn encourages the effective allocation and usage of resources [37].The Diamond accountability model is presented here as a solution for cyber-physical systems (CPS) that are enabled by blockchain technology. The authors suggest an innovative strategy that makes use of the technology behind blockchains in order to promote accountability and transparency inside CPS. This model's intention is to improve the safety and dependability of CPS by providing a record of interactions and data that cannot be altered in any way [38].The authors of this research study investigate the use of blockchain technology in conjunction with federated learning in the context of the Internet of Things (IoT). They highlight current breakthroughs in merging these technologies as well as future difficulties that need to be overcome in order to improve trust, privacy, and security in IoT networks. In this work, the potential advantages of combining blockchain technology with federated learning in a variety of Internet of Things applications are examined [39].

4. PROPOSED METHOD

Figure 7 depicts the proposed architecture, which utilizes Blockchain technology to ensure security and authorization for accessing smart city resources. The diagram showcases the key components of the architecture and outlines the step-by-step process for authorized access to these resources. To avoid any confusion regarding the taxonomy and functioning of different entities, the nomenclature adheres to the specifications outlined by the Internet Engineering Task Force (IETF).

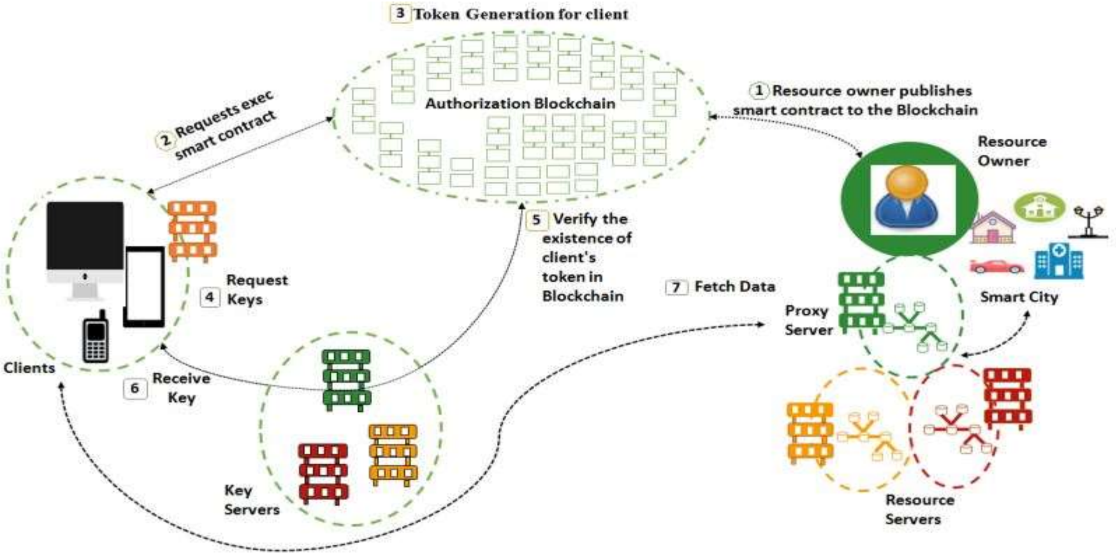

3 Token Generation for client
Authorization Blockchain
2 Requests exec smart contract
1 Resource owner publishes smart contract to the Blockchain
Resource Owner
Smart City
4 Request Keys
5 Verify the existence of client's token in Blockchain
7 Fetch Data
Proxy Server
Clients
6 Receive Key
Key Servers
Resource Servers

Figure 7. A schematic representation of the suggested solution.

- **Authorization Servers:** Their primary function is to generate access tokens.
- **Resource Servers:** These servers generate and store protected resources.
- **Proxy Servers:** Acting as access interfaces for clients, proxy servers store protected resources.
- **Resource Publishers/Owners:** These entities are the original owners of resource servers and the corresponding resources.
- **Key Servers:** Their role involves generating encryption/decryption keys for the protected resources.
- **Client/User Application:** Representing third-party entities, these applications request access to the protected resources.

4.1. Authorization Blockchain

The concept of blockchain may be conceptualized as an immutable ledger that stores records in blocks that are timestamped. These blocks are referred to by their cryptographic hash values, which uniquely identify them and reference the hash of the block that came before them. They are used to store transactions. The distributed character of the Blockchain is ensured by the fact that it is maintained by nodes that keep copies of the most recent n blocks.

When it comes to the Blockchain for authorisation, the nodes may be broken down into three categories: key servers, authorization servers, and clients. There is no need that every node keep a copy of the whole blockchain or take part in the consensus process. Full nodes, which include authorization servers and key servers, are responsible for storing the whole Blockchain. Authorization servers also perform the function of miners inside the network. Key servers are set up by resource owners and are accountable for the information on resource servers that is connected to keys. A unique Blockchain address, which is represented by a pair of asymmetric keys, is used to identify each node in the network.

The owner of the resource will first build a smart contract to explain the access rights, and the smart contract will then produce access tokens on its own. To engage with smart contracts or to promote communication between customers and resource owners, transactions are employed. These transactions are then broadcast to the network of nodes that make up the Blockchain after being signed with the nodes' individual private keys.

These transactions are subject to validation by the authorization servers, often known as miners, and storage in blocks. The sealing of these blocks is carried out with the assistance of a consensus mechanism, and after that, they are added to the Blockchain. In order to be trusted as legitimate, key servers and resource servers must each have a valid certificate in their possession.

When a client wants to decrypt certain resources, the client will connect with a key server in order to make the request. The client's authorisation is validated by the key server, which then investigates the blockchain for any smart contracts that may have been stored there. If the client can be trusted, the key server will hand over the necessary decryption key to the client. This will allow the client to successfully download and decode the resource of their choosing.

In this study, an authorisation Blockchain for smart city resources is constructed using the Ethereum [54] Blockchain network. Records accounts and information pertaining to consumers and contractors are timestamped and saved in blocks inside the Ethereum blockchain. It is a permissionless public Blockchain that is accessible to the public and designed for use with decentralized apps. The justifications for using the Ethereum Blockchain are as follows:

- A prominent and distinguishing feature of Ethereum is its programmability. It allows for the inclusion of contracts and agreements within its code, enabling automatic execution of transactions. Through its smart contracts, users can exchange valuable assets such as money, property, and more. These contracts have the capability to invoke other contracts and can be designed with multiple conditions and supported formats.
- Ethereum's scope extends beyond cryptocurrency transactions; it encompasses a wide range of applications, including supply chains, energy grids, real estate, government registries, and various other use cases.
- Ethereum has demonstrated its resilience against security attacks, making it a robust and secure platform.

- The Ethereum ecosystem is characterized by openness and flexibility, supporting both permissionless and permissioned implementations to cater to different needs.
- With the capacity to accommodate hundreds of nodes and millions of users, Ethereum surpasses many of its competitors in terms of scalability.
- Ethereum consortia are not reliant on a single vendor; instead, they are interoperable, allowing for collaborative and decentralized networks.

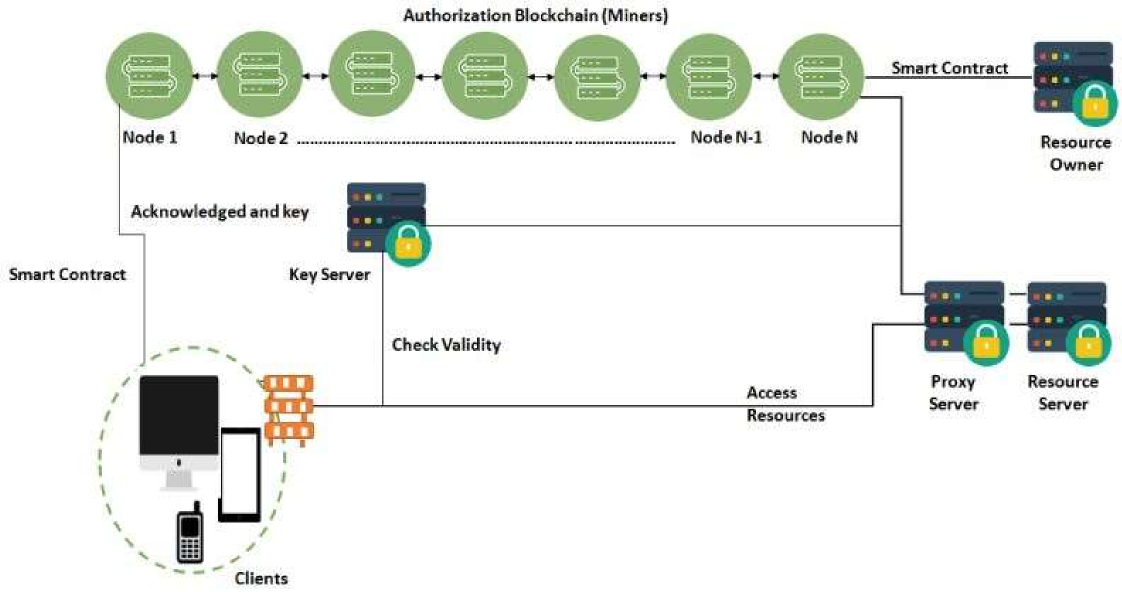
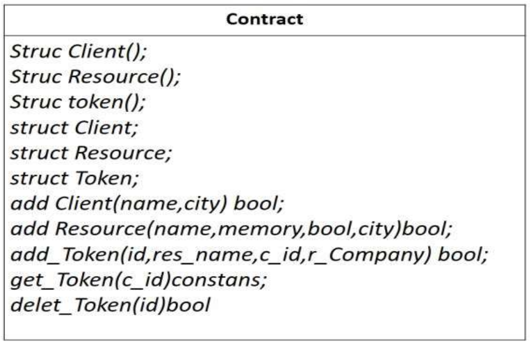


Figure 8. Blockchain infrastructure.



```
Contract
Struc Client();
Struc Resource();
Struc token();
struct Client;
struct Resource;
struct Token;
add Client(name,city) bool;
add Resource(name,memory,bool,city)bool;
add_Token(id,res_name,c_id,r_Company) bool;
get_Token(c_id)constans;
delet_Token(id)bool
```

Figure 9: The following is an example of a smart contract that is currently being executed on the blockchain.

4.2. Resource Servers

Resource servers are responsible for the storage and generation of the protected resources of smart cities. The resources that go into creating smart cities include things like traffic lights, security surveillance cameras, power plants, components of the water supply network, and many other things.

4.3. Resource Owners/Publishers

The resource owners have the legal right to ownership of the resource servers and any resources that are created by such servers. These resource owners are often government agencies or other governing bodies in smart cities.

4.4. Overall Working Flow

The generation of the necessary keys that are used to encrypt and decode resources is the responsibility of the key servers. When users complete the terms of the contract, a unique key is issued for them to use for a certain amount of time. This key grants them access to the resources for the allotted time period.

4.5 Proposed Algorithm

Step 1 : Smart Contract Creation

- Develop a smart contract that will handle IoT device trust checking on the Blockchain network.
- Define the necessary data structures and functions for trust verification.

Step 2 : IoT Device Registration

- Allow IoT devices to register themselves on the Blockchain network as participants.
- Each device generates a unique identifier and provides essential information like device type, location, and public key.

Step 3 : Trust Initialization

- Initialize a trust score for each registered IoT device in the smart contract.
- Set an initial trust level, such as 100, indicating full trust.

Step 4 : Interaction Recording

- Record each interaction between IoT devices and other participants in the Blockchain ledger.
- Include details like timestamp, transaction type, and the parties involved.

Step 5 : Trust Update Mechanism

- Implement a mechanism for updating the trust scores of IoT devices based on their behavior and interactions.
- Define factors that contribute to trust, such as successful transactions, reliability, and data accuracy.

Step 6 : Trust Evaluation

- Periodically evaluate the trust scores of IoT devices based on their recorded interactions.
- Calculate the trust score by analyzing historical data and interactions.

Step 7 : Trust Thresholds

- Set trust thresholds to determine different levels of trustworthiness for IoT devices.
- For instance, a trust score above 70 might be considered trustworthy, while below 50 may indicate low trust.

Step 8 : Trust Verification

- When a new interaction occurs involving an IoT device, the smart contract verifies the trust level of the device.

- If the trust score falls below a specified threshold, additional security measures or restrictions may be imposed.

Step 9 : Handling Trust Violations

- Implement mechanisms to handle trust violations or malicious behavior.
- For instance, temporarily suspend a device's access or notify administrators of suspicious activities.

4.6 Trust criteria

1. **Transaction History:** Analyzing the historical behavior and interactions of the IoT device in previous transactions. A positive track record of successful and reliable transactions may increase trust.
2. **Data Accuracy:** Assessing the accuracy and consistency of the data provided by the IoT device. Consistently providing accurate and reliable data may enhance trust.
3. **Reliability:** Evaluating the device's consistency in fulfilling its intended functions and obligations. Reliable devices are more likely to be trusted.
4. **Security Measures:** Examining the security measures implemented by the IoT device to protect sensitive data and prevent unauthorized access.
5. **Device Integrity:** Ensuring the device has not been compromised or tampered with, ensuring the authenticity of its operations.
6. **Endorsements and Review**s: Considering endorsements or positive reviews from other trusted entities within the system.
7. **Identity Verification:** Verifying the identity of the IoT device and ensuring it is a legitimate and authorized participant.
8. **Compliance with Regulations:** Ensuring that the IoT device complies with relevant regulations and standards.
9. **Consistency:** Evaluating whether the device's behavior is consistent with its declared capabilities and intended purposes.

10. **Response Time:** Measuring the device's responsiveness to requests and interactions. Faster response times may contribute to higher trust.
11. **Error Handling:** Assessing the device's ability to handle errors and recover from failures.
12. **Communication Protocols:** Evaluating the use of secure communication protocols to safeguard data during transmission.

5. EXPERIMENTAL ANALYSIS AND DISCUSSION

Within this part, we will present an analysis of the effectiveness of the essential components that make up the suggested framework.

5.1. Evaluation of Key Server and Resource Server : The performance of the many components that make up the Blockchain-based security mechanism is evaluated using a variety of tools and technologies, such as the Mocha framework and the Chai assertion library. This is done in order to determine how well the individual components function together.

Mocha is a JavaScript-based framework that is used for testing each individual module as well as evaluating its overall performance. On the other hand, Mocha makes use of the Chai assertion library to allow GET, POST, or PUT actions across the servers. To be more specific, the GET method is used to make a request for data from a particular resource, while the PUT method is used to transfer data to a server, which enables the construction of a resource or the update of an existing one.

The amount of time that, on average, it takes to complete the handshake procedure is depicted in Figure 10. It is unmistakable that the typical amount of time spent shaking hands lengthens with an increase in the number of customers. To be more specific, the average amount of time spent shaking hands with 60 customers is 200 milliseconds, but this amount of time more than doubles to 400 milliseconds when there are 120 customers.

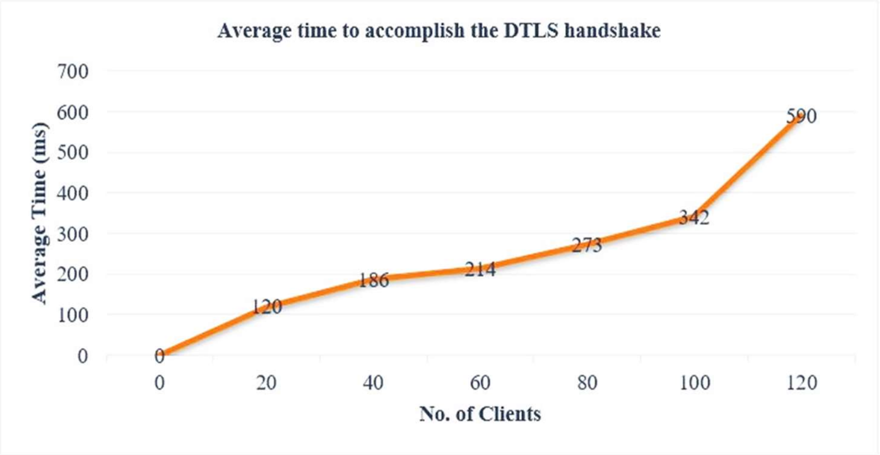


Figure 10. The average amount of time required to complete the DTLS handshake.

Figure 11 illustrates the response times, measured in milliseconds (ms), for a PUT and GET request made by a client to a resource server. Additionally, it displays the time taken to complete the DTLS handshake between the resource server and the key server, represented by the AUTH operation.For the PUT operation, the slowest observed response time was 12 ms, the average response time was 23 ms, and the fastest response time was 174 ms.Regarding the GET operation, the quickest response time recorded was 68 ms, the average response time was 146 ms, and the longest response time was 487 ms.Concerning the AUTH operation, the shortest completion time was 34 ms on record, the average completion time was 67 ms, and the longest completion time was 98 ms.

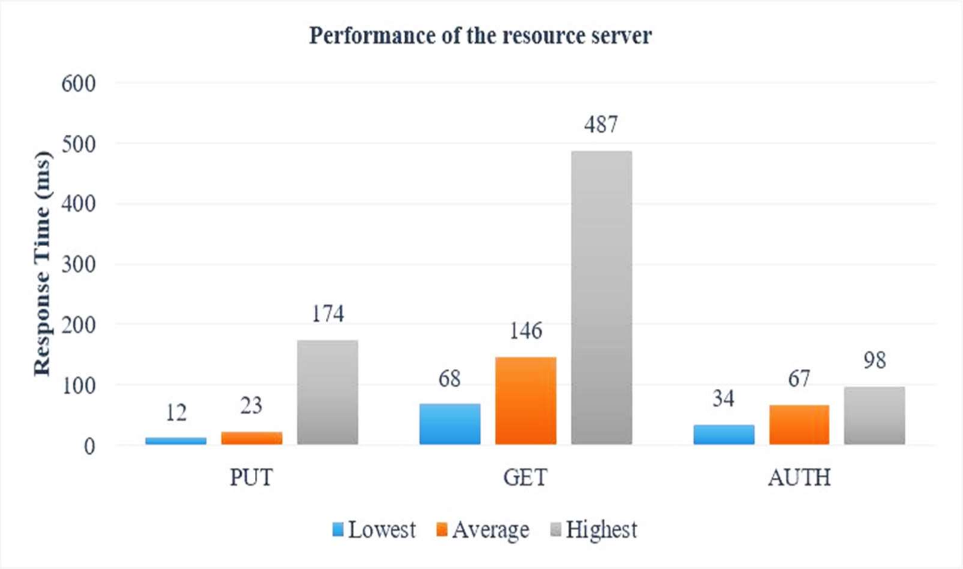


Figure 11. The effectiveness of the resource server.

In order to carry out an exhaustive analysis of the efficacy of the system that was presented, a number of experiments were carried out. These tests included the modification of several factors, such as the transaction rate and the block size. To be more specific, the transaction rate, also known as TR, was adjusted to a range that included anything from fifty to three hundred transactions per second (transactions/sec). For each TR value, the average latency, which will be indicated by the symbol LTAVG, was computed. Figure 12, which depicts the transaction commit time, shows the outcomes of the experiment that we did.According to the data, there is a direct correlation between an increase in the transaction rate and a corresponding rise in the average latency (LTAVG).

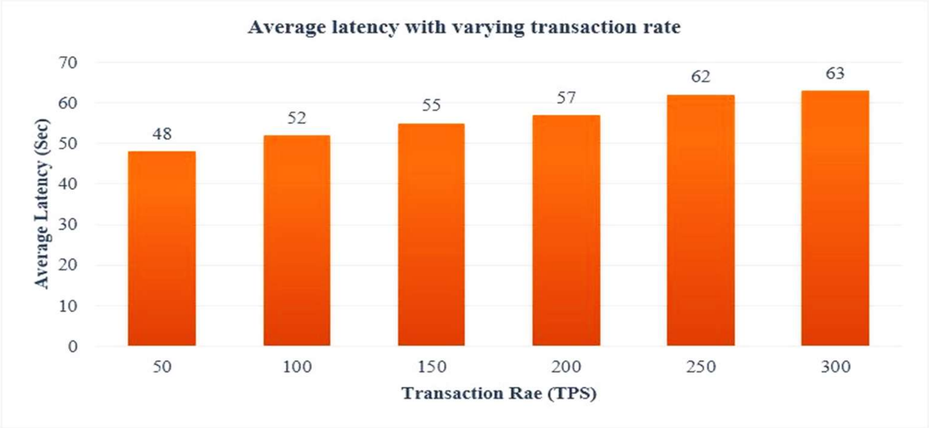


Figure 12. Variable average latency and transaction rate.

In the second experiment, we kept the transaction rate (TR) at a constant level of 150 while progressively increasing the number of nodes that were responsible for completing these transactions from 1 to 8. The visual representation of the transaction average may be seen in Figure 13.The findings make it abundantly evident that an increase in the average latency (LTAVG) is going to occur if there is a rise in the number of nodes that are taking part in transactions. When just one node is processing all 150 transactions per second, the LTAVG is at its lowest point; however, it reaches its greatest point when eight nodes are working together to process all 150 transactions.

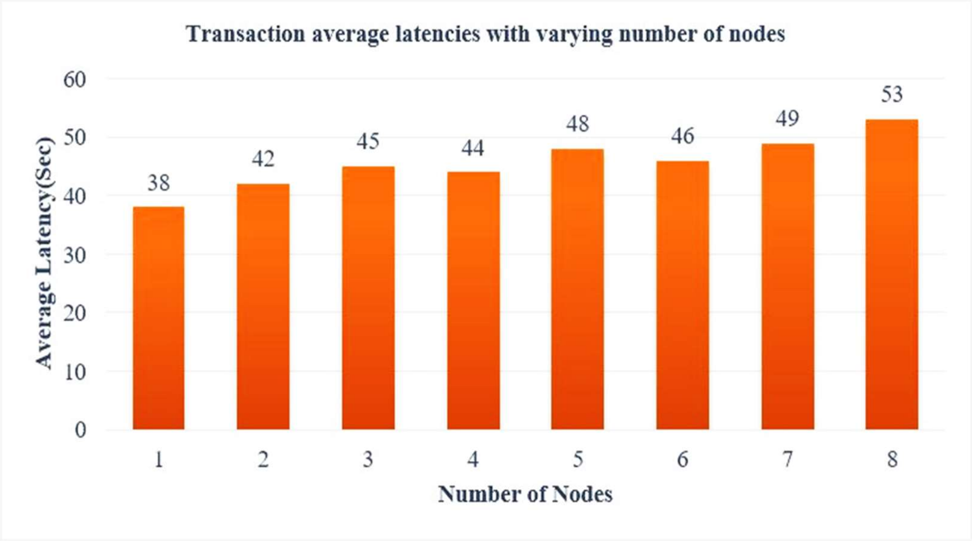


Figure 13. Average transaction latency with varying numbers of nodes.

The transaction throughput drops down noticeably as the number of nodes that take part in the transaction rises, as shown in Figure 14. This is a tendency that can be seen.

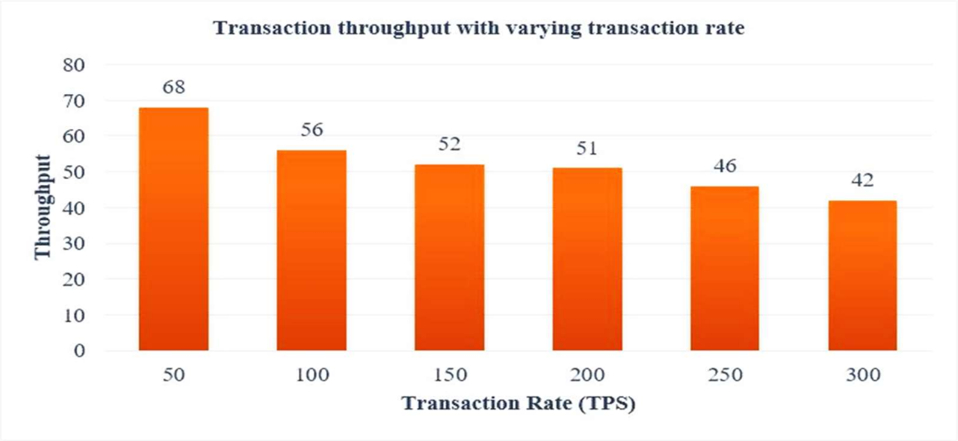


Figure 14. The throughput of transactions despite a variable transaction rate.

Figure 15 illustrates a further finding that is worthy of attention about the use of the CPU when the transaction rate (TR) is changed. It indicates that the CPU use steadily rises with an increase in TR, despite the fact that the block size remains same.

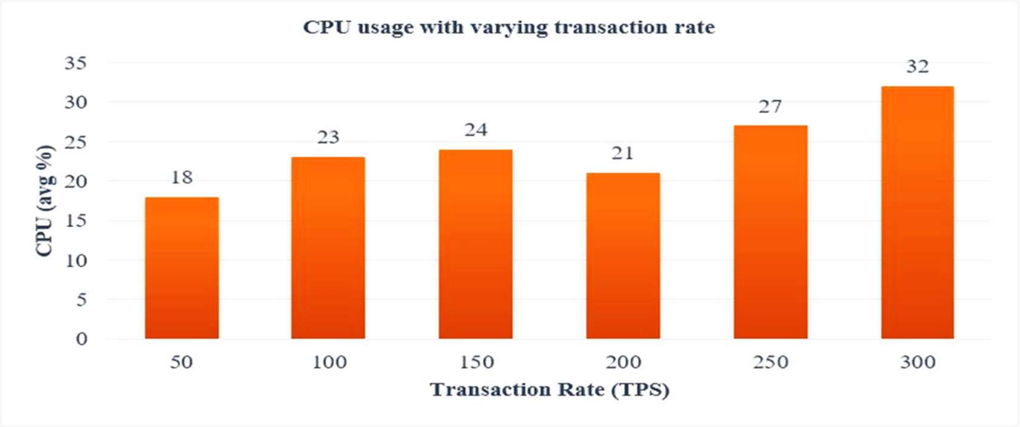


Figure 15. CPU consumption with a variable number of transactions per second.

Throughout the whole of the aforementioned trials, both the transaction size and the block size were kept unchanged at their default levels of 2 transactions per block. Read mode was used for each and every transaction that was processed. In the following collection of observations, we experimented with various block sizes while measuring the average latency (LTAVG) for a variety of transaction rates (TR). Figure 16 illustrates the findings of the study.In the beginning, the block size was determined to be 5, and the LTAVG values were determined by adjusting TR. After then, the size of each block was raised to 10, and the LTAVG values that corresponded to that increment were recorded. It is important to note that a block size of 10 displayed a little lower latency than a block size of 5. This can be seen in the observations. It is essential to emphasize once again that the read transaction mode was kept throughout all of the trials.

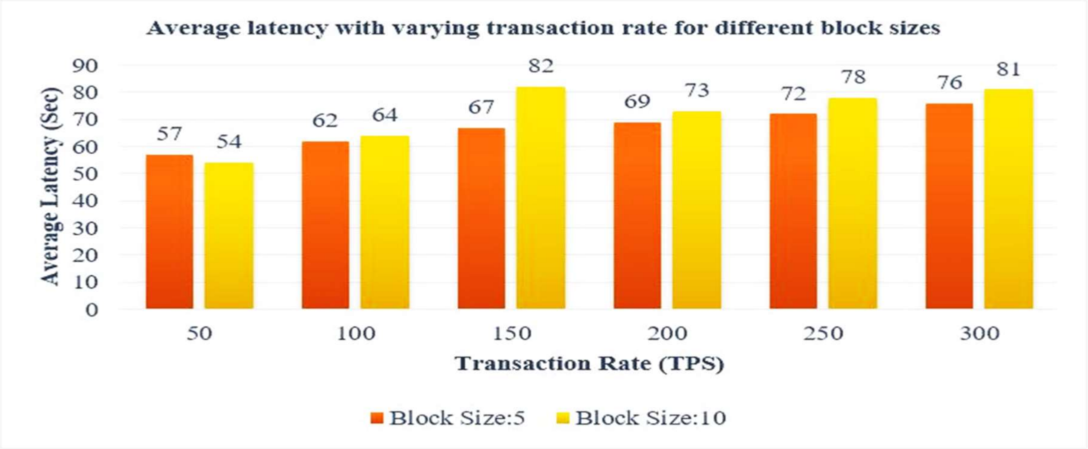
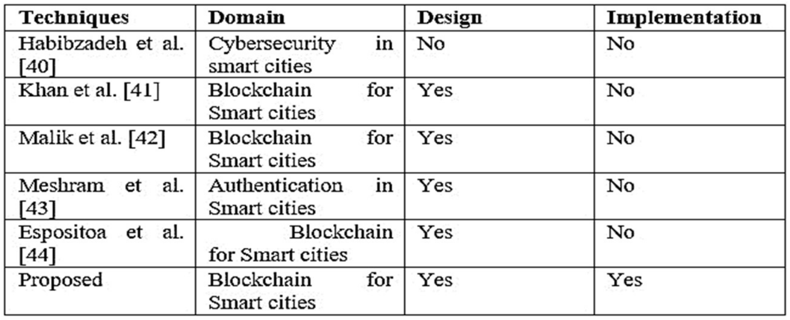


Figure 16. The average delay, with the transaction rate ranging across the board for the various block sizes.

5.2. Comparative Analysis

In this research, a comparative comparison of suggested and current approaches is carried out. The techniques are assessed in terms of the domain for which they are intended, the design, the implementation, and the user interface. The aspects that are covered by both the proposed solution and the current solutions are outlined in Table 1, which reveals that the majority of the existing solutions are missing some key features. The work that has been presented, on the other hand, stands out since it takes an all-encompassing approach that incorporates design, implementation, and user application.This paper presents a technique for ensuring the safe authorisation of smart city resources that is based on Blockchain technology. This method includes the design and implementation of an authorisation Blockchain that is based on the Authentication and Authorization for Constrained Environments (ACE) framework, as well as the Object Security Architecture for the Internet of Things (OSCAR) object security model. The deployment of blockchain technology offers a decentralized permission system that is both flexible and trustworthy. On the other hand, OSCAR makes use of a public ledger to set up multicast groups for clients who have been granted permission to do so.

Table 1 : A comparison of the proposed framework with already existing solutions based on the features of both.



Techniques	Domain	Design	Implementation
Habibzadeh et al. [40]	Cybersecurity in smart cities	No	No
Khan et al. [41]	Blockchain for Smart cities	Yes	No
Malik et al. [42]	Blockchain for Smart cities	Yes	No
Meshram et al. [43]	Authentication in Smart cities	Yes	No
Espositoa et al. [44]	Blockchain for Smart cities	Yes	No
Proposed	Blockchain for Smart cities	Yes	Yes

6. CONCLUSIONS

This article presents a dependable and safe authentication and trust management technique that has been adapted specifically for smart cities. A dependable infrastructure for the safe authorisation of smart city resources is provided by the suggested solution, which makes use of the potential offered by the Blockchain technology. It is illustrated how a realistic implementation of this Blockchain-based security mechanism can be carried out, which enables increased security measures for the resources of smart cities. In addition, a hybrid application is built in order to improve user experience and accessibility. This application provides a user-friendly interface and is able to accept a variety of technologies that are used inside smart cities. Users are given the ability to engage with and exert control over a variety of smart city technologies, such as traffic lights and surveillance systems, via the usage of this program.As we look to the future, the work that we have done has a strong potential for development into a variety of fields, including healthcare, hospitality, pharmaceutical, education, and other fields. If this method were modified and used across a wider range of industries, it may be possible to further contribute to the development of smart city technologies and the uses of those technologies in the future.

REFERENCES

1. Buzachis, A., Celesti, A., Galletta, A., Fazio, M., Fortino, G., & Villari, M. (2020). A multi-agent autonomous intersection management (MA-AIM) system for smart cities leveraging edge-of-things and Blockchain. *Information Sciences*, *522*, 148-163.
2. Zeng, P., Liu, A., Zhu, C., Wang, T., & Zhang, S. (2022). Trust-based multi-agent imitation learning for green edge computing in smart cities. *IEEE Transactions on Green Communications and Networking*, *6*(3), 1635-1648.

3. Afanasyev, I., Kolotov, A., Rezin, R., Danilov, K., Kashevnik, A., & Jotsov, V. (2019). Blockchain solutions for multi-agent robotic systems: Related work and open questions. *arXiv preprint arXiv:1903.11041*.
4. Belkeziz, R., & Jarir, Z. (2021). Using Blockchain in the Internet of Things Coordination. *International Journal of Advanced Computer Science and Applications*, *12*(8).
5. Anthopoulos, L. G. (2015). Understanding the smart city domain: A literature review. *Transforming city governments for successful smart cities*, 9-21.
6. Su, K., Li, J., & Fu, H. (2011, September). Smart city and the applications. In *2011 international conference on electronics, communications and control (ICECC)* (pp. 1028-1031). IEEE.
7. Camero, A., & Alba, E. (2019). Smart City and information technology: A review. *cities*, *93*, 84-94.
8. Rahman, M. A., Rashid, M. M., Hossain, M. S., Hassanain, E., Alhamid, M. F., & Guizani, M. (2019). Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city. *Ieee Access*, *7*, 18611-18621.
9. Biswas, K., & Muthukkumarasamy, V. (2016, December). Securing smart cities using blockchain technology. In *2016 IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS)* (pp. 1392-1393). IEEE.
10. Majeed, U., Khan, L. U., Yaqoob, I., Kazmi, S. A., Salah, K., & Hong, C. S. (2021). Blockchain for IoT-based smart cities: Recent advances, requirements, and future challenges. *Journal of Network and Computer Applications*, *181*, 103007.
11. Singh, S., Sharma, P. K., Yoon, B., Shojafar, M., Cho, G. H., & Ra, I. H. (2020). Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city. *Sustainable cities and society*, *63*, 102364.
12. Li, S. (2018, August). Application of blockchain technology in smart city infrastructure. In *2018 IEEE international conference on smart internet of things (SmartIoT)* (pp. 276-2766). IEEE.
13. Vujičić, D., Jagodić, D., & Ranđić, S. (2018, March). Blockchain technology, bitcoin, and Ethereum: A brief overview. In *2018 17th international symposium infoteh-jahorina (infoteh)* (pp. 1-6). IEEE.
14. Kushwaha, S. S., Joshi, S., Singh, D., Kaur, M., & Lee, H. N. (2022). Systematic review of security vulnerabilities in ethereum blockchain smart contract. *IEEE Access*, *10*, 6605-6621.
15. Yavuz, E., Koç, A. K., Çabuk, U. C., & Dalkılıç, G. (2018, March). Towards secure e-voting using ethereum blockchain. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1-7). IEEE.
16. Ferretti, S., & D'Angelo, G. (2020). On the ethereum blockchain structure: A complex networks theory perspective. *Concurrency and Computation: Practice and Experience*, *32*(12), e5493.

17. Buzachis, A., Celesti, A., Galletta, A., Fazio, M., Fortino, G., & Villari, M. (2020). A multi-agent autonomous intersection management (MA-AIM) system for smart cities leveraging edge-of-things and Blockchain. *Information Sciences*, *522*, 148-163.
18. Wang, Y., & Chen, H. (2022). Blockchain: A potential technology to improve the performance of collaborative emergency management with multi-agent participation. *International Journal of Disaster Risk Reduction*, *72*, 102867.
19. Chen, F., Xiao, Z., Cui, L., Lin, Q., Li, J., & Yu, S. (2020). Blockchain for Internet of Things applications: A review and open issues. *Journal of Network and Computer Applications*, *172*, 102839.
20. Seid, A. M., Lu, J., Abishu, H. N., & Ayall, T. A. (2022). Blockchain-Enabled Task Offloading With Energy Harvesting in Multi-UAV-Assisted IoT Networks: A Multi-Agent DRL Approach. *IEEE Journal on Selected Areas in Communications*, *40*(12), 3517-3532.
21. Samuel, O., Javaid, N., Khalid, A., Imrarn, M., & Nasser, N. (2020, December). A trust management system for multi-agent system in smart grids using blockchain technology. In *GLOBECOM 2020-2020 IEEE Global Communications Conference* (pp. 1-6). IEEE.
22. Qi, R., Feng, C., Liu, Z., & Mrad, N. (2017). Blockchain-powered internet of things, e-governance and e-democracy. *E-democracy for smart cities*, 509-520.
23. Algarni, S., Eassa, F., Almarhabi, K., Almalaise, A., Albassam, E., Alsubhi, K., & Yamin, M. (2021). Blockchain-based secured access control in an IoT system. *Applied Sciences*, *11*(4), 1772.
24. Buzachis, A., Celesti, A., Galletta, A., Fazio, M., & Villari, M. (2018, December). A secure and dependable multi-agent autonomous intersection management (MA-AIM) system leveraging blockchain facilities. In *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)* (pp. 226-231). IEEE.
25. Sabir, B. E., Youssfi, M., Bouattane, O., & Allali, H. (2020). Towards a New Model to Secure IoT-based Smart Home Mobile Agents using Blockchain Technology. *Engineering, Technology & Applied Science Research*, *10*(2).
26. Zulfiqar, M., Kamran, M., & Rasheed, M. B. (2022). A blockchain-enabled trust aware energy trading framework using games theory and multi-agent system in smat grid. *Energy*, *255*, 124450.
27. Alhasnawi, B. N., & Jasim, B. H. (2021). A new internet of things enabled trust distributed demand side management system. *Sustainable Energy Technologies and Assessments*, *46*, 101272.
28. Abou-Nassar, E. M., Iliyasu, A. M., El-Kafrawy, P. M., Song, O. Y., Bashir, A. K., & Abd El-Latif, A. A. (2020). DITrust chain: towards blockchain-based trust models for sustainable healthcare IoT systems. *IEEE access*, *8*, 111223-111238.
29. Singh, S., Sharma, P. K., Yoon, B., Shojafar, M., Cho, G. H., & Ra, I. H. (2020). Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city. *Sustainable cities and society*, *63*, 102364.
30. Oliveira, T. A., Oliver, M., & Ramalhinho, H. (2020). Challenges for connecting citizens and smart cities: ICT, e-governance and blockchain. *Sustainability*, *12*(7), 2926.

31. Gadekallu, T. R., Pham, Q. V., Nguyen, D. C., Maddikunta, P. K. R., Deepa, N., Prabadevi, B., ... & Hwang, W. J. (2021). Blockchain for edge of things: Applications, opportunities, and challenges. *IEEE Internet of Things Journal*, *9*(2), 964-988.
32. Wu, X., & Liang, J. (2021). A blockchain-based trust management method for Internet of Things. *Pervasive and Mobile Computing*, *72*, 101330.
33. Bhuyan, M., Kashihara, S., Fall, D., Taenaka, Y., & Kadobayashi, Y. (2022). A survey on blockchain, SDN and NFV for the smart-home security. *Internet of Things*, 100588.
34. Samal, T., & Dutta, R. (2021, December). Blockchain-inspired Distributed Trust in a Smart Vehicular Network System. In *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* (pp. 1-6). IEEE.
35. Seid, A. M., Erbad, A., Abishu, H. N., Albaseer, A., Abdallah, M., & Guizani, M. (2023). Blockchain-Empowered Resource Allocation in Multi-UAV-Enabled 5G-RAN: A Multi-agent Deep Reinforcement Learning Approach. *IEEE Transactions on Cognitive Communications and Networking*.
36. Al Ridhawi, I., Aloqaily, M., & Karray, F. (2022). Intelligent blockchain-enabled communication and services: Solutions for moving internet of things devices. *IEEE Robotics & Automation Magazine*, *29*(2), 10-20.
37. Kamran, M., & Rasheed, M. B. A Blockchain-Enabled Trust Aware Energy Trading Framework Using Games Theory and Multi-Agent System in Smat Grid. *Available at SSRN 4024957*.
38. Kanak, A., Ugur, N., & Ergun, S. (2020, September). Diamond accountability model for blockchain-enabled cyber-physical systems. In *2020 IEEE International Conference on Human-Machine Systems (ICHMS)* (pp. 1-5). IEEE.
39. Ali, M., Karimipour, H., & Tariq, M. (2021). Integration of blockchain and federated learning for Internet of Things: Recent advances and future challenges. *Computers & Security*, *108*, 102355.
40. Habibzadeh, H.; Nussbaum, B.H.; Anjomshoa, F.; Kantarci, B.; Soyata, T. A survey on cybersecurity, data privacy, and policy issues in cyber-physical system deployments in smart cities. Sustain. Cities Soc. 2019, 50, 101660.
41. Khan, P.W.; Byun, Y.-C.; Park, N. A Data Verification System for CCTV Surveillance Cameras Using Blockchain Technology in Smart Cities. Electronics 2020, 9, 484.
42. Malik, M.N.; Azam, M.A.; Ehatisham-Ul-Haq, M.; Ejaz, W.; Khalid, A. ADLAuth: Passive Authentication Based on Activity of Daily Living Using Heterogeneous Sensing in Smart Cities. Sensors 2019, 19, 2466.
43. Meshram, C.; Ibrahim, R.W.; Deng, L.; Shende, S.W.; Meshram, S.G.; Barve, S.K. A robust smart card and remote user password based authentication protocol using extended chaotic maps under smart cities environment. Soft Comput. 2021, 25, 10037–10051.
44. Espositoa, C.; Ficco, M.; Gupta, B.B. Blockchain-based authentication and authorization for smart city applications. Inf. Process. Manag. 2021, 58, 102468