future internet

Review

An In-Depth Review on Blockchain Simulators for IoT Environments

Jason Zheng [1], Chidinma Dike [1], Stefan Pancari [1], Yi Wang [1,*], George C. Giakos [1], Wafa Elmannai [1] and Bingyang Wei [2]

[1] Electrical and Computer Engineering Department, Manhattan College, Riverdale, NY 10471, USA; jzheng05@manhattan.edu (J.Z.); cdike01@manhattan.edu (C.D.); spancari01@manhattan.edu (S.P.); george.giakos@manhattan.edu (G.C.G.); wafa.elmannai@manhattan.edu (W.E.)
[2] Computer Science Department, Texas Christian University, Fort Worth, TX 76109, USA; b.wei@tcu.edu
* Correspondence: yi.wang@manhattan.edu

Abstract: Simulating blockchain technology within the IoT has never been as important. Along with this comes the need to find suitable blockchain simulators capable of simulating blockchain networks within an IoT environment. Despite there being a wide variety of blockchain simulators, not all are capable of simulating within an IoT environment and not all are suitable for every IoT environment. This article will review previously published works and present a list of suitable blockchain simulators as well as a few untested simulators that have the potential to simulate blockchain networks within an IoT environment. A total of 18 blockchain simulators are presented and discussed in this paper. In addition, a comprehensive list of the advantages and limitations of each simulator is presented to demonstrate the best situation in which simulators should be used. Finally, recommendations are made on when each simulator should be used and in what situation it should be avoided.

Keywords: IoT; blockchain; simulator

check for updates

Citation: Zheng, J.; Dike, C.; Pancari, S.; Wang, Y.; Giakos, G.C.; Elmannai, W.; Wei, B. An In-Depth Review on Blockchain Simulators for IoT Environments. *Future Internet* **2022**, *14*, 182. https://doi.org/10.3390/fi14060182

Academic Editors: Vojislav B. Mišić and Jelena Mišić

Received: 8 May 2022
Accepted: 7 June 2022
Published: 10 June 2022

1. Introduction

The Internet of Things (IoT) is a recent technology that uses smart connected systems to create a global network of physical devices that exchange and communicate data with each other. While IoT technologies have already been widely successful and popular in different sectors, they lack a secure and unified method of communication and data transfer. IoT devices typically have hardware constraints which make it much more difficult to implement strong security measures. As a result, trying to implement strict security protocols with minimal processing power and low-energy usage can be very difficult.

Blockchain technology has recently gained a lot of recognition and attention due to its implementation in cryptocurrencies such as Bitcoin and Ethereum. Blockchain is essentially a system for recording data that makes it much more difficult to change or hack. Blockchain uses a distributed networking system of machines that replicate and create a chain of data. This chain of data can be considered a ledger, with each of these becoming a block. This chain of data is turned into a block which is linked to the previous block creating a chain of blocks, hence the name blockchain. Every block is further reinforced with blocks that come after it. This is due to the fact that the blocks that come after the previous one will contain information on the last block. Additionally, each block is propagated within the network, allowing each machine to view the entire chain and all of its data allowing for multiple verifications to happen. This ensures that the data is authentic, and that the block's integrity is not compromised.

The distributed nature of blockchain allows it to be a decentralized transaction system that is transparent and secure. By being independent of a centralized network, blockchain technology has become an increasingly promising foundation for future development of the internet and data processing. As the world becomes more interconnected and "smart", it will ultimately become more reliant on data storage and data processing. This brings up the need for a faster and more efficient means of securing data and providing a trustworthy system. These smart and connected systems can leverage blockchain, which has the ability to provide data security using its distributed and immutable structure.

A proposed solution to this is to implement blockchain within IoT. Blockchain, being decentralized and distributed in nature, allows it to be highly resistant to tampering. It also takes less processing power from each individual node or device, making it suitable for a minimal processing device typically seen in IoT.

Blockchains have been successfully used in IoT applications, as shown in Figure 1, including smart homes [1], smart cities [2], smart agriculture [3], smart power grids [4], smart transportation and automotives [5], smart healthcare [6], and smart manufacturing. It also aids in developing future applications such as a cloud constellation of nanosatellites forming a sort of data center in orbit where companies can upload their data and bypass the terrestrial network. However, despite this positivity regarding blockchain technology, only recently have we seen enough development of IoT devices to implement it. For instance, blockchain in satellites would create transparency, trust, and efficiency, in the satellite value chain for logistics purposes. Indeed, blockchain over satellite eliminates the dependence on terrestrial infrastructure for the movement, storage, or computation of data, and removes a significant vulnerability for data breach or the compromise of data. Blockchain can provide data privacy, prevent fraud, improve transparency, and ease record keeping.

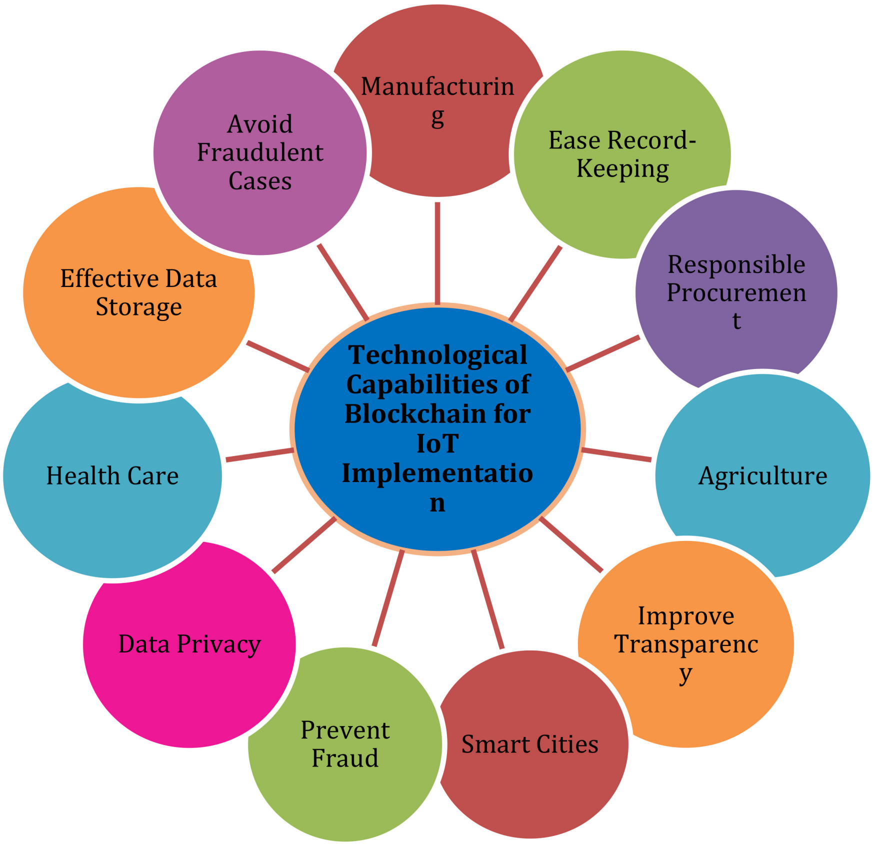

Figure 1. Capability of blockchains for IoT implementation in different domains.

In order to test the effectiveness and security of the blockchain being implemented in IoT, the most cost-effective way is to use simulations. Blockchain simulators have been developed which are capable of demonstrating the different layers of blockchain, as well as applications in IoT. One example for the use of an efficient simulation would be real-time data assembly and processing of shipments. This could be a major advantage for multiple industries that traditionally send their data post-shipment completion to a third party for data aggregation. Using a blockchain-based application connected to the IoT collection devices will allow companies to react to errors and initiate improvements in real-time. A simulator is of utmost importance in order to test such a system beforehand and to make specific adjustments prior to implementation. The blockchain application is perfect for compliance tracking and assurance while being an additional layer on top of already existing data systems. With the use of "smart contracts", the blockchain system can establish and enforce rules between two parties.

To the best of our knowledge, this is the first work to give an in-depth review of different blockchain simulators in IoT environments in order to determine their strengths and weaknesses. This will allow us to discover the most appropriate and best-suited simulator for developing and simulating the IoT environments. We will analyze key features of different simulators that have been used to simulate blockchain in IoT environments, as well as simulators that have the potential to be used in IoT environments in the future. This study will be relevant for researchers and practitioners who are interested in implementing blockchain within IoT or analyzing similar applications. The main contributions of this paper are as follows:

1. By providing a systematic review and comparison of select blockchain simulators that have been used in previous research.
2. By analyzing simulators that can simulate different layers of the blockchain and has the potential to simulate it within an IoT environment.
3. By identifying the strengths and weaknesses of discussed simulators and providing suggestions for future research.

The remaining parts of the paper will be organized into seven sections: Section 2 will present an overview of blockchain technology and its five distinct layers as well as an overview of IoT technology. Section 3 will illustrate the proposed methodology, while Section 4 will provide a systematic review and comparison of experimentally tested simulators as well as a discussion of the field of IoT it was tested on. Section 5 will discuss potential blockchain simulator candidates that are compatible with IoT applications, while Section 6 will discuss implications for future research and Section 7 will conclude our paper.

2. Background

In this section, we will be discussing blockchain technology along with its components, as well as its distinct layers. Along with that, we will also discuss IoT and how blockchain technology can be implemented in IoT technology such as smart homes, smart farming, and many others.

2.1. Blockchain Components

Blockchain technology, as previously stated, is characterized by its blocks which are formed into chains, hence the name blockchain. However, blockchain technology is much more complicated than just a collection of blocks and chains. It requires many other components to actually build the block and make sure that they will not be tampered with and will remain safe. Some of these important technologies include cryptographic hash functions, asymmetric-key cryptography, and ledgers [7].

The first main component of blockchain technology is the cryptographic hash functions. This is applied to data in a method called hashing. Hashing is a method used to calculate a unique output for an input of any size. The data is encrypted into a secure format which is unreadable unless the recipient has the keys. This allows individuals to take input data and hash the data to derive the same results. This proves that there has been no change to the data [7]. One of the most widely implemented hash functions in blockchain technology would be the Secure Hash Algorithm with an output of 256 bits, otherwise, known as SHA-256. The SHA256 algorithm takes inputs that have a length of less than 2^{64} bits and releases an output that has a length of 256 bits. It has a block size of 512 bits which are represented by sixteen 32-bit words. This block of 512 enters a message compression function in 32-bit words through a message scheduler. The message scheduler then expands the 512-bit message block into sixty-four 32-bit words. The SHA256 hashing algorithms are then performed on words that are 32 bits in length, using eight working variables that are also 32 bits in length. The values of the working variable are computed at every round and this is continued until 64 rounds have been completed [8].

SHA256 also takes a 256-bit initialization vector which is fixed for the first message block. The intermediate message digest obtained at the end of the first 64 rounds is used as the initialization vector for the next message block. The SHA256 hash function is built using the Davies–Meyer construction where the initialization vector is added to the output of 64 rounds. After 64 rounds of message compression and the addition of the initialization vector, the algorithm produces an intermediate message digest of 256 bits. After the whole message block has been hashed, a value of 256 bits is obtained that is the final message digest of the input message. The SHA256 hashing algorithm is thus similar to a block cipher with a 256-bit message block size and a 512-bit key that is expanded into sixty-four 32-bit round keys using the message scheduler for each of the 64 rounds of this cipher [8].

A second important component of blockchain is asymmetric-key cryptography also known as public-key cryptography [9]. Asymmetric-key cryptography uses a pair of keys: one public and one private. The main purpose of this component is to be used in transactions such as those done in cryptocurrency. The public key is used to secure operations of the blockchain and give everyone access to the knowledge stored in the block such as the address of a single cryptocurrency in the entire network. The private key is much more restrictive and is used by an individual to digitally sign transactions.

The third important component of blockchain would be the ledger. A ledger would simply be a collection of transactions. These ledgers traditionally were centralized and operated by a single party. However, the distributed ledger is much more common in the case of blockchain [7]. Distributed ledgers are digital ledgers that are distributed across a network to all the nodes, which results in all the nodes having the same copy of the ledger. This ledger will update all nodes or holders on the network simultaneously. Distributed ledgers also authenticate information through cryptographic signature [10]. In the case of blockchain, distributed ledgers are made of blocks and all these blocks form a chain to create the entire ledger. Figure 2 displays all different layers in a blockchain, which will be explained in detail in the next section.

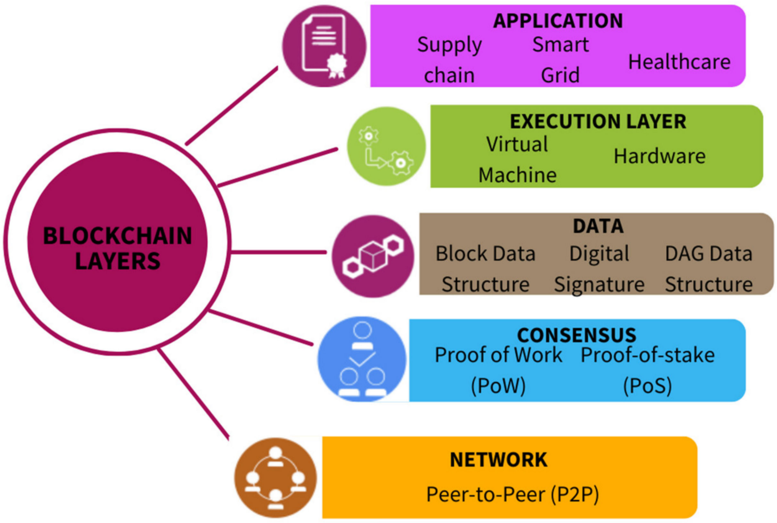

Figure 2. Five Different Layers of Blockchain.

2.2. Blockchain Layers

2.2.1. Network Layer

The first and bottom layer of the multi-layered Distribute Ledger Technology (DLT) stack would be the network layer. This layer consists of a peer-to-peer network in which participants share resources without a central authority, i.e., it is decentralized. While all participants in a P2P network are considered equal, participants are split into two basic types of nodes: light/lightweight nodes and full nodes. Full nodes are the main type of node that stores a complete copy of the ledger and takes care of all the mining, validations, and execution of consensus rules. Lightweight nodes are not able to act as a full ledger but rather are meant to supplement the full nodes of the network. Lightweight nodes store block headers and act as clients to issue transactions. The network layer is critical and takes care of peer discovery, transactions, and block propagation. Depending on the size of the blockchain, the speed of peer discovery, network delays, and propagation may have an impact on the performance of the DLT [11].

2.2.2. Consensus Layer

The consensus layer is very important in the DLT state system as its role is to get all the nodes of the system to reach an agreement. Two main consensus algorithms used by DLT systems are proof based and PBFT [12].

The proof-based consensus was first introduced on the bitcoin network as PoW or proof of work. This is the core mechanism on which the bitcoin network was based, a mechanism which relies on competition between nodes to compete to be the first to solve a mathematical problem [13]. This calculated block would then be broadcast by a node to other nodes, which must then mutually confirm the correctness of the hash value. Once this has been achieved, other miners would add this new block to their own blockchain. Quite similarly, this PoW is used by other cryptocurrencies such as Ethereum and Dogecoin. A drawback would be the waste of considerable computing resources. Proof of Stake (PoS) and Proof of Authority (PoA) are other two proof-based consensus algorithms.

PBFT or Practical Byzantine Fault Tolerance is an algorithm intended to handle up to 1/3 malicious byzantine replicas. The PBFT algorithm is divided into three phases: pre-prepare, prepare and commit. The pre-prepared and preparation phases are used to order requests sent in the same view even when the primary is faulty. The prepare and commit phase is used to ensure that requests that commit are totally ordered across views. For each phase, a node needs 2/3 votes from all nodes to proceed from one phase to the next. The PBFT algorithm relies on a fault tolerance calculated by the formula $(n-1)/3$ to ensure activity and safety. An important feature of the PBFT consensus algorithm is that nodes are only partially trusted [14]. Hyperledger Fabric is an example that uses this type of consensus.

2.2.3. Data Layer

The data layer of the blockchain is typically used to describe the physical layer of the blockchain or DLTS. Included in this layer is practically the entire underlying technology of the blockchain. This includes data block and chain structure, hash function, Merkle tree, asymmetric public key data encryption, and time stamp technology [15]. Despite all of this technology included within the data layer, the most important aspect of this layer is storing data. The data layer is the blockchains database and safely stores all information in the form of data blocks. These data blocks, which are formed into chains, can be accessed by any full node.

In regard to data security, the blockchain system uses the previously mentioned Merkle tree structure to record transactions. Hashes of transactions are computed using the Merkle tree data structure and are stored as Merkle root. The Merkle root, previous hash, timestamps, and the decentralized nature of blockchain make it incredibly difficult to tamper with the system. Along with the security nature of the Merkle tree structure, it also allows transactions to be carried out safely between nodes in the case of decentralization. A drawback is that it can be very energy demanding and have slow processing [11].

2.2.4. Execution Layer

The execution layer has runtime environments such as virtual machines (VMs), containers, and compilers that are installed on nodes. This layer also implements smart contracts, through which it implements trust. These smart contracts run on the local VMs in each individual node on the network. The network then collects self-executing computer instructions to ensure mutual consent between non-trusting parties [16]. A drawback of smart contracts would be the waste of computing resources due to the aborted transactions.

2.2.5. Application Layer

The application layer is the top layer of the blockchain network and is used to connect decentralized applications with the underlying blockchain technology [15]. The most popular use of blockchain technology is cryptocurrency. Typically, along with the cryptocurrency comes a lot of applications such as crypto wallets, smart contracts, and various other decentralized applications [12]. Smart contracts are widely used in cryptocurrency; however, they are designed to facilitate, verify, and enforce the execution of the contract.

Outside of cryptocurrency, the applicability of blockchain can be applied to IoT. Some examples are smart cars, smart healthcare, smart farming, and even smart cities. It is in the application layer that blockchain technology can be applied to IoT.

2.3. IoT Technology

IoT, which is known as the Internet of Things, simply refers to the physical objects and devices that are capable of connecting and exchanging data with other devices through the internet or other communication networks. IoT is one of the most important areas of future technology and is beginning to be implemented in multiple industries [17]. IoT devices are now not only able to connect to a network and communicate with one another but are also capable of collecting data from the environment and sharing that data to other devices for analytics, applications, and communication [18]. This is important for creating a smart environment as the connection of multiple devices and sharing information is a must.

A few examples of IoT being explored to be used in unexpected parts of our lives are transportation systems [19] and in supply chain operations [20]. When it comes to transportation systems, it has been suggested that we use sensors on vehicles, roads, and infrastructure to collect and store huge amounts of data, collectively known as big data. Using this big data traffic control, road conditions, and scheduled travel time can all be viewed and managed with the data collected by the IoT sensors [19]. Similarly, in supply chains, there have been some proposals to use machine learning and artificial intelligence algorithms in conjunction with smart sensors to monitor and collect data on remote equipment and provide suggestions on when maintenance is needed [20]. While both of these examples are not in use yet, we see a common trend of using IoT devices to collect large amounts of data and process them into more useful information. Implementing blockchain technology into the IoT environment has the potential to make collecting, storing, and sharing data more efficient than other proposed technologies. In addition, it may lower cost when it comes time to implement IoT technologies.

2.4. Blockchain Implementation in IoT

When it comes to implementing blockchain within an IoT environment, it can play an important role in more than just security. When used along with a smart contract, it can be used in managing, controlling, and securing IoT devices [21]. An example of this is for wines and spirit, where blockchain technology is being recommended for use in labeling and tracing these products to ensure quality and to prevent illegal trading and adulteration [22]. Blockchain technology in this case is being used to manage and control the smart sensors used to keep track of these liquors and wines. In addition to its capability to make IoT device management more efficient, it allows for IoT devices to be removed from the control of a centralized authority who can manipulate or stop the system from working [23]. This makes attacks against the network a lot more difficult since the network does not revolve around an individual. In addition, the data received from IoT devices and stored in the blockchain network would also be less susceptible to plaintext and cipher attacks due to the hashing of data in the blockchain.

In Figure 3, we present the proposed architecture for an IoT blockchain platform. It is composed of a large number of IoT devices and sensors, user devices, full nodes acting as local bridges, and data storage, all of which are linked to a peer-to-peer blockchain network. The IoT devices and sensors can be connected directly to the blockchain network or can connect to it through a full node. These IoT devices will collect useful data via sensors or user inputs and can request specific transactions through the blockchain network. Data can be sent or received by the IoT devices along with transactions. These data are then stored within the data storage which itself can be stored in two places. One place can be direct data storage, whether it be hardware or software. The second is the blockchain, where data are stored as blocks and can be viewed by anyone. Transactions, on the other hand, must be validated by a group of miners who in turn will receive some sort of reward for validating these transactions. These transactions will then be stored in existing blockchains and form new blocks that will be added to the ledger.

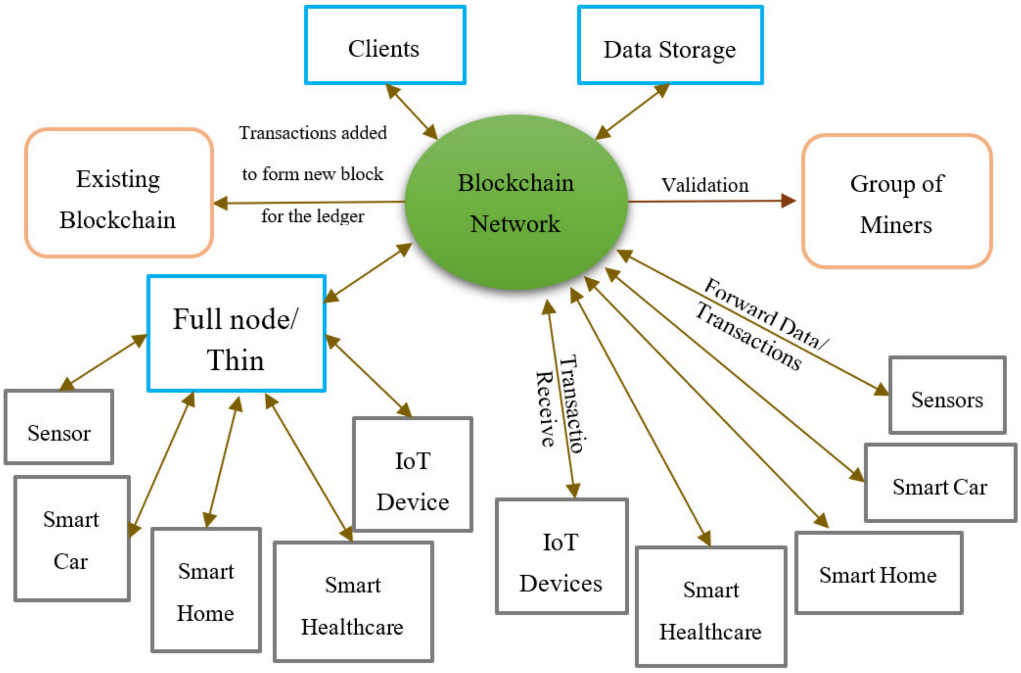

Figure 3. Proposed architecture for an IoT blockchain platform.

3. Methodology

Our systematic review is broken down into four phases: planning, quantitative research, qualitative research, and documenting. To make this study more accurate and increase its overall potential, we followed the four phases to gather information and form it into a cohesive and accurate article. In the following sections, we will go through the four phases of the systematic review. The data were all found by performing a literature search in scientific databases. Many of the articles found came from the scientific publisher IEEE Access, along with MDPI, ELSEVIER, Research Gate, and other similar sources. To find publications necessary to conduct our research, we searched for articles analyzing different blockchain simulators, comparing and contrasting multiple simulators, articles using blockchain simulators to conduct their study, as well as articles explaining blockchain networks in IoT applications. These articles supplied the necessary data to gather both quantitative and qualitative research. However, after reviewing the articles found, it was necessary to narrow down our search due to the sheer abundance of papers. The papers were filtered by the following criteria: they must specifically mention the simulator being discussed or used, they must be related to IoT, and must perform blockchain. As a result, a collection of fifty-five scholarly articles were found that can be used to collect sufficient data for our study.

3.1. Planning

In this phase we analyzed the problem we wanted to explore while setting research goals that we would like to achieve in this review. This paper's purpose is to explore the most prominent simulators that can be used for an IoT network, in conjunction with blockchain technology. These simulators will also be discussed in a comprehensive manner to get a better understanding of their characteristics, such as their advantages and limitations.

3.2. Qualitative Research

The research we did was split into two sections. The first was quantitative research in which we explored more than fifty articles obtained using the research method explained above. The main purpose of this phase was to identify potential simulators out of the hundreds of simulators found before narrowing down the search. For the tested simulators, we ended up with fifteen simulators, out of which four were selected to be discussed in depth due to their popularity and advantages. The advantages and limitations of all fifteen simulators were determined so that a decision on which four to choose could be made. This way, the reader will have a good understanding of the most prominent fifteen simulators that can be used, along with the four simulators that stand out the most. We also selected three potential simulators that were able to solve problems in IoT devices such as network security and scalability.

3.3. Quantitative Research

We used information from the articles collected to find quantitative data to further support our study. For the first graph, it was necessary to review the publication years of the articles used. This highlights the trend of blockchain IoT articles published throughout the years and also validates the need for a comprehensive review of the most popular IoT simulators. Next, a second graph was created to better depict which IoT blockchain layers are simulated by each simulator. This narrowed down our search to the most important simulators out of the fifteen selected. This was done by assuming that the most useful simulators will simulate at least two of the most popular layers.

3.4. Documentation

The final phase of this review was documentation. This phase comprises the results of the literature review. We compiled all the data that we reviewed and organized them properly into their respective sections. We also analyzed the simulators that we had done research on and compared them to one another.

4. A Systematic Review of Blockchain Simulators in IoT

This section will first explain the growing trend of blockchain IoT research and then systematically review and compare blockchain simulators, considering the capability to simulate desired environments and their technical features in order to provide a reference for selecting the most suitable blockchain simulator. It will be split into two main sections: the first being simulators that have been previously used to simulate various aspects of blockchain in IoT, and the second being simulators that have not been used in IoT but have the potential to do so.

4.1. Growing Trend of Blockchain IoT Research

In Figure 4, the growing trend of blockchain and IoT publications is displayed. This graph was constructed by looking at the fifty-five scholarly articles we picked and collecting their years of publication. The data shows that most of the articles collected are from years greater than 2018, validating that the experimental results obtained in this study are relevant and will be a useful reference for future research. It should be noted that the number of papers published in 2022 are slightly smaller than the growing trend. This is likely due to the fact that studies currently being worked on are not yet completed. It is assumed that after the end of this year, the number of IoT papers will be around the same, if not more than the expected eight to ten.

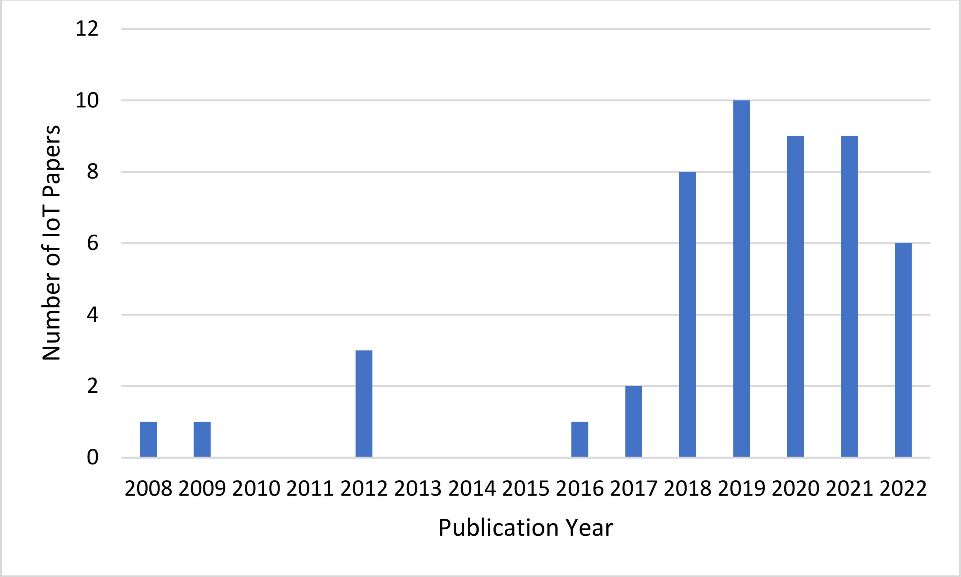

Figure 4. Number of IoT papers used versus their publication dates.

4.2. Overview of Used Blockchain Simulators

In Table 1, we provide the names of the simulator, the article from which the simulator was used, and the layer of the blockchain that it is capable of simulating. In Table 2, we provide further information about the simulators: the language that the simulator runs on, the last date of commit, and links to the source code.

Table 1. A representation of simulator article and simulator type.

Simulator	Author and Sources	Title	Simulator Type
Blocksim	Alharby and Moorsel [24]	BlockSim: An Extensible Simulation Tool for Blockchain Systems	Network, Consensus, Incentives
Simblock	Banno and Shudo [25]	Simulating a Blockchain Network with SimBlock	Consensus, network
Omnet++	Gupta et al. [26]	The Applicability of Blockchain in the Internet of Things	Network
NS3	Dedeoglu et al. [27]	A Trust Architecture for Blockchain in IoT	Network, Consensus
NS2	Yazdinejad et al. [28]	Decentralized Authentication of Distributed Patients in Hospital Networks using Blockchain	Network
Matlab	Moon et al. [29]	Home IoT device management blockchain platform using smart contracts and a countermeasure against 51% attacks	Network
Ethereum	Augusto et al. [30]	An Application of Ethereum smart contracts and IoT to logistics	All 5 layers
iFogSim	Gupta et al. [31]	iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments	Network
Hyperledger Sawtooth	Vangala et al. [32]	Smart Contract-Based Blockchain-Envisioned Authentication Scheme for Smart Farming	Consensus

Table 1. *Cont.*

Simulator	Author and Sources	Title	Simulator Type
Hyperledger Fabric	Assaqty et al. [33]	Private-Blockchain-Based Industrial IoT for Material and Product Tracking in Smart Manufacturing	Consensus
Hyperledger Iroha	Ray et al. [34]	Blockchain for IoT-Based Healthcare: Background, Consensus, Platforms, and Use Cases	N/A
Ganache	Kushal et al. [35]	Machine Learning for Secure Device Personalization Using Blockchain	All 5 layers
CORE	Oham et al. [36]	B-FERL: Blockchain based Framework for Securing Smart Vehicles	Network
Cooja	Pavithran et al. [37]	Towards building a blockchain framework for IoT	Network
Sniper	Kreku et al. [38]	Evaluating the Efficiency of Blockchains in IoT with Simulations	Network

Table 2. Simulator source code, language, and last date of Commit.

Simulator	Codes	Language	Date of Last Commit
Blocksim	https://github.com/carlosfaria94/blocksim	python	N/A
Simblock	https://github.com/maher243/BlockSim	Java	20 January 2020
Omnet++		C++	13 April 2022
NS3	https://gitlab.com/nsnam/ns-3-dev/	C++ & Python	1 October 2021
NS2	https://github.com/sajidhasanapon/NS2	C++/OTCL/TCL	4 November 2011
Matlab	https://github.com/robotarium/robotarium-matlab-simulator	C++	9 March 2022
Ethereum	https://github.com/ethereum	Solidity	1 December 2020
iFogSim	https://github.com/Cloudslab/iFogSim	Java	18 October 2021
Hyperledger Sawtooth	https://github.com/hyperledger/sawtooth-poet	Any	28 January 2022
Hyperledger Fabric	https://github.com/hyperledger/fabric-private-chaincode	Java	28 January 2022
Hyperledger Iroha	https://github.com/hyperledger/iroha	C++	17 February 2022
Ganache	https://github.com/trufflesuite/ganache	Java script	
CORE	https://github.com/coreemu/core	Python	22 March 2022
Cooja	https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja	C & C++	N/A
Sniper	https://github.com/duttresearchgroup/sniper-mem/blob/master/README	N/A	15 April 2021

When it comes to simulator types, most simulators are capable of only simulating one layer with a mix of network and consensus. Exceptions to the one-layer simulators would be Simblock, NS3, and Blocksim. NS3 and BlockSim are both capable of simulating network and consensus while Blocksim has the extra incentives layer [24]. This might be relevant for those who are looking to incentivize their IoT devices, so more people are willing to use them.

Figure 5 shows a bar graph of which layers are more likely to be simulated with respect to the fifteen simulators shown in Table 1. Most of the simulators are simulating the network layer, with the second most simulating the consensus layer. Only a few of the simulators are operating under the data, execution, and application layer. However, when comparing this data to the qualitative data found in Table 1, it is shown that the simulators simulating layers other than the network and consensus are not exclusively running these layers, but instead running all five layers. Since the network layer is the most popular layer to run, this should be a determining factor in deciding which simulators to look at in depth. In addition to simulating the network layer, being able to simulate the consensus layer is also important, but not completely necessary since the majority only simulate one. Despite this, simulators that have more than one layer cover a wider range of uses.

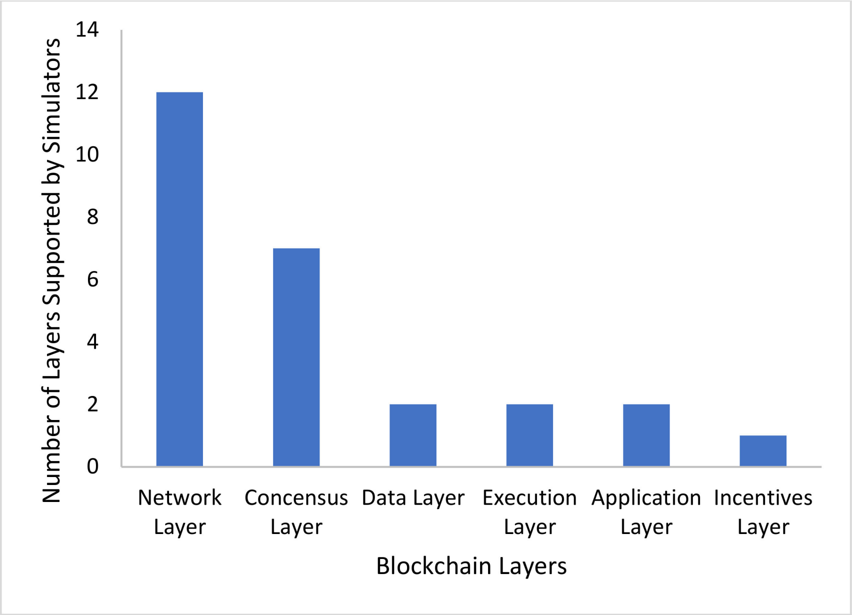

Figure 5. The number of blockchain layers supported by the 15 simulators versus the blockchain layers.

In terms of source code and last commit, most of the simulators were accessible and had their source codes on GitHub. Most simulators have had some type of version update within the past year, but a few of the simulators have not been updated for 3+ years. This included NS2 and Simblock. However, it is important to note that NS2 does have a new version, NS3, which might be the reason for NS2 not being updated.

Finally, in terms of language, the most common were C++, python, and Java, while NS2 was capable of using both C++ and OTCL/TCL.

4.3. Comparative Analysis

In this subsection, we will perform an in-depth analysis of five selected simulators. In terms of selecting the blockchain simulators for our analysis, the main criteria were: (a) widespread use of the simulator in an IoT environment and (b) the ability to simulate multiple layers or a distinct layer that few other simulators simulate. The five simulators that we examine in-depth are Hyperledger Fabric, Blocksim, and NS3. We also discuss Ethereum along with the Ganache simulator since Ganache is specifically used to simulate an Ethereum-based environment and the two are used together quite often.

4.3.1. Hyperledger Fabric

Hyperledger Fabric is a simulator that simulates only the network layer and its native language is Java. The reason this simulator was selected is its high popularity and relative ease when it came to finding articles. The main components of the Hyperledger Fabric architecture are peer nodes, ordering nodes, and client applications with identities of components being generated from certificate authorities. It is well suited to simulate blockchain networks that are supported by smart contracts [39]. However, some downsides include not being able to simulate traditional blockchain consensuses such as proof based or PBFT.

In terms of simulating blockchain within IoT environments, it has been tested with IoT for material and product tracking in smart manufacturing, IoT data management, IoT edge device security, and IoT fish farm platform. Despite not having any mainstream consensus, Hyperledger Fabric is a relatively popular simulator. In terms of material and product tracking, the simulator was used to build a blockchain network in which different numbers of devices were simulated and one device was simulated to send ten concurrent transactions [33]. Within the IoT data management simulation, the simulator was used as the blockchain network with five nodes deployed on Vultr VPS connecting to this blockchain network directly [40]. Transactions were tested between the network-to-peer and peer-to-peer, with nodes receiving data from Hyperledger Fabric clients. In the IoT edge security devices, the simulator was implemented as permissioned blockchain to secure the edge computing device by employing a local authentication process. It has been stated that Hyperledger was used due to its lower processing complexity [41]. Lastly, in the final simulation of implementing a secure fish farm platform, the infrastructure of the blockchain network was built on the simulator. The network consisted of four peer nodes and one order node along with a smart contract that was deployed to all peers [42]. The research team used Hyperledger Fabric to collect and graph data to get a better understanding on how their network should run. For example, Hang et al. graphed the effect of the send rate on the blockchain network transaction throughput [42] to calculate the optimal transaction throughput for their network. By referring to Figure 6, they determined the optimal rate for transaction throughput to be 1100 tps since the average throughput drops after that. This is critical information to know about their network to ensure that it is running optimally.

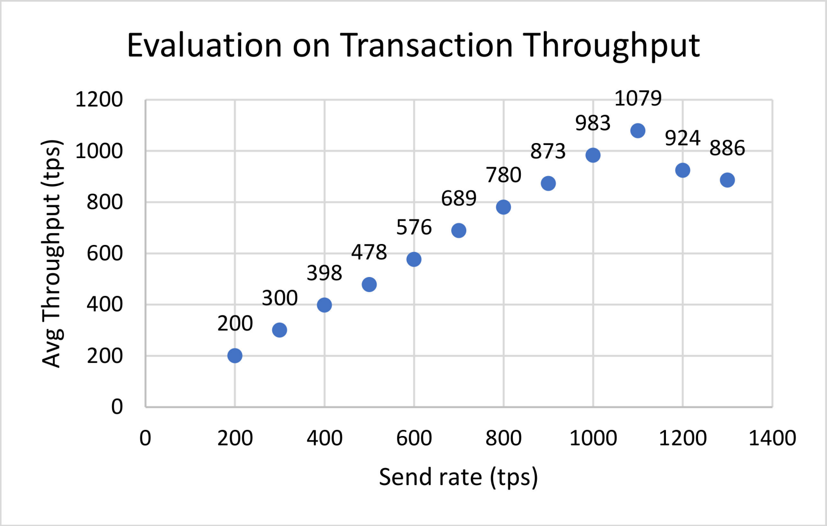

Figure 6. Effect of the send rate on blockchain network transaction throughput using Hyperledger Fabric. Reprinted/adapted with permission from Ref. [42]. 2020, Elsevier B.V.

4.3.2. Blocksim

Blocksim is a simulator that simulates network, consensus, and incentive layers which is built using Python language [27]. In addition to its obvious benefit of being able to simulate three distinct layers, Blocksim is user-friendly and can be used for simulating large-scale networks. Blocksim achieves the three objectives of generality, extensibility, and simplicity [43]. Simply, Blocksim is capable of simulating different blockchain models and designs. It is easily manipulated by users and the data provided is easily understood and unwanted data can be hidden. However, some limitations of this simulator include the amount of data that can be stored, and the number of concurrent users allowed to access the simulations.

In terms of IoT environments, the simulator has been used in smart grids and IoT-based health care. In both articles, Blocksim provides the simulation framework to analyze Ethereum-based architecture. Blocksim has been used to simulate gas consumptions as well as certificate generation and verification transactions within [44]. For example, Blocksim toolkit has been used to generate Table 3 for the proposed system by Namasudra et al. The

Blocksim toolkit was used to generate certificates and verification transactions at random, to help the team have a better understanding of their proposed architecture [44]. Using Table 3, they determined that the certificate generation required more gas consumption than the verification transaction for their proposed architecture. The determined cause of this is the large number of computations done in the certificate generation transaction, when compared to the verification transaction [44]. This is useful information to know about their architecture, and was all simulated using Blocksim.

Table 3. Smart contract cost analysis generated using Blocksim. Reprinted/adapted with permission from Ref. [44]. 2021 IEEE.

No. of Certificates	Gas Limit (Units)	Gas Cost	Gas Price (CGWEI)
1	22,645	147,633	100
10	218,730	1,389,224	1000
20	419,660	2,752,440	2000
30	619,290	4,128,660	3000
40	825,720	5,504,880	4000
50	1,032,150	6,881,100	5000
100	2,064,300	13,762,200	10,000

Blocksim's main task in this simulation was to provide the environment for simulating the blockchain network, deploying smart contracts, and executing tests or experiments for evaluating performance parameters, such as latency, processing time, throughput, and computation time [44]. In the smart grid simulation, Blocksim was used to measure block interval, transaction time, and uncle rate. This was done through an Ethereum network simulation that ran on Blocksim [45].

4.3.3. NS3

NS3 is a simulator that has the ability to simulate the network and consensus layer and can be used in both C++ and python. This simulator is quite interesting as it has a scalability feature in which packets can have virtually zero bytes or "dummy bytes". Also, the nodes have optional features, which allow for minimal memory wasted in IPv4 stacks that nodes do not need. These features are incredibly useful as they save memory which can directly lead to shorter simulation times. However, the NS3 simulator does lack a GUI and has weak visualization that is still experimental [46].

In terms of simulating blockchain within IoT, it is being tested in IoT security and anonymity as well as building trust within IoT networks. NS3 was used in simulation to evaluate latency, processing time, and resilience against cyberattacks. In this simulation, it was used to simulate up to 50 nodes as well as transactions between these nodes. It has been noted that NS3 is widely used for peer-to-peer networks [47]. Similarly, in the second simulation, NS3 was used to simulate 48 nodes, in which three gateway nodes generated blocks every 4.5 s, while others did not generate any blocks [27]. Finally, in the third simulation, both the network and consensus layer were simulated. For their initial simulation, the raft consensus was chosen, while in their network simulation, a PoW consensus was simulated [48].

4.3.4. Ethereum + Ganache

Ethereum is a decentralized, open-source blockchain with smart contract functionality. Ganache is a test blockchain simulator meant to simulate the Ethereum network and aid in the development of an application meant to be implemented on the Ethereum network. This is done by creating a user-friendly test network that can circumvent the need to set up an Ethereum client such as Geth or openEthereum [49].

The Ethereum network is a well-established blockchain network that allows people to actively develop new and innovative applications and products that directly tie to Ethereum and use its native cryptocurrency, ether. This allows people to use a blockchain network that already has nodes and circumvent the need to set up self-hosted nodes for testing. However, some shortcomings include long transaction times and high gas prices.

Ganache, being a development tool used by the Ethereum network, provides an easy testing ground for people trying to develop applications meant to be used on the Ethereum network. This limits Ganache to simulate blockchains that are based on Ethereum, meaning it might be difficult to simulate blockchains with different parameters [49].

In terms of IoT environments that have used the Ethereum network or Ganache simulators to simulate blockchain, we see them being implemented in IoT devices, IoT communication, and IoT logistics. For the simulation of blockchain in IoT communications, Ethereum was the sole simulator. While in the simulations of blockchain within IoT devices and logistics, Ganache was used alongside Ethereum.

In the secure IoT communication simulation, Ethereum was used purely to simulate blockchain security. This meant that Ethereum was used in hashing and encoding [50]. In smart devices, Ethereum was used as the base network for storing data via smart contracts and generating hashes [35]. All of this was simulated on Ganache since it is much more user-friendly than directly applying the simulation to the Ethereum network. Finally, in IoT logistics, Ganache was used to simulate a local blockchain while truffle was used to generate smart contracts. In this environment, the network simulated by Ganache was used to test data integrity stored on the blockchain, data tracking, and clearance check [30], with clearance checking referring to the process that ensures items being tracked fit the quality requirements.

4.3.5. Analysis

Hyperledger Fabric, BlockSim, NS3, and Ethereum/Ganache all had their pros and cons, but all seemed to be able to simulate a blockchain within an IoT environment relatively successfully. In terms of simplicity and overall implementation of blockchain within an IoT environment, Ethereum/Ganache would be a great choice since they encompass a complete blockchain prebuilt and already in use. It provides a holistic simulation allowing you to simulate essentially all five layers of the blockchain. NS3 would be a great simulator for large simulations with hundreds of nodes and transactions since it has the ability to create "dummy bytes" and nodes with optional features allowing simulations to run with less memory, which requires less computing power. Hyperledger Fabric would be a simulator that small blockchains can be simulated on and for those who are not looking for traditional consensuses such as proof-based or PBFT consensuses. Finally, Blocksim is a simulator for those looking to build their own blockchain to test on without trying to base it on Ethereum architecture. It is user-friendly and is capable of simulating different blockchain models on multiple layers of the blockchain.

5. Potential Simulators

The relationship between blockchain simulators and IoT is extremely crucial as these simulators have proven their benefits to IoT's security and scalability challenges. Generally, the goal of a blockchain simulator is to construct simulation structures that are simple, hide unnecessary detail, and are easy to alter so that they may be used to solve a variety of blockchain design and deployment problems that are related to performance, reliability, and security [51]. IoT deals with vulnerable security issues that cause them to be victims of distributed denial of service (DDoS) attacks. In this attack, A cybercriminal floods a network with high volumes of data requests so that the network stops operating or communicating as it usually would. IoT's scalability issue is due to the increasing number of devices connected to an internet of things network, the existing systems that handle these tasks will eventually become overloaded, causing the entire network to go down [52]. This would require a huge number of servers to handle the exchange of information. A real-life example of a blockchain simulator securing internet of things devices is the implementation of the Ethereum blockchain simulator smart contract. Smart contracts are a series of programs in a blockchain that run when predetermined conditions are met.

As DDoS causes network congestion, the Ethereum blockchain simulator smart contract makes intruders pay based on the size of the data or packet before sending it off [53]. This prevents the intruders from sending unlimited amounts of data or packets. As technology advances, more blockchain simulators are in development. This part of the paper examines how the following simulators could potentially work in the IoT world: JABS, CBlockSim, and Cardano. Tables 4 and 5 list the three potential simulators discussed as well as provides the source code and relevant articles for each.

Table 4. Potential blockchain simulators for IoT environments.

Simulator	Authors	Title	Simulator Type
JABS	Habib Yajam [54]	Introducing JABS: Just Another Blockchain Simulator	consensus
CBlockSim	Ma et al. [55]	CBlockSim: A Modular High-Performance Blockchain Simulator	consensus
Cardano	Rakesh Sharma [56]	Cardano Aims to Create a Stable Cryptocurrency Ecosystem	consensus

Table 5. Potential blockchain simulators source code, language, and last date of commit.

Simulator	Codes	Language	Date of Last Commit
JABS	https://github.com/hyajam/jabs?ref=hackernoon.com	Java	19 April 2022
CBlockSim	https://github.com/xuyangm/CBlockSim	C++	20 March 2022
Cardano	https://github.com/input-output-hk/cardano-sl	Haskell	25 June 2020

5.1. JABS Simulator

The first blockchain simulator to have the potential to simulate within an IoT environment is JABS. JABS, just another blockchain simulator, is a high-performance and scalable blockchain consensus algorithm simulator. Java is used to create JABS. This project's major purpose is to provide a complete tool for modeling, testing, and comparing consensus methods on a massive, near-real-time scale [54]. The JABs simulator was built to simulate the network layer of the multi-layered DLT stack. It consists of peer-to-peer connection modules, which are when multiple PCs are connected and share resources without going through a third party. The transition from a centralized to a P2P distributed design will eliminate some central sources of failure and bottlenecks. Other advantages of decentralizing the design include increased fault tolerance and system scalability. It would also help to break down IoT silos while also helping to improve IoT scalability [54].

The network latency in the JABS simulator is constructed such that each node has a certain average delay with the other global regions, based on its geographic location. The Pareto distribution is the random distribution used to simulate the latency in this network. In peer-to-peer networks, this random distribution is commonly used for end-to-end communication delay modeling. According to the creator, the fat-tailed distribution is preferred for more accurate modeling of the delays associated with longer periods. JABS often looks to increase the performance of blockchain networks, specifically fixing delays in the blockchain networks. This skill will be useful in the IoT environment because IoT devices suffer from network congestion. Though it is still new, if JABS implements its code into IoT devices, it may help improve them in the scalability aspect in that it can decrease the delays found in IoT devices.

5.2. CBlockSim Simulator

Similarly, CBlockSim is another blockchain simulator that has the potential to work with IoT. CBlockSim is an extension of the simulator BlockSim: Alharby. CBlockSim is a performance and scalability-focused blockchain simulator that simulates a large number of blockchain components [55]. The performance aspect of CBlockSim can benefit the IoT world because it can help enhance any IoT device that is prone to attacks such as a middle-man attack, which is a cyberattack in which an attacker eavesdrops or pretends to be a genuine participant to intercept an existing conversation or data transfer. Moreover, the creators rewrote BlockSim: Alharby in C++ and created a binary transaction pool data structure. This, allowed them to use bitwise operations in C++ to speed up the simulation. The tests show that CBlockSim reduces run time by an order of magnitude and boosts scalability by an order of magnitude when compared with other simulators. CBlockSim can help with IoT's scalability issue. Specifically, it can help clear the traffic in the IoT network as it often looks for a shorter path to get a specific task done.

5.3. Cardano Simulator

Cardano is a new blockchain that investors are paying attention to. The network is being used as a simulator in a similar way to Ethereum and Bitcoin [56]. Cardano intends to address issues such as scalability, interoperability, and sustainability in cryptocurrency. The issues in cryptocurrency are similar to the issues found in IoT devices.

With problems relating to scalability, networks have become sluggish and costs have gone up due to increased transaction volumes. Ouroboros, a Cardano algorithm, is the proposed solution for its scalability issues. To save energy and enable rapid transaction processing, Ouroboros uses a proof of stake (PoS) technique. Cardano's blockchain reduces the number of nodes in a network by choosing leaders who are responsible for checking and approving transactions from a set of nodes rather than having a copy of separate blockchains on each node; the leader node then sends transactions to the primary network. Like IoT devices, a large number of devices would be connected to one IoT network. This causes the other devices on the same network to become slow, resulting in the network shutting down. If the creators of Ouroboros were to modify its algorithm to fit IoT devices, then it may solve that problem.

In interoperability, it is the mobility of a cryptocurrency both inside its native environment and in its interaction with the current global financial ecosystem. As of right now, there is no mechanism to execute cross-chain cryptocurrency transactions or a smooth transaction between cryptocurrencies and the world's financial ecosystem. The only middlemen are exchanges that collapse or demand high fees. Cardano wants to facilitate cross-chain transfers using side chains, which let two parties perform transactions off-chain. Applying this to IoT will help the communication between IoT devices happen much more effortlessly. It also prevents attacks or viruses that may hinder a successful connection between the devices.

Finally, there are problems relating to sustainability. This refers to governance mechanisms that give rewards to miners and other stakeholders, as well as the development of a cryptocurrency's self-sustaining economic model. Cardano's dedication to creating a sustainable cryptocurrency environment can transfer to IoT devices. Cardano can help IoT reduce our carbon footprint by enabling different applications related to energy management and resource efficiency such as water management or waste management.

6. The Implication of Future Research

Since each simulator has its individual strengths, there are certain cases in which one simulator might be preferred over another. In this section, we make recommendations for when to use each simulator that was discussed in detail as well as when the potential simulators might be used. In addition, Tables 6 and 7 will show the advantage and disadvantages of the simulators and potential simulators.

Table 6. Advantages and limitations of blockchain simulators for IoT environments.

Simulator	Advantages	Limitations
Sniper	- parallel multicore simulator [38] - fast simulation speed at around 2 MIDS - at most 25% error	- does not model system-level code [57] - does not model the internals of a superscalar processor that performs out-of-order execution [57]
Simblock	- can simulate thousands of nodes on a single computer [25] - the behavior of each node can be easily changed [25]	- higher CPU usage when compared with other simulators [58]
Omnet++	- can be used to simulate a wireless network with little impact on performance [26]	- more nodes are needed for a stable network [26]
NS3	- scalability features including “dummy bytes” and nodes with optional features - Minimal waste of memory which directly leads to shorter simulation time	- lack of GUI and weak visualization
NS2	- flexible and modular - can output a text-based or animation-based simulation result [59]	- users may need to make their own C++ objects for advanced simulations [59]
Matlab	- has a large library of functions - has a compiler	- need a license to run
Ethereum + Ganache	- well-established blockchain network - circumvents the need to set up self-hosted nodes	- high gas prices and transaction time (not very efficient) - unable to simulate any network except Ethereum-based ones.
iFogSim	- based on fog computing which reduces network traffic [60] - scalable network with little congestion and limited risk of network degradation [60]	- does not support mobility [60] - does not support fault injection [60]
Hyperledger Sawtooth	- allows for parallel transactions [61] - flexible and modular architecture [61] - smart contracts do not need to know the design of the core system to specify business rules for applications [61]	- fully permission-based which could be a downside for some applications [61]
Hyperledger Fabric	- well suited to simulate blockchain networks that are supported by smart contracts	- unable to simulate tradition proof based or PBFT consensus
Hyperledger Iroha	- general-purpose private network and is simple to deploy [61]	- command query separation makes the system more complex [61]
Core	- open source [62] - supports distributed emulation over multiple machines [62] - -has an easy-to-use GUI	- the performance is largely hardware dependent; the number of processes and packets that can be sent is dependent on the processor speed [62]
Cooja	- can simulate wireless sensor networks without any particular mote [63]	- before stating simulation, one of the four given models must be selected. The simulation desired must fall under one of these models [63]
Blocksim	- capable of simulating different blockchain models and designs - achieves generality, extensibility, and simplicity	- limited amount of stored data and the number of concurrent users.

Table 7. Advantages and limitations of potential blockchain simulators for IoT environments.

Simulators	Advantages	Limitations
JABS	consists of peer-to-peer connection modules to eliminate central sources of failure and bottlenecks and increase the fault tolerance and system scalability in IoT	Still new and in its testing phase
CBlockSim	Being written in C++ reduces run time by an order of magnitude and boosts scalability by an order of magnitude when compared to other simulators	Is a proposed simulator. Therefore, it hasn't been tested in real-world cases
Cardano	Dedication to creating a sustainable cryptocurrency environment can help IoT reduce our carbon footprint	Unable to help IoT in interoperability issues because it lacks mechanisms for IoT devices to communicate properly with each other

6.1. Hyperledger Fabric

For those trying to simulate a blockchain environment with certain criteria and have the knowledge to build a network Hyperledger Fabric is a useful simulator. This simulator is best used to simulate the network layer of the blockchain and implement smart contracts. However, if one is trying to simulate consensus, which is common for most blockchain, this simulator should not be used. This simulator should only be used when one wishes to simulate the network layer of blockchain within the IoT; for any other layer, different simulators should be considered.

6.2. Blocksim

Blocksim is a simulator that is capable of simulating three distinct, user-friendly environments and is capable of building different models and designs. Further, data received from the simulation are easily manipulated to display what is desired. Blocksim is best used when one is trying to simulate a unique blockchain network with specific requirements. It is also great for those who want a simulator that will easily allow manipulation of the simulation. In addition, it is capable of simulating proof based and PBFT consensus which are not available in Hyperledger Fabric. Drawbacks include the limited data and the number of concurrent users allowed to access a simulation at the same time. However, if there is a small team running simulations, Blocksim would be a great simulator as it is easy to use and easy to read.

6.3. NS3

NS3's greatest advantage is its ability to send "dummy bytes" and having optional features for nodes, which in turn reduces the amount of memory and computational power required. This simulator is best suited for those who have limited amounts of memory and computational power but need to simulate a large network. This will allow for the simulation of large networks with limited resources. However, the drawbacks include weak GUI and visualizations. If this weakness can be addressed, NS3 could be an excellent simulator for those with limited resources.

6.4. Ethereum + Ganache

Ethereum's greatest advantage is its widespread use and prebuilt network. Ethereum is heavily used already as a cryptocurrency and Ganache is a simulator that is used to simulate Ethereum networks. Ethereum is proof based and is great for those trying to run simulations that have networks similar to Ethereum and for those who do not want to set up and run self-hosted nodes. Ethereum is very versatile and circumvents the need to set up one's own network as it is all built and in use. People running simulations who want to have networks similar to Ethereum should consider it, as it can save considerable time that is needed to build the blockchain network. In addition, people should use Ganache rather than the Ethereum network directly as it has easier interfaces and is more user-friendly. This simulator should not be used if the simulations are not planned for an Ethereum-based network.

6.5. Potential Simulators

In terms of the potential simulators, we explore JABS, CBlockSim, and Cardano. JABS should be used for those trying to explore consensus. CBlockSim, being an extension to BlockSim, is written in C++ and is more efficient than BlockSim which is written in Python. This simulator can be used to simulate blockchain components used to be tested against security threats in IoT devices. Finally, Cardano is similar to Ethereum as it is a blockchain on which a cryptocurrency is run. However, it is more eco-friendly and runs more efficiently than Ethereum. It can be used in simulations that use Ethereum. However, these simulators have not been used or not nearly as much as the four simulators stated previously and need more simulations in order to garner more accurate information on them.

7. Conclusions

This paper summarized some of the most popular blockchain simulators and provided a review of the ideal simulator for a specific situation. It also showed that there is an ideal simulator that will fit the requirements for every simulation, but a select few that it is recommended to pick based on the user's needs. When it comes to implementing blockchain in IoT, multiple simulators have been used and tested. Specifically, Hyperledger Fabric, Blocksim, NS3, and Ethereum/Ganache were some of the simulators that stood out in the countless number of simulators discovered. Each of the named simulators have their advantages and limitations and, depending on the simulation, one might be more ideal than the other. As stated, Hyperledger Fabric is a popular platform with a large number of resources. It is an effective simulator to use if only simulating the network layer is desired; however, if other layers are needed, one should look elsewhere. Blocksim can simulate the consensus, and incentive layer in a simple way. In addition, if one desires to easily simulate a large network, Blocksim is a good choice. Despite these capabilities, Blocksim is limited by the number of users who can be part of the network and the amount of data that can be stored. NS3 is a good simulator to use when security and minimal memory waste is crucial to the network. This simulator is capable of running the network and consensus layer. If a user desires to run a different layer, a variant simulator should be picked. NS3 also lacks a GUI which can make it difficult for some users. Although this is true, it still has the benefits previously mentioned. Lastly, Etherium/Ganache was discussed. Ethereum, in conjunction with Ganache, is a good simulator to choose if one does not want to build a network from scratch, but instead use a prebuilt network. However, transaction times are long, which may not be well suited for most IoT networks. JABS, CBlockSim, and Cardano are just three of the many non-IoT integrated simulators with the potential to work in an IoT environment. These simulators have shown that some of their components can combat the scalability and security issues in IoT devices.

A limitation of this paper is that the parameters, such as the block size distribution and geographical node distribution, of the IoT simulators were not discussed in detail. The support of existing or potential consensus algorithms, such as PoW and PoS, were also not a large factor in our study. It should also be mentioned that there are many other simulators that can be discussed, but it is out of the scope of this paper to provide an in-depth review of all of them. This paper aimed to thoroughly provide the advantages and limitations of some of the most promising blockchain simulators for an IoT network, not all of them. One possibility for the future is to direct research to analyze more simulators to find which ones fit the needs of users not mentioned in this paper. Another possibility for future research is to follow the updates and changes made to each prominent blockchain simulator mentioned, to track their improvements and developments. Some key limitations mentioned in this study may no longer exist in the years, or even months to come. By tracking the progress, or lack thereof, users can stay up to date with the simulators and determine which one to use.

Author Contributions: Conceptualization, Y.W.; methodology, J.Z., C.D. and Y.W.; validation, J.Z., S.P., Y.W., W.E. and B.W.; formal analysis, J.Z., C.D., S.P. and Y.W.; investigation, J.Z., C.D., S.P. and Y.W.; resources, Y.W., W.E., G.C.G. and B.W.; data curation, J.Z., C.D., S.P., Y.W. and W.E.; writing—original draft preparation, J.Z., C.D., S.P. and Y.W.; writing—review and editing, J.Z., S.P., Y.W., W.E., G.C.G. and B.W.; visualization, J.Z., S.P. and Y.W.; supervision, Y.W., W.E. and G.C.G.; project administration, Y.W.; funding acquisition, G.C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alam, M.R.; Reaz, M.B.I.; Ali, M.A.M. A Review of Smart Homes—Past, Present, and Future. *IEEE Trans. Syst. Man Cybern. Part C* **2012**, *42*, 1190–1203. [CrossRef]
2. Batty, M.; Axhausen, K.W.; Giannotti, F.; Pozdnoukhov, A.; Bazzani, A.; Wachowicz, M.; Ouzounis, G.; Portugali, Y. Smart Cities of the Future. *Eur. Phys. J. Spec. Top.* **2012**, *214*, 481–518. [CrossRef]
3. Dagar, R.; Som, S.; Khatri, S.K. Smart Farming—IoT in Agriculture. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; pp. 1052–1056. [CrossRef]
4. Shahinzadeh, H.; Moradi, J.; Gharehpetian, G.B.; Nafisi, H.; Abedi, M. IoT Architecture for Smart Grids. In Proceedings of the 2019 International Conference on Protection and Automation of Power System (IPAPS), Tehran, Iran, 8–9 January 2019; pp. 22–30. [CrossRef]
5. Menon, V.G.; Jacob, S.; Joseph, S.; Sehdev, P.; Khosravi, M.R.; Al-Turjman, F. An iot-enabled intelligent automobile system for smart cities. *Internet Things* **2022**, *18*, 100213. [CrossRef]
6. Baker, S.B.; Xiang, W.; Atkinson, I. Internet of things for smart healthcare: Technologies, challenges, and opportunities. *IEEE Access* **2017**, *5*, 26521–26544. [CrossRef]
7. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. Blockchain Technology Overview. Gaithersburg, MD: National Institute of Standards and Technology. *Comput. Secur. Div. Inf. Technol. Lab.* **2018**, *31*. [CrossRef]
8. Naik, R.P.; Courtois, N.T. Optimising the SHA256 Hashing Algorithm for Faster and More Efficient Bitcoin Mining. *MSc Inf. Secur. Dep. Comput. Sci. UCL* **2013**, 1–65.
9. Puthal, D.; Malik, N.; Mohanty, S.P.; Kougianos, E.; Das, G. Everything You Wanted to Know about the Blockchain: Its Promise, Components, Processes, and Problems. *IEEE Consum. Electron. Mag.* **2018**, *7*, 6–14. [CrossRef]
10. Deshpande, A.; Stewart, K.; Lepetit, L.; Gunashekar, S. Distributed Ledger Technologies/Blockchain: Challenges, Opportunities and the Prospects for Standards. *Overv. Rep. Br. Stand. Inst.* **2017**, *40*, 40.
11. Paulavičius, R.; Grigaitis, S.; Filatovas, E. A Systematic Review and Empirical Analysis of Blockchain Simulators. *IEEE Access* **2021**, *9*, 38010–38028. [CrossRef]
12. Polge, J.; Ghatpande, S.; Kubler, S.; Robert, J.; Le Traon, Y. BlockPerf: A Hybrid Blockchain Emulator/Simulator Framework. *IEEE Access* **2021**, *9*, 107858–107872. [CrossRef]
13. Kaur, M.; Khan, M.Z.; Gupta, S.; Noorwali, A.; Chakraborty, C.; Pani, S.K. MBCP: Performance Analysis of Large Scale Mainstream Blockchain Consensus Protocols. *IEEE Access* **2021**, *9*, 80931–80944. [CrossRef]
14. Nolan, S. PBFT—Understanding the Algorithm. Coinmonks (Blog). Available online: https://medium.com/coinmonks/pbft-understanding-the-algorithm-b7a7869650ae (accessed on 27 August 2020).
15. Xinyi, Y.; Yi, Z.; He, Y. Technical Characteristics and Model of Blockchain. In Proceedings of the 2018 10th International Conference on Communication Software and Networks (ICCSN), Chengdu, China, 6–9 July 2018; pp. 562–566. [CrossRef]
16. Hao, Y.; Li, Y.; Dong, X.; Fang, L.; Chen, P. Performance Analysis of Consensus Algorithm in Private Blockchain. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 280–285. [CrossRef]
17. Lee, I.; Lee, K. The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises. *Bus. Horiz.* **2015**, *25*, 431–440. [CrossRef]
18. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
19. Gayialis, S.P.; Konstantakopoulos, G.D.; Kechagias, E.P.; Papadopoulos, G.A. An Advanced Transportation System Based on Internet of Things. In Proceedings of the 10th Annual International Conference on Industrial Engineering and Operations Management (IEOM 2020), Dubai, United Arab Emirates, 10–12 March 2020; pp. 10–12.
20. Gayialis, S.P.; Kechagias, E.P.; Konstantakopoulos, G.D.; Papadopoulos, G.A. A Predictive Maintenance System for Reverse Supply Chain Operations. *Logistics* **2022**, *6*, 4. [CrossRef]

21. Khan, M.A.; Salah, K. IoT Security: Review, Blockchain Solutions, and Open Challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411. [CrossRef]
22. Gayialis, S.P.; Kechagias, E.P.; Konstantakopoulos, G.D.; Papadopoulos, G.A.; Tatsiopoulos, I.P. An Approach for Creating a Blockchain Platform for Labeling and Tracing Wines and Spirits. In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Nantes, France, 5–9 September 2021; Springer: Cham, Switzerland, 2021; pp. 81–89. [CrossRef]
23. Alkhateeb, A.; Catal, C.; Kar, G.; Mishra, A. Hybrid Blockchain Platforms for the Internet of Things (IoT): A Systematic Literature Review. *Sensors* **2022**, *22*, 1304. [CrossRef]
24. Kreku, J.; Vallivaara, V.A.; Halunen, K.; Suomalainen, J.; Ramachandran, M.; Muñoz, V. Evaluating the Efficiency of Blockchains in IoT with Simulations. *IoTBDS* **2017**, *820*, 216–223. [CrossRef]
25. Banno, R.; Shudo, K. Simulating a Blockchain Network with SimBlock. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 May 2019. [CrossRef]
26. Gupta, Y.; Shorey, R.; Kulkarni, D.; Tew, J. The Applicability of Blockchain in the Internet of Things. In Proceedings of the 2018 10th International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, 3–7 January 2018; pp. 561–564. [CrossRef]
27. Dedeoglu, V.; Jurdak, R.; Putra, G.D.; Dorri, A.; Kanhere, S.S. A Trust Architecture for Blockchain in IoT. In Proceedings of the Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Houston, TX, USA, 25 June 2019; ACM: New York, NY, USA, 2019; pp. 190–199. [CrossRef]
28. Yazdinejad, A.; Srivastava, G.; Parizi, R.M.; Dehghantanha, A.; Choo, K.K.R.; Aledhari, M. Decentralized Authentication of Distributed Patients in Hospital Networks Using Blockchain. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2146–2156. [CrossRef]
29. Moon, H.S.; Song, J.; Shin, H.; Jang, J. Home IoT Device Management Blockchain Platform Using Smart Contracts and a Countermeasure against 51% Attacks. In Proceedings of the 2022 4th Asia Pacific Information Technology Conference, New York, NY, USA, 14 January 2022; pp. 191–195. [CrossRef]
30. Augusto, L.; Costa, R.; Ferreira, J.; Jardim-Gonçalves, R. An Application of Ethereum Smart Contracts and IoT to Logistics. In Proceedings of the 2019 International Young Engineers Forum (YEF-ECE), Costa da Caparica, Portugal, 10 May 2019; pp. 1–7. [CrossRef]
31. Gupta, H.; Vahid Dastjerdi, A.; Ghosh, S.K.; Buyya, R. IFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments. *Softw. Pract. Exp.* **2016**, *47*, 1275–1296. [CrossRef]
32. Vangala, A.; Sutrala, A.K.; Das, A.K.; Jo, M. Smart Contract-Based Blockchain-Envisioned Authentication Scheme for Smart Farming. *IEEE Internet Things J.* **2021**, *8*, 10792–10806. [CrossRef]
33. Assaqty, M.I.; Gao, Y.; Hu, X.; Ning, Z.; Leung, V.C.; Wen, Q.; Chen, Y. Private-Blockchain-Based Industrial IoT for Material and Product Tracking in Smart Manufacturing. *IEEE Netw.* **2020**, *34*, 91–97. [CrossRef]
34. Ray, P.P.; Dash, D.; Salah, K.; Kumar, N. Blockchain for IoT-Based Healthcare: Background, Consensus, Platforms, and Use Cases. *IEEE Syst. J.* **2021**, *15*, 85–94. [CrossRef]
35. Singla, K.; Bose, J.; Katariya, S. Machine Learning for Secure Device Personalization Using Blockchain. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018; pp. 67–73. [CrossRef]
36. Oham, C.; Michelin, R.A.; Jurdak, R.; Kanhere, S.S.; Jha, S. B-FERL: Blockchain Based Framework for Securing Smart Vehicles. *Inf. Process. Manag.* **2007**, *58*, 10528. [CrossRef]
37. Pavithran, D.; Shaalan, K.; Al-Karaki, J.N.; Gawanmeh, A. Towards Building a Blockchain Framework for IoT. *Clust. Comput.* **2020**, *23*, 2089–2103. [CrossRef]
38. Alharby, M.; Van Moorsel, A. BlockSim: A Simulation Framework for Blockchain Systems. *ACM Sigmetr. Perform. Eval. Rev.* **2019**, *46*, 135–138. [CrossRef]
39. Nasir, Q.; Qasse, I.A.; Abu Talib, M.; Nassif, A.B. Performance Analysis of Hyperledger Fabric Platforms. *Secur. Commun. Netw.* **2018**, *2018*, e3976093. [CrossRef]
40. Jiang, Y.; Wang, C.; Wang, Y.; Gao, L. A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management. *Sensors* **2019**, *19*, 2042. [CrossRef]
41. Honar Pajooh, H.; Rashid, M.; Alam, F.; Demidenko, S. Hyperledger Fabric Blockchain for Securing the Edge Internet of Things. *Sensors* **2021**, *21*, 359. [CrossRef]
42. Hang, L.; Ullah, I.; Kim, D.H. A Secure Fish Farm Platform Based on Blockchain for Agriculture Data Integrity. *Comput. Electron. Agric.* **2020**, *170*, 105251. [CrossRef]
43. Alharby, M.; van Moorsel, A. BlockSim: An Extensible Simulation Tool for Blockchain Systems. *Front. Blockchain* **2020**, *3*, 28. [CrossRef]
44. Namasudra, S.; Sharma, P.; Crespo, R.G.; Shanmuganathan, V. Blockchain-Based Medical Certificate Generation and Verification for IoT-Based Healthcare Systems. *IEEE Consum. Electron. Mag.* **2022**, 1. [CrossRef]
45. Son, D.H.; Quynh, T.T.T.; Khoa, T.V.; Hoang, D.T.; Trung, N.L.; Ha, N.V.; Niyato, D.; Nguyen, D.N.; Dutkiewicz, E. An Effective Framework of Private Ethereum Blockchain Networks for Smart Grid. In Proceedings of the 2021 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, Vietnam, 14–16 October 2021; pp. 312–317. [CrossRef]

46. NS-3 Network Simulators. Available online: https://www2.nsnam.org/tutorials/NS-3-LABMEETING-1.pdf (accessed on 22 May 2022).
47. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. LSB: A Lightweight Scalable Blockchain for IoT Security and Anonymity. *J. Parallel Distrib. Comput.* **2019**, *134*, 180–197. [CrossRef]
48. Foytik, P.; Shetty, S.; Gochhayat, S.P.; Herath, E.; Tosh, D.; Njilla, L. A Blockchain Simulator for Evaluating Consensus Algorithms in Diverse Networking Environments. In Proceedings of the Spring Simulation Conference (SpringSim 2020), Fairfax, VA, USA, 20 May 2020. [CrossRef]
49. Ganache 7 Ethereum Simulator—Building on Web3 Is Now Easier and Faster than Ever before—Truffle Suite. Available online: https://trufflesuite.com/blog/introducing-ganache-7/ (accessed on 29 April 2022).
50. Fakhri, D.; Mutijarsa, K. Secure IoT Communication Using Blockchain Technology. In Proceedings of the 2018 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, 23–24 October 2018; pp. 1–6. [CrossRef]
51. Rashmeet, K. Blockchain Simulator: What Is It and How IS It Built? Available online: https://medium.datadriveninvestor.com/blockchain-simulator-what-is-it-and-how-is-it-built-811b122075a (accessed on 5 January 2022).
52. Switzerland, D. Can Blockchain Accelerate Internet of Things (Iot) Adoption? Available online: https://www2.deloitte.com/ch/en/pages/innovation/articles/blockchain-accelerate-iot-adoption.html (accessed on 25 June 2019).
53. Patruni, M.R.; Saraswathi, P. Securing Internet of Things Devices by Enabling Ethereum Blockchain Using Smart Contracts. Available online: https://journals.sagepub.com/doi/full/10.1177/01436244221078933 (accessed on 9 April 2022).
54. Habib, Y. Introducing JABS: Just Another Blockchain Simulator. Available online: https://hackernoon.com/introducing-jabs-just-another-blockchain-simulator (accessed on 14 February 2022).
55. Ma, X.; Wu, H.; Xu, D.; Wolter, K. CBlockSim: A Modular High-Performance Blockchain Simulator. Available online: https://arxiv.org/pdf/2203.05788.pdf (accessed on 11 March 2022).
56. Kenneth, R. The Rise of the New Blockchains. Where Are Investors and Developers Turning? Available online: https://www.forbes.com/sites/kenrapoza/2022/01/17/the-rise-of-the-new-blockchains-where-are-investors-and-developers-turning/?sh=4affa9c61425 (accessed on 17 January 2022).
57. Heirman, W.; Sarkar, S.; Carlson, T.E.; Hur, I.; Eeckhout, L. Power-aware multi-core simulation for early design stage hardware/software co-optimization. In Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques, New York, NY, USA, 19–23 September 2012; pp. 3–12. [CrossRef]
58. Hanggoro, D.; Sari, R.F. Performance Comparison of SimBlock to NS-3 Blockchain Simulators. In Proceedings of the 2021 4th International Conference on Circuits, Systems and Simulation (ICCSS), Kuala Lumpur, Malaysia, 26–28 May 2021; pp. 45–50. [CrossRef]
59. Issariyakul, T.; Hossain, E. *Introduction to Network Simulator NS2*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
60. Abreu, D.P.; Velasquez, K.; Curado, M.; Monteiro, E. A Comparative Analysis of Simulators for the Cloud to Fog Continuum. *Simul. Model. Pract. Theory* **2020**, *101*, 102029. [CrossRef]
61. Khan, A.A.; Uddin, M.; Shaikh, A.A.; Laghari, A.A.; Rajput, A.E. MF-Ledger: Blockchain Hyperledger Sawtooth-Enabled Novel and Secure Multimedia Chain of Custody Forensic Investigation Architecture. *IEEE Access* **2021**, *9*, 103637–103650. [CrossRef]
62. Ahrenholz, J.; Danilov, C.; Henderson, T.R.; Kim, J.H. Core: A Real-Time Network Emulator. In Proceedings of the MILCOM 2008-2008 IEEE Military Communications Conference, San Diego, CA, USA, 16–19 November 2008; pp. 1–7. [CrossRef]
63. Mehmood, T. COOJA Network Simulator: Exploring the Infinite Possible Ways to Compute the Performance Metrics of IOT Based Smart Devices to Understand the Working of IOT Based Compression & Routing Protocols. *arXiv* **2017**, arXiv:1712.08303.