PuPrChain: A Public-Private Cluster-Based Distributed Blockchain System

Lalithsai Posam
Evergreen Valley High School
lsmadhava@gmail.com

Faisal Nawab
University of California, Irvine
nawabf@uci.edu

***Abstract*—Current large-scale IoT systems suffer from several security concerns due to varying system configurations, participation of multiple organizations across different control and trust domains, and centralization of data. We propose *Public-Private Chain* (PuPrChain), a cluster-based distributed blockchain system that is able to address security and trust issues within IoT systems while also increasing speed and scalability. PuPrChain consists of a hybrid public-private, hierarchical system structure in which clusters interact across a global network with all communication stored in a public blockchain, but each cluster also contains its own network maintained by a private blockchain. This allows PuPrChain to achieve a secure and decentralized environment while also maintaining intra-cluster and inter-cluster coordination.**

***Index Terms*—blockchain, data management, IoT**

I. Introduction

Blockchain is a decentralized data infrastructure used to record transactions [7]. These transactions are recorded using several blocks linked together in an immutable fashion, providing high levels of system security and trust; this immutability creates an environment for different nodes to upload, share, and store data safely [8].

Current Internet of Things (IoT) systems suffer from security and trust issues that blockchain can address. One problem is that IoT devices may transfer user data to malicious participants due to a cyberattack [14]. Another issue is the lack of authorization when adding new devices into the network, which can lead to trust issues between nodes [6]. Moreover, the centralized nature of current IoT systems puts the data of over 14 billion IoT devices [10] at risk in the hands of the companies behind these central authorities. The secure and decentralized characteristics of blockchain make it an ideal solution to address these concerns.

However, blockchain implementation with IoT systems is challenging for multiple reasons. IoT systems generate thousands of requests per second, which prevents blockchain from being able to scale to IoT demands [1]. Furthermore, simultaneous, conflicting transactions pose an issue for ordering transactions within logs [3]. IoT devices also have to constantly update their personal logs with new transactions, rapidly filling their storage. Furthermore, the sequential execution of transactions, mining, and consensus within blockchains are intensive tasks that can cause high latency [15]. Permissioned blockchains are a possible solution, but the underlying protocol within those systems fails to keep up with large applications [9].

In this paper, we propose a cluster-based blockchain system that is able to address two major issues with the integration of blockchain and IoT while also providing a high level of security that IoT systems currently lack. PuPrChain utilizes a combination of edge data centers, clusters, and public and private blockchains to create a distributed system that supports the demands of modern IoT systems.

The main contribution of PuPrChain is its hierarchical cluster structure that allows local networks to be compartmentalized, while also providing customized access and roles. Clusters have the ability to represent entities in a complex process. The use of clusters allows for separate protocols to be developed based on the characteristics of each cluster, such as size or use case. The supply chain example [4] shows the advantages of PuPrChain's cluster-based structure.

A traditional supply chain consists of several groups. It begins with the supplier, who collects and sends the raw materials to the manufacturer. The manufacturer builds the product and sends it through a transportation company to a wholesaler. The wholesaler manages the stock of the product, and then forwards the product to the retailer who sells the product. This process can be integrated into PuPrChain's system. Each of the entities can be represented with its own cluster, and they can engage in inter-cluster communication in order to receive real-time, secure updates regarding ethical sourcing of the raw materials, changes in stock of the product, quality control, and automated transportation updates [11].

Offloading and local edge data centers are two more of PuPrChain's design contributions that allow for quicker computation times and increased scalability. Local edge data centers are assigned to each cluster and because of the close proximity of the data centers, transaction times are much faster when storing and retrieving off-chain data. Offloading occurs when an IoT device requests the assistance of a powerful computer with large-scale transactions, which helps the overall speed of the transaction and scalability as the number of devices in the cluster increases.

A combination of public and private clouds for communication between public and private machines has been explored [13], but PuPrChain takes this a step further through the use of *public and private blockchains*. Each cluster contains its own private blockchain, which enforces an environment where the

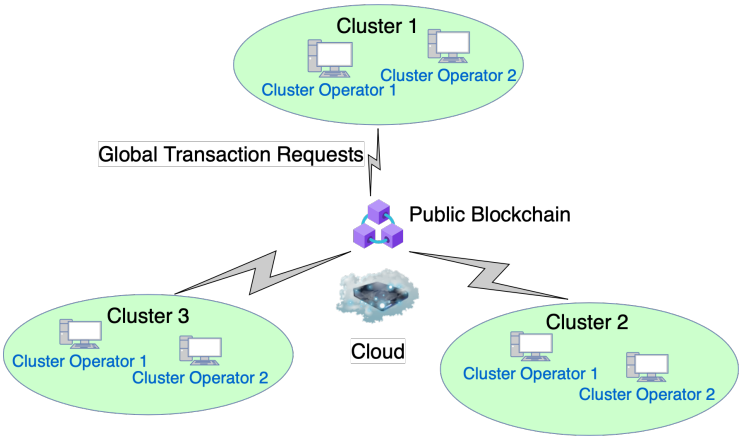

Fig. 1. Cluster-Based Network Model

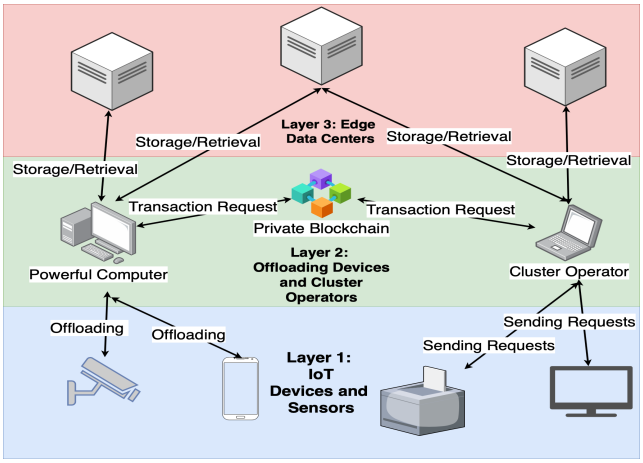

Fig. 2. Structure of a Cluster in PuPrChain. The private blockchain is not included in Layer 2; it is at the center of the entire cluster, but not in any particular layer

real identities of each cluster node are known. This allows clusters to act as trusted, real-world organizations. The public blockchain is for global communication across clusters.

Sharding is another research direction that can improve blockchain-IoT system performance. It involves partitioning the data into several shards that are maintained by different groups of nodes that allows the database to scale horizontally to the number of nodes [2]. Databases would be split such that data is as independent as possible to reduce the number of cross-shard transactions. Although PuPrchain does not directly shard the data, it uses similar principles of local and global data sharing to boost performance of the system. The use of clusters as local, trusted data sharing environments allows nodes within the same cluster to quickly transfer data to one another, and the cluster acts as a security blanket to prevent unauthorized users from viewing the data.

Another approach that has been explored to boost the security of IoT-blockchain systems is side chain [5]. In some IoT systems, devices can travel over long distances such as those on an aircraft or ship. The main issue is that the data from the traveling IoT devices can be integrated in a blockchain outside of its home network, which can be categorized as an unauthorized transaction. Furthermore, the transition of the blockchain transactions back to the home network can be difficult due to the immutable nature of blockchain [16]. Side chains offer a solution that can transfer assets and data between blockchains in a decentralized manner, allowing the data generated by the traveling devices to be transferred back to the home network. PuPrchain does not have a designated data transfer scheme for traveling IoT devices; however, the cluster structure itself allows devices to remain in their home cluster and have access to intra-cluster data sharing methods. Cluster operators would authorize these transactions and ensure that the traveling IoT device belongs in the cluster. This bypasses the issue of the an IoT device initiating a transaction in a cluster outside of its designated cluster.

II. Background

This section introduces the technologies used in PuPrChain.

A. Public Blockchain

A public blockchain indicates an open membership system in which all information regarding executed transactions is public to any interested party and accessible through the blockchain itself [12].

B. Private Blockchain

A private blockchain is a system that only allows users who are authorized by those currently in the network. The ability to transfer information is much faster because the real-world identity of each node in the network is known. Moreover, computationally expensive consensus algorithms are not needed, allowing for high performance and scalability [12].

C. Smart Contracts

Smart contracts are programs that execute certain tasks once the predefined conditions are reached, and they can facilitate the agreement between two parties automatically [18].

III. Cluster-Based Design and Prototype

A. System Model and Design Considerations

PuPrChain consists of a group of clusters (each cluster runs a private blockchain), one public blockchain, edge data centers for local storage, and the cloud for global storage. Within each cluster (Figures 1 and 2), there are three layers: IoT Devices and Sensors (Layer 1), Offloading Devices and Cluster Operators (Layer 2), and the edge computational data centers (Layer 3).

In the rest of this section, we describe the cluster model and operation followed by the intra-cluster and inter-cluster coordination.

Layer 1: IoT Devices and Sensors. Layer 1 consists of IoT devices and sensors. These are low-power devices and constantly make decisions regarding the collection, processing,

and filtering of data. These devices have minimal storage capabilities, which are mainly used for logging all transactions [1] occurring in the cluster. The IoT devices will be in constant communication with upper layers, containing the offloading devices and edge data centers, which have much larger data storage capabilities.

Each IoT device will have its own log, in which there is a copy of all transactions that have occurred in the cluster so far. All IoT devices are expected to have the same order of transactions, and if there are any discrepancies, cluster operators intervene to ensure that all devices have the same order.

Layer 2: Offloading Devices and Cluster Operators. There are two different devices in layer 2 of the cluster: offloading devices and cluster operators. Offloading devices' main task is to assist any IoT devices with a transaction that may be too computationally expensive for the IoT device itself. The IoT device can communicate with the offloading device and send information to help complete the transaction; global transactions specifically can be computationally expensive due to having large amounts of data that need to be transmitted to another cluster. Other tasks, such as the generation of hashes and encryption, can be outsourced to an offloading device. Offloading will assist with scaling as the cluster increases in the number of devices and the number of transactions, and it still maintains a high level of security because this process is overlooked by cluster operators.

Cluster operators serve the administrative role in PuPrChain. When a device would like to join the cluster, the cluster operators will cooperatively decide whether to allow the device to join. Moreover, cluster operators are able to assign roles to devices and designate devices as IoT devices, cluster operators, or offloading devices, which are visible to all devices in the network.

Layer 3: Edge Data Centers. The third layer consists of edge data centers that will store off-chain data such as images, videos, and documents. These high power data centers provide local storage for all off-chain data. IoT devices can issue a local transaction through the cluster operator, which stores the off-chain data in the edge data center. Edge data centers will also lead to enhanced security since devices would not have to store cluster-based information in a public cloud, and cluster operators have greater control over their own cluster-based data.

Private Blockchain Each cluster will also have its own private blockchain, which will serve as a sequential and immutable ordering of all transactions within the cluster. The cluster can represent a trusted and verified organization because of the transparency of identities of each device in the cluster. A private blockchain is also utilized to avoid malicious nodes from entering the system and causing security issues, and only authorized roles are able to access certain information.

[1]In this paper, the term "transactions" refers to actions done by any device, such as data retrieval, storage, or transfer

Nodes responsible for data generation and storage within a cluster are assumed to be in the same security and control domain, and thus, trust each other. Untrust is in inter-cluster coordination, and is the reason why we need trustful solutions for their coordination.

Cluster Network and Public Blockchain Clusters participate in a public blockchain that tracks global transactions, as shown in Figure 1. Cluster operators will lead all global communication, so any IoT device that would like to transfer data to an IoT device in another cluster must verify their identity with the cluster operator. Then, the transaction must be validated by the public blockchain. This creates two layers of security and masks malicious acts within a cluster via the private blockchain. This ensures that malicious transactions are not being sent to the public blockchain, effectively lowering the number of transactions that have to be evaluated.

Within cross-cluster communication, each cluster will have its own log that tracks both global and local transactions. This increases efficiency and eases storage requirements for IoT devices as they no longer have to store global communication within their storage; they only have to store transactions occurring in their own cluster. IoT devices are able to access the log of the cluster they are a part of and will have different levels of access depending on their assigned roles.

The Cloud. The cloud is the main storage system for cross-cluster communication, and it hold all global off-chain data, such as images, documents, or tabulated data. If a device requests data to be stored in the cloud, it will receive an encrypted hash pointing to the off-chain data once the transaction has been validated and added to the public blockchain. If a device requests data to be sent to another device in a separate cluster, then off-chain data will be stored in the cloud and a hash pointing to the data will be returned to both devices for future data access.

B. Protocol Details

Figure 3 displays the pathway for a local, cluster-based transaction in PuPrChain, which can either be a request to store data in a local edge datacenter or transfer the data to another IoT device. The device begins by gathering all transactional data (Step 1) and sends it to the cluster operators, whom will verify the identity of the requester (Step 2) and then forward it to the private blockchain that will either validate or not validate the transaction (Step 3). If the transaction is validated, it is grouped into the next available block (Step 4), and an encrypted hash is generated for the off-chain data (Step 5). Next, the block is added to the cluster's blockchain (Step 6) and the off-chain data is stored in the edge data center (Step 7). The encrypted hash used to access the data is returned to the IoT device that requests the transaction. and the event is updated on every device's log. If there was a recipient for the data transfer, then a hash to access the data is also sent to the destination IoT device as well (Step 8). The process ends with all of the devices within the cluster updating their personal logs (Step 9), along with the cluster operators updating the cluster log.

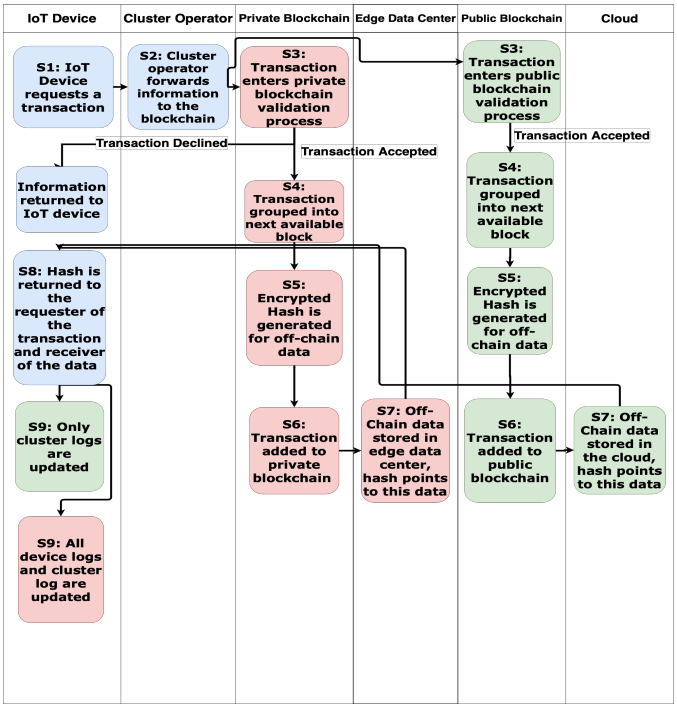

Fig. 3. Flowchart of Local and Global Communication in PuPrChain, Red: Local Protocol, Green: Global Protocol.

Figure 3 also displays the pathway for a global, inter-cluster transaction in PuPrChain. The IoT device requests a transaction to be sent to the blockchain along with data to be stored in the cloud (Step 1). The IoT device also sends its public key and digital signatures to the cluster operator so that the cluster operator (Step 2) can approve the transaction to continue to the public blockchain for validation (Step 3). If the transaction is accepted, then it is sent to the smart contracts, which will format the transaction and add it with several other transactions until the maximum block size is reached. If the transaction cannot fit into the current block, then a new block is created and the transaction is added to the new block (Step 4). After the transaction has been formatted, the smart contracts generate a unique hash to represent the data associated with the transaction (Step 5). The block is then added to the blockchain (Step 6) and the off-chain data associated with the transaction is stored in the cloud (Step 7). This hash is stored in the cloud for quick retrieval by the clusters whenever the information needs to be accessed. The encrypted hash used to access the data is returned to the IoT device that requests the transaction and the event is updated on every device's log. If there was a recipient for the data transfer, then a hash to access the data is also sent to the destination IoT device as well (Step 8). The process ends with only cluster logs being updated (Step 9).

IV. Experimental Evaluation

In this section, we introduce the experimental design and the tools used to create the PuPrChain prototype.

A. Prototype Tools

Ganache Ganache is an Ethereum [17] client framework that can simulate an Ethereum network to test smart contracts. This generated network was chosen because of its similarities to our cluster-cluster network design; both networks contain a public blockchain with open membership.

Web3.py Web3.py is a Python library used to directly interact with the Ethereum network hosted by Ganache. It helps initiate transactions, interact with smart contracts, and reads block data.

Truffle and Solidity Truffle is an environment that allows for quick development and testing of these smart contracts. Solidity is the object-oriented programming language used to write smart contracts.

B. Experimental Design and Results

To artificially simulate three separate cluster networks, the default accounts that were automatically generated in Ganache's environment serve as endpoints within the network. In this prototype, the accounts were split into three separate clusters, and each cluster has its own separate set of smart contracts maintained by the cluster operators. All of the clusters were connected to the same Ethereuem network hosted by Ganache, and two separate experiments were conducted based on varying transaction sizes.

The baseline for the experiments is a traditional blockchain-IoT system with several IoT devices and one public blockchain, without any form of clusters. The Ganache accounts serve the role of IoT devices, and they send transactions back and forth through direct communication with the public blockchain. All transactions were storeed on their personal logs. In the traditional blockchain-IoT system, all transactions were initiated from a particular IoT device and sent to another IoT device in the network, and the remaining IoT devices would follow the broadcast-update procedure once the transaction was confirmed to the blockchain. In PuPrchain, the transactions were initiated from a particular IoT device in a specific cluster and sent to an IoT device in another cluster, and the remaining cluster operators would update their cluster-based logs.

Figure 4 shows the duration of time required to broadcast and update logs with global transactions in a traditional blockchain-IoT system versus PuPrChain. PuPrChain has a significant advantage because of the use of local clusters; the proposed system only has to update the cluster based logs and then forwards access of the cluster log to all nodes in the cluster. This leads to an average of a 65.8% decrease in times required to broadcast the transaction and update all logs. On the other hand, traditional blockchain-IoT integrations have to to broadcast the transaction to all of the nodes and update all of their personal logs, leading to high broadcast and update times. This is the main roadblock that prevents current baseline systems from expanding to larger-scale systems, and PuPrchain avoids this completely.

Figure 5 displays the storage space taken by all logs in traditional blockchain IoT systems versus PuPrChain. Because

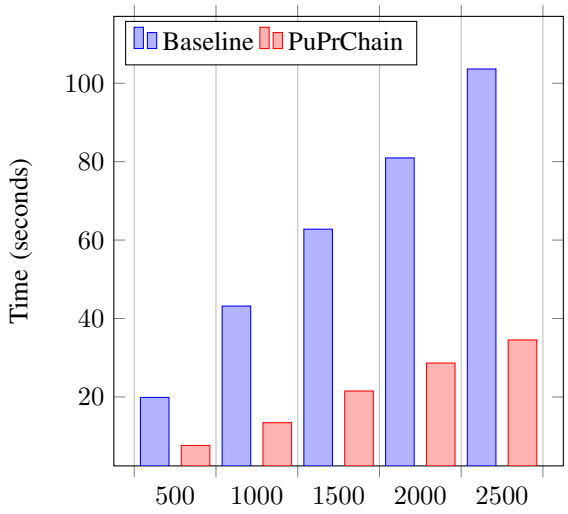

Fig. 4. Time required to update logs for PuPrChain versus baseline over varying transaction sizes

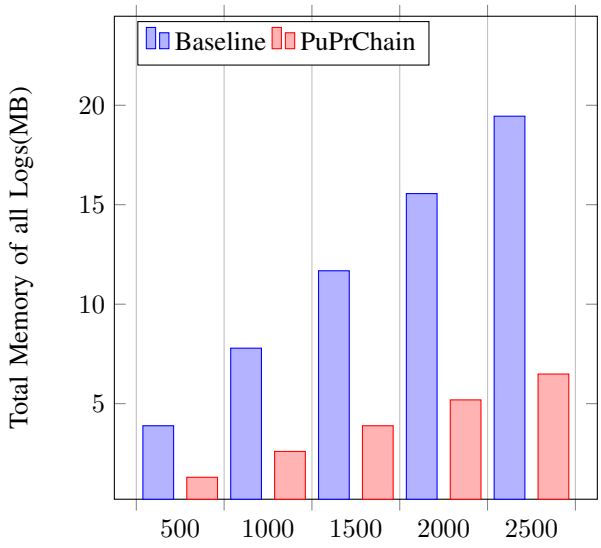

Fig. 5. Storage Space Required for all Logs for PuPrChain versus baseline over varying transaction sizes

PuPrchain only broadcasts transactions to cluster logs, the amount of memory required to store information about all of the transactions reduces by over 66.2% on average across all transaction sizes. The baseline is forced to have all the nodes' logs store information from the transactions, while PuPrchain takes advantage of its cluster structure to only store information in the cluster logs. This drastically reduces the amount of memory required, which improves scalability as the number of clusters increase.

V. Conclusions

In our paper, we presented PuPrChain. PuPrChain is a hybrid solution that integrates public and private blockchains in a cluster network. PuPrChain is able to achieve faster log update times and minimize storage requirements by using established communication with groups of devices in clusters, effectively lowering the number of devices that need to receive active updates and broadcasts of transactions. The private blockchains hosted in each cluster create a secure and trusted environment that can represent real-world entities.

References

[1] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. "CAPER: A Cross-Application Permissioned Blockchain". In: *Proceedings of the VLDB Endowment* 12 (2019), pp. 1385–1398. DOI: 10.14778/3342263.3342275.

[2] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. "On Sharding permissioned blockchains". In: *2019 IEEE International Conference on Blockchain (Blockchain)* (2019), pp. 282–285. DOI: 10.1109/blockchain.2019.00044.

[3] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. "Parblockchain: Leveraging transaction parallelism in permissioned Blockchain Systems". In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (Oct. 2019). DOI: 10.1109/icdcs.2019.00134.

[4] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. "Permissioned blockchains: Properties, techniques and applications". In: *Proceedings of the 2021 International Conference on Management of Data* (June 2021). DOI: 10.1145/3448016.3457539.

[5] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Timón Jorge, and Pieter Wuillex. *Enabling blockchain innovations with pegged sidechains*. Oct. 2014. URL: https://www.blockstream.com/sidechains.pdf.

[6] Ahmed Banafa. *IOT and Blockchain Convergence: Benefits and challenges*. Jan. 2017. URL: https://iot.ieee.org/newsletter/january-2017/iot-and-blockchain-convergence-benefits-and-challenges.html.

[7] Christian Cachin and Marko Vukolic. "Blockchain Consensus Protocols in the Wild (Keynote Talk)". In: *31st International Symposium on Distributed Computing (DISC 2017)*. Ed. by Andréa W. Richa. Vol. 91. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 1:1–1:16. ISBN: 978-3-95977-053-8. DOI: 10.4230/LIPIcs.DISC.2017.1. URL: http://drops.dagstuhl.de/opus/volltexte/2017/8016.

[8] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan, and Neeraj Kumar. "A survey on privacy protection in Blockchain System". In: *Journal of Network and Computer Applications* 126 (2019), pp. 45–58. DOI: 10.1016/j.jnca.2018.10.020.

[9] Suyash Gupta, Jelle Hellings, Sajjad Rahnama, and Mohammad Sadoghi. "Building high throughput permissioned Blockchain Fabrics". In: *Proceedings of the*

VLDB Endowment 13.12 (Sept. 2020), pp. 3441–3444. DOI: 10.14778/3415478.3415565.

[10] Mohammad Hasan. *State of IOT 2022: Number of connected IOT devices growing 18% to 14.4 billion globally*. May 2022. URL: https://iot-analytics.com/number-connected-iot-devices/.

[11] Tiana Laurence. *Blockchain for Supply Chain: Track and Trace*. 2019. URL: https://aws.amazon.com/blockchain/blockchain-for-supply-chain-track-and-trace/.

[12] C. Mohan. "State of public and private blockchains". In: *Proceedings of the 2019 International Conference on Management of Data* (2019), pp. 404–411. DOI: 10.1145/3299869.3314116.

[13] Kerim Yasin Oktay, Sharad Mehrotra, Vaibhav Khadilkar, and Murat Kantarcioglu. "SEMROD: Secure and Efficient MapReduce Over HybriD Clouds". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (May 2015), pp. 153–166. DOI: 10.1145/2723372.2723741.

[14] Nisha Panwar, Shantanu Sharma, Guoxi Wang, Sharad Mehrotra, Nalini Venkatasubramanian, Mamadou H. Diallo, and Ardalan Amiri Sani. "IOT notary: Attestable Sensor Data Capture in iot environments". In: *ACM Transactions on Internet of Things* 3.1 (2021), pp. 1–30. DOI: 10.1145/3478290.

[15] Ana Reyna, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. "On blockchain and its integration with IOT. challenges and opportunities". In: *Future Generation Computer Systems* 88 (2018), pp. 173–190. DOI: 10.1016/j.future.2018.05.046.

[16] Xu Wang, Xuan Zha, Wei Ni, Ren Ping Liu, Y. Jay Guo, Xinxin Niu, and Kangfeng Zheng. "Survey on blockchain for internet of things". In: *Computer Communications* 136 (2019), pp. 10–29. DOI: 10.1016/j.comcom.2019.01.006.

[17] Gavin Wood. "Ethereum: A secure decentralised generalised transaction ledger". In: ().

[18] Weiqin Zou, David Lo, Pavneet Singh Kochhar, Xuan-Bach Dinh Le, Xin Xia, Yang Feng, Zhenyu Chen, and Baowen Xu. "Smart contract development: Challenges and opportunities". In: *IEEE Transactions on Software Engineering* 47.10 (2021), pp. 2084–2106. DOI: 10.1109/tse.2019.2942301.