2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

A Blockchain-based Decentralized Data Storage and Access Framework for PingER

Saqib Ali†, Guojun Wang*
School of Computer Science and Technology, Guangzhou University, Guangzhou, P. R. China, 510006
Emails: {saqibali,csgjwang}@gzhu.edu.cn

Bebo White
Stanford Linear Accelerator Center
P.O. Box 4349,
Palo Alto, California 94309-4349
Email: bebo@slac.stanford.edu

Roger Leslie Cottrell
Stanford Linear Accelerator Center
P.O. Box 4349,
Palo Alto, California 94309-4349
Email: cottrell@slac.stanford.edu

***Abstract*—The blockchain is an innovative technology which opened doors to new applications for solving numerous problems in distributed environments. In this work, we design a blockchain-based data storage and access framework for PingER (worldwide end-to-end Internet performance measurement project) to remove its total dependence on a centralized repository. We use the permissioned blockchain and Distributed Hash Tables (DHT) for this purpose. In the proposed framework, metadata of the files are stored on the blockchain whereas the actual files are stored off-chain through DHT at multiple locations using a peer-to-peer network of PingER Monitoring Agents. This will provide decentralized storage, distributed processing, and efficient lookup capabilities to the PingER framework.**

***Index Terms*—Permissioned blockchain, Distributed ledger technology, PingER, Decentralized system**

I. Introduction

The blockchain is a peer-to-peer distributed ledger in which records called blocks are linked and secured using a cryptographic hash [1]. By design, blockchains are decentralized, secure, immutable and extremely fault tolerance making them suitable for record management activities i.e., financial transactions, identity management, provenance and authentication [2]. Blockchain can be deployed as permissionless (e.g., Bitcoin [3] or Ethereum [4]) or permissioned blockchain e.g., Hyperledger Project by The Linux Foundation. In permissionless or public blockchain the actors in the system are not known. Anyone can join or leave the blockchain network at any time, which may raise security risks in the network. However, in permissioned or private blockchain only known and identifiable set of participants are explicitly admitted to the blockchain network [5]. This reduces the presence of malicious actors within the network. As a result, only authenticated and authorized actors can participate in the network which increases the security of the system as required by the enterprise applications [6].

The concept of permissioned blockchain is gaining a lot of interest especially for non-financial use cases (other than the cryptocurrencies) in which the users are authenticated and authorized to participate in the network [7]. The interesting non-financial areas that leverage the opportunities of permissioned blockchains include health, government services, supply chain management, Internet of Things, peer-to-peer cloud storage and many more [8]–[10]. The P2P cloud storage (e.g., STORJ[1], Sia[2], Filecoin[3]) is an interesting application of blockchain as it provides a decentralized data storage facility without involving any trusted third party or a client-server architecture. The decentralized data storage will help to eliminate the most traditional data failures and outages by increasing the security, privacy, and control of the data [11], [12].

PingER (Ping End-to-end Reporting)[4] is a worldwide end-to-end Internet performance measurement framework developed and managed by the SLAC National Accelerator Laboratory USA [13]. The client-server architecture of PingER consists of 50 active Monitoring Agents (MAs) in 20 countries of the world. These MAs probe 700 remote sites located in 170 countries of the world [14]. The ping statistics for each MA-remote site pair are stored on a local MA. The data archived by each MA is fetched daily by SLAC to a centralized repository of text archives as shown in Figure 1.

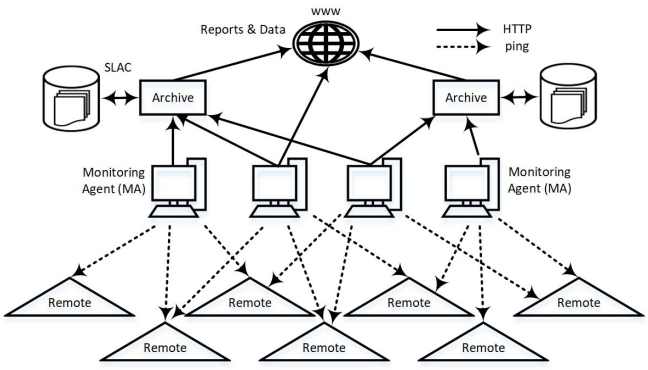

Fig. 1. PingER: Data storage & access architecture

The current architecture of the PingER is highly dependent on the SLAC computing resources, e.g., archival space,

†Department of Computer Science, University of Agriculture, Faisalabad, Pakistan, 38000. Email: saqib@uaf.edu.pk
*Correspondence to: csgjwang@gzhu.edu.cn

[1]https://storj.io
[2]https://sia.tech/
[3]https://filecoin.io/
[4]www-iepm.slac.stanford.edu/pinger/

processing and uptime. Therefore, there is a need to ensure the future of the PingER data if/when SLAC support goes away. Further, different PingER nodes have different tasks, different privileges and different software installed on them, relationships among them is highly asymmetrical, and the whole network is organized around a SLAC central repository. An interesting solution to these problems might be the PingER blockchain. Theoretically, the second tier (SLAC, Archives, etc. in Figure 1) can be eliminated and the upward paths from the monitoring agents could be replaced by write-access data entries on a PingER blockchain. This would make the PingER architecture decentralized and remove the project dependence upon SLAC (or other archive sites).

The major contributions of this paper are four-fold.

- We design a PingER data storage and access framework by storing the metadata of daily PingER files on the permissioned blockchain whereas the actual data is stored off-chain on multiple MAs.
- We design an off-chain storage of PingER data files using distributed hash tables on multiple MAs in which contents are accessed through hashes with high throughput and low latency.
- We outline the implementation requirements for the development of the blockchain-based PingER framework.
- Finally, we explore the capabilities that the framework can provide aside from the PingER decentralized data storage.

The remaining paper is organized as follows. The preliminaries of blockchain are discussed in Section II. Section III describes the proposed blockchain-based data storage and access framework for PingER. The implementation requirements are given in Section IV. The significance of the framework is outlined in Section V, and finally, Section VI concludes this paper.

II. PRELIMINARIES: BLOCKCHAIN

Blockchains are write-only data structures with no administrative permissions for editing or deleting of the data. The data structures are known as blocks and are distributed in a P2P network. Each block contains the cryptographic hash function of the previous block and is used to develop a link between them. The linked blocks form a complete chain, hence the term *blockchain*. The hash function maintains the security, integrity, and immutability of the blockchain. A typical example of a blockchain is shown in Figure 2. The process of creating new blocks is known as mining. The new blocks are always appended at the end of the blockchain [6], [15]. The main components of the blockchain include Transactions, Blocks, Cryptography, Smart Contracts, Consensus Algorithms, and P2P network. Each component is explained as follows.

i) Transactions & Blocks: The record of an event, cryptographically secured with a digital signature, that is verified, ordered, and bundled together into blocks, form the transactions in the blockchain. Thus, each block is composed of transaction data along with the timestamp, cryptographic hash of the previous block (parent block) and a nonce. A nonce is a random number or bit string which is used to verify the hash. The hash values are unique and help to maintain the integrity of the entire blockchain from the first block (genesis block) to the last in the network [10].

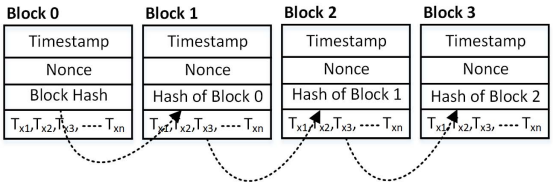

Fig. 2. A blockchain structure

ii) Cryptography: It plays a key role in blockchain by providing the security, immutability and rightful ownership of the transactions being stored on the block. It provides the security and immutability by linking the blocks in a chronological order using the hash function. Note that, the hash only provides the encrypted form of the original transaction from which it is not possible to drive the original transaction data. The examples of hash functions include the family of Secure Hash Algorithms (i.e., SHA1, SHA128, SHA256, SHA512, etc.) [2]. The rightful ownership is provided to the transactions using digital signatures. Further, it helps the receiver to verify the authenticity and integrity of the transactions on the network. For example, Bitcoin uses Elliptic Curve Digital Signature Algorithm (ECDSA) based on asymmetric cryptography [16].

iii) Smart Contracts: *Nick Szabo* coined the term of Smart Contracts for the first time [17]. They are the computer programs that run automatically when certain criteria are met within the system. They are used to transfer value of any kind between the peers in a blockchain without the service of the trusted third party [18]. Today, the Ethereum smart contracts are designed to run on all nodes of the Ethereum network. Similarly, Hyperledger Fabric[5] smart contracts are called *chaincode*. They enable the user to create transactions in the shared ledger of the network.

iv) Consensus Algorithms: They are used in blockchains to achieve the agreement on a single state of the data in a distributed network. They ensure that the same copy of the data is available to all peers in the network. Further, they help to prevent the malicious nodes from changing the state of the data in a distributed environment. Mainly the consensus algorithms are either lottery-based (Proof of Work, Proof of Elapsed Time) or voting based (Simplified Byzantine Fault Tolerance) depending on the unique requirement of the system and level of fault tolerance [19]. Other available consensus algorithms are Proof of Stake, Proof of Deposit, Proof of Burn, Proof

[5]https://www.hyperledger.org/projects/fabric

of Authority, etc. The Bitcoin blockchain uses Proof of Work (PoW) consensus algorithm for creating new blocks by solving a computational puzzle which is hard to create but is easy to verify by other peers in the network [20].

v) Peer-to-peer (P2P) networks: Blockchains run on a P2P network via the Internet without any central server. As a result, they do not have a single point of failure or attack. Each peer contributes to the storage and processing power for the upkeep of the network.

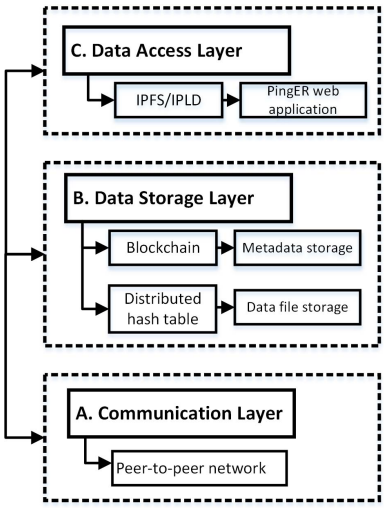

Fig. 3. Blockchain-based data storage and access framework for PingER

III. Blockchain-based Decentralized Data Storage and Access Framework for PingER

In every PingER Monitoring Agent (MA), each sample measurement set is sent every 30 minutes. The MA goes through its list of remote sites by sending up to thirty 100-byte pings at one-second intervals until ten echo replies are received or it times out in 30 seconds. This is then repeated for 1000-byte pings. The data collected for each set of pings consists of an MA name, remote sites, IP addresses, payload in ping request, timestamp, number of ping packets sent & received, response sequence number, minimum, average and maximum Round Trip Time (RTT). All these raw measurements are stored in flat text files at each MA. The centralized data repository at SLAC fetches all the text archives from each MA on daily basis. The data is analyzed and stored with a specific naming pattern containing the name of performance metric, packet size, MA name and the date of the measurement at SLAC data repository e.g., `average_rtt-100-pinger.slac.edu-2018-01-28.txt.gz` [21]. The analyzed data is referred as hourly data from all MAs and is used to generate sixteen Internet performance metrics on a daily, monthly and yearly basis. The data is public and can be downloaded from the *pingtable* web interface[6] (running at the SLAC web server) or by anonymous FTP[7]. Thus, in the current PingER framework, data storing, and processing is centralized and entirely dependent on SLAC resources for analyzing, archiving and reporting.

The above mentioned centralized data storage and access framework of PingER can be replaced with a fully decentralized system using the key features of permissioned blockchain. Therefore, in this work, we adopted a layered approach to translate the current PingER framework into a blockchain-based decentralized data storage and access framework as shown in Figure 3. The proposed design is composed of three layers i.e., Communication layer, Data storage layer and Access layer. Each layer is explained as follows.

A. Communication Layer

In the current PingER deployment, there are nearly 50 MAs located at different geographical locations of the world [14]. Each MA has a client status with no contribution towards data storage, processing, analyzing and reporting. Therefore, in the proposed framework, the status of an MA is raised from a client to a peer by establishing a P2P network of MAs. As a result, each MA will not only help in data collection but also be the place of data storage, processing and facilitating the web services. Moreover, it will provide the communication and storage facilities to the distributed hash-tables and blockchain deployment (discussed in the next layer) as they run on a P2P network via the Internet without any central server.

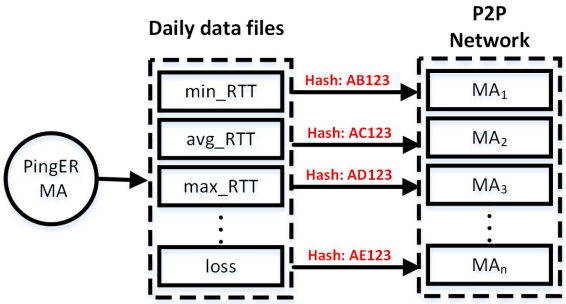

Fig. 4. Off-chain storage of PingER data files

B. Data storage Layer

Blockchains are not general-purpose databases. It is not possible to store daily PingER data on the blockchain as it will quickly scale to an unmanageable size and cause the problem of Blockchain Bloat [22]. Therefore, we use the blockchain to manage the identity and access control on the network by storing only a small amount of metadata information of the PingER data files. Meanwhile, the actual data files are stored off-chain at K-storage locations on the network by using erasure coding `(K-of-M)` [23]. The erasure coding will help to add redundancy to the system against failures. Thus, if a specific MA is off-line, the erasure coding will guarantee the

[6]http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl
[7]ftp://ftp.slac.stanford.edu/users/cottrell/

availability of the data for analysis from other storage locations on the network.

B-1. Distributed Hash Table (DHT): In the proposed framework, each MA analyzes the collected raw data locally based on the performance metrics (e.g., minimum RTT, average RTT, maximum RTT, and packetloss) and stores it with a filename of specific format after every 24 hours i.e., `average_rtt-100-MA_name-2018-01-28.txt.gz`. Note that, each MA creates a separate file for each performance metric every day and stored it locally. Before directly transmitting and storing the files of a specific day at K-storage locations on the network, each file is hashed with a cryptographic function as shown in Figure 4. The hashes of all the files are combined to create a Merkle tree [24] as shown in Figure 5.

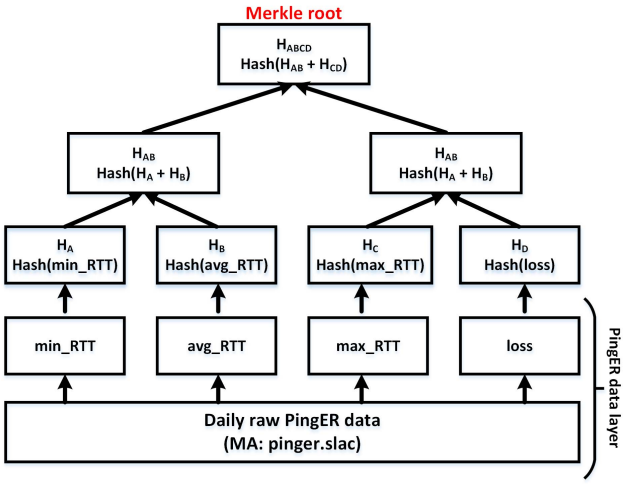

Fig. 5. PingER data Merkle tree

The Merkle root is stored as a part of the metadata in the blockchain along with the K-storage locations of the files. Later, each MA can carry out audits of the files (stored at K-storage locations of the network) periodically by verifying the Merkle root to ensure the proof of storage of the data. Further, Merkle root will help to check the immutability status of the files without analyzing the data in the files. If a node is off-line or fails to provide the proof of storage, the replication process will copy the data from the other file location and placed it on a new MA to satisfy the proof of redundancy based on `(K-of-M)` erasure coding as shown in Figure 6.

On the network, the files are stored off-chain through a Distributed Hash-table (DHT) [25], [26]. Note that only the references to the data are stored in DHT (not the actual data itself) and are accessible through blockchain. For example, the Kademlia [27] is a popular DHT protocol which provides efficient lookups through massive networks with low coordination overhead in P2P networks.

B-2. PingER Blockchain: The components of PingER blockchain are explained as follows:

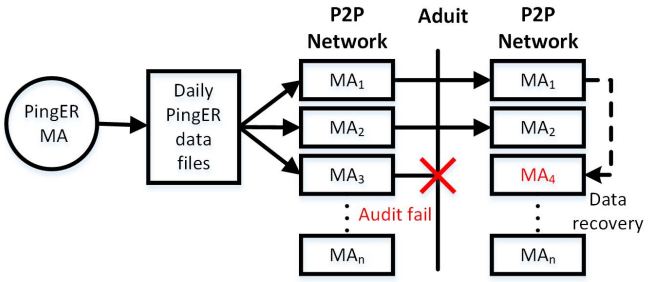

Fig. 6. PingER data audit and recovery

i) **Transaction:** The PingER blockchain transaction structure is illustrated in Figure 7. Each transaction consists of a file hash, file size, timestamp, Merkle root, upload locations of the file and an MA signature. Each MA signature is composed of an MA name and IP address. Each MA broadcasts the transaction on the network. All the MAs in the network verify the transaction through MA signature.

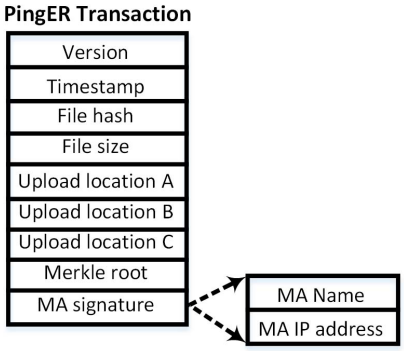

Fig. 7. PingER blockchain transaction

ii) **Block:** The verified transactions are ordered and batched into blocks. Note that, in the original blockchain, there are thousands of transactions in a single block. However, in the proposed framework, day-to-day metadata of each MA is stored in the blockchain (only once in a day). Therefore, the total number of the transaction in a single block becomes equal to total number of MAs available in the network. This will help to simplify the data access layer of the framework. Each block is composed of a block header and list of verified transactions of the day. The block header contains the timestamp, block index in the blockchain, Merkle root hash created using the hashes of all the transactions in a block and hash of the previous block. Finally, the hash of the preceding block turns this sequence of blocks into an immutable data structure. A typical PingER block is shown in Figure 8.

iii) **Consensus:** Consensus algorithm assists all MAs to share the exact same copy of the data on the network. A good choice for PingER permissioned blockchain will

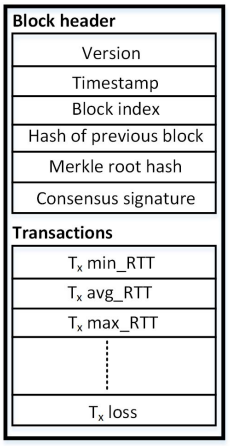

Fig. 8. PingER block & transactions

be the Simplified Byzantine Fault Tolerance (SBFT) consensus algorithm. SBFT is an implemented version of the Practical Byzantine Fault Tolerant (PBFT) algorithm [28]. The algorithm is based on a single block generator which has a privileged role in the system. The block generator collects all transactions from the MAs on the network. The block generator orders and groups all verified transactions periodically i.e., after every 24 hours in case of PingER. Afterwards, the generator broadcast the new block to all MAs on the network. Each MA acts a block signer and only signs the block which is already signed by the generator. Consensus about the new block is achieved as a result of a minimum number of other nodes in the network ratifying the new block. In a system, Byzantine fault tolerance can be achieved, if 2f+1 nodes out of 3f+1 nodes (where f is the number of faults in the system) reach a consensus.

C. Data Access Layer

In this layer, existing PingER scripts which generate hourly, monthly, and yearly reports can work seamlessly. This is because, in this framework, contents are addressed through hashes which is a widely used means of connecting data in a distributed network (e.g., data accesses in git work on the same principle). For example, PingER scripts can run over the Inter Planetary File System (IPFS) [18], [29] which is a peer-to-peer hypermedia protocol using distributed hash tables and a self-certifying namespace. An important feature of IPFS is that the objects can be traversed with a string path. Paths work as they do in traditional UNIX filesystems and the Web. For example, `/ipfs/BLYkgq61ZYaQ8NhkcqyU7rLcnSa7dSHQ16x/min_rtt-100-pinger.slac.edu-2018-01-28.txt.gz`. Thus, all data files are always accessible via their hash. This approach still works even if a few MAs are offline as the files are located in multiple locations for redundancy. Thus, IPFS provides smarter, faster and permanent web services to the PingER data access stack as shown in Figure 9.

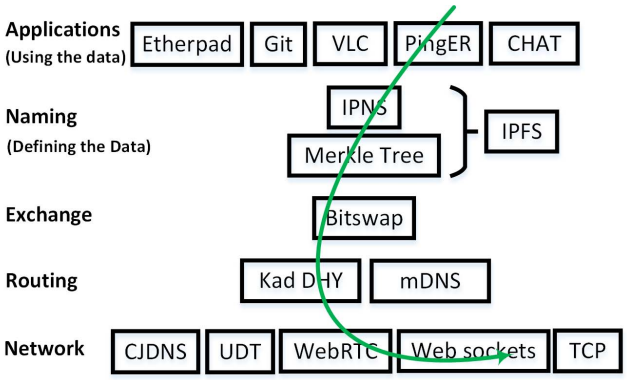

Fig. 9. PingER data access stack

IV. Implementation Requirements

The development of the proposed framework can be carried out in the Hyperledger Project. It includes different open source blockchain frameworks and related technologies hosted by the Linux Foundation in collaboration with industry leaders in technology, finance, banking, supply chain management, telecommunication providers, consumer electronics manufacturers, and more [30]. Specifically, Hyperledger Fabric provides a non-cryptocurrency based distributed ledger technologies with auditable & immutable ledgers, modular approach, support for smart contacts and choice of multiple consensus algorithms depending on the requirements of the system. Moreover, peers (MAs in case of PingER) in Hyperledger Fabric can be added dynamically and programmatically, rather than statically. The technical requirements of Hyperledger Fabric are cURL[8], Node.js[9], npm package manager[10], Go Language[11], Docker[12], and Docker Compose[13]. All these tools will facilitate the development of PingER-blockchain framework in the Hyperledger Fabric.

V. Significance

In addition to decentralized data storage, the framework could possibly have the following capabilities.

i) There is no single point failure problem since no single machine holds all the data record in the network.
ii) As a P2P network, all the PingER monitoring sites would have equal status with write access to the blockchain; the tiered structure that is currently used and dependent upon the uptime and connectivity would be replaced.
iii) The relationship between all MAs become symmetrical as all of them are running same tasks, installed with similar software and executing similar privileges.

[8]https://curl.haxx.se/
[9]https://nodejs.org/en/
[10]https://www.npmjs.com/
[11]https://golang.org
[12]https://www.docker.com/
[13]https://docs.docker.com/compose/

iv) One advantage might be the near real-time data collection and analysis across the blockchain members i.e., MAs. In the current architecture, SLAC fetches data overnight. Thus, the most updated date goes back to 24 hours of time which make it impossible for PingER to capture events in real time.

v) PingER web application becomes fully distributed and even if a few MAs are offline, data can be accessed, and reports can be generated.

VI. Conclusion

In this paper, we design a decentralized data storage and access framework for PingER using permissioned blockchain technology. The proposed framework eliminates the need for the centralized repository as the upward paths from the monitoring agents are replaced by write-access data entries on the permissioned blockchain. This approach decentralizes the PingER framework and removes the project dependence upon centralized computing resources for storing, processing and uptime. The resulting architecture will help in scaling up sustainable and large-scale implementation of the project. This, in turn, will help in improving the performance monitoring of the Internet needed to maintain the quality-of-service required for present day and future technologies of the Internet.

Acknowledgment

This material is partially based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under Contract DE-AC02-76SF00515. Further, this work is supported in part by the National Natural Science Foundation of China under Grants 61632009 & 61472451, in part by the CERNET Innovation Project NGII20170102, in part by the Guangdong Provincial Natural Science Foundation under Grant 2017A030308006 and High-Level Talents Program of Higher Education in Guangdong Province under Grant 2016ZJ01.

References

[1] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, thirdquarter 2016.

[2] M. D. Pierro, "What is the blockchain?" *Computing in Science Engineering*, vol. 19, no. 5, pp. 92–95, 2017.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.

[5] M. E. Peck, "Blockchain world - do you need a blockchain? this chart will tell you if the technology can solve your problem," *IEEE Spectrum*, vol. 54, no. 10, pp. 38–60, October 2017.

[6] D. T. T. Anh, M. Zhang, B. C. Ooi, and G. Chen, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1–1, 2018.

[7] T. Aste, P. Tasca, and T. D. Matteo, "Blockchain technologies: The foreseeable impact on society and industry," *Computer*, vol. 50, no. 9, pp. 18–28, 2017.

[8] S. Underwood, "Blockchain beyond bitcoin," *Commun. ACM*, vol. 59, no. 11, pp. 15–17, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2994581

[9] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?a systematic review," *PLOS ONE*, vol. 11, no. 10, pp. 1–27, 10 2016.

[10] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, Jun 2017. [Online]. Available: https://doi.org/10.1007/s12599-017-0467-3

[11] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *Journal of the American Medical Informatics Association*, vol. 24, no. 6, pp. 1211–1220, 2017. [Online]. Available: http://dx.doi.org/10.1093/jamia/ocx068

[12] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of iot data," in *Proceedings of the 2017 on Cloud Computing Security Workshop*, ser. CCSW '17. New York, NY, USA: ACM, 2017, pp. 45–50.

[13] W. Matthews and L. Cottrell, "The pinger project: active internet performance monitoring for the henp community," *IEEE Communications Magazine*, vol. 38, no. 5, pp. 130–136, May 2000.

[14] S. Ali, G. Wang, and R. L. Cottrell, "Internet Performance Analysis of South Asian Countries using End-to-end Internet Performance Measurements," in *16th IEEE International Conference on Ubiquitous Computing and Communications (IUCC 2017)*, Guangzhou, China, 2017, pp. 1–8.

[15] S. Ines, J. Ubacht, and M. Janssen, "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," *Government Information Quarterly*, vol. 34, no. 3, pp. 355 – 364, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0740624X17303155

[16] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[17] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997. [Online]. Available: http://ojphi.org/ojs/index.php/fm/article/view/548

[18] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[19] V. Gramoli, "From blockchain consensus back to byzantine consensus," *Future Generation Computer Systems*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17320095

[20] D. Kraft, "Difficulty control for blockchain-based consensus systems," *Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 397–413, Mar 2016.

[21] T. Barbosa, R. Souza, S. Cruz, M. Campos, and R. Les Cottrell, "Applying data warehousing and big data techniques to analyze internet performance," in *2015 NETAPPS 4th Int. Conf. on Internet Applications, Protocols and Services*, 2015, pp. 31–36.

[22] S. Wilkinson, J. Lowry, and T. Boshevski, "Metadisk a blockchain-based decentralized file storage application," Tech. Rep., 2014.

[23] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *CoRR*, vol. abs/1506.03471, 2015. [Online]. Available: http://arxiv.org/abs/1506.03471

[24] R. C. Merkle, "Protocols for public key cryptosystems," in *1980 IEEE Symposium on Security and Privacy*, April 1980, pp. 122–122.

[25] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 161–172, Aug. 2001.

[26] A. Roehrs, C. A. da Costa, and R. da Rosa Righi, "Omniphr: A distributed architecture model to integrate personal health records," *Journal of Biomedical Informatics*, vol. 71, pp. 70 – 81, 2017.

[27] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 53–65.

[28] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002. [Online]. Available: http://doi.acm.org/10.1145/571637.571640

[29] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[30] M. Vukolić, "Rethinking permissioned blockchains," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, ser. BCC '17, 2017, pp. 3–7. [Online]. Available: http://doi.acm.org/10.1145/3055518.3055526