Integration of Blockchains with Management Information Systems

Ka Ching Chan*, Xujuan Zhou*, Raj Gururajan*, Xiong Zhou†, Mustafa Ally*, Michael Gardiner*
*School of Management and Enterprise
University of Southern Queensland, Springfield, Australia
Email: kc.chan@usq.edu.au, xujuan.zhou@usq.edu.au, raj.gururajan@usq.edu.au
mustafa.ally@usq.edu.au, michael.gardiner@usq.edu.au
†Fujian Vocational College of Agriculture
Fujian, P R China
Email: 328475505@qq.com

***Abstract*—In the era of the fourth industrial revolution (Industry 4.0), many Management Information Systems (MIS) integrate real-time data collection and use technologies such as big data, machine learning, and cloud computing, to foster a wide range of creative innovations, business improvements, and new business models and processes. However, the integration of blockchain with MIS offers the blockchain trilemma of security, decentralisation and scalability. MIS are usually Web 2.0 client-server applications that include the front end web systems and back end databases; while blockchain systems are Web 3.0 decentralised applications. MIS are usually private systems that a single party controls and manages; while blockchain systems are usually public, and any party can join and participate. This paper clarifies the key concepts and illustrates with figures, the implementation of public, private and consortium blockchains on the Ethereum platform. Ultimately, the paper presents a framework for building a private blockchain system on the public Ethereum blockchain. Then, integrating the Web 2.0 client-server applications that are commonly used in MIS with Web 3.0 decentralised blockchain applications.**

***Index Terms*—blockchain, distributed ledger technology, private blockchain, consortium blockchain, permissioned blockchain, smart contract, Ethereum**

I. Introduction

In the era of the fourth industrial revolution, IoT [1], RFID, and QR Codes are inputs or sensors for real-time data collection. The integration of these input technologies with traditional client-server Web 2.0 systems, and key technologies such as big data, machine learning, and cloud computing fosters a wide range of creative innovations, business improvements, and new business models and processes.

In this environment, blockchain offers the trilemma of security, decentralisation and scalability, where balancing the three alternatives depends on the needs for the blockchain.

The focus of this paper is on the integration of traditional Web 2.0 client-server systems with Web 3.0 blockchain technology and the need for a single source of immutable truth, all of which are ideal characteristics for many applications.

Blockchain technology allows digital information to be distributed, but not copied, meaning that a blockchain is a linking of a series of time-stamped immutable records of data that is not owned by a single entity. This makes the data stored on the block secure and yet transparent. This technology addresses the issues of trust as the immutable ledger is shared and replicated across all nodes.

Benefits of this technology include the development and use of smart contracts, and yet remains secure to participants who hold accounts to access the blockchain in real-time.

While the concept of integrating blockchain and MIS sound alluring, there are three significant obstacles to the integration of blockchain with MIS. The first is the many differing blockchain frameworks. Developing a suitable framework for a given application can draw upon the work of [2], which explored many blockchain applications to develop recognisable blockchain frameworks. The second obstacle, even with the blockchain framework, is the many configurations and variations, which makes the process of designing and implementing a blockchain for a given situation complicated [3]. The third obstacle is the multiple IT systems speaking different languages, often leading to a painful reconciliation process and with a high risk of error [4], a problem that can be solved using blockchain with the MIS existing software.

Thus, this paper explores and explains how Web 2.0 and Web 3.0 systems can be interfaced while maintaining the benefits of both systems and addressing the weaknesses of each system.

II. Types of Blockchains

In this section, we will first provide an overview and discuss the differences between the three types of blockchains - public, private, and consortium. In the next section, we will then describe how client-server management information systems can be integrated with Ethereum based blockchains.

A. Public, Private and Consortium Blockchains

Systems can be categorised into centralised, decentralised, and distributed systems. In a centralised system, there is a single point of authority controlling and commanding the operations of all systems. The client-server architecture is a typical centralised system architecture. Most of the current web applications and TCP/IP applications fall into this category. In a decentralised system, there is no single system

that is in control and command of any other systems. All systems are run independently and in parallel. And there is no single point of failure. The peer-to-peer and master-slave architectures are typical decentralised system architectures. Typical decentralised applications include blockchains and cryptocurrencies. In a distributed system, the computation is shared among a number of servers. Typical distributed architectures include peer-to-peer, client-server, and n-tier architecture. Typical applications include cluster computing and grid computing [5], [6].

Not all systems fall into one of the above three categories. Some systems are centralised and distributed; some other systems are decentralised and distributed. Many existing online services are run by centralised and distributed systems; while cryptocurrencies such as Bitcoin and Ethereum, are decentralised and distributed systems. Ethereum was designed as a public blockchain running in a fully decentralised network that is not controlled by any single entity and is exceptionally secured by using cryptographic and consensus algorithms such as proof-of-work and proof-of-stake. Private blockchains and consortium blockchains can be implemented on Ethereum with configuration and/or access control.

1) Public Blockchains: Ethereum was initially designed as a public blockchain [7], [8]. Any individual and organisation can participate, and any computer can join the Ethereum network and become a node without permission required by any other node. Therefore, the public Ethereum is a permissionless blockchain where all nodes are considered equal, and no node is in control of any other node. These nodes form a peer-to-peer network [9]. As a decentralised system, every node contains a copy of the blockchain that is synchronised with the entire transaction history. Due to a large number of nodes, the redundancy of Ethereum is very high, and any node can be taken out without affecting the continuing operation of the chain.

Blockchain enables trustless transactions without intermediaries such as clearing house, agents or central organisations. The blockchain network allows users to remain anonymous and perform transactions on a peer-to-peer basis [9], [10]. As a public ledger, the transaction data recorded on the Ethereum main chain is transparent to the public; and is immutable and cryptographically secured. All records go through a mining process for verification before being added to the chain permanently. The computational cost involved in the mining process is very high, and the processing speed very slow. The mining process that runs a consensus algorithm to approve transactions is considered wasteful of energy and computing resources. Currently, the Ethereum main chain can only process roughly 15 transactions per second [11]. Due to this waste, the scalability is still the most challenging problem among the blockchain trilemma of trade-off among security, decentralisation, and scalability.

There are three main types of blockchain transactions - send, approve, and read. The access privileges of these transactions are critical differentiators of public, private, and consortium blockchains [12], [13]. Although there are many different

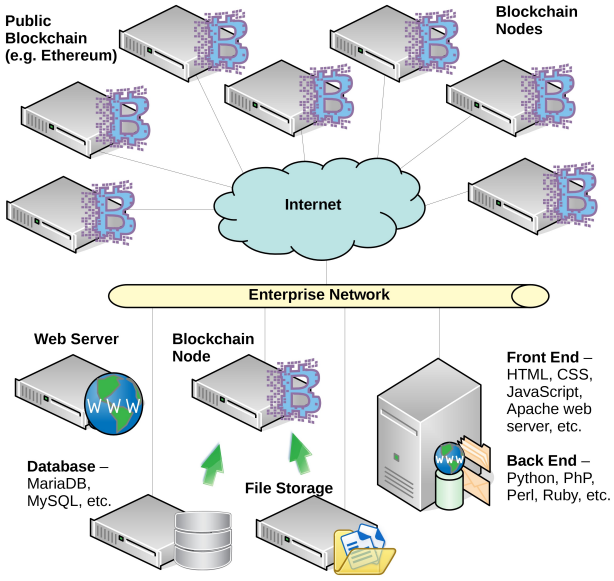

Fig. 1. Public blockchain.

ways to build blockchain applications, these access privileges fundamentally define whether the blockchain is public, private, or consortium. In a public blockchain, any node can send transactions to and read from the blockchain. Further, any node can participate in the consensus process to approve transactions. Table I compares the three types of Ethereum based blockchains.

2) Private Blockchains: The ideology of blockchain, being a decentralised and permissionless system that supports trustless transactions and total transparency, is core to cryptocurrencies. But this ideology may have to be compromised with the reality of practical needs in industry. Business enterprises would not be interested in total transparency, i.e. to share the ledger records with other businesses; but want to keep their data and information private and confidential unless disclosure is needed to meet regulatory obligations or other specific purposes. Business enterprises would not want other businesses or competitors to participate in their blockchain transactions, but want to be the only party to have full control of the blockchain. Therefore, private blockchains with modified characteristics may be required to meet the needs of various enterprise applications. These private blockchains can be integrated with the management information systems for storage of secure and immutable vital records, to enable secure and trustless transactions among internal departments, and to improve or lower the cost of a particular business process. The private blockchains can also play an essential role in being the single source of truth for key data.

The design and implementation of private blockchains can take many forms, as a trade-off of a number of factors such as cost, speed, scalability, decentralisation, transparency, choice of software, ease of implementation, and so on. One approach is to set up a blockchain network entirely within the boundary of the enterprise's network, as shown in Fig. 2.

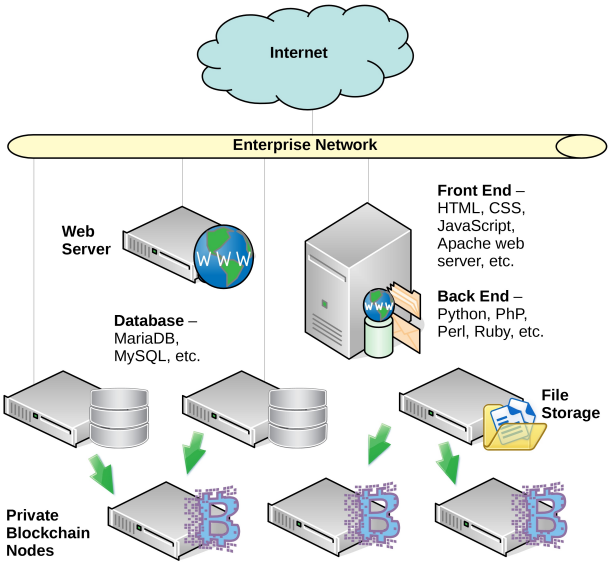

Fig. 2. Private blockchain network within an enterprise network.

All nodes are physical computers or virtual machines that are under the control and administration of the enterprise, and the ledger data is stored in a distributed database across a number of nodes in different locations. The Open Linux Foundation project Hyperledger would be one solution for such an approach [14].

The second approach is to set up a private blockchain network on a public platform like Ethereum [15]–[17]. This setup is similar to that shown in Fig. 1 with one or more blockchain nodes set up within the enterprise network. The process of setting up a private Ethereum chain is similar to setting up a test Ethereum chain (testnet) that is separate from the main chain. User accounts and access control are put in place during the setup and configuration stage to control who is allowed access to the private chain.

A private Ethereum chain has its own genesis block. To process transactions on the private chain, a low mining difficulty can be set in the genesis block to enable fast mining of fake Ether. Fake Ether does not have any value and cannot be used on the main chain or traded on cryptocurrency exchanges, but the processing performance can be significantly improved. The proof of work (PoW) consensus algorithm used in Ethereum is highly wasteful of energy and computational power. In a very recent article, Microsoft suggested using proof of authority (PoA) as a more efficient consensus algorithm for building private Ethereum chains [18].

In an Ethereum based private blockchain, the privileges to send and approve transactions are centralised to a single organisation; while the permission to read is limited to selected nodes internal or external to the organisation [13]. How to integrate private blockchains on Ethereum with management information systems is the focus of this expository paper.

3) Consortium Blockchains: Consortium blockchains are controlled and operated by a group of organisations. Provenance of supply chains is a typical application of consortium blockchain. The members of the consortium chain are suppliers and buyers along the supply chain, and they share certain information and store the transaction records on the chain. The information allows users to check the provenance of the products that they purchase. The immutability and single source of truth that a consortium blockchain can offer are fundamental to any supply chain provenance.

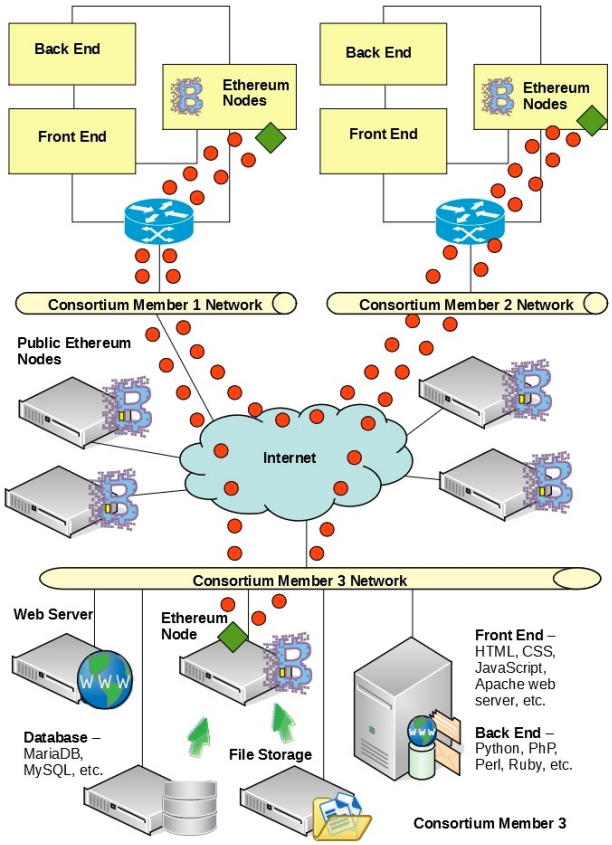

Fig. 3. Consortium blockchain.

Each consortium member runs and controls its network of Ethereum nodes somewhere in the world. The challenge of building consortium blockchains is to connect these networks of Ethereum nodes anywhere in the world via the public Internet, and at the same time to provide mechanisms to ensure the permission rights of the consortium participants. The chain database is decentralised and distributed among all participating parties. All participating organisations are involved in the consensus and decision-making process. The governance and control of the chain must be agreed and rules defined.

The Ethereum network setup for each consortium organisation is similar to that for a private network (Fig. 2). Each consortium blockchain requires individual Ethereum networks to be connected peer-to-peer, where permission rights are configured by software. Figure 3 shows a consortium blockchain network with three parties. Each party runs its own Ethereum nodes which can be physical servers or virtual machines. These

individual Ethereum networks (marked with a green diamond in Fig. 3), communicate with one another via peer-to-peer connections (the orange dotted paths in Fig. 3) through the Internet.

In a consortium chain, the send, approve and read rights are limited to a certain number of parties within the consortium network [12]. A successful example of consortium blockchain was implemented by Webjet to improve the settlement process between hotel suppliers and travel partners [4]. The blockchain enables any two parties to verify that booking data matches. Microsoft also offers a set of solution templates (Ethereum consortium blockchain in Azure Marketplace) that allow users to configure multi-region and multi-member Ethereum consortium blockchain networks with a simple multi-step process through the Azure portal or command line. In this approach, all nodes are within the same virtual network, or each member's nodes are put into individual virtual networks communicating through application gateways [19]. For consortium blockchains, proof of authority (PoA) can be used as a more efficient consensus algorithm than the proof-of-work (PoW) algorithm [18]. Another Open Source project Quorum adds transaction privacy on top of the Ethereum transactions to build consortium chains for enterprises [20].

III. Integration of MIS and Ethereum Blockchain

Blockchain is a Web 3.0 technology, and the integration of Web 2.0 business applications with Web 3.0 blockchains is a pragmatic approach during this period of maturing of Web 3.0. There are many practical applications that can benefit from such an approach. For example, in elections, votes are collected through web applications and immediately added to a blockchain to ensure data is immutable once submitted.

There are many approaches using different technologies and platforms to build a blockchain application. This section aims to describe a generic approach that uses the most popular software that are freely available to integrate the traditional management information systems running on client-server network architecture (Web 2.0) with private decentralised Ethereum blockchains (Web 3.0). Although they are in two different paradigms, the integration that mainly involves data transfer between database and blockchain, is straightforward. In the proposed integration approach, the operations of the two remain highly independent and separate with limited interaction. In fact, the blockchain component can be simply integrated as an add-on to any existing management information systems or business web systems.

Most business web applications are centralised applications that consist of front end and back end systems. The functionalities and user interfaces are provided by the front end applications while data is stored in a central database server. The programming languages commonly used include HTML, CSS, JS for front end development and Python, Perl, PhP, and Ruby for back end development.

As an example, a developer may design a front end web page using HTML with PhP code embedded in the HTML page to access a MySQL database. When a visitor opens the page, the server processes the PhP code retrieving the relevant data from the database table, generates a dynamic page, and sends it back to the visitor's web browser via the Internet or any TCP/IP network for display.

TABLE I
Comparison of Public, Private and Consortium Ethereum Blockchains

	Management Information Systems (Web 2.0)		
Network Architecture	Client-server		
System Architecture	Centralised or distributed		
Front End	HTML, CSS, JS, and Apache web server		
Back End	Python, PhP, Perl, Ruby		
Database	MariaDB, MySQL		
Ethereum API	web3.js, web3.py, web3.php		
	Blockchains (Web 3.0)		
	Public	***Private***	***Consortium***
Network Architecture	Peer-to-peer		
System Architecture	Decentralised and distributed		
Ethereum Client	Go Ethereum (geth)		
Control of Chain	No one	Centrally controlled	All participating parties
Node	Any computer connected to the Internet	Computers of a single organisation	Computers of a selected numbers of organisations
Permission	Permissionless	Permissioned	Permissioned
Send Permission	Any node	Centralised	Limited
Approve Permission	Any node	Centralised	Limited
Read Permission	Any node	Limited	Limited
Blockchain Platform	Ethereum (main chain)	Ethereum (testnet)	Ethereum (Vnet) or Ethereum with Access Control
Consensus Algorithm	Proof of Work	Proof of Work or Proof of Authority	Proof of Work or Proof of Authority
Gas Cost [21]	Ether	Fake Ether (Free)	Fake Ether (Free)
Performance	Processing $\approx$15 transactions/sec [11]	Processing instantly	Processing instantly

There are two main steps to integrate such a web application with an Ethereum blockchain:

1) Setting up nodes of the private or consortium blockchain on the Ethereum platform.
2) Programming the web application - using the Web3.js API and Ethereum client Geth to read/write data from/to the Ethereum blockchain.

These two steps will be explained in more details in the following sections.

Fig. 4 shows the integration framework with the web applications at the top and the blockchain at the bottom. It should be noted that this framework applies to all types of Ethereum blockchains - public, private and consortium. For

example, if the blockchain is a consortium blockchain to support supply chain provenance, each of the companies along the supply chain can have its own web servers and databases, and connects to the same blockchain via the web3.js API and Geth client.

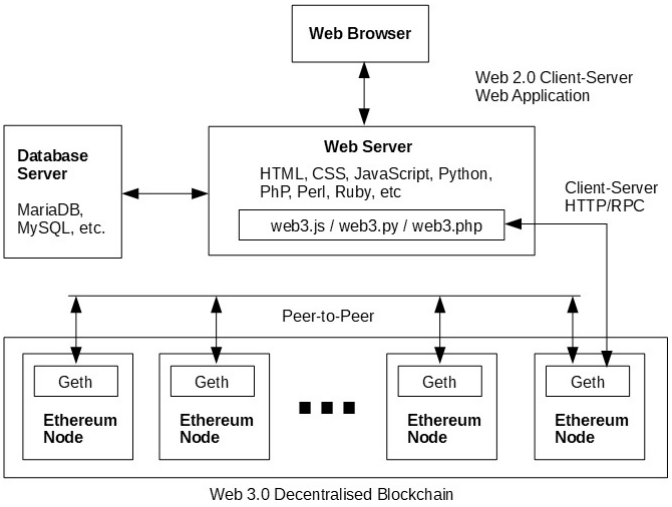

Fig. 4. Integration with blockchain.

A. Setting up Ethereum Nodes Using Geth

To set up an Ethereum node, access the chain, and participate in chain transactions, an Ethereum client is needed. Go Ethereum (Geth) is one of the original Ethereum clients that are available for download and can be installed on most operating systems [22]. As shown in Fig. 4, Geth is required on each Ethereum node connected to the blockchain.

Once installed, Geth provides a Command Line Interface (CLI) for running an Ethereum node. Geth can be used to perform all the main tasks, such as:

- creating Ethereum nodes (private or public),
- creating Ethereum accounts to store Ether,
- mining, i.e. creating blocks on the blockchain and miner receiving Ether as a reward,
- performing transactions, i.e. transferring Ether between accounts,
- creating and executing smart contracts, and
- providing a HTTP-RPC server to communicate with web3.js clients.

If the application involves smart contracts, then the Remix Integrated Development Environment (IDE) would be an easier option for development. Remix is a web based browser IDE that supports coding, testing, debugging, compiling and deploying of smart contracts [23], [24]. Smart contracts enable business transactions to take place in a trustable or trustless, secure, and automated manner. The Ethereum smart contracts that define programmatically rules of business contracts are coded in Solidity, a simple programming language with Javascript like syntax [25], [26]. Remix has a built-in Solidity compiler to generate machine-level bytecode that can be executed by an Ethereum node.

When preparing the blockchain for integration, multiple Ethereum nodes running Geth should be deployed, enabling the decentralisation of the chain, and distribution of data and processing resources across every device connected to the blockchain.

B. Programming Web Application Using Web3.js

When the private blockchain is ready, the next step is to program the web application to communicate and interact with an Ethereum node using a HTTP or Inter-Process Communication (IPC) connection such as Named Pipes or Remote Procedure Calls (RPC). There are multiple Ethereum Application Programming Interfaces (API) available to support different programming languages, such as web3.js for JavaScript [27], web3.py for Python [28], and web3.php for PhP.

The library web3.js, which was the first API developed, defines the rules and formats of how blockchain data should be exchanged between web3.js and Geth. The web3.js API provides a convenient RPC interface to allow communicating with an Ethereum node running Geth from inside a JavaScript application. The web3.js RPC or HTTP request causes a procedure to execute in the remote Ethereum node.

The JavaScript application does not interact with the blockchain directly. It only interacts with one Ethereum node and that node, in turn, interacts with the blockchain. Therefore, the connection between web3.js and Geth is a client-server connection where web3.js takes the client role and Geth the server role. At the blockchain level, all nodes are equal and communicate with the whole Ethereum chain on a peer-to-peer basis.

IV. Discussion and Conclusion

By integrating management information systems with private or consortium blockchains, business enterprises can store their key records in the blockchain that offers data immutability and serves as the single source of truth for the enterprises and/or across the supply chain.

In this paper, we have discussed the key concepts and illustrated with figures the implementation of private and consortium blockchains on the public Ethereum platform. The access rights (send, read, approve) are the key differentiators of blockchain type (public, private, consortium). The main issues of running enterprise applications on the global public Ethereum platform are slow processing performance (15 transactions/sec) and scalability. However, for private and consortium blockchains, mining with fake Ether significantly speeds up the process and performance is not an issue.

Then we have presented a framework for the integration of traditional Web 2.0 applications with Web 3.0 blockchains. We have also introduced the leading software libraries (Geth and web3.js) that are required to connect the web application to an Ethereum node. Although they are in two different paradigms, the integration that mainly involves data transfer and initiating remote commands are straightforward. We believe the

integration framework provides a clear blueprint for future development of innovative applications and improvements of both intra- and inter-company processes.

REFERENCES

[1] D. Miller, "Blockchain and the Internet of Things in the industrial sector," IT professional, vol. 20, no. 3, pp. 15–18, May/June 2018.

[2] B. A. Scriber, "A framework for determining blockchain applicability", IEEE Software, vol. 35, no. 4, pp. 70–77, July/August 2018.

[3] X. Xu., et al., "A taxonomy of blockchain-based systems for architecture design," Proc. IEEE Int. Conf. Software Architecture (ICSA), 2017.

[4] J. Guscic and L. Oldfield, "Webjet Limited - Introducing Rezchain and Rezpayments", Ord Minnett Investor Session, Melbourne, Australia, 18 June 2019. [Online]. Available: https://www.asx.com.au/asxpdf/20190618/pdf/445xj8c6v932qx.pdf. [Accessed: 12-Aug-2019].

[5] P. Hooda, "Comparison Centralized, Decentralized and Distributed Systems", 2018. [Online]. Available: https://www.geeksforgeeks.org/comparison-centralized-decentralized-and-distributed-systems/. [Accessed: 19-Sep-2019].

[6] A. Drury, "Centralized vs Decentralized: Whats the difference?", 2018. [Online]. Available: https://blocklr.com/guides/centralized-vs-decentralized/. [Accessed: 19-Sep-2019].

[7] Ethereum Foundation, "Ethereum blockchain app platform", 2018. [Online]. Available: https://www.ethereum.org/. [Accessed: 27-Feb-2019].

[8] A. M. Antonopoulos and G. Wood, Mastering Ethereum: Building Smart Contracts and DApps. Sebastopos, CA: O'Reilly Media, 2019.

[9] L. Hu, "Understanding Ethereums P2P Network", 2018 [Online]. Available: https://medium.com/shyft-network-media/understanding-ethereums-p2p-network-86eeaa3345. [Accessed: 28-Aug-2019].

[10] C. Grundy, "Explained: How do Ethereum transactions work?", 2018 [Online]. Available: https://thecoinoffering.com/learn/ethereum-transactions/. [Accessed: 14-May-2019].

[11] A. Hertig, "How Will Ethereum Scale?", 2018 [Online]. Available: https://www.coindesk.com/information/will-ethereum-scale. [Accessed: 28-Aug-2019].

[12] J. Poon, "Building a Private Ethereum Consortium", Microsoft Developer Blog, 1 June 2018. [Online]. Available: https://www.microsoft.com/developerblog/2018/06/01/creating-private-ethereum-consortium-kubernetes/. [Accessed: 14-Aug-2019].

[13] V. Buterin, "On Public and Private Blockchains", Ethereum Foundation Blog, 6 August 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/. [Accessed: 15-Aug-2019].

[14] The Linux Foundation, "Hyperledger", 2016. [Online]. Available: http://https://www.hyperledger.org/. [Accessed: 28-Aug-2019].

[15] Mercury Protocol, "How To: Create Your Own Private Ethereum Blockchain", 2017. [Online]. Available: https://medium.com/mercuryprotocol/how-to-create-your-own-private-ethereum-blockchain-dad6af82fc9f. [Accessed: 10-Sep-2019].

[16] O. S. Hiremath, "Ethereum Private Network Create your own Ethereum Blockchain!", 2019. [Online]. Available: https://www.edureka.co/blog/ethereum-private-network-tutorial. [Accessed: 10-Sep-2019].

[17] A. Toluhi, "Ethereum: Setting Up A Private Blockchain", 2018. [Online]. Available: https://medium.com/coinmonks/ethereum-setting-up-a-private-blockchain-67bbb96cf4f1. [Accessed: 10-Sep-2019].

[18] Microsoft TechNet, "Ethereum proof-of-authority consortium", 2019. [Online]. Available: https://docs.microsoft.com/en-us/azure/blockchain/templates/ethereum-poa-deployment. [Accessed: 10-Sep-2019].

[19] Microsoft TechNet, "Ethereum Multi-Member Consortium Blockchain in Azure Marketplace", 2017. [Online]. Available: https://gallery.technet.microsoft.com/Ethereum-Multi-Member-96bad3bd. [Accessed: 10-Sep-2019].

[20] Quorum, "Evolve with Quorum. The proven blockchain solution for business", 2019. [Online]. Available: https://www.goquorum.com/. [Accessed: 10-Sep-2019].

[21] Aziz, "Guide to Ethereum: What is gas, gas limit and gas price?", 2018 [Online]. Available: https://masterthecrypto.com/ethereum-what-is-gas-gas-limit-gas-price/. [Accessed: 15-May-2019].

[22] Ethereum Foundation, "Go Ethereum", 2013–2016. [Online]. Available: https://geth.ethereum.org/. [Accessed: 28-Feb-2019].

[23] Ethereum Foundation, "Remix", 2019. [Online]. Available: https://remix.ethereum.org/. [Accessed: 30-Aug-2019].

[24] Remix, "Remix, Ethereum-IDE", 2019. [Online]. Available: https://remix-ide.readthedocs.io/en/latest/settings.html. [Accessed: 30-Aug-2019].

[25] Ethereum Foundation, "Solidity", 2016–2019. [Online]. Available: https://solidity.readthedocs.io/en/v0.5.4/. [Accessed: 27-Feb-2019].

[26] R. Modi, Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain. Birmingham, UK: Packt Publishing, 2018.

[27] Ethereum Community, "web3.js - Ethereum JavaScript API", 2016. [Online]. Available: https://web3js.readthedocs.io/en/1.0/. [Accessed: 28-Feb-2019].

[28] P. Merriam and J. Carver, "Web3.py", 2018. [Online]. Available: https://web3py.readthedocs.io/en/stable/. [Accessed: 28-Feb-2019].