Algorithm based-platform for decentralized applications in blockchain

Hamssah Hussein Abdul-Sada, Furkan Rabee
Department of Computer Science, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq

Article Info

Article history:

Received Mar 5, 2023
Revised Jul 6, 2023
Accepted Jul 8, 2023

Keywords:

Blockchain
DApps
Goerli test network
Smart contract
SONR DApps

ABSTRACT

Over the past several years, blockchain technology has been applied in numerous fields. Blockchain (BC) is based on the idea of decentralized applications (DApps) and facilitates their use. This increases the transparency of the apps, distributed, and flexible. Within this context, this paper presents a procedure and algorithm to build a DApp on Ethereum blockchain based platforms. simulation of nature reserve (SONR) is a smart contract code that was built to simulate the work of a nature reserve to take advantages of the characteristics of the genetic algorithm, and then deploy this contract on one of the Ethereum test networks, the Goerli test network, to customize a new block with a unique address for this DApps on that network, the contract creation time is within 12 seconds, and the average transaction time is 8.1 seconds with a 2.5 gwei gas cost.

Corresponding Author:

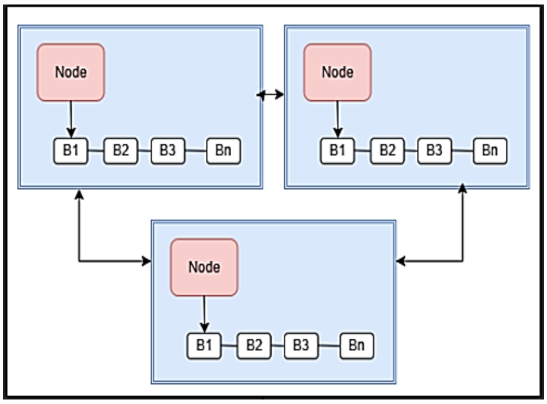
Hamssah Hussein Abdul-Sada
Department of Computer Science, Faculty of Computer Science and Mathematics, University of Kufa
Najaf, Iraq
Email: hamssah.alhassnawe@student.uokufa.edu.iq

1. INTRODUCTION

Researchers and industrial interest in blockchain has recently increased. Initial blockchain announcements were made in 2008 by Satoshi Nakamoto. Recently, blockchain has drawn more attention from academics and business. Initial blockchain announcements were made in 2008 by Satoshi Nakamoto [1]. The first industry to use blockchain was lash currency. Blockchain applications are currently being used in various industries with the development of smart contracts. The most widely used blockchain platform is now Ethereum. Smart contracts are used in conjunction with Ethereum decentralized applications (DApps). A different protocol for building decentralized apps is what Ethereum aims to build. The advantage of quick code development time with security measures is said to be offered. Decentralized apps and smart contracts are increasingly on the agenda as more businesses and developers attempt to find novel business models by utilizing the blockchain infrastructure [2]. With the help of DApps, a lot of data was produced. The development of applications based on the Blockchain platform is not without its challenges. On the internet, a variety of development environment solutions are given individually.

As shown in Figure 1, Numerous nodes running Ethereum clients make up the Ethereum network. Each of these nodes has a copy of the blockchain, which has a complete record of all network transactions. Nodes are able to stop fraudulent activities like bitcoin cloning and fraud as a result, and they can also maintain an auditable record of all network transactions.

Due to the decentralized structure of the network and the fact that each user's identity is determined by a public key, the Ethereum architecture preserves user pseudonymity (a less secure type of anonymity). Those who use the platform may now complete jobs like money transfers, buying and selling, and many other things thanks to this. The platform's native coin is called Ether, or ETH. In terms of market capitalization, Ether is the only cryptocurrency after Bitcoin. Anyone may launch irreversible, centralized, and permanent applications on Ethereum for user interaction [1]. The accounts that make up the Ethereum state each have a 20-byte address and go through state changes. The concept of a global state involves mapping addresses to account statuses. Ethereum supports both externally owned (private key controlled) and contract accounts. The four components of an Ethereum account are nodes, ether balance, contract code hash, and storage root [3].



Ethereum networks are employed for data storage and money transfers.
The Ethereum network ecosystem is quite diverse.
Nodes, which can be one or more, create networks.
Each node is a computer that uses the Ethereum client.
Anybody can manage a node.
Each node may have a complete copy of the blockchain.
A database called "blockchain" has a record of each transaction that has ever occurred.

Figure 1. An Ethereum network

As seen in Figure 2, developers utilizing the web3.js application programming interface (API) have access to the Ethereum network via this platform. This enables web apps to send transactions to the Ethereum network and decode messages delivered by the network. Ethereum also provides a user interface for customers using browsers and add-ons that support blockchain technology, such as the Bing browser and Metamask for Google Chrome. Through these user interfaces, users may access their Ethereum wallets and use Ethereum to pay for things online. The range of applications created across several fields is displayed in Table 1. The most advanced applications were found to be those for gaming [4].

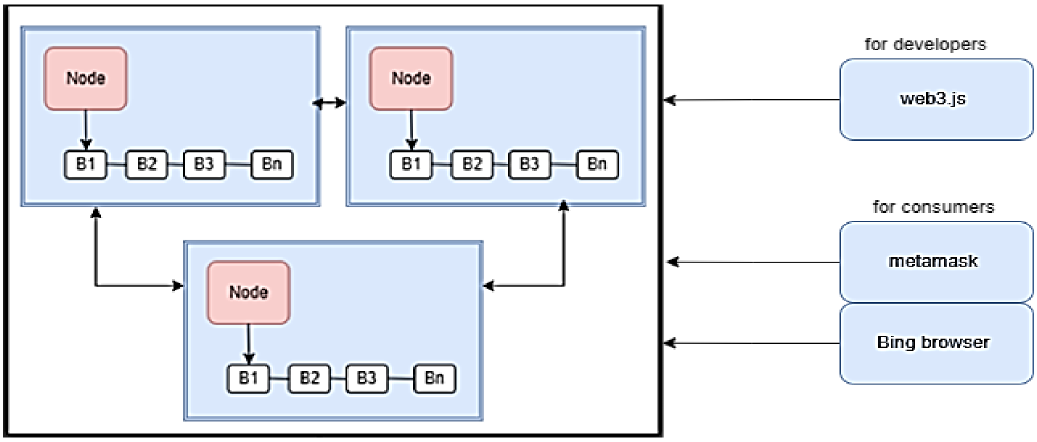


Figure 2. Interfacing with an Ethereum network

Table 1. DApps categories in Ethereum [5]

Type of Dapp	Number of Apps
Game	526
Casino	326
Exchange	95
High Risk	279
Finance	22
Other	325
Social	40

Digital apps or programs known as DApps run on a peer-to-peer (P2P) network of computers rather than a single computer [4]. The architecture on which modern internet applications are constructed inherently contains single points of failure. By distributing essential infrastructure or data storage components among a number of peers or nodes, DApps attempt to address these issues. Therefore, factors like security, expense, and usability should be considered while developing a DApp [6]. Certain characteristics must be present in DApp apps, such as [4]:

- Improved performance (low lag, high throughput).
- Reasonably cheap transaction fees.
- Maintainability flexibility.

User data shouldn't be copied or stored by DApps. The lack of a single point of vulnerability is the cornerstone of blockchain security. Users should only be able to manage their separate, individual identities through the application. It reduces reliance on top government officials. The user interface of the program should be simple to use and not unduly complicated. Avoid adding more code since it increases expenses and might introduce security problems.

A DApp is a web application that is "blockchain enabled" and operates on a peer-to-peer network of computers as opposed to a single server. Each node may utilize it independently, Furthermore, it has a front end and a back end. A "blockchain enabled" DApp in our context is a web application that is "blockchain enabled" and "runs on a peer-to-peer network of computers rather than a single server." Each node may utilize it independently, and it has a front end and a back end. The front-end interface is frequently built using the same technology as other apps. The program's use of a smart contract to connect to a blockchain network is the sole significant distinction. A smart contract, which executes itself, contains the terms of forging the contract between the buyer and seller in direct conformity with computer codes. Smart contracts allow for trustworthy transactions and agreements to be made between several anonymous parties without the need for a central authority, a legal system, or an external implementation mechanism [6].

The consensus algorithm is a type of algorithm that validates data. The algorithm has generated a hash value for each block that has been produced on the blockchain. This hash value may be produced using the SHA-256 hash algorithm, which alters inputs, reference hashes, and random integers to produce a hash value with a particular pattern [7]. Ethereum, however, often uses Keccak-256 as its hashing algorithm. This hash is applied to Ethereum addresses that were obtained by public keys or contracts. The last 20 bytes of the public key's Keccak-256 hash are used to represent the hexadecimal numbers that make up Ethereum addresses. In contrast to Bitcoin addresses, which are input into the client's user interface or presented to have a built-in checksum to prevent mistakes, Ethereum addresses are written in raw hexadecimal without any checksums. may give the three following factors to help DApps choose consensus models that meet their needs [8]:

- Deciding whether or not participants may remain anonymous.
- Classifying those with the authority to verify and sign transactions.
- Decentralizing administration of present business operations.

The proof-of-authority (PoA) consensus: the private Ethereum network's standard PoA also, this consensus performs well [9]. The PoA consensus uses a unique mechanism since anybody may join the blockchain network as a node. It varies from the type of blockchain that uses node-level privileges as a consequence. One of the test networks that use the PoA consensus is Goerli, which might be used to build a system. A PoA node that has shown its legitimacy to operate is granted the authority to produce new blocks. These nodes, called "Validators," are run by software that enables them to add transactions to blocks. Validators must maintain their computers in excellent working condition even if the procedure is automated, so they are not need to keep an eye on them constantly. PoA is appropriate for both public and private networks. The operation of this protocol may be summed up as follows: The validating devices generate the new blocks of transactions that need to be added to the network. Any blockchain network's configuration affects how new blocks are accepted and mined in the network [10]. Using PoA has a number of benefits. It can be utilized more sustainably since it does not require huge calculations like PoW, has a reasonably high risk tolerance, and has predictable block production times. The validator can be seen, which makes it possible for outside parties to manipulate the system. This is one disadvantage of PoA.

Many researchers work on the Ethereum network to create decentralized applications, including:

- Bogner *et al*. [11] shows off a DApps for sharing common items that is built on an Blockchain smart contract on Ethereum. Through the use of this Agreement, Users may register and rent Devices without the aid of a trusted third party (TTP), the disclosure of any personal information, or a prior membership to the service.
- Bang and Choi [7] suggests a monitoring system that gathers and examines data from a blockchain network in order to identify and track illicit activities. Also, among the Blockchain monitoring systems, was offered a productive storage system constructed by Apache Kafka and Apache Storm.

- Blum *et al*. [12] suggests an organized method utilizing already-existing architectural ideas employs the Meta-Transaction design pattern and other things like strategies, tactics, and design patterns to demonstrate how these may be used. Meta-transactions are function calls that are cryptographically signed and sent by a user to a backend. The user's costs are paid by the backend, which also uploads the transaction to the blockchain.
- Shabi and Al-Qarafi [13] and others investigate the specific idea of the face of significant security issues, a decentralized mechanism for managing data —the blockchain—was chosen. It also talks about current methods that may be utilized to defend against those attacks. By summarizing the salient features of these solutions, the strategies for enhancing blockchain security are discussed in this study. A multitude of blockchain systems and security tools that register defenselessness may be built using these crucial qualities. Finally, this study covers the open questions and future research directions for blockchain-based internet of things (IoT) systems.
- Min and Cai [14] makes an effort to identify DApp users using publicly accessible data. Creating several datasets with by 230,000 addresses, more than 73.8 million transactions were generated using Ethereum as an example. By converting hexadecimal addresses into accessible program names, researchers were able to assess user behavior in light of several DApp categories.
- Taş and Tanrıöver [3] discusses our experience creating a DApp using Ethereum, one of the most well-liked blockchain platforms, with an emphasis on architectural design for novices. He also suggests a method for determining which components of the local application architecture might profit from using Ethereum.
- Abdelhamid *et al*. [15] scholarly publications examining software engineering issues with blockchain technology for government applications have been systematically reviewed for this study. To demonstrate and pinpoint the problems usually raised in scholarly literature, the articles are first inductively analyzed. Additionally, a theoretical framework is explored using models from conventional software development, which is then followed by deductive analysis to outline the blockchain's use cases and potential future developments related to the challenges.
- Akazue *et al*. [16] by first guaranteeing that SmartRice, a sensor-based blockchain architecture, encourages decentralization providing decision-support for the value chain process of the food supply. Accurate information about harvested goods is captured and transmitted through SmartRice. In order to eliminate the many types of fraud that are prevalent in the current centralized system, the research recommends a decentralized architecture, limit corruption due to its sensor-based layered architecture, and reduce reported data mistakes along the value chain.
- Yue *et al*. [17] propose Blockchain-based DApps offer a lack of confidence in authority while addressing the crucial issues of security and privacy. Exploring frameworks and paradigms for decentralizing applications is a major focus of blockchain research, encouraging a variety of novel solutions spanning from network topologies to business models.
- Midaoui *et al*. [18] provide a strategy that combines a configurable IoT-based system with a distributed ledger (blockchain) to account for all necessary data, including the unique case of urban areas, and puts an open data platform at the disposal of investors, authorities, and consumers to efficiently evaluate, guarantee, and uphold traceability and transparency.
- Kumar *et al*. [19] provide the option of creating a decentralized blockchain for IoT data exchange that is trustworthy. People may connect with one other without a trusted mediator or middleman thanks to the blockchain's core criteria for verifying transactions. With the help of blockchain, we can create a distributed digital ledger. Blockchain technology ensures data immutability, location and shipment tracking, humidity, and open access to temperature records using ZigBee internet of things (IoT) sensor technology.
- Pujari *et al*. [20] suggested solution uses blockchain technology, smart contracts, and interplanetary communication to provide a safe decentralized consensus application in an untrusted environment. File system (IPFS) keywords.

2. METHOD

This study builds a decentralized application that mimics a nature reserve making use of Ethereum's blockchain-based smart contracts. The absence of a middleman and primary points of failure provides an advantage over the traditional paradigm. The evolutionary technique was developed in this paper using Solidity Smart Contracts and Remix-IDE V0.29.0. Gas is a predetermined quantity of ether that is given to the node carrying out the transaction and is inversely proportional to the transaction's processing complexity. Gas provides a financial incentive to finish a deal. Due to this incentive model, system load and contract complexity have an impact on how quickly contracts develop. Similar to this, transaction time is influenced by both the amount of active transactions throughout the whole system and the complexity of the application logic. Figure 3 displays the recommended DAPP's architectural layout. We focus on the architectural design of blockchain-oriented apps in this study and offer a technique to simulate a nature reserve.

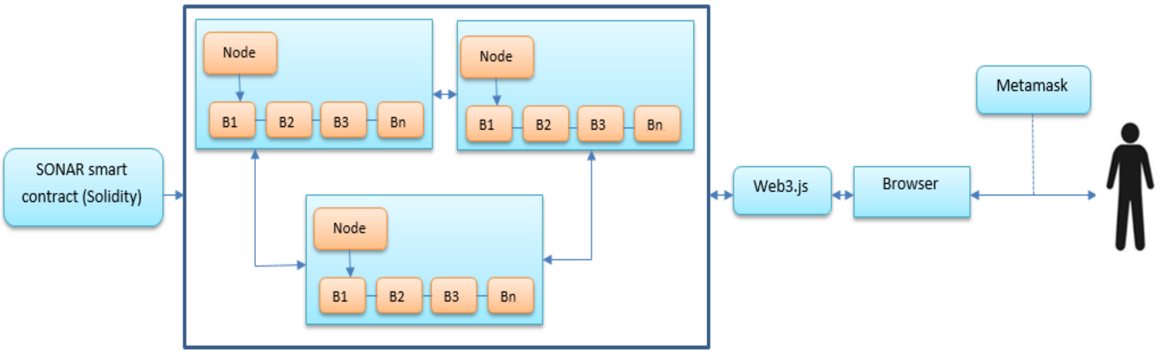


Figure 3. SONR DApps architecture

2.1. Create and deploy a (SONR) contract on the Goerli test network

First, DApps apps create SONR smart contracts using solidity [21], [22]. Solidity is a programming language with object-oriented characteristics that is primarily used to create Ethereum contracts [5]. Remix is an appropriate platform for solidity program testing and debugging. An open-source application called the remix makes it possible to create Solidity contracts directly from a web browser. Solidity contracts can be written and debugged more easily. Because Remix is built in JavaScript, it continues to be used both locally and in browsers. Smart contracts are deployed, tested, and troubleshooted using Remix [23].

The general structure of the SONR contract with genetic algorithm implementation is shown in the Algorithm 1 [24]:

Algorithm 1. Pseudocode of SONR contract

```
Input: ⊳ The population of four sequence as an ID's.
Output: ⊳ eight of sequences
while population not converged do
 selection
 crossover
 add new offspring to original population
end while
end procedure
```

First, the resilience code needs to be compiled into the Bytecode for the Ethereum virtual machine (EVM) before distributing the smart contract. We get two objects after building the solidity code:

- Byte code/EVM code
- Application binary interface (ABI)

Secondly, connect to the Ethereum network by logging in to an Alchemy account to create an API key. Thirdly, install the Metamask and initiate an Ethereum address. We need to install a Hardhat package to complete deploying our SONR contract. Finally, we'll get an address for the block that has been allocated to SONR contract on the Goerli network.

3. RESULTS AND DISCUSSION

Data was gathered via the Goerli test network. The Goerli test network offers ether to test apps and comprehensive data that may be used to monitor how long a transaction takes to complete and how much gas it uses [25]. The cost of gas was assessed to evaluate the transactional computing complexity of the application. The Ethereum network calculates the gas consumption, as illustrated in Figure 4, to be proportionate to both the transaction duration and the computational complexity of a transaction [26]. The reason for this is that nodes tend to favor executing less computationally demanding activities, which slows down transactions that demand a lot of gas to complete. According to Figure 5, the typical transaction takes 8.1 seconds and uses 2.5 gwei.

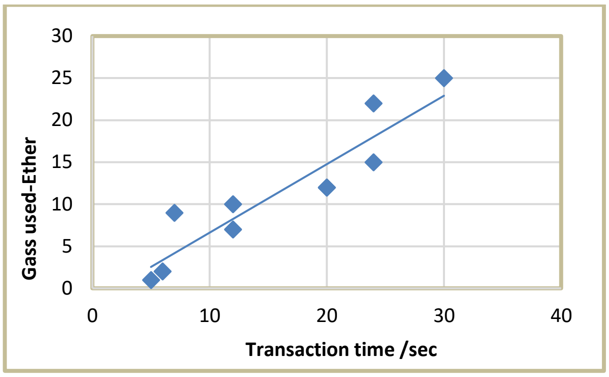


Figure 4. Gas Cost in comparison to transaction time

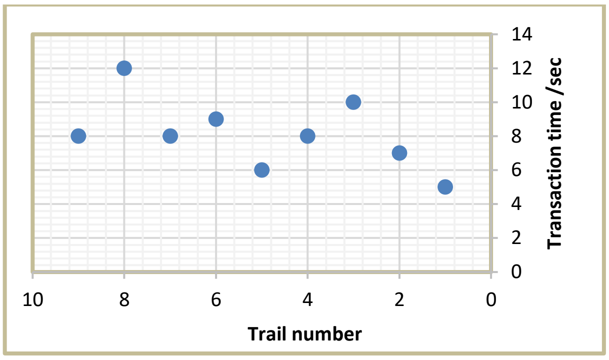


Figure 5. Test network measurements at the time are to blame for the variation in transaction runtimes

The creation of the contract took precisely 12 seconds. This demonstrates that the test network complies with the application's transaction requests rather quickly and effectively. These gas measurements reveal the application's complexity. Variations demonstrate how the load on the test network has varied. Figure 6 total gas usage on the Goerli network after 9 transactions was finished was 46.2 gwei, which is a fair amount for an Ethereum application. Later optimization may reduce the total amount of gas used. Each of these transactions was performed nine times, as shown in Figure 6, for an average transaction time of 8.1 seconds.

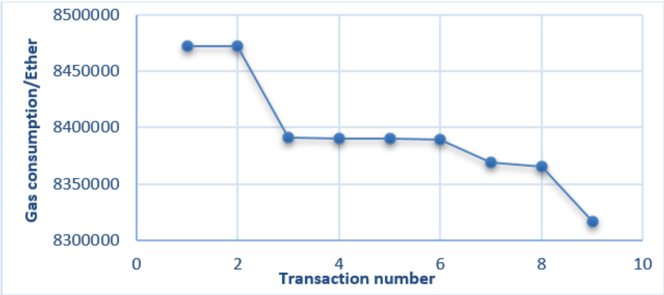


Figure 6. The cumulative gas consumption

Different test-network measures that were in effect such as network performance and latency at the time, might be to blame for the variation in these results. These runtime analyses also took into account how much gas each of these transactions used. By multiplying the quantity of gas utilized by Remix's default gas pricing (1 gas=100 gwei) [18] and Metamask's suggested gas price (1 gas=40 gwei), each transaction was translated into US dollars [27], the network statistics are tracked by the Ethereum project [28]. Table 2 displays these gas prices.

Table 2. Gas cost for each transaction

Transaction type	Gas cost (remix)	Gas cost (metamask)
Deployment	2621353	562.03600000
getDNA	180070	0.000843054010116648
Fitness	284287	0.196571580314008492
Crossover	399319	292.548.074429916375534914
Evolution	421852	0.000843054

4. CONCLUSION

Recently, Ethereum infrastructure has caught the attention of developers who are considering building apps on blockchain platforms in many industries. The enticing traits of DApps are what we've discussed. Next, a potential set of tools needed to quickly create apps using the Ethereum architecture is demonstrated. A smart contract called the SONR is described in detail, along with an application development environment. Further research will focus on scalability issues and ensuring that the built application complies with generally accepted security standards for the education domain.

REFERENCES

[1] T. R. Chowdhury, M. Yusuf, P. Kundu, S. Chakraborty, and N. Biswas, "Crypto pay: design of public blockchain platform," *American Journal of Electronics & Communication*, vol. 3, no. 3, pp. 11–15, Jan. 2023, doi: 10.15864/ajec.3303.

[2] G. Wood., "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[3] R. Taş and O. O. Tanrıöver, "Building a decentralized application on the Ethereum blockchain," in *3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2019 - Proceedings*, Oct. 2019, pp. 1–4, doi: 10.1109/ISMSIT.2019.8932806.

[4] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung, "Decentralized applications: the blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53019–53033, 2018, doi: 10.1109/ACCESS.2018.2870644.

[5] P. Zhang, J. White, D. C. Schmidt, and G. Lenz, "Design of blockchain-based apps using familiar software patterns to address interoperability challenges in healthcare," *22ND HILLSIDE Proc. of Conference on Pattern Languages of Programs*, 2017.

[6] K. Wu, "An empirical study of blockchain-based decentralized applications," *arXiv preprints*, Feb. 2019.

[7] J. Bang and M. J. Choi, "Design and implementation of storage system for real-time blockchain network monitoring system," in *2019 20th Asia-Pacific Network Operations and Management Symposium: Management in a Cyber-Physical World, APNOMS 2019*, Sep. 2019, pp. 1–4, doi: 10.23919/APNOMS.2019.8892967.

[8] T. C. Dao, B. M. Nguyen, and B. L. Do, "Challenges and strategies for developing decentralized applications based on blockchain technology," in *Advances in Intelligent Systems and Computing*, vol. 926, 2020, pp. 952–962, doi: 10.1007/978-3-030-15032-7_80.

[9] "Proof of authority consensus," GeeksforGeeks, 2022, Accessed: Feb. 14, 2023, [Online]. Available: https://www.geeksforgeeks.org/proof-of-authority-consensus/.

[10] S. Joshi, "Feasibility of proof of authority as a consensus protocol model," *arXiv preprints*, Aug. 2021.

[11] A. Bogner, M. Chanson, and A. Meeuw, "A decentralised sharing app running a smart contract on the Ethereum blockchain," in *Proceedings of the 6th International Conference on the Internet of Things*, Nov. 2016, vol. 07-09-Nove, pp. 177–178, doi: 10.1145/2991561.2998465.

[12] F. Blum, B. Severin, M. Hettmer, P. Huckinghaus, and V. Gruhn, "Building hybrid DApps using blockchain tactics -The meta-transaction example," in *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020*, May 2020, pp. 1–5, doi: 10.1109/ICBC48266.2020.9169423.

[13] M. Al-Shabi and A. Al-Qarafi, "Improving blockchain security for the internet of things: challenges and solutions," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 5, pp. 5619–5629, Oct. 2022, doi: 10.11591/ijece.v12i5.pp5619-5629.

[14] T. Min and W. Cai, "Portrait of decentralized application users: an overview based on large-scale Ethereum data," *CCF Transactions on Pervasive Computing and Interaction*, vol. 4, no. 2, pp. 124–141, Jun. 2022, doi: 10.1007/s42486-022-00094-6.

[15] I. R. Abdelhamid, I. T. A. Halim, A. E. A. Ali, and I. A. Ibrahim, "A survey on blockchain for intelligent governmental applications," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 31, no. 1, pp. 501-513, Jul. 2023, doi: 10.11591/ijeecs.v31.i1.pp501-513.

[16] M. I. Akazue, R. Elizabeth Yoro, B. O. Malasowe, O. Nwankwo, and A. A. Ojugo, "Improved services traceability and management of a food value chain using block-chain network: a case of Nigeria," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 29, no. 3, pp. 1623-1633, Mar. 2023, doi: 10.11591/ijeecs.v29.i3.pp1623-1633.

[17] K. Yue *et al.*, "A survey of decentralizing applications via blockchain: The 5G and beyond perspective," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 4, pp. 2191–2217, 2021, doi: 10.1109/COMST.2021.3115797.

[18] M. El Midaoui, E. M. Ben Laoula, M. Qbadou, and K. Mansouri, "Logistics tracking system based on decentralized IoT and blockchain platform," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 23, no. 1, pp. 421–430, 2021, doi: 10.11591/ijeecs.v23.i1.pp421-430.

[19] S. Ganesh Kumar, B. Sriman, A. Murugan, and B. Muruganantham, "IoT-smart contracts in data trusted exchange supplied chain based on block chain," *International Journal of Electrical and Computer Engineering (IJEECS)*, vol. 10, no. 1, pp. 438–446, Feb. 2020, doi: 10.11591/ijece.v10i1.pp438-446.

[20] C. Pujari, B. Muniyal, and C. C. B., "A decentralized consensus application using blockchain ecosystem," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, p. 6399, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6399-6411.

[21] Q. Xu, Z. He, Z. Li, and M. Xiao, "Building an Ethereum-based decentralized smart home system," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, Dec. 2018, pp. 1004–1009, doi: 10.1109/PADSW.2018.8644880.

[22] "Solidity — Solidity 0.5.11 documentation," *Solidity*, 2016, Accessed: Feb. 06, 2023. [Online]. Available: https://docs.soliditylang.org/en/v0.5.11/index.html.

[23] "Remix - Ethereum IDE," *REMIX*, Accessed: Dec. 19, 2022, [Online]. Available: https://remix.ethereum.org/.

[24] H. H. Abdul-Sada and F. Rabee, "The genetic algorithm implementation in smart contract for the blockchain technology," *Al-Salam Journal for Engineering and Technology*, pp. 37–47, Feb. 2023, doi: 10.55145/ajest.2023.02.02.005.
[25] "Goerli Testnet," goerli, Accessed: Feb. 08, 2023, [Online]. Available: https://goerli.net/.
[26] G. Wood, "Ethereum: a secure decentralised generalised transaction ledgers," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.
[27] V. P. Ranganthan, R. Dantu, A. Paul, P. Mears, and K. Morozov, "A decentralized marketplace application on the ethereum blockchain," in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, Oct. 2018, pp. 90–97, doi: 10.1109/CIC.2018.00023.
[28] "Ethereum Gas Tracker," Etherscan, Accessed: Feb. 14, 2023, [Online]. Available: https://etherscan.io/gastracker.

BIOGRAPHIES OF AUTHORS

Hamssah Hussein Abdul-Sada acquired the Bachelorism's degree in computer science from the University of Babylon/Faculty of Computer Science in (2006/2007). She was accepted to study a master's degree in computer science at the University of Kufa/College of Computer Science and Mathematics in (2021/2022), and She is currently in the stage of writing scientific research on the topic of blockchain technology to obtain a master's degree in the above specialization. She can be contacted at email: hamssah.alhassnawe@student.uokufa.edu.iq.

Furkan Rabee is a staff member in the Department of Computer Science, Faculty of Computer Science and Mathematics, at the University of Kufa, Iraq. He got B.Sc. and M.Sc. in Computer Engineering from AL-Nahrian University in 2000 and 2008, Bagdad, Iraq. He got Ph.D. in Computer Science and IT from the School of Computer Science and Engineering, UESTC, Chengdu, China 2015. The research interests include real-time scheduling algorithms, real-time locking protocols, operating systems, parallel processing, distributed system, computer network, IoT, RFID design, mobile computing, cloud computing, and smart cities. He can be contacted at email: furqan.rabee@uokufa.edu.iq.