EasyChair Preprint

№ 10565

Performance Investigation of the Ethereum Blockchain Network During the Process of Data Exchang in Shopping Carts

Mogeeb A. Saeed, Amjaad S. Ali, Abeer Sh. Alhakimi, Aya A. Alaghbari, Neerfan M. Alzoureki and Hamza Mutaher

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 15, 2023

Performance Investigation of the Ethereum Blockchain Network During the Process of Data Exchang in Shopping Carts

Saeed Mogeeb A.
Faculty of Al-Saeed for Engineering and Information Technology
Taiz University
Taiz, Yemen
mogeeb1982@gmail.com

Amjaad Sameer Murshed Abdo Ali
Faculty of Al-Saeed for Engineering and Information Technology
Taiz University
Taiz, Yemen
amjaadalmorshid@gmail.com

Abeer Shaif Abdulrazzaq Ahmed Al_Hakimi
Faculty of Al-Saeed for Engineering and Information Technology
Taiz University
Taiz, Yemen
alhakimiabeer22@gmail.com

Aya Abdullah Abdulelah Abdulrazzak Al_Aghbari
Faculty of Al-Saeed for Engineering and Information Technology
Taiz University
Taiz, Yemen
ayaalaghbari7@gmail.com

Neerfan Mounir Taha Alzoureki
Faculty of Al-Saeed for Engineering and Information Technology
Taiz University
Taiz, Yemen
nervanalzoureki@gmail.com

Hamza Mutaher
School of Computing and Innovative Technologies
British University Vietnam
Hanoi, Vietnam
hamza.a@buv.edu.vn
0000-0001-8798-7538

***Abstract*—This In the ever-evolving landscape of technology, the concept of blockchain first emerged in the early 1990s and has since captured significant attention and adoption in various industries. One such industry is E-commerce, which has witnessed a surge in popularity due to the widespread availability of the internet. However, with the convenience of online shopping comes concerns regarding customer privacy and data security, especially during the payment process. To address these challenges and provide a secure and efficient solution, we have embarked on a research to develop a digital shopping cart using blockchain technology, specifically leveraging the Ethereum network. By harnessing the power of blockchain, our solution ensures the safety of customer data, eliminates the risk of unauthorized changes or fraudulent activities, and offers a seamless and fast payment experience. Through our implementation, we aim to enhance data security and improve overall efficiency and speed in E-commerce transactions. Our digital shopping cart is designed to provide customers with peace of mind, knowing that their personal information is safeguarded and their transactions are executed with integrity. We envision expanding the network performance to accommodate more nodes and incorporating advanced features such as multiple operations in a single block.**

***Keywords*— Blockchain, Ethereum, Decentralized Application, Smart Contracts t**

I. Introduction

E-commerce offers several benefits over conventional purchasing and is growing worldwide. More than 95% of buyers prefer Internet shopping due to its ease, lower prices, and 24/7 availability. COVID-19 and overcrowded shops prompted this change. E-commerce began in the 1970s with e-money transactions. In the late 1990s, shopping carts facilitated online shopping, selection, and payment. The growing number of e-commerce users has raised risks, including card hacking, money theft, and illegal access to personal data, putting consumers at risk. As a way to mitigate risks, digital currencies, particularly blockchain, have become popular. Blockchain, which was initially connected to cryptocurrencies, evolved into an adaptable technology for decentralized money transactions and Smart Contracts [1]. Blockchain securely connects computers worldwide. Data integrity is maintained through real-time copying on all network machines. Blockchain secures huge amounts of data by validating transactions and decentralizes systems as distributed ledgers [2]. Blockchain uses Proof of work (PoW) and Proof of stake (PoS) to secure and legalize information. These technologies verify and safeguard blockchain content. The software we have developed employs a Decentralized Application (DApp) and utilizes Ethereum Smart Contracts. DApps are online apps that run independently on many nodes on a Peer-to-Peer computer network and cover the front and back [3]. The Ethereum platform provides a blockchain-based virtual machine, known as the Ethereum Virtual Machine (EVM), which facilitates the implementation of Smart Contract capabilities [4].

Smart Contracts automatically execute when criteria are satisfied. They allow financial transfers, registration, alerts, and ticket issuing on blockchains. Blockchain transactions are immutable and regulated [5]. The shopping cart is an essential part of E-commerce websites that makes it easier for buyers to buy things. It includes looking for products, putting them in the cart, looking at the details of the products, choosing shipping methods and addresses, and then paying for the products with Visa cards or other suitable methods [6]. But this payment can cause significant problems, like account hacks and money being

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

taken out without permission. This is a substantial flaw in E-commerce. Also, buyers must give private personal information when purchasing, which can cause problems if it gets out without permission. There is also a need to eliminate third parties like banks from the payment process to make it safer, faster, and more convenient. Following are our research questions:

- How do we eliminate the third party (the bank) in the payment process?
- How can the Blockchain network make the shopping cart process more reliable and secure?
- How can we apply the concepts of shopping carts using the Blockchain network?
- How is the process of tracking data in the Blockchain network?
- How will the performance of the Blockchain network be during the data-sharing process?

In this research, we will focus on studying the performance of Ethereum Blockchain during the customer data-sharing process while purchasing, and the following are the objectives of this research:

- Investigate the mechanism that preserves client money and privacy in the Ethereum Blockchain network.
- Improve the efficiency, security, and power of the shopping cart.
- Reduce the parties of the purchasing process to two parties (the seller and the buyer only) without a third party (the bank).

The rest of the paper is organized as follows

II. Related Work

First, Trust in digital transactions is one of the most important aspects the developers must ensure to complete safe transactions. Khan et al.[7] presents a method using Ethereum Blockchain technology to protect digital content and transactions. It utilizes encryption and transparency to prevent forgery and enhance security in digital transactions. To explore the trust in crypto-token DApps, Toufaily[8] proposed a comprehensive framework and provided insights for policymakers and developers to enhance trust and drive adoption in blockchain applications. The framework also provides a model for policymakers, educators, platforms, and developers to enhance trust and drive adoption in Web 3.0 economies.

Marchesi et al.[9] focus on industrial blockchain applications, comparing public blockchains. The proposed Ethereum-based solution incorporates efficient consensus algorithms and a blockchain explorer and addresses the drawbacks of public blockchains. The authors also introduced a new approach for easily customizing Ethereum-based smart contracts in the agri-food industry. It aims to shorten development time while ensuring safety and reliability. A honey production case study showcases the approach, enabling secure traceability and preventing food fraud. Future work includes expanding to other supply chains and blockchain platforms [10]. To ensure transaction traceability and transparency, Kravenkit et al.[11] proposed a blockchain-based traceability system for product recall (TSPR) to enhance recall process transparency. TSPR integrates stakeholders into a single blockchain, storing traceability events, recall data and status. The system is validated using Ethereum smart contracts, ensuring accuracy and visibility. Moreover, Canessane et al.[12] implemented an electronic voting application on the Ethereum Blockchain network, ensuring transparency and reliability. A smart contract prevents multiple voting and enables real-time vote tracking. The application offers time and effort savings for voters, allowing them to vote securely from personal devices.

For the private proof of authority (PoA), Toyoda et al.[13] analyze bottlenecks in private Ethereum blockchain systems and proposes improvements. Using a custom toolset, the authors measure function execution time and identify key bottlenecks, including underutilized multi-threading. The study underscores the need for enhancing performance in private Ethereum blockchains. On the other hand, Madhwal et al.[14] developed an open-source supply chain management blockchain focusing on the Proof of Delivery (PoD) process. Smart contracts improve efficiency and reliability, but complexity limits performance measurement. Pre-contractual steps, further smart contract development, and off-chain transaction recording for dispute resolution are recommended. For performance analysis, Rouhani et al.[15] analyzed blockchain transactions on the Ethereum network, comparing the performance of Parity and Geth clients. Parity is 89.8% faster than Geth under the same system configuration. To optimize the transaction cost, Laurent et al.[16] proposed a method to optimize transaction fees in Ethereum by using a Monte Carlo approach to predict mining probabilities within a given time limit. Experimental results, based on real-life data from Ethereum Constantinople, validate the proposed method's effectiveness. The next sections will elaborate on our contribution to the Ethereum area of research.

III. Prepare Your Paper Before Styling

Based on our analysis of the shopping cart system, we designed diagrams showing how the Ethereum blockchain network works, what its tasks are, and how they relate to each other. Fig 1 shows our project's information flow. It is divided into two parts; client-specific and admin tasks. The blockchain checks and updates client data. Clients buy things while the admin makes smart contracts and products.

Fig 2 explains the functions of each node in the blockchain network where the seller node can add or update the entry (products), and the customer node can invoke select, buy, and pay functions. In Fig 3, the project's process sequence is depicted. The admin requests to enter a new product, verified by the smart contract. The data is saved on the blockchain, making the product available to customers. Customers browse and search for products, view the product details, and enter purchase information. The smart contract validates the data, records the transaction on the blockchain, updates the user interface, and sends a sale confirmation to the customer.

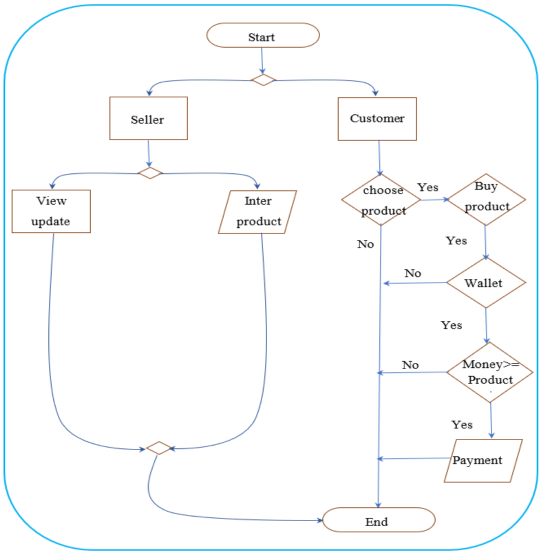

Fig. 1. Sequence of information.

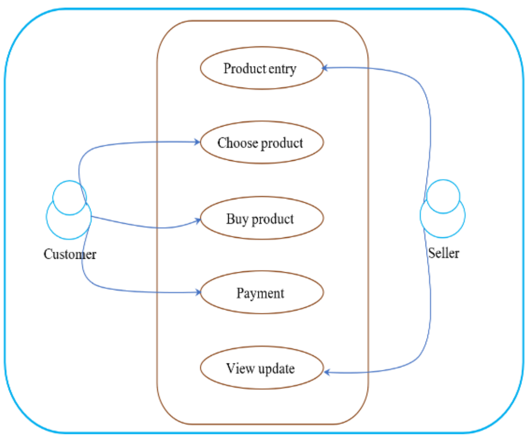

Fig. 2. Node functions.

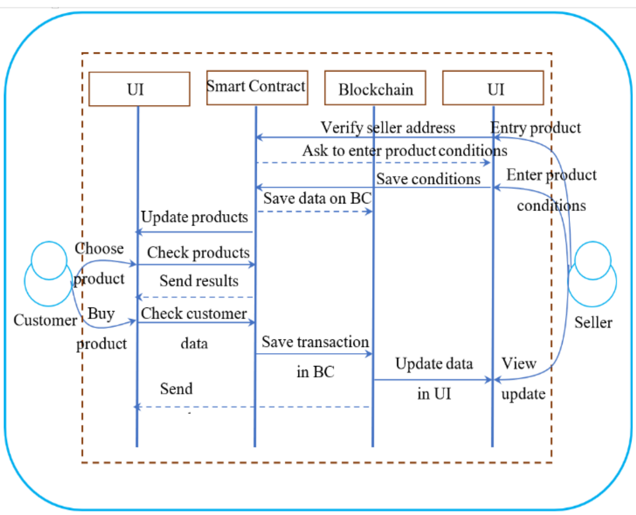

Fig. 3. Sequencing process.

The Fig 4 illustrates the key elements of our project and their interconnections. It depicts two types of users (normal and administrative), the link between products and users, as well as the association between the process, movements, and users.

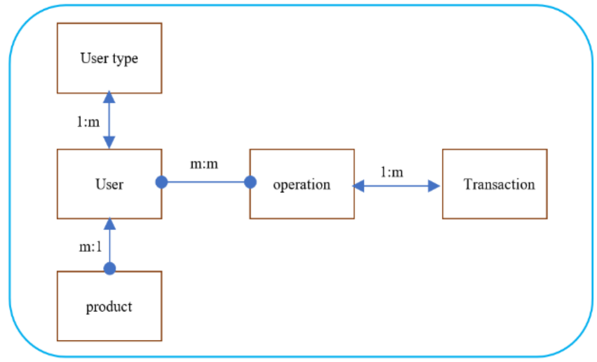

Fig. 4. Sequencing process.

A. Methods

The This section will discuss our projects' essential methods and concepts for developing, deploying, and interacting with the Ethereum blockchain network. These include the following:

- Bootstrap: A front-end development framework that simplifies website creation through predefined templates and a responsive grid system [17].
- Truffle: A framework used for implementing and interacting with Ethereum smart contracts, facilitating their deployment and connection on the Ethereum network [18][19].
- MetaMask: A browser extension wallet that enables secure management and transfer of ether on the Ethereum blockchain [20].
- Web3.js: A JavaScript library employed to establish connections between applications and the Ethereum network, allowing for communication and interaction with Ethereum-based smart contracts [21].
- Ganache: A program designed for creating and testing private Ethereum blockchains, particularly useful for local Ethereum network testing [21].
- Solidity: A high-level contract language that uses JavaScript-like syntax for developing smart contracts on the Ethereum platform [19].
- Contract Address: A unique combination of public and private keys that ensures transaction authenticity on the Ethereum network [22].
- Blocks: Collections of linked transactions forming a chain on the blockchain, containing information such as timestamps, fees, difficulty, and transaction data [22].
- Gas: The fee required to execute transactions or contracts on the Ethereum network, denoted in Gwei [22].

$$
\begin{gathered}
Gas = Measures\ how\ much\ computation\ is\ done \\
Gas\ price = How\ much\ you're\ willing\ to\ pay\ per\ gas \\
of\ work \\
Gas\ limit = Max\ gas\ you're\ willing\ to\ use\ for\ a \\
transaction \\
Tx\ cost = Gas\ used * Gas\ price \quad (1) \\
Gas\ block\ limit = Max\ gas\ allowed\ in\ a\ block
\end{gathered}
$$

IV. Implementation

After In the implementation section, we will utilize the mentioned tools and frameworks to bring the idea to life. Fig 5 below illustrates the sequence of the processes involved in the implementation.

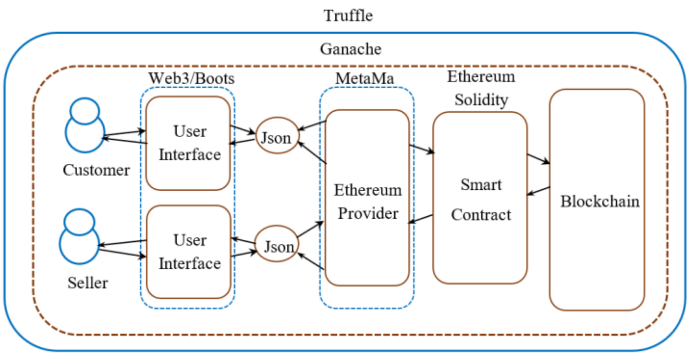

Fig. 5. Implementation Process.

We have two network nodes to simplify the concept: the customer and the seller. The seller can add and view sales listings, while the customer can access and add items to their shopping cart. The following tools and frameworks were used for the implementation and testing.

- Truffle: A developer-friendly tool for building the network and compiling the Smart Contract. It provided ten nodes, each with 100 Eth.
- Ganache: A GUI tool for managing and monitoring the nodes and their transactions.
- MetaMask: A wallet and Ethereum provider used for registration, login, and accessing the Ethereum network.
- Web3: Used to connect the Smart Contract and the Blockchain, allowing functions to be called within the UI framework.

We used Bootstrap as a CSS framework for the UI to create an aesthetically pleasing website. See Fig 6.

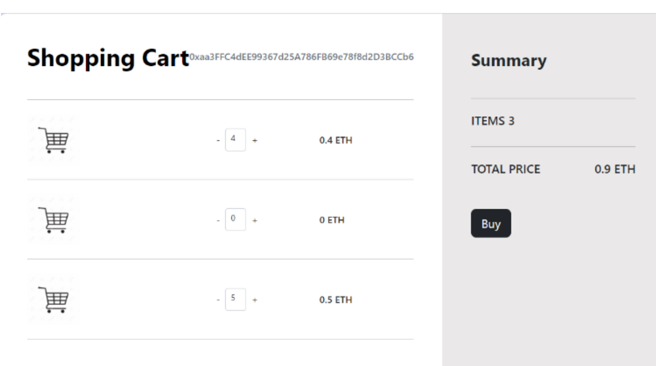

Fig. 6. User interface

V. RESUTL

On listing 1, can you see the results of the transactions

```
Listing 1: Result
Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x608060)
2_SHOPPINGCART.js
================
   Deploying 'SHOPPINGCART'
   ------------------------
   > transaction hash:
0x867a024c47a813836b31b726ee5b4e658581f6b89940a500dac9fb4cf677d515
   > Blocks: 1            Seconds: 9
   > contract address:
0xaBF05aff5Fa8A7f2E010875A6Aab50d523e91606
   > block number:        4
   > block timestamp:     1660318637
   > account:
0x6265791e6EF2a42fc14FC6f06a6d30c3A19Dd286
   > balance:             99.965552
   > gas used:            27338 (0x608060)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.13443950 ETH
```

To calculate the total cost using equation (2) with the data from listing 1, we can substitute the values into the equation as follows

$$total\ cost = gas\ limit \times gas\ price\ per\ unit \quad (2)$$

$$total\ cost = 6721975 \times 20\ gwei = 134439500\ gwei$$

$$total\ cost\ in\ ETH = \frac{total\ cost}{10^9} \quad (3)$$

$$total\ cost\ in\ ETH = \frac{134439500}{10^9} = 0.13443950\ ETH$$

To execute the Account's Transaction, we utilized the Ganache platform with 10 accounts (the maximum), each containing 100 ETH. This allowed us to analyze the network performance during the data exchange process, as depicted in Fig 7.

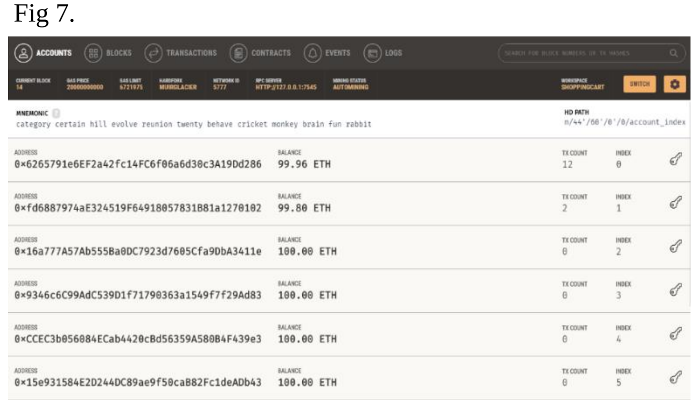

Fig. 7. Accounts

In our implementation, we utilized two accounts to examine the exchange of transactions between them. The results of this analysis are presented in Table I.

In the shopping cart contract, the balance is 99.985552 ETH, which is equivalent to 9,998,555,200,000,000 wei. In Fig 8, it can be observed that Block 0 (Genesis) does not contain any transactions. For more detailed information, please refer to

Table II, The subsequent blocks have been mined and indicate the time when transactions were added to the blocks or submitted to the transaction pool. In Figure 7, Block 3 and Block 4 are executed simultaneously, indicating that different miners can generate blocks at the same time due to the blockchain's working mechanism. Due to this delay, multiple miners may simultaneously solve for the same block and attempt to add it to the blockchain network, leading to a temporary and unsettled state. Nodes within the network must then reach a consensus on which of the newly identified blocks to accept and continue with, while rejecting others, see Table III.

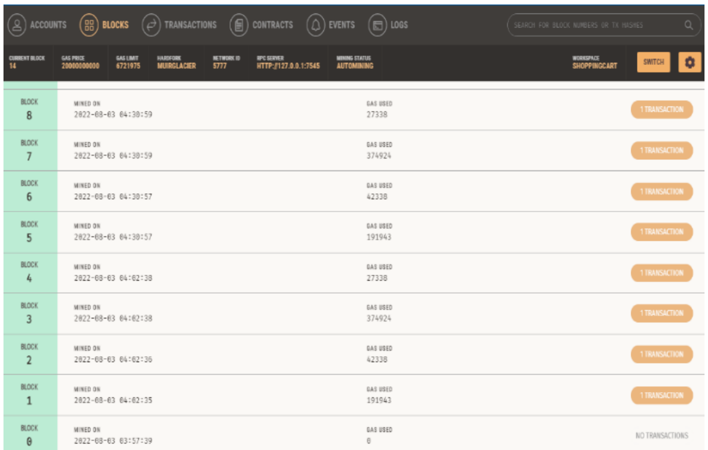

Fig. 8. Blocks

TABLE I. ACCOUNT'S TRANSACTION EXECUTION

Accounts Address	0x6265791e6EF2a42fc14FC6f06a6d30c3A19Dd286		0x6265791e6EF2a42fc14FC6f06a6d30c3A19Dd286	
Balance	99.96 ETH		99.80 ETH	
Block Number	12		2	
T_X Count	12		2	
Tx Value	0.00 ETH		0.10 ETH	
Tx Execution	Yes	No	Yes	No
	6	6	2	0

TABLE II. ACCOUNT'S TRANSACTION EXECUTION

Block	Genesis Block (0)	Block $_{i+1}$(14)
Block Has0068	0x33c8625d1308298ed07af2ad22c7933400ada9df395b6ebd34c74d183b04ed0e	0xaefd632c64286adeccc8c7c0bd0cf308a20610d16892444aa3db70f6aaaa027d
Gas used	0	44998
Value	0.00 ETH	0.10 ETH
Transaction	Not Found.	1
Sender Address	Not Found.	0xfd6887974aE324519F64918057831B81a1270102
Contract Address	Not Found.	0xaBF05aff5Fa8A7f2E010875A6Aab50d523e91606

TABLE III. ACCOUNT'S TRANSACTION EXECUTION

Block Mined in	**Block (3)**	**Block (4)**
Transaction Type	**Contract Creation**	**Contract Call**
T_X Hash	0xb1e5e790a80bd51dd0f0ed3be67488b50f2a512d069da4c18882ca1c6b60eb30	0x867a024c47a813836b31b726ee5b4e658581f6b89940a500dac9fb4cf677d515
Sender Address	0x6265791e6EF2a42fc14FC6f06a6d30c3A19Dd286	0x6265791e6EF2a42fc14FC6f06a6d30c3A19Dd286
Contract Address	0xa56DE901cd742D06A4E9638B9E41d278B0Cbbb4c	0x57A8B91179ac2cA83392B6E22Ff9d9DAF8765766
Mined on (date)	2022-08-03 04:02:38	2022-08-03 04:02:38
T_X Data	0x6080604052336000806101000a81548173ffff...etc	0xfdacd5760000000000000000000000000000000000000000000000000000000000000002
Gas used	374924	27338
Gas limit	6721975	34172
Contract Name	Not found	Migrations
Function	Not found	setCompleted(completed: uint256)
Inputs	Not found	2 (the shopping Cart Smart Contract)

VI. PERFORMANCE EVALUATION

The latency of a transaction TX is the delay between the transmission of TX through the Ethereum client and the creation of the block that includes TX. In this network, the average block generation time is around 10 to 19 seconds.

Throughput shows the number of successful Transactions per second in the blocks. Throughput is determined by Equation.4:

$$Throughput(tps) = \frac{NT}{Block\ Period} \qquad (4)$$

Where NT is the total number of transactions submitted, Block Period is the average block generation time and tps is transactions per second.

For example, according to our results, we can calculate Throughput as follows:

$$Throughput = \frac{14}{10} = 1.4 \approx 1\ tps$$

VII. CONCLUSION

In this paper, we analyzed the performance of the Ethereum blockchain network for data exchange between two nodes (customer and seller) using a shopping cart. The Ethereum blockchain demonstrated fast, reliable, cost-effective, and high-performance features, enhancing the efficiency of the shopping cart. Our simulation focused on a two-node scenario, but we aim to expand the network performance analysis to include more nodes and enable multiple operations in a single block. Additionally, we plan to develop the shopping cart to integrate it with real-world websites.

References

[1] Julija Golosova and Andrejs Romanovs, "The Advantages and Disadvantages of the Blockchain Technology," in IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), 2018.

[2] G. Habib, S. Sharma, S. Ibrahim, I. Ahmad, S. Qureshi, and M. Ishfaq, "Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing," Future Internet, vol. 14, no. 11, Nov. 2022, doi: 10.3390/fi14110341.

[3] Ruhi Taş and Ömer Özgür Tanrıöver, "Building A Decentralized Application on the Ethereum Blockchain," in 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2019.

[4] E. Hildenbrandt et al., "KEVM: A complete formal semantics of the ethereum virtual machine," in Proceedings - IEEE Computer Security Foundations Symposium, IEEE Computer Society, Aug. 2018, pp. 204–217. doi: 10.1109/CSF.2018.00022.

[5] Alkhansaa Abuhashim and Chiu C. Tan, "Smart Contract Designs on Blockchain Applications," in IEEE Symposium on Computers and Communications (ISCC), 2020. doi: 10.1109/ISCC50000.2020.9219622.

[6] O. Mohammed, A. Rababah, and F. A. Masoud, "Key Factors for Developing a Successful E-commerce Website," 2010. [Online]. Available: http://www.ibimapublishing.com/journals/CIBIMA/cibima.html

[7] U. Khan, Z. Y. An, and A. Imran, "A Blockchain Ethereum Technology-Enabled Digital Content: Development of Trading and Sharing Economy Data," IEEE Access, vol. 8, pp. 217045–217056, 2020, doi: 10.1109/ACCESS.2020.3041317.

[8] E. Toufaily, "An integrative model of trust toward crypto-tokens applications: A customer perspective approach," Digital Business, vol. 2, no. 2, p. 100041, 2022, doi: 10.1016/j.digbus.2022.100041.

[9] L. Marchesi, M. Marchesi, R. Tonelli, and M. I. Lunesu, "A blockchain architecture for industrial applications," Blockchain: Research and Applications, vol. 3, no. 4, Dec. 2022, doi: 10.1016/j.bcra.2022.100088.

[10] L. Marchesi, K. Mannaro, M. Marchesi, and R. Tonelli, "Automatic Generation of Ethereum-Based Smart Contracts for Agri-Food Traceability System," IEEE Access, vol. 10, pp. 50363–50383, 2022, doi: 10.1109/ACCESS.2022.3171045.

[11] S. Kravenkit and C. So-In, "Blockchain-Based Traceability System for Product Recall," IEEE Access, vol. 10, pp. 95132–95150, 2022, doi: 10.1109/ACCESS.2022.3204750.

[12] Canessane R. Aroul, N. Srinivasan, Beuria Abinash, Singh Ashwini, and Kumar B. Muthu, "Decentralised Applications Using Ethereum Blockchain," in Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), 2019. doi: 10.1109/ICONSTEM.2019.8918887.

[13] K. Toyoda, K. Machi, Y. Ohtake, and A. N. Zhang, "Function-Level Bottleneck Analysis of Private Proof-of-Authority Ethereum Blockchain," IEEE Access, vol. 8, pp. 141611–141621, 2020, doi: 10.1109/ACCESS.2020.3011876.

[14] Y. Madhwal, Y. Borbon-Galvez, N. Etemadi, Y. Yanovich, and A. Creazza, "Proof of Delivery Smart Contract for Performance Measurements," IEEE Access, vol. 10, pp. 69147–69159, 2022, doi: 10.1109/ACCESS.2022.3185634.

[15] Rouhani Sara and Deters Ralph, "Performance analysis of ethereum transactions in private blockchain," in 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017. doi: 10.1109/ICSESS.2017.8342866.

[16] A. Laurent, L. Brotcorne, and B. Fortz, "Transaction fees optimization in the Ethereum blockchain," Blockchain: Research and Applications, vol. 3, no. 3, Sep. 2022, doi: 10.1016/j.bcra.2022.100074.

[17] A. Paudyal, "Developing Video Chat Application with ReactJs And WebRTC," 2021.

[18] R. Delhougne, "Concolic-Fuzzing of JavaScript Programs using GraalVM and Truffle," 2021. [Online]. Available: https://github.com/Z3Prover/z3

[19] Kushal Chaganlal Choudhary, Saurabh Achal Agrawal, Mihir Manohar Gadhe, and Rohini Pise, "Decentralised voting with Ethereum blockchain," International Journal of Innovation Engineering and Science Research, vol. 5, pp. 2581–4591, 2021, [Online]. Available: www.ijiesr.com

[20] S. Shakya and V. Kapoor, "A Decentralized Polling System Using Ethereum Technology," Journal of Information Technology Management, vol. 14, pp. 1–8, 2022, doi: 10.22059/JITM.2022.85645.

[21] S. K. Panda and S. C. Satapathy, "An Investigation into Smart Contract Deployment on Ethereum Platform Using Web3.js and Solidity Using Blockchain," 2021, pp. 549–561. doi: 10.1007/978-981-16-0171-2_52.

[22] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, "SmartCheck: Static analysis of ethereum smart contracts," in Proceedings - International Conference on Software Engineering, IEEE Computer Society, May 2018, pp. 9–16. doi: 10.1145/3194113.3194115.