# W2E (Workout to Earn): A Low Cost DApp based on ERC-20 and ERC-721 standards

Do Hai Son[†]
dohaison1998@vnu.edu.vn

Nguyen Danh Hao[‡]
haonguyen.uet@gmail.com

Tran Thi Thuy Quynh[‡]
quynhttt@vnu.edu.vn

Le Quang Minh[†]
quangminh@vnu.edu.vn

[†] VNU Information Technology Institute, Hanoi, Vietnam
[‡] VNU University of Engineering and Technology, Hanoi, Vietnam

***Abstract*—Decentralized applications (DApps) have gained prominence with the advent of blockchain technology, particularly Ethereum, providing trust, transparency, and traceability. However, challenges such as rising transaction costs and block confirmation delays hinder their widespread adoption. In this paper, we present our DApp named W2E - Workout to Earn, a mobile DApp incentivizing exercise through tokens and NFT awards. This application leverages the well-known ERC-20 and ERC-721 token standards of Ethereum. Additionally, we deploy W2E into various Ethereum-based networks, including Ethereum testnets, Layer 2 networks, and private networks, to survey gas efficiency and execution time. Our findings highlight the importance of network selection for DApp deployment, offering insights for developers and businesses seeking efficient blockchain solutions. This is because our experimental results are not only specific for W2E but also for other ERC-20 and ERC-721-based DApps.**

***Index Terms*—W2E, Decentralized application, Gas consumption, Transaction execution time**

## I. Introduction

DApps have been conceptualized long before blockchain technology. However, this term is only widely known along with the launch of Ethereum (Eth) blockchain network in 2014 [1]. DApp is a distributed application that operates on Peer-to-Peer (P2P) blockchain networks. Nowadays, the types of DApps are very diverse, e.g., finance, social, games, storage, and so on, supported on a website or mobile device. Since running on blockchain platforms, DApp inherits its security properties, e.g., trustworthiness, traceability, transparency, etc.

Nevertheless, there are a few challenges that lead to DApps not being as widely utilized as personal computer and mobile applications at present. The first challenge is the rapid increase in transaction costs, also known as gas in Ethereum [2]. To illustrate, the average transaction cost for a regular Ethereum native token (ETH) transfer is $2.07 USD [3] on Jan. 2024. This number of ERC-20 standard-based transfer token functions is $4.07 USD. The cost for an essential function such as transferring is too high compared to the fee-free banking transfer service. The second challenge is the delay due to block confirmation times. For example, Ethereum fixed the block confirmation times at 12 seconds. That means, in the worst case, a transaction needs up to 12 seconds to be confirmed and ready for further steps in DApps. Many blockchain protocols, consensus mechanisms, and smart contract (SM) standards are proposed to deal with these challenges [4]. Based on the official statistics [5], DApps are deployed on a lot of blockchain networks. In [6], the authors explored factors that should guide users' choices and ensure a seamless match between the DApp and the blockchain platform, such as simplicity, cost, performance, and so forth. However, there is currently no practical research available on the selection of an optimal network for DApps. Especially for small businesses, choosing an unsuitable deployment network can lead to unnecessary fees and time on the end users' side.

In this work, we focus on developing a mobile DApp called W2E - Workout to Earn. The objective of this application is to motivate users to exercise by rewarding them with Non-fungible tokens (NFTs) upon the completion of workout missions. W2E is a kind of general Web & Contract as a DApp according to classification in [1]. In detail, developers create a web or application front-end to link between smart contracts and typical missions, e.g., games, trading, or workouts, as in our project. The incentives are not directly real-world money but decentralized tokens, which can be exchanged for real-world money on trading markets, i.e., Binance. The tokens are generated through SMs, which are defined in Ethereum improvement proposals (EIPs). In this study, we use two popular token standards, i.e., ERC-20 and ERC-721. The former standard enables users to generate new tokens akin to ETH and subsequently conduct transfers, approvals, or minting. The latter standard represents ownership of NFT items, which can be images, items, and so forth. The link between ERC-20 and ERC-721 in W2E is that users do not directly buy ERC-721-based items by ETH but through our defined ERC-20 tokens, named DMD. DMD tokens can be purchased from the market by ETH or rewarded after completing workout missions. We then consider choosing an effective Ethereum-based blockchain network to deploy

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project "Agricultural IoT based on Edge Computing" and financially supported by NICT http://www.nict.go.jp/en/index.html.
Corresponding author: Tran Thi Thuy Quynh (quynhttt@vnu.edu.vn).

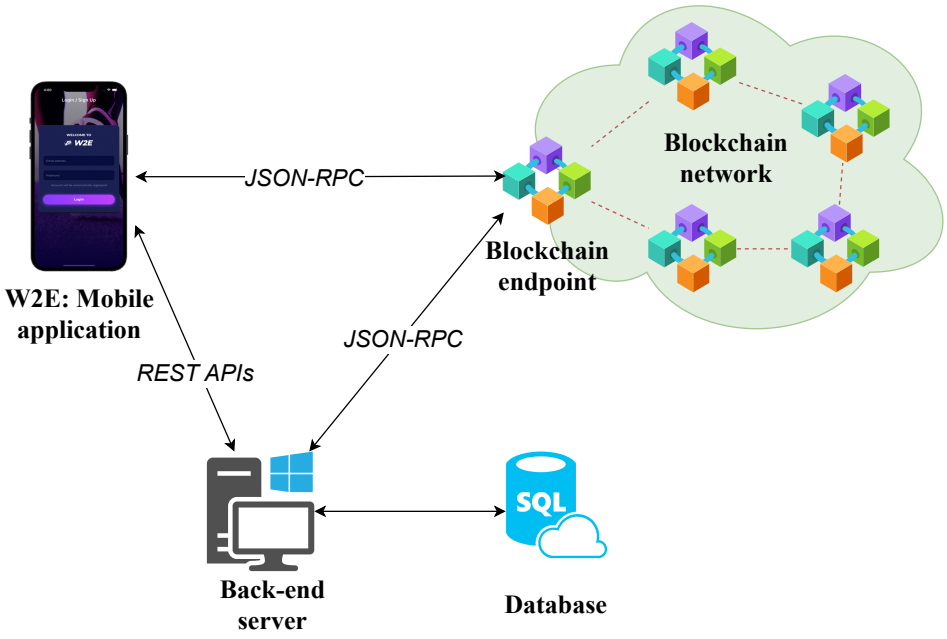



Fig. 1: Architecture of W2E application.

W2E DApp. Ethereum is now not only a standalone network but is the backbone of other decentralization solutions, i.e., SideChain (SC) and Layer 2 (L2) [7]. Hence, we deploy our W2E on several Ethereum-based networks, i.e., Eth 1.0 testnet (Proof-of-Authority consensus mechanism - PoA), Eth 2.0 testnet (Proof-of-Stake consensus mechanism - PoS), Polygon testnet (L2), and Optimism testnet (L2). Moreover, besides testnet and mainnet on the public internet, a private Ethereum network is a good solution for specific purposes. For example, businesses can host a private Ethereum network and deploy W2E in this network, which has been proven to provide good efficiency in terms of gas and validation time [8]. In this project, we built small-scale private Ethereum 1.0 and 2.0 networks [8] to illustrate the private Ethereum solutions. As mentioned above [6], we consider two important factors, i.e., gas consumption (money) and time consumption (delay), on these Ethereum-based blockchain networks. Our experiments reveal significant discrepancies in gas usage and operation time across various networks. Note that since W2E is developed based on well-known ERC-20 and ERC-721 standards, our experimental results also partly reflect practical DApps that use these two ERC standards.

The main contributions of this work are as follows: (i) the method to build W2E - a general DApp, its smart contracts source code is provided in [1]; (ii) analysis of the gas and time consumption for several popular token standards on different Ethereum-based networks. The architecture and functions of W2E are presented in section II. The deployment analysis of W2E's SM is shown in section III. Section IV concludes this work and offers some potential research directions.

[1]https://github.com/DoHaiSon/W2E_smart_contract

## II. Workout to Earn (W2E)

W2E is a DApp that encourages exercise running on the iOS and Android platforms. W2E aims to reward users with a number of tokens (DMD) after finishing a workout, i.e., running. Users can sell their tokens (ERC-20) or use them to buy NFT items (ERC-721) on W2E's market. Both DMD tokens and NFT items (Pets) are tradeable.

Fig. 1 shows the architecture of the W2E application. There are three main components in W2E, as follows:

1) W2E mobile application: this is the user interface of W2E where users can record their workout, earn DMD tokens, and trade NFT items. This component links to the back-end server to register and authenticate user accounts through the REST API protocol. On the other hand, the W2E application also connects to blockchain endpoints, i.e., Ethereum, Polygon, etc., through JSON-RPC connections. This link takes responsibility for sending transactions and loading users' balance information. W2E is developed on the Flutter framework.
2) Blockchain endpoint: this is a node in the interested blockchain network, e.g., Ethereum mainnet/testnet/private networks, Solana, and so on. The node acts as a bridge to broadcast transactions of the W2E application and back-end server into the blockchain network via the synchronous process.
3) Back-end server: this server accounts for several purposes, i.e., deploying smart contracts and storing users' account information and workout history in a centralized database. Additionally, the back-end server always listens to events from W2E smart contracts from the blockchain endpoint. That is a cache database for the

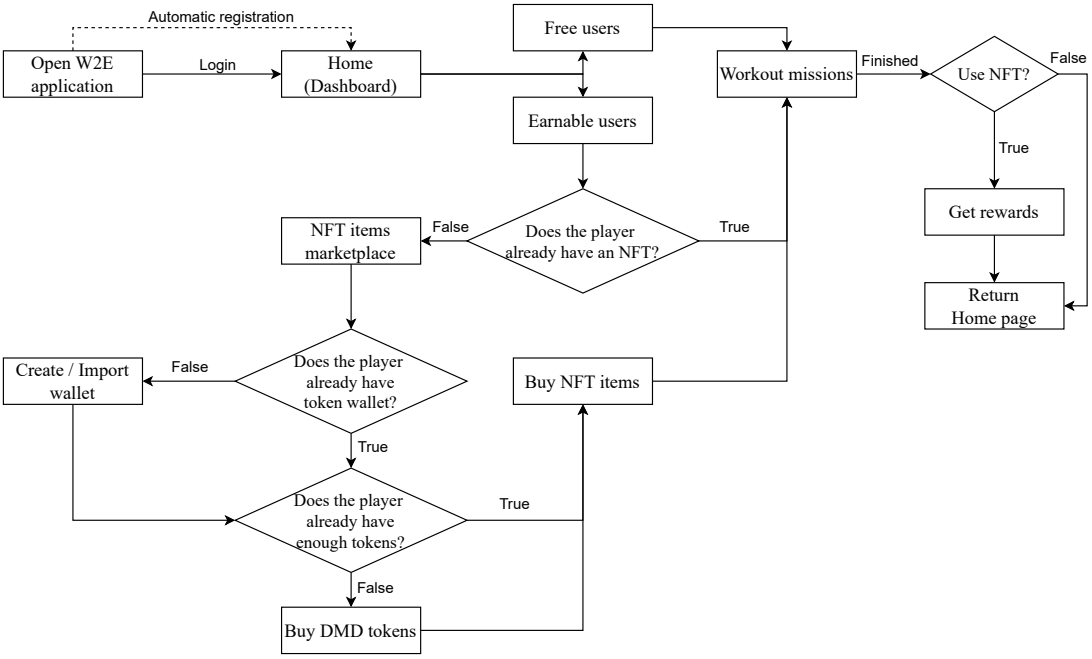



Fig. 2: W2E: Users' flowchart.

W2E mobile application to load tokens and NFT information faster.

Fig. 2 is the users' flowchart of W2E. Our application requires both an internet and GPS connection to track the users' velocity and distance. There are five main steps in W2E, as follows:

1) Users log into their W2E account with email and password. W2E automatically registers an account if this device has never logged in before.
2) Users import or create their external-owned account (EOA) into the W2E application. The creation phase is based on the mnemonic generation algorithm - BIP39 [9].
3) Users require at least one pet to be able to receive rewards from workout missions. We call them "Earnable users". Each pet corresponds to a different reward rate for the user's training process. If users do not have any pets, they can buy them from the NFT items marketplace or act as non-profit users. Pets and DMD can be purchased with DMD tokens and ETH, respectively.
4) Users engage in training missions. W2E captures metrics such as running time, distance, speed, and step count to help users track their workout progress.
5) At the end, W2E rewards users an amount of DMD tokens based on their workout record and the bonus rate for the selected pet.

In detail, smart contracts of ERC-20-based DMD tokens and ERC-721-based NFT items are provided in our public repository: github.com/DoHaiSon/W2E_smart_contract. There are two approaches to determining the price of DMD tokens, which serve as monetary rewards for users. First, the business that released the W2E application sets the value of DMD tokens, using them as incentives for their users. Alternatively, DMD's price can be influenced by market dynamics, reflecting the demand and supply generated by players who trade this token, a common approach adopted by various DApp gaming applications. Fig. 3 is some real screenshots captured from the W2E application. The dashboard and NFT marketplace interfaces are shown in Fig. 3a. Then, Fig. 3b is a record of a running mission on the W2E application. The interfaces of buying DMD tokens and acquiring pets are illustrated in Fig. 3c and Fig. 3d, respectively.

## III. Low Cost Deployment Analysis

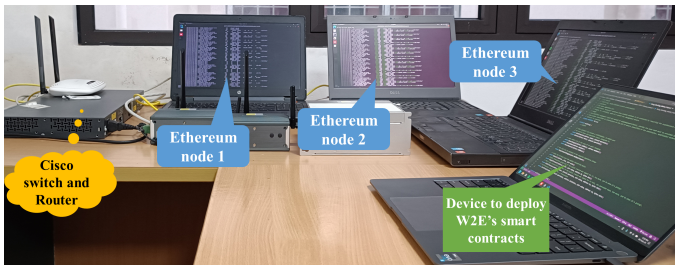
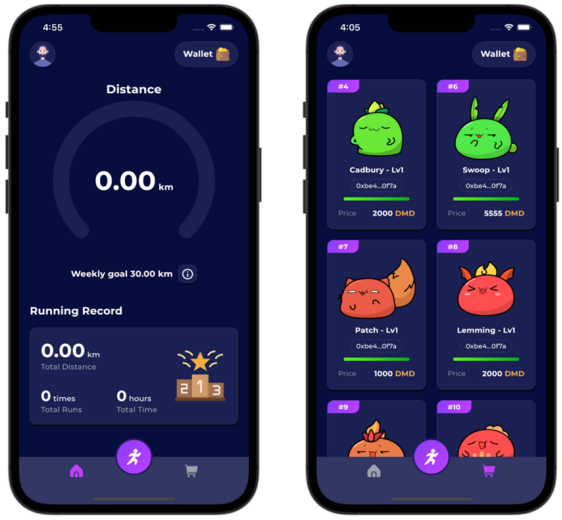
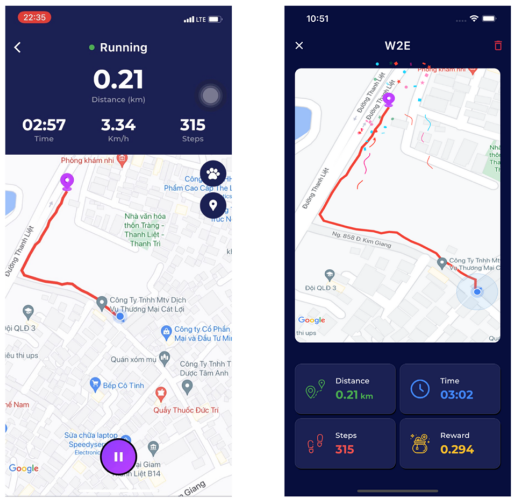



Fig. 4: A small-scale private Ethereum 1.0 and 2.0 network configuration.

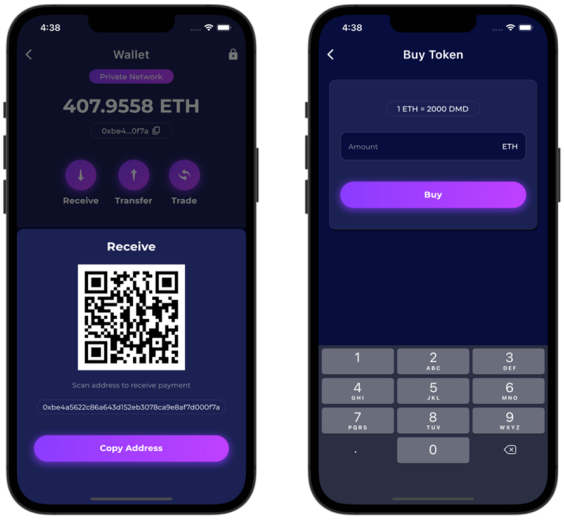
In this section, we deploy W2E smart contracts on several Ethereum-based networks to evaluate performance discrepancies across various implementations. Table I presents a list of six networks that are considered in this work. Four of these networks are public Ethereum-based blockchain testnets available on the internet. Their configurations closely resemble



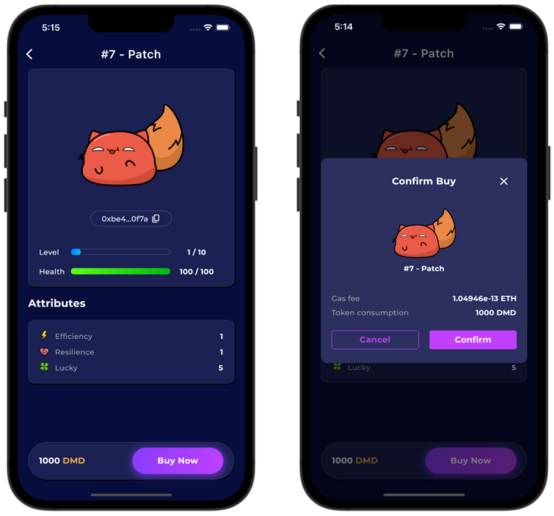

(a) Dashboard and Market




(b) Workout record and rewards




(c) Buy DMD tokens (ERC-20)




(d) Buy an NFT item (ERC-721)

Fig. 3: Interfaces of main functions on W2E application.

the mainnet, but their tokens are non-valuable. We do not set up these networks but use APIs of infura.io as a bridge between the W2E application and the blockchain endpoint. The remaining two networks are private, managed by us, and configured differently from both the mainnet and testnet. They are set up in our laboratory, as shown in Fig. 4. This small-scale setup has three Ethereum nodes and a device for developing the W2E application. All devices are connected to a local network by a Cisco switch. However, any business can adopt and host a blockchain network. The private Eth 1.0 network is launched using an older version of Ethereum and modified to reduce the block confirmation times, according to our previous work [8].

We first examine the gas consumption of a few ERC-based smart contracts, i.e., ERC-20, ERC-721, ERC-1155, and ERC-777. As shown in Table II, the fees for deploying W2E's and other SMs on networks exhibit significant discrepancies. When considering the testnets, it is evident that two L2 networks significantly reduce gas consumption. L2 networks were specifically designed to lower transaction costs, which sets them apart from Ethereum 1.0 and 2.0 networks. Particularly noteworthy is the Polygon testnet, which requires only about 10% of the gas necessary to deploy SMs compared to the Eth 2.0 testnet. Eth 1.0 testnet horribly consumes gas, which led to the Eth 2.0 upgrade. In contrast, private networks, owing to custom configurations, entail trivial deployment fees. Specifically, deploying ERC-721, ERC-1155, and ERC-777 contracts incurs fees of approximately 0.04 Gwei.

While the deployment fee for SMs may be substantial, it is a one-time expense. Therefore, our primary concern lies in the transaction costs associated with activities, such as purchasing and selling NFTs or DMD tokens. Table III shows the execution fees for W2E's smart contract functions, i.e., buy, sell, and cancel NFT items. The results are close to deployment fees

TABLE I: The list of deployment networks.

| Name | Network config | Source code / API | Version |
|---|---|---|---|
| **Eth 1.0 testnet** | Goerli testnet | Infura.io [10] | - |
| **Eth 2.0 testnet** | Ethereum-Sepolia testnet | Infura.io [10] | - |
| **Private Eth 1.0** | Custom config [8] | go-ethereum [11] | 1.10.4 |
| **Private Eth 2.0** | Mainnet config | go-ethereum [11]<br>prysm [12] | 1.10.22<br>3.2.0 |
| **Polygon testnet** | Mumbai testnet | Infura.io [10] | - |
| **Optimism testnet** | Optimism-Sepolia testnet | Infura.io [10] | - |

TABLE II: Gas consumption to deploy the W2E and other ERC-based smart contracts in Gwei.

| Token / Network | **ERC-20** (W2E) | **ERC-721** (W2E) | **ERC-1155** | **ERC-777** |
|---|---|---|---|---|
| **Eth 1.0 testnet** | 5644679.62 | 5611030.26 | 5543731.55 | 5602617.92 |
| **Eth 2.0 testnet** | 395379.94 | 378555.26 | 353318.25 | 370142.93 |
| **Private Eth 1.0** | 6.84 | 6.35 | 6.44 | 6.58 |
| **Private Eth 2.0** | 0.05 | 0.04 | 0.04 | 0.04 |
| **Polygon testnet** | 48791.57 | 50474.04 | 41220.46 | 42902.93 |
| **Optimism testnet** | 82440.92 | 92535.73 | 79917.22 | 117772.75 |

TABLE III: Gas consumption to execute W2E's functions in Gwei.

| Function / Network | **Buy NFT** | **Sell NFT** | **Cancel NFT** |
|---|---|---|---|
| **Eth 1.0 testnet** | 6511150.56 | 5737215.35 | 4130458.56 |
| **Eth 2.0 testnet** | 529977.37 | 429029.3 | 319668.89 |
| **Private Eth 1.0** | 10.85 | 7.95 | 4.78 |
| **Private Eth 2.0** | 0.09 | 0.07 | 0.03 |
| **Polygon testnet** | 74028.59 | 59727.61 | 43744.16 |
| **Optimism testnet** | 126185.09 | 92535.73 | 74869.82 |

where Polygon testnet and private Eth 2.0 outperform other networks. We can conclude that if businesses want to deploy their applications onto Ethereum-based networks, they should consider SC or L2 solutions, e.g., Polygon or Optimism, to reduce gas consumption. On the other hand, private network solutions may fit specific purposes, e.g., small businesses want to control their blockchain network themselves and do not want users to bear transaction fees.

We then investigate time consumption/delay due to transaction confirmation time. This directly impacts the user experience, as simple operations such as buying tokens can take a lot of time. Table IV is the deployment time of the DMD token at five different timestamps. Initially, there was a difference in smart contract deployment time at different timestamps, which lasted up to over $40$ seconds. This is understandable because the deployment transaction can come at a time when the block is almost to be validated (good case) or take a long time for the block to be validated (bad case). Considering different networks, the private Eth 2.0 still stands out as the best solution, with a delay of almost less than $4$ seconds. In contrast to gas consumption, the Optimism testnet outperforms the Polygon and two Eth testnets. This performance variation reflects a trade-off between gas consumption and time delay considerations of the Polygon and Optimism networks.

Finally, we consider the time consumption to execute the buy NFT transactions of the W2E application at five timestamps. The experimental results in Table V show similarities with Table IV. If testnets are used, W2E publishers consider using L2 networks to reduce transaction latency. On the contrary, although it reduces decentralization, private Eth 2.0 networks provide significant efficiency compared to testnet solutions.

## IV. Conclusion

This study introduced the W2E application and investigated the implementation costs across various blockchain solutions. We first described the architecture and users' flowchart of the W2E application. We then deployed W2E's smart contracts onto six Ethereum-based networks. Through experiments, L2 solutions or private networks offer enhanced efficiency in gas usage and transaction execution time. Since W2E is a general DApp, developers can adopt the idea for their project. In the future, we will consider the performance of a DApp when the number of blockchain nodes is scaled up.

TABLE IV: Time consumption to deploy the DMD token SM at different timestamps.

| Network \ Timestamp | 1st (ms) | 2nd (ms) | 3rd (ms) | 4th (ms) | 5th (ms) | Average (ms) |
|---|---|---|---|---|---|---|
| **Eth 1.0 testnet** | 38489 | 38887 | 36808 | 40094 | 35500 | 37955.6 |
| **Eth 2.0 testnet** | 20484 | 20491 | 21660 | 18660 | 17578 | 19774.6 |
| **Private Eth 1.0** | 7057 | 5104 | 5758 | 5980 | 6830 | 6145.8 |
| **Private Eth 2.0** | 3849 | 3849 | 3775 | 3539 | 4104 | 3823.2 |
| **Polygon testnet** | 10788 | 10484 | 9771 | 8074 | 10920 | 10007.4 |
| **Optimism testnet** | 5880 | 6196 | 5021 | 4275 | 5024 | 5279.2 |

TABLE V: Time consumption to execute the W2E's buy NFT function at different timestamps.

| Network \ Timestamp | 1st (ms) | 2nd (ms) | 3rd (ms) | 4th (ms) | 5th (ms) | Average (ms) |
|---|---|---|---|---|---|---|
| **Eth 1.0 testnet** | 39492 | 38004 | 38808 | 42560 | 41224 | 40017.6 |
| **Eth 2.0 testnet** | 18600 | 18112 | 19750 | 21320 | 19682 | 19492.8 |
| **Private Eth 1.0** | 6168 | 6100 | 4867 | 6480 | 6941 | 6111.2 |
| **Private Eth 2.0** | 4250 | 4200 | 3744 | 3899 | 4007 | 4020 |
| **Polygon testnet** | 11000 | 12982 | 10026 | 10145 | 9860 | 10802.6 |
| **Optimism testnet** | 5120 | 6087 | 5000 | 4384 | 4932 | 5104.6 |

## REFERENCES


[1] P. Zheng, Z. Jiang, J. Wu, and Z. Zheng, "Blockchain-based decentralized application: A survey," *IEEE Open Journal of the Computer Society*, vol. 4, pp. 121–133, Mar. 2023.

[2] C. Liu, J. Gao, Y. Li, H. Wang, and Z. Chen, "Studying gas exceptions in blockchain-based cloud applications," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–25, Jun. 2020.

[3] N. Sawinyh, "Ethereum Gas Fees," Accessed: Jan. 18, 2024. [Online]. Available: https://defiprime.com/gas-today

[4] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Computing Surveys*, vol. 54, no. 8, Oct. 2021.

[5] DappRadar, "Top Blockchain Dapps," Accessed: Feb. 2, 2024. [Online]. Available: https://dappradar.com/rankings

[6] S. Nanayakkara, M. Rodrigo, S. Perera, G. Weerasuriya, and A. A. Hijazi, "A methodology for selection of a blockchain platform to develop an enterprise system," *Journal of Industrial Information Integration*, vol. 23, p. 100215, Sept. 2021.

[7] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16 440–16 455, Jan. 2020.

[8] D. H. Son, T. T. T. Quynh, T. V. Khoa, D. Thai Hoang, N. L. Trung, N. Viet Ha, D. Niyato, D. N. Nguyen, and E. Dutkiewicz, "An effective framework of private ethereum blockchain networks for smart grid," in *International Conference on Advanced Technologies for Communications (ATC)*, Oct. 2021, pp. 312–317.

[9] P. Marek, R. Pavol, V. Aaron, and B. Sean, "Mnemonic code for generating deterministic keys," Accessed: Feb. 5, 2024. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

[10] Infura Inc, "Web3 Development Platform — IPFS API & Gateway — Blockchain Node Service," Accessed: Jan. 10, 2024. [Online]. Available: https://www.infura.io

[11] Ethereum, "Official Go implementation of the Ethereum protocol," Accessed: Dec. 14, 2023. [Online]. Available: https://github.com/ethereum/go-ethereum

[12] Prysmatic Labs, "Prysm: An Ethereum Consensus Implementation Written in Go," Accessed: Jan. 10, 2024. [Online]. Available: https://github.com/prysmaticlabs/prysm/tree/v3.2.0