Effective Ethereum Staking in Cryptocurrency Exchanges

Yuto Takei
Mercari, Inc., and
Tokyo Institute of Technology

Kazuyuki Shudo
Kyoto University

***Abstract*—Cryptocurrency staking has become a popular investment option, and multiple cryptocurrency exchanges have been offering staking services to customers. Among the various proof-of-stake cryptocurrencies, Ethereum has become one of the most attractive choices. However, due to the complex architecture of Ethereum, exchanges face several challenges in designing and operating their staking systems while ensuring security and efficiency. In this paper, we analyze the solo-staking method on Ethereum and identify four challenges that exchanges are likely to encounter: wallet configuration, validator key security, stable validator node operation, and profitability. To address these challenges, we propose certain solutions, such as implementing a multi-tiered wallet configuration for customer assets and conducting validator operations on cloud platforms. We have implemented some of these proposed methods on the cloud, and successfully achieved stable operation for the Holesky testnet. We have also identified additional challenges that need to be addressed. We summarize them as open challenges and show the research direction.**

***Index Terms*—Ethereum, staking, cryptocurrency, exchange, security, wallet, cloud, proof of stake (PoS)**

1. Introduction

Staking in cryptocurrency was initially known among cryptocurrency enthusiasts as a technique to make profits from coins in possession during the early days of proof-of-stake (PoS) cryptocurrencies around 2017. It has now become accessible to general consumers through hardware wallets or other means. Several major cryptocurrency exchanges offer staking options as financial services, making it appealing to individuals interested in investments. There are also financial benefits for exchanges from profits.

While it may vary depending on jurisdiction or applicable licenses, cryptocurrency exchanges may generally provide staking services through the following schemes, in the order of increasing levels of risk for customers:

1) *Custody agreements*: Exchanges securely hold customer funds deposited at relatively low interest rates. This is similar to a savings account at a bank, with the principal guaranteed.
2) *Lending agreements*: Customers lend their cryptocurrencies to the exchange at pre-agreed interest rates, which are often higher than those of custody agreements. While the exchange can reinvest the principal at its discretion, the principal is repaid to customers along with the interest at maturity.
3) *Investment agreements*: Exchanges charge fixed management fees to invest customers' assets on their behalf. Customers have the potential to earn higher yields, but there is also the same chance of risk of losing principal.

However, financial institutions are often prohibited or regulated from making high-risk reinvestments using customer funds. Among them, cryptocurrency staking has become an attractive option, which may be considered to have a lower risk compared to token swapping or lending through DeFi. For example, exchanges can collect block generation fees by staking on Tezos, Avalanche, and Solana.

Ethereum is another popular PoS cryptocurrency, with the second-largest market capitalization after Bitcoin. Exchanges may need to consider the following requirements because the staked ethers are locked and staking activity comes with the certain risk of penalties or loss of funds:

- Safely and securely stake customer assets, including preventing the loss of cryptographic keys.
- Be able to adjust the amount of ethers in the exchange's reserve as liquidity, so as to promptly respond to customer withdrawal requests.
- Aim for higher yield rates to ensure profitability or to attract more deposits from customers.

To achieve these goals, exchanges need to understand Ethereum's consensus mechanisms, cryptographic key management, and wallet configurations. Since we have found a limited number of literature that covers these topics, we aim to provide an overview of each, propose the suitable architecture, and provide analysis and insight in this paper. Specifically, our contributions include:

- Explaining the fundamental challenges of staking on Ethereum and filling the gaps in the literature.
- Proposing a set of general techniques to improve Ethereum staking stability and efficiency.
- Conducting an experiment on the testnet and showing the performance through evaluation.
- Presenting newly revealed open challenges and outlining the future research direction.

Proc. 7th IEEE Int'l Conf. on Blockchain
(IEEE Blockchain 2024), August 2024

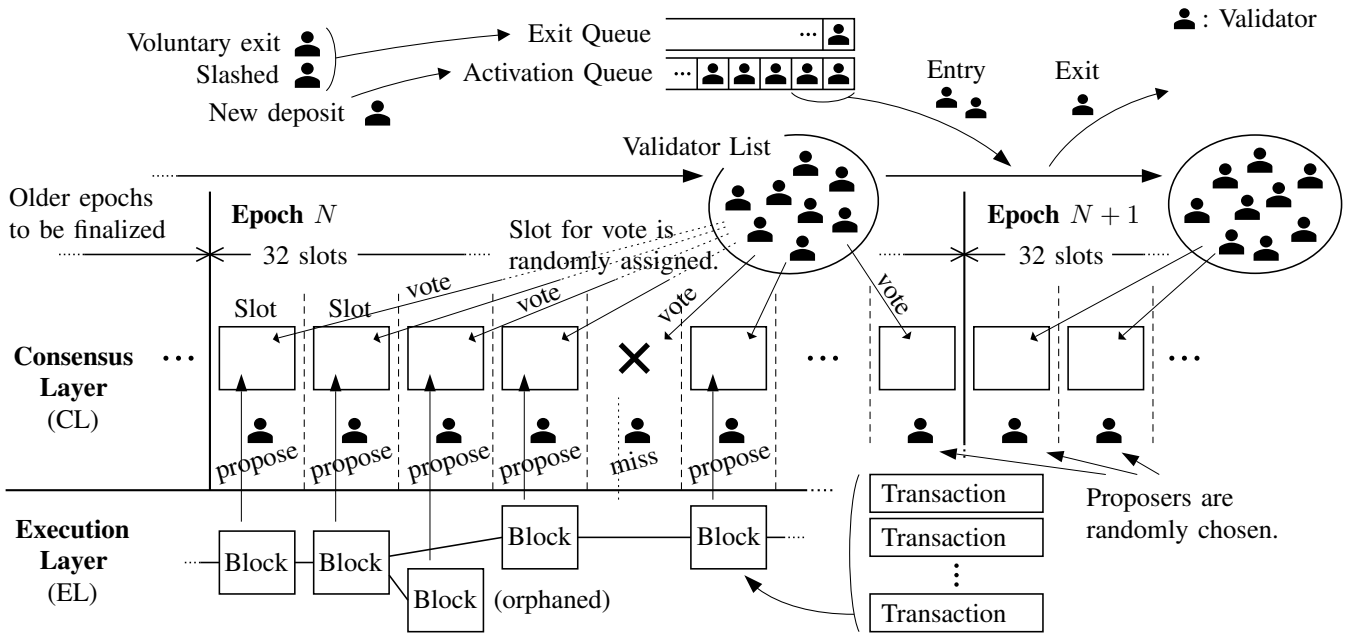

Figure 1. Overview of Ethereum Architecture

In Section 2, we will review previous research and development on consensus algorithms until the modern Ethereum. In Section 3, we will provide an explanation of the overall concept of Ethereum architecture and staking, which will serve as the foundation for the subsequent discussions. In Section 4, we will identify the challenges that exchanges may encounter during basic staking. In Section 5, we will propose several techniques to address these challenges. In Section 6, we will introduce our experimental setup for the Holesky testnet and evaluate its performance. In Section 7, we will discuss additional issues that emerged from the experiment. Lastly, in Section 8, we will summarize our proposal, highlight the takeaways from the experiment, and outline open challenges for more effective staking.

2. Related Work

Cryptocurrency transactions are typically recorded on a blockchain. In permissionless blockchains, network participants need to agree on appending records, i.e., blocks with transactions. Bano et al. provided a comprehensive overview of various consensus algorithms [1].

While voting seems like a straightforward way to reach consensus, it is susceptible to Sybil attacks [2] in a permissionless environment. To address this issue, cryptocurrencies like Bitcoin employ a proof-of-work (PoW) mechanism, where network participants, known as miners, compete by using their computational resources to maintain the blockchain. This architectural design of using a computational puzzle to authorize requests originates from Hashcash [3]. The PoW blockchains have known issues such as vulnerability to attacks like selfish mining by a party with over 50% of the entire computational power [4], [5], and their non-negligible negative impact on the environment [6]. Despite these challenges, PoW is used in Bitcoin and other cryptocurrencies, and research is actively conducted [7].

PoS has long been proposed as an improvement over PoW [8]–[10]. Early PoS algorithms adjust the difficulty of the PoW computational puzzle based on the miner's balance, i.e., *stake*, of currency held on the chain. More advanced PoS algorithms completely eliminate the need for competition based on computational resources by optimizing the distribution of block generation rights, making it a more environmentally friendly approach. Some PoS algorithms had issues, e.g., the nothing-at-stake problem [11], resulting in chain divergence, and known attacks like the long-range attack [12] by branching from an early block. To address these, several amendments to the algorithms like uncle blocks, slashing, and checkpoints have been proposed.

Different from permissionless blockchains, permissioned ones only need to distribute the voting rights to a predetermined set of participants, and Paxos [13], PBFT [14], or other well-studied algorithms have been suitable options for that. Those are extended to various protocols such as Tendermint [15] or HoneyBadgerBFT [16].

Ethereum is a cryptocurrency that has transitioned from PoW to PoS through a process called *The Merge*. It was driven by the need to solve Ethereum's scalability issues [17]. Currently, a special smart contract called the *deposit contract* is used to participate in staking. Park et al. conducted the formal verification of the deposit contract [18].

In the post-merge PoS Ethereum, the reward system has been renewed, instead of mining in PoW [19]. Among the significant rewards that a block proposer can earn is miner extractable value (MEV), which has been the subject of active research [20], [21]. Buterin et al. have introduced

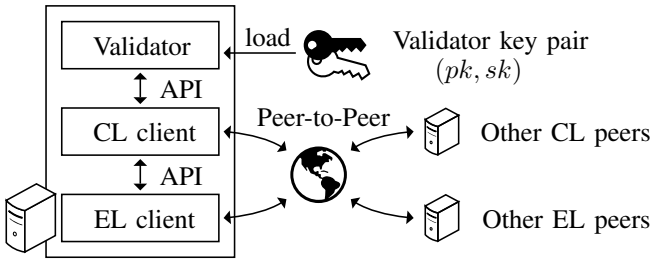

Figure 2. Minimum Software Configuration (*left*) for Staking

the idea of proposer-builder separation (PBS) to mitigate aggressive exploits, such as sandwich attacks, which target arbitrage trades [22]. Other protocol-level enhancements are being discussed through Ethereum improvement proposals (EIPs), with EIP-7251 currently receiving significant attention for its potential impact on the staking ecosystem.

3. Background on Ethereum Staking

We show a brief architectural overview of Ethereum as of 2024 in Figure 1. It is composed of two interconnected subsystems due to its historical background.

- The *Consensus Layer (CL)* is responsible for finalizing the history and providing checkpoints with the voting mechanism among a set of participants.
- The *Execution Layer (EL)* is responsible for executing transactions in blocks to transfer ethers and execute smart contracts, maintaining the state obtained by applying blocks up to the chain tip.

As the theoretical foundation, the current Ethereum adopts the PoS consensus algorithm, Gasper [23], based on the chain selection rule GHOST [24] with the LMD (latest message driven) modification, combined with the Casper FFG [25]. The CL's participants, called *validators*, are populated every epoch, which is a batch of 32 slots occurring every 12 seconds. The set of validators is constantly updated using the activation and exit queues.

As the primary benefit of staking, validators can earn rewards from mainly following two roles.

- Proposing a new EL block to the slot, if assigned.
- Attesting the proposed block for the assigned slot.

A validator may be penalized instead of receiving a reward if it fails voting, which is called an *attestation miss*. Also, to further encourage consensus soundness, multiple block proposals or multiple votes are subject to an eviction penalty called *slashing*, when reported by whistleblowers.

If a large-scale network failure occurs and leads to chain separation, the stake of inactive nodes as seen from the CL will be penalized and automatically reduced, called an *inactivity leak*, and eventually removed from the validator set. This design prioritizes availability over partition tolerance in the CAP theorem [26], where the security of the chain is compromised to a degree as analyzed by Pavloff et al. [27].

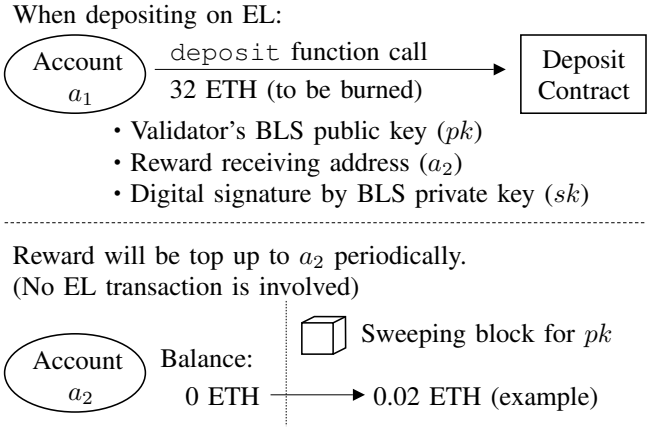

Figure 3. Deposit and Reward

In summary, there are roughly three types of penalties:

- Minor penalty for attestation misses or wrong votes.
- Major penalty and slashing for multiple proposals or votes, which violate the protocol.
- Inactivity leak in case of a network partition.

The minimum required software configuration for the validator is shown in Figure 2. CL uses BLS cryptography [28] to identify accounts, and it is necessary for the operator to generate a unique BLS key pair (pk, sk) for each validator. When the operator issues a transaction to transfer 32 ETH along with a reward receiving address on EL (a_2), the validator's public key (pk) and signature, as shown in Figure 3, the validator will be added to the activation queue and eventually be included in the validator set.

To unstake and withdraw all the validator's stake, the validator on CL broadcasts a *voluntary exit* message signed by its key (sk). Once the message is processed, the validator goes through the exit queue. The balance will eventually be accounted for in the reward receiving address (a_2) after a minimum waiting time of 256 epochs, due to the protocol design to prevent immediate exit after a slashable offense.

4. Challenges in Exchanges

As our first contribution, we identify several challenges that cryptocurrency exchanges may face as follows.

Challenge 1: Lack of Asset Liquidity. Exchanges need to hold and manage a certain amount of non-staked ethers in order to process customer withdrawal requests.

As explained previously, when exchanges perform a voluntary exit to unstake, it may take at least 27 hours until the funds are released. The staked ethers cannot be refunded immediately to the customer, and therefore, the exchange must maintain some as a reserve.

Challenge 2: Security Risks of Validator Keys. Every message on CL must be signed by the validator's key. There are two types of security risks associated with validator keys:

compromise and loss. Exchanges have to securely manage their validator keys, with particular emphasis on preventing key loss. Here are examples of potential consequences.

- *Key compromise*: A malicious attacker who steals the key can intentionally cause slashing of the validator.
- *Loss of key*: A validator can no longer continue attestation duties or unstake at worst.

Key compromise, i.e., the key being leaked or stolen, can occur due to vulnerabilities or backdoors in the software being used in Figure 2, or unauthorized actions by human operators. A malicious attacker can exploit compromised keys to engage in double voting, which results in slashing. The standard penalty is a deduction of 1 ETH deposit, with the remaining funds being locked for several weeks.

Loss of key can be caused by equipment failures or operational errors, for example. The validator would no longer sign any messages on the CL, being unable to initiate the exit at worst[1]. The validator is forcibly evicted from the CL when its balance drops to 16 ETH due to penalties, which with only attestation misses may take several decades.

These two types of risks have been long known as inherent risks associated with cryptographic keys. As suggested by Blakley [29], the countermeasures are sometimes contradictory: tighter protection and redundancy. Our earlier research provides comprehensive risk analysis on key management for exchanges [30].

Challenge 3: Stable Operation of Validators. Exchanges have the responsibility of operating secure and stable validator nodes with minimum downtime.

There are occasional upgrades to Ethereum protocols through hard forks, which require the validator, CL, and EL clients to be updated beforehand. Additionally, security vulnerabilities of the underlying system should be addressed.

If a validator experiences downtime, there will be a recurring penalty for every failed vote per epoch, i.e., approximately 7,000 gwei every 384 seconds. If a validator key is accidentally used on multiple machines during maintenance, it may result in slashing due to double voting.

Challenge 4: Increasing Revenue. While exchanges aim to maximize their revenue by staking, they may face the following circumstances that impact profitability.

- A validator may fail to effectively broadcast their attestation, causing a reduced reward or penalty.
- Profitable transactions may not exist in the mempool, resulting in less fee income from block proposals.
- The operational costs may exceed the revenue.

In addition, there is a certain wait in the activation queue before a validator starts generating profits after the deposit.

1. To avoid such a case, exchanges can presign a voluntary exit message and keep it separate from the key. The message remains valid indefinitely as per EIP-7044 after the Dencun upgrade as of March 2024.

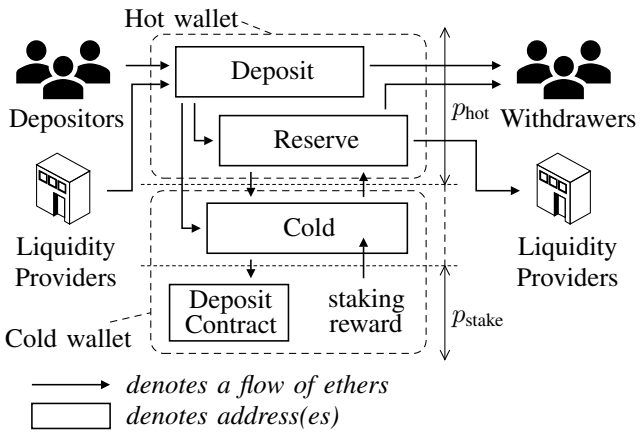

Figure 4. Wallet Configuration for Exchanges

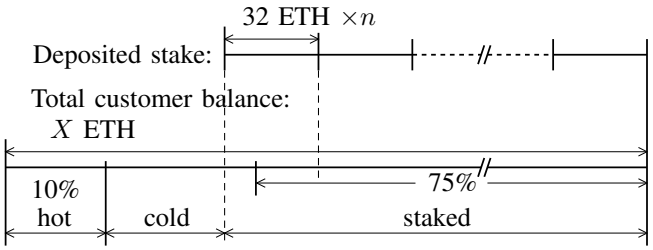

Figure 5. Allocation Example

5. Proposed Techniques

As our main contribution, we propose a set of techniques to resolve or relax the previously mentioned challenges.

5.1. Wallet Management

In Challenge 1, we mentioned the issue of limited customer asset liquidity during staking. To address this problem, we propose a wallet configuration for exchanges as shown in Figure 4. Exchanges often use cold wallets for enhanced security. In our proposal, staking is performed from a cold wallet, and the rewards are deposited back into the cold wallet. To further enhance liquidity, exchanges shall also utilize liquidity providers (LPs).

Suppose that the total amount of customer deposits is X ETH. To determine the appropriate balance of the hot and cold wallets, the exchange sets the following parameters:

- p_{hot}: The minimum reserve rate for the hot wallet.
- p_{stake}: The target rate for staking.
- T: The rebalance interval.

The amount in each wallet is adjusted every T. The number n of 32 ETH units to stake can be calculated as:

$$n = \left\lfloor \frac{X \cdot p_{\text{stake}}}{32} + 0.5 \right\rfloor \quad \text{if } X > \frac{16}{1 - (p_{\text{hot}} + p_{\text{stake}})}$$

An example of wallet allocation is shown in Figure 5 for the case where $p_{\text{hot}} = 0.1$ and $p_{\text{stake}} = 0.75$. There

are security incidents related to hot wallets, as surveyed by Oosthoek et al. [31], and it is preferable to keep p_{hot} low to improve security. Some jurisdictions set a maximum limit.

To enhance profitability, it is desirable to set p_{stake} as large as possible, while the maximum withdrawable amount without unstaking is limited to $X(1 - p_{\text{stake}})$ ETH. In practice, p_{stake} should be determined based on the average of customer withdrawal totals within T under the regular trends.

The choice of T should consider factors such as operational costs, transaction fees, ease of accounting, and business requirements. We consider $T = 24$ hours is realistic in actual operations. In the unlikely event that the hot and cold wallets are emptied for customer withdrawals, such as during sudden market movements, exchanges should rely on LPs for an additional supply of ethers.

5.2. Secure Software and Utilities

In Challenges 2, we mentioned the importance of secure management and the use of validator keys in order to prevent disruptions in validator operations.

As a holistic solution to these challenges, we suggest implementing general security measures when building the system architecture for staking. These include:

- Reviewing the software's audit reports as well as the activities of its developers and user community.
- Validating integrity by verifying digital signatures.
- Checking the vulnerabilities of software packages in dependencies and investigating whether any high CVSS cases have been reported.
- Conducting additional inspections of the source code for crucial tools and building them when necessary.

Since CL and EL codebases are typically large-scale and used in an online environment, exchanges should implement security hardening based on defense-in-depth strategies for the network and the entire system stacks.

We also recommend employing extra tools, such as slashing protection, to prevent violating signing attempts. It should be correctly configured and enabled to provide safety in scenarios where nodes are migrating to another or when recovering from failures, as examples.

5.3. Use of Cloud Environments

We also propose running validators in a cloud environment as a means to contribute to the solutions for Challenges 2 to 4. This not only has a positive effect on cost reduction through operational automation but also provides the following three benefits.

Secure validator key management. Many cloud vendors provide secret manager products or functionality that can securely store sensitive information as binary objects. Some of these functions also offer geographic redundancy.

Exchanges can keep BLS key pairs in those secret managers and prevent the loss of validator keys due to natural disasters or equipment failures.

Simplified node reconstruction. Thanks to the checkpoint feature by Gasper, nodes can be synchronized to the latest state within a few hours. This means that even if a node becomes inoperable for some reason, the node operator can quickly rebuild validators by spinning up new nodes in a cloud environment, as long as the keys are intact.

This also eases the provisioning of different types of nodes, which can help avoid failures caused by specific software [32]. In general, client diversity in a network system contributes to the resilience of the entire stability [33], which also applies to the Ethereum network.

Fast and reliable network. Communication stability is crucial for Challenge 4. It is essential to deliver a validator's own votes and proposals quickly to adjacent validators in order to maximize rewards. By hosting validators in the cloud, exchanges can benefit from fast network backbones.

5.4. MEV-boosting

It is a major goal for mass node operators, such as exchanges, to increase revenue from block proposals. As mentioned in Section 3, they can increase staking profits as MEV by optimizing transactions within blocks.

MEV-boost is software that supports off-protocol PBS and works with the validator client when proposing a block. It is still important to have a fallback mechanism in place to ensure that the validator client can always receive block proposals from the CL client in case MEV-boost fails.

5.5. Use of Staking Pool

Staking pools are DeFi platforms that accept smaller deposits from users and combine them for collective staking. Exchanges, when using staking pools, have the advantage of being able to flexibly control staking activity with higher liquidity by depositing smaller fractions of their customers' funds, as compared to a multiple of 32 ETH locked when performing regular staking, also called *solo-staking*.

Moreover, many staking pools issue an ERC-20 token known as a liquid staking derivative (LSD) in exchange for the deposit. The LSD acquired from staking pools is a valuable ERC-20 token, which can be utilized as collateral for other reinvestment purposes. This process, referred to as *restaking*, is an emerging field in DeFi. However, reinvesting LSD can significantly increase systemic risk.

6. Experiments and Evaluation

We conducted the experiment on Holesky testnet to show the effectiveness of our proposal particularly in Section 5.3.

6.1. Setup

We created two virtual machines ($\texttt{VM}_1$, $\texttt{VM}_2$) on the Microsoft Azure cloud at `D4as_v5` class (4 vCPUs, 16 GiB memory) in the East US region with a 512 GiB SSD each. The architecture detail is shown in Figure 6.

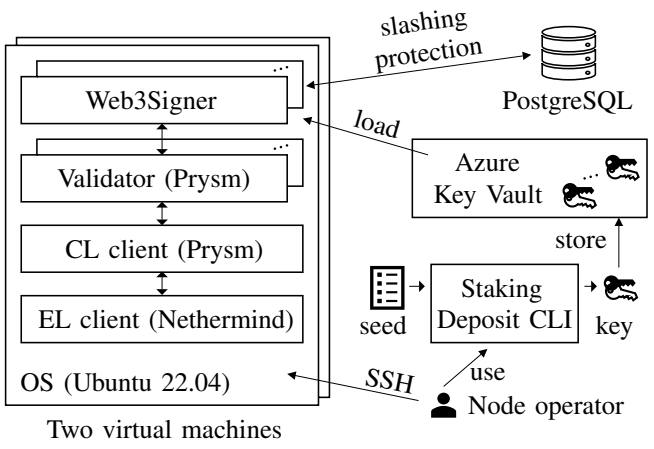

Figure 6. System Setup of our Experiment

To compare the performance differences, we configured 4 validator keys for $\texttt{VM}_1$, and 16 keys for $\texttt{VM}_2$. To ensure security, we performed the following during the setup:

- Inspected the Staking Deposit CLI's code to check the proper use of the random number generator.
- Derived keys from the seed mnemonic and uploaded the generated keys to Key Vault as per Section 5.2.
- Utilized Web3Signer for signing CL messages with a PostgreSQL backend to enable slashing protection.

Due to the project constraints of the experiment, we did not simulate the customer deposit and did not consider the wallet configuration, MEV-boost, or staking pools.

6.2. Results and Evaluation

VM Resource Usage. The summary of the collected VM metrics is presented in Table 1. We observed no significant difference in the resources usage between $\texttt{VM}_1$ and $\texttt{VM}_2$. This implies that the architecture can be scaled without being affected by the number of validators.

To provide more details on the result, we found no regular patterns in CPU utilization, and it remained stable through the entire time. The activity of the EL client, CL client, and validators did not have a major impact on CPU usage. In terms of storage consumption, we noticed that the EL client synchronized the full blockchain data (approximately 80 GiB) initially when we started, and both the EL and CL clients increased their storage consumption as we continued. Throughout the experiment, there was consistent usage of storage IO and network bandwidth.

Based on these metrics, we conclude that a standard class VM with at least 0.5 vCPU units and around 16 GiB of memory in any cloud is sufficient for solo-staking in a cloud environment, regardless of the number of validators or BLS keys hosted. Additionally, more than 1 TiB of storage may be necessary from the beginning of operation due to the size of the mainnet EL blockchain data.

TABLE 1. SUMMARY OF VM RESOURCE USAGE

Total CPU Utility (95th percentile)	0.32 vCPU (minimum) 0.84 vCPU (maximum) 0.48 vCPU (average)
Memory usage (average)	8.93 GiB (Nethermind, EL client) 2.54 GiB (Prysm, CL client) 0.04 GiB (Prysm, per each validator)
Storage used	85 GiB (Nethermind, EL client) 55 GiB (Prysm, CL client)
Total storage IO per minute (average)	170 MiB (write) 20 MiB (read)
Total network usage per minute (average)	39 MiB (incoming) 43 MiB (outgoing)

Node Migration and Updates. To examine the maintenance scenario, we rebooted both of our running VMs during the experiment conducted over a week. The procedure involved (1) OS software updates, (2) shutdown of node clients, (3) VM reboot, and (4) restarting each software in order. All steps were completed in a total of eight minutes on both VMs. Each validator resumed operation after a maximum of two attestation misses, with some validators not missing any. EL and CL clients were able to synchronize with the latest state in approximately one minute after the reboot.

To examine a node reconstruction scenario as mentioned in Section 5.3, we migrated the validators to another region, West Europe. The procedure went as: (1) we launched a new VM and completed the synchronization of EL/CL clients while the existing VM was running, (2) we shut down the existing running validators, and (3) we waited for at least one epoch to avoid duplicate votes before resuming signing on the newer VM. Eventually, there were a maximum of two attestation misses, but no slashable offenses were observed.

The first outcome suggests that node operators are able to quickly restore validator operations after a brief interruption. Our second outcome suggests that in the event of a longer interruption, it is feasible to initiate a new VM to synchronize with the most recent checkpoint.

Profitability. After the experiment, the daily cost of operating the infrastructure was 11.0 USD. Each validator made an average profit of 0.00146 ETH per day. The attestation success rate was approximately 99%, and each validator missed 1.3 attestations per day on average. As of June 2024, the mainnet had 1.0 million validators compared to the 1.5 million validators on the testnet. This could potentially result in 1.5 times higher rewards on the mainnet.

Unfortunately during this experiment, no validator was assigned the role of a proposer. On the mainnet, the current proposer reward is approximately 0.0440 ETH. If validators had an equal chance of obtaining a slot every 12 seconds, they could expect to receive this reward 2.6 times per year.

We have summarized the cost and estimated profit on the mainnet in Table 2. By using the following variables:

- Daily operational cost: x USD / VM
- Daily staking reward: y ETH / validator
- Exchange rate: z USD / ETH
- # of validators on a VM: n

TABLE 2. COST AND PROFIT BREAKDOWN (MAINNET ESTIMATE)

Cost	VM instance	4.2 USD / day
	SSD Storage	1.2 USD / day
	Network usage	5.2 USD / day
	PostgreSQL	0.4 USD / day
	Total	11.0 USD / day
Profit	Attestation reward	0.0023 ETH / validator, day
	Proposal reward	0.0440 ETH (2.6 times per year)
	Total	0.0027 ETH / validator, day

the annual yield rate A can be formulated as:

$$A = \frac{365(y - x/nz)}{32}$$

Based on the values of x and y obtained from Table 2, and assuming $n = 16$ validators running on a single VM with an exchange rate of $z = 3,300$ USD/ETH, we speculate that the annual percentage yield would be approximately 2.84%. It is worth noting that this yield could potentially be even higher if the MEV-boost was appropriately implemented and configured.

Security and Operational Implications. By benefiting from the security features provided by the cloud infrastructure, such as network security, fine-grained access control, and audit logs, we were able to rapidly build our architecture. However, we faced the necessity of developing custom-built scripts or configurations that were tailored to integrate with specific products, such as Key Vault. This may limit the portability to other platforms or on-premise environments.

From an operational perspective, we have gained insights about considering geographic distribution and utilizing multiple cloud vendors to ensure resilience. For instance, we recognized the possibility of VMs being automatically throttled or evicted when using economic VM instances, which can cause validators to stop attesting. We also realized the potential risk of losing cryptographic keys in the event of datacenter-wide failures or disasters. These findings emphasize the need to strike a balance between profitability and the expenses to maintain a reliable infrastructure.

7. Open Challenges

Through the experiment, we have identified the following open challenges that could lead to further improvement.

Reducing Attestation Misses. Despite the operational stability of our validators, we observed occasional attestation misses without any visible difference in the logs and network activity. We were unable to identify the causes during this study, but our hypothesis is that this was attributed to the connectivity or performance of neighboring CL nodes, based on our findings of higher offline rates on the testnet among non-exited validators (example below as of June 30, 2024):

- Testnet: 8.3% (126,003 offline / 1,518,332 total)
- Mainnet: 0.2% (2,339 offline / 1,027,448 total)

Although our experiment should be conducted against the mainnet, we speculate that operating more CL clients could improve the deliverability of attestations even on the testnet. We aim to study CL gossip behavior through simulation and compare stable and unstable networks.

Comparison with Other Infrastructure Options. We proposed the use of cloud environments in the operation of validators and chose the Azure cloud for our experiment. As explained, this choice was reasoned by the goal of using cloud functionality to enhance the security of validator keys and to utilize stable network and compute infrastructure at a lower operational cost. However, exchanges may have additional motivations to evaluate cost efficiency, security, stability, fault tolerance, and other factors, such as:

- To ensure system portability, it is desirable to avoid strong dependence on a specific cloud vendor.
- If the number of validators increases, on-premise operations may have relatively cost advantages.

For example, other configuration options are available from different cloud providers. GCP provides compute products specialized in blockchain nodes that the user can delegate the operation of both CL and EL clients to. AWS provides VM instances with secure enclaves to provide confidential computing capabilities, enabling the execution of Web3Signer inside the enclave to improve the security of handling BLS keys. We are also aware of non-virtualized configuration options. For example, devices like Raspberry Pi [34] or FPGA [35] can be used to run blockchain nodes.

Optimal Wallet Allocation using Mainnet Trends. We proposed allocating 70% of the entire customer funds for staking (p_{stake}) as an example. However, this value was arbitrarily chosen based on the author's experience, and no numerical analysis was given at this stage of the paper.

To determine a more appropriate allocation among wallets, statistics from actual exchange activity are essential. One can infer the historical deposit and withdrawal amounts of various exchanges by blockchain analysis. By combining this data with market trends, it may be possible to estimate the correlation between customer withdrawals and market fluctuations or deflations, thus creating a numerical model for optimal wallet allocations. Similarly, evaluating LSDs and ether prices could help determine whether solo-staking or using a staking pool is appropriate.

8. Conclusions

We reviewed solo-staking technique on Ethereum and the challenges faced by cryptocurrency exchanges. We focused on the problem of securing the operational reserve of customer deposits, ensuring the safety of validator keys, simplifying node operations, and enhancing profitability.

To address these issues, we introduced a multi-tiered wallet setup that defines the reserve amount. We also recommended several security precautions and operational strategies for cloud-based environments. Additionally, we

discussed the use of supplementary software to maximize MEV and the use of staking pools to increase profits directly.

Through our experiments, we were able to successfully operate multiple validators in a cloud environment, contributing to the overall performance on the Holesky testnet. We discovered that the architecture is scalable regardless of the number of validators. Additionally, during the experiment, we identified new challenges to further improve profitability. To achieve this, we need to refine the allocation of funds to different wallets and optimize operations. Exploring the possibility of using alternative devices may also help reduce costs. Furthermore, analyzing network behavior and other factors may be useful in reducing attestation misses.

References

[1] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Sok: Consensus in the age of blockchains," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies (AFT 2019)*.

[2] J. R. Douceur, "The Sybil attack," in *International workshop on peer-to-peer systems (IPTPS 2002)*. Springer Berlin Heidelberg, 2002, pp. 251–260.

[3] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Advances in Cryptology — CRYPTO' 92*. Springer Berlin Heidelberg, 1993, pp. 139–147.

[4] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, vol. 8437. Springer Berlin Heidelberg, 2014, pp. 436–454.

[5] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in Bitcoin," in *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2017, pp. 515–532.

[6] M. Wendl, M. H. Doan, and R. Sassen, "The environmental impact of cryptocurrencies using proof of work and proof of stake consensus algorithms: A systematic review," *Journal of Environmental Management*, vol. 326, 2023.

[7] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Čapkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 3–16.

[8] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2016, pp. 142–157.

[9] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities," *IEEE Access*, vol. 7, 2019.

[10] D. Karakostas, A. Kiayias, and M. Larangeira, "Account management in proof of stake ledgers," in *Security and Cryptography for Networks*. Cham: Springer International Publishing, 2020, pp. 3–23.

[11] W. Li, S. Andreina, J. M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Cham: Springer International Publishing, 2017, pp. 297–315.

[12] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE Access*, vol. 7, 2019.

[13] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, 1998.

[14] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *3rd Symposium on Operating Systems Design and Implementation (OSDI 99)*. New Orleans, LA: USENIX Association, Feb. 1999.

[15] J. Kwon, "Tendermint : Consensus without mining," 2014. [Online]. Available: https://tendermint.com/static/docs/tendermint.pdf

[16] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 31–42.

[17] A. Kudzin, K. Toyoda, S. Takayama, and A. Ishigame, "Scaling Ethereum 2.0s cross-shard transactions with refined data structures," *Cryptography*, vol. 6, 2022.

[18] D. Park, Y. Zhang, and G. Rosu, "End-to-end formal verification of Ethereum 2.0 deposit smart contract," in *Computer Aided Verification*. Cham: Springer International Publishing, 2020, pp. 151–164.

[19] M. Cortes-Goicoechea, T. Mohandas-Daryanani, J. L. Muñoz-Tapia, and L. Bautista-Gomez, "Autopsy of Ethereum's post-merge reward system," in *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Dubai, United Arab Emirates.

[20] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 910–927.

[21] B. Weintraub, C. F. Torres, C. Nita-Rotaru, and R. State, "A Flash(bot) in the pan: Measuring maximal extractable value in private pools," in *Proceedings of the 22nd ACM SIGCOMM Internet Measurement Conference*, 2022.

[22] V. Buterin, "Proposer/block builder separation-friendly fee market designs," Jun. 2021. [Online]. Available: https://ethresear.ch/t/proposer-block-builder-separation-friendly-fee-market-designs/9725

[23] V. Buterin, D. Hernandez, T. Kamphefner, K. Pham, Z. Qiao, D. Ryan, J. Sin, Y. Wang, and Y. X. Zhang, "Combining GHOST and Casper," 2020. [Online]. Available: https://arxiv.org/abs/2003.03052

[24] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin," in *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2015, pp. 507–527.

[25] V. Buterin and V. Griffith, "Casper the friendly finality gadget," 2019. [Online]. Available: https://arxiv.org/abs/1710.09437

[26] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *ACM SIGACT News*, vol. 33, 2002.

[27] U. Pavloff, Y. Amoussou-Genou, and S. Tucci-Piergiovanni, "Byzantine attacks exploiting penalties in Ethereum PoS," 2024. [Online]. Available: https://arxiv.org/abs/2404.16363

[28] P. S. Barreto, B. Lynn, and M. Scott, "Constructing elliptic curves with prescribed embedding degrees," in *Security in Communication Networks*. Springer Berlin Heidelberg, 2003, pp. 257–267.

[29] G. R. Blakley, "Safeguarding cryptographic keys," in *1979 International Workshop on Managing Requirements Knowledge (MARK)*, 1979, pp. 313–318.

[30] Y. Takei and K. Shudo, "Pragmatic analysis of key management for cryptocurrency custodians," in *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Dublin, Ireland.

[31] K. Oosthoek and C. Doerr, "From hodl to heist: Analysis of cyber security threats to Bitcoin exchanges," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*.

[32] A. P. Pereira, "Ethereum' s beacon chain updated after finality issues," May 2023. [Online]. Available: https://cointelegraph.com/news/ethereum-s-beacon-chain-is-updated-after-finality-issues

[33] B. Littlewood, P. Popov, and L. Strigini, "Modeling software design diversity - a review," *ACM Computing Surveys*, vol. 33, 2001.

[34] M. Cortes-Goicoechea, T. Mohandas-Daryanani, J. L. Muñoz-Tapia, and L. Bautista-Gomez, "Can we run our Ethereum nodes at home?" *IEEE Access*, 2024.

[35] J. Ktari, T. Frikha, M. Hamdi, and H. Hamam, "Enhancing blockchain consensus with FPGA: Accelerating implementation for efficiency," *IEEE Access*, 2024.