The staking mechanisms in Ethereum

Elias Jaghjough Lamrani

June 30, 2024

Resum– El mecanisme de consens d'Ethereum, *Proof-of-Stake* (PoS), es la base fonamental que composa la seva infraestructura blockchain. Els validadors comprometen una participació econòmica per validar i proposar nous blocs, obtenint recompenses per un comportament honest i afrontant penalitzacions per accions deshonestes. Aquest treball s'endinsa en un anàlisi exhaustiu dels diferents mecanismes d'*staking*, incloent *solo*, *pooled* i *Software-as-a-Service* (SaaS). L'estudi identifica meticulosament les propietats de seguretat i privadesa inherents a cada mecanisme i distingeix les suposicions de confiança dins dels seus respectius models. Avaluant aquestes metodologies, aquesta tesi pretén revelar els seus respectius punts forts i febles, oferint una visió sobre les seves implicacions en la seguretat i la privadesa de la xarxa. A més, l'estudi proposa un sistema dissenyat per millorar aquests aspectes, reduint així la dependència de la confiança en tercers.

Paraules clau– Proof-of-Work, Proof-of-Stake, Staking, Assumpcions de confiança, Node validador, Holešky, Proposta de sistema.

Abstract– The consensus mechanism of the Ethereum cryptocurrency, Proof-of-Stake (PoS), forms the foundation of its blockchain infrastructure. Validators commit an economic stake to validate and propose new blocks, earning rewards for honest behaviour and facing penalties for dishonest actions. This thesis provides a comprehensive analysis of various staking mechanisms, including solo, pooled, and Software-as-a-Service (SaaS) approaches. It meticulously identifies the security and privacy properties inherent in each mechanism and examines the trust assumptions within their respective models. By evaluating these staking methodologies, the research aims to uncover their strengths and weaknesses, shedding light on their implications for network security and privacy. Additionally, the study proposes a system designed to enhance these aspects, thereby reducing reliance on third-party trust.

Keywords– Proof-of-Work, Proof-of-Stake, Staking, Trust assumptions, Validator node, Holešky, Sistem proposal.

1 Introduction

In the evolving landscape of blockchain technology, the consensus mechanism [1] plays a crucial role in shaping the efficiency, security, and sustainability of decentralized networks. Among the diverse consensus mechanisms, Proof-of-Stake (PoS) [2] has emerged as a significant alternative to the Proof-of-Work (PoW) [3] model. In the Ethereum ecosystem, PoS stands out as a prominent force, reshaping the validation of transactions and the addition of new blocks to the blockchain.

• Contact address: eliaslamrani2002@gmail.com.
• Specialization: Information and Communication Technology (ICT).
• Supervised by Jordi Herrera Joancomarti.
• Course 2023/24.

At the heart of the PoS model lies the concept of staking, where validators commit a certain economic stake to validate and propose new blocks on the blockchain. Unlike PoW, which relies on computational work and energy-intensive mining, PoS leverages validator's economic incentives to ensure network security and integrity. This fundamental shift not only enhances energy efficiency but also lowers barriers to entry, fostering greater inclusivity and participation within the network.

Despite its potential benefits, the PoS consensus mechanism poses unique challenges and considerations. One of such areas of focus lies in the diverse array of staking mechanisms employed within PoS networks. From solo staking [4] to pooled [5] and even Staking-as-a-Service (SaaS) solutions [6], the landscape of staking methodologies is rich and vast. Each approach brings its own set of advantages and trade offs, needing a deep understanding of their secu-

rity, privacy, and trust implications.

In view of these complexities, this thesis embarks on a comprehensive exploration of the staking mechanisms, aiming to unravel their hidden implications. By meticulously analyzing the security and privacy properties inherent in each staking approach, this research aims to clarify their respective strengths and weaknesses. Additionally, it seeks to explain the intrinsic trust assumptions embedded within these models, providing valuable insights into their impact on the network.

Beyond analysis, a validator node will be implemented with a focus on enhancing its security measures. By leveraging the findings, the study designs and implements a system to improve security and privacy levels in PoS networks. These efforts aim to reduce reliance on third-party trust and enhance the robustness of decentralized networks, paving the way for a safer and more trustworthy blockchain ecosystem.

2 Objectives

This thesis focuses on analyzing Ethereum staking mechanisms to enhance security and privacy while minimizing third-party trust. To achieve this, specific goals are outlined:

- **Understanding PoS:** Examine its principles, operational dynamics, and implications for security, privacy and decentralization.
- **Investigate staking mechanisms:** Explore solo, pooled, and SaaS staking methodologies, evaluating their functionalities, advantages, and drawbacks.
- **Analyze security and privacy properties of staking mechanisms:** Identify potential vulnerabilities, evaluate security features, and analyze trust assumptions of different staking methods.
- **Implement validator nodes on the testnet blockchain:** Deploy validator nodes on a testnet, selecting mechanisms based on hardware requirements and privacy measures, incorporating identified security considerations.
- **Propose a system designed to reduce third-party trust:** Develop a system that enhances user convenience and asset security.
- **Evaluate practical security assumptions in Ethereum P2P network:** Assess and validate security assumptions within the test blockchain's peer-to-peer network.

3 State of Art

Ethereum's transition from Proof-of-Work (PoW) to Proof-of-Stake (PoS) [7] signifies a pivotal moment in its evolution, driven by the dual objectives of reducing energy consumption and enhancing decentralization. The latter is particularly crucial for maintaining the blockchain's integrity and resilience against censorship. However, while PoS offers numerous advantages over PoW, it also presents certain drawbacks.

In contrast to PoW, where miners solve computational puzzles, PoS requires validators to stake (lock) the native cryptocurrency token, necessitating a higher level of technical expertise to set up a validator node. This transition has introduced additional security concerns at the consensus layer, including the concept of maximal extractable value (MEV) [8], which refers to the maximum value extractable through transaction manipulation within a block. The switch to PoS could potentially exacerbate vulnerabilities, such as the sandwiching attack, wherein malicious actors exploit validators' flexibility in block creation to manipulate transaction ordering for personal gain, potentially harming unsuspecting users.

Nevertheless, PoS brings significant benefits, notably in energy efficiency by eliminating the need for the energy-intensive computations inherent in PoW. Additionally, PoS reduces hardware requirements, making participation more accessible and fostering a more distributed network. Moreover, it introduces economic penalties for dishonest behavior, making 51% style attacks [9] more costly for potential attackers compared to PoW systems. While validators are required to invest 32 Eth, this requirement also incentivizes honest behavior due to the fear of losing the deposited funds.

By significantly reducing energy consumption compared to its predecessor, PoS aligns with the growing focus on environmental responsibility. Additionally, PoS lowers barriers to entry by eliminating the need for specialized hardware, allowing for broader participation in securing the network. Moreover, PoS incentivizes honest behavior through economic penalties, further enhancing network security. By embracing this innovative approach, Ethereum sets a positive precedent for the future of blockchain technology, paving the way for a more sustainable, inclusive, and secure ecosystem.

PoS aligns with the increasing emphasis on environmental responsibility by significantly reducing energy consumption compared to its predecessor. Despite some technical complexities, the overall benefits of PoS, such as fostering greater decentralization and resilience, outweigh these challenges. Furthermore, PoS incentivizes honest behavior through economic penalties, enhancing network security. Ethereum's embrace of this innovative approach sets a positive precedent for the future of blockchain technology, paving the way for a more sustainable, inclusive, and secure ecosystem.

4 Methodology and Planning

Among the various methodologies available, adopting agile methodologies has proven advantageous due to their iterative, incremental, and evolutionary traits. The Kanban [10] methodology was ultimately selected for its effectiveness in managing individual projects.

Kanban is a visual management system used to optimize workflow in projects by representing tasks on boards segmented into columns: *To Do*, *In Progress*, *Done*, and *Blocked*. The *To Do* column lists tasks awaiting initiation, the *In Progress* column contains tasks under development, the *Done* column includes completed tasks, and the *Blocked* column identifies tasks that cannot proceed due to dependencies or issues. Each task is represented as a card that

transitions between columns to reflect its status, typically including a description, title, and priority level.

Kanban boards are often constructed using project management platforms like Jira [11]. This software allows users to create and customize Kanban boards to meet project-specific needs, contributing to its selection for this thesis.

The thesis has been structured into two to four-week sprints, encompassing research and practical sections. Tasks derived from main objectives are tracked using a Gantt chart and reviewed at each sprint's conclusion. Given a 14-week timeframe, parallelizing tasks in the research section is feasible. The thesis timeline is organized as follows:

1. Research Section (**6 weeks**)
   (a) Explore proof of stake: **2 weeks**.
   (b) Investigate staking mechanisms: **2 weeks**.
   (c) Examine the strengths and weaknesses of each staking mechanism: **2 weeks**.
   (d) Examine the security and privacy assumptions: **2 weeks**.
2. Practical Section (**8 weeks**)
   (a) Examine validator setup process: **2 weeks**.
   (b) Deploy validator nodes: **4 weeks**.
   (c) Create a voluntary exit message in advance: **2 weeks**.
   (d) Analyze the testnet network: **2 weeks**.

Refer to the next figure for a detailed view of the Gantt chart.

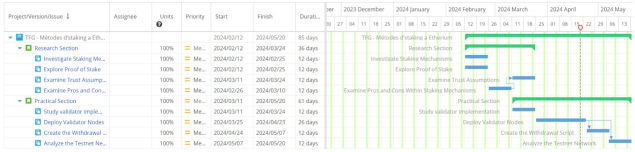

Fig. 1: This Gantt chart shows task duration in workdays.

5 Proof of Stake

In Proof of Stake (PoS), validators stake a portion of their cryptocurrency as collateral, incentivizing honesty through potential loss of holdings if they behave dishonestly. Ethereum's operation now comprises two layers: the *execution layer*, handling transaction validation and execution similar to the previous PoW protocol, and the *consensus layer*, built upon the Beacon Chain [12]. This consensus layer aims to achieve agreement among validators, who verify the validity of newly generated blocks and may occasionally create and propagate new blocks themselves.

5.1 Validators

Becoming a validator entails depositing 32 ETH into the deposit contract and configuring three distinct software components: an execution client, a consensus client, and a validator client. Upon depositing their ETH, the user enters an activation queue, which controls the rate at which new validators can join the network. Before delving further, it is crucial to clarify the concept of time within the Ethereum network. Unlike PoW, where block timing depends on mining difficulty, PoS operates at a fixed tempo. Time in Ethereum is divided into slots, each lasting 12 seconds, and epochs, consisting of 32 slots (approximately 6.4 minutes). Once the validator joins the network, they are tasked with three primary responsibilities [13]:

- **Block proposal:** In each time slot within the Ethereum PoS system, the *RANDAO* algorithm [14] randomly chooses a validator to propose the next block. This selection process helps deter manipulation attempts by malicious actors. It is important to note that a slot may or may not contain a block.
- **Block attestation:** Attestation [15] entails validators verifying the validity and accuracy of data within a block. They periodically confirm the state of the latest validated block (head of the beacon chain) to ensure network consensus. Validators are organized into committees of 128 for fairness and network efficiency. While a new committee is randomly chosen for each slot, all active validators participate in attesting to the latest validated block once per epoch, rather than every individual block. Moreover, to optimize network bandwidth and enhance scalability, some validators occasionally aggregate attestations from others, consolidating multiple confirmations into a single message.
- **Sync committee participation:** Sync committees [16], consisting of validators, produce signatures to authenticate the state of the beacon chain. These signatures enable light clients to efficiently ascertain the latest validated block without needing to download the entire blockchain.

Validators primarily aim to earn economic rewards in the form of ETH. To accomplish this, they withdraw the ETH staked in the deposit contract, commonly known as staking withdrawal [17]. Any excess reward payments beyond the 32 ETH threshold are automatically sent to a specified withdrawal address linked to each validator. Users define this address during the setup of the validator node. Additionally, users can opt to exit staking entirely, unlocking their entire validator balance for withdrawal. Furthermore, users seeking to exit the staking process must initiate it by signing and broadcasting a *voluntary exit message* using their validator keys. This message, facilitated through the validator client and submitted to the consensus node, eliminates the need for validators to manually submit a transaction to withdraw a specific amount of ETH, thus avoiding gas fees. The duration of the validator exit process varies depending on the number of validators exiting simultaneously. Once the exit process is complete, the account will no longer have validator responsibilities.

5.2 Consensus Security

Unlike PoW, which relies on miners, PoS entrusts validators with maintaining network integrity. Therefore, robust security mechanisms are crucial for safeguarding Ethereum

against various threats and building trust among users. Byzantine Fault Tolerance (BFT) plays a pivotal role in securing Ethereum's PoS system, enabling it to withstand Byzantine failures, where nodes exhibit malicious or unpredictable behavior. BFT achieves this by incentivizing honest behavior and penalizing malicious activity through a reward and penalty system.

Validators earn consensus layer rewards for proposing blocks, attesting to their validity, participating in sync committees, and reporting misbehavior [18]. Additionally, block proposers receive execution layer rewards, including optional priority fees and direct user tips. However, validators also face penalties for engaging in dishonest behavior, which fall into two main categories:

- **Slashing:** This penalty occurs when a validator proposes conflicting blocks in the same slot, attests to contradictory blocks within the same epoch, or attempts to *vote against history* by casting a vote surrounded by a previous one. The severity of slashing, measured by the amount of ETH destroyed, depends on the number of validators being slashed simultaneously, known as a *correlation penalty*. This deters coordinated attacks by making them financially unattractive.
- **Inactivity:** Inactive validators face lesser penalties proportional to their offline duration. This distinction acknowledges that inactivity is often due to factors like power outages or hardware failures, rather than intentional network harm.

6 Staking Methods

6.1 Solo Staking

In the realm of staking, solo staking embodies self-reliance. It entails running an Ethereum node and committing a significant economic stake to act as a validator, thereby directly enhancing the chain's security and integrity. While providing extensive control and potential rewards, solo staking also presents distinct advantages and disadvantages.

6.1.1 Advantages

1. **Enhanced decentralization:** Solo staking fortifies the network by decentralizing validation power to individual nodes, reducing dependence on centralized entities. This enhances censorship resistance and strengthens defenses against attacks.
2. **Complete rewards:** Solo stakers earn rewards directly from the protocol, bypassing intermediaries and potentially capturing higher returns compared to other methods.
3. **Transparency and control:** This method guarantees complete control over the validator and its operations, ensuring full transparency and alignment with personal preferences.
4. **Strengthened privacy:** Because of its decentralized nature, solo staking ensures that only the individual participant holds their private keys, guaranteeing exclusive control over their staked assets.
5. **Increased security:** Enhanced security against unauthorized access by eliminating the reliance on centralized servers. In contrast, other mechanisms that involve storing funds on centralized servers increase vulnerability to potential security breaches.
6. **Increased client diversity:** Most validators predominantly choose clients like Geth and Prysm, but an increasing number are opting for minority clients like Erigon and Besu. This trend aims to diversify client usage, promoting a more secure environment.

6.1.2 Drawbacks

1. **Economic barrier:** The 32 ETH deposit requirement represents a significant financial hurdle, limiting accessibility for many potential participants.
2. **Technical complexity and ability:** Setting up, maintaining, and securing a validator node demands advanced technical knowledge, including expertise in blockchain protocols, software configuration, and network security.
3. **Hardware limitations:** Running a validator node demands a dedicated computer with specific hardware requirements, which can be costly and impractical for some.
4. **Strict maintenance:** Unlike other methods, solo stakers are solely responsible for maintaining the uptime and functionality of their node around the clock. Neglecting this responsibility can result in penalties and loss of rewards.
5. **Slashing risk:** Malicious or negligent behavior, even unintentional, can result in slashing. Even poor internet connection can result in this penalty.

6.1.3 Trust Assumption

Solo staking, beckons with the promise of high rewards and unparalleled autonomy. But venturing down this path requires traversing a web of trust assumptions, each posing its own unique challenge.

The first layer of trust lies with the chosen wallet, where the staker's digital assets are stored. Secure Element (SE) wallets provide added security, but individuals bear full responsibility for safeguarding validator keys. Any lapse in judgment or unforeseen vulnerability could result in significant financial losses, underscoring the importance of robust security measures and constant vigilance.

Further down the path, reliance on open-source client software is another trust assumption. These programs, categorized into execution and consensus clients, are essential for the staking process. While thorough audits and a vibrant development community provide some reassurance, the potential for bugs or vulnerabilities requires trust in both the software itself and the teams behind it.

The anonymity offered by blockchain technology presents a nuanced challenge in the context of solo staking, introducing a degree of trust. While directly linking individuals to their staking activities based on on-chain data might

be difficult, there are ways to potentially gather information that could compromise privacy. Validator nodes communicate via peer-to-peer protocols, which may involve exchanging IP addresses for tasks like data synchronization and block proposals. This exposure raises the risk of attacks like the theorized *validator sniping attack* proposed by the HOPR team [19]. This attack involves launching a short-burst Denial-of-Service (DoS) attack against a validator node just before it proposes a block. A successful attack could diminish validator effectiveness and increase attacker rewards, as they could potentially propose the next block themselves and claim the block reward. While solutions like VPNs or TOR can obfuscate IP addresses and enhance privacy, they come with a trade-off. These tools can introduce latency or connection issues, potentially leading to missed block proposals and attestations, reducing the overall effectiveness of a validator.

Weak subjectivity [20], or block synchronization, introduces trusted checkpoints in blockchain networks. Unlike systems where every node downloads the entire history, weak subjectivity relies on checkpoints, trusted snapshots of the blockchain at specific points. When new or returning nodes join the network, they fetch the latest checkpoint and rely on other nodes for missing blocks since then, assuming the checkpoint reflects a valid state. While this speeds up onboarding, it involves trust assumptions. However, the risk of receiving a faulty checkpoint is minimal, as it can be cross-verified against independent sources like block explorers or multiple nodes. Despite cryptographic verification ensuring data integrity, there is still reliance on the honesty of others.

While self-managed validator software offers control and potentially lower costs, some users prefer Virtual Private Servers (VPS) for staking. VPS providers offer preconfigured environments tailored for staking, simplifying setup and potentially improving uptime compared to self-managed hardware. This option is especially appealing for users less comfortable with technical aspects of validator operation or unwilling to invest in hardware. However, choosing a VPS for staking entails important considerations. Entrusting staking keys to a third party introduces trust, as the risk of penalties for bad behavior depends on the provider's infrastructure, security practices, and the VPS's performance.

6.2 Non-custodial staking

Non-custodial staking services such as Stakefish [21], Allnodes [22] and Staked [23] provide the infrastructure for staking, empowering users to maintain control over their assets. These services manage validator nodes using the staking keys, while users directly engage with smart contracts to stake their assets using their withdrawal keys. However, it is important to note that users typically still need to invest 32 ETH to stake with these services.

6.2.1 Advantages

1. **Enhanced user convenience:** These platforms handle validator software, infrastructure, and network interaction for users, streamlining the staking process and eliminating the need for technical expertise, dedicated hardware, and ongoing maintenance.
2. **Diverse providers options:** Users have a wide array of staking services to choose from, each offering unique features and benefits. When selecting a staking service, users should consider key attributes like whether it is open-source, audited for security, battle-tested in real-world scenarios, offers multiple client options, and retains custody of validator credentials.
3. **Moderate security for user assets:** Since withdrawal keys are exclusively held by the user, the service lacks the capability to misappropriate any ETH deposited in the smart contract.

6.2.2 Drawbacks

1. **Fees:** SaaS platforms usually charge fees for their services, deducted from the staking rewards users earn. The specific fees can vary depending on the platform.
2. **Limited security risks:** While users do not relinquish any assets to the service, the service retains the ability to slash the deposited ETH because it possesses the staking keys.
3. **Limited Control:** These services deprive users of direct control over their validator keys and the manner in which their stake interacts with the network.
4. **Centralisation risk:** Users place their trust in the staking provider, which decreases decentralisation.
5. **Minimum technical knowledge required:** Users engaging in non-custodial staking services must deposit the minimum required amount to delegate validator responsibilities, requiring a certain level of technical knowledge. Additionally, users should understand how to securely store their private keys in a wallet.

6.2.3 Trust Assumption

The adoption of staking services introduces complex trust dynamics between users and providers. This section examines the critical trust assumptions inherent in these services, delving into the trade-offs between convenience and control.

In non-custodial services, users delegate validator responsibilities while retaining possession of their withdrawal keys and, consequently, control over their funds. While this setup enhances user trust to some extent, there remains the potential for the service to engage in dishonest behavior, leading to slashing penalties. As a result, users must exercise a degree of trust in the service provider, although it is reduced compared to custodial arrangements.

Regardless of the service type, whether custodial or non-custodial, users must trust the security measures and practices of the service provider. The integrity of the platform's infrastructure, encryption protocols, and adherence to security best practices becomes crucial. A breach in the service provider's security could result in catastrophic consequences, including the slashing of users staked assets. Thus, users must carefully assess the reputation, track record, and security protocols of staking providers before entrusting them with their valuable assets.

6.3 Pooled Staking

Staking pools serve as collaborative networks of cryptocurrency holders who pool their assets to engage in the staking process collectively. This allows smaller investors to participate in staking without needing a significant amount of cryptocurrency individually. In terms of reward distribution, each participant's remuneration is based on their contribution to the staking pool. Typically, staking pools appoint operators responsible for utilizing the pooled funds for native staking and distributing rewards proportionally to each participant's contribution. For instance, some platforms may opt for a more centralized approach with a single entity serving as the operator, while others may adopt a decentralized model involving decentralized autonomous organizations (DAOs) and self-executing smart contracts. Additionally, staking pools such as Lido [24] and Rocket Pool [25] operate as staking pools governed by on-chain communities. Lido benefits from the contribution of 30 carefully vetted companies offering staking services to the protocol. In contrast, Rocket Pool boasts a network of over 2,500 permissionless node operators entrusted with staking user funds.

6.3.1 Advantages

1. **Accessibility for small investors:** Pooled staking allows investors with smaller holdings to participate in staking. By combining their assets in a pool, they can meet the minimum stake requirements that might be too high for individual staking.

2. **Enhanced convenience:** Staking pools handle the technical aspects of staking, such as validator selection and maintenance. This simplifies the process for users who do not want to manage the technical complexities themselves.

3. **Reward distribution:** Staking pools distribute the staking rewards among all pool members proportionally to their delegated stake, after deducting a fee for the pool operator.

4. **Diversification:** Token holders can delegate their stake to multiple pools, diversifying their risk and potentially increasing their rewards.

5. **Option to serve as a Staking Pool Operator:** Platforms such as *Rocket Pool* provide the opportunity for individuals or entities to act as pool operators. These operators manage a pool of combined holdings contributed by users to participate in staking processes, earning commissions from other users only if the node operates smoothly.

6.3.2 Drawbacks

1. **Centralization:** Its centralized structure contradicts the decentralized purpose of PoS.

2. **Reduced rewards:** It also offer investors the opportunity to earn rewards without the responsibility of managing validator nodes. However, this convenience comes at a cost; fees deducted from earned rewards typically result in lower returns compared to individual staking.

3. **Lack of control:** Pools enforce rules governing the withdrawal of assets, restricting investors' autonomy over their stakes. This lack of control becomes particularly concerning during market volatility, when investors may need flexibility to manage risks.

4. **Security risks:** Staking pools face various security threats, including potential delays or freezes on withdrawals. Furthermore, vulnerabilities in the pool's infrastructure could be exploited by malicious actors, endangering investors' assets.

6.3.3 Trust Assumptions

Pooled staking offers an attractive entry point for users to participate in staking. Unlike solo staking, which requires a significant minimum stake and technical expertise, pooled staking allows participation with smaller contributions. However, this convenience comes with a layer of complexity: trust in the staking pool operator. While these operators can be facilitators who streamline staking, the possibility of misplaced trust exists.

Accessibility is a key benefit of this staking mechanism, lowering the barrier for participation. Yet, users must place trust in the operator managing their funds. While operators manage the staking process, ensuring secure storage and wallet management of delegated funds is paramount. Expertise in maintaining validator nodes is essential, including timely software updates to mitigate vulnerabilities, proactive monitoring for suspicious activity, and regular maintenance to prevent slashing risks. Some operators enhance security by geographically distributing validator nodes to mitigate localized threats.

However, the role of the operator also introduces the risk of malicious behavior. While most operators act in good faith, there is a potential risk of them exploiting this trust. Malicious actors could disappear with delegated funds in an exit scam, manipulate validator configurations to steal rewards, trigger slashing penalties, or distribute rewards unfairly. Vigilance and thorough vetting of operators are crucial to mitigate these risks.

6.4 Custodial staking

Custodial services like Binance [26], Coinbase [27], and Kraken [28] provide convenience by handling the technical complexities of staking. However, this convenience comes at the cost of users relinquishing control over their private keys and entrusting their assets to the service provider. While these platforms simplify the staking process, users must place trust in the provider's security measures and policies. Additionally, custodial staking services often involve fees for their management and security services.

6.4.1 Advantages

1. **Accessibility for small investors:** Custodial staking services often enable participation with smaller amounts of cryptocurrency compared to the conventional stake requirement.

2. **Enhanced user convenience:** These platforms manage the validator software, infrastructure, and network interaction on behalf of the user. This streamlines the staking process and eliminates the need for technical expertise, dedicated hardware and ongoing maintenance.

3. **Diverse providers options:** When choosing a service, it is essential to assess several attributes, as outlined in the non-custodial service section.

6.4.2 Drawbacks

1. **Fees:** SaaS platforms typically charge fees for their services, which are deducted from the staking rewards users receive. These fees may vary depending on the platform.

2. **Security risks:** In custodial staking, users cede control of their cryptocurrency to the platform. Choosing a reputable provider with strong security measures is crucial to mitigate the risk of fund loss due to security breaches.

3. **Limited Control:** Users using these services do not retain direct control over their validator keys or the management of their stake.

4. **Centralization Risk:** By relying on a staking provider, users contribute to centralization risks in the network.

6.4.3 Trust Assumption

The adoption of staking services introduces nuanced trust dynamics between users and service providers. Specifically, the custodial alternative emerges as the one with the most significant trust assumptions.

Custodial providers are distinguished by assuming control of and safeguarding customers assets throughout the staking process. In essence, these services manage both the user's staking and withdrawal keys. Therefore, users entrust their assets to the custody of the service provider, exposing themselves to potential risks such as mismanagement, insider threats, or external breaches. The custodial service's possession of withdrawal keys grants them considerable power, potentially enabling unauthorized access to user's funds. Moreover, control over the staking keys opens avenues for malicious activities that could result in slashing penalties. Overall, users place complete trust in the service; in the event of malicious activity, there is little the user can do. It is precisely for this reason that custodial services entail the most significant trust assumptions.

After gaining a deep understanding of the staking mechanisms, the following table provides a concise overview of their respective advantages and disadvantages.

	Solo	Non-custodial	Pooled	Custodial
Decentralization	✓	×	×	×
Complete rewards	✓	✓	×	×
Control	✓	∼	×	×
Convenience	×	✓	✓	✓
No Economic barrier	×	×	✓	✓
No Fees	✓	×	×	×
Enhanced Security	✓	∼	∼	×

TABLE 1: COMPARISON OF STAKING MECHANISMS

7 VALDIATOR SET-UP

7.1 Previous Considerations and Implementation

Before setting up a validator node, several considerations must be addressed: selecting appropriate hardware, determining the testnet to use, and choosing the execution and consensus client software.

Hardware requirements increase over time as the database size grows weekly, which varies by execution client. A basic setup initially demands decent hardware equipment. In this regard, a Virtual Private Server (VPS) with 12 vCPU cores, 48GB RAM, and 2TB SSD storage was rented.

Multiple testnets complement Ethereum's mainnet, providing developers with sandbox environments to experiment and validate innovations before live deployment:

- **Sepolia:** The recommended default testnet for Ethereum application development [29].
- **Goerli:** Once crucial for infrastructure, staking, and protocol updates, Goerli [30] now faces scalability issues, including a shortage of native testnet tokens and a limited number of validators. It has been deprecated and is no longer maintained.
- **Holešky:** A robust alternative to Goerli, supporting over 1.4 million validators and offering over one billion testnet ETH, ensuring an abundant and uninterrupted testing environment [31].
- **Rinkeby / Ropsten:** Both testnets are currently sunsetted [32].

Moreover, a full validator node consists of two clients:

- *Execution client*, such as Geth or Nethermind, are responsible for executing smart contracts and maintaining the Ethereum state, essentially mirroring the entire network's data.
- *Consensus clients*, such as Lighthouse or Prysm, handle the consensus mechanism, verifying transactions and securing the network through PoS.

Careful evaluation of execution and consensus client selection is crucial. It is important to avoid *supermajority clients*, defined as one used by over two-thirds of all active validators. These clients can independently finalize the chain, posing a supermajority risk [33]. This scenario creates a single point of failure, where any bug could lead to severe consequences such as network splits, slashing risks, and chain disruption. Geth may currently fall into this category, as shown in Figure 7 in the Appendix.

Furthermore, it is crucial to consider additional factors: the RAM usage of each client, minimum required database size, database growth rate, synchronization time, and the size of the community supporting each client. Geth is identified as a supermajority client and is therefore excluded from consideration. According to Figure 5 in the Appendix, Besu requires significant RAM usage, despite the dedicated hardware planned for implementing the validator exceeds these requirements. Furthermore, Erigon's beta status raises stability concerns. By contrast, Nethermind boasts a large user community, and although its database grows rapidly, this challenge could be mitigated with synchronization checkpoints. Hence, Nethermind [35] emerges as the preferred choice. Regarding consensus clients (Figure 6 in the Appendix), a minority client is also preferred. While theoretically any client could be used, Lighthouse [36], developed in Rust with an emphasis on security and efficiency, has been selected.

Due to space constraints, a detailed explanation of the validator setup process is not possible here. Instead, a comprehensive walkthrough has been created. For those interested in setting up an Ethereum validator node independently, please visit [38].

7.2 Results

After depositing the 32 ETH into the deposit contract, the validator must await confirmation and transition to the pending state, which can take 16 to 24 hours initially and extend for several weeks. Once this period ends, the validator node will engage in attestation and block proposal activities, earning ETH rewards over time. When a predefined threshold is reached, excess ETH beyond the initial 32 will be automatically withdrawn to the designated address. For an illustration of a completed validator setup, please refer to Figure 2.

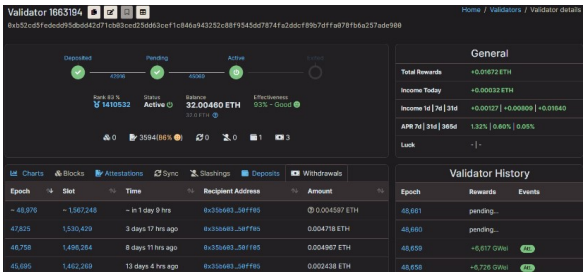

Fig. 2: An active validator's rewards metrics [39].

After setting up the validator, it is recommended to install Grafana and Prometheus. For more information, visit Figure 8 in the Appendix.

8 System Proposal

Throughout the thesis, it is clear that setting up a validator is the most secure option and offers significant benefits, but it lacks user convenience. Developing a system that prioritizes security, minimizes reliance on third parties, and enhances user convenience would be compelling.

Such a system could resemble a Staking-as-a-Service platform with a few modifications (refer to Figure 4). Clients should generate their own staking keys to ensure trust and security, especially since inputting the withdrawal address is a critical step in the process. Tools like staking-deposit-cli or Wagyu GUI produce two files: *keystore.json*, which contains the validator's private key, and *deposit_data.json*, which includes the validator's public key and staking details.

Once the keys are generated, the user deposits 32 ETH via the Ethereum Launchpad. After confirming the deposit, the user sends the staking keys (*keystore.json* and *deposit_data.json*) to the staking service, which uses them to establish the validator node. The user can monitor the validator's status and profitability through a blockchain explorer like beaconcha.in. Subsequently, the user should delete the staking keys, particularly the *keystore.json* file, to avoid slashing penalties for reusing the keys in another validator. At this point, three main situations could occur:

1. **The staking service operates with integrity:** In this scenario, the service adheres to correct validator behavior and will execute the withdrawal process upon request.

2. **The staking service disappears:** Although the validator remains operational, the service fails to facilitate the withdrawal process upon request.

3. **The staking service acts maliciously:** Due to the system's infrastructure, the service cannot steal funds directly. However, with the staking keys, the service can initiate slashing activities. There is an incentive for this: the slashing whistleblower reward [40], which gives up to 0.0625 ETH for evidence of slashing (1/8 for the proposer, 7/8 for the attester). Thus, a service operating multiple validators could profit from slashing activities despite the reputational cost.

The system still needs a mechanism to withdraw the user's ETH in the second and third scenarios. In the third situation, where the user incurs a slashing penalty, no action is necessary. When a validator is slashed, it transitions automatically to the exited state, signaling the completion of its operations. However, if the validator has been slashed, it incurs two additional penalties. Firstly, the duration spent in the exited state before funds can be withdrawn is extended to 36 days. Secondly, halfway through this period, the validator faces an additional deduction based on the number of other validators slashed in the surrounding 18 days. If the service operates many validators, the potential amount of ETH slashed could be considerable due to the correlation penalty, making this third situation a bottleneck in the system. For a comprehensive overview of the validator's lifecycle, refer to Figure 3.

For the second option, the user can prepare a voluntary exit message in advance using ethdo [42], a tool for managing wallets and accounts, and interacting with Ethereum consensus nodes. This process involves creating two interdependent files: *offline-preparation.json* and *exit-operation.json*. The creation of exit-operation.json requires offline-preparation.json, which necessitates a synced beacon node connection. Although a default mainnet offline-preparation.json can be generated without a synced beacon node, this is not possible for testnets. For a detailed walkthrough, consult [43], and refer to Figure 9 in the Appendix

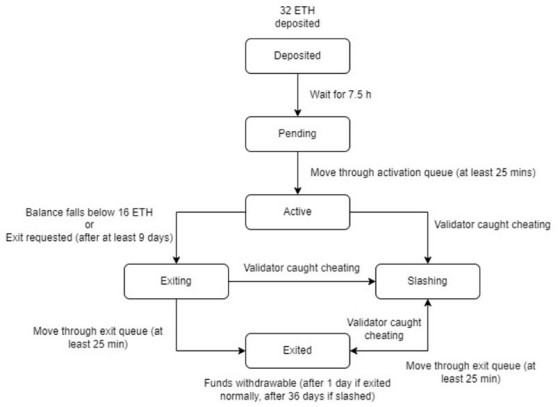

Fig. 3: Validator's lifecycle [41].

to view a validator after the exit message has been submitted.

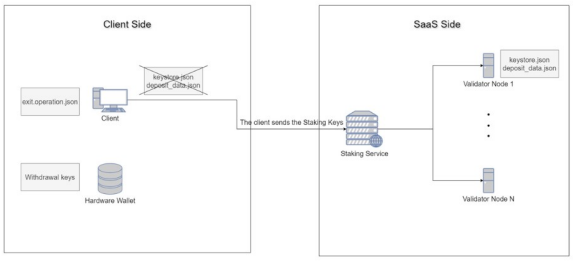

Fig. 4: The service stakes for the user while the user holds the withdrawal message.

9 Conclusions

The primary objectives of this thesis have been achieved. he Proof-of-Stake consensus mechanism and its impact on Ethereum's blockchain have been comprehensively understood. Security and privacy aspects of staking mechanisms were analyzed, assessing their advantages, disadvantages, and trust assumptions. A hardened staking node was deployed on the Holešky testnet, and a staking service proposal was made to diminish reliance on third parties. However, during the execution of the final objective, it was found to lack substantive information and was rather generic. Furthermore, this objective largely overlaps with the third objective, which evaluates deep security assumptions within the P2P network.

Efforts were made to automate the exit message generation for voluntary exits using a script, but this was hindered by dependencies on the beacon node. Addressing this challenge remains a potential area for future exploration and resolution with continued study and time. While MEV-Boosting [37] has the potential to enhance validator node performance, the primary focus was on effectively hardening the node. Thus, this project is feasible for a computer engineer with access to online resources.

The proposed staking system shows promise as a future standard for staking services. However, it must simplify the technical requirements for users, who currently face challenges in creating their own keys and withdrawal messages. These issues can be tackled over time with community support and further development efforts.

Acknowledgments

I am deeply grateful to my thesis tutor, Jordi Herrera Joancomarti, whose guidance has been fundamental in the development of this project. He provided essential advice on structuring my thesis and managing my time effectively, while also directing me to valuable resources for my research. I would also like to extend my thanks to the Ethstaker community, which has supported me by addressing doubts encountered during the setup of the validator node.

References

[1] "Consensus mechanisms." Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/developers/docs/consensus-mechanisms

[2] "Proof-of-stake (PoS)." Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos

[3] "Proof-of-work (PoW)." Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/developers/docs/consensus-mechanisms/pow

[4] "Solo stake your ETH." Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/staking/solo#what-is-solo-staking

[5] "Pooled staking." Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/staking/pools#what-are-staking-pools

[6] "Staking as a service." Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/staking/saas

[7] S. Lin, "Proof of Work vs. Proof of Stake in Cryptocurrency", Highlights in Science, Engineering and Technology, vol. 39, pp. 953-961, April 2023.

[8] G. Damalas and P. Ambrus, "An introduction to maximal extractable value on Ethereum", EY, March 2023.

[9] "51% Attack." Accessed: Jun. 28, 2024. [Online]. Available: https://academy.binance.com/en/glossary/51-percent-attack

[10] "Kanban." Accessed: Jun. 28, 2024. [Online]. Available: https://es.wikipedia.org/wiki/Kanban

[11] "Jira — Software de seguimiento de proyectos e incidencias." Accessed: Jun. 28, 2024. [Online]. Available: https://www.atlassian.com/es/software/jira

[12] "The Beacon Chain." Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/en/roadmap/beacon-chain/#what-is-the-beacon-chain

[13] ”Ethereum Proof-of-Stake Consensus Layer: Participation and Decentralization” in the 5th Workshop on Coordination of Decentralized Finance (CoDecFin), Willemstad, Curaçao, March 2024.

[14] “Block Proposal.” Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/es/developers/docs/consensus-mechanisms/pos/block-proposal/

[15] “Attestations.” Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/attestations/

[16] “Sync Committee Protocol.” Accessed: Jun. 28, 2024. [Online]. Available: https://docs.telepathy.xyz/telepathy-protocol/sync-committees

[17] “Staking withdrawals.” Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/staking/withdrawals

[18] “Recompensas y penalizaciones de la prueba de participación.” Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/es/developers/docs/consensus-mechanisms/pos/rewards-and-penalties/

[19] Dr. Sebastian Bürgel. ”Proof of Stake Validator Sniping Research”. [Online]. Available: https://medium.com/hoprnet/proof-of-stake-validator-sniping-research-8670c4a88a1c

[20] ”Subjetividad débil.” Accessed: Jun. 28, 2024. [Online]. Available: https://ethereum.org/es/developers/docs/consensus-mechanisms/pos/weak-subjectivity/

[21] “Stakefish.” Accessed: Jun. 28, 2024. [Online]. Available: https://stake.fish

[22] ”Allnodes.” Accessed: Jun. 28, 2024. [Online]. Available: https://www.allnodes.com

[23] ”Staked.” Accessed: Jun. 28, 2024. [Online]. Available: https://staked.us

[24] ”Lido.” Accessed: Jun. 28, 2024. [Online]. Available: https://lido.fi

[25] ”Rocket Pool.” Accessed: Jun. 28, 2024. [Online]. Available: https://rocketpool.net

[26] ”Binance.” Accessed: Jun. 28, 2024. [Online]. Available: https://www.binance.com/es

[27] ”Coinbase.” Accessed: Jun. 28, 2024. [Online]. Available: https://www.coinbase.com/es/

[28] ”Kraken.” Accessed: Jun. 28, 2024. [Online]. Available: https://www.kraken.com/es-es

[29] ”eth-clients/sepolia.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/eth-clients/sepolia

[30] ”eth-clients/goerli.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/eth-clients/goerli

[31] ”eth-clients/holesky.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/eth-clients/holesky

[32] ”Ethereum’s Testnets Explained [2024] Holešky, Goerli, Sepolia, and More.” Accessed: Jun. 28, 2024. [Online]. Available: https://www.ankr.com/blog/ethereum-testnets-ultimate-guide/

[33] ”The Ethereum Supermajority Risk.” Accessed: Jun. 28, 2024. [Online]. Available: https://supermajority.info

[34] ”Client Diversity.” Accessed: Jun. 28, 2024. [Online]. Available: https://clientdiversity.org/#distribution

[35] “NethermindEth/nethermind.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/NethermindEth/nethermind

[36] “sigp/lighthouse.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/sigp/lighthouse

[37] ”How to setup a validator for Ethereum staking on testnet HOLESKY.” Accessed: Jun. 28, 2024. [Online]. Available: https://www.coincashew.com/coins/overview-eth/testnet-holesky-validator

[38] ”Not-Elias/stakingWalkthrough.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/Not-Elias/stakingWalkthrough/blob/main/README.md

[39] ”Open Source Ethereum Blockchain Explorer - beaconcha.in.” Accessed: Jun. 28, 2024. [Online]. Available: https://beaconcha.in

[40] A. Yang, “A Comprehensive Study of Ethereum Staking Cryptoeconomics”, January 2024.

[41] ”Understanding the Validator Lifecycle.” Accessed: Jun. 28, 2024. [Online]. Available: https://www.attestant.io/posts/understanding-the-validator-lifecycle/

[42] ”wealdtech/ethdo.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/wealdtech/ethdo/tree/master

[43] ”Not-Elias/stakingWalkthrough.” Accessed: Jun. 28, 2024. [Online]. Available: https://github.com/Not-Elias/stakingWalkthrough/blob/main/withdrawalProcess.md

APPENDIX

Client	RAM Use	Minimum Database Size	Database Growth	Time to sync
Nethermind	8 GB	1.2 TB	30 GB/week	Fastest, 4hrs+
Besu	12 GB	1.2 TB	10 GB/week	Medium, 16hrs+
Geth	8 GB	1.2 TB	8 GB/week	Fast, 8hrs+
Erigon	8 GB	1.2 TB	8GB/week	Medium, 16hrs+
Reth	10 GB	1.2 TB	4GB/week	Medium, 16hrs+

Fig. 5: Comparison of execution clients [37]

Client	CPU Use	RAM Use	Database Size	Time to sync head
Lighthouse	Medium	6 GB	120 - 150 GB	Instant via checkpoint
Lodestar	Medium	8 GB	120 - 150 GB	Instant via checkpoint
Teku	Medium	10 GB	120 - 150 GB	Instant via checkpoint
Nimbus	Low	3 GB	120 - 150 GB	Instant via checkpoint
Prysm	Medium	6 GB	120 - 150 GB	Instant via checkpoint

Fig. 6: Comparison of consensus clients [37].

Fig. 7: This figure shows the client usage percentages [34].

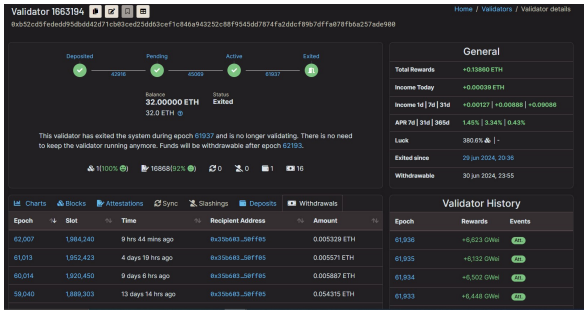

Fig. 9: Validator status following submission of the voluntary exit message.

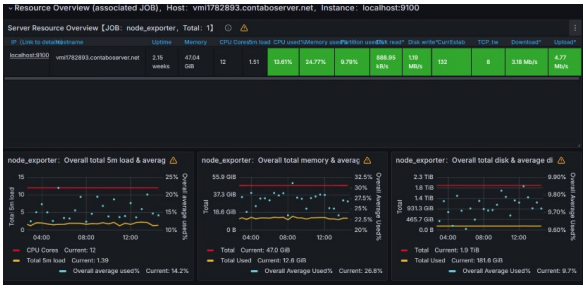

Fig. 8: Prometheus gathers metrics from monitored targets by scraping metrics HTTP endpoints, while Grafana provides a dashboard for visualizing the collected data. Downloading three dashboard files is advisable: one for monitoring the execution client, another for the consensus client, and one for general system monitoring. This figure shows a server resource overview confirming proper functioning of all components.