International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 2, Mar-Apr-2025, ISSN (Online): 2395-566X

Token Stacking and Earn Reward System

Shubham Gogdani, Harmit Vagadiya, Utsav Padaliya, Dhrumil Karena, Mukesh Patidar
Parul University Vadodara, India

Abstract- This paper introduces a novel approach to decentralized finance (DeFi) through a Token Stacking and Earn Reward System. By utilizing blockchain technology and smart contracts, this system allows users to stack tokens to earn rewards securely and transparently. The process is fully automated, eliminating intermediaries, which improves both efficiency and trust. This research focuses on the architecture, mechanisms, and advantages of such a system, highlighting the significant role of decentralized technologies in the evolution of financial systems.

Index Terms- Token stacking, blockchain, decentralized finance, smart contracts, reward systems, DeFi, transparency, automation.

I. INTRODUCTION

Blockchain has transformed various industries, offering transparency, security, and decentralization. In the financial sector, decentralized finance (DeFi) has emerged as a disruptive model, enabling financial services without intermediaries like banks. DeFi systems are built on blockchain technology, where transactions are governed by smart contracts that automate processes such as lending, borrowing, and staking.

The Token Stacking and Earn Reward System leverages DeFi's principles to allow users to lock (or stack) their tokens and earn rewards in return. The system eliminates centralized control, instead using smart contracts that automate the process and ensure the system operates in a transparent and secure manner. This paper aims to explore the system's design and implementation, as well as the technological and financial impact of blockchain in DeFi reward mechanisms.

II. BACKGROUND

1. Blockchain Technology

Blockchain is a decentralized and distributed ledger technology, where each transaction is recorded in a block and linked together to form a chain. Blockchain's key characteristics include immutability, transparency, and security. These features make it ideal for financial applications like token stacking, where trust and accuracy are crucial. In this system, blockchain ensures that all token transactions, staking operations, and rewards are recorded and verified without the need for intermediaries.

2. Decentralized Finance (DeFi)

DeFi refers to a set of financial services built on blockchain, designed to function without central authorities. Services like lending, borrowing, trading, and earning interest are facilitated by smart contracts—self-executing programs on the blockchain that enforce the terms of agreements automatically. In DeFi, users can engage in financial transactions without intermediaries like banks. In this system, DeFi principles allow users to stack their tokens in a decentralized manner while earning rewards transparently.

Immutability

Immutability refers to the property of a blockchain that prevents any alteration of data once it has been recorded. This feature is critical for maintaining the integrity of the ledger.

Mechanism of Immutability

Each block in the blockchain contains a unique cryptographic hash that links it to the previous block. This chain of hashes means that if any block is altered, the hash changes, disrupting the entire chain. As a result, the consensus of the network must be reached to approve any changes, making unauthorized alterations highly difficult.

Practical Implications

This characteristic ensures that all transactions are permanent and can be audited at any time, providing a reliable history of activities. In financial applications, this promotes confidence among users, as they can trust that the transaction history is accurate and tamper-proof.

Transparency

Transparency is another fundamental characteristic of blockchain technology. In a blockchain network, all participants have access to the same version of the ledger, enabling them to view all transactions that have occurred.

Public vs. Private Blockchains

In public blockchains (like Bitcoin), anyone can view the transaction history, fostering an open environment.

Conversely, private blockchains restrict access to authorized users, but still maintain a level of transparency among those participants.

Impact on Trust
This shared visibility allows users to independently verify transactions, thereby reducing the likelihood of fraud. The transparent nature of blockchain can significantly enhance accountability, as all actions are recorded and can be traced back to their origin.

Security
Security in blockchain systems is achieved through various means, including cryptographic techniques and decentralized architecture.

Cryptographic Protection
Each transaction is secured using cryptographic algorithms, which ensure that data remains confidential and protected from unauthorized access. Additionally, the decentralized nature of blockchain means there is no single point of failure; attacks would need to compromise multiple nodes simultaneously, which is extremely challenging.

Consensus Mechanisms
Mechanisms like Proof of Work (PoW) and Proof of Stake (PoS) play a critical role in validating transactions and adding them to the blockchain. These mechanisms help ensure that only legitimate transactions are processed, further enhancing the overall security of the network.

3. Smart Contracts
Smart contracts are self-executing contracts with the terms of the agreement written into lines of code. They are designed to automatically enforce, verify, and execute transactions. In this project, smart contracts govern the process of staking tokens, calculating rewards, and distributing the earnings once the staking period ends. This removes the need for trust between users and intermediaries, as the contract itself enforces all terms transparently and automatically.

Key Features of Smart Contracts
Decentralized Execution
Smart contracts function on decentralized blockchain networks, ensuring that no single entity controls the contract or its execution. This decentralization enhances the security and integrity of transactions, making it difficult for any party to manipulate or alter the contract after deployment.

Automatic Execution
One of the most significant advantages of smart contracts is their ability to execute automatically once certain predefined conditions are met. For instance, in the Token Stacking and Earn Reward System, once a user locks their tokens in the smart contract, the system will automatically initiate the reward calculation process at the end of the staking period without any manual input.

Self-Verification
Smart contracts possess built-in mechanisms to verify compliance with the terms of the agreement. They can assess whether the stipulated conditions have been satisfied before executing any actions. For example, the contract can verify that the required staking period has elapsed before processing reward distributions.

Transparency and Immutability
All interactions with smart contracts are recorded on the blockchain, creating a transparent and immutable ledger of all transactions. This feature allows all parties involved to track their transactions and verify the contract's actions, thus fostering trust and accountability. Once a smart contract is deployed, its code cannot be altered without consensus, ensuring that the terms remain consistent and reliable over time.

Functionality in the Token Stacking System
In the Token Stacking and Earn Reward System, smart contracts are integral to managing the staking process and ensuring fair and automated reward distribution. The following outlines how smart contracts function within this framework:

Token Locking Process
When users choose to stake their tokens, they interact with a specific smart contract designed to manage the staking process. The contract securely locks the tokens for a predetermined period, preventing any access or transfer until the staking duration concludes. This mechanism not only secures the tokens but also contributes to liquidity in the ecosystem.

Reward Calculation Mechanism
Smart contracts include algorithms that automatically calculate the rewards based on various criteria. These criteria may encompass:

- **Amount of Tokens Staked:** Users who stake more tokens can receive a higher proportion of rewards.
- **Staking Duration:** Longer staking periods often result in increased rewards, incentivizing users to commit their tokens for extended durations.
- **Market Factors:** The contract can be programmed to adjust reward rates based on the overall performance of the staking pool or prevailing market conditions, creating a dynamic reward system that adapts to external factors.

Automated Reward Distribution
At the conclusion of the staking period, the smart contract automatically executes the reward distribution process. This means that users receive their original staked tokens along

with any accrued rewards directly in their wallets without any delays or manual intervention. This efficiency improves user satisfaction and encourages continued participation in the staking program.

Auditability and Verification
Each action taken by the smart contract is recorded on the blockchain, resulting in a transparent and verifiable record of all transactions. Users can review their staking history and rewards at any time, enhancing accountability and trust in the system. This level of transparency is crucial in financial applications where users need assurance that the process is fair and reliable.

Advantages of Implementing Smart Contracts
Cost Reduction
By removing intermediaries from the transaction process, smart contracts significantly lower operational costs. Users can benefit from increased returns on their investments due to the absence of fees typically associated with traditional financial institutions.

Increased Efficiency
Smart contracts enable faster transaction processing. The automated execution of transactions eliminates the delays that often occur with manual processing, allowing for real-time updates and interactions within the ecosystem.

Trust and Reliability
The transparent nature of smart contracts instills confidence among users. Participants can engage with the system knowing that the terms of their agreements will be executed as programmed, without the risk of fraud or manipulation.

Innovative Financial Solutions
The programmability of smart contracts allows for the creation of sophisticated financial products and mechanisms. For instance, the Token Stacking and Earn Reward System can implement tiered reward structures or offer bonuses for users who stake for longer periods, creating a more engaging and rewarding user experience.

Challenges and Considerations
While smart contracts offer numerous advantages, there are also challenges that need to be addressed:

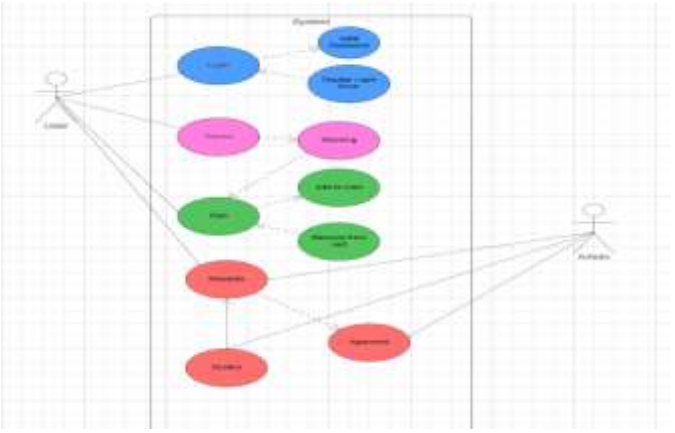
Vulnerabilities in Code
The effectiveness of smart contracts hinges on the quality of their code. Bugs or vulnerabilities can lead to exploits, resulting in potential financial losses for users. Comprehensive testing and audits are essential to identify and rectify any issues before deployment.

Regulatory Concerns
The legal landscape surrounding smart contracts and DeFi is still developing. Uncertainties regarding compliance with existing regulations can pose risks to developers and users. Staying informed about regulatory changes and ensuring adherence to applicable laws is crucial for the long-term viability of the project.

User Education and Accessibility
Many potential users may not fully understand how smart contracts operate or the implications of engaging with them. Providing educational resources and support can help demystify the technology, leading to broader adoption and increased participation in the Token Stacking and Earn Reward System.

III. SYSTEM ARCHITECTURE

1. Token Stacking Mechanism
In the Token Stacking and Earn Reward System, users can lock or stack their tokens into a smart contract for a specified period. The tokens are held securely within the smart contract and cannot be accessed until the staking period is over. This stacking mechanism encourages users to hold tokens, which helps maintain stability in token supply and demand within the ecosystem.



2. Reward Calculation Engine
The reward calculation engine is an integral part of the system, implemented through the smart contract. This engine determines the rewards based on the number of tokens staked, the duration of staking, and other factors such as the current supply-demand dynamics of the token. The calculations are performed automatically, ensuring transparency and fairness in reward distribution.

3. Reward Distribution
Upon the completion of the staking period, the smart contract distributes the initial token amount along with the calculated rewards to the user's wallet. This process is automated and

does not require any manual intervention, ensuring that all users receive their rewards promptly and without discrepancies.

IV. IMPLEMENTATION

1. Token Contract

The system uses an ERC-20 token contract on the Ethereum blockchain, which follows the widely adopted ERC-20 standard. The token contract handles the locking and unlocking of tokens, staking operations, and reward distribution. It ensures that the tokens remain locked in the contract until the staking period ends.

Key Features of the Token Contract

Standardized Token Protocol (ERC-20)

The ERC-20 standard provides a blueprint for creating fungible tokens on the Ethereum blockchain. By adhering to this standard, the Token Contract ensures compatibility with various wallets, exchanges, and decentralized applications (dApps), thereby enhancing its usability and accessibility. The ERC-20 specification includes essential functions such as:

- Transfer: Facilitates the transfer of tokens between users.
- BalanceOf: Allows users to check their token balance.
- Approve and TransferFrom: Enables users to approve third-party contracts to spend tokens on their behalf, which is useful in scenarios where tokens are needed for staking or other applications.

Minting and Burning Mechanisms

The Token Contract may also include functions for minting new tokens and burning existing tokens. Minting allows the contract owner or designated addresses to create additional tokens, which can be used for various purposes, such as incentivizing early adopters or rewarding users who participate in the staking program. Conversely, burning tokens reduces the total supply, which can help stabilize the token's value and combat inflation.

Tokenomics

The design of the Token Contract includes carefully crafted tokenomics that dictate how tokens will be distributed, utilized, and managed within the ecosystem. Key components of tokenomics may include:

- **Total Supply:** The maximum number of tokens that will ever be created. This cap can influence the perceived value of the token and help control inflation.
- **Initial Distribution:** A strategy for allocating tokens during the initial launch, which may involve a public sale, private sale, or distribution to early supporters and stakeholders.
- **Incentives for Staking:** The contract may outline how rewards are calculated and distributed to users who lock their tokens for staking, thereby encouraging participation in the system.

Figure 1: Wallet selection

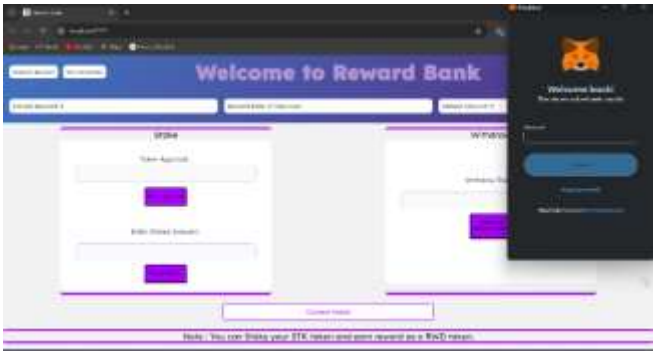


Figure 2: Connect MetaMask wallet

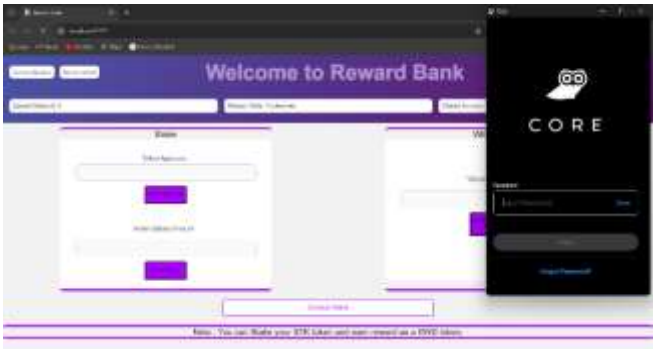


Figure 3: Connect Core wallet


Figure 4: Connected wallet

Figure 5: Earned rewards

Functions of the Token Contract

Token Transfer Functions

The Token Contract implements functions that allow users to transfer tokens seamlessly. These functions are crucial for facilitating transactions within the platform, enabling users to send and receive tokens when engaging in staking or trading activities. The contract ensures that all transfers are validated, and that balances are updated accordingly.

Staking Interface

The Token Contract includes a specialized interface for staking functionalities. When users wish to stake their tokens, they interact with this part of the contract, which manages the locking of tokens and tracks each user's staked balance. This interface ensures that tokens remain locked for the specified duration and prevents any premature withdrawals.

Reward Calculation

Integrated within the Token Contract is the logic for calculating and distributing rewards based on the staking parameters. This includes:

- **Accruing Rewards:** The contract tracks the amount of tokens staked by each user and calculates rewards based on the duration and total staked amount.
- **Automatic Distribution:** Once the staking period concludes, the contract automatically distributes the earned rewards back to the users' wallets. This seamless process eliminates the need for manual calculations or interventions.

Security Considerations

Security is paramount in the design and implementation of the Token Contract. Several measures are taken to ensure its robustness:

Auditing and Testing

Before deployment, the Token Contract undergoes extensive testing and security audits to identify and rectify any vulnerabilities. This process is crucial to ensure that the contract functions as intended and that users' funds are secure.

Access Control

The contract may implement access control measures to restrict certain functions to authorized addresses, such as the owner or governance entities. This prevents unauthorized access and protects the integrity of the tokenomics.

Error Handling

The contract includes error handling mechanisms to manage potential issues that may arise during token transfers, staking, or reward calculations. This ensures that users receive appropriate feedback and that the contract remains operational even in unexpected scenarios.

Staking Smart Contract

The staking smart contract is responsible for recording the staked tokens, enforcing the locking period, and calculating rewards. The contract ensures that tokens are securely locked and cannot be transferred until the staking conditions are met. Once the staking period ends, the contract automatically calculates the rewards and distributes them.

Reward Distribution and Gas Efficiency

On Ethereum, transactions incur gas fees, which vary depending on network congestion. The staking contract is designed to minimize gas fees during the reward calculation and distribution processes. This optimization ensures that the system remains cost-effective for users, even during periods of high network activity.

Mechanisms of Reward Distribution

Automated Reward Calculation

The system utilizes smart contracts to automate the reward calculation process, ensuring that it is both transparent and efficient. The calculation typically depends on several factors:

- **Staking Duration:** The length of time tokens are staked plays a crucial role in determining the rewards. Users who commit their tokens for extended periods generally earn higher rewards, thereby encouraging longer participation.
- **Amount Staked:** The total number of tokens a user stakes also affects their rewards. Users staking larger amounts may receive proportionally greater rewards, which serves as an incentive for higher investment.
- **Dynamic Adjustment of Rewards:** The smart contract can be designed to adjust reward rates based on various factors such as network performance, total value locked in staking, or market conditions. This flexibility allows the system to adapt to changing circumstances and maintain user interest.

Timely and Efficient Distribution

Once the staking period concludes, the smart contract facilitates the immediate distribution of rewards to users' wallets. This automated mechanism enhances user experience by:

- **Instant Access to Earnings:** Users receive their rewards promptly, reinforcing trust in the system. Immediate distribution eliminates delays often associated with manual processing.
- **Batch Processing of Distributions:** To optimize efficiency, the system can process multiple reward distributions in a single transaction. This approach minimizes the number of transactions required, enhancing overall system efficiency.

Gas Efficiency Considerations

Gas fees represent a significant concern in blockchain environments, especially on networks like Ethereum, where transaction costs can vary widely based on network congestion. Managing gas efficiency is critical for ensuring that the staking and reward distribution system remains cost-effective for users.

Optimizing Smart Contract Design

The architecture of the smart contract plays a crucial role in minimizing gas costs. Key strategies include:

- **Efficient Code Practices:** Developing efficient algorithms and data structures within the smart contract can reduce the amount of computational work required, thereby lowering gas fees. Simplifying logic and avoiding unnecessary computations are essential considerations.
- **Modular Functions:** Structuring the contract with modular functions allows for reuse of code, thereby decreasing redundancy and optimizing performance. This modularity can help reduce gas costs for common operations.

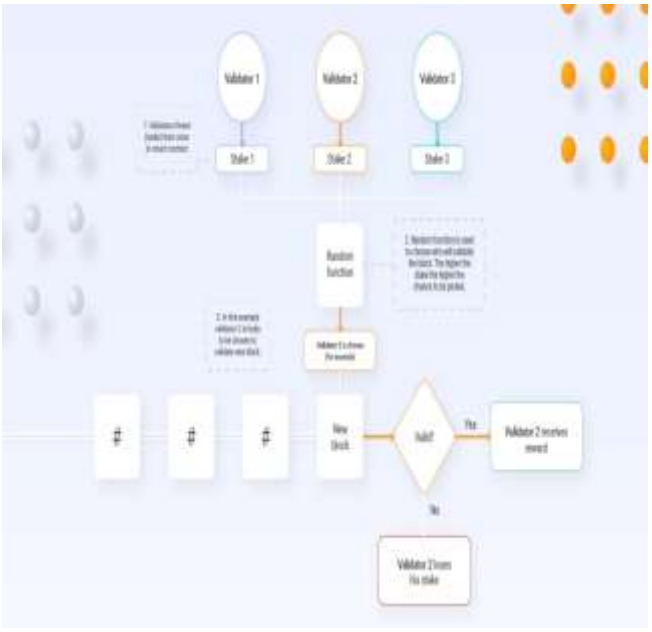
Batching Transactions

Implementing transaction batching can significantly improve gas efficiency:

- **Collective Reward Distributions:** Instead of executing individual transactions for each user, the system can aggregate rewards and distribute them collectively. This method reduces the total number of on-chain transactions, which helps lower overall gas costs.
- **Scheduled Distributions:** Timing reward distributions during periods of lower network congestion can help minimize gas fees. By analyzing network traffic and strategically timing distributions, the system can save costs and enhance the user experience.

User Awareness of Gas Costs

Educating users about gas fees and how to manage them can significantly enhance their engagement with the staking platform. Providing resources and guidance on optimal times for staking and withdrawing can help users avoid high transaction costs, leading to a more satisfying experience.



V. SECURITY AND EFFICIENCY CONSIDERATIONS

1. Security

Smart contract security is a top priority in decentralized finance systems due to the large amount of value handled by these contracts. The Token Stacking and Earn Reward System undergoes rigorous security audits, including formal verification methods to ensure the correctness of the contract's code. Additionally, the system employs mechanisms like time-locks and multi-signature wallets to enhance security further.

2. Efficiency

The system is designed to be efficient by automating the staking and reward distribution process, eliminating intermediaries, and reducing manual intervention. However, the use of Ethereum for staking comes with challenges such as high gas fees, especially during times of network congestion. To address scalability concerns, the system can explore Layer-2 solutions like Optimistic Rollups, which offer lower gas fees and faster transaction processing.

VI. RESULTS AND ANALYSIS

1. Test Deployment

The system was deployed on an Ethereum test network (Ropsten) for initial evaluations. Users were able to stack tokens, and rewards were distributed automatically based on the conditions set in the smart contract. The test network provided valuable insights into the performance of the system, particularly in terms of transparency and automation.

2. Key Observations

- **Transparency:** All staking transactions and reward distributions were recorded on the blockchain, allowing for full transparency.
- **Automation:** The system successfully automated the entire staking and reward distribution process, reducing human intervention.
- **Gas Fees:** While the system functioned as expected, gas fees on Ethereum posed a challenge, especially for smaller transactions.

VII. CONCLUSION

The Token Stacking and Earn Reward System demonstrates the potential of combining blockchain technology and DeFi principles to create a transparent, efficient, and secure financial solution. By automating the staking and reward distribution processes, the system enhances user trust and reduces operational costs. Future research will focus on improving scalability and exploring alternative blockchain networks or Layer-2 solutions to make the system more accessible.

REFERENCES

1. Iyer, V. (2022). An Examination of Stacks Protocol Performance Compared to Other LayerOne Blockchains.
2. Freni, P., Ferro, E., Moncada, R. (2022). Tokenomics and blockchain tokens: A designoriented morphological framework. Blockchain: Research and Applications, 3(1), 100069.
3. Batog, B., Boca, L., Johnson, N. (2023). Scalable reward distribution on the ethereum blockchain. Working paper. http://batog. info/papers/scala ble-reward-distr ibuti on. pdf. Accessed 12 August.
4. McLaughlin, T. F., Williams, R. L. (1988). The token economy. In Handbook of behavior therapy in education (pp. 469-487). Boston, MA: Springer US.
5. Falligant, J. M. (2018). Preference for Accumulated and Distributed Token Exchange-Production Schedules: Effects of Task Difficulty and Variable Token-Production Schedules (Doctoral dissertation, Auburn University).
6. Hangen, M. M. (2020). An Assessment of Token Value and Effectiveness (Doctoral dissertation, University of Kansas).
7. Argueta, T. (2022). Establishing and Evaluating Tokens as Conditioned Reinforcers (Doctoral dissertation, University of Florida).
8. Naderi, H., Shojaei, A., Ly, R. (2023). Autonomous construction safety incentive mechanism using blockchain-enabled tokens and vision-based techniques. Automation in Construction, 153, 104959.
9. Reddy, K. R., Farhad, S., Kotti, J., Sonule, V., Muniyandy, E., Verma, A. (2024). IoT-Driven Big Data Analytic to Automate Blockchain Adaptation. International Journal of Intelligent Systems and Applications in Engineering, 12(14s), 600-608. 46 CHAPTER 9. CONCLUSION
10. Bhatia, A., Giometti, R., Nicolas, A. (2018). Decentralized social news platform (pp. 1- 52). Tech. Rep.
11. Singh, S. DECENTRALIZED FINANCE (DEFI): EXPLORING THE ROLE OF BLOCKCHAIN AND CRYPTOCURRENCY IN FINANCIAL ECOSYSTEMS.
12. Kumar, V. S., Lee, J. J., Hu, Q. (2023). Non-fungible token-based health record marketplace. IET Blockchain.
13. Li, S. N., Spychiger, F., Tessone, C. J. (2023). Reward distribution in proof-of-stake protocols: A trade-off between inclusion and fairness. IEEE Access, 11, 134136-134145.
14. Direr, A., Doursat, R., Laurent, B., Biton, D. (2022). A Data-Driven and Principled Approach to Designing the Tokenomics of a New Blockchain-Based Game. Available at SSRN 4187993.
15. Lesavre, L., Varin, P., Yaga, D. (2021). Token Design and Management Overview.
16. Rossello, R. (2024). Blockholders and strategic voting in DAOs' governance. Available at SSRN 4706759.
17. Ali Khan, I. (2021). Blockchain technology and its implementation with a staking application in an Ethereum network.
18. Hall, O. J., Shiaeles, S., Li, F. (2024). A Study of Ethereum's Transition from Proof-of-Work to Proof-of-Stake in Preventing Smart Contracts Criminal Activities. Network, 4(1), 33-47.