

Towards Proxy Staking Accounts Based on NFTs in Ethereum

Author Name

Affiliation

Affiliation

City and Country

email

Abstract—Blockchain is a technology that is often used to share data and assets. However, in the decentralized ecosystem, blockchain-based systems can be utilized to share information and assets without the traditional barriers associated with solo responsibility, e.g., multi-sig wallets. This paper describes an innovative approach to blockchain networks based on a non-fungible token that behaves as an account (NFTAA). The key novelty of this article is using NFTAA to leverage the unique properties of NFTs to manage your ownership better and effectively isolate them to improve the security, transparency, and even interoperability possibilities. Additionally, the account-based solution gives us the ability and flexibility to cover regular use cases such as staking and liquid equities, but also practical composability. This article offers a simple implementation, which allows developers and researchers to choose the best solution for their needs in demand of abstract representation in any use-case.

Index Terms—Ethereum, solidity, NFT, liquidity, proxy accounts

I. INTRODUCTION

When Ethereum transitioned to the Proof-of-Stake consensus, it was a huge signal for the ecosystem [1]. This switch from the Proof-of-Work (PoW) consensus mechanism is being done for several reasons. Firstly, PoS is more energy efficient and environmentally friendly than PoW [2]. It also allows for faster transaction processing and improved scalability. PoS allows for more decentralized decision-making, as the power to validate transactions is distributed among multiple “mining” nodes rather than concentrated in a few large mining pools [3].

To ensure correct protocol mechanisms, asset staking was invented. Staking refers to holding and committing assets as collateral to participate in the validation of transactions and the production of new blocks. Stakers, also known as validators, are producing new blocks for blockchain. If a validator produces a valid block, they are rewarded with a portion of the transaction fees and a block reward. However, if they produce an invalid block, they may be penalized, and their staked assets may be slashed.

Yet, staking has a few problems, and in addition to discussing and speculating about its decentralization or securing, one of them is the illiquidity of staked assets and the impossibility of changing their owner. The inability to change the owner of staked assets can be very inconvenient, for example, if we urgently need to exchange our current staked collateral for other assets. This paper closely looks at this problem, analyzes the existing solutions, and describes a new concept that introduces the staking assets through an NFT account (NFTAA), i.e., transferring the ownership of the stake to the NFT account, which allows changing the owner of the stake (through a change of ownership of the NFTAA) in case that we need.

The rest of the paper is organized as follows: In Section II, a closer look at the example of Ethereum and its technical specifications is provided. Existing related works are analyzed in Section III. Next, the solution is presented in the form of NFT proxy accounts described in Section IV, and our evaluation is described in Section V. We summarize the paper, the discussion, and future work in Section VI.

II. ETHEREUM INTRODUCTION

Ethereum stands tall as the unrivaled titan in the realm of smart contract platforms, commanding its position as the foremost and most sophisticated hub for decentralized financial ecosystems across the globe.

This pioneering blockchain network is purpose-built to facilitate the seamless operation of decentralized applications (dApps) while steadfastly pursuing continuous enhancements in scalability, an ongoing mission undertaken by its vibrant community of developers and stakeholders. Ethereum’s commitment to scalability underscores its unwavering dedication to delivering a future-proof infrastructure that can efficiently accommodate the ever-expanding demands of the decentralized world. Through its tireless innovation, Ethereum endeavors to remain the largest, most adaptable, and responsive smart contract platform, serving as the bedrock of the decentralized digital economy.

A. Proof of stake

In proof of stake (PoS), compared to proof of work (PoW) used in Bitcoin, we do not have miners but rather validators. Validators are nodes that validate blocks and their data, such as transactions. Validators are not competing to find the correct hash for a block and do not receive a reward in the form of a coinbase transaction as miners in Bitcoin. Instead, validators receive a reward through transaction fees from the transactions included in the block they validate. Having a lot of computing power to be a validator is unnecessary, and the process is less energy-intensive [4].

Validators are motivated to refrain from cheating because they must stake a certain amount of coins to become validators. These staked coins are used to secure the network and safeguard against cheating. If a validator is caught cheating, they lose their staked coins and can no longer serve as a validator. Since the reward for validating a block is typically less than the amount of staked coins, it is not financially worthwhile for a validator to cheat. In Ethereum, the amount of staked coins required to become a validator is set at 32 ETH [5]. Staking this amount is quick; however, unstaking is slightly different and can take days; recently, it was more than five days [6]. Therefore, selling the account with staked 32 ETH is quicker and gives users easy and fast access to money. However, you do not want to sell your original account because you could have history and other liquidity associated with it. Thus, we are introducing staking through an NFT which works as a proxy and isolates your stake, so you can easily sell it to anyone without risk of selling more than your stake.

III. RELATED WORK AND BACKGROUND

As mentioned, proof-of-stake can provide security for the network by other participants. As a bonus, there is also gaining of some profit from the staking. However, PoS has some drawbacks. First, to unstake the tokens, there is a need to wait for the withdrawal period.

The current withdrawal mechanism for validators has shown significant scalability concerns as the network grows. At present, the system only accommodates 16 withdrawals within a single block. This translates to a maximum processing capability of 115,200 validator withdrawals daily under the ideal scenario of no missed slots. This limitation becomes particularly pronounced in light of rising validator numbers: 800,000 withdrawals take an entire week to process. This rigid structure poses a severe challenge as it correlates the number of validators with the withdrawal processing time.

Moreover, the inherent risk of missed slots further complicates the landscape. While the system is designed to bypass validators without eligible withdrawals, thereby potentially saving some time, the downside of missed slots has a pronounced effect. Each missed slot can lead to a proportional delay in processing, pushing the timeline even further. Such inefficiencies can negatively impact user experience, potentially dissuading prospective validators from participating and undermining the very foundation of the network's decentralization principle. This bottleneck, if not addressed, threatens both the operational efficacy and the perceived reliability of the platform.

Fortunately, already existing solutions can help solve these problems. Generally speaking, there is a familiar concept: Lock ETHs and get a token representing the staked amount (derivate). Getting derivate from the staked asset is also called liquid staking. It represents the idea of making locked assets liquid. A liquid asset represents an asset that can be converted to cash. In DeFi terms, liquid assets can be swapped for other assets, such as stablecoins, or provide additional liquidity on the various decentralized exchanges.

A. Liquid Staking Protocols

Liquid staking providers empower you to tap into the liquidity of your staked assets. In return, these providers typically claim a share of the staking rewards ranging from 5% to 10% as compensation for rendering this valuable service. With the Shanghai upgrade, the appeal of Ethereum Liquid Staking Derivatives (LSDs) is set to soar, as it alleviates the inherent risks and inconveniences associated with conventional ETH staking. This surge in LSD adoption translates into amplified revenue streams for liquid staking platforms, instilling a bullish sentiment across the liquid staking sector. Anticipating this upward trajectory, MetaMask has thoughtfully integrated staking directly into its wallet interface, streamlining the process for users to stake their ETH. Users can look forward to earning approximately 5% annual rewards in ETH by leveraging this feature, with support currently extended to Lido and Rocket Pool and the possibility of more integrations in the pipeline [7].

B. Lido

Lido made its mark as a trailblazer in the field of liquid staking. In 2020, with the launch of Ethereum's Beacon chain, Lido became the first provider to offer liquid staking services, pioneering and popularizing this innovative approach. Since its inception, Lido has consistently maintained its position as a market leader.

Lido's liquid staking solution revolves around its token known as stETH, which represents users' staked Ethereum (ETH). The process is simple: users deposit their ETH on Lido's staking platform and receive stETH in return. This transformative token empowers users by preserving the liquidity of their staked assets. With stETH in hand, users can explore a myriad of DeFi opportunities. Holders of stETH can choose to engage in various DeFi activities, including trading, lending, providing liquidity, and using it as collateral. Additionally,

stETH holders earn ETH staking rewards in the form of more stETH tokens, further enhancing their potential yield.

Lido operates on a sustainable revenue model. The platform collects a 10% fee from the ETH staking rewards generated. This fee is divided between Lido's node operators, who ensure the network's security, and the DAO (Decentralized Autonomous Organization) treasury.

Despite its outstanding achievements, Lido's dominant position has raised concerns within the Ethereum community. Concentrating a substantial amount of Ether within a single organization poses potential risks to the network. A centralized attack on the Ethereum network could become feasible if Lido's control surpasses 33%. This scenario, known as a 1/3 centralization attack, is a recognized challenge in Proof-of-Stake ecosystems. In response, Lido and the Ethereum community are actively exploring avenues to enhance the decentralization of the protocol, mitigating these potential risks [7, 8].

C. Rocket Pool

Rocket Pool is a groundbreaking Ethereum PoS protocol designed to be community-owned, decentralized, trustless, and compatible with Ethereum's staking system. The project was conceived in late 2016 and has been live since October 2021. Rocket Pool caters to two primary user groups:

- **Tokenized Stakers:** Individuals who want to participate in Ethereum staking using rETH with deposits as low as 0.01 ETH.
- **Node Operators:** Users who stake ETH and run a node in the Rocket Pool network to earn higher returns compared to staking independently outside the protocol. This includes experienced users and those who may not have 32 ETH for solo staking.

The core premise of Rocket Pool is to ensure the network remains decentralized, aligning with Ethereum's ethos and promoting self-sovereignty. Rocket Pool comprises several key components that create a decentralized Ethereum staking ecosystem. **Smart contracts** facilitate ETH deposits, allocate them to node operators based on demand, and manage various tokens and rewards. Rocket Pool's decentralized network of specialized Ethereum nodes runs the **Smart Node** software. These nodes communicate with the protocol's smart contracts and provide the network consensus required by the Ethereum Beacon Chain. **Minipool validators** are smart contracts created by node operators who deposit 8 or 16 ETH on their node. When the total ETH in a Minipool validator reaches 32 ETH, a new validator is created to perform consensus duties.

Rocket Pool's commitment to decentralization, security, and inclusivity makes it an attractive solution for those looking to participate in Ethereum staking without the traditional barriers associated with solo staking. The decentralized protocol ensures resilience, scalability, and more equitable distribution of rewards within the network [7, 9].

D. EIP-6551

The world of non-fungible tokens (NFTs) is on the brink of a significant transformation with the introduction of the ERC-6551 token standard. This new standard seeks to enhance the functionality of NFTs by equipping them with smart contract capabilities, allowing them to represent unique digital assets and act as smart contract wallets. The ERC-721 standard, the foundation for creating unique digital assets on the Ethereum blockchain, has its limitations. While it paved the way for myriad applications, from art to virtual real estate, its scope is restricted. NFTs based on ERC-721 can only be owned and transferred; they cannot own other assets or interact dynamically with other smart contracts. Proposed as an Ethereum Improvement

Proposal (EIP) in February 2023, ERC-6551 aims to address these limitations. It introduces a system where every NFT can be assigned an Ethereum account, termed a token-bound account (TBA). This account enables the NFT to own assets, interact with applications, and execute arbitrary operations, all without necessitating changes to existing smart contracts or infrastructure. Key features:

- **Smart Contract Capabilities:** Beyond representing unique assets, ERC-6551 tokens can function as smart contract wallets, holding other tokens and NFTs.
- **Interactivity:** TBAs can seamlessly interact with various entities on the Ethereum network, from decentralized exchanges to gaming platforms.
- **Composability:** This feature allows NFTs to bundle with related assets, enhancing user experiences, especially in Web3 gaming environments.
- **Identity & Provenance:** Each NFT, with its distinct identity, can interact independently with decentralized applications (dApps). Additionally, users can trace the complete transaction history of an asset, offering a clearer picture of its past interactions and uses.

The ERC-6551 standard represents a pivotal moment in the evolution of NFTs. Bridging the gap between unique digital assets and smart contract functionality opens up a realm of possibilities in the Web3 space, from gaming to art and beyond. As the crypto community navigates its challenges, the potential of ERC-6551 to redefine digital ownership remains immense [10, 11, 12].

But this solution also has some gaps. Firstly, the existence of TBA is not documented in the NFT, meaning we could possess an NFT without any knowledge of an associated account until we check the TBA register. We can also sell an NFT with possession of different assets by mistake and thus lose our precious liquidity. The second major gap is the lack of atomicity in the creation process. Of course, this is because TBA applies to all existing NFTs. Still, we think that creating an NFT and wallet in one atomic transaction, ensuring that the NFT is fully bound to the wallet proxy contract from its inception and that the NFT has information about its connection to the wallet contract, is more transparent.

IV. SOLUTION DESIGN

In the case of the existing solutions (except ERC-6551), the ETH collateral is often protected via a DAO / multi-signature account. However, this means that the security of the assets is not directly in the hands of the user because, in case of a problem or a third-party devaluation, our liquid assets representing our stake are depreciated. This section delves into the cutting-edge notion of staking via NFTs, which serve as distinct accounts within the system. Although this method is not yet common, we bring a few insights into why it is a good idea.

The term "non-fungible token" refers to a unique and distinct asset that can represent various physical items with varying values. On the other hand, NFTs primarily operate on the asset side of the owner-asset relationship and lack the potential to own other assets. On the other hand, accounts are used to represent ownership of assets. In an owner-asset relationship, the account can be on the first or second side. After composing these two relations, it follows that it is possible for an account and an NFT to be represented by a single object that will be uniquely identifiable and will have the nature of an asset that can be owned - change its owner and at the same time can own other assets. We represent a non-fungible token as an account, NFTAA in short. Most Ethereum is based on a single account architecture consistent with the NFTAA

philosophy and facilitates connection with existing infrastructure. We may easily exploit existing infrastructure for smooth communication if NFTAAs are handled as conventional accounts. Furthermore, if needed, NFTAAs can function as asset owners or in a multi-sig scheme.

Another key characteristic of NFTAA is the ownership possibilities and the transferability of this ownership to any other account, potentially other NFTAAs. This means that the private keys used to alter a certain NFTAA will change. This is feasible thanks to the well-known proxy smart contracts [13, 14]. Figure 1 shows the relationships between NFTAA and other logical entities.

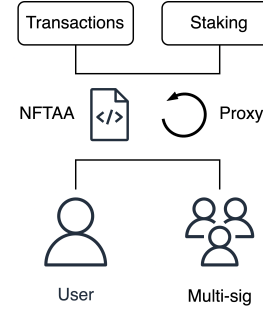


Fig. 1. NFT as an account interaction

We can create a proxy account (smart contract) to perform specific tasks on behalf of another account. For example, Bob's account can have a proxy account named Alice. Alice can send transactions on Bob's behalf, but only for the calls Bob has approved, as Bob is the only person currently authorized to access the proxy. This means that if Alice sends a transaction, it is initiated by Bob, so Bob appears to be signed by Bob, even though Alice is technically sending it.

A. DeFi and DAO use-cases

When we discuss accounts, NFTs, assets, staking, ownership, and owners - DeFi comes into play, and because of the nature of NFTAA, we will also discuss DAO. As stated before, if we own ETH and decide on liquidity staking, we have staked the ETH in exchange for other assets we can continue utilizing. Compared to futures or tokens as a reward for staking, NFT may appear to be one large illiquid piece of value. This is not the case because fractionalizing NFT is not novel in the DeFi world [15]. The ideal number of fractions equals the number of ETH tokens staked on behalf of the NFTAA account. However, for whatever reason, we may be unable to fold NFTs distributed in this manner back to their original condition. We could remedy this by holding a buyout auction or depositing a sum greater than the reported staking value. We can still perform an unstake if our NFTAA becomes fully illiquid for any reason and cannot be sold on the secondary market or used as collateral for a loan, and after this, the NFTAA may (or may not) be burnt.

Another intriguing application of NFTAA is the ability to manage an account using numerous proxy accounts, each with the capacity to make certain calls. We can think of NFTAA as a company with employees who have specific roles. Specific subaccounts represent particular roles, but the company signs all transactions, and NFTAA, i.e., the corporation owns all other assets. NFTAA, like any other corporation, can be acquired. The NFTAA also functions as the company's ledger.

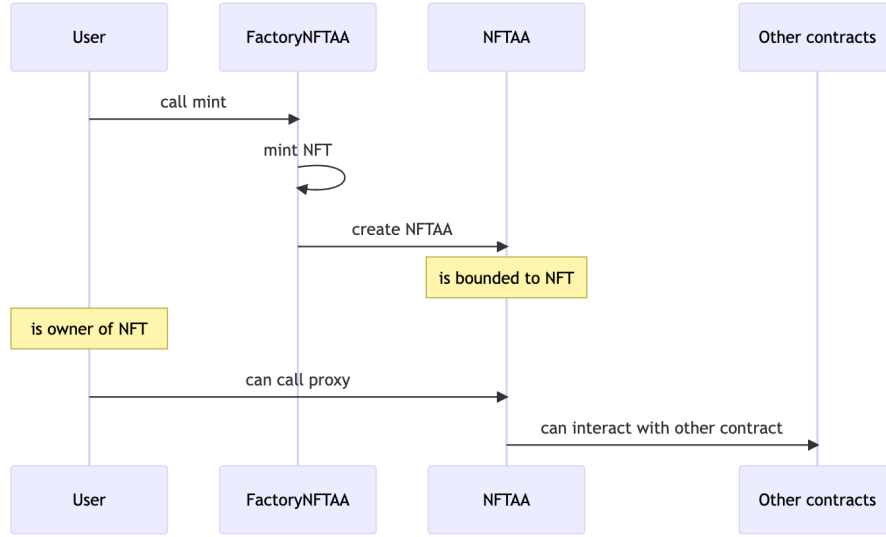


Fig. 2. NFTAA Solidity sequence diagram

B. Architecture

Figure 2 shows the key elements of the architecture. The foundation of the entire implementation is the NFTAA smart contract, which has ERC-721 as the underlying implementation. Subsequently, the NFT mining functionality is enriched by the creation of the NFTAA smart contract, which is interleaved from the NFT that has been mined. The NFTAA contract contains a proxy function that allows the owner of the pinned NFT to perform transactions through the NFTAA account.

Listing 1 shows how we can modify a function to be successfully called only if the caller owns a bound NFT.

```

1 modifier onlyNFTOwner() {
2     require(
3         msg.sender == ERC721(_binded_nft_addr).
4             ownerOf(_binded_nft_id),
5         "Caller is not the owner of the NFT"
6     );
7 }

```

Listing 1. NFT owner only modifier

We created a web application that allows users to engage with the NFTAA manufacturing contract. The application is built on a template for developing web apps that interface with web3 applications [16]. This template is useful, but it only works with the Talisman wallet. Unfortunately, the Talisman documentation does not state if it allows signing EVM transactions [17, 18]. To alleviate this constraint, we have added Metamask wallet support. Finally, for ease of use, as we can see in Figure 3, we created a dApp using the NFTAA concept for staking. Via this decentralized application, we can: Stake, Add other funds to the stake, Unstake, Get a balance of the stake, Get the staker address, Mint new NFTAA, Transfer ownership of NFTAA.

NFTAA introduces a novel approach to account management. Each user is issued a unique NFT that can represent their account. This NFT can encode specific attributes, permissions, or properties related to the user, such as staking rights, access to certain functionalities, or governance privileges within a decentralized ecosystem. In particular, the use-case of proxy-staking accounts leverages this NFT-based approach for staking mechanisms. Instead of the conventional method of staking tokens in a wallet, users stake their NFT-based accounts' tokens. The NFT encapsulates staking parameters, delegation prefer-

ences, and other relevant details for participating in a proof-of-stake (PoS) or delegated proof-of-stake (DPoS) consensus mechanism. Through this model, the NFT serves as a representation of ownership and a means to determine the user's staking weight and influence within the network. The more tokens staked through a bound NFT account, the more significant the user's impact on consensus, rewards, and potentially, governance decisions.

Furthermore, NFTAA can facilitate delegation strategies, allowing users to delegate their staking rights or influence to other accounts or entities while still maintaining ownership of their NFT-based account. This flexibility enhances the overall decentralization and participation in the network. This proposed framework does not emphasize the preservation of backward compatibility with existing NFT contracts, abstaining from alterations to the ERC-721 standard. However, it is theoretically possible to choose the standard to be used when creating NFTAA so that we could use ERC-165, but of course, we prefer to use ERC-721. Alternatively, a partial subset of the ERC-721 interface could be used (e.g., ENS NameWrapper names).

C. NFTAA vs TBA

As highlighted in Section III-D, EIP 6551 presents certain limitations. Our solution, however, does not prioritize connecting existing NFTs but enables bidirectional binding of the NFT and Proxy contract. Consequently, upon ownership of an NFT, we can discern from the NFT metadata that it is associated with another smart contract, indicating its status as an NFTAA. Furthermore, we enhance transparency by ensuring the atomic creation of the NFT and proxy contract within a single transaction, thereby promoting seamless integration. Likewise, our design does not need the registrar service to create the TBA address as opposed to TBA. In our case, the factory contract creates the NFTAA contract with functionality that also works as the NFTAA address (TBA uses one more address than NFTAA). Another difference is that in TBA, you could have more accounts on one NFT, which in our case is not possible due to the atomic creation of exactly one NFT at the time of creating the NFTAA smart contract (we first create the account contract and exactly after that the unique NFT). However, theoretically, you could use already existing NFTAA as the NFT in TBA :-). Regarding security, there are at least two differences between NFTAA and TBA. The first is the

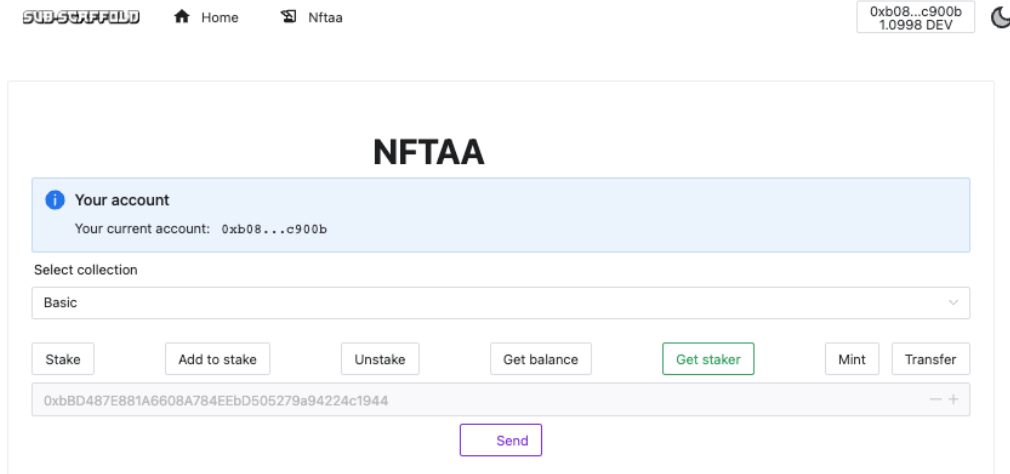


Fig. 3. NFTAA dApp

possibility of generating the address of TBA sooner without having the deployed contract (as the create2 function for creating contracts is deterministic, i.e., you can calculate the future contract address). In this case, if the upcoming smart contract is badly developed (does not contain an execute function for byte calls to other contracts) or is not upgradeable, it can happen that assets or liquidity sent to the TBA will not be withdrawable and become locked forever (even the NFT bounded in TBA can be sent by a mistake (if not safeguarded) to the address of TBA. Therefore you lock yourself from it). The problem is that TBA uses only the address of any contract you want in the registry contract and does not look for any details on how the contract looks or if it is already deployed. In our case, the account form is pre-defined in the factory contract, and even after deployment, we guarantee upgradeability as it is a proxy account. The second security issue is potential fraud, which is described in the EIP 6551 specification¹ that you withdraw assets from the TBA and at the same time sell the NFT to someone who fought that they will also have the assets in the TBA. In our case, we provide the security measures for not withdrawing the assets from the NFTAA while also selling the NFT, which is bound to the NFTAA smart contract. A small difference is also in the address where you receive the assets (unstake in the case of staking). In TBA, the unstaked amount will go to the TBA address; in NFTAA, the unstaked amount will go to the NFTAA smart contract address.

V. EVALUATION AND TESTING

When testing decentralized applications, the most important thing to focus on is the core functionality performed on the blockchain. The most significant part is the smart contract. Testing smart contracts is more complicated than testing regular software, but many tools make this easier nowadays. One of them is the Ethereum development environment Hardhat [19]. This development tool allows us to run the blockchain network locally and, among other things, provides extensive support for testing and mocking data to simulate a real blockchain environment.

A. Smart contract testing

Based on these functionalities, we built our tests documented in Figure 4. The tests were built to evaluate the key elements of our concept. To this end, the tests mainly focus on:

- Flow test - determine whether the system can successfully mint a new NFTAA contract, emit a 'NewNFTAA' event, and check if the minted NFT is bonded.
- Validations - the system ensures that transactions attempting to mint an NFTAA contract with an empty or excessively large description are correctly reverted.
- Staking - ensures that users can stake their funds, increase their stakes, and unstake them. Additionally, it verifies that users cannot unstake before the system reaches the unlock time.
- Attributes - the system's ability to return the proper bound NFT ID and NFTAA note is tested in these tests.
- Proxy - examine if the system correctly revokes actions taken by parties other than the contract owner and whether it issues a "ProxyResponse" event whenever a legitimate function call performed via the proxy is received.

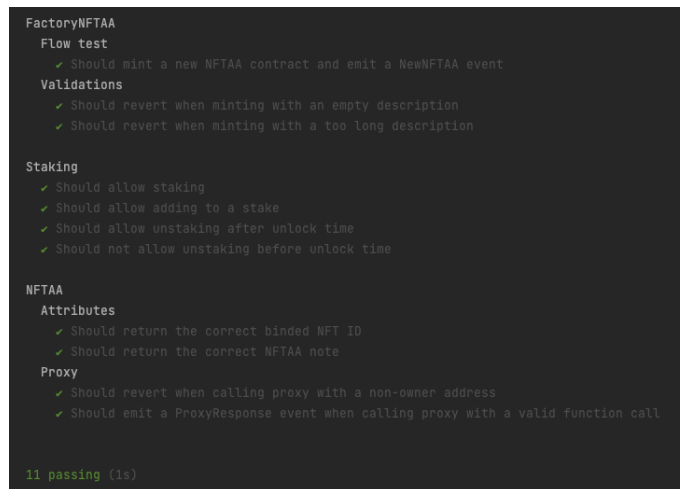


Fig. 4. Results of smart contracts testing

Eleven subtests passed the testing phase. By running these automated tests, we can ensure that the smart contract system performs as expected and adheres to our expectations.

¹<https://eips.ethereum.org/EIPS/eip-6551#fraud-prevention>

B. Requirements evaluation

This section evaluates how our system meets the requirements defined by analysis.

The evaluation of requirements, as depicted in Table I, demonstrates the successful implementation of key features within the system. The initial requirement (Request 1.) ensuring users can connect their wallets has been met, establishing a fundamental connection between users and the platform. Subsequently, the creation of NFTAAs (Non-Fungible Token Asset Accounts) is confirmed successful (Request 2.), emphasizing the system’s capability to generate unique and distinct token accounts. Transactions are seamlessly facilitated with the successful transfer of NFTAAs between wallets (Request 3.), affirming the platform’s robustness in handling asset transfers. Stakeholder engagement is supported through functionalities such as staking (Request 4.) and unstaking (Request 5.) via NFTAA, reflecting the system’s commitment to user participation in blockchain-based activities. Moreover, users can conveniently monitor their NFTAAs (Request 6.), fostering transparency and awareness of individual token holdings. The system’s ability to verify ownership of NFTAA (Request 7.) adds an additional layer of security and accountability. The rest of the requirements are non-functional. We did a thorough, smart contract security audit by tools Slither² and MythX³ to evaluate non-functional requirement number fourteen. There were no critical or severe issues in the security model. As a result, the acceptance testing of the non-functional requirements is successful. The evaluation underscores the system’s effective incorporation of essential functionalities, meeting or exceeding the specified requirements and providing users with a comprehensive and reliable blockchain experience.

TABLE I
EVALUATION OF REQUIREMENTS

Request identifier	Description of the requirement	Evaluation
1.	The system shall allow users to connect their wallet	Successful
2.	The system shall allow users to create an NFTAA	Successful
3.	The system shall allow users to transfer NFTAA to another wallet	Successful
4.	The system shall allow users to do stake via NFTAA	Successful
5.	The system shall allow users to do unstake via NFTAA	Successful
6.	The system shall allow users to check their NFTAAs	Successful
7.	The system shall allow users to check owner of some NFTAA	Successful
8.	The system core functionality shall be implemented by smart contract	Successful
9.	The system shall provide a user-friendly interface, with clear and concise instructions and error messages	Successful
10.	The system shall be scalable to handle a large number of users and data storage requirements	Successful
11.	The system shall be used by any wallet	Successful
12.	The system shall be compatible with modern web browsers	Successful
13.	The system shall be compliant with relevant industry standards and best practices	Successful
14.	The system shall be secure and protected against cyber-attacks	Successful

VI. CONCLUSION AND FUTURE WORK

Non-fungible tokens as accounts have massive potential in the Ethereum ecosystem. They can provide a valuable asset that can be used as an account in the runtime and traded on the market. One of the best features of the NFTAA system is that any account type can cooperate with NFTAA: externally owned accounts, smart contracts, proxies, or multi-signature accounts. Staking is an excellent use-case example for NFTAA. Unlike classic liquid staking, NFTAA does not rely on any DAO, multi-signature accounts, or entity that manages the token; instead, the owners are in full custody of their assets. Additionally, the network does not lose any security since no funds are unstaked and can be withdrawn. Moreover, non-fungible

tokens can be used in other DeFi applications, such as lending, borrowing, and derivatives, so they do not lose any flexibility. From the implementation perspective, the NFTAA can be implemented with current and available Solidity tooling. However, the applicability of NFTAA is not only in staking use-case but also company owning, asset and portfolio management, delegation of rights to another wallet, or even identity and role management (e.g., you could have a game profile or persona linked to the NFT and in future sell it with every item in the itinerary). The possibilities of NFTAA are nearly endless.

In summary, non-fungible tokens as accounts, specifically in the context of proxy-staking or token-bound accounts, introduce a paradigm shift in account management and staking within blockchain ecosystems. Integrating unique NFTs with account functionalities enhances personalization, participation, and governance within decentralized networks. It reflects the evolving landscape of blockchain technology, where innovative applications continually redefine the boundaries of traditional concepts. This paper offers design and architecture decisions with a simple implementation, which allows developers and researchers to choose the best solution for their needs.

Future work includes examining further use-cases for NFTAA and implementing the NFTAA in different networks, e.g., the Polkadot ecosystem, where there is already an implementation of proxy accounts and the unstaking lasts 28 days. Implementing NFTAA in the Substrate environment means bringing this concept in a completely different technology. Also, the plan is to extend the research to more enterprise use-cases with great potential for business and blockchain fusion.

REFERENCES

- [1] Elie Kapengut and Bruce Mizrach. “An Event Study of the Ethereum Transition to Proof-of-Stake”. In: *SSRN Electronic Journal* (Oct. 2022). DOI: 10.2139/ssrn.4247091.
- [2] Christophe Schinckus, Canh Phuc Nguyen, and Felicia Hui Ling Chong. “Cryptocurrencies’ hashrate and electricity consumption: evidence from mining activities”. In: *Studies in Economics and Finance* 39.3 (Apr. 2022), pp. 524–546. ISSN: 10867376. DOI: 10.1108/SEF-08-2021-0342.
- [3] Benjamin Johnson et al. “Game-theoretic analysis of DDos attacks against bitcoin mining pools”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8438. Springer Verlag, 2014, pp. 72–86. ISBN: 9783662447734. DOI: 10.1007/978-3-662-44774-1_6.
- [4] Fahad Saleh. “Blockchain without Waste: Proof-of-Stake”. In: *The Review of Financial Studies* 34 (3 Feb. 2021), pp. 1156–1190. ISSN: 0893-9454. DOI: 10.1093/RFS/HHAA075. URL: <https://academic.oup.com/rfs/article/34/3/1156/5868423>.
- [5] Elie Kapengut and Bruce Mizrach. “An Event Study of the Ethereum Transition to Proof-of-Stake”. In: *SSRN Electronic Journal* (Oct. 2022). DOI: 10.48550/arxiv.2210.13655. URL: <https://arxiv.org/abs/2210.13655v1>.
- [6] Margaux Nijkerk. *Ethereum Validators Forced to Wait Days to Unstake Amid Celsius Withdraws*. en. Section: Technology. Jan. 2024. URL: <https://www.coindesk.com/tech/2024/01/05/ethereum-validators-forced-to-wait-days-to-unstake-amid-celsius-withdraws/> (visited on 03/04/2024).
- [7] CJ. *What are Liquid Staking Derivatives (LSDs) and How Do They Unlock Liquidity?* URL: <https://www.coingecko.com/learn/what-is-liquid-staking-liquid-staked-derivatives-you-need-to-know>.

²<https://github.com/crytic/slither>

³<https://mythx.io/>

- [8] Lido. *What is Lido*. URL: <https://docs.lido.fi/>.
- [9] Rocket Pool. *Explainer Series*. URL: <https://docs.rocketpool.net/overview/explainer-series.html>.
- [10] Jayden Windle et al. *ERC-6551: Non-fungible Token Bound Accounts*. URL: <https://eips.ethereum.org/EIPS/eip-6551#example-account-implementation>.
- [11] John Isige. *ERC-6551: The Future Token Standard for NFTs and Gaming?* URL: <https://www.techopedia.com/erc-6551-the-future-token-standard-for-nfts-and-gaming>.
- [12] Ivan Cryptoslav. *What Is ERC-6551? Explaining the New NFT Token Standard*. URL: <https://coinmarketcap.com/academy/article/what-is-erc-6551>.
- [13] Ethereum Foundation. *UPGRADING SMART CONTRACTS*. 2023. URL: <https://ethereum.org/en/developers/docs/smart-contracts/upgrading/>.
- [14] OpenZeppelin. *Proxies*. 2023. URL: <https://docs.openzeppelin.com/contracts/4.x/api/proxy>.
- [15] Wonseok Choi, Jongsoo Woo, and James Won-Ki Hong. “Gas cost analysis of fractional NFT on the ethereum blockchain”. In: *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE. 2023, pp. 1–6.
- [16] Branislav Hozza et al. *Sub-scaffold*. URL: <https://github.com/kodadot/sub-scaffold>.
- [17] Talisman. *Joining EVM Networks*. 2023. URL: <https://docs.talisman.xyz/talisman/navigating-the-paraverse/ethereum-features/joining-evm-networks>.
- [18] Talisman. *Talisman FAQ*. 2023. URL: <https://docs.talisman.xyz/talisman/help-and-support/faq#q-can-i-use-talisman-extension-to-send-funds-to-an-evm-address>.
- [19] Nomic Foundation. *Hardhat*. URL: <https://hardhat.org/>.