ReviewRewards Prototype Design testing

ReviewRewards introduces a flexible framework for journals to tailor review reward schemes to their unique criteria and review processes. This platform acknowledges that different journals can have different review methodologies and incentive preferences, offering them the freedom to decide upon the most appropriate reward mechanism.

The prototype is built using React.js for its user interface, chosen for its component-based architecture that simplifies the development of a dynamic and responsive UI (*React*, n.d.). ReviewRewards is supported by REST API built with Node.js that provides a robust connection to the Oracle autonomous database (*What Is Oracle Autonomous Database?*, 2024). It supports an interface for three user types: author, editor, and reviewer.

- **Authors** can submit their manuscripts to the journal for review
- **Editor** verifies that the manuscript is of sufficient quality and assigns reviewers. For simplicity, the first iteration of the prototype considers the journal's editorial role with a single editor account.
- **Reviewers** submit reviews for their assigned manuscripts.

Blockchain

The application is built on blockchain technology to maintain a ledger of all forms of rewards received by the reviewers. The inherent property of blockchain ensures that the rewards once awarded to the reviewer will remain perpetual, i.e. DP2. The public nature of blockchain ensures that the reward mechanism is transparent, i.e. DP3, while maintaining privacy adhering to the confidentiality standard of the review process.

Incentive Engineering

Incentivization is at the core of the application, designed to motivate high-quality reviews. The system's goal is to incentivize based on the motivating factors of the majority of reviewers. Different outlets follow different forms of peer-review processes and prefer to reward their reviewers with different incentivization policies. Hence, the designed system empowers editors to customize settings to align with their journal's specific requirements. Currently, the prototype supports a double-blind peer review process, with future plans to accommodate additional methodologies. This study identifies two primary incentive models: non-transferable certificates and utility tokens. Non-transferable certificates serve as immutable records of achievements, affiliations, or identities, while utility tokens offer tangible rewards and facilitate various transactions within the system.

1. *Non-transferrable certificates:* Non-transferrable certificates are cryptographic tokens that serve as proof of achievement or credentials. Unlike traditional tokens, they can't be freely transferred or exchanged. These tokens are similar to digital badges or certificates, as they are tied to a specific individual's or entity's identity, ensuring that the achievements or credentials they represent remain with the original recipient. Issuance of such non-transferable certificates will be particularly valuable for researchers at early career stages by acknowledging their contribution to academic progress.
2. *Utility tokens:* Utility tokens are digital assets designed to be used within a specific blockchain ecosystem, providing access to goods or services offered by the platform (*What Is a Utility Token? Utility Token Definition*, n.d.). Unlike non-transferable certificates, utility tokens are transferable and can be traded on various exchanges. The design of utility tokens focuses on creating intrinsic value within the ecosystem, such as payment for services, or access to premium features. Such utility tokens can be used as credits for subscription-based journals or as currency required for authors to make a manuscript submission themselves.

Prototype Implementation

This section details the components of the ReviewRewards prototype. The general overview of the system architecture is shown in Figure 1.

User Interface

ReviewRewards' user interface consists of the following pages

- *Login page and Dashboard*: Users are first directed to the login page, which requires authentication using Metamask as shown in Figure 3(a). The system assumes that the user will have the Metamask browser extension installed on their preferred browser. As an individual can have different roles based on different journals, each user can assume the role of author, editor, or reviewer.

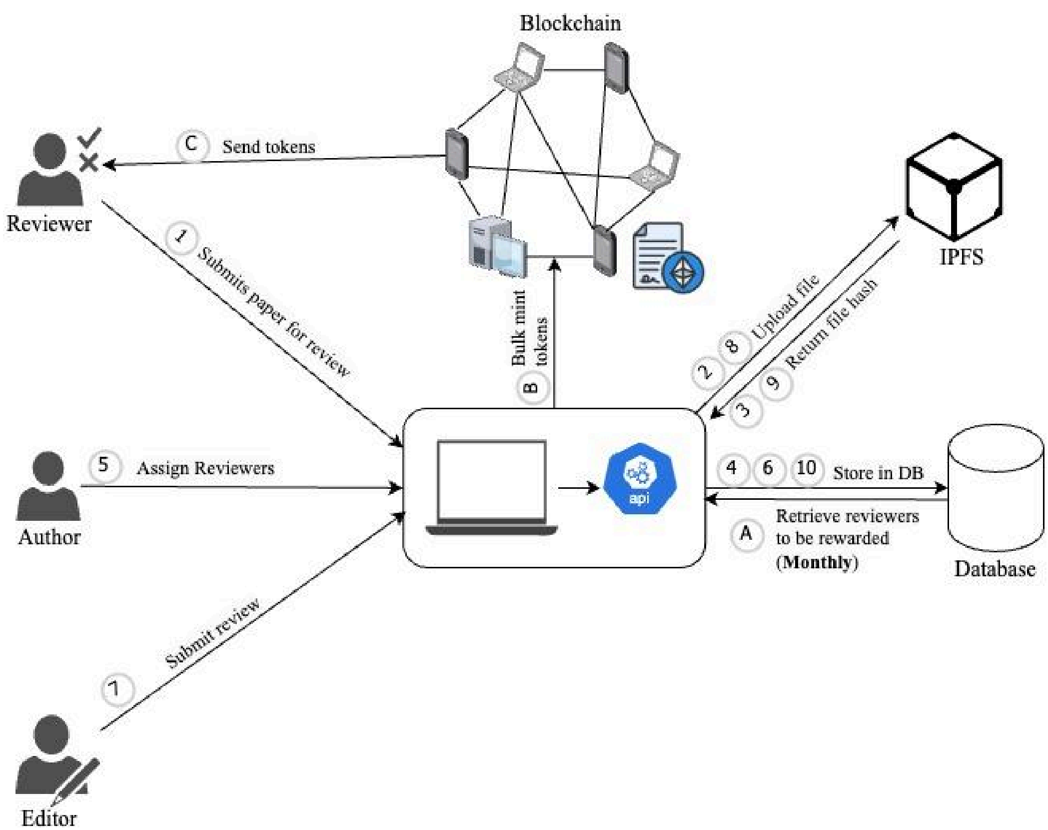


Figure 1 Architecture design of ReviewRewards

- *Author dashboard:* Authors can submit their manuscripts to their chosen journals from this page, shown in Figure 3(b) and Figure 3(c).
- *Editor dashboard:* Editors can view the manuscripts submitted to their journal from this page. They can assign manuscripts to multiple reviewers and set deadlines for review submissions, as shown in Figure 3(d).
- *Reviewer dashboard:* Reviewers can view the manuscripts assigned to them from the reviewer dashboard, Figure 3(e). They can upload reviews along with additional information as in Figure 3(f).
- *Settings page:* The settings page offers editors the flexibility to customize the incentive structure of the review process. This is shown in Figure 3(h). Here, editors can enable or disable the issuance of utility tokens (RRTs) and outline the reward policy, including the allocation of different tokens based on the timeliness of review submissions. This feature allows for the incentivization of prompt reviews and the imposition of penalties for late submissions.
- *Reputation page:* This page enables users to view the non-transferrable certificates and utility tokens they have earned, serving as a digital portfolio, as shown in Figure 3(g).

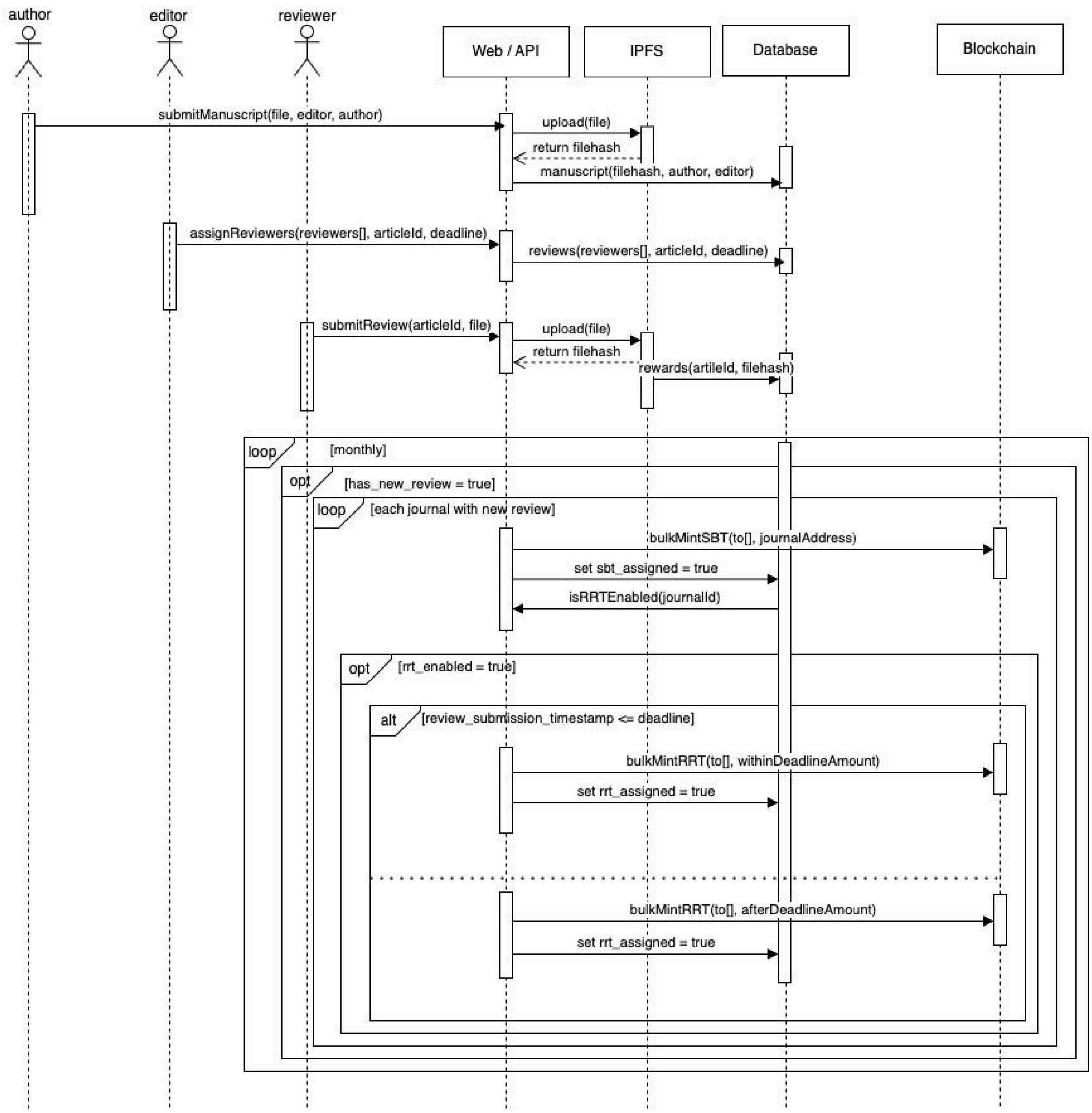
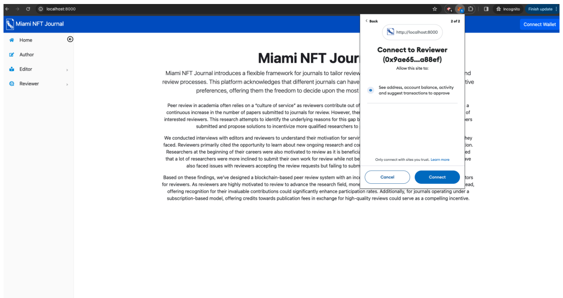
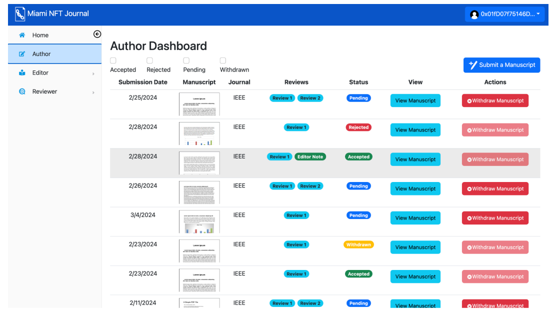


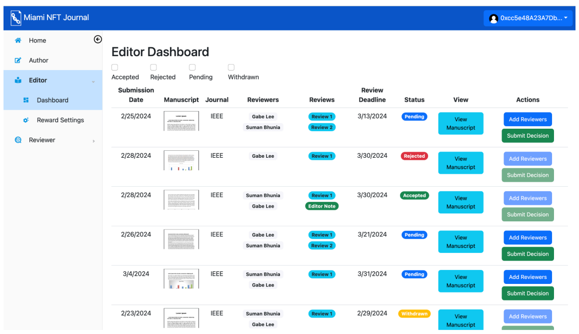
Figure 2 Sequence diagram

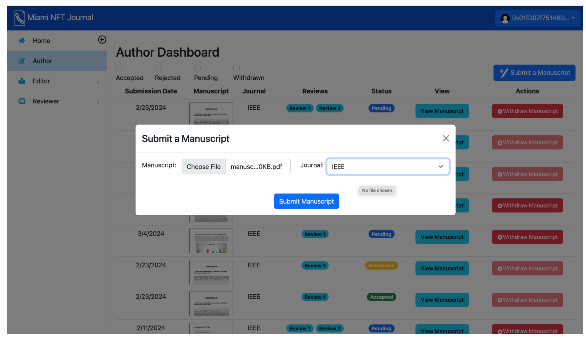




(a) Login page with Metamask

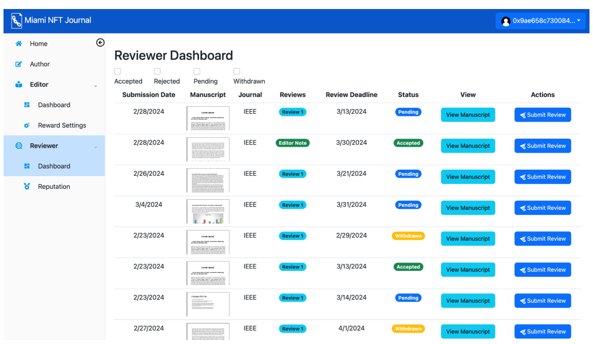
(b) Author dashboard

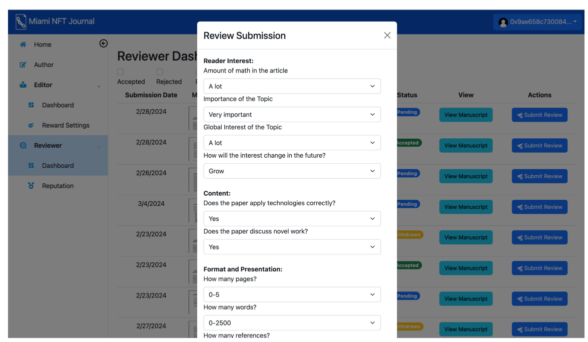




(c) Manuscript submission

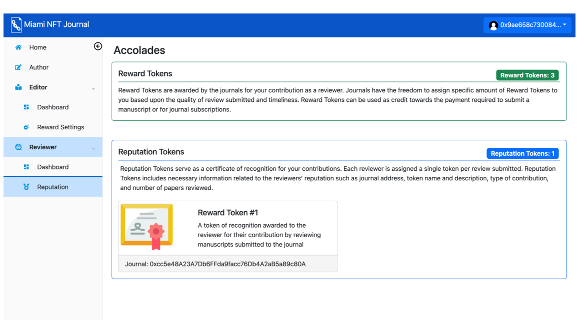
(d) Editor dashboard

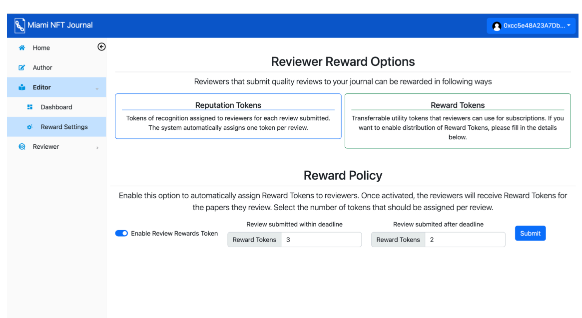




(e) Reviewer Dashboard

(f) Review submission popup





(g) Reputation page

(h) Settings page

Blockchain

There are multiple blockchain networks available with their own set of specialized functionalities, pros and cons. While selecting a blockchain network, there are several things to consider such as transaction costs, transaction throughput, decentralization, and security. Ethereum is a widely popular and accepted blockchain network with a very active and strong ecosystem with a larger developer community and extensive documentation. However, Ethereum lags in terms of scalability and affordability with its competitors such as Avalanche and Solana. Although Solana offers lower costs and higher throughput, it provides less degree of decentralization as it has a smaller set of validators (*Solana Vs. Ethereum: A Comprehensive Comparison*, 2023). Avalanche is a close competitor to Ethereum providing almost the same level of security with better scalability and lower transaction costs, however, Ethereum is much older and has a strong history of security testing, and also is more robust for non-fungible tokens (*Avalanche (AVAX) Blockchain - Getting to Know the Ethereum Rival*, n.d.). As Soulbound tokens, a special type of NFTs, play a major role in ReviewRewards, Ethereum appears to be the best choice. Later, if there is a need for better transaction throughput and lower costs, a transition can be made to an Ethereum Layer 2 solution such as Polygon (Werth et al., 2023).

Ethereum supports smart contracts, which are executable code that automates transactions when specified conditions are met. These smart contracts are responsible for minting the non-transferable certificates and reward utility tokens for the corresponding reviewer addresses. Smart contracts are written using the Solidity programming language, which is specifically designed to target the Ethereum Virtual Machine (EVM) (*Solidity*, n.d.). They are deployed in Ethereums' Goerli and Sepolia testnet for testing and verification.

Cryptowallet

Cryptowallets are digital wallets that allow users to manage, store and transfer cryptocurrencies securely. They hold the cryptographic keys used to sign transactions on the blockchain, enabling users to prove ownership of their digital assets. These cryptowallets provide reviewers with a secure mechanism to receive and manage their earned reward tokens.

Metamask is a widely used crypto wallet (*Get Started With MetaMask Portfolio*, n.d.). It stands out as the optimal choice for this system because of its strong security standards, wide accessibility, and ease of use. It offers a user-friendly browser extension that is compatible with most popular browsers, making it easily accessible for users regardless of their preferred web platform. This ease of access is crucial for a system that relies on broad participation from authors, reviewers, and editors. Additionally, MetaMask's robust security features and straightforward interface simplify the process of interacting with the blockchain to claim and manage tokens. Its widespread adoption and support make it a reliable and convenient tool for users to engage with the token-based incentive mechanism of the peer review system.

Review Process Database

The inherent transparency and immutability of blockchain pose significant challenges for maintaining the anonymity requirements of a single or double-blind peer review process. The reviewers can be easily traced back to the reviews they submitted by looking at the transaction details and timestamps of token generation. Hence, a hybrid approach as mentioned (Tasca & Tessone, 2017) is beneficial. Blockchain is used to handle token minting and transactions, ensuring transparency and immutability for incentive mechanisms. All the sensitive data related to authors' and reviewers' identities are securely managed in an off-chain database. SQL databases excel in efficiently handling complex and structured data. Oracle autonomous database is selected as it allows efficient data retrieval and manipulation with the added benefits of advanced encryption techniques, automated backups, maintenance, and scalability.

The database stores all sensitive information such as authors, papers submitted, reviewers assigned, and reviews submitted represented by Figure 4. *Manuscripts* table stores details related to manuscript submission. All the information related to reviewers assigned for each manuscript and review submission is stored in the *reviews* table. The *reward_settings* table stores reward policies set up by the journal. Once a review is submitted, the reviewers' address and flags indicating whether they have been appropriately rewarded are stored in the *reward_allocation* table.

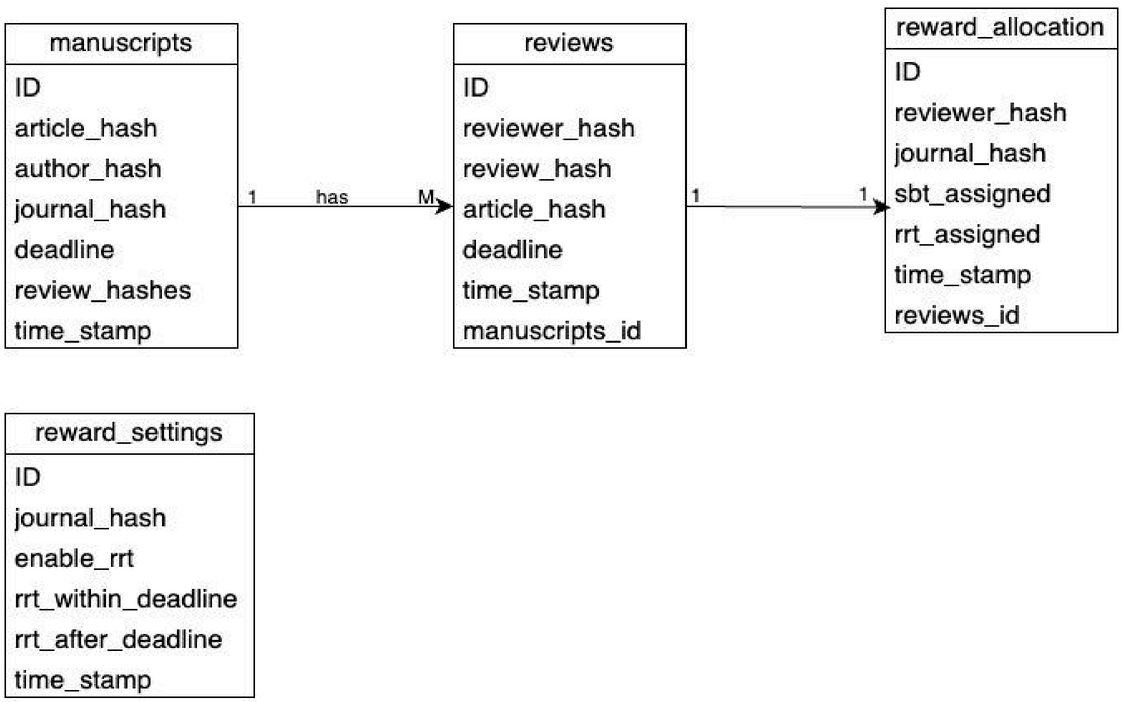


Figure 4 ReviewRewards database ER diagram

Tokenization

The incentives for review submissions are distributed in the form of blockchain tokens. Non-transferrable certificates are realized through Soulbound tokens whereas the standard ERC-20 tokens termed Review Reward tokens (RRT) are used as a form of utility tokens.

Soulbound Tokens

Soulbound Tokens are unique digital tokens designed as an enhancement of the ERC-721 standard for non-fungible tokens (NFTs) (*ERC-721 Non-Fungible Token Standard*, 2023). Soulbound tokens are non-transferrable tokens that once minted and assigned to an entity are permanently tied to them, serving as an immutable record of one's experiences, skills, memberships, and achievements within the digital ecosystem (Weyl et al., 2022, #). These tokens are enabled at default and serve as a certificate of recognition for reviewers' contributions. Each reviewer is assigned a single token per review submitted. SBT includes necessary information related to the reviewers' reputation such as journal address, token name and description, type of contribution, and number of papers reviewed.

Review Reward Tokens

RRTs are ERC-20 (*ERC-20 Token Standard*, 2023) based tokens that can be redeemed for different purposes within the system, such as subscription fee credits. These tokens are disabled by default. Editors can enable the use of these tokens in their policy settings page. They also have the flexibility to adjust distribution amounts for the reviewers who submit within the deadline and after the deadline, encouraging timely submission of reviews. Later iterations of the system will have the option for editors to add weightage to the reviews submitted so that tokens can be distributed based on the review quality.

Once the outlet updates its incentivization policy, the system will regularly (once a month) check for new reviews submitted to the journal. The scheduled job checks for pending rewards in *reward_allocation* table. Tokens will be distributed to the reviewers appropriately based on the journal policy.

Review Reward Algorithm

Each month check for new review submissions (reward_allocation.rrt_assigned = false or sbt_assigned = false)

    For each journal with new review submissions and submission_timestamp < 1 month

        trigger **bulkMinkSBT(toReviewerAddresses[], journalAddress)**

Set sbt_assigned = true

If journal has RRT tokens enabled (reward_settings.rrt_enabled = true)

Separate reviews submitted before and after deadline

For submission_timestamp <= deadline

Trigger **bulkMintRRT(toAddresses[], withinDeadlineAmt)**

Set rrt_assigned = true

For submission_timestamp > deadline

Trigger **bulkMintRRT(toAddresses[], afterDeadlineAmt)**

Set rrt_assigned = true

File storage

InterPlanatory File System (IPFS) is a decentralized, peer-to-peer file-sharing protocol (*What Is IPFS?*, 2023). It facilitates the storage and sharing of files in a distributed network of nodes, moving from a traditional, centralized web server model to a more resilient, distributed system. This eliminates the reliance on single servers and single points of failure, enhancing the reliability against data loss. Using IPFS on the ReviewRewards system ensures that even if there's a network outage, manuscript files and reviews are always accessible. This approach not only improves the security and availability of data but also reduces the bandwidth costs associated with accessing files from a centralized location.

Prototype Evaluation

Token rewards allocated to reviewers are permanently recorded on the blockchain, ensuring their perpetuity independent of the operational state of any centralized server, satisfying DP2. The process of minting tokens on a blockchain network demands significant computational resources. The prototype is evaluated by estimating the costs associated with executing various smart contract functions, and measuring these expenses in terms of gas prices. Gas prices represent the fees required for conducting transactions or executing smart contracts on the blockchain. The gas price for the ReviewRewards prototype is measured using the Foundry tool.

The analysis, as depicted in Figure 5, examines the estimated gas prices for different smart contract methods under varied load scenarios (1, 20, 100, 200, and 500 API calls). The individualMintRRT API mints RRT tokens individually whereas bulkMintRRT API mints a large batch of tokens at once. Similarly, safeMintSBT API mints a single SBT token, and bulkMintSBT API mints a batch of tokens. The bulk mint methods for SBT and RRT tokens are tested by requesting to mint 3 tokens each in a single API call. From the figure, it is apparent that minting methods associated with SBT tokens have minimal fluctuation with varying loads, whereas RRT tokens minting could have some variability. Similarly, the other remaining APIs such as balanceOf which returns the token balance of the user, and tokenURISBT which returns the token URI containing the metadata of the SBT tokens perform consistently under varying load factors. However, getTokensOwnedByAddress has some variability. As these contract methods will be called numerous times from the application, it is necessary to have as minimal variability as possible for the API to be more stable.

As one of the core technical aspects of the prototype is its ability to mint a large batch of tokens in a given interval of time, it is necessary to reduce the costs associated with bulk minting as possible. Further detailed in Figures 6 through 9 are the minimum, maximum, median, and average gas prices for minting batches of 3, 10, 20, and 30 tokens, respectively. The design of the API call facilitates the minting of up to 30 tokens in a single operation, optimizing the process to reduce the costs associated with multiple transactions, thereby maintaining efficiency while keeping the cost within an acceptable range. These figures clearly illustrate that the cost of bulk minting is directly proportional to the number of tokens minted. Ethereum network is notorious for high gas fees. Considering a consortium or permissioned blockchain as an alternative could offer the academic community all the advantages of blockchain technology without the high expense. This approach would ensure the immutable recording of reviewer incentives while mitigating the financial burden of gas fees.

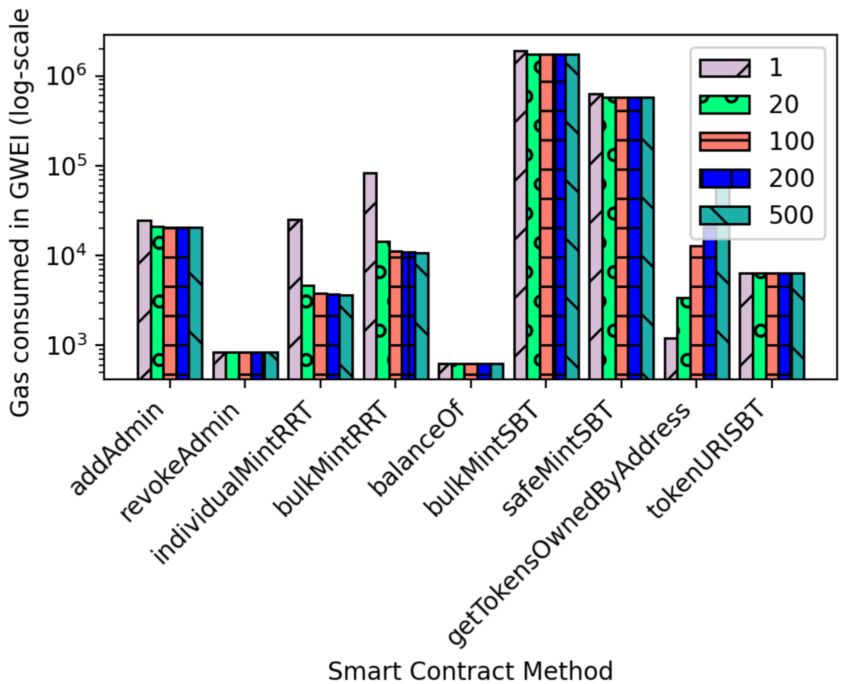


Figure 5: Gas estimate while bulk minting tokens for 3 reviewers with varying load factor

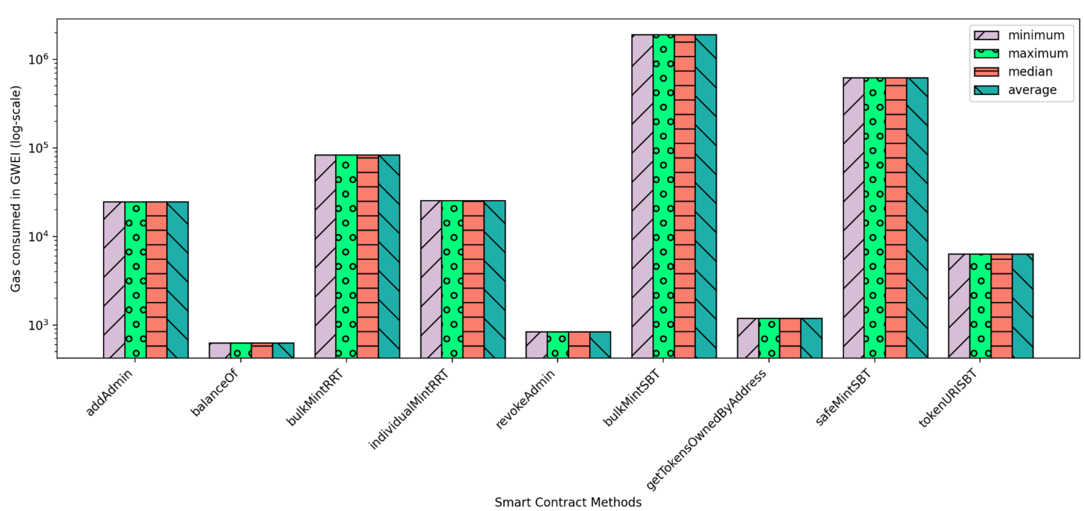


Figure 6: Gas estimate while bulk minting tokens for 3 reviewers

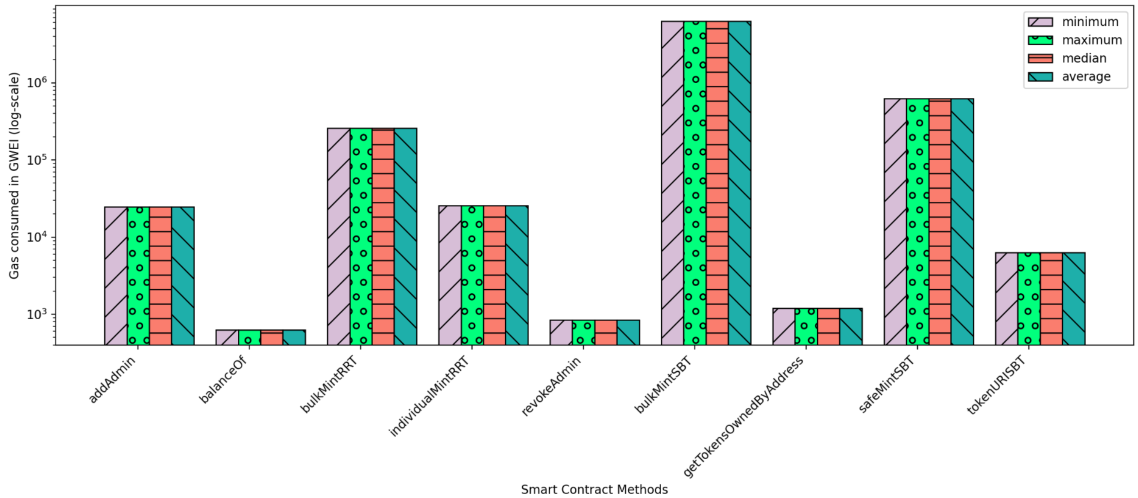


Figure 7: Gas estimate while bulk minting tokens for 10 reviewers

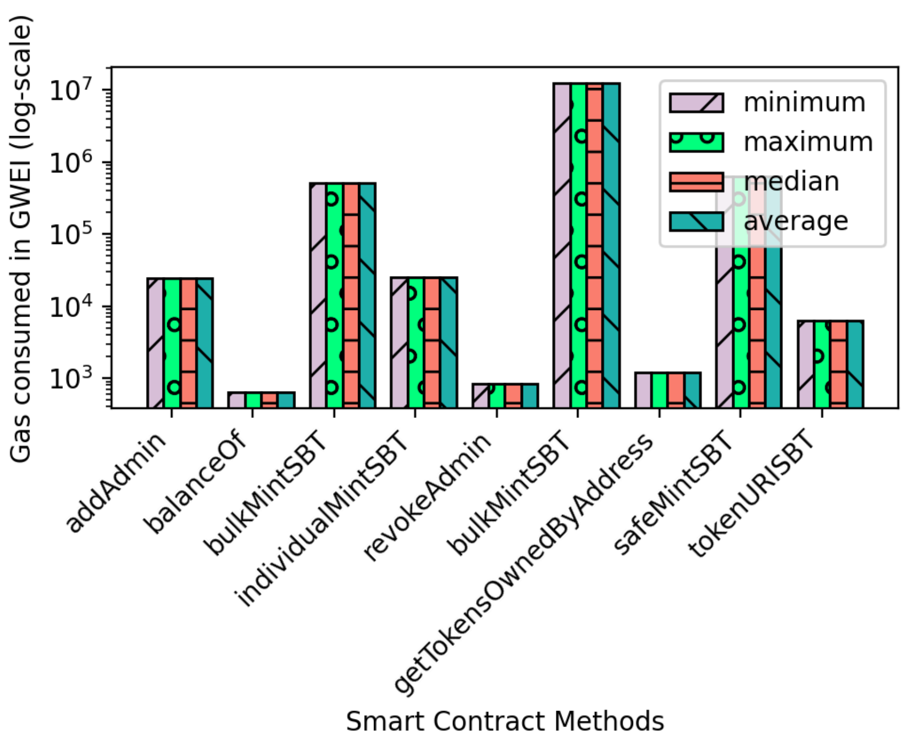


Figure 8: Gas estimate while bulk minting tokens for 20 reviewers

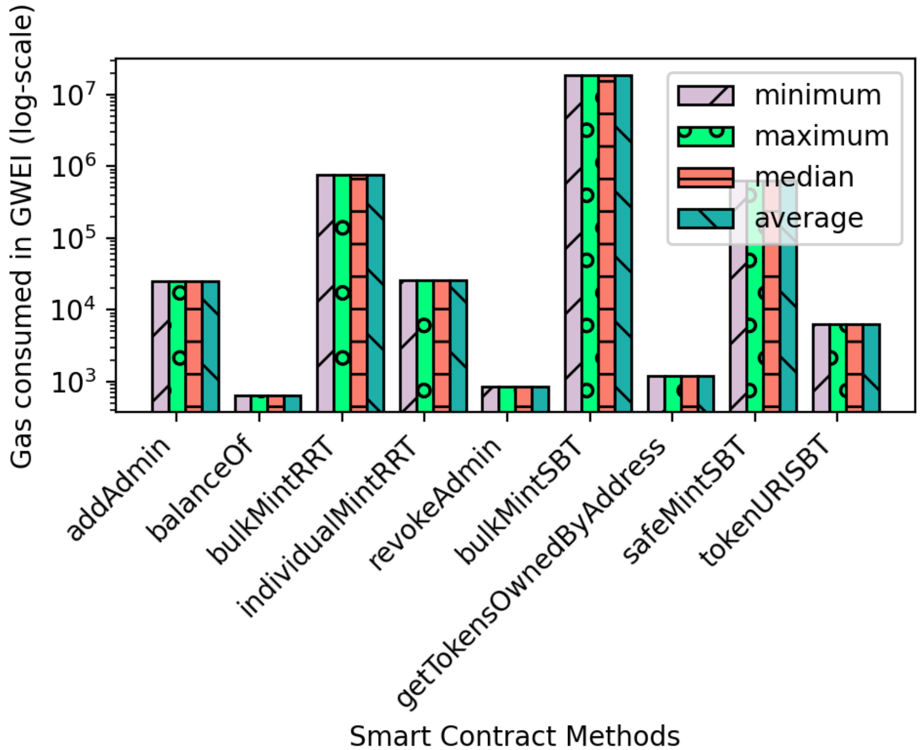


Figure 9: Gas estimate while bulk minting tokens for 30 reviewers

References

Avalanche (AVAX) blockchain - getting to know the Ethereum rival. (n.d.). Atato Custody. Retrieved February 19, 2024, from https://www.atato.com/avalanche-avax-blockchain-getting-to-know-the-ethereum-rival

ERC-20 Token Standard. (2023, November 19). ethereum.org. Retrieved February 19, 2024, from https://ethereum.org/developers/docs/standards/tokens/erc-20

ERC-721 Non-Fungible Token Standard. (2023, November 19). ethereum.org. Retrieved February 19, 2024, from https://ethereum.org/developers/docs/standards/tokens/erc-721

Get started with MetaMask Portfolio. (n.d.). MetaMask: The Ultimate Crypto Wallet for DeFi, Web3 Apps, and NFTs. Retrieved February 19, 2024, from https://metamask.io/

React. (n.d.). The library for web and native user interfaces. Retrieved February 19, 2024, from https://react.dev/

Solana vs. Ethereum: A Comprehensive Comparison. (2023, September 21). ZenLedger. Retrieved February 19, 2024, from https://www.zenledger.io/blog/solana-vs-ethereum/

Solidity. (n.d.). Solidity 0.8.24 documentation. Retrieved February 19, 2024, from https://docs.soliditylang.org/en/v0.8.24/

Tasca, P., & Tessone, C. J. J. (2017). AuthEthe. *arXiv preprint arXiv:1708.04872*. https://arxiv.org/pdf/1708.04872.pdf

Werth, J., Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N., & Pahl, C. (2023). A Review of Blockchain Platforms Based on the Scalability, Security and Decentralization Trilemma. *ICEIS (1)*, 146--155.

Weyl, E. G., Ohlhaver, P., & Buterin, V. (2022). Decentralized society: Finding web3's soul. *SSRN 4105763*. https://www.metaneo.fr/content/files/2022/06/SSRN-id4105763.pdf

What is a Utility Token? Utility Token Definition. (n.d.). CoinSmart. Retrieved February 19, 2024, from https://www.coinsmart.com/articles/what-are-utility-tokens/

What is IPFS? (2023, October 2). IPFS Docs. Retrieved February 19, 2024, from https://docs.ipfs.tech/concepts/what-is-ipfs/

What is Oracle Autonomous Database? (2024, January 23). Oracle Help Center. Retrieved February 19, 2024, from https://docs.oracle.com/en/cloud/paas/autonomous-database/serverless/adbsb/autonomous-intro-adb.html