BARIT: BLOCKCHAIN-BASED ANONYMOUS REVIEWER INCENTIVE TOKEN

Pratiksha Shrestha

A THESIS

Presented to the Faculty of Miami University in partial
fulfillment of the requirements
for the degree of

Master of Science

Department of Computer Science and Software Engineering

The Graduate School
Miami University
Oxford, Ohio

2024

Dr. Suman Bhunia, Advisor
Dr. Arthur Carvalho, Reader
Dr. Karen Davis, Reader
Dr. Liudmila Zavolokina, Reader

©

Pratiksha Shrestha

2024

ABSTRACT

Peer review is an integral part of academic publication necessary to maintain high standards and novelty of published research. Despite its importance, peer reviewers are rarely provided with incentives, leading to journals facing difficulties in finding reviewers willing to accept invitations and submit reviews on time. This thesis proposes a Blockchain-based Anonymous Reviewer Incentive Token (BARIT) to incentivize peer reviewers. We conducted interviews with academic researchers to identify the system requirements and understand the factors that motivate them to contribute as reviewers or editors. BARIT introduces flexible incentive schemes that provide both recognition and tangible benefits for reviewers' contributions while preserving the anonymity required by the venue's review process (open, single, or double-blind). By leveraging blockchain technology to record reward tokens, their permanence, immutability, and acceptance across different publishers are ensured. The incentive model aims to encourage researchers' involvement as reviewers, reduce invitation refusal rates, and prompt the timely submission of review reports. We demonstrated the designed solution to experts to evaluate its usability, benefits, and efficiency from their perspective.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my deepest gratitude to my thesis advisor, Dr. Suman Bhunia, whose insightful guidance, patience, and unwavering support have been instrumental throughout this research. I am also profoundly grateful to my committee members, Dr. Arthur Carvalho, Dr. Karen Davis, and Dr. Liudmila Zavolokina, for their valuable feedback and constructive criticism that significantly improved the quality of this thesis. A special thank you goes to Dr. Chad Anderson and Dr. Gabe Lee for their visionary input in shaping the direction of this project and providing consistent support. Their contributions were crucial in bringing this research to fruition.

I am forever grateful to my parents, whose unconditional love, encouragement, and sacrifices have been the driving force behind my academic pursuits. Their unwavering belief in me has been a constant source of motivation. To my sisters, Poonam and Pooja, thank you for being my pillar of strength, supporting and encouraging me whenever I felt lost. I would also like to extend my heartfelt thanks to my friends Aayog, Ashika, and Sampada, for always being there for me. Their friendship has been a source of comfort and encouragement.

Finally, I would like to acknowledge all those who have directly or indirectly contributed to this work. Their support and assistance have been invaluable, and I am truly grateful.

Chapter 1

Introduction

Publication of academic research plays a crucial role in advancing the scientific community. Researchers publish papers in various outlets like scientific journals, conferences, and other venues to share findings, discoveries, and innovations. These papers serve as valuable resources for professionals to improve their practices while allowing researchers to build upon previous work and make novel contributions [12]. Peer review by expert reviewers ensures that these papers are significant enough to warrant dissemination. Many researchers consider peer review to be essential to the evaluation and publication of new scientific knowledge [13]. Scholars have noted that the purpose of peer review is to reduce publication-related errors, such as the acceptance of papers that lack value to the academic community and the rejection of papers that would have been appreciated by the community [14].

1.1 Motivation

Academic journals and conferences, as outlets for the dissemination of knowledge, rely on experts to evaluate submitted manuscripts. These experts assess the relevance, methodological rigor, significance of the findings, clarity of writing, and adherence to ethical standards. They typically give their time to review papers with the expectation that their peers will provide the same service for their submissions. The process depends upon having enough experts willing to review the papers that are submitted, and the outlet editors being able to identify those willing reviewers. Unfortunately, editors have seen a significant increase in the number of submitted papers in recent years, which outpaces the corresponding number of reviewers willing to accept review invitations [15]. *Publons* data reveals a significant workload disparity in peer review. Just 10% of reviewers conduct nearly 50% of peer review submissions [16]. For the 2022 peer review cycle, *Cyberpsychology: Journal of Psychosocial Research on Cyberspace* reported needing to contact 22.6 reviewers on average to obtain two reviews per article - almost double the number compared to the previous year [17]. Of the requests sent, a substantial portion (33.1%) was declined, with an even larger percentage (52.4%) simply ignored whereas only 14.5% were accepted, and of those, 11.7% ultimately failed to deliver a review [17]. This situation hinders the ability of publication venues to perform a timely and thorough evaluation of the growing volume of research output.

Reviewing scientific work is a time-consuming process that competes with researchers' already heavy workload with teaching, research, and administrative duties. Reviewing is considered a voluntary act for the scientific community, termed a "culture of service" [18]. Yet, this intrinsic satisfaction of contributing to the advancement of knowledge can be overshadowed by the intense pressure of the "publish or perish" culture prevalent in academia. The number of papers requiring review vastly exceeds the number of available reviewers, as researchers typically prioritize publishing their own work over reviewing others'. Accepting a review invitation doesn't guarantee prioritization, and lack of motivation often leads to delayed reports, hindering the entire review process and eventual publication [19]. Remarkably, slow review times are one of the major complaints against the peer review process [20].

Few publishers offer monetary and non-monetary rewards to encourage active participation from their reviewers. These rewards include public recognition, free or discounted access to journals or conference proceedings, and submission fee credits. For instance, *Wiley Journals* and *American Psychology Association (APA)* collaborate with *Web of Science* to record and showcase peer review contributions [21, 22]. *MDPI* provides vouchers that can be used for manuscript submissions [23]. Additional incentives offered by other publishers are discussed in Section 2.1.5. These programs often face limitations, such as being confined to individual platforms and raising data security concerns due to their reliance on centralized servers.

We conducted interviews with researchers who have fulfilled roles as reviewers and editors across various journals and conference proceedings. Our aim was threefold: first, to discern the driving factors behind their engagement; second, to pinpoint shortcomings within current frameworks; and third, to devise strategies for constructing a system that empowers editors to efficiently identify and incentivize appropriate experts for manuscript review. The feedback revealed that researchers have different opinions on their motivation to perform reviews, all of which are important. Furthermore, editors commonly encounter challenges in recruiting willing reviewers who can deliver high-quality assessments in a timely manner.

1.2 Contributions

This research proposes a peer review system that provides editors enough flexibility in how the system is operationalized and allows them to implement appropriate incentives for reviewers while maintaining trust in the system's functionality. Blockchain-based Anonymous Reviewer Incentive Token (BARIT) stores the review process in a private database that provides flexibility to operate peer reviews with varying levels of anonymity. Editors are granted the ability to choose their preferred level of anonymity for peer review. We use blockchain smart contracts to manage review incentives. For each review submission, the reviewer is awarded a perpetual recognition certificate along with crypto tokens that can be redeemed for different services within the system. The issuance and value of these tokens are predetermined by the respective journal for which the review is conducted.

This thesis makes the following contributions:

- Introduces a public platform independent of individual publication outlets to facilitate engagement in the peer review process.
- Development of a flexible framework capable of supporting various peer review processes and incentivization models tailored to the varied needs of different publication venues.
- Suggests a hybrid system combining blockchain technology with an off-chain database to

ensure robust support for all peer review models with incentives while safeguarding reviewer anonymity.

- Introduces flexible incentivization schemes based on non-transferable certificates of recognition and spendable utility tokens. These tokens have varied values to address the different motivating factors that drive various reviewers.
- Implement and demonstrate the feasibility of the proposed design and make the code "open-source" through public repository [5].

Chapter 2

Background & Related Work

In this chapter, we discuss the necessary background for this research. The background includes an introduction to peer review and current challenges with the review process and incentivization for reviewers in Section 2.1 followed by blockchain technology, its features, and limitations in Section 2.2. In Section 2.3 we discuss the Ethereum network with smart contracts and concepts of tokenization in Section 2.4. We then describe the essential technologies such as cryptowallets and InterPlanetary File System. Some hybrid technologies with blockchain and traditional databases are discussed in brief in Section 2.7. Finally, in Section 2.8, we discuss some of the related works.

2.1 Peer Review Process

Peer review is a critical process in the academic publication that involves examining an author's scholarly work by experts in the same domain to evaluate its credibility, validity, and novel contributions [24]. It ensures that submitted manuscripts meet the high standards of the research domain and adhere to ethical and moral guidelines, thereby ensuring that research findings are fit for dissemination to the scientific community and the public. Peer review as the standard for determining what research is published in academic journals and conferences developed out of the increasing specialization of scientific work and growing competition for space in academic outlets [25]. Today, peer review is a core component of the academic research process as most scholarly papers are published only after being evaluated by a panel of experts who verify the quality of the work and its contribution to knowledge. A survey conducted by Mulligan et al. demonstrated that 84% of the researchers consider peer review a vital process in scientific publishing [26].

2.1.1 Purpose of Peer Review

The purpose of peer review can be broadly summarized into two primary objectives:

1. Peer review ensures that accepted papers meet specific criteria for quality, originality, adherence to standard procedures, and contribute to advancing scientific knowledge [24]. This helps maintain the integrity and trustworthiness of scholarly publications.

2. Peer review provides valuable suggestions and feedback to the authors to improve the quality of their paper and identify any errors that need to be corrected before publication [24]. Peer review facilitates an environment for constructive criticism, enhancing the research's clarity, accuracy, and overall impact.

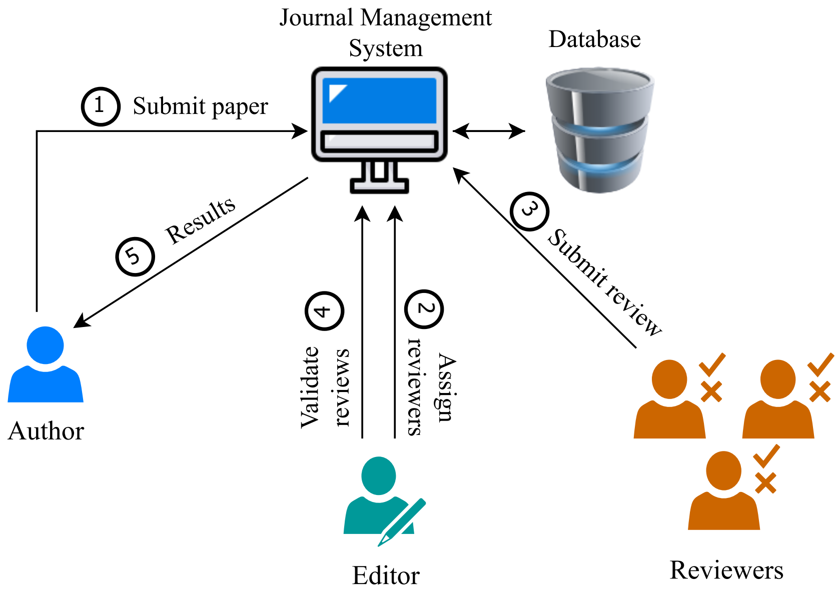


Figure 2.1: Visual representation of peer review process

2.1.2 Overview

A typical peer review process follows a series of well-defined steps that may vary slightly depending on the publication venue. Figure 2.1 illustrates the visual representation of the generalized peer review process. Once the research is complete with results and analysis, the author submits the paper to an appropriate publication venue. Many publishers use journal management software to facilitate the process and store submitted papers and other relevant details in a centralized database. Each venue has a specific set of requirements that must be met for a paper to be published.

The detailed flow of the peer review process is illustrated in Figure 2.2. The editor of the venue first analyzes the submitted paper to determine if it meets all the necessary criteria. The editor also checks for any signs of plagiarism. If the paper does not meet the venue's standards, the editor rejects it and informs the author of the outcome. If the paper meets the venue's requirements, the editor searches and sends review invitations to qualified experts. The reviewers read the papers carefully, examining the validity of the scientific hypothesis and ideas, implemented methodologies, and correctness of the reported results. Reviewers also evaluate the originality of the research and its potential contributions to the advancement of the field. The references and any minor or significant errors made in the paper are also checked. The detailed review reports are then sent back to the editor.

The editor examines the reports provided by all the reviewers. Based on the reviewers' recommendations, the paper may be accepted, rejected, or returned to the author for revisions. The paper is accepted if the report is positive and suggests no improvements or changes are necessary. If there are suggestions for improvements or error fixing, the editor informs the author and requests them to

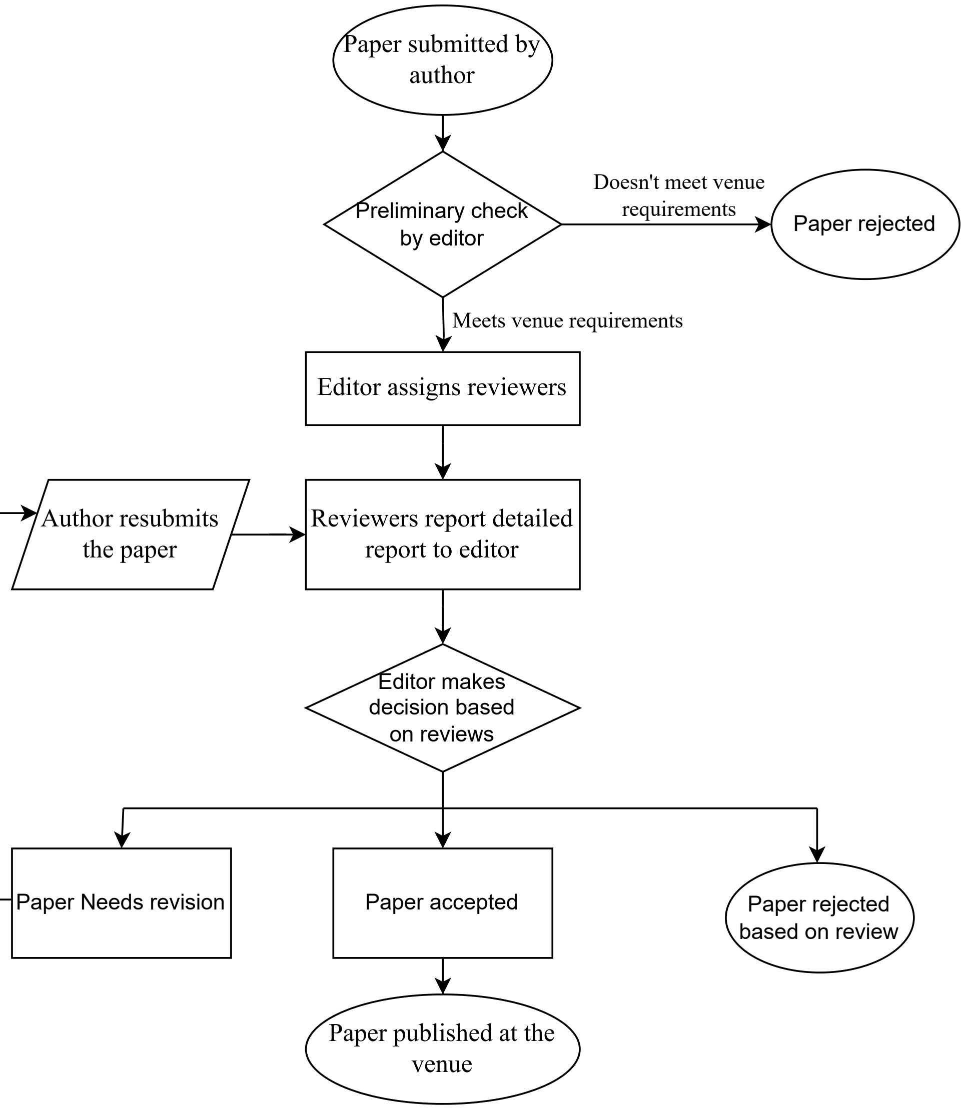


Figure 2.2: General publication flow with peer review [7]

make necessary changes and resubmit the paper. The resubmitted paper then undergoes the same review process. It is rejected if the report suggests that the paper should not be accepted. Finally, once a paper is accepted, it may undergo minor changes before being published in the venue.

2.1.3 Type of Peer Review

A peer review is often conducted in one of three formats: open, single-blinded, or double-blinded.

1. **Open peer review:** Open peer review is the process in which both authors' and reviewers' identities are fully disclosed. This approach can lead to more constructive feedback and

thoughtful critiques from reviewers [27]. However, reviewers might feel intimidated when reviewing work by prominent authors and also tend to tone down the criticisms [27].

2. **Single blind peer review:** In a single-blind peer review, reviewers are aware of the authors' identities, but the authors do not know who the reviewers are. As the reviewer is anonymous, they are more willing to provide honest feedback without feeling intimidated or influenced by the author's reputation [27]. However, there is a possibility that the reviewer may offer overly harsh criticism or delay the publication of the paper [27] to give their own research an advantage.

3. **Double blind peer review:** In a double-blind peer review, the identities of both the authors and reviewers are kept secret from each other. This review process has all the benefits of the anonymous reviewer and can reduce bias as the author's reputation doesn't influence the feedback [27]. Nevertheless, the reviewers might still provide unwarranted negative feedback. There's also a possibility of the reviewers being able to deduce the author's identity based on their writing style and niche field of study [27].

2.1.4 Incentivizing Peer Reviewers

The primary motivation for the reviewers is known to be intrinsic led by the satisfaction of serving the academic community that has supported them in the past. But, with the fear of "publish or perish" culture in academia, the number of papers that need to be reviewed far exceeds the number of available reviewers. Studies have shown that monetary rewards can sometimes negatively impact reviewer motivation and the quality of peer reviews [28, 29]. In contrast, non-monetary incentives, such as public recognition, free or discounted access to published articles, and credits towards publishing fees have been found to be more effective in motivating reviewers to accept review invitations [28, 30]. These incentives appeal to both early-career and experienced researchers by providing tangible benefits and recognizing their contributions to the peer-review process.

While outside the scope of this research, it is also important to consider incentives for honest, reliable reviews. Carvalho et al. discuss the use of proper scoring rules to encourage honest evaluations, especially when the quality of manuscript submissions is subjective [31]. They propose a Bayesian model for the peer review process, with rules that reward reviewers based on how closely their reported reviews align with those of their peers [31].

2.1.5 Incentive Models Used by Publishers

Table 2.1 provides a detailed list of incentives offered by popular publishers. Many publishers such as the *American Psychology Association (APA)* [22], *Association for Computing Machinery (ACM)* [33], *Wiley Journals* [21], *Sage Publishing* [40] partner with platforms such as *Web of Science* (previously known as Publons) [42] and *ORCID* [43]. These platforms allow researchers to track and showcase their peer review and editorial contributions across journals. Publishers such as *Copernicus* [35], *Elsevier* [36], and *Institute of Electrical and Electronics Engineers (IEEE)* [39]

Table 2.1: Different monetary and non-monetary incentive models used by publishers

Publisher	Recognition	Rewards
American Psychology Association (APA) [22]	APA provides integration with the Web of Science to provide credit for the reviewer's contribution.	–
American Society of Plant Biologists (ASPB) [32]	Associate editors are sent personal thank you notes.	Reviewers are awarded Journal Miles for each review they submit. The miles are redeemable for free merchandise and memberships.
Association for Computing Machinery (ACM) [33]	Reviewers can opt-in for Web of Science (Publons) service to track and record their review activities.	–
Colabra: Psychology [34]	–	Editors and reviewers have the option to either receive payment for themselves or volunteer their contribution to waiver funds or their institutions' open access fund.
Copernicus [35]	They publish an annual list of reviewers. They also have integration with Web of Science.	–
Elsevier [36]	Elsevier provides a Reviewer Recognition Platform where reviewers can create a public profile detailing all of their review contributions. Reviewers can also benefit from the end-of-the-year review report published by the publication house.	The reviewers receive one month of access to Scopus and ScienceDirect along with a range of discounts based on different 'badges' that they earn [28].
Hindwai [37]	Each year, reviewers who have reviewed for Hindwai receive reviewer badges and certificates as acknowledgment of their contribution.	–
Institute of Electrical and Electronics Engineers (IEEE) [38, 39]	Reviewers who reviewed for different IEEE journals or conferences are listed in the respective venue's reviewers list.	–
Multidisciplinary Digital Publishing Institute (MDPI) [23]	MDPI integrates with the Web of Service to provide reviewer recognition. The reviewers are awarded a Personalised reviewer certificate for their contribution.	Reviewers are eligible to receive discount vouchers that can be used for future submissions. Exceptional reviewers are awarded the Outstanding Reviewer Award which includes monetary values and discounts.
Sage Publishing [40]	Sage is connected with the Web of Science and ORCID for reviewer recognition.	For each review, reviewers receive a 60-day access to all its journals, a 25% discount on any SAGE book, and a 20% discount on Sage Author Services [28].
Springer [41]	Reviewers can opt-in for Clarivate services to record their research activities.	–
Wiley Journals [21]	The reviewers' review history can be synced with Web of Science and ORCID. Reviewers also receive a Reviewer Recognition Certificate once they complete the review assignment.	Reviewers who review for certain health journals are eligible to receive Continuing Medical Education (CME) credits; a non-curricular academic activity credit [28].

acknowledge their reviewers by publishing a yearly list of reviewers. Some journals such as *Hindwai* [37] and *Multidisciplinary Digital Publishing Institute (MDPI)* have also experimented with the distribution of reviewer badges and certificates. A culture of providing tangible rewards to reviewers apart from giving recognition is gaining more traction. In 2015, *Colabra: Psychology* offered a new reward system for researchers where the reviewers and editors could either "elect to pay themselves" [34] or volunteer their contributions to waiver funds or their institutions' open access fund. Similarly, other publishers such as *Elsevier*, *MDPI*, *Sage Publishing*, and *Wiley Journals* provide benefits such as discounts on their products or subscriptions, vouchers for future submissions, and academic activity credit [28]. These benefits are typically confined to individual publisher platforms. The reliance on centralized servers in such systems also poses the risk of a system outage, potentially resulting in permanent loss of accumulated rewards. This research aims to design a system that can effectively incentivize expert reviewers and resolve the drawbacks associated with current reviewer recognition and reward programs.

2.1.6 Current Challenges in Peer Review

Despite being the standard for scientific publication, the peer review process is subject to criticism for several limitations. This section aims to highlight some of these limitations.

- **Difficulty Finding Qualified Reviewers**: Peer review is a voluntary process whereby scholars give their time to review the work of others. The decision to participate in the peer review process is multi-faceted but is often based on feelings of reciprocity that because others have given their time to evaluate a scholar's work, that scholar must give their time to review the work of others. A growing expectation by universities to raise the bar for research productivity by their faculty, a phenomenon known as publish or perish [44], is resulting in many more papers being submitted for review. This increasing pressure to publish research results in a decreasing proportion of scholars participating as reviewers which makes the task of finding qualified reviewers willing to review very difficult.

- **Lack of Incentives**: Reviewing a paper is time-consuming and arduous, requiring considerable expertise and knowledge. However, there is hardly any mechanism to reward or compensate the reviewers for their contributions. Most reviewers perform such reviews due to a sense of duty and obligation, while others do it to be in sync with the latest advancements in their field [24]. A lack of proper incentives might make reviewers feel unappreciated, leading to decreased motivation and a decline in the quality of reviews.

- **Publisher Platform Specific Rewards**: There are a few publishers that provide some form of monetary or non-monetary compensation to the reviewers for their efforts as described in Section 2.1.5. These incentives are however limited to the platform used by the publishers. All the rewards earned are only valued within the publisher's platform. Moreover, all the rewards and associated information are stored in central servers which pose the risk of data loss due to unexpected system failures or outages.

- **Manipulation of Submitted Manuscripts**: The traditional peer review system is susceptible to manipulation by unethical authors. Such authors might make small changes to the original paper and resubmit it as a new one [45]. There have also been instances of authors gaming the system by reviewing their own paper [46].

2.2 Distributed Ledger and Blockchain Technology

Distributed Ledger Technology (DLT) is a system that operates without relying on a central authority to facilitate transparent sharing of information, even in cases where parties do not necessarily

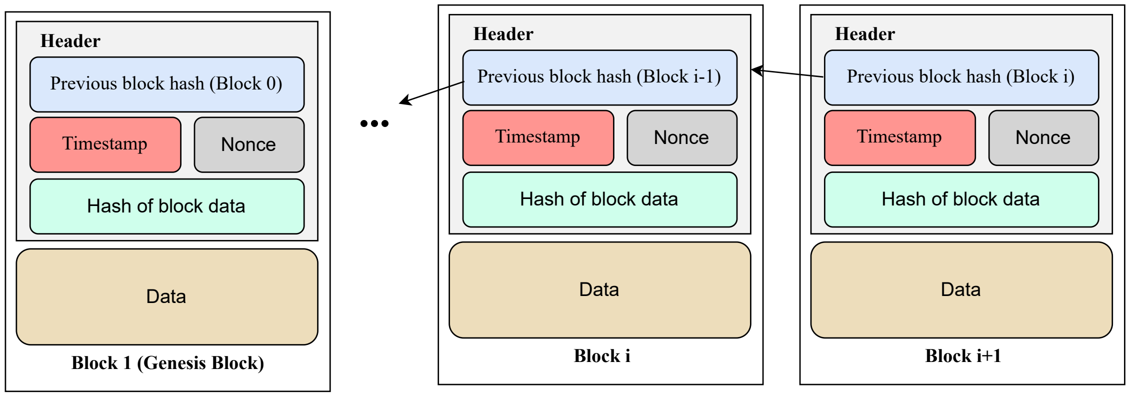


Figure 2.3: General structure of blockchain [8]

trust each other [47, 48]. One of the primary objectives of DLT is to eliminate the need for a trusted intermediary, such as a bank, to facilitate transactions. The consensus mechanism on which DLT is built ensures that all parties agree on the validity of shared data or transactions [47]. This consensus is achieved through complex algorithms that ensure the integrity and security of the data, even in an environment where some participants may be unreliable or malicious.

Blockchain is one of the most widely adopted forms of DLT which allows secure and transparent transactions without needing a trusted third party. In a blockchain, a distributed network of nodes maintains a database of transaction records. Each transaction is cryptographically signed and added to the chain as a new block [49]. The block also contains the link to the previously signed transaction. The structure of blockchain is represented in Figure 2.3. For legitimacy, validator nodes must validate each transaction before adding it to the chain. Blockchain is "append-only" and "immutable," meaning it can't be modified or deleted once a block is added. There are replicated copies of the ledger across the network. Any conflicts or discrepancies when adding a new block are resolved using a consensus mechanism [8].

2.2.1 Features of Blockchain

Blockchain has many desirable features that make it popular in wide sectors [48]. The primary features of blockchain are described below:

- **Decentralized:** Blockchain uses a distributed network of nodes, each maintaining a copy of the global ledger [48, 49]. As there is no central authority, there is no single point of failure.

- **Distributed:** All the users in the system are responsible for managing the ledger on the network. As the network is able to utilize the distributed computation power across the computers, the transactions are efficient [48, 49].

- **Consensus:** A pre-agreed algorithm is used to reach a consensus on the ledger's state [8]. This ensures that the replicas throughout the network are synchronized with minimal manual effort.

- **Secure:** Every block in the blockchain is secured using a cryptographic hash, making it highly difficult to tamper with [49].

- **Immutable:** In addition to being secured with cryptographic hashes, each block in the network is linked to the previous block [49]. Any modification made to a block would cause the chain to break and render all subsequent blocks invalid. Furthermore, because of its distributed nature, any attempt to modify a new block would require a consensus, and other nodes would detect the tampering and reject it.

- **Transparent:** All nodes in the blockchain can view and validate every transaction recorded in the network. Each transaction is cryptographically signed, providing a comprehensive record of data provenance [49].

Blockchain technology was initially the basis for Bitcoin and continues to provide the underlying infrastructure for a growing number of cryptocurrencies, but due to its properties such as decentralization, auditability, and security, it has also been applied in many other contexts including healthcare mutual aid services [50], internet of things sensor data [51], supply chain demand forecasting [52], used car sales [53], and videogame loot boxes [54]. Trust is a key requirement for many of these applications making blockchain an effective solution based on the trust that is inherent in blockchain's decentralized and distributed architecture. Specifically, blockchain is considered to be trust-free due to the fact that parties transacting on a blockchain have access to the same data. Consequently, they do not need to trust one another regarding on-chain data accuracy and integrity.

2.2.2 Types of Blockchain

Blockchain networks can either be fully accessible to everyone or have a specific permission model in place to determine who can have access. They can be divided into two categories: permissionless and permissioned. Table 2.2 provides a side-by-side comparison of permisonless and permissioned blockchain.

Permissionless

A permissionless blockchain network is public and open to everyone. In this type of network, anyone can join, issue transactions, and add blocks [1, 8]. The process of adding new blocks to the blockchain network is called mining. First, the miner nodes validate and include a transaction in a block. The permissionless blockchain network then uses a consensus mechanism such as proof-of-work or proof-of-stake to agree on which block should be appended next to the chain. Using a consensus mechanism helps reduce malicious behavior by incentivizing nodes that create legitimate blocks [8]. However, it requires the nodes to expend considerable computing resources and is considerably slow.

Bitcoin and Ethereum are some of the popular permissionless blockchain networks.

Table 2.2: Permissionless vs permissioned blockchain [1]

	Permissionless	**Permissioned**
Access Control	Anyone can participate.	User needs to be authorized.
Consensus Mechanism	Computationally intensive. e.g; proof-of-work, proof-of-stake	Less resource intensive e.g; Practical Byzantine Fault Tolerance (PBFT)
Network Performance	Cannot handle high transaction throughput. Has high latency and is less scalable.	Handles high transaction throughput. Has low latency and is highly scalable.
Security	Less secure.	More secure because of restricted access.
Governance	Truly decentralized with no central authority.	Has varying levels of decentralization as the network is governed by the organization.

Permissioned

A permissioned blockchain network differs from a permissionless one in that it is private and only accessible to a group of pre-approved users. An authority is responsible for granting permission levels to each user, allowing for tailored access to the network. Some users may be given access to only read while others may be able to validate transactions [1, 8]. In a permissioned blockchain, all transactions are still distributed and transparent, but the identity of the participating nodes is not anonymous, which enhances the trust between the parties. A consensus mechanism is still used but doesn't need to be as resource-intensive as the permissionless blockchain network.

Permissioned blockchains are helpful in industries where control, privacy, and confidentiality are crucial, such as finance and healthcare. Additionally, permissioned blockchains can be designed to comply with legal and regulatory frameworks, making it easier for businesses to meet their obligations. They are also more energy-efficient and faster than permissionless blockchains, as they require less computing power to reach consensus.

2.2.3 Consensus Models

The consensus mechanism is a process used in blockchain networks that ensures a single, consistent global ledger representing the actual state of all the transactions performed on-chain [8]. It is a critical component that enables trust between participants in the absence of a central authority. The consensus mechanism is necessary for blockchain because it ensures that transactions are validated and recorded accurately and that double-spending or other malicious activities are prevented.

Consensus is achieved when the majority of nodes on the network agree that a particular transaction is valid, allowing it to be added to the blockchain. Different blockchain networks use different consensus mechanisms. This document will discuss two commonly used consensus mechanisms, i.e., Proof-of-Work (PoW) and Proof-of-Stake (PoS).

Proof-of-Work (PoW)

In a Proof-of-Work consensus, the miner nodes mining a block must solve a complex mathematical challenge to validate the block. A significant amount of computational power is required to solve such a problem. When a block is mined, the miner can receive certain predetermined rewards in return for the resources exhausted [55].

A PoW consensus favors the ones with extensive computational powers and heavily wastes resources that could benefit other sectors. Moreover, PoW works on the assumption that the majority of the miners are truthful, making it susceptible to “51% attack” [56] in which a group of mining pools can gain a majority and carry out malicious activities like adding fraudulent transactions to the blockchain.

Proof-of-Stake (PoS)

In a Proof-of-Stake consensus model, users place stakes in the network instead of using computational resources. The nodes act as a validator. The validator for the next block is randomly selected, although nodes with higher stakes have a greater chance of being selected. Since nodes are required to place their stake, they are incentivized to be truthful while validating the block, as any fraudulent action can lead to the loss of their stakes. Validators receive specific transactional fees associated with the block once it is added to the blockchain [55].

Compared to PoW, the PoS model is less energy intensive as there is no need for an expensive setup. It is also more robust to “51% attack” as attaining a majority in PoS is very difficult. However, PoS is considered less scalable and unsuitable for large-scale applications, according to some researchers [56].

2.2.4 Blockchain vs. Centralized Databases

Traditionally, data has been stored in a single location with a central point of truth, utilizing a centralized architecture. In this setup, authorized users can create, read, modify, or delete data within traditional databases, with access levels typically granted and managed by a central authority [2]. The differences between blockchain and centralized databases are outlined in Table 2.3.

Blockchain technology, on the other hand, offers a more secure method of storing data due to its inherent characteristics of immutability, transparency, and decentralization. Despite these advantages, traditional databases are still more scalable and provide easier maintenance of user privacy.

There is no definitive answer as to which data storage option is superior. The choice between blockchain and centralized databases depends on the specific requirements of the application being developed. In many cases, applications leverage both technologies, using centralized databases alongside blockchain to benefit from both systems by appropriately distributing data between them.

Table 2.3: Blockchain vs centralized database [2]

	Blockchain	**Centralized Database**
Architecture	Peer to peer	Centralized client-server
Control	Decentralized without needing a trusted third party.	Centralized with administrators managing access privileges.
Access	Data is accessible by anyone in the network.	Users have permission-based access to data.
Immutability	Data cannot be altered once stored.	Data can be modified and deleted by authorized users. It is also susceptible to being exploited by malicious entities.
Robustness	Highly robust as data is distributed among nodes.	Data is stored in a central entity.
Performance	Slower due to decentralized propagation.	Faster and highly scalable.
Use Cases	Areas that require trust and transparency such as voting systems, supply chain, and notaries.	Storage of personal and confidential information, relational data, and large-scale data storage.

2.2.5 Limitations of Blockchain Technology

The advantages offered by blockchain have brought widespread interest in different sectors. However, blockchain faces several limitations that could impede its adoption which can be broadly categorized into scalability issues, resource consumption, privacy concerns, and regulatory challenges [57].

- **Scalability Issues:** Scalability is a major challenge in the implementation of blockchain technology. The consensus mechanisms used to validate each record can lead to congestion and increased latency during periods of high demand. As a result, the transaction processing speed of blockchain networks is significantly slower than that of traditional centralized systems. For example, traditional databases in the banking industry can handle approximately 1,700 transactions per second, whereas Bitcoin's blockchain can only manage about 7 transactions per second [57, 58]. Additionally, blockchain technology requires extensive computational resources to perform the complex calculations necessary for storing and validating information within the network. Efforts to enhance blockchain scalability, such as off-chain transactions and sharding, are currently in their early stages and demand substantial research to become viable solutions [57].

- **Resource Consumption and Cost:** As new users join a blockchain network, congestion increases, leading to greater infrastructure demands for processing transactions and consequently raising transaction costs [57]. The high costs associated with achieving Proof-of-Work (PoW) consensus have driven Ethereum's migration to Proof-of-Stake (PoS) [57].

However, even blockchain networks using PoS consensus can incur significant transaction costs, making them unsuitable for applications with a large user base performing thousands of transactions. Moreover, the append-only nature of blockchain means that there will always be a need for more and more storage [58].

- **Privacy Concerns:** While blockchain provides transparency and trust, it can also pose privacy challenges. In public blockchains, all transactions are publicly visible, which can conflict with privacy requirements for sensitive information [57, 58]. Although transactions are pseudonymous, sophisticated analysis techniques can potentially de-anonymize users, compromising their privacy. Private and permissioned blockchains aim to address these issues but often do so at the expense of decentralization and security.

- **Regulation and Compliance Risks:** The regulatory landscape for blockchain and cryptocurrencies is still evolving and remains fragmented across different jurisdictions. Without any central authority, it makes monitoring and enforcement of laws and regulations quite difficult. Moreover, different judicatory organizations have established different laws regarding blockchain [57]. This lack of uniformity creates uncertainty and can hinder the widespread adoption of blockchain technologies. Enforcement of rigid laws for regulation of cryptocurrencies as started by many countries will be helpful to establish trust and understanding.

2.3 Ethereum Network

The Ethereum network is an open-source, decentralized blockchain platform launched in 2015 [59]. Co-founded by Vitalik Buterin, it is designed to be a flexible and programmable blockchain. Ethereum allows developers to build applications and program logic that can automate the transfer of assets, manage digital identities, and enable peer-to-peer interactions [59]. A worldwide network of nodes supports the Ethereum network. Each node stores a copy of the blockchain and participates in the consensus mechanism, which verifies transactions and generates new blocks.

Ethereum network has the ability to create custom tokens. These tokens can represent any asset or value, including cryptocurrencies, commodities, or even virtual items in games. These tokens can be developed using the Ethereum platform's token standard, ERC-20, allowing for the creation of numerous new cryptocurrencies and Initial Coin Offerings (ICO) that utilize Ethereum's blockchain for their token issuance and management. Ethereum's native cryptocurrency, Ether (ETH), is used to pay transaction fees and incentivize nodes to secure the network.

2.3.1 Smart Contracts

Smart Contracts are computer programs that automatically enforce the digital contractual clauses when a certain condition is met [60]. Smart contracts were first introduced in 1994 as a computerized transaction protocol that executes the terms of an agreement [61], a decade before the invention of blockchain. Szabo compared smart contracts to a vending machine; just as vending machines can distribute sodas when a dollar bill is inserted without any intermediaries, smart contracts can be used to automate complex transactions in the financial domain [61]. However, at that time, the actual implementation of smart contracts was considered impossible as they were at risk of being breached by central authorities. After the introduction of Bitcoin, the possibility of real-world application of smart contracts was discussed once more with Ethereum being the first blockchain network to support the creation of smart contracts.

The smart contract offers the following benefits [60]:

- Smart contracts once deployed on blockchain cannot be altered. Smart contracts reduce the risk of malicious activities as every contract execution event is traceable and auditable.
- The costs associated with the intervention of third-party investigators or mediators can be reduced as smart contracts create trust in the execution of predetermined conditions automatically.
- Without the necessity of an intermediary, contract terms can be executed immediately resulting in a faster, more efficient turnaround time.

The terms of an agreement or a contract are first finalized by all the authorities involved. These contract terms are then converted into executable computer programs that represent all the clauses along with any logical connection between the terms. A smart contract is simply a digital agreement between parties that stores information, processes inputs, and produces outputs through predefined functions. For instance, a constructor function creates the contract, making the transaction sender the owner, while a self-destruct function typically allows only the owner to delete the contract [62]. Smart contracts are similar to a class in object-oriented programming with variables, functions, modifiers, events, and structures. Hosting a smart contract involves invoking the constructor through a transaction, storing the final code on the blockchain, and enabling users to call functions by sending transactions [62].

The life cycle of smart contracts is completed in four consecutive phases: creation, deployment, execution, and completion [60].

1. **Creation:** All the stakeholders discuss the rights, obligations, and prohibitions, drafting the initial agreement. Software engineers convert this agreement into computer language following a design, implementation, and validation process. This process undergoes several rounds of negotiation and iteration.
2. **Deployment:** Smart contracts are deployed to blockchain platforms making them immutable and accessible to all parties. If any modifications are needed, a new smart contract needs to be deployed. The digital assets of involved parties are locked by freezing their digital wallets.
3. **Execution:** After deployment, smart contracts monitor and evaluate their clauses, automatically executing functions when conditions are met. These contracts consist of declarative statements with logical connections that trigger transactions, which are then validated by miners and stored on the blockchain along with the updated states.
4. **Completion:** Once a smart contract is executed, the new states of all parties are updated, and all the changes are stored on the blockchain. Digital assets are transferred between the parties, unlocking the assets and completing the smart contract's life cycle.

2.3.2 Programming Languages

Different programming languages have been designed and developed for blockchain development. Solidity, Move, and Motoko are examples of languages built specifically for blockchain and smart contract development targeting specific blockchain networks [63]. Some popular all-purpose programming languages such as Rust, Go, and C++ are also popular in the blockchain community.

Solidity

Solidity is the most commonly used programming language for smart contract development, designed specifically for the Ethereum Virtual Machine (EVM) [64]. This high-level, object-oriented language features a syntax similar to JavaScript and TypeScript, making it accessible and familiar to web developers. Solidity's accessibility, precision, and flexibility contribute to its popularity among blockchain developers [63].

The code sample below is written in Solidity. It provides an example of the core concepts of Solidity such as constructors, variables, functions, modifiers, and events.

```
contract AdminContract {
    // Define a struct for admin record
    struct Admin {
        uint id;
        string name;
    }

    // Define state variables
    address public owner;
    uint public adminCount;
    mapping(address => Admin) admins;

    // Event declaration
    event AdminCreated(uint count, string name, address addr);
    event AdminRevoked(address addr);

    // Creating modifier to limit access to owner
    modifier onlyOwner() {
        require(msg.sender == owner, "Only the owner can perform this action");
        _;
    }

    // Constructor
    constructor(string memory _name) {
        owner = msg.sender;
        adminCount = 1;
        admins[msg.sender] = Admin(adminCount, _name);
    }

    // Add a new admin
    function addAdmin(string memory _name, address _account) public onlyOwner {
```

```
        adminCount++;
        admins[_account] = Admin(adminCount, _name);
        emit AdminCreated(adminCount, _name, _account); \\ emitting event on
            admin creation
    }

    // Remove an admin
    function revokeAdmin(address _account) public onlyOwner {
        delete admins[_account];
        emit adminRevoked(_account); // emitting event when admin is revoked
    }
}
```

2.3.3 Gas Fees

The term "gas" refers to the computational cost of executing smart contracts or transactions on the Ethereum network. Every operation on the Ethereum network requires a certain amount of computational power [65]. The users must pay for this power in terms of gas fees regardless of whether the transaction is successful or fails. There can be fluctuations in gas fees based on network demand and can be affected by factors like the complexity of the smart contract or the speed at which the transaction needs to be processed. Gas fees are an essential consideration for developers building on Ethereum or any other blockchain network and can significantly impact the cost and speed of their applications.

The native cryptocurrency of the Ethereum network is called ether or ETH. The transaction costs are usually a small fraction of ETH. Hence, these costs are usually represented in fractions of the cryptocurrency using denominations to make the calculation easier and economical [66]. Wei is the smallest denomination of ETH. 1 ETH is equal to 10^{18} wei. Gwei is a denomination that is equal to 10^{-9} ETH. Gwei is a middle denomination used to represent the gas fees as it can be used for values higher and lower. In May 2024, the median transaction cost on Ethereum was 5 Gwei which is equal to 0.000000005 ETH [66]. Other blockchain networks such as Avalanche and Solana have their own cryptocurrency and denominators.

2.3.4 Decentralized Applications (DApp)

A decentralized application or DApp is a web or smartphone-accessible application that connects with a blockchain network and smart contracts in the backend [67]. The backend runs on EVM whereas the users can interact via a web browser or smartphone application. The frontend interface can be built using any available frontend technologies such as JavaScript, HTML, and CSS [67].

2.4 Tokenization

A token is a digital asset that can represent any asset or value in the physical and digital realm [68]. Tokens include digital information that specifies the agreed-upon value or asset they represent, secured by cryptographic protocols. In blockchain, tokens are implemented with the help of smart contracts, that define the set of protocols and algorithms to facilitate the token creation, validation, and transfer. The smart contract validates the representative value and uniqueness of a token [69]. Tokenization is the process of transforming a real-world asset or data into a unique digital representation made up of a randomized sequence of characters i.e., a token [68, 69]. Tokens themselves do not hold economic value directly, as their market value is determined externally. Thus, tokens act as symbols validated by smart contracts within the blockchain system and can be used in various applications or traded.

A token is simply a digital reference that maps to the physical or virtual asset or services [69]. Tokens can be used to represent tangible assets such as currencies, physical assets (real estate, precious metals), or intangible assets such as rights, access, and services. Tokens can be broadly classified into two types based on their fungibility, i.e. whether they can be replaced or interchanged with another identical token.

2.4.1 Fungible Tokens

Fungibility of tokens means that each token content is similar as the other token content, making them interchangeable or replaceable such that they can be easily substituted by another token of equal value [69]. These tokens are identical to each other and can be split into smaller units without impacting their value. Technically, a fungible token is implemented as a list of blockchain addresses (user accounts) with associated quantities and a set of methods to manage that list, such as transferring tokens between addresses, along with rules governing who can manipulate the list [70]. In the Ethereum blockchain, ERC-20 (Ethereum Request for Comments #20) is a standard established by the community outlining specific functionalities and criteria for fungible tokens to operate correctly. Tokens following ERC-20 guidelines are divisible and indistinguishable from other tokens, promoting interoperability within the Ethereum blockchain community [69].

Fungible tokens are a great way to represent fiat currencies, cryptocurrencies such as Bitcoin and Ether (ETH), reward points, utilities, and services to name a few. However, fungible tokens can't be used to represent unique assets that can't be divided or exchanged.

2.4.2 Non Fungible Tokens

Non-fungible tokens (NFTs) are unique cryptographic tokens that have distinctive information and attributes that distinguish them from others [69, 71]. NFTs cannot be interchanged or divided, making them best suited to track asset ownership. The ERC-721 standard provides the framework necessary to represent, transfer, and track non-fungible tokens on the Ethereum network. ERC-721 defines that each NFT must have a universally unique identifier, which can be tracked and transferred using metadata [69]. NFTs represent ownership of digital or physical assets, allowing for individual tracking. With the help of blockchain networks as well as capabilities for delegation, transfer of ownership, and revocation [69].

Soulbound Tokens

The concept of non-transferable NFT tokens was first introduced by Vitalik Buterin, Ethereum's co-founder in 2022. Soulbound tokens (SBTs) are non-transferrable non-fungible tokens that are held by the accounts or wallets, known as souls [72]. Once delegated to a 'soul', SBT tokens can not be transferred but it is still possible to be revoked by the issuer. Soulbound tokens are used to represent assets that are permanently tied to an entity and cannot be traded or sold such as digital identity, awards and certificates, gaming progress, or in-game items.

2.5 Crypto Wallets

Crypto wallets are software applications that enable users to securely store and manage their cryptocurrencies, tokens, and other digital assets [73]. It serves as a bridge between the user and the blockchain network, allowing them to manage their digital assets by sending, receiving, and storing them. Crypto wallets store private and public keys that enable users to access and manage their digital assets on a blockchain network. The private keys are essentially the user's password and must be kept safe and secure to prevent unauthorized access to the wallet and its contents.

Saving private keys outside of a wallet is not recommended as it poses a significant security risk. Private keys are sensitive information that, if stolen or lost, can lead to the loss of the user's digital assets. Saving private keys on an internet-connected device or in a file on a computer can leave them vulnerable to hacking, malware, and other security breaches. Crypto wallets use various security measures such as encryption, multi-factor authentication, and cold storage to protect private keys and prevent unauthorized access to the user's digital assets. When private keys are stored in a physical devices, it is known as 'cold wallet' [73]. Cold wallets aren't connected to the internet, making them slow and expensive but very secure. Hot wallets are a piece of software that stores the private keys online, making them faster and more convenient for trading, but less secure.

MetaMask

MetaMask is a hot crypto wallet that allows users to interact with the Ethereum network and different blockchain applications directly through their web browser [74]. MetaMask is available either as a browser extension or as a desktop application. Users can use MetaMask to securely buy, send, and receive Ethereum tokens, store and manage their keys, and interact with smart contracts. It also allows for convenient switching between different Ethereum networks, including the mainnet, testnets, and private networks. MetaMask has become one of the most popular wallets for Ethereum users due to its ease of use, security, and compatibility with many popular decentralized applications (dApps). Table 2.4 provides a comparison of MetaMask with other popular hot wallets.

Table 2.4: Comparison of different hot wallets [3, 4]

	MetaMask	**Coinbase Wallet**	**Trust Wallet**
Platform	Browser extension, mobile app	Mobile app	Mobile app
User Interface	User-friendly interface but can be a bit technical.	Simple intuitive interface specifically targeted to Coinbase users.	Easy to use interface for beginners.
Security	High security as private keys are stored locally with strong passwords and backup seed phrases. Supports hardware wallets for added security.	High security with encrypted cloud backup of keys for recovery.	High security as private keys are stored locally. Supports hardware wallets for added security.
Privacy	No ID verification is required, preserving anonymity.	ID verification is required which compromises anonymity.	No ID verification is required, preserving anonymity.
dApp Integration	Extensive integration with Ethereum-based dApps. Often the default choice for interacting with decentralized finance (DeFi) platforms and NFT marketplaces.	Good support for dApps.	Features built-in dApp browser with focus on Binance Smart Chain.
Network	Specifically designed for Ethereum based networks. Can be extended to Polygon and Binance Smart Chain.	Supports multiple blockchain networks.	Supports multiple blockchain networks.
Support	Has a large user base and active community support.	Backed by Coinbase, offering robust customer support and resources.	Supported by Binance, with a large community and extensive resources.
Use Cases	Best for users deeply involved in the Ethereum ecosystem and interacting with dApps and DeFi platforms.	Ideal for users who use Coinbase exchange and want an integrated wallet experience with broad cryptocurrency support.	Great for users looking for a versatile wallet and strong integration with Binance Smart Chain.

2.6 InterPlanetary File System (IPFS)

InterPlanetary File System (IPFS) is the implementation of a distributed decentralized network for file storage and transfer based on systems such as Git and BitTorrent [75, 76]. IPFS works in a peer-to-peer network where all the files and other data are stored as IPFS objects in the local storage of the nodes. The nodes communicate with each other to initiate and transfer objects.

In a traditional database or file storage, location-based addressing is used where a file is retrieved based on its location. However, IPFS uses content-based addressing. In content-based addressing, file retrieval is done by requesting what data you want instead of pointing to a location.

Some of the properties of IPFS are described below:

- **Decentralization:** Instead of using a central server, files are stored across several nodes in the IPFS network [76].
- **Content-addressing:** IPFS uses content-addressing to identify and address files based on their content rather than their location on the network [76].
- **Versioning:** IPFS allows for the versioning of files, which means that different versions of the same file can be stored and accessed through unique content addresses [75].
- **Deduplication:** Even if multiple people publish the same file on the IPFS network, the file will be created only once [75].
- **Caching:** IPFS nodes cache frequently accessed files, making it faster to access them in the future.
- **Tamper resistance:** If a published file is modified, it will be detected by IPFS [75].
- **Peer-to-peer:** IPFS is a peer-to-peer network, meaning that nodes communicate directly with each other rather than through a central server [76].

2.6.1 Data Storage Implementation

IPFS objects are structured as Merkle Directed Acyclic Graphs, a cryptographically authenticated data structure. For each file, a unique cryptographic hash is created based on its content. IPFS stores all data as IPFS objects in chunks of 256 KB. If a file is larger than 256 KB, it is divided into chunks and thus has multiple IPFS objects. An IPFS object per file is created that contains the link to all the other IPFS objects that constitute the file [77]. The general structure of IPFS objects is shown in Figure 2.4.

Each file in an IPFS network has a unique hash. All the file retrieval and transfers are made by referencing the unique hash of the file. e.g., a node initiates a file request by using its unique hash. When the file is retrieved, its hash is compared with the requested hash value to ensure integrity of the file. If the hash of requested and retrieved files matches, it's safe to use. If the hashes do not match, it indicates that the file has been tampered with.

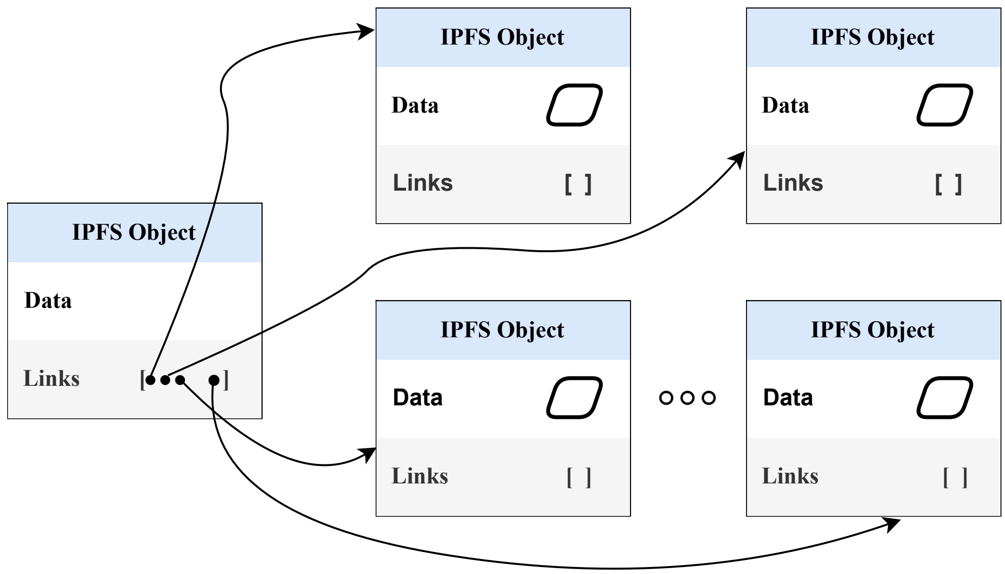


Figure 2.4: Representation of IPFS objects and links for multiple chunks adapted from [9]

When a node requests a file in the network, it is downloaded and stored in its own local storage. When another node requests the same file, it can be retrieved from all the nodes that currently have cached the file in their system [77], similar to torrent download.

2.6.2 File Versioning

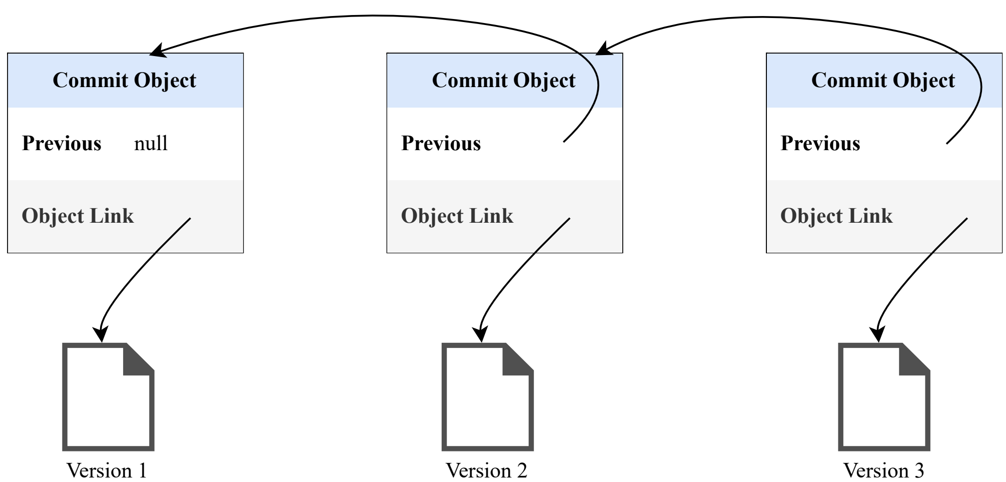


Figure 2.5: Example of IPFS object versioning adapted from [10]

Content-based addressing ensures that once a file is added to the network, it is immutable. IPFS supports file versioning for any changes made by storing all versions of the file [77]. When a file is published in the network, IPFS will create a new "commit object," which contains information on the previous commit and the links to the IPFS object of the file. The previous commit field is empty for the first version of the file. When an updated file is added to IPFS, a new commit object is added, which links to the previous commit. A graphical representation of file versioning in the IPFS network is shown in Figure 2.5.

The reliability, immutability, and file versioning capabilities of IPFS make it highly preferable for designing academic peer review systems. It is necessary to ensure that the paper cannot be tampered with after submission but can be revised and resubmitted when changes are suggested after the first round of review. The advantages of IPFS over traditional data stores are compared in detail in Table 2.5.

Table 2.5: IPFS vs. traditional data store

	InterPlanetary File System	**Traditional Data Store**
Resilience	IPFS network is distributed and decentralized. It means that there is no single point of failure.	All data is stored in a central server or location. In cases of outage or failures, the data becomes no longer accessible.
Geo-redundancy	Files once retrieved from the network are cached in the nodes' local storage. Nodes are distributed across different geographic regions. Hence, files in IPFS network have inherent geo-redundancy making them more reliable [78].	Traditional data store may implement redundancy and replications among multiple disks. But it is still limited to a certain geographic location [78]. An outage in that particular location will still make data inaccessible.
Verifiability	Due to content-addressing, any changes made to the file content after publication are easily detectable, providing data validation and verification.	Changes made to the file can not be detected.
Location dependency	Files are addressed based on their content. Changes to the file location don't have any impact.	If a file is moved to another location, its address changes and cannot be accessed with the previous address.
Performance	Data and files are replicated in multiple regions across a network of nodes. A node can fetch required data from the closest node that has the file, improving performance [79].	All the data is hosted in a single server making retrieval for users in distant locations slow.
Censorship	Files are distributed across multiple nodes in the network, making it highly difficult to censor content.	Traditional data stores are typically controlled by a single entity, making them vulnerable to censorship.

2.7 Integration of Blockchain Networks with Off-chain Databases

Blockchain technology offers several attractive features, such as decentralization, high security, data immutability, and provenance tracking, making it appealing for many data-based applications. However, it also presents challenges related to privacy concerns, performance limitations, and scalability issues, which traditional databases handle more effectively. Neither blockchain nor databases alone can adequately address all data requirements. However, by combining these two technologies, systems can harness their collective strengths in security, privacy, and efficiency [58].

Recent trends have increasingly focused on integrating blockchain with external databases to capitalize on their respective advantages while mitigating their challenges. One notable application of this integration is MedChain, a healthcare data-sharing framework [80]. MedChain allows for the efficient and secure sharing of medical data, ensuring its integrity and accessibility exclusively to authorized parties through blockchain technology. The actual patient data is stored in an external healthcare database. Similarly, Hjálmarsson et al. proposes a blockchain-based e-voting system where all transactions related to the voting process are securely recorded on the blockchain, while voter credentials are verified through a government-centralized database or identity verification services [81]. These applications show that combining blockchain and databases can facilitate the development of robust, efficient, and resilient systems.

2.8 Related Work

Significant research has been done on the application of blockchain in different sectors. This thesis focuses on incentivizing researchers to act as reviewers in the peer review process by offering rewards in the form of blockchain tokens. We discuss the related works in three scopes: the use of blockchain tokens for recognition and certification, blockchain in publication systems, and research done on motivating peer reviewers through incentives.

2.8.1 Blockchain-based Publication Systems

Following the emergence of Bitcoin [82], the first blockchain-based cryptocurrency system, researchers have explored the potential of blockchain for developing other systems that can benefit from its unique capabilities. Blockchain technology offers a promising solution for the academic publishing industry by making the entire process more transparent, accountable, and efficient. In a review by Leible et al. [83], the advantages of blockchain for scientific publishing are discussed, including efficient dissemination of research, prevention of single points of failure, promotion of equality in participation, and security from intellectual property plagiarism through timestamp based transaction recording to the chain as soon as they are created. Mackey et al. [84] proposed a blockchain-based shared-governance framework for scientific publishing, which implements a proof-of-authority (POA) consensus mechanism. Confidential information and files, such as submitted papers and reviews, are stored in an off-chain database. The paper aims to make the overall publishing process transparent and democratic by ensuring the inclusion of all stakeholders.

ARTiFACTS [85] aims to leverage blockchain to create a platform for researchers to keep permanent, immutable records of all their scholarly contributions, from ideation to review, revisions, and citations. It focuses on providing proof-of-existence of research work from the beginning and encourages research sharing, tracking, accessibility, and accreditation of the scientific work achieved.

In CryptSubmit, manuscripts submitted by the authors are securely timestamped along with the reviews and other submissions using blockchain technology [86]. The primary objective of this approach is to safeguard intellectual property by preserving a tamper-proof record of timestamped manuscript submissions that can be traced back to the original owner, i.e., the author. This approach ensures that scientific work remains protected against potential plagiarism due to reviewer dishonesty or data leaks.

TimedChain, based on a permissioned blockchain system, proposes to create a secure, decentralized, and efficient platform for managing the editorial process of academic publications [87]. Smart contracts are used to automate various aspects of the editorial process. It integrates effortlessly with the existing database of the publishers. Publishers themselves would be responsible for appending new blocks in the blockchain. They would be encouraged with a Proof-of-Authority based incentive mechanism that provides a reputation score based on their contribution to managing and maintaining the publication network.

In [88], Tenorio-Fornes et al. propose a blockchain and IPFS-based open peer review publication platform for transparent and immutable open-access publications. The research provides a proof-of-concept and also discusses the possibility of an incentive mechanism to reward reviewers as well as security and privacy for single and double-blind peer reviews. However, these topics are only mentioned and not explored in detail.

All research introduced in this section addresses various challenges in research publication using blockchain technology, such as plagiarism protection, fair participation, and accessibility with blockchain technology. However, they do not specifically focus on the peer review process or on incentivizing the reviewers, which is the main objective of our research.

2.8.2 Blockchain Tokens for Recognition and Certification

Recently, there has been much research exploring the potential of emerging blockchain technology to enhance review and recognition systems. A decentralized rating framework using a public blockchain network is introduced in [89] to reward users for submitting reviews. Reviewers earn fungible utility tokens redeemable for discounts at registered businesses like restaurants and shops. The number of tokens awarded is proportional to the reviewer's reputation score, calculated based on past review activity [89]. The Soulbound Token Certification (SBTCert) Verification System utilizes the decentralized nature of blockchain to issue and verify educational certificates as non-transferable soulbound tokens [90]. In [91], the use of blockchain-based tokens with incentivization schemes is proposed to attract qualified peer reviewers in information system conferences.

2.8.3 Incentivizing Peer Reviewers

In PubChain, an incentive scheme based on blockchain is proposed to motivate the primary stakeholders in the publishing ecosystem (authors, reviewers, and readers) to perform their respective tasks [92]. The paper proposes the development of a free open-access platform that can be trustworthy through the implementation of a rigorous peer review process. The reviewers are rewarded with financial benefits in the form of PubCoins that will be pegged against popular cryptocurrency tokens such as BitCoin. Spearpoint also proposes the use of cryptocurrency to provide a monetary incentive to the reviewers [93]. However, adding financial incentives to scholarly work could have a negative impact, especially since reviewing papers is considered prestigious volunteer work for reputed researchers.

Pluto proposed a decentralized publishing platform that utilizes smart contracts and tokens on Ethereum's blockchain to control copyright over submitted scientific works using Digital Object Identifiers [94]. It implements a "reputation score" based on the user's contribution, such as submitting research works, and proper evaluation (review) of others' work. The review process includes a blind period when the authors and reviewers are anonymous, followed by a public period when all the reviews are made public for evaluation. The platform also shares the content directly on its online platform, bypassing the traditional "journal" approach. This could limit adoption by scientists who focus on publication in recognized academic media such as journals and conference proceedings.

Khan et al. [95] proposed a Peer-To-Peer publication model, which allows authors to publish parts of their research incrementally, using a consortium-based permissioned blockchain network led by publishers. The approach aims to reduce the pressure of publishing "good results" and improve researchers' portfolios. Reviewers are granted authorship of the paper they review to encourage fair and constructive feedback, and reputation scores based on research and review ratings are weighed based on citations.

EUREKA is another decentralized peer review application that provides a token-based incentive mechanism to motivate authors and reviewers [96]. When a paper is accepted, the author, linked researchers, and reviewers are rewarded with crypto-tokens named Eureka Tokens (EKA) based on their contributions. The EKA tokens can be redeemed for a variety of actions, such as submitting papers for review, submission fee payment, and voting for awards, to name a few. Similarly, Orvium [97] also provides a token-based incentive mechanism (Orvium tokens) in a transparent and open peer review system. Orvium aims to improve transparency, reduce access costs to scholarly publications, and better reward mechanisms for reviewers.

The current approaches for incentivizing reviewers are primarily focused on providing tangible benefits, such as financial rewards [93], discounts, and coupons that can be redeemed for certain services on the platform [96] or by building their reputation score in an open peer review system. However, these methods may only partially align with researchers' expectations and requirements. Additionally, the reviewed literature primarily discusses incentivization in an open peer review system or, in the case of Pluto, only a blind period. This research proposes an incentive mechanism appropriate for a double-blind peer review system and also prioritizes the reviewers' reputation and prestige along with financial rewards.

2.8.4 Privacy-preserving Peer Review Systems

Blockchain technology has been increasingly explored in recent research works to promote trust and fairness in the publication system through open-access publication and open peer-review models. While blockchain's distributed, decentralized, and transparent nature has the potential to promote trust and fairness in the publication system, maintaining complete transparency also poses certain disadvantages. The double-blind peer review process is highly regarded by many researchers as the gold standard for scientific publication, with anonymity and privacy being crucial for both authors and reviewers. However, preserving anonymity and privacy in an open and transparent blockchain system is challenging.

Open-Pub [98] is currently the only known study that addresses privacy concerns in blockchain-based peer review systems. Open-Pub uses Threshold identity-based group signature (TIBGS) and asymmetric encryption to ensure the privacy and anonymity of authors and reviewers in a double-blind peer review system. TIBGS is a variation of group signature and has n group managers instead of one. Instead of a single group manager generating a group private key, each manager holds a portion of the private key. Group users need to get private key portions from threshold k managers to sign anonymously on the group's behalf. Authors use TIBGS to submit their work anonymously. The validators (editors) then encrypt the submitted paper using the reviewers' public key and send it for review. Reviewers then submit the review with their real identity without remaining anonymous. Once the paper is accepted, validators reveal the identity of the author. This way, authors and reviewers remain anonymous during the review period but are revealed to the public along with reviews submitted by the reviewers once the review period is over. OpenPub, in this regard, does not offer a "truly" blind peer review system but nevertheless provides a strong privacy mechanism.

Although these endeavors collectively aim to enhance transparency, reduce access costs to scholarly publications, and improve reward mechanisms for reviewers, they predominantly operate within an open peer review framework or a partial blind review. Additionally, they often adopt a rigid structure for the review process and incentive model, either as standalone platforms or by mandating conformity from journals and other publication venues. The novelty of this research lies in its emphasis on providing a flexible platform for participating journals, offering customizable review processes and reward options, all while upholding the integrity and permanence of rewards facilitated by blockchain technology. Journals can decide the amount and value of the tokens distributed as rewards. Once reviewers are rewarded, they are free to redeem accumulated tokens for benefits from any participating publisher that accepts them. Table 2.6 provides an overview of the scope covered by related research works and compares it to the overall scope of our designed solution.

Table 2.6: Past research on blockchain-based peer review system and tokenization for reviewer incentives

Artifact Name: Title	Open	Single-blind	Double-blind	Tokenization	Recognition	Rewards	Multi peer-review support	Standalone platform	Configurable Incentives	Summary
A framework proposal for blockchain-based scientific publishing using shared governance [84]	✓	✓	✓	✓	×	×	✓	×	–	Blockchain based shared-governance model (agnostic to specific publisher) for scientific publishing, implementing proof-of-authority (POA) consensus mechanism. Uses private sidechain depending on the type of review process and off-chain database for files. All information is revealed after peer review is complete and added to the public blockchain.
ARTiFACTS Launches First-Ever Blockchain-Based Platform for Scientific and Scholarly Research [85]	–	–	–	×	×	×	–	✓	–	Leverages blockchain to keep permanent, immutable records of all scholarly contributions, from ideation to review, revisions, and citations. Provides proof-of-existence of research work from the beginning and encourages research sharing, tracking, accessibility, and accreditation of the scientific work achieved.
CryptSubmit: Introducing Securely Timestamped Manuscript Submission and Peer Review Feedback Using the Blockchain [86]	✓	×	×	×	×	×	×	✓	–	Safeguard intellectual property by preserving a tamper-proof record of timestamped manuscript submissions that can be traced back to the original owner, i.e., the author. This approach ensures that scientific work remains protected against potential plagiarism due to reviewer dishonesty or data leaks.
Towards a decentralized process for scientific publication and peer review using blockchain and IPFS [88]	✓	×	×	×	×	×	×	✓	×	Proposes a blockchain and IPFS-based open peer review publication platform for transparent and immutable open-access publications. The possibility of reviewer incentive mechanism and security and privacy concerns for anonymized peer review systems is briefly discussed without implementation.
Pubchain: A decentralized open-access publication platform with participants incentivized by blockchain technology [92]	✓	×	×	✓	×	✓	×	✓	×	Proposes an incentive scheme where reviewers are rewarded with financial benefits in the form of PubCoins. Submission fees paid by the author are distributed among reviewers, authors of papers cited, and miners. Authors and reviewers are paid rewards based on the reader scores of their papers and reviews. Incentive mechanism is only proposed but not implemented yet.
Pluto: White Paper ver 0.4 [94]	×	×	✓	✓	–	–	×	✓	×	Pluto shares the content directly on its online platform, bypassing the traditional "journal" approach. It implements a "reputation score" based on the user's contribution, such as submitting research works, review submissions, etc. The review process includes a blind period when the authors and reviewers are anonymous, followed by a public period when all the reviews are made public for evaluation.
A Peer-To-Peer Publication Model on Blockchain [95]	✓	✓	✓	✓	✓	×	×	✓	×	Publishers can create their own consortium-based permissioned blockchain model. Allows authors to publish parts of their research incrementally. Reduces the pressure of publishing "good results" and improves researchers' portfolios. Authors get reputation scores based on their research whereas reviewers get their scores based on the user's rating for their review report's quality. Reviewers also get authorship on the papers they review.
EUREKA: A blockchain-based scientific publishing platform [96]	✓	×	×	✓	✓	✓	×	✓	×	Provides a blockchain-based publication platform where authors can submit their manuscript for review by paying tokens. Once submitted, the system (convenience layer) selects an adequate list of reviewers who can then submit reviews. The numeric rating is stored in a smart contract whereas the raw text is stored in the convenience layer (MongoDB, Node.js server, private Ethereum node). When a paper is accepted, the author, linked researchers, and reviewers are rewarded with crypto-tokens named Eureka Tokens (EKA) based on their contributions. EKA tokens can be used to represent awards, credit for article submission fee payments, and vote for awards and prizes.
Orvium Whitepaper: Accelerated scientific publishing [97]	✓	×	×	✓	×	✓	✓	×	×	Orvium provides a transparent and open peer review platform with the aim to reduce the publishing and access costs to scholarly publications. Organizations, authors, or anyone can form a Decentralized Autonomous Journal and determine the governance rules. Authors have full control and copyright of their research submissions. When submitting an article, authors can skate Orvium (ORV) tokens to encourage peer review. Reviewers can review manuscripts at any time and if there is a remaining stake, they are rewarded Orvium (ORV) tokens for their contribution. ORV tokens can be used for manuscript submission later.
Open-Pub: A Transparent yet Privacy-Preserving Academic Publication System based on Blockchain [98]	✓	✓	✓	✓	–	–	×	✓	×	Open-pub provides a blockchain-based transparent and fair peer-review platform with privacy-preserving mechanisms. Open-Pub uses Threshold identity-based group signature (TIBGS) and asymmetric encryption to ensure the anonymity of authors and reviewers in double-blind peer reviews. Authors use TIBGS to submit their work anonymously. The validators (editors) encrypt the submitted paper using the reviewers' public key and send it for review who review them with their real identity. Once the paper is accepted, validators reveal the identity of the author. This way, authors and reviewers remain anonymous during the review period but are revealed to the public along with reviews submitted by the reviewers once the review period is over. The incentive mechanisms aren't discussed in this paper.
BARIT: Our proposed solution	✓	✓	✓	✓	✓	✓	✓	✓	✓	BARIT provides a flexible peer review platform supporting all peer review types. The reviewers are incentivized in two forms: 1) soulbound tokens as perpetual certificates of recognition for their contributions; and 2) fungible reward tokens that can be redeemed for various publication-related fees. Publishing outlets have the flexibility to set reward policies that reward timeliness in review submissions.

Chapter 3

Understanding User Perspective on Peer Review

This research aims to leverage blockchain technology in designing an incentive model that can be appealing to expert reviewers. The resulting platform Blockchain-based Anonymous Reviewer Incentive Token (BARIT) is a flexible and trustworthy peer review system that motivates the reviewers by providing them with recognition and rewards while keeping their identity preserved. In this chapter, we discuss our motivation for BARIT, our goals and objectives along with the key requirements and design principles derived from expert interviews.

The research is carried out using design science research methodology (DSRM), proven effective for solving difficult problems through the creation of innovative artifacts [99]. Specifically, the DSRM proposed by Peffers et al. [11] is used, which involves six steps: problem identification, defining solution objectives, artifact design and development, artifact demonstration, artifact evaluation, and communication. The adapted methodology is illustrated in Figure 3.1.

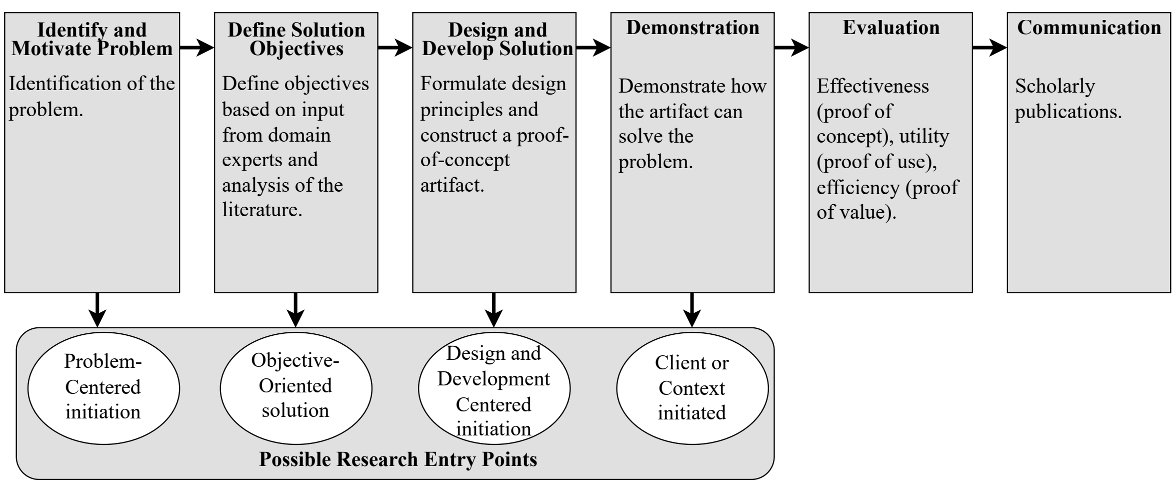


Figure 3.1: Design science research methodology applied to this research [11]

3.1 Problem Identification from Interviews with Experienced Researchers

As the first step for this research, we conducted interviews with experienced researchers to gain more insights on the issues that are relevant for academic community members with regard to their participation in the peer review process. This research has been reviewed and approved by the Miami University Research Ethics and Integrity Office. All the interviews and data handling are conducted in accordance with the policy set by Miami University's Human Subjects Research policy[1].

3.1.1 Recruitment of Participants and Data Collection

In order to better understand the current state of the peer review process, we conducted semi-structured interviews with ten individuals from the academic community who had experience as reviewers and/or editors for academic publications. These participants were invited through email invitations and represented a diverse range of roles, including reviewers, associate editors, and editor-in-chief. Their academic ranks span from Assistant Professor to Full Professor, ensuring a broad spectrum of perspectives from scholars at various stages of their careers. Table 3.1 provides an overview of the characteristics of our study participants.

Table 3.1: Study participants

Participant	Reviewing Role	Discipline	Academic Position	Gender
P1	Reviewer	Computer Science	Assistant Professor	Male
P2	Reviewer	Computer Science	Assistant Professor	Male
P5	Reviewer	Computer Science	Assistant Professor	Female
P7	Reviewer	Computer Science	Assistant Teaching Professor	Male
P6	Associate Editor	Computer Science	Associate Professor	Male
P8	Associate Editor	Gerontology	Assistant Professor	Female
P9	Associate Editor	Computer Science	Associate Professor	Male
P3	Editor-in-Chief	Information Systems	Professor (retired)	Male
P4	Editor-in-Chief	Information Systems	Professor	Male
P10	Editor-in-Chief	Statistics	Professor	Female

Researchers' perspectives on the academic reviewing process can vary widely, influenced by their diverse experiences across different venues. Different intrinsic and extrinsic motivations drive researchers to participate as reviewers, shaped by their career trajectory, aspirations, and other factors. Researchers who have served as reviewers may have different insights on potential system

[1]https://miamioh.edu/policy-library/academics/research/research-involving-human-subjects.html

improvements compared to those who have also participated as editors. As editors are well-versed in the challenges of finding qualified reviewers, their insights for improvements could differ significantly from scholars who have only worked as reviewers.

It is necessary to have a broad understanding of different issues and challenges faced by all participants in peer review irrespective of their roles. This study primarily aims to devise effective incentives for reviewers, thereby boosting review acceptance rates and ensuring high-quality submissions. However, motivating reviewers alone isn't enough to gain support from publishers. The system should align with the operating standards and incentivization policies of participating journals, conferences, and workshops, while also facilitating the process of finding expert reviewers for editors.

We tailored our interview questions to reflect the distinct roles of our participants. For reviewers, our focus encompassed the following key areas:

1. Their motivation for serving as a reviewer.
2. Current state of peer reviewing platforms along with any forms of compensation they've received for their contribution as a reviewer.
3. Their perspective on compensating reviewers and their preferences regarding forms of rewards.

Similarly, interviews with editors were focused on these points:

1. Their motivation for serving as an editor.
2. Current state of peer review and the challenges they've experienced in finding expert reviewers in time.
3. Their perspective on compensating reviewers, forms of rewards as well as metrics that could be beneficial for evaluating the quality of the submitted review.

All interviews were conducted via Zoom[2], with participants' consent obtained for recording. Transcripts were anonymized and stored following strict security protocols. The interview structure is detailed in Appendix A.1.

3.1.2 Findings from the Interviews

We asked participants what motivates them to contribute their time as reviewers or editors. We found that responses fit into one of four categories: learning, promotion requirement, quid pro quo, and service to the community. For some, the opportunity to see the latest research topics and methods and discover new ideas for their own research was a strong motivator. This aligns with the findings of other studies that have found relevance of the topic to scholar's own work and the opportunity to learn something new as motivating factors for accepting a review request [30]. The feeling that participation in the peer review process was an important and necessary service to

[2] `https://zoom.us/`

the academic community also motivated many of our participants. This too aligns with previous studies that found over 90% of scholars were motivated to contribute time to peer review to benefit their academic community [13]. Additionally, some of the junior faculty saw their participation in the peer review process as an obligation for promotion and tenure and some took a quid pro quo perspective contributing their time to review the work of others because others had contributed time to review their work. A selection of quotes associated with each category is provided in Table 3.2.

Table 3.2: Motivations to engage as reviewers or editors

Participant	**Category**	**Quote**
P1	Learning	To see what other researchers are doing. I see reviewing as a platform that I can continue my research.
P5	Learning	I can understand the development of the current research area or my related research area and sometimes I can get new ideas.
P4	Learning	I can keep up with all those new concepts, research methods and the idea and the topics. I think that really benefit me, allows me to stay up to date about the current research.
P2	Promotion requirement	Something that I can put on the annual review, on the tenure review.
P7	Promotion requirement	Reviewing papers is a service and service is counted toward tenure and promotion.
P1	Quid Pro Quo	It feels bad when you spend so much time for your research, you submit your paper and there is no reviewer accepting to do the review for your work. So, I try to help in that capacity to participate in this process.
P3	Quid Pro Quo	You do reviews as a service and in return, someone reviews your work as a service. So, it's really a social capital issue. You make contributions to the pool of intellectual assets that will be reviewing work and you'll receive reviews.
P5	Service to the community	I think serving as a reviewer is a contribution to this community.
P6	Service to the community	Nurturing the next generation of researchers to see their work published.
P10	Service to the community	You have to contribute to your discipline in increasing levels of authority where you can use your experience, so it's just a natural progression.
P4	Service to the community	I feel like it is a contribution to the field. I can help the authors go through the review process and publish their good research.

Several of the editors we interviewed noted the increasing number of submissions they are required to handle. P8 explained that in her editorial role the journal *"sends me emails almost every single day"* and P9 said he receives 3 requests per day from just one journal. P10 said she handles an average of 170 papers per year, but the most extreme case was P4 who as editor-in-chief said, *"in a year, I tend to handle between 900 and 1,000 manuscripts."* For editors, the challenge is not just an increasing volume of papers. Our participants also noted an increasing difficulty in finding enough qualified reviewers to perform the necessary reviews. One thing making that more difficult was the growing problem of free riders. P9 explained that *"What I have found is that we get a lot of submissions from some people who will never review for us. So, you have a lot of people who suck blood out of the system because everyone else needs to review their stuff but they will never do anything."* The result is that inviting reviewers is becoming a larger component of an editor's role. P4 explained that *"finding the appropriate reviewers, that's kind of difficult to start with. Then keeping track of the reviewers and making sure they produce a timely and constructive review, that's also a challenge."* P9 noted that *"I usually have to invite about 30 people in order to get three who accept"* and *"the three who accept may not deliver…they just drop dead, no response, nothing, full on ignore after they've accepted."*

When asked about how reviewers might be incentivized to perform more and better-quality reviews, there was a range of responses. One that was mentioned by multiple respondents was monetary compensation. P1 noted that MDPI offers a monetary incentive system for reviewers where the *"incentive is in the form of credit that a reviewer can use if they want to submit their paper"* so if you *"review 10 papers where you get the incentives, then you can make the publishing fee basically free."* P1 liked this incentive and felt it motivated him to review for MDPI. There were others, however, who felt that paying reviewers would only weaken the intrinsic motivation to review as a service to the community. A potential incentive that many felt could also be useful is recognition by outlets for the efforts of their reviewers. This aligns with the findings of other studies that non-financial incentives like acknowledgments of reviewer's work by institutions and journals would encourage scholars to accept review requests [30].

3.2 Issues with Current Peer Review Practices

The interviews with academic researchers along with the study of past literature helped us identify major issues with the current peer review practices. The identified deficiencies are summarized below:

1. **Difficulty Finding Qualified Reviewers:** Most of the participants we interviewed identified the increasing difficulty in finding appropriate reviewers as the major challenge in their role as an editor. A significant portion of their time is dedicated to sourcing qualified individuals willing to undertake review responsibilities.

2. **Reviews Not Completed Within Due Date:** Reviewers who agree to submit review reports also do not complete their assignment with due diligence. Delays in submission are prevalent, and in some cases, reviewers neglect to respond altogether.

3. **Lack of Incentivization:** While many researchers are motivated to review manuscripts out of a sense of community service, they often receive little to no incentives for their efforts. Some outlets acknowledge reviewers in annual reports or on their websites. However, there is no standardized method for formally recognizing and compensating reviewers for their contributions.

4. **Lack of Transferability of Rewards:** Although a few journals offer forms of compensation, such as subscription fee credits, these rewards typically remain confined to the publisher's platform. This lack of transferability prevents researchers from accessing diverse research communities across various journals, conferences, and workshops.

3.3 Objectives of BARIT

Recognizing the shortcomings of existing peer review systems, we define the solution objectives by outlining the requirements for a system that addresses current challenges through proper incentivization mechanisms. We then present the design principles and key features of BARIT, our proposed solution. The system requirements, design principles, and the derived features for BARIT are outlined in Figure 3.2.

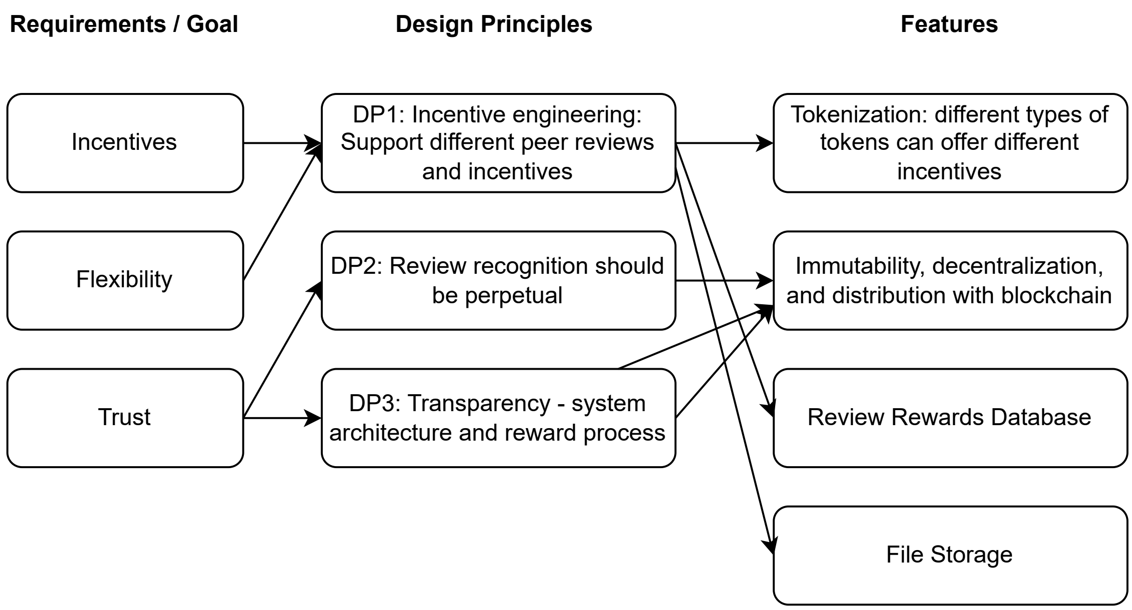


Figure 3.2: Requirements, design principles, and key features of BARIT

3.3.1 Research Question

How can we develop a peer review system that gives editors enough flexibility in how the system is operationalized to implement appropriate incentives for reviewers and provides editors and reviewers with sufficient trust in the system's functionality and data management?

3.3.2 System Requirements

We derived the design requirements for this research from our understanding of the problem, the literature, and the interviews we conducted. To address our design problem, we have kept both the user and organizational perspectives in mind as we developed our requirements, which include incentives, flexibility, and trust.

- **Incentives:** Peer review is structured as a voluntary service where potential reviewers can accept or decline requests to participate in the process. Therefore, designing a system to support peer review requires an understanding of the incentives that lead reviewers to voluntarily contribute their time to review manuscripts.

  Several theories address the role of incentives in motivating behavior including expectancy theory [100], reinforcement theory [101], rational choice theory [102], and self-determination theory [103]. In expectancy theory, valence is the extent to which individuals value different rewards. Thus, individuals are differently motivated to perform work depending on the type of reward they expect to receive for that work [100]. In self-determination theory, motivation is, at its most basic level, a function of goal orientation where a person is motivated either intrinsically (i.e., doing something because it is inherently interesting or enjoyable) or extrinsically (i.e., doing something because it leads to a separable outcome) [103]. Many scholars have viewed intrinsic motivation as more effective than extrinsic motivation but extrinsic motivation can come in many different forms with some much more effective and long-lasting than others [103]. This is important because not all activities are inherently interesting to the people who must engage in those activities.

  Through our interviews we identified several ways in which academics are incentivized to participate in the peer review process. These include the opportunity to learn about new research (intrinsic), requirements for promotion (extrinsic), a sense of quid pro quo (extrinsic), and a desire to serve the community (extrinsic). The system must be capable of incentivizing the extrinsic motivations of reviewers. The variety of incentives that could motivate reviewers to participate in the peer review process leads to our next design requirements, which is flexibility.

- **Flexibility:** The fact that reviewers vary in their motivations to participate in the peer review process indicates that for a system to enable editors to incentivize enough reviewers, that system must be flexible. Flexibility as a design requirement focuses on the needs of a system to support functionality that is customizable based on the needs of the system's stakeholders. In addition to the varied motivations of reviewers, different journals will have various methods they employ to structure their reviews. For example, most journals structure their reviews as a double-blind process in which the authors do not know the reviewers and the reviewers do not know the authors. However, some journals use a single-blind process in which the authors do not know the reviewers, but the reviewers do know the identity of the authors and other journals use an open structure in which the authors also know the identity of the reviewers. Journal editors will also have a range of needs and expectations regarding how to operationalize reviewer incentives. Some journals wish to incentivize reviewers with monetary rewards while other journals feel that monetary incentives would devalue the reviewing process and undermine other motivations like a desire to serve the community. Therefore, the system should be designed with the flexibility to support different peer review methods and provide a range of incentive capabilities.

- **Trust:** Trust by reviewers in the reviewing system is essential because many of the benefits reviewers receive from reviewing are managed by the system. All users (authors, reviewers, and journals) should be able to trust the system to assign rewards and protect their earned rewards. The system must reliably deliver the promised rewards.

3.3.3 Design Principles

The design principles with serve to guide the design and development of the proposed solution. Three design principles are derived from the system requirements: incentives engineering, perpetual rewards, and transparency.

- **Design Principle 1: Incentives Engineering -** Editors should have the ability to configure the review process and associated incentives based on their needs. The system should be able to support all peer review types (open, single, and double-blind) depending upon the process followed by each outlet. Incentives could include, but would not be limited to, monetary rewards, certificates of recognition, and credits for journal submission. Editors should be able to apply a weighted value to each review based on, among other attributes, the quality of the review and the timeliness of the review submission. This satisfies the incentives and flexibility requirements.

- **Design Principle 2: Perpetual Rewards -** Rewards should be perpetual regardless of the journal or conference venue. This ensures trust and encourages participation from all stakeholders.

- **Design Principle 3: Transparency -** All users should have access to transparent information about the system architecture for creating, distributing, and maintaining incentives. This would not preclude the editors from keeping the review process itself opaque to non-editors.

3.3.4 Features

Leveraging blockchain as the underlying technology would be optimal for aligning with the above-mentioned design principles. The following features satisfy the design principles for BARIT:

- **Blockchain:** BARIT utilizes blockchain technology to manage incentives. It maintains a permanent ledger that tracks all rewards received by reviewers. This immutability ensures rewards are perpetual, verifiable once awarded to the reviewer, and can't be tampered with. The public nature of blockchain ensures the transparency of the reward mechanism while adhering to confidentiality standards by anonymizing reviewers as needed.

- **Tokenization:** Tokenization allows for the customization of digital assets into crypto-tokens [69]. These tokens can hold varying values, supporting diverse incentive forms such as recognition and redeemable tokens. The actual value of each token will be determined by the outlet based on their system and what is appropriate for them.

- **Review Rewards Database:** As the system needs to support different review processes with varying anonymity levels, it is necessary to protect the privacy of the authors and reviewers. Storing all the information related to the peer review process on a private database ensures that such information will remain confidential.

- **File Storage:** The system needs to handle a high volume of file uploads and downloads, including manuscripts from authors, review reports, and editor notes, which can vary in size from kilobytes to megabytes. Directly storing large files in a database or blockchain network can be expensive. Therefore, a dedicated file storage system offers reduced resource costs and improved throughput.

Chapter 4

BARIT System Design

This chapter presents an overview of the BARIT system design; including the intended user roles, major technological concepts, and user story. BARIT presents a framework designed to facilitate all types of peer review processes and encourage active reviewer participation through a versatile incentive model that meets reviewers' motivating factors. Recognizing the varied methodologies and incentive preferences of journals, this platform empowers them to determine the most suitable reward mechanism for their needs.

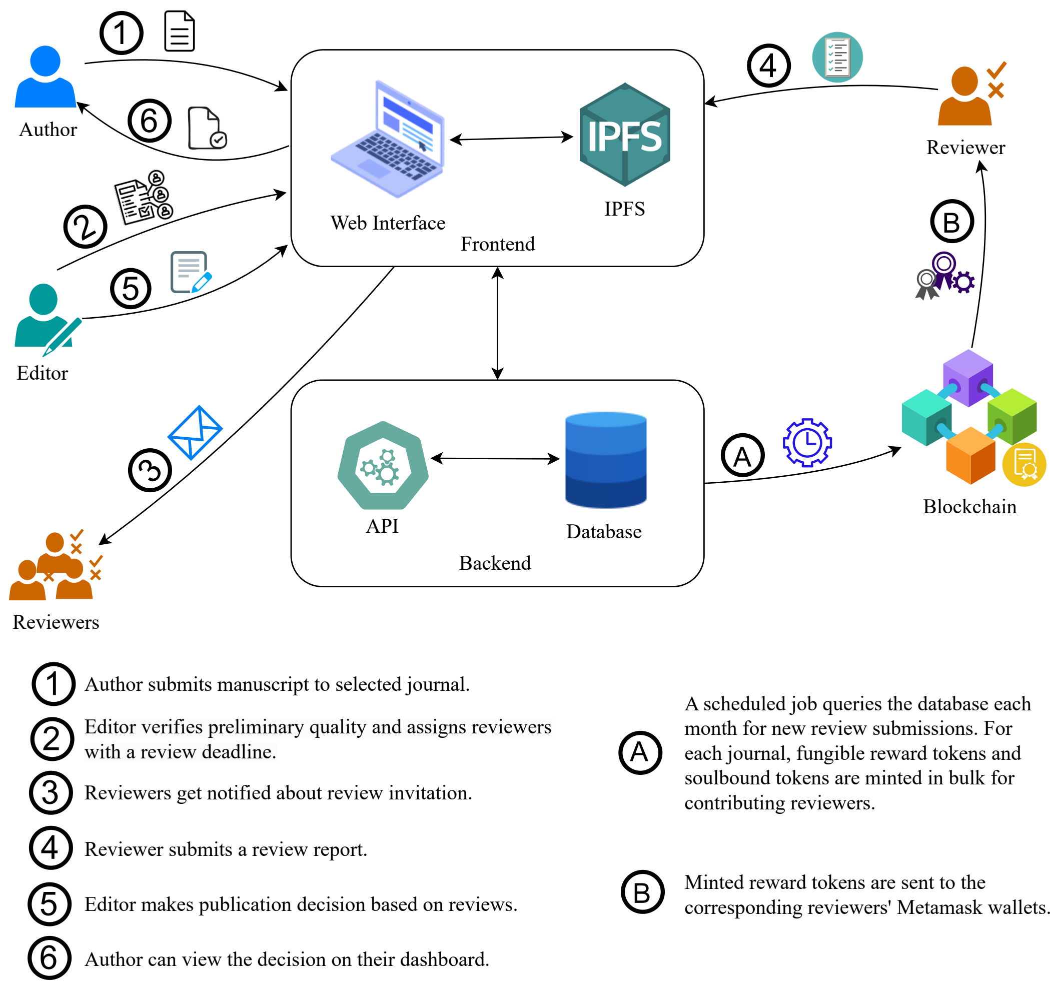


Figure 4.1: System diagram of BARIT

Figure 4.1 illustrates the architecture design for BARIT. BARIT provides an interface for three user roles: authors, editors, and reviewers, who interact with the system frontend. The backend consists of blockchain technology, the Interplanetary File System (IPFS) [104] for file storage, and an off-chain database. Communication between the frontend and backend components is facilitated through REST API.

4.1 Users

BARIT provides an interface for the following user roles:

1. **Authors** who submit their manuscripts to the journal for review and track submission decisions.

2. **Editors** who verify the quality of manuscripts and assign reviewers.

3. **Reviewers** who can view and submit reviews for the manuscripts assigned to them. With each manuscript submission, reviewers get a certificate token and possibly utility tokens.

4.2 Blockchain

Blockchain allows secure and transparent transactions without the need for a trusted third party [8]. There are a variety of available public blockchain networks with their own set of functionalities, advantages, and shortcomings. The key factors for network selection are transaction costs, throughput, decentralization, and security. Ethereum [59], a popular and most widely recognized network, has a robust ecosystem, an extensive developer community, and provides comprehensive documentation.

However, Ethereum faces challenges in scalability and affordability compared to its competitors, such as Avalanche and Solana. Solana, for instance, offers lower costs and faster transaction processing than Ethereum. However, this advantage comes at the cost of a certain degree of decentralization, due to its relatively smaller validator set [105]. Avalanche is a strong contender for Ethereum as it has similar levels of security as Ethereum with scalability and transaction cost efficiency.

Despite newer competitors such as Avalanche and Solana, Ethereum's proven security track record remains a significant advantage. Additionally, it offers advanced functionality and support for non-fungible tokens (NFTs) such as Soulbound tokens [106], making it a better choice for applications like ours that involve such tokens.

4.3 Cryptowallets

A user can manage their Ethereum account with a private key. Ethereum accounts have an address where all their funds are held. Users need their private key to sign transactions that verify their identity when transferring funds from one address to another. Hence, it is necessary to keep these keys secret. Crypto wallets are a piece of software or hardware that securely stores these cryptographic keys, acting as a digital wallet that proves their ownership of digital assets and authorizes each transaction on the blockchain. Reviewers benefit from the secure infrastructure provided by crypto wallets, which allows them to securely receive and manage their earned reward tokens.

4.4 Reviewer Anonymity with Hybrid Approach

Blockchain transparency is crucial for network integrity but raises privacy concerns when dealing with identifiable information [107]. Transactions often contain sensitive information that should not be exposed to unauthorized parties. Although obfuscation methods like zero-knowledge proofs and cryptographic techniques can help protect privacy, they often require substantial resources and tech-savvy users with a basic understanding of how these techniques work.

In blinded peer review processes, anonymity preservation of the involved participants is a non-negotiable requirement. However, the small size of the scientific community and transaction timestamps on blockchain make full anonymization difficult. Even with identity obfuscation, timestamps for token distribution done after the submission of reviews could potentially link reviewers to their reviews and the manuscripts they assessed.

To address these privacy concerns, we adopt a hybrid approach as suggested in [108]. This approach involves using blockchain for incentive management while storing user identities in an off-chain database. Author and reviewer identities, along with other sensitive data such as transactions related to the review process (manuscript submission, reviewer assignment, and review report submissions), are logged in the off-chain database. This ensures sensitive information remains secure and is only disclosed to relevant parties based on the peer review process (open, single-blind, or double-blind). The incentive scheme is built upon blockchain technology using smart contracts and tokenization. The system periodically queries the database for new reviews. For each submission, reviewers are allocated crypto-tokens as incentives. Token distribution is recorded on the blockchain, making the rewards perpetual and publicly displaying accumulated tokens, associated journals, and reviewer addresses. Reviewer anonymity regarding reviewed papers is maintained because all identifying information remains off-chain. Distributing tokens to a large pool of reviewers each month makes it nearly impossible to trace papers back to reviewers using timestamps alone.

As relational databases are faster and more scalable compared to blockchain solutions, using a relational database for the peer review process has the added benefit of keeping the system highly performant while benefitting from the permanent tamper-proof nature of blockchain for review rewards.

4.5 Incentive Engineering

Incentivization is at the core of BARIT, designed to motivate high-quality reviews. The system empowers editors to customize settings to fit their journals' specific requirements. We explore two primary incentive models: non-transferable certificates and utility tokens.

4.5.1 Non-transferable Certificates

Non-transferable 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 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.

4.5.2 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 [109]. 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.

4.5.3 Token Distribution Algorithm

To maintain reviewer anonymity, as outlined in Section 4.4, reviewers are not immediately rewarded upon submitting a review report. Instead, a flag is activated in an off-chain database table upon review submission, signifying that the reviewer has fulfilled their task and is eligible for compensation. Subsequently, a scheduled monthly job queries the database and allocates recognition and utility tokens to the reviewers in accordance with the journal's policy. Algorithm 1 provides the pseudocode for token distribution.

4.6 User Story

Figure 4.2 illustrates the overall interaction between different users and BARIT components. The peer review process commences with the submission of a manuscript by an author to a publishing venue for review. The system periodically checks for new review submissions and mints reward tokens based on the venue's policy. The anonymity of the author and reviewer in the peer-review process depends on the participating journal's policy. The sequence of steps in the overall process is outlined below:

4.6.1 Manuscript Submission by Authors

When an author submits a manuscript to a journal, the manuscript file is uploaded to the designated file storage. The author and manuscript details are stored in an off-chain database.

Algorithm 1: Algorithm for review reward token distribution

Reviewer r submits a review report s for journal j;
$s^j_r.timestamp \leftarrow CURRENT\_TIMESTAMP$;
$s^j_r.sbt \leftarrow 0$; `/* sbt refers to non-transferable certificates */`
$s^j_r.frt \leftarrow 0$; `/* frt refers to utility tokens */`
$s^j_r.rewarded \leftarrow 0$;
Token distribution is scheduled to run once a month;
for *Each journal* j **do**
  List of reviewers who submitted reviews within the month $r[] \leftarrow []$;
  if $s^j_r.rewarded = 0$ **then**
    Insert r into $r[]$;
  end
  if $r[].length > 0$ **then**
    Bulk mint sbt tokens $bulkMintSBT(r[], j)$ and send them to r wallet;
    $s^j_r.sbt \leftarrow 1$;
  end
  if FRT *is enabled for journal* j; $j_{frt} = true$ **then**
    if $s^j_r[].timestamp <= s_{deadline}$ **then**
      Bulk mint frt tokens $bulkMintFRT(r[], frt_{within\_deadline\_amount})$ and send them to r wallet;
      $s^j_r[].frt \leftarrow 1$;
    else
      Bulk mint frt tokens $bulkMintFRT(r[], frt_{after\_deadline\_amount})$ and send them to r wallet;
      $s^j_r[].frt \leftarrow 1$;
    end
  end
  $s^j_r[].rewarded \leftarrow 1$
end

4.6.2 Assign Reviews by Editors

The editor gains access to the submitted manuscript. If the manuscript meets the requisite standards, the editor assigns one or more reviewers, along with a review deadline. Editors have the option to allocate different amounts of utility tokens from their settings, depending on whether the review is submitted on time or after the deadline.

4.6.3 Submit Reviews by Reviewers

Reviewers receive review invitations along with deadlines for submission. Reviewers upload their review reports, accompanied by questions related to the paper they reviewed. Details related to the review submission are stored in the database.

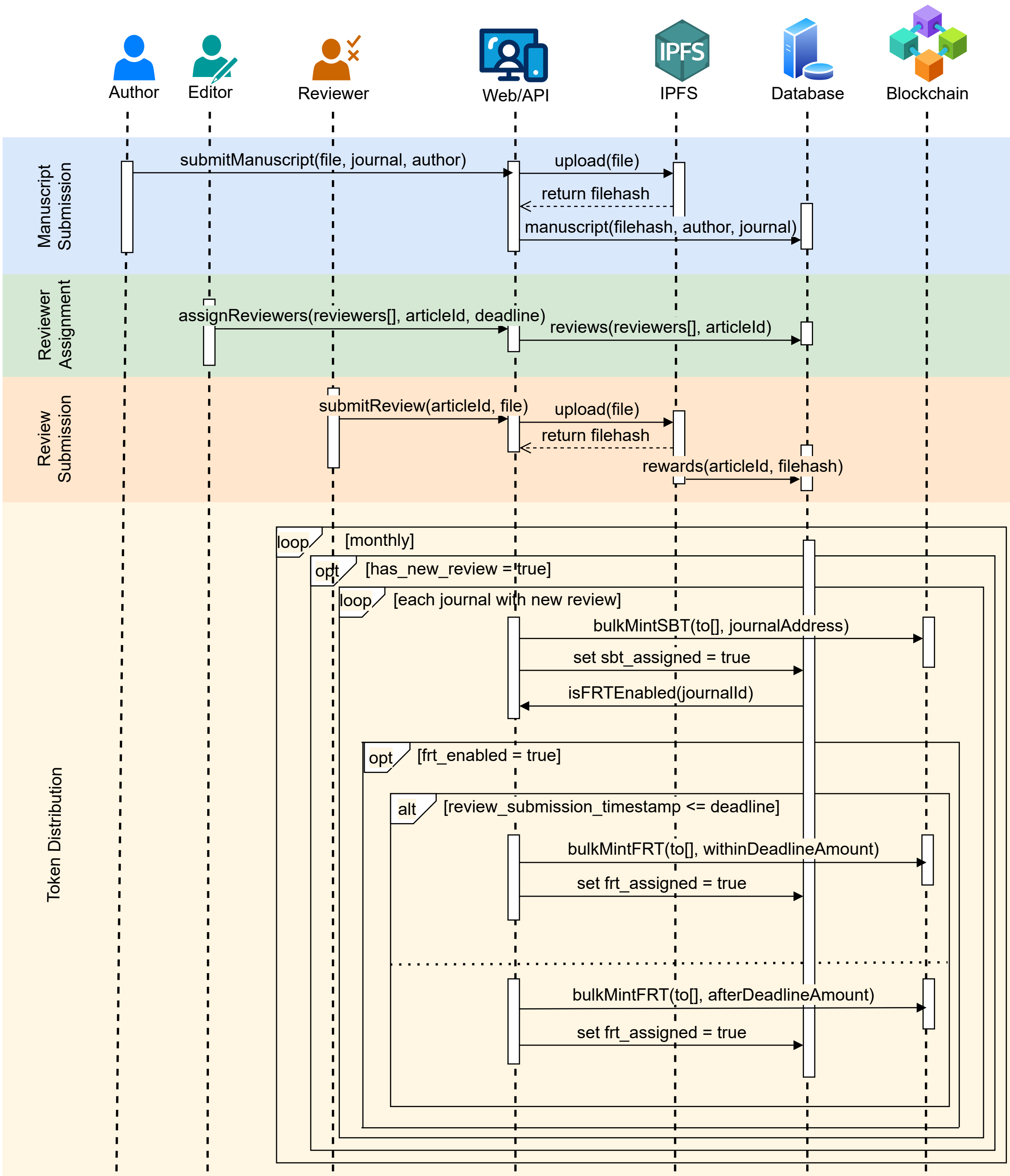


Figure 4.2: Sequence diagram of BARIT

4.6.4 Manuscript Final Decision Taken by Editor

The submitted reviews are accessible to the editor. Based on the reviews, they make an informed decision to either accept or deny the paper. Authors can now view the status of their paper's acceptance along with the review reports. The anonymity of the reviewer may be maintained, depending on the review process followed by the journal.

4.6.5 Reward Distribution

The system periodically queries for new reviews submitted on the platform. Upon finding new reviews, reviewers are awarded non-transferable recognition tokens (SBTs) linked to the journal they reviewed for. If the journal's incentivization policy allows, then reviewers are also assigned transferable utility tokens (FRTs) whose value and amount depend upon the journal's policy and may vary depending on review timeliness.

Chapter 5

Prototype Implementation

In the previous chapter, we discussed the system design for BARIT. This chapter provides the implementation detail for the BARIT prototype. The system architecture, along with implementation details for the hybrid peer review and incentivization platform, is illustrated in Figure 5.1 Users engage with the frontend, constructed using the `React.js` library, via their web browsers. Authentication is facilitated through the `MetaMask` wallet, enabling transactions on the blockchain network. HTTP requests from users are directed to the backend `Node.js` server, responsible for managing interactions with the `Oracle` database. All data pertaining to the peer review process is stored within this database. Additionally, the Node server executes scheduled tasks to engage with the blockchain network and periodically generate reward tokens.

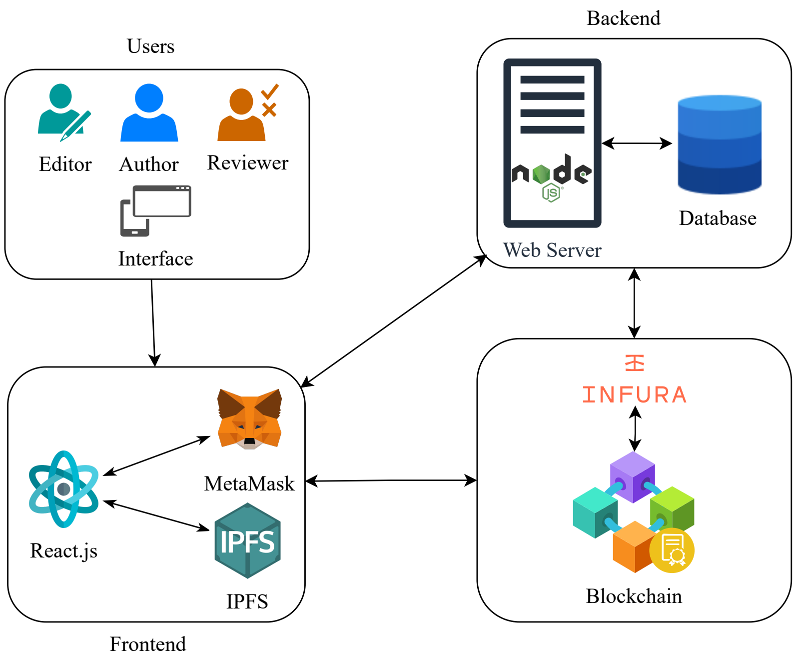


Figure 5.1: System architecture diagram of BARIT

All the tools used for the development of BARIT prototype are listed in Table 5.1.

Table 5.1: Tools used for prototype development. All the code is published at [5].

Functionality	Chosen tool
UI library	React.js v18.x [110]
Programming language	Node.js v18.x [111]
Smart Contract development	Solidity v0.8.4 [64]
Web Application Framework	Express.js v4.16.x [112]
Database	Oracle Autonomous Database [113]
Crypto wallet	MetaMask [74]
Web Server	Nginx [114]
Web hosting platform	OpenStack [115]
Containerization	Docker v20.10.17 [116]
Soulbound Token (SBT)	ERC-721 modified to be Soulbound [71]
Fungible Reward Token (FRT)	ERC-20 [71]

5.1 Blockchain Network

We are using blockchain technology exclusively to handle the allocation and transfer of incentive tokens. Using blockchain, incentives can be permanently recorded without any risk of data loss or tampering. Tokens accumulated by researchers for reviewing papers can also be used with any publishers within the platform, providing them more flexibility to freely decide upon the token usage. This encourages a more engaging peer review environment where reviewers will be more open to review papers from all venues based on their expertise and gain benefits for their contributions.

Among several blockchain networks available, we've selected the Ethereum [59] network for the peer review prototype. Ethereum is a widely popular and accepted blockchain network with a very active and strong ecosystem and large developer community and extensive documentation. The cryptocurrency used in Ethereum is called Ether (ETH). Any entity that holds ETH tokens are called accounts which can either be Externally Owned Accounts handled by users that can initiate transactions or contract accounts controlled by code [117]. Ethereum is an older network with a history of strong security along with being a pioneer in self-executable contracts known as smart contracts. They can be used for different transactions in blockchain networks such as the deposit and withdrawal of tokens and cryptocurrency. We use smart contracts to mint and allocate incentive tokens to the reviewers as well as to display and transfer the tokens to redeem platform-specific benefits. One of the major benefits of using the Ethereum network is its extensive support in securely managing and modifying both fungible and non-fungible crypto tokens, which is necessary for providing flexible incentivization capability.

Most client-facing API calls either interact with the private database or with read-only smart contract methods, which are usually fast and do not incur gas costs. Therefore, high blockchain throughput is not a big concern. Security and reliable management of fungible and non-fungible crypto-tokens are essential due to the token-based incentive structure. With Ethereum's pioneering support for non-fungible tokens and established standards, such as `ERC-20` and `ERC-721` [71], the platform provides a reliable framework for minting, ownership, and transferring tokens. Ethereum's smart contract capabilities facilitate the creation of non-transferable fungible tokens. Moreover, if increased transaction throughput or lower costs are required in the future, a transition can be made to Layer 2 solutions like Polygon [118] or EVM-compatible chain such as Avalanche [106].

BARIT's smart contracts are built using `Solidity v0.8.4`, an object-oriented programming language specifically designed for Ethereum Virtual Machine (EVM) [64]. These smart contracts are deployed on the `Ethereum Sepolia` test network.

5.2 Crypto Wallet

The prototype is integrated with `MetaMask` wallet, as it provides extensive integration with Ethereum-based networks. MetaMask provides stringent security and a straightforward interface to claim and manage crypto tokens [74]. MetaMask doesn't require ID verification which helps preserve the privacy of BARIT users. The MetaMask browser extension used to connect with dApps is supported by most popular browsers, making it easily accessible for users regardless of their preferred web platform. This allows our diverse user base the freedom to choose their preferred web browser for claiming and managing tokens.

5.3 Tokenization for Incentives

The incentives for review submissions are distributed in the form of blockchain tokens. We support two incentive models: non-transferable certificates of recognition built using Soulbound tokens and transferable fungible reward tokens.

5.3.1 Soulbound Token (SBT)

Soulbound Tokens are an extension of the `ERC-721` standard for non-fungible tokens (NFTs) designed to serve as a digital certification or badge. They are unique non-fungible tokens with modifications making them non-transferable. SBTs are assigned to an address upon minting that permanently ties the token to that particular 'soul' [72]. These tokens are enabled by default and act as public certificates of recognition for reviewers' contributions. Reviewers earn one SBT per completed review, serving as an immutable record of their contributions and a public acknowledgment of their work. SBT metadata stores the token name, description, type of contribution, journal name, and address, all of which allow the user holding the token to have verifiable proof of acknowledgment of their contribution. Researchers can display their Soulbound Token (SBT) tokens on their reputation page in the platform along with being able to handle them from their MetaMask wallet.

5.3.2 Fungible Reward Token (FRT)

In addition to review certificates, separate Fungible Reward Token (FRT)s are used to serve as remuneration for reviews. Fungible Reward Tokens are transferable fungible tokens adhering to the `ERC-20` standard. Editors have the flexibility to opt-in to reward their reviewers with FRTs from their policy settings page. To encourage timely submission of review reports, editors can adjust the token distribution amount such that reviewers are rewarded more generously for completing their assignment within the provided deadline. Journals can assign values to FRTs according to their policies, such as offering them as credits towards subscription fees, submission fees, or other publication ecosystem incentives.

Distribution of Tokens

As illustrated in Figure 4.2, reviewers are periodically awarded SBTs and FRTs based on reviewed journal policies. Each blockchain transaction incurs a computational overhead associated with its initialization. When individual transactions are created for each token minting process, transaction overhead can significantly elevate cumulative gas costs. To mitigate this issue and enhance cost-effectiveness, all tokens of the same type are minted within a single transaction. This is achieved by consolidating the necessary information, including the reviewer's addresses, into a single blockchain method, from which all tokens are then minted. This approach reduces gas fees, minimizes resource utilization, and alleviates blockchain congestion. Additionally, various Solidity cost-optimization techniques such as using memory variable instead of storage whenever possible and the use of *unchecked for operations that cannot overflow or underflow*, are implemented to further optimize efficiency. The code fragment below shows the implementation of *unchecked for operations* in bulkMintSBT smart contract method.

```
function bulkMintSBT(address[] memory _tos, string memory journalString)
    public onlyOwner {
     require(_tos.length > 0, "No tokens to mint");
     uint256 currentTokenId = _tokenIdCounter;
     unchecked {
         for(uint256 i=0; i< _tos.length; i++) {
             currentTokenId += 1;
             _safeMint(_tos[i], currentTokenId);
             ownerTokenIds[_tos[i]].push(currentTokenId);

             string memory tURI = _generateTokenURI(
                 currentTokenId,
                 journalString
             );
             _setTokenURI(currentTokenId, tURI);
         }
     }
     _tokenIdCounter = currentTokenId;
}
```

5.4 File Storage - IPFS

Peer review process involves frequent exchange of files among authors, editors, and reviewers. Authors upload their manuscripts, while editors and reviewers access the uploaded manuscripts and submit review reports and editor notes on submission decisions. All of these files are stored in a dedicated external file storage system for easy retrieval and to prevent transaction overhead and storage costs on the blockchain network and Oracle database.

The system utilizes the InterPlanetary File System (IPFS), a distributed data storage and sharing protocol [104]. With IPFS, the system is no longer dependent on central servers and single points of failure, thus enhancing the reliability against data loss. Files can be stored permanently on IPFS, with faster retrieval for frequently accessed ones. IPFS uses content-addressable storage, ensuring that the files once uploaded can't be tampered with, thus maintaining the data integrity of the submitted manuscripts and review reports. We used Pinata as the IPFS provider to handle storage and dedicated gateways for faster retrieval.

The use of IPFS for file storage can remain optional, allowing the publication venue to continue using its existing file storage system.

5.5 Review Rewards Database

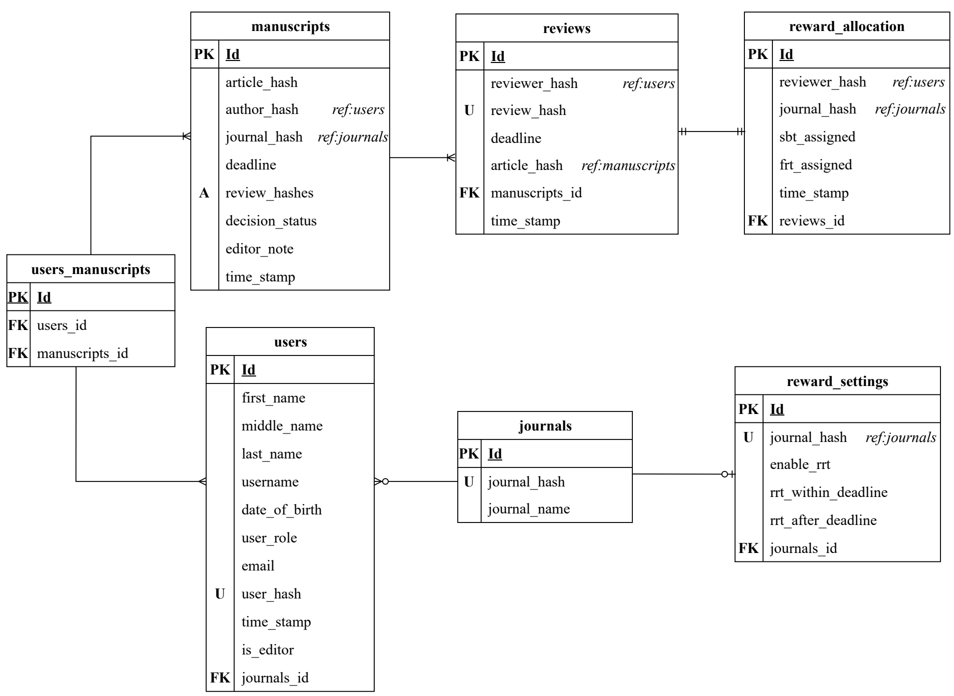


Figure 5.2: Entity relationship diagram for review rewards database

The confidential information related to the review process along with all sensitive data related to authors' and reviewers' identities are securely managed in an off-chain Oracle Autonomous Database [113]. Oracle autonomous database supports fast query performance without having to dedicatedly manage its hardware or software resources. It is auto-scalable and automatically manages backups, maintenance, database tuning, and routine management tasks eliminating the need for database administrators.

All sensitive information such as authors, papers submitted, reviewers assigned, and reviews submitted are stored in Oracle. The database is structured using seven tables: *users*, *journals*, *reviews*, *manuscripts*, *users_manuscripts*, *rewards_allocation*, and *reward_settings*. The *users* and *journals* tables contain identifying information about different users and journals respectively. The *manuscripts* table stores all logs related to manuscript submission and decision whereas the *reviews* table contains information about review assignments. The *users_manuscripts* table contains the list of authors for each manuscript submission. Journals' reward policies are stored in *reward_settings* table. The *reward_allocation* table logs token allocation information for all reviewers with flags indicating whether they've received reward tokens for their contribution. The entity-relationship diagram for the database is illustrated in Figure 5.2. In the diagram, ***PK***, ***FK***, and ***U*** represent primary keys, foreign keys, and unique keys, respectively. The database design incorporates some non-relational features. The array type is denoted by ***A***, with the attribute *review_hashes* being an example. Additionally, references to tables are indicated by the format *ref:<table_name>*, signifying that the attribute refers to the specified table.

Database Backup: The database undergoes daily backups to facilitate seamless data recovery in the event of unexpected data loss caused by system failures or disruptions.

In this initial design, the platform owner manages the database that stores all data related to the review process and reward distribution. Once the prototype is integrated with existing peer review management systems, data related to the review process will be managed by the respective journals, while BARIT will be responsible for the rewards database and distribution.

5.6 RESTful API

The `REST API` is used to interact with the backend components of the prototype i.e. SQL database and smart contract methods. The API is developed using the lightweight and minimalist `Express.js v4.16.x` framework for `Node.js v18` [112]. Express.js simplifies request handling and routing for client requests and allows easy integration with other modules and libraries. Additionally, Express.js's extensive documentation and active community support make it a safe choice for the system's longevity. Scheduled jobs to bulk mint reward tokens are listed in Table 5.2. Different API methods used for the create, read, update, and write (CRUD) operations on SQL database as well as for uploading and downloading files from IPFS are listed in Table 5.3.

Table 5.2: Cron jobs to bulk mint SBT and FRT tokens

Job	**Schedule**	**Description**
bulkMintSBTTokens	Monthly	Mint soulbound tokens in bulk. One blockchain transaction is made per journal that assigns SBT tokens to eligible reviewers.
bulkMintFRTTokens	Monthly	Mint fungible reward tokens in bulk. One blockchain transaction is made per journal that assigns FRT tokens to eligible reviewers.

Table 5.3: API call methods implemented on BARIT for database and IPFS interaction

Path	Method	Body	Query	Description
/manuscript-submission	POST	author_hash, journal_hash, file_hash	–	This API call creates a new entry for the submission of a manuscript to a selected journal or conference proceeding. It receives the IPFS address of the uploaded manuscript file along with the addresses of the author and the journal, and stores this information in the database.
/get-assigned-reviewers	GET	–	article_hash	This API call returns all reviewers assigned to an article. The user sends a request to the API through the user interface with the article hash as a query parameter. The response includes a list of reviewers assigned to that article.
/add-reviewewrs	POST	reviewer_hashes, article_hash, deadline	–	This method adds reviewers to a selected manuscript. Editors can call this API to assign one or multiple reviewers to the manuscript. The reviewers are then stored in the database after validating that each reviewer is assigned to unique article hashes.
/get-manuscripts-by-author	GET	–	author_hash	This API call returns a list of manuscripts submitted by an author. The author provides their address hash as a query parameter. The backend server queries the database to retrieve and return the list of manuscripts submitted by the user as an author.
/get-manuscripts-by-reviewer	GET	–	reviewer_hash	This method returns a list of manuscripts assigned to a user for review. The user's address is sent as a query parameter. The backend server queries the database to find and retrieve the manuscripts assigned to that user's address for review.
/get-manuscripts-by-journal	GET	–	journal_hash	This method returns a list of manuscripts submitted to the publication outlet. Editors send a request with the journal address to query the database and retrieve the list of manuscripts submitted for review.
/review-submission	POST	article, reviewer, review_hash, journal	–	This API call records the review hash and the review submission timestamp. When a reviewer successfully submits a review file to IPFS, this API call is triggered with the reviewer's, journal's, and article's addresses as body parameters. The backend server then executes a query to store the review hash and submission timestamp in the database. Additionally, this API sets a flag to indicate that the reviewer has submitted a review for the specified journal.
/get-token-settings	GET	–	journal_hash	This method returns the reward policy set by the journal.
/update-review-settings	POST	journal_hash, enable_rrt, amt_per_rev_within-_deadline, amt_per_rev_after_deadline	–	This API call updates the outlet's reward settings. The editor of the outlet provides details such as the enabled or disabled status of RRT tokens, along with the allocated amount of tokens for reviews submitted within the deadline and after the deadline. The backend server then stores this information in the database.
/get-journals	GET	–	–	This method returns a list of all journals that have signed up on the platform.
/get-reviewers	GET	–	–	This method returns a list of all available reviewers.
/get-journal-detail	GET	–	journal_hash	This API call returns details of a particular journal. The user interface makes the API request with journal address as query parameter. The API queries the database and returns the journal name and other details.
/update-decision-status	POST	decision_status, manuscript_hash, editor_note	–	This method updates the decision status for a paper submission. Editors can use it to make a final decision and submit an editor's note, while authors can use it to withdraw their submission. The API call receives the decision status, manuscript hash address, and editor's note (optional) as parameters, and stores this information in the system database.
/file-upload	POST	file, name	–	This API call uploads and pins file to IPFS. This is used by the users to upload the manuscript, review reports, and editor notes.
/file-download	GET	–	ipfs_hash	This API call returns file from IPFS network. It is used by the users to view the submitted manuscript, review reports, and editor notes.

5.7 Smart Contracts

We are using two smart contracts for review reward management. Minting and distribution of SBT tokens are handled by the SoulBoundToken smart contract whereas transactions related to FRT tokens are handled by the FungibleRewardToken smart contract. The minting and distribution of tokens are handled by cron jobs whose gas costs are covered by the contract owner. Users do not need to incur gas fees when calling view functions (functions that do not change state). The implementation details of the smart contracts are discussed below.

5.7.1 SoulBoundToken Smart Contract

The SoulBoundToken smart contract handles all the transactions related to the Soulbound Tokens. This smart contract defines the individual and bulk minting of SBTs, metadata present in the minted tokens, and allocation of SBTs to the respective crypto addresses of reviewers. All methods implemented in this smart contract are discussed in Table 5.4. As Soulbound Tokens should not be transferable, this smart contract builds upon the `ERC-721` standard and modifies the `_beforeTokenTransfer` method to prevent token transfer between two crypto addresses. To prevent unauthorized minting of SBT tokens, only the smart contract owner or admins are allowed to execute the `bulkMintSBT` and `singleMintSBT` methods as a security measure.

Table 5.4: Description of SoulBoundToken smart contract methods

Function	**Callers**	**Description**
bulkMintSBT	Cron job from Node.js API	Mints a batch of SBT tokens and assigns them to corresponding reviewers.
singleMintSBT	Cron job from Node.js API	Mints a single SBT token and assigns it to the corresponding reviewer.
getTokensOwned	Reviewer	Retrieve a list of SBT token IDs accumulated by the reviewer.
tokenURISBT	Reviewer	Returns the token URI containing the metadata of the given SBT token ID.
_generateTokenURI	Private	Generate the token URI for SBT token based upon the journal address, NFT image, and recognition details. Called from functions singleMintSBT and bulkMintSBT.
burn	Reviewer	Burn/spend the accumulated SBT. Only allowed for the owner of the token.
updateTokenImage	Contract Owner	Updates the default image used for the SBT tokens.
_beforeTokenTransfer	Internal	Overrides the contract behavior when a token transfer is initiated. Prevents the transfer of SBT tokens between entities.

5.7.2 FungibleRewardToken Smart Contract

The FungibleRewardToken smart contract is responsible for all Fungible Reward Token transactions. Functionalities such as individual and bulk minting of FRTs, token allocation, and transfer are handled by FungibleRewardToken smart contract. The methods supported by this smart contract are detailed in Table 5.5.

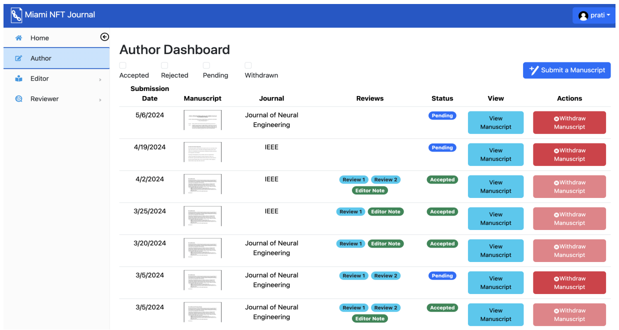
Table 5.5: Description of FungibleRewardToken smart contract methods

Function	**Callers**	**Description**
addAdmin	Admin	Add an admin to the contract.
revokeAdmin	Admin	Remove an admin from the contract.
bulkMintFRT	Cron job from Node.js API	Mints a batch of FRT tokens and assigns them to corresponding reviewers.
singleMintFRT	Cron job from Node.js API	Mints a single FRT token and assigns it to the corresponding reviewer.
balanceOf	Reviewer	Check the FRT balance of the user.
transfer	Reviewer	Transfer FRT tokens to another MetaMask wallet address.

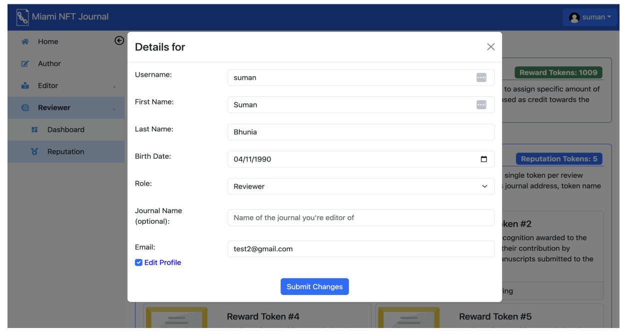
5.8 User Interface

The prototype for BARIT is designed to be simple and intuitive for all users. We used the `React.js v18` library to build a clean and scalable user interface. React.js offers a component-based architecture and has efficient rendering capabilities [110]. This ensures a seamless and responsive user experience along with efficient development and maintenance due to its reusable components. Users can interact with six distinct pages: login, dedicated dashboards for author, editor, and reviewer, policy settings page for editors, and reputation page for reviewers. The screenshots of the prototype are shown in Figure 5.3. The details of the review process such as the identity of the author and reviewer may be hidden or visible depending upon the review process followed by the journal. The general process flow with respect to these pages is discussed below:

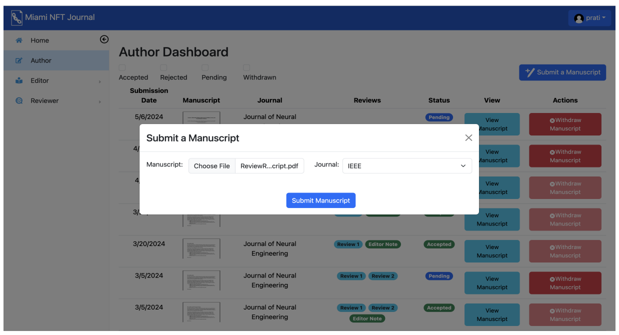
- *Logging in*: All the users need to log in to the system using their MetaMask account. After successfully authenticating, they are redirected to their respective dashboard based on their user role. They can update their profile from the profiles page.
- *Setting incentivization policy*: Editors have the option to opt-in to reward their reviewers with Fungible Reward Tokens. From the setting page, they can enable or disable issuance of FRTs and manage the amount of tokens distributed depending upon review submission timeliness.



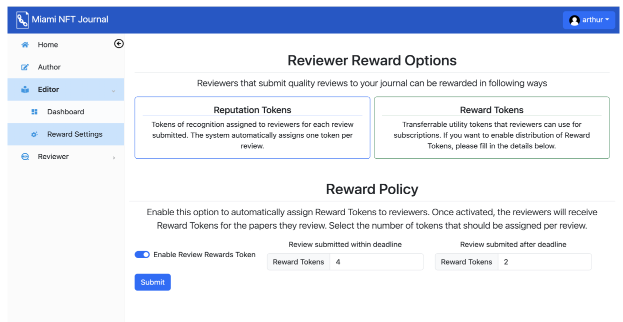
(a) Author Dashboard



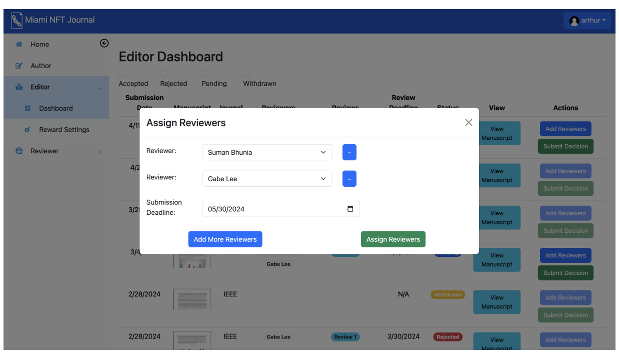
(b) User profile update



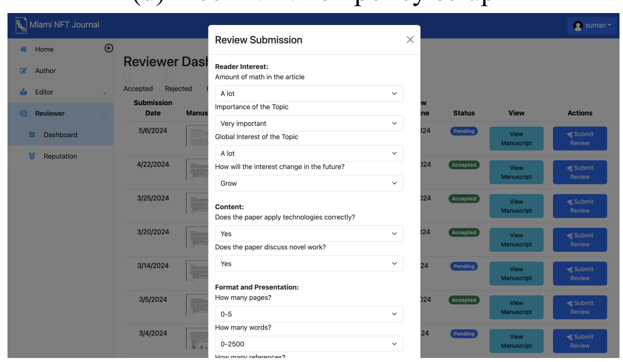
(c) Manuscript Submission



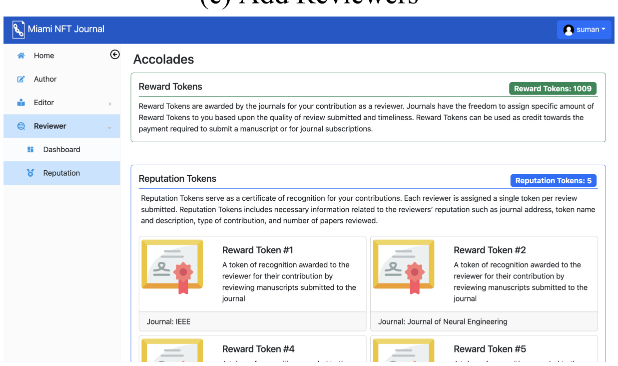
(d) Incentivization policy setup



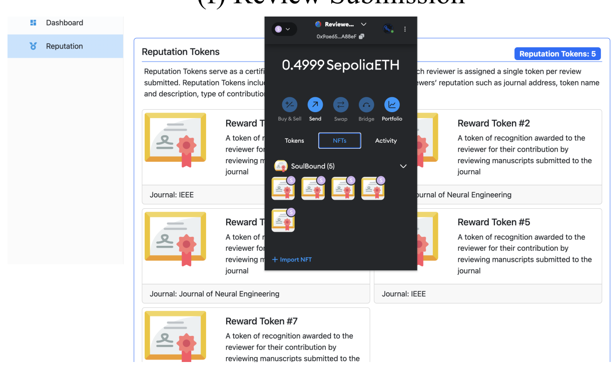
(e) Add Reviewers



(f) Review Submission



(g) Reputation page



(h) SBT tokens on MetaMask wallet

Figure 5.3: Screenshots of BARIT User Interface

- *Manuscript submission*: Authors can submit their manuscripts to the participating journals from their dashboard. Upon submission, they can view the decision status along with any reviews or editor notes that have been assigned to the manuscript.

- *Preliminary check and reviewer assignment*: Upon submission of the manuscript, editors can view submitted papers for review from their dashboard. They can view the file, reject the manuscript if it doesn't adhere to the venue's standards, or assign reviewers with submission deadline.

- *Review report submission*: Reviewers can access the manuscripts assigned to them from their reviewer dashboard. They are asked to upload a review document along with some additional information related to the manuscript for a successful review report.

- *Final decision*: The editor reads the review reports and makes a decision to either accept or reject the manuscript. Additionally, they can add an editor note with suggestions for changes. The author can view the decision from their dashboard.

- *Reviewer reputation*: Within a month of review submission, the reviewer will receive a Soulbound Token per review as a public acknowledgment of their contribution to the journal they reviewed for. If enabled, they will also receive FRTs based on the journal's policy. Reviewers can display their earned tokens on their reputation page, serving as a digital portfolio, and redeem FRTs with any publisher in the platform.

5.9 Deployment of User Interface and REST API

Both the user interface and REST API are containerized with `Docker` [116] and hosted on `Ubuntu` virtual machine running on `OpenStack` hypervisor [115]. Docker containerization simplifies deployment, guarantees a consistent environment, and optimizes resource utilization. This setup supports reliability and security, providing a stable and accessible platform for the peer review system's operation and future growth. We are using `Nginx` as a web server and reverse proxy as it can handle large volumes of simultaneous connections, and provides faster response times and high performance, all while taking fewer resources [114].

Chapter 6

Evaluation and Validation

This chapter discusses the evaluation of BARIT prototype by measuring the estimated gas price and latency of API calls. The prototype is also validated based on the feedback from potential users to determine if the system requirements are met.

6.1 Evaluation

All the functionalities of the smart contract methods were tested for accuracy by writing unit tests using `Hardhat`, an Ethereum development environment [119]. The smart contract methods are evaluated based on the estimated gas price for their execution and the delay in receiving a response from the backend API server or smart contract methods. These evaluations help us determine the robustness of the designed system and its ability to effectively scale to accommodate a growing user base. Tools used for evaluation are listed in Table 6.1.

6.1.1 Gas Price Estimates

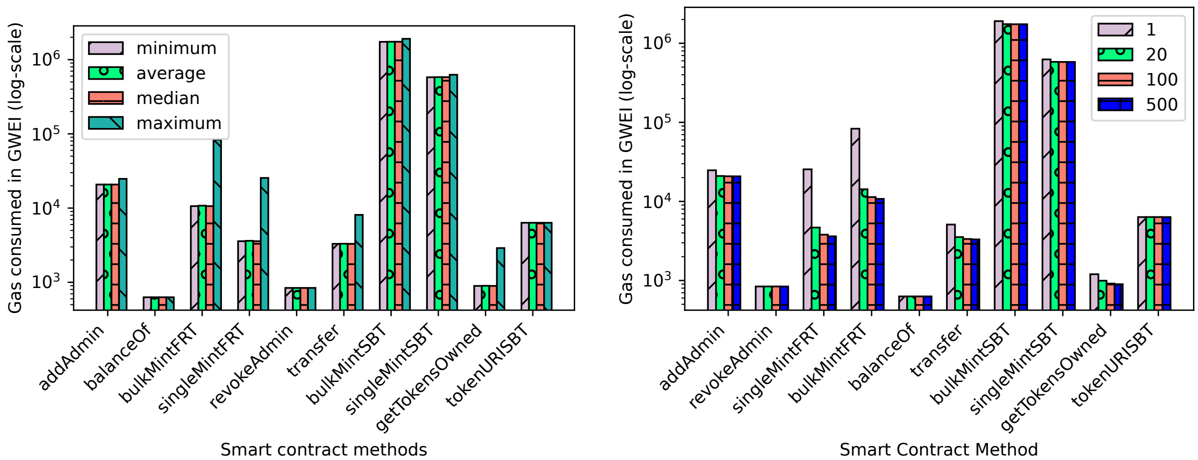
We integrated `Foundry tool-chain` [120] with Hardhat to estimate gas prices for smart contract methods. The gas price consumed is measured in GWEI.

Transaction Cost for Different API calls

The estimated gas prices for the SoulBoundToken and FungibleRewardToken smart contract methods are shown in Figure 6.1a. Each method is invoked 500 times to evaluate the minimum, average, median, and maximum gas price consumption estimates in GWEI (log-scale). The `balanceOf`, `getTokensOwned`, and `tokenURISBT` API calls are read-only methods that do not modify state variables. Since they merely retrieve data from the blockchain without altering the state, they consume the least amount of gas. However, among these methods, `tokenURISBT` consumes slightly more gas as it returns a larger amount of data, including all the metadata information of the SBT.

Table 6.1: Tools used for the experiment. All the code is published at [5].

Functionality	Chosen tool
Gas price estimate	Foundry Toolchain v1.1.1 [120]
Latency test	AutoCannon v7.15.0 [121]
Unit testing	Hardhat v2.11.0 [119]



(a) Gas estimate with 500 API calls

(b) Gas estimate with varying load (1-500 API calls)

Figure 6.1: Gas estimate for BARIT smart contracts

The `transfer`, `bulkMintFRT`, `singleMintFRT`, `addAdmin`, `revokeAdmin`, `bulkMintSBT`, and `singleMintSBT` methods require higher gas consumption because they modify the blockchain state. Minting soulbound tokens involves more complex computations and storage compared to minting simple fungible tokens, resulting in a significantly higher cost for minting SBT tokens compared to the minting and transfer of FRT tokens.

Scalability of the System with Varying Load Factor

To evaluate the scalability of the system, we measured the gas consumed in GWEI (log-scale) for the smart contract methods under varying load factors, ranging from 1 to 500 API calls. The results are plotted in Figure 6.1b. The legend indicates the number of API calls made. Most smart contract method calls exhibit consistent gas costs across different loads. The `balanceOf` API call returns the total amount of FRTs available to the user. The transfer API is responsible for token transfers. These methods will be used frequently for different token transactions in the system. Their low gas cost variance ensures predictable transaction costs and minimizes the risk of unexpected fees. The `geTokensOwned` API retrieves a user's SBT IDs for displaying earned SBTs with metadata obtained from `tokenURISBT` API. The gas costs for these API calls are also consistent across varying loads ensuring that the system can be easily scaled to handle growing user traffic without incurring an increase in gas costs.

Methods such as `bulkMintFRT` and `bulkMintSBT` facilitate the bulk minting and assignment of FRTs and SBTs, while `singleMintFRT` and `individualMintSBT` handle the minting and assignment of single tokens. These minting methods for SBTs have consistent gas prices. However, single and bulk minting of FRTs shows some variance, as the cost of running a single transaction is unexpectedly higher than the average cost of running 20, 100, or 500 transactions concurrently. Token minting can be scheduled for low gas cost periods to optimize efficiency. Low variance in gas costs is a desired attribute for accurate transaction cost estimation, enhancing system reliability.

Gas Price Estimates with Current Crypto Prices

Table 6.2 presents the cost of executing smart contract methods on various blockchain networks, converted to USD. The costs are estimated based on the average gas price for 500 API calls, using market prices for Ethereum, Solana, and Avalanche as of June 2, 2024. The execution cost is lowest on the Avalanche network and highest on the Ethereum network. While we've selected the Ethereum network for testing, we can move to the EVM compatible Avalanche network in the future to benefit from lower transaction costs.

Table 6.2: Average gas price of BARIT smart contract API calls (average of 500 API calls) in USD for Ethereum, Solana, and Avalanche blockchain network. The prices for Ethereum, Solana, and Avalance are $\$3482.25$, $\$147.90$, and $\$31.33$ respectively as of June 11, 2024 [6].

Function	GWEI	Ethereum($)	Solana($)	Avalance($)
addAdmin	20776	0.072347226	0.0030727704	0.00065091208
revokeAdmin	840	0.00292509	0.000124236	0.0000263172
bulkMintFRT	10776	0.037524726	0.0015937704	0.00033761208
singleMintFRT	3623	0.01261619175	0.0005358417	0.00011350859
balanceOf	629	0.00219033525	0.0000930291	0.00001970657
transfer	3310	0.0115262475	0.000489549	0.0001037023
bulkMintSBT	1742043	6.066229237	0.2576481597	0.05457820719
singleMintSBT	580809	2.02252214	0.0859016511	0.01819674597
getTokensOwned	898	0.0031270605	0.0001328142	0.00002813434
tokenURISBT	6341	0.02208094725	0.0009378339	0.00019866353

Measuring Gas Price Difference While Minting Varying Number of Tokens

Figure 6.2 illustrates a direct correlation between gas costs and the number of tokens minted per API call. It also displays the gas estimate for the `getTokensOwned` method when the given address holds a varying number of tokens. The execution cost for the `getTokensOwned` method increases with the number of accumulated tokens, which could complicate system scaling. Similarly, the cost of bulk minting rises proportionally with the number of tokens minted. If the tokens minted per API call are unchecked, it can lead to exorbitant gas prices. To prevent this, the API limits minting to a maximum of 30 tokens per transaction, ensuring affordable transaction costs while minimizing the overhead of multiple transactions. Future exploration could involve alternative solutions like consortium or permissioned networks to leverage blockchain benefits without significant financial constraints.

6.1.2 API Call Latency Estimates

We assessed delays for API calls involving database queries, smart contract interactions, and file transfers. We utilized `AutoCannon`, a benchmarking tool known for its accuracy and reliability

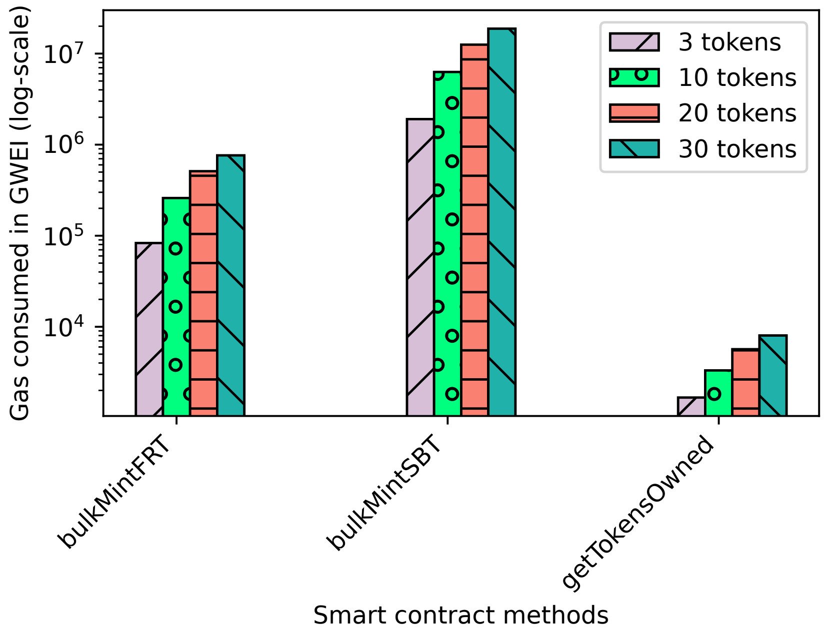


Figure 6.2: Gas estimate for bulk minting SBTs and FRTs with varying number of tokens minted in one call and varying amounts of tokens owned by the address

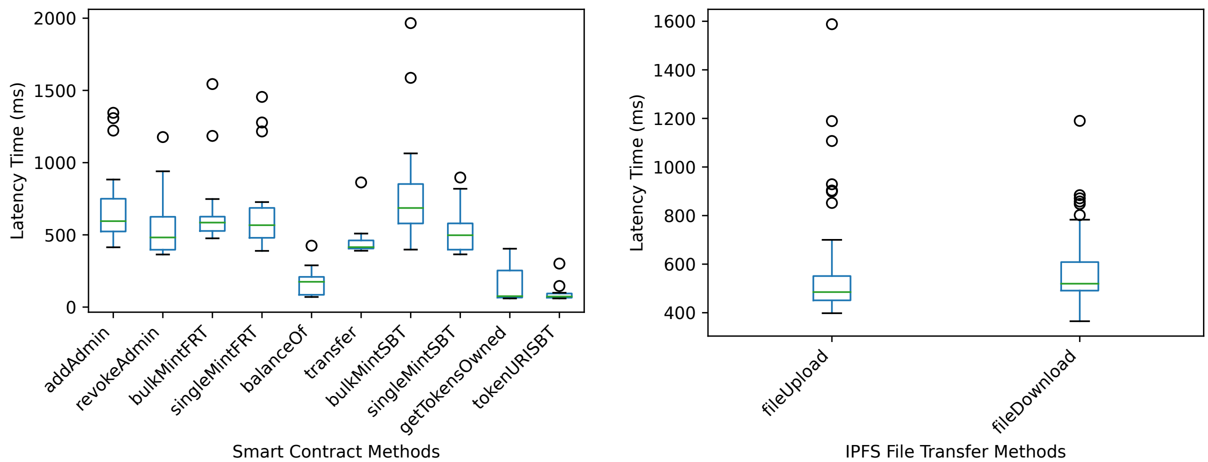
to measure the request latency of API calls [121]. The prototype underwent load testing with 50 concurrent requests per second for each API call, simulating a high user volume.

Latency while Invoking Smart Contract Methods

Figure 6.3a shows the latency of various smart contract methods with 15 concurrent API calls on the Sepolia test network. All read-only methods exhibit acceptable latency, ranging from 60 to 425 ms. The low latency of the `balanceOf` method, below 500 ms, and the transfer method, below 900 ms, ensures smooth token trading even under high network traffic. Users will also experience minimal delay displaying their accumulated SBTs, as the `tokenURISBT` and `getTokensOwned` methods have latencies under 500 ms. Bulk minting FRT tokens using the `bulkMintFRT` method takes between 476 and 1544 ms, with an average of 672.94 ms. Similarly, bulk minting SBT tokens have a slightly higher average delay of 795.26 ms, with some outliers reaching up to 1966 ms. Since token minting occurs once a month in the back end, the higher response delay does not directly impact the user experience. However, as latency has not been tested on the Ethereum Mainnet, results may vary.

Latency of File Transfer Methods

The latency for file upload and download from the IPFS network is illustrated in Figure 6.3b. The average latency for file upload was recorded at 573.32 ms, while for file download, it was slightly



(a) API calls to smart contract methods (15 concurrent requests)

(b) API calls for IPFS file transfer (50 concurrent requests)

Figure 6.3: Request latency of BARIT API calls for smart contract methods and IPFS file transfer

higher at 578.36 ms. The majority of requests for both methods fell within the range of 400 ms to 800 ms, with a few outliers reaching up to 1600 ms.

Latency of API Calls to SQL Database Queries

The latency and performance metrics of API calls responsible for various database queries are illustrated in Figure 6.4. The system maintained consistent performance under multiple concurrent requests, with minimal delay times averaging between 15-80 ms for most GET methods. However, calls such as get-journals and get-reviewers, responsible for retrieving all registered journals and reviewers, showed slightly higher average latency in the range of 140-150 ms. Despite this, the API latency remained sufficiently low for users not to perceive any noticeable wait times. The `POST` methods responsible for the submission of manuscripts, review submissions, decision submissions, and setting updates exhibited higher average latencies ranging from 426 ms to a maximum of 2.4 sec (for review submission). Considering the nature of form submissions, delays below 3 sec are generally acceptable. It's important to note that this experiment was conducted with a load of 50 requests per second. Considering that the majority of the time of all the stakeholders will be spent reading, editing, and reviewing the manuscripts rather than on the platform itself, We do not anticipate BARIT's user traffic to exceed the load of 50 concurrent requests per second.

6.2 Validation

After designing the prototype based on the requirements gathered from the initial interviewees, we demonstrated it to the same participants and collected their feedback through open-ended questions. Each participant received a follow-up email inviting them to participate in a user study with the

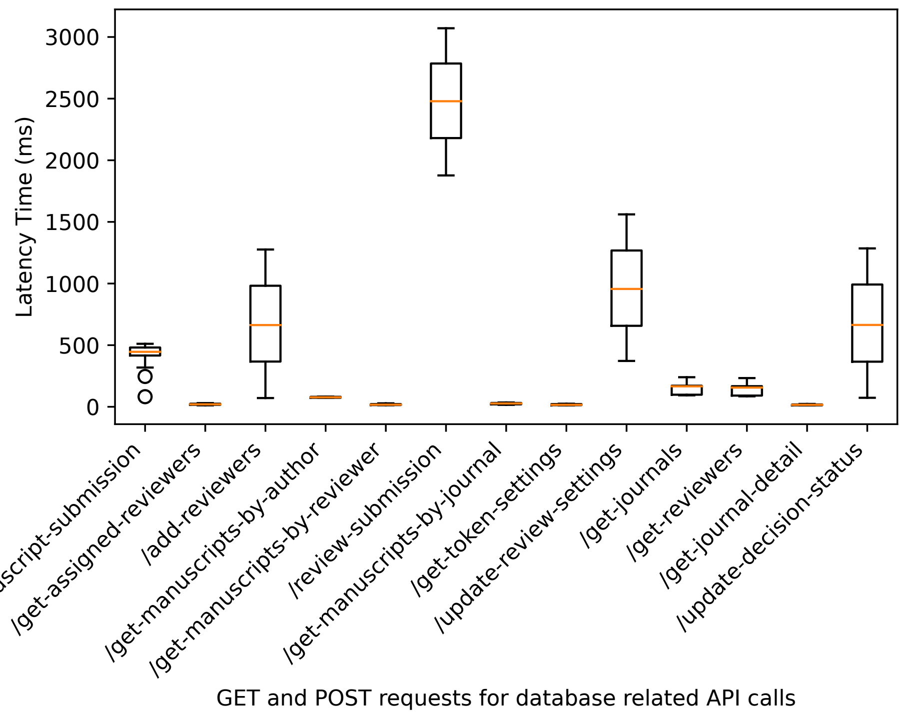


Figure 6.4: Request latency of API calls for database connection with 50 concurrent requests

developed solution. We conducted the study with 8 of the original participants. During these interviews, we briefed participants on the research goal and showed them a video demonstration of the prototype. We then asked open-ended questions aimed at:

- Validating whether the design requirements were satisfied.
- Verifying the usability and ease of use of the developed prototype.
- Gathering suggestions for system improvements.

The structure of the user study is detailed in Appendix A.2.

6.2.1 Requirements Satisfaction

In the earlier phase, we defined three system requirements: incentives, flexibility, and trust. To validate that these requirements were met, we asked the participants a series of questions, prompting them to answer yes, no, or maybe, and to provide their reasoning.

Table 6.3 shows the responses of participants to the questions related to requirement satisfaction. All participants expressed confidence in the system's ability to incentivize reviewers to be more active and perform timely reviews. The responses indicate that using two tokens; one for certificates of recognition and another for redeemable reward tokens effectively encompasses various

Table 6.3: Response to requirement validation questions

Requirement	Question	Answer
Incentives	Do you think that reviewers will be motivated by the incentives provided through the platform?	All 8 participants answered ‘yes’.
Flexibility	Do you think the system is customizable to the needs of participating journals?	7 participants answered with ‘yes’ or ‘possibly’ whereas 1 participant answered ‘no’.
Trust	Do you believe the system to be trustworthy and transparent without disrupting the requirements of a blinded peer review process?	7 participants answered with ‘yes’ whereas 1 participant answered with ‘I don’t know’.

motivating factors for researchers to act as reviewers. P4 highlighted the benefits of the designed soulbound tokens for their career progression:

> *“Through public blockchain, it would be a way to showcase the service to their employers, you know to the department head whenever a faculty goes for tenure. So, I think that is a good way to keep track of such reviews.”*

Many participants were encouraged by the possibility of reducing their own publishing and subscription fees with the accumulated credits. P8 specifically addressed the lack of incentives for their contributions and welcomed the research’s objective to provide such credits:

> *“I do feel in general, we don’t get any incentive for being editor or being a reviewer. In fact, when you are trying to get your work published, you have to pay for either... you have to go for subscription based journal, or if it is open access, then you have to pay for APC (Article Processing Charge) charges. So, I really like the idea of giving some sort of token that can be eventually added up and when you are ready to publish your own work, you can use those in exchange of like article processing charges.”*

P3 shared similar sentiments and expressed a preference for cashing out the received credit for any use:

> *“If you can use it to pay for future publication costs, then yes, that would definitely help reviewing for the place that you want to submit to which means I could lower the cost a bit then. Yes, if you could use it for actual money and kind of put that money wherever you want, then even more so, of course. So yeah, I think the rewards would definitely be a little more incentivizing for actually doing reviews.”*

Regarding P3’s comment, while receiving actual money is not currently possible, reward tokens accumulated from reviewing for one publisher can be redeemed for services from another publisher.

Seven out of eight participants believed that the system could be customized to the needs of participating journals. *"the editor seems to have a decent amount of control"*, P1 praised the system's flexibility in supporting various peer review processes (open, single, and double-blind), and P6 remarked, *"I think the journals can decide, they can decide how to give reward"*. This indicates that the system offers substantial flexibility and control to publishers over the review process and implementation of specific reward policies. However, P2 raised a concern that many journals partner with vendors specializing in sophisticated peer-review platforms. The main aim of BARIT is to incentivize peer reviewers, and as P2 suggested, our priority should be integrating the tokens with existing systems rather than reinventing the entire platform.

Similarly, 7 participants believed the system is trustworthy, attributing their trust to the use of blockchain technology:

> *"With the help of blockchain technology, I think it's trustworthy and transparent. As it is immutable, the technology itself can guarantee this."*

Participants not familiar with blockchain had mixed reactions. One participant expressed uncertainty as they weren't familiar with blockchain technology and would expect some proof or certifications in order to trust the system.

Overall, the majority of participants (7 out of 8) are fully confident that the developed BARIT prototype satisfies the system requirements.

6.2.2 Usability and User Interface Evaluation

We asked the participants some questions to understand their likelihood of using the system and their perceived benefits of it. All the participants expressed a positive attitude towards using the system once available, as it would not only benefit reviewers but could potentially make the review process more efficient and easier for editors.

P3 noted that the system would be highly beneficial to reviewers, stating: *"you're finally getting some kind of reward for the time spent reviewing one or more papers for whatever publication you're reviewing them for rather than just volunteering my time."* Reviewers generally found the designed solution quite helpful, as BARIT provides a streamlined platform to track and showcase their review contributions for different journals. They felt reassured that these tokens of recognition would remain available regardless of the continuity of the journals. P4 added that being able to reduce costs related to paper submission or conference registration in return for volunteering their time to review would be highly welcomed. With reviewers motivated to review, editors should find it easier to secure willing qualified reviewers, reducing delays in the peer review process. P6, who serves as an editor, noted the possibility of the system making the review process more efficient:

> *"If the reviewers are rewarded, then they will be more interested to do the reviews, and that way reviewers will not reject or they will be interested. They will participate and that will, you know, make the reviewing time faster, backlog slower, and overall trust improvement will be there. So overall, the reviewing process and the quality of the peer-reviewed publications will also improve."*

Similarly, adjusting reward tokens based on the timeliness of review submissions should encourage prompt reviews from assigned reviewers:

> *"I think you might get people responding to the review deadline if they're getting tokens for it."*

Participants were quite satisfied with the user-friendliness of the designed system. One participant stated, *"I think it looks pretty slick very well designed. And the interface is much better than the existing options out there, such as EasyChair, or EDAS, which are very outdated."* Participants felt that the interface is intuitive and hassle-free to navigate, and the interactions are well-thought-out.

6.2.3 Concerns

All the questions asked in the study were open-ended and semi-structured, with the goal of keeping the conversation flowing naturally and discerning any unique perspectives or concerns of the participants about the system without being constrained by a rigid structure.

One participant expressed concern about user privacy, as a reviewer's identity could potentially be deduced based on the time when they receive their token. P5 stated:

> *"It can be said there might be some information leakage regarding the reviewer. Because, for example, let's say, X reviewed the paper and got the token, and that token was recorded on blockchain. Then, others can see that that token is on the blockchain now, and that it belongs to him. Then we can't say it's not blind anymore."*

As discussed in the previous chapters, we have considered the possibility of a reviewer's information being inferred from the token transfer timestamp if tokens are distributed immediately upon review submission. To address this, all pending reward tokens are distributed once a month, hiding the actual review submission date from the blockchain network. Hundreds of reviewers will receive tokens for their contributions to different journals simultaneously, making it incredibly difficult to pinpoint the reviewer who reviewed a specific manuscript in the past month.

Another major concern brought forward was the possibility of gamification. Reviewers are often intrinsically motivated by the desire to serve the community. Offering extrinsic motivations can have a negative impact on the otherwise voluntary activity of peer reviewing, as discussed by P3:

> *"I worry about human incentive for greed and gaming the system. I could probably review more papers that get more tokens kind of thing, and then my review quality will go down because I'm speeding up review processes for the sake of getting more tokens."*

A possible way to prevent gamification is to delegate the responsibility of approving the submitted review to the editor. Editors can verify that the review meets the minimal standards set by the journal before tokens are assigned to the reviewer. Additional metrics, such as the average speed at which papers are reviewed, can be helpful in detecting dishonest submissions.

6.2.4 Suggestions for Improvement

There were several recommendations made by the participants to improve the system. The participants were specifically asked to share any improvements that could make the system more appealing to the journal editors and reviewers. They suggested both major features to increase efficiency and minor changes to improve usability:

- Tie rewards not only for meeting deadlines but also for other measurable qualities, such as the length and complexity of the reviewed manuscript.
- Provide editors with detailed reviewer statistics, including acceptance rates for review requests, timeliness in submitting reviews, and quality scores based on editorial feedback. These insights would facilitate the identification of qualified and reliable reviewers, enabling informed decisions during the assignment process.
- Integrate the solution with existing publishing platforms used by various journals and conferences for easy adoption.
- Allow publishers to provide reviewers with tailored guidelines, questionnaires and rubrics specific to their journal or conference. This customization ensures that reviewers are well-informed about the expected criteria and format, promoting consistent and high-quality reviews.

Chapter 7
Conclusion

Peer review plays a significant role in the publication of quality research papers to advance the scientific community. However, it is often criticized for its slow pace and inadequate incentives for reviewers. We conducted interviews with experienced researchers to identify key challenges in the peer review process and understand what motivates them to participate as reviewers. We discovered that journals struggle with low acceptance rates for review invitations, as researchers feel undervalued and under-incentivized for their time and effort. By analyzing the different forms of motivations preferred by different reviewers, we designed a blockchain-based peer review platform that aims to provide proper incentives while upholding the integrity of the peer review process.

Our solution leverages a hybrid on-chain/off-chain database system to address the challenge of finding expert reviewers. Recognizing that reviewers are motivated by different factors, we introduce two token incentives: 1) perpetual soulbound tokens that serve as certificates of recognition; and 2) fungible reward tokens that offer financial compensation redeemable for publication-related fees. Soulbound tokens provide verifiable proof of reviewers' contributions, particularly beneficial for early-career academics. To tackle delays in the review process, we provide flexibility to journals to manage the distribution of fungible reward tokens based on the timeliness of review submissions, thereby encouraging prompt feedback. The system is designed to be agnostic to peer review types and preserves user anonymity in order to encourage broader adoption of the system. This prototype follows a double-blind approach which can be easily adjusted to support open and single-blind review systems. The evaluation results demonstrate the system's capability to handle high user traffic and has received positive user perception within the academic community.

7.1 Limitations

One limitation of this research is the relatively small and homogeneous participant pool, primarily composed of individuals from the computer science field. This lack of diversity could introduce bias, reflecting only a narrow subsection of the academic community. Additionally, due to time and resource constraints, we were unable to conduct a quantitative evaluation of the proposed system's impact on improving the quality of peer review responses. While the user study assesses the perceived benefits of the system, actual changes in peer review quality, frequency, and timeliness can only be measured once the system is integrated with existing peer review platforms used by publication venues. Another limitation of the current model is the lack of exploration of the economic aspects of the tokens. Each journal or conference venue has the autonomy to determine the quantity and value of tokens they distribute, which could lead to disparities in token value when exchanged between venues. This discrepancy could affect the perceived fairness and efficacy of the token-based incentive system, highlighting the need for further investigation to ensure equitable token exchange across various publication platforms.

7.2 Future Work

Some of the possible future research work and improvements are discussed below:

- In the future, the reward distribution mechanism could be made more fair and appropriate by enhancing flexibility with metrics such as review quality, length of the paper being reviewed, and experience level of the reviewer along with a penalty when the reviewer fails to submit a review after accepting a review invitation.

- Current version of the prototype requires publishers to sign up without an option to integrate into their existing platform. In the future, integration with conference management systems such as Papercept to utilize accumulated reward tokens towards publication-related fees could make the acceptance of the platform easier.

- A social recovery model could be implemented to address situations where a reviewer loses access to their crypto wallet. In this model, trusted individuals or organizations, such as the journal that originally issued the soulbound tokens, would have the authority to change the associated wallet keys.

- Future research needs to be conducted on setting base guidelines and rules for the number of tokens distributed by all outlets. This could help regulate the use of tokens earned from one publisher to another.

- Integrating with the ORCID API to allow users to link their ORCID accounts with their MetaMask wallet addresses would enable reviewers to publicly verify their review history.

- Based on the user study feedback, a filter option for editors to select reviewers based on their desired expertise and experience level along with their review statistics is a possible future enhancement. This can help editors find the best reviewers from the large review pool.

Appendix A

Interview Questionnaire

This chapter includes the question structure used for the preliminary survey with the researchers and the final prototype evaluation.

A.1 Preliminary Interview

The interviews were semi-structured with open-ended questions. This questionnaire set an opening for the participants to delve deeper into their experiences as reviewers and editors and gain insights for potential solutions. Interviewees were asked different questions based on their experience of serving as a reviewer or an editor. Following subsections lists the questions asked to the interviewees:

A.1.1 Experience Level

1. Designation (student, assistant professor, professor, etc.)
2. How many years have you been involved in academic research?
3. Have you served in an editorial capacity as an associated editor, senior editor, or editor-in-chief?

A.1.2 Questions to Editor

1. For which journals or conferences have you served as an editor?
2. How many years have you serviced in each of those editorial roles?
3. What inspires you to serve as an editor?
4. On average, how many papers does your editorial role require you to handle each year?
5. Which types of reviewing systems have you experienced as an editor (open, single-blind, double-blind, triple-blind)?
   (a) Which type do you think is best for editors? Why?
6. Are there any journals or conferences that stand out as having an effective reviewing system for you as an editor?

(a) If so, what makes the review system effective?

7. Are there any journals or conferences that stand out as having a poor reviewing system for you as an editor?

    (a) If so, what makes the review system poor?

8. What are the challenges in obtaining quality reviews within time. If they mention "finding quality reviewers," then ask

    (a) Why it is challenging to find qualified reviewers?
    (b) What is your strategy for finding qualified reviewers?

9. Do you think it should be standard practice to compensate reviewers?

    (a) If yes, what form of compensation would you find most appropriate? Why?
    (b) What factors should be considered for determining compensation (timeliness of the review, quality of the review, length of the paper being reviewed, quantity of reviews performed for that outlet, quality of enumerate journal/conference)?

A.1.3 Questions to Reviewer

1. What is your motivation for serving as a reviewer?

2. How many different journals and conferences do you regularly review?

3. On average, how many reviews do you perform each year?

4. Which types of reviewing systems have you experienced as a reviewer (open, single-blind, double-blind, triple-blind)?

    (a) Which type do you think is best for reviewers? Why?

5. Are there any journals or conferences that stand out as having an effective reviewing system for you as a reviewer?

    (a) If so, what makes the review system effective?

6. Are there any journals or conferences that stand out as having a poor reviewing system for you as a reviewer?

    (a) If so, what makes the review system poor?

7. Have you received recognition or compensation for serving as a reviewer?

    (a) If yes, what forms of recognition or compensation have you received (thank you letter, reviewing certificate, journal access, subscription or registration discounts, coupons)?

i. Do you find any of them particularly valuable to you? Why or why not?

8. Should it be standard practice to compensate reviewers?
   (a) If yes, what form of compensation would you find most appropriate? Why?
   (b) What factors should be considered for determining compensation (timeliness of the review, quality of the review, length of the paper being reviewed, quantity of reviews performed for that outlet, quality of the journal/conference)?

A.1.4 General Questionnaire

1. What are some improvements that you would like to see in the peer-review publication system?
2. Do any of your previous opinions differ when you think from the lens of an author instead of a reviewer?

A.2 Second Round Interviews – Prototype Demonstration Feedback

Once the development of the prototype was complete, we asked our previous interviewees to provide their feedback on the developed prototype. Our goal was to determine if the goals are met, measure the usability and ease of use of the prototype, and gather any suggestions for improvements. This section lists the questions asked to the interviewees for prototype feedback.

A.2.1 Requirements Satisfaction

1. Do you think that reviewers will be motivated by the incentives provided through the platform?
2. Do you think the system is customizable to the needs of participating journals?
3. Do you believe the system to be trustworthy and transparent without disrupting the requirements of a blinded peer review process?

A.2.2 Usability and Ease of Use

1. How likely are you to use the software in the future?
2. How likely will the software/application be of benefit to the users?
3. How likely is the system to make the peer review process more efficient?
4. How would you rate the system's trustworthiness for a transparent review process?

5. How likely are you to consider a reviewer's authenticity and contribution based on their recognition tokens accumulated?
6. How likely are you to utilize the utility/credit tokens?
7. How difficult do you anticipate using Metamask would be?

A.2.3 Suggestions for Improvements

1. Are there improvements that could make the system more appealing to journal editors?
2. Are there improvements that could make the system more appealing to reviewers?
3. Do you have any other suggestions to improve the system?
4. Do you have any comments and feedback on the overall use of the application?

Appendix B

Smart Contracts

Appendix B presents the smart contracts designed for BARIT. The public code repository is available in GitHub [5].

B.1 Soulbound Token Smart Contract

Soulbound smart contract is responsible for minting soulbound tokens, allocating them to reviewer addresses, and ensuring that they can't be transferred.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/Strings.sol";
import "@openzeppelin/contracts/utils/Base64.sol";

/**
    @title Soulbound Tokens
    @dev A non-transferable ERC721 token contract for rewarding reviewers.
 */
contract Soulbound is ERC721, ERC721URIStorage, Ownable {
    using Strings for uint256;

    // Counter for token IDs
    uint256 private _tokenIdCounter;

    // Mapping to store token IDs owned by an address
    mapping(address => uint256[]) ownerTokenIds;

    // Image for the token
    string internal _rewardImage;

    // Constructor to initialize the contract with default token name, symbol, and image
    constructor() ERC721("SoulBound", "SBT") {
        _rewardImage = 'https://review-rewards.infura-ipfs.io/ipfs/Qmc8dJ1o7B7kZeuhH8DshyU55FZ1JU7PjJ6ShdwuKzEqqV';
    }

    /**
        @dev Overriding the beforeTokenTransfer function to restrict token transfer
        @param from Address from which the token is being transferred
        @param to Address to which the token is being transferred
        @param tokenId Token ID being transferred
     */
    function _beforeTokenTransfer(address from, address to, uint256 tokenId)
        internal
        override
```

```
    {
        require(from == address(0), "Token can not be transferred to another address");
        super._beforeTokenTransfer(from, to, tokenId);
    }

    /**
        @dev singleMintSBT: Function to mint a token and assign it to the given address
        @param to Address to which the token is being assigned
        @param journalString The name of the journal associated with the token
     */
    function safeMint(address to, string memory journalString) public {
        _tokenIdCounter += 1;
        _safeMint(to, _tokenIdCounter);
        ownerTokenIds[to].push(_tokenIdCounter);

        string memory tURI = _generateTokenURI(_tokenIdCounter, journalString);
        _setTokenURI(_tokenIdCounter, tURI);

    }

    /**
        @dev Function to mint multiple tokens and assign them to the given addresses.
        Tokens are minted in one batch per journal.
        @param _tos Array of addresses to which the tokens are being assigned
        @param journalString The name of the journal associated with the tokens
     */
    function bulkMintSBT(address[] memory _tos, string memory journalString) public onlyOwner {
        require(_tos.length > 0, "No tokens to mint");
        uint256 currentTokenId = _tokenIdCounter;
        unchecked {
            for(uint256 i=0; i< _tos.length; i++) {
                currentTokenId += 1;
                _safeMint(_tos[i], currentTokenId);
                ownerTokenIds[_tos[i]].push(currentTokenId);

                string memory tURI = _generateTokenURI(
                    currentTokenId,
                    journalString
                );
                _setTokenURI(currentTokenId, tURI);
            }
        }
        _tokenIdCounter = currentTokenId;
    }

    /**
        @dev Generates the token URI for a given token ID and journal string.
        @param tokenId The ID of the token
        @param journalString The name of the journal associated with the token.
        @return tokenURi The generated token URI
     */
    function _generateTokenURI(uint256 tokenId, string memory journalString)
        private
        view
        returns (string memory) {
        bytes memory attributesPart = abi.encodePacked(
            '{',
                '"trait type": "Journal",',
                '"value": "', journalString, '"',
            '},',
            '{',
                '"trait type": "Contribution",',
                '"value": "', 'Reviewer', '"',
            '},',
            '{',
```

```
                '"trait type": "No of reviews",',
                '"value": "', '1', '"',
            '}'
        );

        bytes memory dataURI = abi.encodePacked(
            '{',
                '"name": "Reward Token #', tokenId.toString(), '",',
                '"image": "', _rewardImage, '",',
                '"description": "A token of recognition awarded to the reviewer for their contribution by reviewing manuscripts submitted to the journal",',
                '"attributes": [',
                    attributesPart,
                ']',
            '}'
        );

        return string(
            abi.encodePacked(
                "data:application/json;base64,",
                Base64.encode(dataURI)
            )
        );
    }

    /**
        @dev Retrieves the tokens owned by a given address.
        @param _account Address for which the token IDs are being fetched
        @return tokenIds Array of token IDs owned by the address
     */
    function getTokensOwned(address _account) public view returns(uint256[] memory) {
        uint256[] memory tokenIds = ownerTokenIds[_account];
        return tokenIds;
    }

    /**
        @dev Burns a token owned by the caller. Restricts burning to token owner.
        @param tokenId Token ID to be burned
     */
    function burn(uint256 tokenId) external {
        require(ownerOf(tokenId) == msg.sender, "Only owner can burn the token.");
        _burn(tokenId);
    }

    // Allow only owner to burn the token
    /**
        @dev Burns a token.
        @param tokenId Token ID to be burned
     */
    function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
    }

    /**
        @dev Retrieves the token URI for a given token ID.
        @param tokenId Token ID for which the URI is being fetched
        @return tokenURI The URI of the token
     */
    function tokenURI(uint256 tokenId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (string memory)
    {
        return super.tokenURI(tokenId);
```

```
    }

    /**
        @dev Updates the image associated with soulbound tokens.
        @param image The new image URL
     */
    function updateTokenImage(string memory image) public onlyOwner {
        _rewardImage = image;
    }
}
```

B.2 Fungible Reward Token Smart Contract

Fungible reward token smart contract is responsible for minting, distribution, and transfer of fungible reward tokens.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol";

/**
 * @title FungibleRewardToken
 * @dev An ERC20 token contract for rewarding reviewers.
 */

contract FungibleRewardToken is ERC20, ERC20Burnable {
    address payable public owner;

    // Mapping for admin access control
    mapping(address => bool) admins;

    /**
    `   @dev Constructor to initialize the contract with default token name, symbol, and image
        @param reward The reward amount to be given to the reviewers
    */
    constructor(uint256 reward) ERC20("FungibleRewardToken", "FRT") {
        owner = payable(msg.sender);
        admins[msg.sender] = true;

        // Mint initial supply to owner
        _mint(owner, 10000 * (10 ** decimals()));
    }

    /**
        @dev Overriding the beforeTokenTransfer function
        @param from Address from which the token is being transferred
        @param to Address to which the token is being transferred
        @param value Token value being transferred
     */
    function _beforeTokenTransfer(address from, address to, uint256 value) internal virtual
        override {
        super._beforeTokenTransfer(from, to, value);
    }

    /**
        @dev Bulk mints tokens to multiple reviewers.
        @param _tos Array of addresses to mint tokens to
        @param amount Amount of tokens to mint
     */
```

```
    function bulkMintFRT(address[] memory _tos,s uint256 amount) public onlyOwner {
        for(uint256 i=0; i< _tos.length; i++) {
            _mint(_tos[i], amount * (10 ** decimals()));
        }
    }

    /**
        @dev Mint desired amount of tokens to a single reviewer.
        @param to Address to mint tokens to
        @param amount Amount of tokens to mint
     */
    function singleMintFRT(address to, uint256 amount) public onlyAdmin {
        _mint(to, amount * (10 ** decimals()));

    }

    /**
        @dev Adds an admin to the contract
        @param _account The address to be given admin privileges
     */
    function addAdmin(address _account) public onlyAdmin {
        admins[_account] = true;
    }

    /**
        @dev Revokes admin privileges from an account
        @param _account The address to revoke admin privileges from
     */
    function revokeAdmin(address _account) public onlyAdmin {
        admins[_account] = false;
    }

    function destroy() public onlyOwner {
        selfdestruct(owner);
    }

    /**
        @dev Modifier to restrict access to only the contract owner
     */
    modifier onlyOwner {
        require(msg.sender == owner, "Only the owner can call this function");
        _;
    }

    /**
        @dev Modifier to restrict access to only the contract admin
     */
    modifier onlyAdmin {
        require(admins[msg.sender], "Only admin can call this function");
        _;
    }
}
```

Acronyms

BARIT Blockchain-based Anonymous Reviewer Incentive Token. iii, vii, viii, 2, 30, 35, 37, 39–42, 44, 46, 48, 51, 52, 54, 55, 57–59, 61, 64, 73

DLT Distributed Ledger Technology. 9, 10

FRT Fungible Reward Token. vii, viii, 45, 47–49, 51, 53, 54, 56, 58, 60

IPFS InterPlanetary File System. 4, 22–24

PoS Proof-of-Stake. 13

PoW Proof-of-Work. 13

SBT Soulbound Token. vii, viii, 45, 47–49, 51, 53, 55–58, 60

References

[1] Permissioned Vs Permissionless Blockchains. https://101blockchains.com/permissioned-vs-permissionless-blockchains/, May 2020.

[2] Steve Walters. Blockchain vs. Database: How are They Different? https://www.coinbureau.com/education/blockchain-vs-database/, Mar 2023.

[3] Ela Kolis. Comprehensive Guide to Choosing Between Metamask and Coinbase Wallet. https://medium.com/@elakolis/comprehensive-guide-to-choosing-between-metamask-and-coinbase-wallet-faa3402674b6, Apr 2024.

[4] Trust Wallet vs MetaMask: An In-Depth Feature Comparison. https://www.historytools.org/vs/trust-wallet-vs-metamask, March 2024.

[5] Pratiksha Shrestha. Github Repository for the project. https://github.com/wsl-miami/peer-review-rewards, April 2024.

[6] Today's Cryptocurrency Prices by Market Cap. https://coinmarketcap.com/, June 2024.

[7] The peer review process. https://authorservices.wiley.com/Reviewers/journal-reviewers/what-is-peer-review/the-peer-review-process.html, [Accessed March 2024].

[8] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. *arXiv preprint arXiv:1906.11078*, 2019.

[9] Soufiane Bouchaara. IPFS in-depth(Myths and Misunderstandings). https://blog.kalvad.com/myths-about-ipfs/, Dec 2021.

[10] Lakshya Kandpal. Unleashing Web Freedom: IPFS. https://bitwisely.hashnode.dev/unleashing-web-freedom-ipfs, May 2023.

[11] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[12] Susan J Henly and Molly C Dougherty. Quality of manuscript reviews in nursing research. *Nursing Outlook*, 57(1):18–26, 2009.

[13] Mark Ware. *Peer review: benefits, perceptions and alternatives*. Citeseer, 2008.

[14] Detmar W Straub. Editor's comments: Type ii reviewing errors and the search for exciting papers. *MIS Quarterly*, pages v–x, 2008.

[15] Joel Henrique Ellwanger and José Artur Bogo Chies. We need to talk about peer-review—Experienced reviewers are not endangered species, but they need motivation. *Journal of Clinical Epidemiology*, 125:201–205, 2020.

[16] Global State of Peer Review 2018. `https://clarivate.com/lp/global-state-of-peer-review-report/`, [Accessed March 2024].

[17] Decisions and Reviews Statistics for 2022. `https://cyberpsychology.eu/announcement/view/3948`, July 2023.

[18] Travis I Lovejoy, Tracey A Revenson, and Christopher R France. Reviewing manuscripts for peer-review journals: a primer for novice and seasoned reviewers. *Annals of Behavioral Medicine*, 42(1):1–13, 2011.

[19] Janine Huisman and Jeroen Smits. Duration and quality of the peer review process: the author's perspective. *Scientometrics*, 113(1):633–650, 2017.

[20] J Lotriet Cornelius. Reviewing the review process: Identifying sources of delay. *The Australasian medical journal*, 5(1):26, 2012.

[21] Recognition for Reviewers. `https://authorservices.wiley.com/Reviewers/journal-reviewers/recognition-for-reviewers/index.html`, April 2024.

[22] Get Recognized for Peer Review. `https://www.apa.org/pubs/journals/resources/peer-review-recognition`, January 2023.

[23] Katherine Bosworth. Reviewer Benefits with MDPI. `https://blog.mdpi.com/2021/06/11/reviewer-benefits/`, June 2021.

[24] Jacalyn Kelly, Tara Sadeghieh, and Khosrow Adeli. Peer review in scientific publications: benefits, critiques, & a survival guide. *Ejifcc*, 25(3):227, 2014.

[25] John C Burnham. The evolution of editorial peer review. *Jama*, 263(10):1323–1329, 1990.

[26] Adrian Mulligan, Louise Hall, and Ellen Raphael. Peer review in a changing world: An international study measuring the attitudes of researchers. *Journal of the American Society for Information Science and Technology*, 64(1):132–161, 2013.

[27] What is Peer Review? `https://www.elsevier.com/reviewers/what-is-peer-review`, Jan 2022.

[28] Armen Yuri Gasparyan, Alexey N Gerasimov, Alexander A Voronov, and George D Kitas. Rewarding peer reviewers: maintaining the integrity of science communication. *Journal of Korean medical science*, 30(4):360, 2015.

[29] Pankaj Kumar Garg, Armen Yuri Gasparyan, Alexey N Gerasimov, Alexander A Voronov, and George D Kitas. Financial incentives to reviewers: double-edged sword. *Journal of Korean Medical Science*, 30(6):832–833, 2015.

[30] Leanne Tite and Sara Schroter. Why do peer reviewers decline to review? A survey. *Journal of Epidemiology & Community Health*, 61(1):9–12, 2007.

[31] Arthur Carvalho, Stanko Dimitrov, and Kate Larson. Inducing honest reporting without observing outcomes: An application to the peer-review process. *arXiv preprint arXiv:1309.3197*, 2013.

[32] Danielle Padula. ISMTE 2016 North American Conference Recap. `https://blog.scholasticahq.com/post/ISMTE-2016-North-American-Conference-Recap/`, August 2016.

[33] Publons Reviewer Recognition Service. `https://authors.acm.org/author-resources/publons`, [Accessed April 2024].

[34] Collabra: Psychology. Peer review: are you actually volunteering your time? `https://medium.com/@CollabraOA/peer-review-are-you-actually-volunteering-your-time-be3ca06b891e`, September 2015.

[35] Reviewer Recognition. `https://publications.copernicus.org/for_reviewers/reward_for_reviewers.html`, [Accessed April 2024].

[36] Bahar Mehmani, Joris van Rossum, and Ursula van Dijk. More Recognition for Reviewers. `https://www.elsevier.com/connect/more-recognition-for-reviewers`, September 2015.

[37] Reviewer Rewards. `https://reviewers.hindawi.com/page/reviewer-rewards`.

[38] Information for Reviewers. `https://ieee-ims.org/publication/ieee-tim/reviewers`, [Accessed March 2024].

[39] Reviewer and Editor Appreciation. `https://www.comsoc.org/publications/journals/ieee-comml/reviewer-and-editor-appreciation`, [Accessed March 2024].

[40] Reviewer Rewards. `https://us.sagepub.com/en-us/nam/reviewer-rewards`, [Accessed April 2024].

[41] Peer Reviewer confirmation (certificate). `https://support.springer.com/en/support/solutions/articles/6000250759-peer-reviewer-confirmation-certificate-`, March 2024.

[42] Web of Science Reviewer Recognition Services. `https://authorservices.wiley.com/Reviewers/journal-reviewers/recognition-for-reviewers/publons.html`, April 2024.

[43] Open Researcher and Contributor ID (ORCID). `https://orcid.org/`, April 2024.

[44] Mark De Rond and Alan N Miller. Publish or perish: Bane or boon of academic life? *Journal of management inquiry*, 14(4):321–329, 2005.

[45] Shantanu Kumar Rahut, Razwan Ahmed Tanvir, Sharfi Rahman, and Shamim Akhter. Scientific paper peer-reviewing system with blockchain, IPFS, and smart contract. In *Research Anthology on Blockchain Technology in Business, Healthcare, Education, and Government*, pages 1029–1060. IGI Global, 2021.

[46] Cat Ferguson, Adam Marcus, and Ivan Oransky. The peer-review scam. *Nature*, 515(7528):480, 2014.

[47] Michel Rauchs, Andrew Glidden, Brian Gordon, Gina C Pieters, Martino Recanatini, François Rostand, Kathryn Vagneur, and Bryan Zheng Zhang. Distributed ledger technology systems: A conceptual framework. *Available at SSRN 3230013*, 2018.

[48] Gwyneth Iredale. 6 Key Blockchain Features You Need to Know Now. `https://101blockchains.com/introduction-to-blockchain-features/`, Nov 2021.

[49] Theodore Kelechukwu Onyejiaku. What are the Characteristics of Blockchain? `https://www.educative.io/answers/what-are-the-characteristics-of-blockchain`, [Accessed March 2024].

[50] Zhen Shao, Lin Zhang, Susan A Brown, and Tuo Zhao. Understanding users' trust transfer mechanism in a blockchain-enabled platform: A mixed methods study. *Decision Support Systems*, 155:113716, 2022.

[51] Mathieu Chanson, Andreas Bogner, Dominik Bilgeri, Elgar Fleisch, and Felix Wortmann. Blockchain for the iot: privacy-preserving protection of sensor data. *Journal of the Association for Information Systems*, 20(9):1274–1309, 2019.

[52] Arthur Carvalho and Majid Karimi. Aligning the interests of newsvendors and forecasters through blockchain-based smart contracts and proper scoring rules. *Decision Support Systems*, 151:113626, 2021.

[53] Benedikt Notheisen, Jacob Benjamin Cholewa, and Arun Prasad Shanmugam. Trading real-world assets on blockchain: an application of trust-free transaction systems in the market for lemons. *Business & Information Systems Engineering*, 59:425–440, 2017.

[54] Arthur Carvalho. Bringing transparency and trustworthiness to loot boxes with blockchain and smart contracts. *Decision Support Systems*, 144:113508, 2021.

[55] Chirag. A Guide to Understand Blockchain Consensus Algorithms. https://appinventiv.com/blog/blockchain-consensus-algorithms-guide/, Jul 2021.

[56] P. Rajitha Nair and D. Ramya Dorai. Evaluation of Performance and Security of Proof of Work and Proof of Stake using Blockchain. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pages 279–283, 2021.

[57] Alden Baldwin. What are the Limitations of Blockchain? https://www.cryptopolitan.com/what-are-the-limitations-of-blockchain/, Feb 2023.

[58] Changhao Zhu, Junzhe Li, Ziyue Zhong, Cong Yue, and Meihui Zhang. A survey on the integration of blockchains and databases. *Data Science and Engineering*, 8(2):196–219, 2023.

[59] Nico. Intro to Ethereum. https://ethereum.org/en/developers/docs/intro-to-ethereum/, August 2023.

[60] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491, 2020.

[61] Nick Szabo. The Idea of Smart Contracts. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html, Apr 1994.

[62] Shafaq Naheed Khan, Faiza Loukil, Chirine Ghedira-Guegan, Elhadj Benkhelifa, and Anoud Bani-Hani. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer Networking and Applications*, 14:2901–2925, 2021.

[63] Aleksei Korobeinikov. 9 Best Programming Languages for Blockchain Development. https://blaize.tech/article-type/5-best-programming-languages-for-blockchain-development/, Oct 2023.

[64] Solidity. https://docs.soliditylang.org/en/v0.8.25/. [Accessed March 2024].

[65] Riley Annon. Gas and Fees. https://ethereum.org/en/developers/docs/gas, Mar 2024.

[66] Carla Tardi. What Is Gwei? The Cryptocurrency Explained. https://www.investopedia.com/terms/g/gwei-ethereum.asp, May 2024.

[67] Chris Dannen. *Introducing Ethereum and solidity*, volume 1. Springer, 2017.

[68] Monique J Morrow and Mehran Zarrebini. Blockchain and the tokenization of the individual: Societal implications. *Future Internet*, 11(10):220, 2019.

[69] Gang Wang and Mark Nixon. SoK: Tokenization on Blockchain. In *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing Companion*, pages 1–9, 2021.

[70] Florian Glatz. A blockchain token taxonomy. *Retrieved November*, 29:2019, 2016.

[71] Ravi Chamria. Introduction to Token Standards: ERC-20, ERC-721, ERC-777, and ERC-1155. `https://www.zeeve.io/blog/introduction-to-token-standards-erc-20-erc-721-erc-777-and-erc-1155/`, October 2022.

[72] E Glen Weyl, Puja Ohlhaver, and Vitalik Buterin. Decentralized society: Finding web3's soul. *Available at SSRN 4105763*, 2022.

[73] What Is a Crypto Wallet? A Beginner's Guide. `https://crypto.com/university/crypto-wallets`, Apr 2022.

[74] Will Morris. Get started with MetaMask Portfolio. `https://metamask.io/`, December 2022. [Accessed February 2024].

[75] Juan Benet. IPFS-content addressed, versioned, P2P file system (DRAFT 3). *arXiv preprint arXiv:1407.3561*, pages 1–11, 2014.

[76] What is IPFS. `https://docs.ipfs.tech/concepts/what-is-ipfs/`, [Accessed April 2023].

[77] Dave McKay. What Is the Interplanetary File System (IPFS) and How Do You Use It? `https://www.howtogeek.com/784295/what-is-the-interplanetary-file-system-ipfs/`, Feb 2022.

[78] The Filebase Team. Comparing IPFS To Traditional File Storage Systems. `https://filebase.com/blog/comparing-ipfs-to-traditional-file-storage-systems/`, journal=Filebase Blog, 8 Mar 2023.

[79] IPFS and the problems it solves. `https://docs.ipfs.tech/concepts/ipfs-solves/`, [Accessed April 2023].

[80] Bingqing Shen, Jingzhi Guo, and Yilong Yang. Medchain: Efficient healthcare data sharing via blockchain. *Applied sciences*, 9(6):1207, 2019.

[81] Friðrik Þ Hjálmarsson, Gunnlaugur K Hreiðarsson, Mohammad Hamdaqa, and Gísli Hjálmtỳsson. Blockchain-based e-voting system. In *2018 IEEE 11th international conference on cloud computing (CLOUD)*, pages 983–986. IEEE, 2018.

[82] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, page 21260, 2008.

[83] Stephan Leible, Steffen Schlager, Moritz Schubotz, and Bela Gipp. A review on blockchain technology and blockchain projects fostering open science. *Frontiers in Blockchain*, page 16, 2019.

[84] Tim K Mackey, Neal Shah, Ken Miyachi, James Short, and Kevin Clauson. A framework proposal for blockchain-based scientific publishing using shared governance. *Frontiers in Blockchain*, 2:19, 2019.

[85] ARTiFACTS. ARTiFACTS Launches First-Ever Blockchain-Based Platform for Scientific and Scholarly Research. `https://www.prnewswire.com/news-releases/artifacts-launches-first-ever-blockchain-based-platform-for-scientific-and-scholarly-research-300615989.html`, 19 Mar 2018.

[86] Bela Gipp, Corinna Breitinger, Norman Meuschke, and Joeran Beel. CryptSubmit: Introducing Securely Timestamped Manuscript Submission and Peer Review Feedback Using the Blockchain. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–4, 2017.

[87] Eman-Yaser Daraghmi, Mamoun Abu Helou, and Yousef-Awwad Daraghmi. A blockchain-based editorial management system. *Security and Communication Networks*, 2021:1–17, 2021.

[88] Antonio Tenorio-Fornés, Viktor Jacynycz, David Llop-Vila, Antonio Sánchez-Ruiz, and Samer Hassan. Towards a decentralized process for scientific publication and peer review using blockchain and ipfs. In *Hawaii International Conference on System Sciences*, 2019.

[89] Andrea Lisi, Andrea De Salve, Paolo Mori, Laura Ricci, and Samuel Fabrizi. Rewarding reviews with tokens: An ethereum-based approach. *Future Generation Computer Systems*, 120:36–54, 2021.

[90] Tarun Vihar Tumati, Yun Tian, and Xunfei Jiang. A soulbound token certificate verification system (sbtcert): Design and implementation. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0345–0350. IEEE, 2024.

[91] Arthur Carvalho, Chad Anderson, and Liudmila Zavolokina. Designing incentives for attracting peer reviewers to information system conferences. *Communications of the Association for Information Systems*, 54(1):33, 2024.

[92] Taotao Wang, Soung Chang Liew, and Shengli Zhang. Pubchain: A decentralized open-access publication platform with participants incentivized by blockchain technology. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–8. IEEE, 2020.

[93] Michael Spearpoint. A proposed currency system for academic peer review payments using the blockchain technology. *Publications*, 5(3):19, 2017.

[94] Pluto Network. White Paper Draft (ver 0.4). Technical report, January 2018. https://assets.pluto.im/Pluto_white_paper_v04_180719_1355_BSH.pdf, Accessed: 22 March 2023.

[95] Imtiaz Khan and Ali Shahaab. A peer-to-peer publication model on blockchain. *Frontiers in Blockchain*, 4:615726, 2021.

[96] Sina Rafati Niya, Lucas Pelloni, Severin Wullschleger, Andreas Schaufelbühl, Thomas Bocek, Lawrence Rajendran, and Burkhard Stiller. A blockchain-based scientific publishing platform. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 329–336. IEEE, 2019.

[97] Whitepaper: Accelerated scientific publishing. https://docs.orvium.io/Orvium-WP.pdf, Sep 2019. Accessed: 23 March 2023.

[98] Yan Zhou, Zhiguo Wan, and Zhangshuang Guan. Open-pub: A transparent yet privacy-preserving academic publication system based on blockchain. In *2021 International Conference on Computer Communications and Networks (ICCCN)*, pages 1–11. IEEE, 2021.

[99] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.

[100] Edward E Lawler III and J Lloyd Suttle. Expectancy theory and job behavior. *Organizational behavior and human performance*, 9(3):482–503, 1973.

[101] Deborah J Stipek. Motivation to learn: From theory to practice. 1993.

[102] Michael Hechter and Satoshi Kanazawa. Sociological rational choice theory. *Annual review of sociology*, 23(1):191–214, 1997.

[103] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[104] What is IPFS? https://docs.ipfs.tech/concepts/what-is-ipfs/, [Accessed April 2024].

[105] reindigital. Solana vs. Ethereum: A Comprehensive Comparison. https://www.zenledger.io/blog/solana-vs-ethereum/, September 2023.

[106] Avalanche (AVAX) blockchain - getting to know the Ethereum rival. https://www.atato.com/avalanche-avax-blockchain-getting-to-know-the-ethereum-rival/.

[107] Li Peng, Wei Feng, Zheng Yan, Yafeng Li, Xiaokang Zhou, and Shohei Shimizu. Privacy preservation in permissionless blockchain: A survey. *Digital Communications and Networks*, 7(3):295–307, 2021.

[108] Paolo Tasca and Claudio J Tessone. Taxonomy of blockchain technologies. Principles of identification and classification. *arXiv preprint arXiv:1708.04872*, 2017.

[109] What Are Utility Tokens? https://www.coinsmart.com/articles/what-are-utility-tokens/. [Accessed April 2024].

[110] React The library for Web and Native User Interfaces. https://react.dev/. [Accessed Februrary 2024].

[111] Run Javascript Everywhere. https://nodejs.org/en, [Accessed September 2023].

[112] Express 4.19.2: Fast, Unopinionated, Minimalist Web Framework for Node.js. https://expressjs.com/. [Accessed February 2024].

[113] Oracle Autonomous Database. https://docs.oracle.com/en/cloud/paas/autonomous-database/index.html. [Accessed February 2024].

[114] Will Morris. What Is NGINX? An Overview of the Basics. https://www.elegantthemes.com/blog/wordpress/what-is-nginx-an-overview-of-the-basics.

[115] Understanding OpenStack. https://www.redhat.com/en/topics/openstack, October 2021.

[116] Docker Builds: Now Lightning Fast. https://www.docker.com/. [Accessed March 2024].

[117] Nico. Ethereum Accounts. https://ethereum.org/en/developers/docs/accounts/, August 2023.

[118] Jan Werth, Mohammad Hajian Berenjestanaki, Hamid R Barzegar, Nabil El Ioini, and Claus Pahl. A Review of Blockchain Platforms Based on the Scalability, Security and Decentralization Trilemma. *ICEIS (1)*, pages 146–155, 2023.

[119] Hardhat Overview. https://hardhat.org/hardhat-runner/docs/getting-started, April 2024.

[120] Foundry Book. https://book.getfoundry.sh/, [Accessed April 2024].

[121] AutoCannon. https://www.npmjs.com/package/autocannon, February 2024.