

Decentralized Escrow and Milestone Verification Systems for Freelance Digital Services Using Smart Contracts

Yash Dhankhar¹, Gagan¹, and Mukta²

¹Department of Computer Science and Engineering (Cyber Security)

²Associate Professor, School of Engineering and Technology

Sushant University, Gurugram, India

¹{yash.220btccse100, Gagan.220btccse133}@sushantuniversity.edu.in, ²mukta@sushantuniversity.edu.in

Abstract—The gig economy is experiencing a revolutionary upheaval across the world. However, such an economic phenomenon continues to be dependent on central intermediaries whose high costs in terms of service fees and arbitration processes pose significant challenges. Traditional gig freelancing websites work as neutral mediators; however, their centralized structure means that high commissions—previously amounting to 20 percent—are commonplace. This paper attempts to develop an automated decentralized escrow solution via Ethereum smart contracts. The first limitation to address with current web3-based solutions is the "subjectivity gap" that arises when validating off-chain performance of digital services. In light of this, Decentralized Oracle Networks (DONs) will be introduced to automate the process of objective milestone verification using cryptographic proof of work through GitHub merge events. Economic feasibility analysis will be done through comparisons between the conventional percentage system of commissions and the Layer-2 flat-rate cost savings provided by gas payments. As a security-centric analysis, a security review will be performed in regards to vulnerabilities such as "prodigal" or "suicidal" code to ensure fund safety amid India's legal and tax environment.

Keywords—Blockchain, Smart Contracts, Decentralized Oracles, Escrow Systems, Web3, Cyber Security, Milestone Verification, Gig Economy, Ethereum, Cryptographic Hashing.

I. INTRODUCTION

There has been an essential paradigm shift in the dynamics of global labor in the past decade with the transition from the traditional concept of employment to a more fluid, digital gig economy [15]. This phenomenon is particularly prominent in the case of the Indian subcontinent where large numbers of specialized professionals such as software developers and digital designers are now independent contractors working for clients globally [15]. The downside of this shift towards global inclusivity is its dependence on centralized architecture which can pose restrictions on finance and trust [3].

Traditionally, centralized freelance markets like Upwork and Fiverr act as central escrow agents dealing with the issue of non-trust between individuals who do not share any pre-existing mutual trust [1]. The problem, however, is that these trusted agents charge what is termed the "Trust Tax" [4]. These agents may levy hierarchical commissions for initial agreements that may be as high as 20%, posing a considerable financial burden on independent developers and freelancers [3]. Additionally, because they control the ledger and the process of dispute resolution, freelancers are subject to client ghosting, appropriation of unpaid services, and arbitrary freeze of funds during the 5-to-14-day security hold period.

The advent of principles behind Web3 can be viewed both theoretically and technically as providing an answer to the presence of such intermediaries [2]. Through the use of blockchain technologies, it becomes possible to perform "trustless" transactions with the security of assets ensured by unchangeable programming code as opposed to the supervision of companies [9]. Prior work has already examined the automation of job bids and payments with Ethereum-based smart contracts and has identified them as self-executing escrow services capable of holding funds cryptographically until certain criteria have been fulfilled [5].

Nonetheless, one of the biggest obstacles facing the widespread implementation of such a system is the problem with Oracle Services [8]. Being, by design, an isolated

network, a smart contract cannot independently verify any off-chain actions, like the merging of code in a GitHub repository or deploying a website [7]. Absent a way to connect these milestones to the blockchain network, the decentralized escrow system can only perform automatic token transfers [8]. This project seeks to address this issue through the implementation of Decentralized Oracle Network architectures (DONs), capable of feeding verified external data to the escrow process [6].

Moreover, considering that the operation of these technologies entails custody of valuable digital assets, security becomes the primary consideration [10]. There are some peculiar vulnerabilities of smart contracts, such as susceptibility to re-entrancy attacks and logic errors, which could result in the loss of funds forever [12]. In order to design a framework for decentralized freelancing services that would be not only secure but also economically efficient, the current study intends to conduct a security audit by means of practical testing [11], [13], [14].

II. PROBLEM DEFINITION AND OBJECTIVES

A. Problem Statement

In today's era, the freelance economy is plagued with inefficient systems that seek profit maximization from the system rather than enabling financial freedom to its participants [15]. As centralized intermediaries have managed to provide global access to talent effectively, yet, there are three critical problems associated with their existence that prevent the development of the decentralized workforce:

1. **Economic Barriers and "Trust Tax":** The high commission rates levied by the intermediaries, historically assessed as 20%, pose a major obstacle for aspiring freelancers in developing countries such as India [3], [15]. There is also an instance where the cost of creating trust outweighs the service fee paid by the client to the freelancer, making the entire transaction inefficient [4].
2. **Buyer and Seller's Problem:** The primary issue here is the absence of a credible third party that ensures trust

*Corresponding author. Permanent contact: yashdhankhar5656@gmail.com

between the two participants of the process [4]. In the absence of any mediation, the client faces the fear of not receiving the services he pays for, whereas the freelancer feels threatened by the possibility of non-payment once the work is completed, including "appropriation of work" done by the clients [1], [14].

3. **Verification Delay and Centralization Bias:** The existing verifications depend on the clients' manual approval or central support teams, leading to delays in payments by 5 to 14 days [1]. Such delays lead to cash flow uncertainty and vulnerability to "client ghosting," with a bias toward favorable decisions for the payers [1], [3].

B. Research Objectives

This paper seeks to address the above-mentioned issues and come up with a technically viable solution in line with the 8th-semester engineering course requirement through the following objectives:

1. **Objective 1:** To conduct an exhaustive examination of the factors making centralized freelance platforms expensive and inefficient based on contemporary economic studies and reviews [1], [13], [15].
2. **Objective 2:** The objective is to create a theoretical model for a dual deposit smart contract that allows for the safekeeping of funds until both parties have a mathematical motivation to complete their responsibilities [4], [5].
3. **Objective 3:** It is essential to determine the automation process by which external factors like the merging of a GitHub repository can be leveraged using Decentralized Oracle Networks (DONs) to validate technical work impartially [6], [7], [8].
4. **Objective 4:** An exhaustive comparison between the cost of blockchain transactions (gas fees) and the existing high-commission 20% fee model is required [3], [15].
5. **Objective 5:** This dissertation will explore the feasibility of the proposed decentralization solution in practical applications with respect to asset security and compliance in the Indian freelance marketplace [11], [12], [15].

III. EVOLUTION TOWARDS THE DECENTRALIZED WEB (WEB3) AND ETHEREUM MECHANISMS

A. Paradigm Shift towards Web3

The evolution of the Internet can be divided into three distinct eras, all of which have transformed the relationship between users, data, and centralized systems. While Web1 era was dominated by static "read only" data and Web2 provided "read write" functionality in addition to creating an opportunity for extreme centralization [3]. Large companies within Web2 work as mediators in both data exchange and financial transactions charging very high fees for their services [3].

Web3, on the other hand, allows users to exercise complete control over their data with its "read write own" concept due to the use of decentralized ledger technology (DLT) making transactions secure, immutable, and free from the risk of a single point of failure [9]. This means that for freelance marketplaces, transactions made with Web3 technologies ensure trustless transactions without relying on any private company's goodwill and reputation [1].

B. Ethereum and Smart Contract Mechanics

The reason why Ethereum remains a core technology for dApps lies in its Turing complete runtime engine called Ethereum Virtual Machine (EVM) [2]. In contrast to initial blockchains created merely for the transfer of funds, Ethereum offers more advanced functionality in form of deploying smart contracts that automatically execute predefined actions depending on the conditions defined within these scripts [9].

When applied to the web3 freelance service, these scripts play the role of an automated escrow agent. Developed in Solidity programming language, the contract code will include all necessary milestones and cryptographic functions that must be fulfilled to release funds [2]. As the contract code is open-source and not changeable after deployment, all parties can agree on its terms in advance, thus eliminating the so-called "trust tax" of human escrows [4].

C. Cryptographic security and trust lessness

One of the most crucial features of the web3 freelancer business model is related to the use of cryptographic mechanisms ensuring data immutability and preventing post-factum manipulation of information [9]. Thanks to secure hashing functions implemented in the architecture, the system will allow creating a reliable connection between off-chain work (commitment to a code change in Git) and on-chain payments [1].

IV. THE "ORACLE PROBLEM" AND DECENTRALIZED ORACLE NETWORKS (DONs)

A. Definition of Oracle Problem

One of the weaknesses of blockchain technology is its deterministic and autonomous nature. By definition, blockchains cannot access any off-chain data source or prove the occurrence of an event that happens outside their ecosystem [8]. The isolation guarantees security and consensus, but creates what is known as the Oracle Problem, a smart contract being incapable of independently verifying the occurrence of a certain trigger like completing a software milestone or uploading files to repositories [8]. A freelancing escrow system cannot function without having a bridge to see its off-chain transactions [7].

B. Architecture of Decentralized Oracle Networks (DONs)

To fill the gap of connectivity between the blockchain and external data sources, decentralized oracle networks provide a mechanism called "off-chain computation" that feeds back off-chain information to the blockchain [6]. In contrast to a centralized oracle that poses a risk of becoming a single point of failure and reintroduces a "trusted intermediary," DONs utilize the services of a network of nodes for fetching and validating the data [6].

For this case study, it will be possible to use Chainlink 2.0, an advanced version that supports advanced oracle triggers [6]. This means that the nodes can listen to external Application Programming Interfaces (APIs), such as GitHub and Figma, and provide a cryptographic proof of successful delivery to the smart contract [7]. This will ensure that the payment is released based on verifiable data rather than arbitrary subjective approvals [1].

C. Ensuring Data Integrity and Verification Logic

In order to avoid the risk of malicious nodes giving incorrect information, the decentralized oracle networks apply the process of consensus building, whereby multiple nodes should agree that the off-chain outcome is acceptable for the smart contract [7]. This is particularly important for the services that have a lot of subjectivity due to their nature and therefore, often cause disputes [8]. For instance, the smart contracts will ensure that the outcome of work that needs to be completed matches the original request through the cryptographic hash [4], which provides the necessary technical bridge between Code as Law and Code as a mediator [9].

V. LITERATURE REVIEW

A. Synthesis of Decentralized Labor Models

Several models have been created within the context of the gig economy to counteract existing centralized labor organizations. The research on DLancer showed that such architecture can incorporate job bids, escrow, and DAO governance [1]. In addition, using Ethereum as a platform is possible for organizing job postings and payments [2]. According to previous studies, centralization allows for greater participation but leads to increased dependency and high financial costs because of higher transaction fees [3].

B. Escrow Protocols and the Buyer-Seller Dilemma

Innovative ideas related to smart contract-based escrow protocols are known for helping in reducing the rate of transaction failures and solving the "Buyer-Seller Dilemma" [4]. It refers to introducing a protocol based on a dual deposit escrow, which will ensure that both the seller and the buyer will be involved in financial transactions [4]. Such approach eliminates the need for human interaction and is vital for addressing the "Trust Tax," which equals 10% to 20% of earnings of a freelancer [5].

C. Research Gap Identification

While the theory behind decentralized escrow is robust, research indicates the presence of critical gaps concerning the automation of digital milestones [13]. The majority of existing implementations depend upon manual verification or basic transfer of tokens [1]. Although Decentralized Oracle Networks have been identified as solutions for off-chain interoperability, the use of such networks to verify technical deliveries, including GitHub pull requests and design asset deployments, is relatively unstudied [7]. This paper intends to fill this gap through the development of a verifiable technical trigger framework.

D. Legal and Security Perspectives

Academic literature on the legal issues surrounding smart contracts asserts that "Code as Law" does not entirely supersede traditional legal principles [14]. The implementation of smart contracts in practice, particularly within the Indian context, is fraught with difficulties related to enforceability and adherence to regulation [15]. Additionally, the safety of smart contracts is critical; the absence of thorough security analysis exposes escrowed funds to the risks of "greedy," "prodigal," and "suicidal" code vulnerabilities that could result in irreversible monetary loss [10], [12].

VI. PROPOSED SYSTEM ARCHITECTURE

A decentralized system will have the three-layer architecture, ensuring a distinct separation between user interaction, blockchain logic, and external data validation [1]. In addition to decentralization, the "Trust Layer" of the system removes human intervention from the process, thus creating an automated system of execution [2].

A. The Three-Layer Architecture

User Interface (UI) Layer: This top-level module provides means for negotiation between a client and a freelancer and serves as an entry point for contract development, milestone definition, and project initiation [2]. As a web application interface to blockchain technology, it makes interactions with the smart contract seamless and convenient [1], [13].

Smart Contract (Escrow) Layer: This is the "Escrow Agent" layer that consists of Solidity-based contracts running on the Ethereum or Polygon blockchain [2], [9]. This module manages the funds, accepts deposits from both parties using the logic of dual-deposit as a game-theoretic guarantee, and releases payments to the recipient once certain predefined events are triggered via the Oracle layer [4], [5].

Oracle Network Layer: This is a middle-layer that mediates the on-chain and off-chain components of the system [6]. This layer consists of independent nodes that monitor progress in relation to the technical aspect of the project in an external ecosystem such as GitHub or Figma [7]. The decentralized consensus approach of the Oracle layer ensures that no single data point or node can trigger a payment transaction fraudulently or provide wrong information about the milestone fulfillment [8].

B. Transaction Process and Logic

The logic of the transactions within the decentralized system works according to an ordered chain of steps that provides security. The first step involves defining milestones that must be completed as part of the work done by the freelancer, e.g., "Repository Merge" or "API Deployment" [1]. After that, the client deposits the project funds in the smart contract and the freelancer – proportional collateral to prevent appropriation of the work done as part of the "Buyer-Seller Dilemma" [4].

Once the task is done, the freelancer uploads a cryptographic hash of the milestone fulfilled to the Oracle nodes. These nodes check whether the uploaded cryptographic hash matches the agreed upon one [7]. Upon consensus being achieved, a transaction is initiated in the form of a call to the smart contract for an immediate settlement without any hold periods (5-14 days) characteristic of centralized systems [1], [6].

VII. SECURITY ANALYSIS AND THREAT MODELING

With the adoption of the suggested decentralized solution, the security of the smart contract layer becomes the key factor affecting the integrity of the stored escrow funds [10]. Being immutable after deployment, the contract may contain logical flaws leading to irrecoverable losses, which is an issue of high importance given the freelance sector that often involves high-value digital assets exchange [12].

A. Classification of Smart Contract Vulnerabilities

For a proper defense-in-depth approach, we conduct a comprehensive analysis of the proposed escrow contract in relation to commonly used patterns of vulnerabilities described in literature:

1. **Transaction-Ordering Dependence (Front-running):** It takes place when the transaction ordering in the block impacts the outcome of a smart contract, and poses significant threats during the disputes stage [10].
2. **Re-entrancy Attacks:** An extremely dangerous vulnerability involving an external contract calling back to the escrow contract before the initial call has finished execution, enabling an attacker to repeatedly withdraw the same amount [11].
3. **Greedy, Prodigal, and Suicidal Contracts:** Based on formal definitions, three types of risks exist, namely, a "greedy" contract keeping funds forever without providing any way of their recovery; a "prodigal" contract sending the funds to a third party by mistake; and a "suicidal" contract capable of being shut down by an adversary leading to the irreversible loss of the funds kept in escrow [12].

B. Formal Verification and Automated Analysis

To overcome the above challenges, Securify principles can be used in the suggested system architecture to provide practical security analysis [11]. By analyzing the bytecode of the contract, it becomes possible to detect whether the `releasePayment()` method can only be called with the confirmed oracle consensus agreement [11]. According to literature analysis, automated testing and symbolical execution become important tools for discovering these types of vulnerabilities during the development stage [13].

C. Game-theoretic Security and Collateral Logic

Besides the code vulnerabilities, a number of behavioral risks must be addressed as well. The dual-deposit escrow mechanism implemented in the system provides a "cheat-proof" game-theoretic environment where the costs of cheating are too high to actually engage in it [4]. Therefore, the financial penalties for breaching the contract terms serve as a strong deterrent in a trustless system [5].

D. Oracle Integrity and Data Risks

Security of the milestone verification process also depends on the integrity of Decentralized Oracle Network (DON) [6]. With a single centralized oracle, there will be a risk of data manipulation, therefore, a distributed DON will be used in our system [8]. As multiple independent nodes reach consensus, they trigger the execution of the `releasePayment()` method by signaling "milestone completed" status [6], [7].

VIII. MILESTONE VERIFICATION AND CRYPTOGRAPHIC LOGIC

The key improvement introduced by the decentralized platform proposal concerns the use of objective technical triggers for release based on cryptography [1]. The system uses cryptographic tools and oracle logic to make sure that the transfer of funds occurs only when the delivery of digital goods is verified [4].

A. Digital Fingerprinting using Cryptographic Hashes

To ensure the deliverable remains intact and the project owner is satisfied with it, a unique digital fingerprint of the deliverable can be obtained using a cryptographic algorithm, such as SHA-256 [1]. As soon as a freelancer completes their technical task, e.g., creates a certain part of the code, the deliverable is hashed and uploaded onto the blockchain [5]. On-chain storage of the file provides a way to avoid any issues regarding the "appropriation of work"; since the client cannot argue that the work delivered was not the same as agreed upon or it was incomplete, the provided files should match the fingerprint already recorded [4], [9].

B. Staged Milestone Logic

For large and long-term technical projects, a staged approach can be used for milestone tracking, making sure the process runs smoothly while reducing the financial risks [1].

1. **Decomposing the Work:** Projects are divided into milestones (UI design, Backend API Development, Integration, etc.) with an assigned portion of escrowed funds [2].
2. **Staggered Payments:** Payments will be paid out only upon successful completion of the milestone, providing incentives to complete the work properly [4].
3. **Milestone State:** The smart contract contains information about the current project status (Pending, In-Verification, Completed); hence, any attempt to pay out prematurely or unlawfully will not be allowed [5].

C. DON-Based Execution Triggers

The change in the project state from "pending" to "completed" will be governed by Decentralized Oracle Networks (DONs) [6]. DONs monitor the technical environment outside the system in order to provide objective evidence of project completion:

1. **Repository Monitoring:** The oracle network will be set up to watch for specific events on version control systems, such as "Pull Request Merge" into the private GitHub repository [7].
2. **Verifying the Truth:** Independent DON nodes obtain information about the event and provide proof in order to resolve the problem of trusting oracles [8].
3. **Broad-casting Results:** When all DON nodes agree that a particular merge criteria is met, a signed transaction will be sent to the smart contract, which will trigger a settlement automatically [6], [7].

With the help of the described procedure, freelancers receive payments instantly, avoiding settlement delays that can last from 5 to 14 days and potential "ghosting" from centralized sites [1], [3].

IX. COMPARATIVE ECONOMIC MODEL

One of the key goals of this research is to measure the potential economic gains resulting from the transition from the centralized labor markets into the decentralized escrow ecosystem. With the help of the comparative analysis of the "Trust Tax", we will be able to assess whether the blockchain settlements are economically viable solution for the digital workforce [1].

A. Centralized "Trust Tax" Model

Freelance platforms charge an overall percentage fee, which typically differs based on the number of jobs completed by the worker for a certain client [3]. Usually, the fees of the largest platforms range from 20% for the first \$500 worth of work [3]. Thus, in the case of a highly-commissioned project for \$1,000, a freelancer would pay a \$200 fee to the third-party [3], [15]. This deduction of the profits from the freelancer's income is seen by the platforms as the cost of escrow, arbitration, and search functions in the labor marketplace [4]. Nevertheless, the fee rate of 20% makes it difficult for independent professionals in India to operate and survive in the gig economy [15].

B. Decentralized Flat-Rate Model

Contrary to the percentage-based economic model of Web2 services, the proposed decentralized system operates on flat-rate costs based on the so-called "gas fees" [3]. As for Layer-1 networks (such as Ethereum), gas fees may fluctuate depending on the current conditions [3]; in contrast, on scalable Layer-2 systems (Polygon and Arbitrum, among others), the cost of performing transactions including the fund locking process, milestone confirmation, and release of funds rarely exceeds \$1 [2], [3].

As seen from the comparative table above, the economic advantage of the decentralized system becomes more significant as the total project price goes up. Namely, a \$1,000 commission on a project will amount to only about the sum paid for a gas fee, leaving 19.9% more profit to the freelancer [3].

C. The Effect of "Opportunity Cost"

Additionally, the economic model of the decentralized platform takes into consideration the opportunity cost of the funds used in the transaction. Specifically, many centralized systems impose a security hold of 5 to 14 days until all the earned money is deposited onto the freelancer's bank account [1]. However, when the freelancer receives a payout in India, he or she has to incur additional expenses associated with international transfers, and exchange unfavorable currency rates [15]. The proposed architecture of the smart contract allows achieving "Instant Settlement", whereby funds are automatically transferred into the wallet immediately after the milestone is verified by the oracle [1], [6].

D. Dual-Deposit Economic Equilibrium

As a way of making sure that the proposed system works in a completely "cheat-proof" fashion, its economic model includes the mechanism of dual deposit [4]. By forcing both parties to lock the corresponding sums of tokens in the contract, one creates a "financial punishment" in case the client refuses to approve a certain merge or, conversely, the freelancer delivers a poor-quality job [5]. The loss of collateral serves as a powerful deterrent against any kind of cheating [4]. In this regard, the decentralized architecture will work efficiently without costly arbitration procedures [4], [14].

X. OBSTACLES TO ADOPTION OF DECENTRALIZED ESCROW IN THE INDIAN CONTEXT

While there are numerous benefits to decentralized escrow, certain socio-technical obstacles and regulatory roadblocks

should be addressed to allow its wide adoption in the context of the Indian gig economy [14], [15].

A. Lack of Legal Clarity

Although the legal recognition of smart contracts has been achieved through the passage of the Information Technology Act of 2000, no provisions concerning self-execution through automated code were made therein [14]. The paradigm of Contracts ex Machina is hindered in courts due to the absence of the trusted mediator which makes liability determination and resolving disputes more difficult [14]. Consequently, unless clear legal recognition of the nature of these contracts is provided by Indian legislature, decentralized escrow will not be able to provide adequate protection against breaches [14].

B. Complexity of Freelancers' Taxation

In addition to potential disputes arising from contract execution, the issue of freelancers' taxation should be considered [15]. Given the complexity of the current system of VAT and Tax Deducted at Source (TDS), implementing a decentralized escrow protocol necessitates the creation of a mechanism of income declaration and payment of professional income taxes [15]. As the case of NITI Aayog clearly demonstrates, tax compliance should be ensured to formalize gig economy in India [15]. Moreover, the issue of cryptocurrency valuation poses problems with tax payments; consequently, the usage of stablecoin-based settlement layer should be considered to avoid complications with currency conversion [8], [15].

C. User Experience Barriers

The main advantage of using a decentralized system in the context of the gig economy marketplaces is its ability to provide similar user experience to the traditional Web2 applications [3], [13]. At the moment, however, the necessity to interact with digital wallets, pay network fees, and maintain personal keys creates a major obstacle for non-tech-savvy freelancers and clients [1]. Transitioning from convenience of centralized escrows to the world of decentralized sovereignty involves education initiatives aimed at ensuring user awareness about risks related to smart contract management [12], [13].

D. Infrastructure Issues

While the availability of broadband internet increased dramatically in recent years, Internet service quality in tier-2 and tier-3 cities is still far from optimal [15]. In order to function properly, the decentralized system will have to interact constantly with the blockchain and DON networks [6], [9]. Consequently, any disruption in the process of information transfer between oracles can lead to delays in payments, which can be a major drawback of decentralized escrow compared to traditional systems [8].

XI. ANALYSIS AND SECURITY AUDIT

As the main objective of this research is to create a framework that is both reliable and cost-effective, this section presents an analytical assessment of the proposed smart contract algorithm based on scholarly benchmarks [13]. Rather than the implementation, we perform a theoretical security audit of the proposed architectural patterns to assess their resistance to any potential failures associated with market factors and technology [11], [13].

A. Logical Proof of the Proposed Escrow Protocol

To determine whether the proposed dual-deposit escrow algorithm guarantees logical integrity in terms of game theory, we use the "Buyer and Seller's Dilemma" criteria [4].

1. **Proof of commitment:** To avoid abandonment risk identified in the decentralized employment model, the freelancer needs to make a deposit before being assigned a task, thereby having an economic motivation to complete it [4], [5].
2. **Automatic settlement:** Instead of relying on manual intervention from clients, the algorithm implements an automatic process through a Decentralized Oracle Network trigger that eliminates a "Subjectivity Gap" and prevents the so-called "client ghosting", where a client keeps withholding the money despite delivering work on time [1], [6].
3. **Cryptographic Verification:** Upon receiving evidence about the completion of a technical milestone (a GitHub merge, in particular), the contract automatically releases funds to the freelancer verified by multiple oracle nodes [7], [8].

B. Theoretical Security Audit

By following the methodology used in a practical security analysis, the proposed Solidity patterns are assessed based on common vulnerabilities in smart contracts [11].

1. **Re-entrance and Overflow Prevention:** To protect the application from reentrance and overflow vulnerabilities that have been observed in various decentralized finance applications, the design is built using "Checks-Effects-Interactions" pattern and uses modern versions of the programming language [10], [11].
2. **Classification of Vulnerabilities:** Auditing helps to guarantee that the application is classified neither as a "greedy" (which may lock all funds indefinitely), nor as "prodigal" (that leaks funds in some way), and "suicidal" (that can be destroyed at will) [12].
3. **Oracle Reliability:** In terms of oracle vulnerability, the multi-nodes consensus approach employed to form DON allows us to assume that if a minority of oracles provide erroneous data, there will still be an exact trigger to the escrow contract [6], [8].

C. Estimated Performance Metrics

Based on data on comparative performance provided by the current gig economy and Blockchain infrastructure, we evaluate the efficiency of the system [3], [15].

1. **Fees:** Based on calculations presented by analytical models, the platform's profit from commission on freelance payments is expected to be 20% compared to less than 1% for blockchain-based Layer-2 network transactions for escrow contracts [3].
2. **Speed of settlement:** Instead of waiting between 5 and 14 days for the release of the funds, the blockchain-based application theoretically requires no more than a few minutes (typically 5) until the block is confirmed and oracle nodes reach consensus [1], [6].
3. **Regulatory compliance:** The projected benefits for India come from its transitioning towards the formalization of gig economy payments, which requires a secure and regulated

platform of payments in line with future digital assets regulation [14], [15].

XII. CONCLUSION AND FUTURE WORK

A. Summary of Findings

This work has developed an all-encompassing technical and economic solution for the decentralization of freelance service exchange based on the Ethereum-based escrow and Decentralized Oracle Network (DON) [1], [2], [6]. By tackling two major issues inherent to centralized platforms ("Trust Tax" and settlement latency), it was determined that a viable alternative exists for the financial sovereignty of freelance workers [3], [15]. This research demonstrates that the "Buyer and Seller's Dilemma" problem can be solved using dual-deposit escrow, thereby providing mathematical incentive for both sides to complete the transaction and avoid involvement of a third party [4], [5].

In addition, DONs were determined to be the key tool enabling automation of milestone-based payments based on off-chain technical events. By connecting a particular GitHub merge commit hash to a certain payment condition, such a solution solves problems with subjective milestone completion criteria and potential "client ghosting" that are typical of the gig economy today [1], [6]. Analytical calculations show that, compared to tiered commissions, Layer-2 solutions for off-chain processing of such transactions significantly decrease operational costs of the system [3].

B. Final Takeaway on Security and Compliance

It has been determined that the security of the described approach lies in immutability and auditing of the code of a smart contract [10]. Using the described approach based on Securify principles and classifying the contract types into "greedy", "prodigal", and "suicidal", it is possible to implement a trustless model in which capital of freelancers and their customers will be protected from any technical attacks [11], [12]. However, its implementation within the Indian legal environment will depend on the progress in recognition of smart contracts and taxation of digital assets as stated in the national policy document [14], [15].

C. Directions for Future Work

Despite the completeness of this analysis, there still exist some directions for further research in order to solve certain problems of decentralized freelance escrow [13]:

1. **Decentralized Reputation System:** Future works must consider creating an on-chain, soulbound token-based reputation system of freelancers working in different protocols without compromising on personal privacy [1], [13].
2. **Privacy-Preserving Verification of Milestones:** Further analysis needs to focus on the use of zero-knowledge proofs for verifying milestones without making the underlying proprietary source code known to the oracle nodes [7], [9].
3. **Integration of Regulatory Aspects:** Research in this area includes creation of "Tax-as-a-Service" automated functions in order to integrate the solution within the requirements of Indian GST/TDS for freelancer incomes [14], [15].

4. **Cross-Chain Escrow Mobility:** Future developments include studying the possibility of implementing cross-chain messaging mechanisms to enable escrow functionality across chains [6], [13].

XIII. REFERENCES

- [1] A. Singh, S. Kumar, R. Kumar, and S. Shenai, "DLancer: A decentralised freelancing platform," SCITEPRESS, 2025.
- [2] "Decentralized freelancing platforms using Ethereum," International Journal of Innovative Research in Technology, 2025.
- [3] "Decentralized gig economy," International Research Journal of Modernization in Engineering Technology and Science, 2026.
- [4] A. Asgaonkar and B. Krishnamachari, "Solving the buyer and seller's dilemma: A dual-deposit escrow smart contract for digital goods," in Proc. IEEE ICBC, 2019.
- [5] S. Zola, K. Arora, and A. Devipriya, "A decentralized escrow protocol based on blockchain," AIP Conf. Proc., 2024.
- [6] S. Ellis, A. Juels, and S. Nazarov, "Chainlink 2.0: Next steps in the evolution of decentralized oracle networks," Chainlink Labs, 2021.
- [7] A. Pasdar, Y. C. Lee, and Z. Dong, "Connect API with blockchain: A survey on blockchain oracle implementation," ACM Comput. Surv., 2023.
- [8] C. Duley, L. Gambacorta, R. Garratt, and P. Koo Wilkens, "The oracle problem and the future of DeFi," BIS Bulletin no. 76, 2023.
- [9] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," IEEE Access, vol. 4, 2016.
- [10] L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in Proc. ACM CCS, 2016.
- [11] P. Tsankov, A. Dan, D. D. Cohen, A. Gervais, F. Buenzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in Proc. ACM CCS, 2018.
- [12] I. Nikolić, A. Kolluri, I. Sergey, P. Saxena, and A. Hobor, "Finding the greedy, prodigal, and suicidal contracts at scale," in Proc. ACSAC, 2018.
- [13] R. Singh, "A systematic literature review on blockchain-based smart contracts: Platforms, applications, and challenges," ACM Comput. Surv., 2025.
- [14] K. Werbach and N. Cornell, "Contracts ex machina," Duke Law J., vol. 67, 2017.
- [15] NITI Aayog, India's Booming Gig and Platform Economy: Perspectives and Recommendations on the Future of Work. New Delhi, India, 2022.