Modeling and implementation of a secure freelance platform based on Ethereum smart contracts

Svitlana Popereshnyak[1,*,†], Maksym Bielikov[1,†] and Anton Bur[1,†]

[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prospect Beresteiskyi, Kyiv, 03056, Ukraine

Abstract

The rapid growth of the freelance market and the adoption of blockchain technologies have created new opportunities for decentralized and secure financial management. Traditional freelance platforms often suffer from high transaction fees, lack of transparency, and centralized control, leading to increased risks of fraud and contractual disputes. This paper presents the modeling and implementation of a secure freelance platform that integrates Ethereum smart contracts to automate financial transactions and improve trust between users. The study analyzes existing centralized and decentralized freelance platforms, formulates functional and non-functional requirements, and proposes a microservices-based architecture for the web application. Smart contracts were developed using Solidity to ensure automated, tamper-proof payment execution. Security, performance, and scalability tests were conducted to validate the system's robustness. A novel mathematical model for fraud risk assessment in smart contract-based transactions was introduced, taking into account users' financial ratings and behavioral patterns. Additionally, a decentralized arbitration mechanism using DAO (Decentralized Autonomous Organization) principles was implemented to resolve disputes fairly and transparently. The results demonstrate that integrating blockchain technologies into freelance platforms significantly enhances transaction security, reduces operational costs, and mitigates fraud risks, offering practical applications for startups, IT companies, and freelance marketplaces.

Keywords

Blockchain, Freelance platform, Ethereum smart contracts, Decentralized arbitration, Fraud risk assessment, DAO (Decentralized Autonomous Organization), Secure financial transactions, Internet of Everything, Microservices architecture, Solidity development, Blockchain-based marketplaces

CEUR-WS.org/Vol-4004/paper12.pdf

1. Introduction

The rapid development of blockchain technologies and the active growth of the freelance platform market open up new prospects for their integration into software. In modern conditions, freelance is becoming an increasingly popular format of work, covering numerous professional areas, including design, programming, marketing and others. However, traditional freelance platforms face problems related to the transparency of transactions and trust between the parties. This makes it urgent to develop solutions based on blockchain technologies that can eliminate these shortcomings. The most popular type of freelance exchanges are web applications, as they provide access from any device with a browser installed.

The goal of the work is to create an effective, transparent and secure platform for freelance, which uses the advantages of blockchain technologies to improve financial interactions between users.

The result of this work is the development of a freelance platform with payment via the Ethereum blockchain, which will ensure a high level of security and transparency of financial transactions. It

MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine

* Corresponding author.

† These authors contributed equally.

spopereshnyak@gmail.com (S. Popereshnyak); bielikov.maksym@lll.kpi.ua (M. Bielikov); bur.anton@lll.kpi.ua (A. Bur)

0000-0002-0531-9809 (S. Popereshnyak);0009-0004-1204-0349 (M. Bielikov); 0009-0004-9320-7445 (A. Bur)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

is expected that it will be in demand among freelancers, service customers and companies looking for performers for various projects.

Thus, the relevance of the development is determined by the need to implement the latest technologies to solve current problems in the freelance sector, which, in turn, will contribute to increasing trust between market participants.

2. Domain analysis

Freelancing, or working on-demand without permanent employment, is one of the fastest growing forms of employment in the world today. According to Demandsage 2024 statistics [1], there are approximately 1.57 billion freelancers in the world out of a total workforce of 3.38 billion. The main advantages of freelancing are flexible schedules, the ability to work on a variety of projects, and reduced operating costs for the employer. However, platform users currently face a number of problems that make it difficult for them to interact effectively. In particular, traditional freelance platforms are often characterized by:

- high commissions for intermediary services;
- lack of transparency in the terms of order fulfillment;
- risks of fraud and delays in payment;
- limited access to the global market due to financial or legal barriers.

Blockchain allows you to overcome these problems. This is a decentralized data storage technology that provides transparency, immutability and automation of transaction execution using smart contracts. A smart contract is a program code that automatically executes the terms of a transaction between parties without the participation of intermediaries. The most common platform for developing smart contracts is Ethereum.

Developing a decentralized freelance platform with the integration of Ethereum smart contracts to ensure transparency, security and automation of financial transactions will allow:

- Improve the reliability of transactions by automatically executing the terms of smart contracts.
- Reduce financial costs by eliminating intermediaries and optimizing commission fees.
- Increase the level of trust through the use of blockchain, which makes it impossible to forge transactions.
- Optimize risk management using a mathematical model for assessing fraudulent transactions.
- Improve the convenience of interaction between customers and executors through an automated task execution control system.

3. Literature review

The continuous development of freelance marketplaces, alongside the rising demand for secure, efficient, and transparent digital interactions, has driven significant research efforts into blockchain integration, risk management, and intelligent system architectures. Blockchain has emerged as a leading technology to enhance trust and security in decentralized environments. Hatim et al. [2] proposed a blockchain-based Internet of Vehicles (BIoV) framework to ensure data integrity and transparency within smart cities, demonstrating the effectiveness of decentralized trust mechanisms — a concept equally critical for freelance platforms aiming to reduce fraud and enforce transparent transactions.

In the context of blockchain-driven economic models, Sukkrajang et al. [3] designed a trade distance and pricing system for electric vehicle charging stations, using blockchain to secure and

verify transactions. Their work underscores blockchain's potential to automate and safeguard financial operations — a fundamental requirement for freelance ecosystems handling user payments.

The integration of blockchain with machine learning for compliance and monitoring purposes has been studied by Shaik et al. [4], who applied intelligent algorithms to enhance regulatory compliance in blockchain-based supply chains. Their findings reveal that machine learning can significantly strengthen fraud detection and risk management capabilities, which is highly relevant for freelance platforms seeking to automate user evaluation and transaction security.

Concerns over data integrity in decentralized infrastructures were addressed by Ravishankar et al. [5], who developed a blockchain-backed database to protect against data tampering in cloud computing environments. This reinforces the potential advantages of integrating blockchain into freelance marketplaces to ensure reliable user data management and transaction histories.

Despite its strengths, blockchain technology also introduces new security vulnerabilities. Ismail and Reza [6] analyzed blockchain-specific risks in supply chains, emphasizing the necessity of comprehensive risk assessment and robust system design — key considerations in building a secure freelance service platform.

Hartmann et al. [7] explored the role of blockchain in decentralized finance and crowdfunding, revealing that blockchain significantly enhances trust and reduces intermediary costs, supporting its application for minimizing operational expenses on freelance platforms.

Applications of blockchain to critical infrastructure, such as rail transit systems, were demonstrated by Li et al. [8], showing how blockchain enables secure, real-time management. Their results offer insights for freelance platforms requiring real-time contract execution and service verification.

Further, Ribeiro and Barbosa [9] introduced a blockchain-specific risk analysis methodology, highlighting the importance of multi-factor risk evaluation when developing decentralized applications — directly informing the fraud prevention model proposed in this study.

Additional studies on secure cryptographic methods [10] and efficient cloud data processing strategies [11] outline the broader technological challenges involved, emphasizing the need for secure, scalable freelance management solutions.

While previous studies have highlighted the transformative potential of blockchain and intelligent systems in various domains, the specific application of AI-driven multi-factor risk assessment models in combination with blockchain-based smart contracts for freelance service management remains insufficiently explored.

The development presented in this study directly addresses these challenges by integrating Ethereum smart contracts to automate payments and improve the security and transparency of freelance interactions. The proposed solution aims to create a decentralized freelance platform that:

- Dynamically evaluates the risks of financial transactions based on user ratings, behavioral factors, and financial indicators using a newly proposed risk assessment model;
- Automates freelancer selection and task verification through secure smart contracts;
- Implements a decentralized arbitration system using DAO (Decentralized Autonomous Organizations) principles to resolve disputes fairly and efficiently;
- Optimizes commission costs through adaptive transaction timing mechanisms.

The research thus aims to design an effective, transparent, and secure freelance environment, leveraging the advantages of blockchain technologies for enhanced financial interaction between users. By applying object-oriented design methods, blockchain analysis, cryptographic techniques, and machine learning algorithms, the platform ensures scalable, secure, and efficient freelance service management. This work contributes to the advancement of intelligent decentralized marketplaces and offers practical applications for startups, IT companies, and freelance exchanges seeking to enhance their operational security and efficiency.

4. Description of business processes

As part of the development of a web application for a freelance platform with payment via the Ethereum blockchain, key business processes were identified and described. They cover the main stages of user interaction with the system, such as registration, authorization, and order fulfillment. Let us consider in more detail the order fulfillment process, for which a BPMN model was built (Fig. 1.)

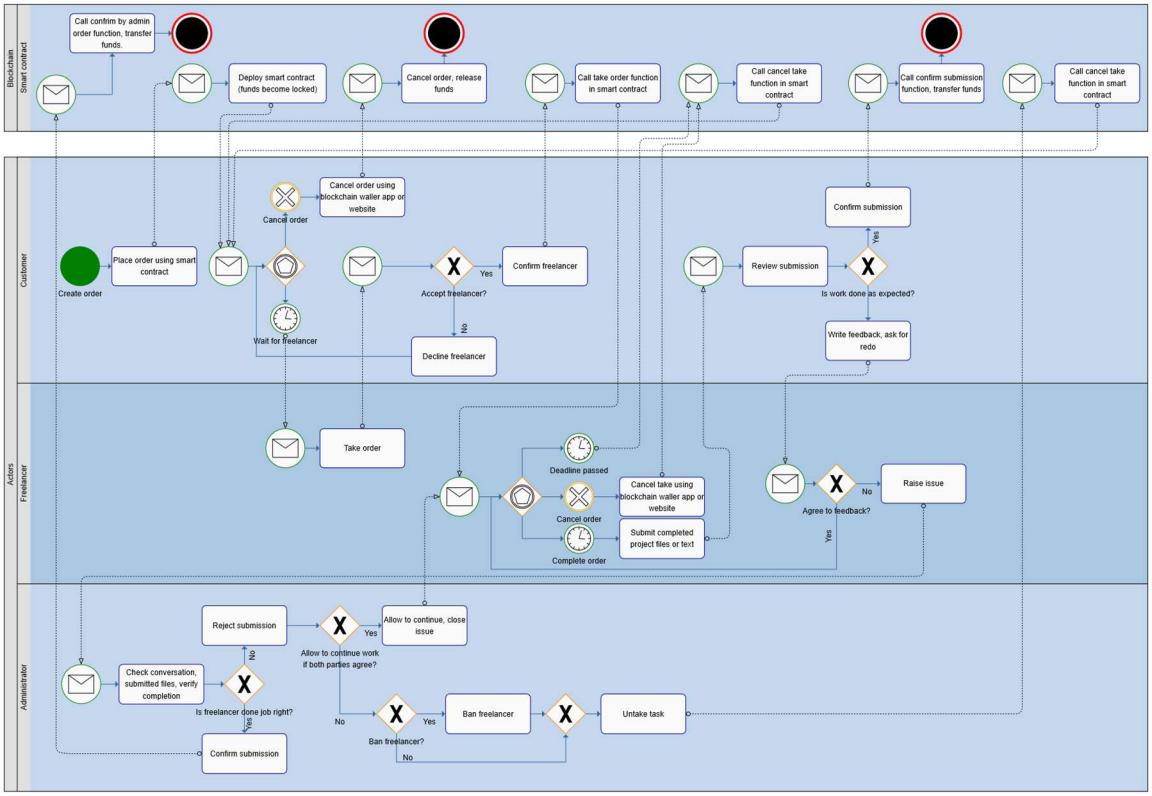


Figure 1: Order fulfillment business process

Description of the order execution sequence (Figure 1):

- the customer creates a task on the platform website, a smart contract is generated to the address of his wallet, which blocks the amount of money he has set (the budget for execution);
- the customer can cancel the order. This can be done both on the website of the web application and by calling the cancellation function independently on the blockchain via a crypto wallet. In this case, the money will be unlocked and the smart contract will be considered completed;
- the customer will wait until one of the freelancers takes the order on our platform. And when this happens, he will receive a corresponding message. The customer must either reject or accept the freelancer who wants to perform the task. As soon as the performer has been selected, a record is created in the blockchain. The security of the election is guaranteed, because the “take order” function can only be called from the wallet that was used to initiate the smart contract;
- the freelancer, in turn, can refuse to perform the task by calling the “cancel order” function through a wallet connected to the platform, but this will worsen his rating. Then, the customer will be able to wait and choose another freelancer, or cancel his task. Automatic cancellation will work when the deadline set by the customer is completed;

- the freelancer must attach the files of the completed task. The customer will check them and, if they agree, the money will be transferred to the performer. If a conflict arises, the platform administrator gets involved, who must check the chat history and decide whether the freelancer has completed the task assigned to him. If so, the money is sent to the performer and the smart contract is considered completed. If not, the administrator communicates with the parties to the conflict about whether to allow the current freelancer to make changes to the task and continue working. In case of agreement, the performer must attach new files to the task in time before the deadline or refuse to perform it;
- if the parties to the conflict do not reach an agreement, the administrator decides whether to allow the freelancer to take new orders on the platform and has the right to block him. The task performer is canceled from the smart contract. The customer can wait for a new freelancer. However, if the task is no longer relevant, he can cancel it. The money will be unblocked and the smart contract will be considered completed.

5. Mathematical model for assessing the risk of fraudulent transactions in smart contracts

5.1. Statement of the problem

Decentralized freelance platforms that use Ethereum smart contracts are at risk of fraud from both contractors and clients. Attackers can:

- Create fake accounts in order to receive an advance payment without completing the task.
- Use deliberately dishonest work evaluation mechanisms.
- Conduct transactions between their own accounts to artificially increase the rating.
- - Resort to manipulations with refunds due to vulnerabilities in smart contracts.

To minimize these risks, we have developed a mathematical model that allows us to estimate the likelihood of fraud in real time by analyzing the behavioral and financial parameters of users.

5.2. Model input parameters and fraud risk formalization

The following parameters are used to assess the risk of a fraudulent transaction:

1. User rating R ($0 \le R \le 1$) is the average score given by customers.
2. Number of successfully executed contracts N_s is an indicator of the contractor's experience.
3. The number of unfulfilled contracts N_f is the frequency of contract violations.
4. Account lifetime S (in days) is the duration of activity on the platform.
5. Average task completion time T_d (in hours) - compared to the average for the platform.
6. Cryptocurrency wallet balance B (in ETH) - the financial stability of the performer.

We also enter thresholds:

- $T_{d_{mean}}$ – the average time for completing tasks on the platform.
- $T_{d_{min}}, T_{d_{max}}$ – are the limits of normal contract execution.

Probability of fraud (P_{fraud}) is defined as the sum of weighted risk functions:

$$P_{fraud} = w_1 \cdot f(R) + w_2 \cdot f(N_s, N_f) + w_3 \cdot f(T_d) + w_4 \cdot f(B) + w_5 \cdot f(S). \quad (1)$$

where w_i are weighting factors that determine the importance of the respective factor.
Here are the formulas for the risk functions:

- Rating factor:

$$f(R) = 1 - R. \quad (2)$$

The lower the rating, the higher the risk of fraud.

- The factor of execution history:

$$f(N_s, N_f) = \frac{N_f}{N_s + N_f + 1}. \quad (3)$$

Add 1 to the denominator to avoid dividing by zero.

- The factor of contract execution time:

$$f(T_d) = \frac{T_{d_{mean}} - T_d}{T_{d_{max}} - T_{d_{min}}}. \quad (4)$$

If the performer completes a task much faster or slower than normal, this may be an anomaly.

- Crypto wallet balance factor:

$$f(B) = e^{-\alpha B}, \quad (5)$$

where α is a parameter that determines the sensitivity to low balances.

- Account age factor:

$$f(S) = e^{-\beta S}, \quad (6)$$

where β is a coefficient that determines the risk of new accounts.

5.3. Thresholds and decision-making

Based on the obtained value of P_{fraud}, the system determines the risk level of the transaction:

- If $P_{fraud} \geq 0.7$is a high risk of fraud, the transaction is blocked or requires additional verification.
- If $0.4 \leq P_{fraud} < 0.7$ – moderate risk, additional verification of identity or deposit of funds is required.
- If $P_{fraud} < 0.4$ – low risk, the transaction is allowed without restrictions.

5.4. Rationale for choosing weighting factors

In the model for assessing the risk of fraudulent transactions in smart contracts, the probability of fraud is calculated using the formula (1), where w_1, w_2, w_3, w_4, w_5 are weighting factors that determine the impact of each factor on the overall risk level.

Basic principles for choosing weighting factors.

- Normalization of the sum of weights. To avoid distortion of the results, the sum of the weighting factors should be equal to 1: $w_1 + w_2 + w_3 + w_4 + w_5 = 1$.
- Consideration of the real impact of factors.

Not all factors have the same impact on fraud. For example:

- Low rating and a large number of unfulfilled contracts are key indicators.

- The wallet balance affects the risk, but less significantly than the history of contract execution.
- Account age also matters, but its influence decreases over time.

Let's consider the main approach to choosing the values of weighting coefficients (Table 1)

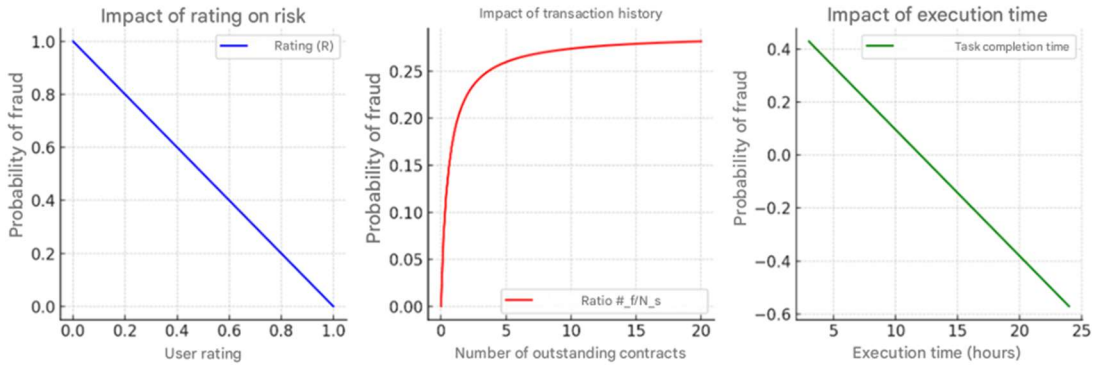
Table 1
Initial weighting coefficients based on expert analysis

Factor	Impact on fraud	Initial weight value
User rating w_1	High impact	0,30
Performance history w_2	High impact	0,25
Execution time w_3	Medium impact	0,15
Wallet balance w_4	Low impact	0,15
Account age w_5	Medium impact	0,15

This distribution of weights ensures that rating and contract history have the greatest impact, while less important factors (balance, account age) receive lower coefficients.

5.5. Risk factor analysis and model comparison

Let us consider the impact of various factors on the probability of fraud in smart contracts (Fig. 2, Fig. 3.):

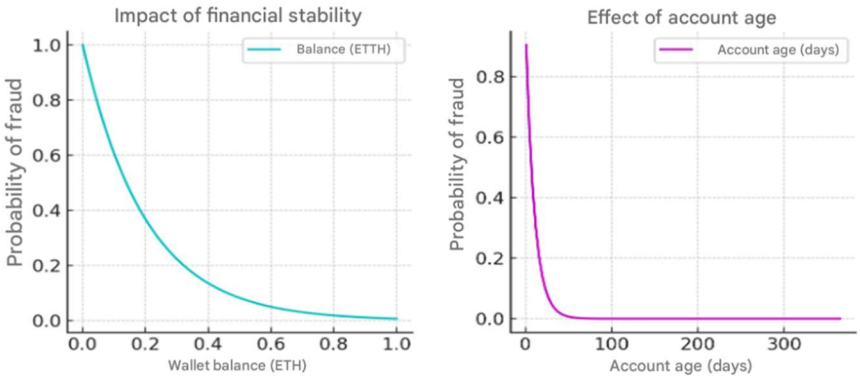


a) b) c)

Figure 2: Influence of factors on the probability: a) User rating; b) transaction history; c) execution time.

In Fig. 2. a) demonstrates the dependence of user rating on the probability of fraud - the higher the rating, the lower the probability of fraud. Fig. 2. b) demonstrates the dependence of transaction history on the probability of fraud - the more unfulfilled contracts, the higher the risk. Fig. 2. c) demonstrates the impact of execution time on the likelihood of fraud - abnormally fast or too slow execution of tasks may indicate fraud.

In Fig. 3. a) demonstrates the dependence of the wallet balance on the probability of fraud - a low balance increases the risk, as fraudsters rarely keep large amounts.



a) b)

Figure 3: Influence of factors on the probability: a) wallet balance; b) account age (days).

In Fig. 3. b) demonstrates the dependence of the age of the transaction account on the probability of fraud - new accounts have a higher risk of fraud, which gradually decreases with increasing time of existence.

Let's look at how the probability of fraud in the two models (traditional and proposed) changes depending on the user's rating (Figure 4).

Red dots (traditional model) — a simple linear approach that determines the risk of fraud based solely on the user's rating. Blue dots (proposed model) — takes into account additional factors, which allows for a more accurate risk assessment.

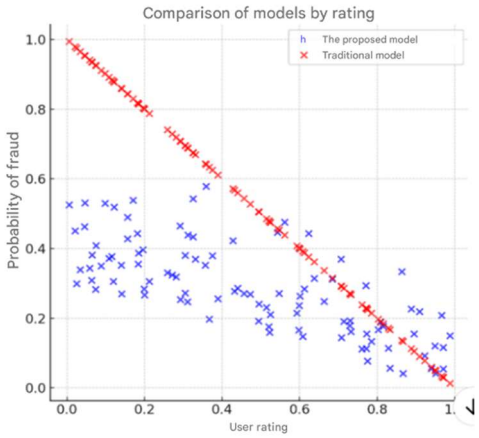


Figure 4: Comparison of models by rating

As you can see, the traditional model ignores users with low balances and new accounts, which can lead to fraud. The proposed model takes into account additional parameters, reducing false positives and false negatives.

6. Software architecture

The software architecture is based on a microservice approach using the principles of Domain-Driven Design. The main architectural pattern is Layered Architecture. This allows for scalability, flexibility, and ease of system maintenance.

Consider the third level of the C4 diagram (Figure 5), which describes the internal components of containers. Each microservice has controllers that accept requests and interact with services. The GRPC protocol is used to interact between microservices, the JSON/HTTPS protocol is used for external interaction with the API, and SMTP is used to send emails.

For payment via the Ethereum blockchain, the integration is performed using the Web3.js library, which allows you to interact with smart contracts. Smart contracts provide payment automation and guarantee transparency and data integrity.

An external SMTP mail service is used to send emails (registration, password recovery, notifications). Integration security is provided through the OAuth2 authentication mechanism.

Requests for text generation, as well as for validation of created orders, use the third-party API of the Groq service, which provides access to the LLM model. The connection is made via the HTTPS protocol.

The choice of database technologies is also based on the requirements for system functionality and reliability. PostgreSQL is used for the authorization microservice, as this DBMS ensures high integrity and security of user data, especially when storing sensitive information such as passwords and tokens. Redis is used for caching and storing temporary data, such as active session tokens or verification codes. This technology allows you to quickly retrieve data by storing it in RAM.

Particular attention is paid to system security, especially in terms of authorization and protection of user data. The use of two-factor authentication through emails and TOTP, as well as captcha protection, is necessary to prevent unauthorized access. To ensure the secure storage of sensitive information, encryption technologies are used at the database level, as well as the use of the bcrypt library for password hashing.

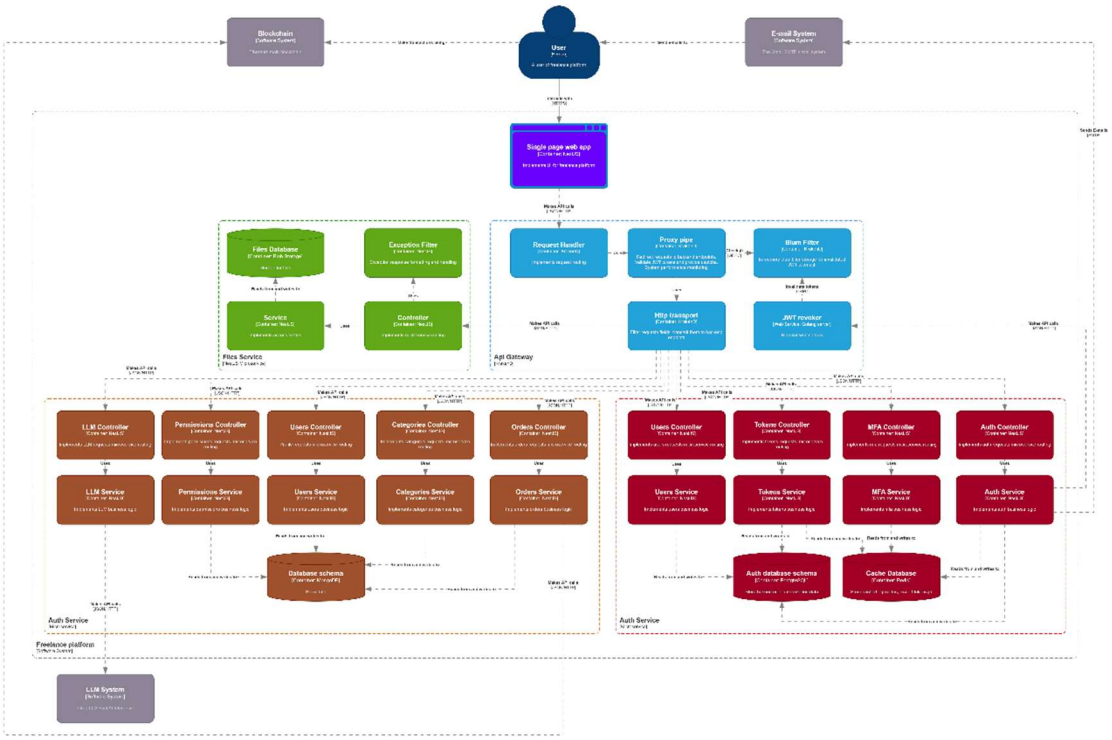


Figure 5: The third level of diagram C4

The secure gRPC protocol is used to transfer data between microservices because it has many advantages. Firstly, gRPC allows for high data exchange speeds through the use of the HTTP/2 protocol, which supports multithreading and efficient connection management. This is especially important in a microservice architecture, where each microservice must connect to others to perform complex operations. In addition, gRPC uses serialization of the Buffers protocol, which reduces the amount of data transferred and increases the efficiency of request processing. This reduces the delay time between requests and responses, which improves overall system performance.

To interact with the API, users use the HTTPS protocol, which provides a high level of security when transferring data between clients and the server. HTTPS uses the TLS protocol to encrypt data, which guarantees protection against information interception and man-in-the-middle attacks.

HTTPS also provides server authentication, which helps to avoid request forgery and promotes trust in the API by users. In addition, HTTPS is the standard for interacting through web browsers, which ensures compatibility with a wide range of browsers and ensures that the connection remains secure when the network changes or users switch to new devices.

7. Data security analysis

Data security analysis in the Web application of a freelance platform paid through the Ethereum blockchain takes into account potential risks and provides protection against common threats through the integration of modern security technologies.

Among the possible attacks that can be directed at the system are SQL injections, cross-site scripting (XSS) attacks, brute force attacks (DDoS), phishing, exploitation of vulnerabilities in third-party software, attacks on blockchain wallets, and data interception through Man-in-the-Middle.

To protect against SQL injections, the system uses parameterized queries and ORM tools that minimize the risk of executing malicious commands in the database. XSS vulnerabilities are neutralized by thoroughly checking and cleaning incoming data, as well as encrypting confidential information before it is displayed on the frontend.

To combat brute force attacks, the registration and authorization stages use a limit on the number of login attempts, captchas, and two-factor authentication. Phishing risks are minimized by encrypting data via SSL TLS and creating an interface that includes direct warnings to users about possible threats. Up-to-date software updates reduce the risk of exploiting vulnerabilities in third-party libraries and frameworks.

To protect the integration with blockchain wallets, we have implemented algorithms for signing transactions, encrypting private keys, and interacting only through official APIs. Protection against Man-in-the-Middle attacks is provided through the use of HTTPS.

The integrity and confidentiality of data is maintained by encrypting the storage of confidential information (passwords, tokens) using the AES-256 and bcrypt algorithms.

8. Conclusions

The aim of the development was to increase the reliability and security of data processing using Ethereum blockchain technologies. As a result, we created a web application that provides efficient and secure interaction between freelancers and customers, automation of many routine processes, including task verification, using the LLM model. All functional requirements have been successfully implemented, and the system provides a high level of security, efficiency in transaction processing, and support for transparency of financial transactions. The development meets all the requirements and is competitive in the market of freelance platforms working with blockchain technologies. Thus, the goal has been achieved.

The study analyzed modern centralized and decentralized freelance platforms and identified their key disadvantages, such as high fees, centralized management of financial flows, lack of transparency of transactions, and the risk of fraud. To solve these problems, a web-based freelance platform application with the integration of Ethereum smart contracts is proposed, which provides payment automation, protection against manipulation, and secure interaction between customers and contractors.

The company has developed a mathematical model for assessing the risk of fraudulent transactions that takes into account user ratings, task history, crypto wallet balance, and other factors. This model allows us to identify potentially fraudulent accounts and transactions, reducing the risks for platform participants.

An adaptive fee management mechanism has been implemented to optimize costs when interacting with the Ethereum blockchain, increasing the efficiency of transactions. A system of automated arbitration involving decentralized autonomous organizations (DAOs) was implemented to resolve disputes between customers and contractors.

The results obtained and the developed software are the basis for creating a competitive freelance platform that can be used in the market for secure interaction between freelancers and customers. The results of the study can be used to create secure decentralized platforms in the field of freelancing, e-commerce, and financial technologies.

Declaration on Generative AI

During the preparation of this work, the authors used AI program Chat GPT 4.0 for correction of text grammar. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Freelance statistics 2024 – number of freelancers & industry size. *DemandSage*. URL: https://www.demandsage.com/freelance-statistics/ .

[2] S. M. Hatim, S. J. Elias, R. M. Ali, J. Jasmis, A. A. Aziz and S. Mansor, "Blockchain-based Internet of Vehicles (BIoV): An Approach Towards Smart Cities Development," 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 2020, pp. 1-4, doi: 10.1109/ICRAIE51050.2020.9358355.

[3] K. Sukkrajang, R. Duangsoithong and K. Chalermyannont, "Trade Distance and Price Model for Electric Vehicle Charging using Blockchain-based Technology," 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Mai, Thailand, 2021, pp. 964-967, doi: 10.1109/ECTI-CON51831.2021.9454741.

[4] A. S. Shaik, M. Mahima, J. Sravanthi, H. M. Ali, R. Agarwal and D. G. V, "Machine Learning Applications for Enhancing Regulatory Compliance in Blockchain-Based Supply Chains," 2024 International Conference on IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2024, pp. 1239-1243, doi: 10.1109/ICICAT62666. 2024.10923070.

[5] B. Ravishankar, P. Kulkarni and M. V. Vishnudas, "Notice of Removal: Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments," 2020 International Conference on Mainstreaming Block Chain Implementation (ICOMBI), Bengaluru, India, 2020, pp. 1-4, doi: 10.23919/ICOMBI48604.2020.9203500.

[6] S. Ismail and H. Reza, "Security Challenges of Blockchain-Based Supply Chain Systems," *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, NY, USA, 2022, pp. 1-6, doi: 10.1109/UEMCON54665. 2022.9965682.

[7] F. Hartmann, G. Grottolo, X. Wang and M. I. Lunesu, "Alternative Fundraising: Success Factors for Blockchain-Based vs. Conventional Crowdfunding," 2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Hangzhou, China, 2019, pp. 38-43, doi: 10.1109/IWBOSE.2019.8666515.

[8] H. Li et al., "Blockchain-Based Data Management and Control System in Rail Transit Security Scenario," 2024 IEEE 11th International Conference on Cyber Security and Cloud Computing (CSCloud), Shanghai, China, 2024, pp. 19-23, doi: 10.1109/CSCloud 62866.2024.00011.

[9] S. L. Ribeiro and I. A. de Paiva Barbosa, "Risk Analysis Methodology to Blockchain-based Solutions," *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, Paris, France, 2020, pp. 59-60, doi: 10.1109/ BRAINS49436.2020.9223309.

[10] S. Popereshnyak, Y. Novikov, Y. Zhdanova Cryptographic system security approaches by monitoring the random numbers generation. (2024) CEUR Workshop Proceedings, 3826, pp. 301-309. doi:

[11] M. Reshetniak, S. Popereshnyak Method for accessing and processing multimedia content in a cloud environment. (2019) 2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019 — Proceedings, art. no. 9061463, pp. 71-76. doi: 10.1109/PICST47496. 2019.9061463