FMDB *Transactions on Sustainable Finance and Data Science*

Blockchain Based Crowdfunding Platform: A Secure and Transparent Approach to Decentralized Fundraising

K. Karthikeyen[1,*], S. Irfan Kasim[2], B. M. Hareez[3], R. Regin[4], M. Sebastian[5], Jedsadakorn Nararattananukul[6]

[1,2,3,4]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
[5]Department of Management Studies, Dhaanish School of Management, Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India.
[6]Faculty of Business Administration, Ramkhamhaeng University, Bang Kapi, Bangkok, Thailand.
kk6505@srmist.edu.in[1], is5154@srmist.edu.in[2], hb2603@srmist.edu.in[3], regin12006@yahoo.co.in[4], maliyakal19@gmail.com[5], uratchayaphonn56@rumail.ru.ac.th[6]

Abstract: The problem is that many current crowdfunding frameworks overly rely on centralized management and lack transparency regarding how funds flow. Filling that vacuum, researchers introduce CrowdChain as an innovative funding framework implemented solely on the Ethereum blockchain. Rather than a centralized server, the system will leverage Smart Contracts written in the Solidity programming language for transactions, voting, and status tracking, which cannot be tampered with later. Instead of loading large files, only their URLs will be stored to avoid additional transaction costs. The primary mechanism will focus on developing a new procedure for approving milestones. In this procedure, the funds will be securely held in online vaults until more than half of the participants approve their release. Researchers will ensure that CrowdChain integrates seamlessly with MetaMask by using Next.js and Ethers.js for the frontend. This will make it easier for users to participate on the platform. On-chain governance, when combined with off-chain data, has been shown to dramatically reduce operational expenses without sacrificing security or traceability, according to the tests conducted.

Keywords: Blockchain and Crowdfunding; Smart Contracts; Decentralized Finance; Crowd-Funding Sites; Security Risks; Security Token Offerings; Initial Exchange Offerings; InterPlanetary File System.

Received on: 12/08/2025, **Revised on:** 05/10/2025, **Accepted on:** 10/12/2025, **Published on:** 09/06/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSFDS

DOI: https://doi.org/10.69888/FTSFD.2026.000697

Cite as: K. Karthikeyen, S. I. Kasim, B. M. Hareez, R. Regin, M. Sebastian, and J. Nararattananukul, "Blockchain Based Crowdfunding Platform: A Secure and Transparent Approach to Decentralized Fundraising," *FMDB Transactions on Sustainable Finance and Data Science*, vol. 1, no. 2, pp. 93–107, 2026.

1. Introduction

1.1. Background and Context

*Corresponding author.

Something new emerged in money matters when people started using websites to collect cash directly from others around the world. Instead of knocking on bank doors or begging investors, folks now show off their plans online - getting tiny bits of help from lots of different people [1]. It opened paths for dreamers and builders who rigid systems had shut out, built long ago. A shift happened quietly at first, then grew stronger after sites like Kickstarter appeared near the end of the 2000s. Suddenly, anyone could back inventions, films, tools, and even community efforts by sending a bit of money toward something they liked. Distance stopped mattering; location no longer blocked chances [2]. One idea at a time, trust moved from institutions into shared belief among strangers, clicking yes and fuelled by small contributions piling up quickly, the worldwide crowdfunding scene now pulls in billions of dollars every year. Ideas from nearly any field can find backing when people decide they're worth a shot. Growth like this hints at something deeper - trust in new ways to fund what matters. Still, most old-style crowdfunding sites run like hubs with a single command center. Whoever runs the platform gets final say on which projects go live, handles money flow, sends out payouts, plus settles conflicts when they arise. Sure, having a single main operator can make things run more smoothly. Yet this setup comes with downsides - making it harder for everyone involved to see what is really happening behind the scenes. Backers rarely get clear updates on how their money is spent once a project wraps up [3].

Creators, too, must wait, stuck following rules set by the site and its banking partners to reach the cash raised. Key influencers of control generally derive handsome profits from charges, leaving the creators with little money. The fact that the system depends heavily on conventional banking systems and traditional transaction systems means that some people cannot participate, especially in nations with strict financial regulations or inadequate banking facilities. When disturbances occur, such as interruptions, fraudulent activities, or unexpected financial regulations, users feel their impact immediately [4]. A distributed ledger system shares information among several parties rather than consolidating it in a single place. Instead of one hub holding all records, copies live on many machines across the chain. It holds information in a way that makes changes tough to hide, so things stay open and safe. With Ethereum came a new room to grow - code could now run on the chain itself. Running on their own, these coded deals follow rules baked right in; no person is needed to push them along. They let apps spread across nodes handle tricky jobs like moving money, counting votes, or giving back funds when due, all without losing clarity or protection [5]. When chains of blocks join up with those running agreements in crowdfunding, pools of support form without a central hand steering, showing every dollar's path clearly and sending payments only at set moments. Trust grows more easily, effort drops - such setups might reshape how groups gather resources.

1.2. Problem Statement

Even though crowdfunding sites let more people reach funding, hidden issues still block smooth results. Not every system runs fairly; most rely on one central team holding the keys. That single group decides who gets approved, when money moves, and how accounts are handled, sometimes changing outcomes without warning. When choices feel unclear, those launching projects rarely understand why a go-ahead or a denial occurred. Decisions during conflicts tend to back the site first, leaving users guessing where they stand [6]. Here's another problem - running a crowdfunding campaign often means spending more than expected. Most platforms take a cut, usually 5 to 10 percent of what you bring in. On top of that, there are extra charges just for handling payments, sometimes tacking on even more cost. Because of these layers, people who run projects might end up keeping far less than they hoped. When totals get big, those taken-out chunks can amount to serious money missing down the line [7].

Even when money moves to project starters, people who gave it usually cannot see where it goes next. Because spending isn't clearly shown, cash might be used poorly, tasks could stall, or perks may never arrive. Reports show that many crowdfunded efforts struggle to deliver on time, leaving backers frustrated and less likely to trust these platforms again. Participation capabilities vary by location and income group. Some systems use financial transfer techniques that cannot be accessed or are prohibited in some areas, hence preventing potential users from contributing to international projects. People living in impoverished nations face additional difficulties because international banking systems may not allow them access. Combining personal details and funds increases security risks because centralization makes these systems more prone to cyberattacks, potentially leading to leaks of personal details and financial records [8]. Processing manual refunds and settling disputes slows down operations, affecting their reputation for impartiality. Given these issues, decentralisation would be more efficient, offering better governance, lower costs, enhanced payment security, and greater accessibility.

1.3. Objectives of the Study

It is proposed to adopt a new crowdsourcing model based on the development of distributed networks that are not centrally managed, using blockchain technology and self-executing contracts. It is necessary to distinguish this idea from previous methods that rely on centralized crowdfunding systems for information management and decision-making. The key idea of the approach is to distribute the control of the entire process between the participants, who will also share responsibilities [9]. Increased transparency is another aspect since all transactions become visible in such networks. The problem of additional fees and delays in delivering funds to the creator is solved by automating transactions in accordance with predefined rules. The code is the basis for trust, which is another key point. Thus, this development may go beyond mere innovation and become something else. The main goal of the research is to develop a crowdfunding framework capable of managing processes such as campaign management, contribution tracking, and eliminating intermediary organizations. The mechanism aims to reduce costs and deliver funds to the creator once the necessary conditions are met. Another objective of this work is to increase the transparency of the fundraising process. All transactions and payments are recorded in a distributed ledger that allows users to track the movement of funds with confidence [10]. Data protection is crucial during this process, as encryption and autonomous contract execution can prevent fraudulent activity. If the campaign does not raise the required funds, the funds are automatically returned to investors. One of the design priorities is creating user-friendly interfaces suitable for even inexperienced users. Even though blockchains are considered efficient, they often become difficult for many users. The main focus of the paper is building a strong underlying structure while keeping the interface simple. In general, experiments are needed to test the effectiveness, operability, and accessibility of the developed system for potential users.

1.4. Significance of the Study

The contributions of this study are evident in the way blockchain technology has been applied to real problems associated with the crowdfunding arrangement. By using blockchain technology in a decentralized manner, without relying on large intermediaries, the framework ensures fairness and transparency throughout the funding process. If other developers and researchers wish to explore similar applications in their work, they now have an example of how such an application should work for learning purposes [11]. The current study highlights several interesting features of its implementation across organizational and coding aspects. Looking closely at user tests reveals where changes could make decentralized setups easier to join. As barriers drop, people who miss out on traditional banking might find new ways to access funds through online crowdfunding. With less overhead and stronger confidence in how things work, small creators and local projects across the globe gain ground. When tested, the model shows that blockchains can reshape how money is raised digitally - clear rules, safer transfers, leaner processes open doors for fresh ideas in finance without central control [12].

2. Literature Review

2.1. Overview of Traditional Crowdfunding Ecosystems

Now here's how it works: people pull money together online, one small piece at a time, building support from many different corners. Long ago, similar ideas circulated through subscriptions, but real shape came only when sites such as Indiegogo appeared in 2008, followed by Kickstarter a year later. Studies, especially a detailed 2014 work by Agrawal and team, split this method into four types: offering rewards, offering shares, lending cash, or simply asking for donations. When someone chooses the reward route, they hand over money in exchange for something physical, maybe an early version of what's being made [13]. That move doubles as both funding and proof that folks want the thing before factories start running. Giving money without expecting anything in return most often happens through donation crowdfunding. Yet when people invest cash hoping for profit, equity or loan setups come into play instead. What stops wider trust, though? Research keeps pointing to weak spots built into central systems. A major problem arises because commissions are deducted, sometimes as much as 10% of every dollar earned. This reduces the funds available for the paper itself. The second problem is the platform owners' lack of responsibility [14]. After the money moves to the maker, site oversight usually ends. No built-in system checks how cash gets used or ensures rewards arrive. That gap - called the principal-agent issue - lets creators hold more knowledge than supporters. Backers stay in the dark while those running projects know more. Some well-known collapses and scams on large sites have weakened confidence across crowdfunding. Because of this, researchers are now exploring designs that build responsibility without requiring a controlling group [15].

2.2. Blockchain Technology: The Trustless Paradigm

Bitcoin emerged in 2008 thanks to someone named Satoshi Nakamoto, tackling the challenge of verifying online money transfers. Instead of banks or intermediaries, this setup used something called blockchain - a shared record kept across many computers that can't be changed after entries are made. Its main trick? Letting people interact without needing to believe in one another [6]. That word "trust less" sounds odd at first, yet here it means nobody must rely on anyone else's honesty; the math behind the scenes holds things together. Later, Swan took this idea further, breaking down uses of blockchains into three waves: coins, smart contracts, and then broader real-world tools. When it comes to raising funds from crowds, the second wave matters most. Logic and deals now shift, not just money. Because everyone holds a copy, what gets written stays clear - no changes after the fact. Old crowd campaigns often hide details; this fixes that gap. Each pledge shows up openly, so anyone can check the balance right away [11]. Still, just showing where funds go won't fix misuse. Even when supporters watch every dollar move, that doesn't mean they can block bad choices. What helps here is a way to tie actions to clear conditions - something smart contracts offer by design.

2.3. Smart Contracts and Automated Governance

When Nick Szabo first talked about "smart contracts" back in 1994, it stayed just an idea - until Ethereum came along in 2014, built by Buterin [1]. The contract runs independently because its rules are directly coded into computers. Such a contract is registered in a distributed, decentralised ledger known as a blockchain. The fundraising process is also affected by the tool and the way people use it to interact with their supporters. If there is a breach of a traditional contract, legal action may be necessary, which would be both expensive and time-consuming. On the contrary, smart contracts implement themselves right away. This is because the contract rules are programmed in the software. According to Chowdhury et al. [16], such contracts run complex operations without requiring any human effort. Take, for example, crowdfunding. Here, the contract serves as an escrow service to collect funds. Money will only be disbursed when certain conditions are met. One agreement can disburse the collected funds to all parties if the set goals are not achieved before the end of the set deadline. By automating transactions, the need for intermediaries, such as custodians, is reduced, thereby lowering costs and fraud risk. Money moves gradually based on achievements that were made along the way [10]. The program holds the money until verification shows completion of goals. Money is released gradually after each milestone is reached. These pre-programmed rules reflect decisions made before. Power, therefore, resides not in one individual but in everyone, and the control follows trust [17].

2.4. Decentralized Fundraising Models

Tokens started popping up when blockchains were used for raising money. Shortly after that, exchanges started trading digital coins for cryptocurrencies like Ether or Bitcoin. Many of these tokens gave people the right to access features or to vote on future decisions. Between 2017 and 2018, significant sums were invested, far more than would typically be seen in startups. However, research found that trust issues persisted. The biggest problem was that many smart contracts lacked systems for payments and progress tracking [14]. People could raise a lot of money just from a white paper, and the money would go straight to the developers, who could use it as they wished. Without any real obligation to complete the project, things would go wrong, with exit scams becoming common; the participants called this practice a “rug pull.” In its exploratory approach, the industry examined both Security Token Offerings (STOs) and Initial Exchange Offerings (IEOs). The experiment failed in some respects: STOs attempted to convert actual assets, such as stocks, into digital tokens but faced strict regulatory constraints. It led to IEOs using reputable exchanges like Binance to validate the startups. Though IEOs were considered relatively safer, they reintroduced intermediation and cost structures like those of existing web-based services. The pattern seems to indicate a lack of one thing: a system that maintains openness during the initial token sale process while adding the verification process associated with regular startup funding.

2.5. Decentralized Autonomous Organizations

Tokens became the method of raising money via blockchain-based fundraising. Tokens in the earlier stage were known as Initial Coin Offerings. They exchanged their digital currencies, such as Ether and Bitcoin, for tokens. Some tokens granted permission to use a specific service in the future [5]. At the same time, tokens enabled voting on network-related issues. From 2017 to 2018, a large sum of money was invested very quickly compared to other technologies. The venture capital community watched valuations rise rapidly, but did not address accountability. These shortcomings were like those seen in traditional crowdfunding scenarios. At the time of deployment, smart contracts lacked the means to facilitate phased payments based on proven progress. Without these features, trust in each project declined considerably. Most projects received significant funding by simply providing an official statement, enabling immediate deposits into developers’ accounts and unrestricted control over the next steps. Since there were no regulations to oversee the allocation of resources, abuse was rampant [18]. Many projects ran for a short while before collapsing, with the founders leaving once they had secured the funds from the community [9]. This practice came to be known as “rug pulls.” The core problem can be summarised as follows. STOs originally aimed to institutionalize tangible assets through digitisation, but quickly became plagued by regulatory issues. In turn, IEOs outsourced the vetting process to major platforms like Binance, which were tasked with validating a startup's authenticity. Trust was restored through the endorsement of a familiar face. Yet these traditional gatekeepers returned, along with their inherent costs, filtering processes, and selection biases reminiscent of the restrictive nature of Web 2.0’s gated community model. The pattern persists because each solution introduces its own set of obstacles. The most important feature, however, is the lack of a crucial component. Namely, an approach that combines openness typical for early crowdfunding projects, transparency in its verification points, and the standards of experienced investors in a private setting [12].

3. Proposed Method and System Architecture

3.1. System Overview

The new crowdfunding technology, Crowd Chain, is built on the Ethereum blockchain. The architecture consists of three blocks that work in unison but remain conceptually separate. Each block has its own function. While split into parts, all these parts are interconnected to work smoothly. The architecture's strength lies in its decentralization. Smart contracts control financial transactions, while software protocols execute operational actions autonomously, meaning there is no central authority controlling everything. There is functional separation among the architecture's elements, although they depend on each other.

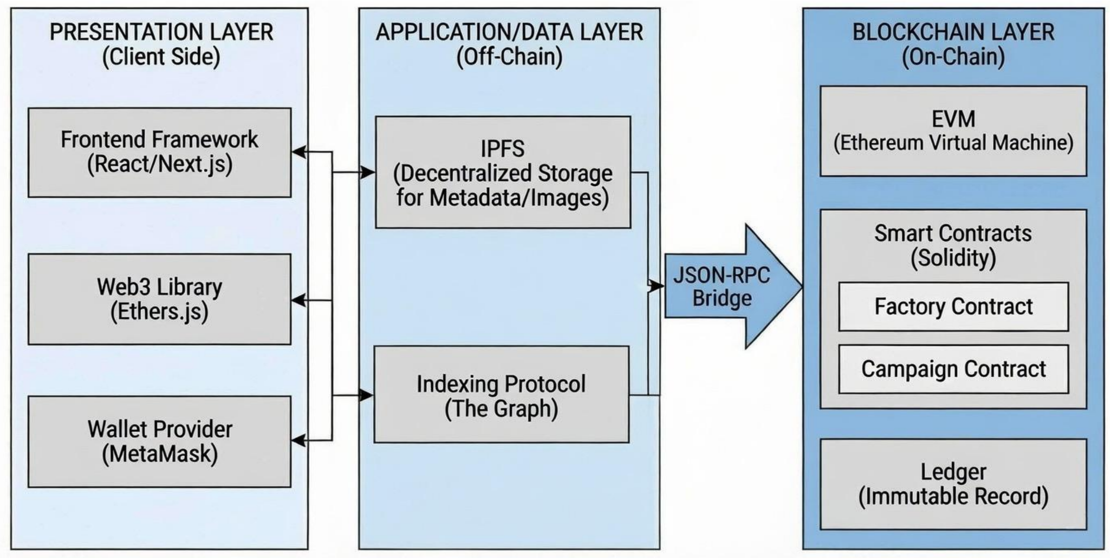

Figure 1: System architecture - three-layer design

The system's tripartition is shown in Figure 1. On top of the architecture sits the Presentation Layer, where all user interactions occur. It is built with Next.js, which runs on React. This framework ensures an immediate response to user actions, such as scrolling or clicking. The link between the presentation and the blockchain is Ethers.js, an application framework for communicating with the blockchain in web applications. Finally, MetaMask handles heavy computations indirectly, storing cryptographic keys and approving actions only if the user agrees. Storing large volumes of data, including images and lengthy campaign descriptions, on the main blockchain becomes more expensive. Instead of storing this data on the Ethereum blockchain, it is moved to the Interplanetary File System (IPFS), which specialises in handling large volumes of data. Due to the need for quick data access, indexing is handled via The Graph. As a result, if initial loading of statistical data is necessary, it is done immediately without making repeated calls to the Ethereum blockchain. The overall speed increases because requests are made using indexed references rather than live nodes. At the base level, the paper uses the Ethereum Virtual Machine (EVM). Smart contracts are written in Solidity and run inside this virtual machine. For external integration, a JSON-RPC bridge is used that allows communication between external software and on-chain logic.

3.2. System Design and Class Structure

Figure 2 below highlights the interlinking of objects using an OOP relationship. Firstly, the User class has two subclasses: Creator and Backer. The Creator subclass is responsible for calling the createCampaign method, while the Backer handles backing actions via methods specific to each backer action. In the middle is the Campaign object that holds properties such as ID, name, funding objective, amount raised thus far, deadline, creator, and status. Another object used here is CrowdFundingFactory, which creates unique Campaign Contract objects. Contracts handle on-chain activities, including depositing, withdrawing, and refunding backers. Interaction between components occurs via a website, where users can start or contribute to existing projects. Campaign creation, editing, and fundraising are done through the campaign management tool. Contributions occur via a transfer process that uses software code embedded in the chain to dictate all activities.

These include enforcing contract terms, monitoring deadlines, logging payments, transferring funds to paper owners once the target is achieved, and refunding when the objective is not reached. All activities are logged permanently, audited by the whole blockchain network, publicly viewable, and cannot be tampered with. This technology operates independently, without any intermediaries. Each user starts as an object of type User, which includes two attributes: Location and Available Balance. On top of this class are two primary subclasses based on the User class, but with different inheritance; the Creator subclass creates projects through campaigns. On the flip side, funding comes from the Backer. Thanks to this setup, shifting between roles feels natural - someone might create one project, yet support another. Switching like this fits how people get involved. Security and reliability are paramount. The system has CrowdFundingFactory, which is used for automated contract deployment. Contracts are not deployed directly; they are deployed by calling deployCampaignContract ().

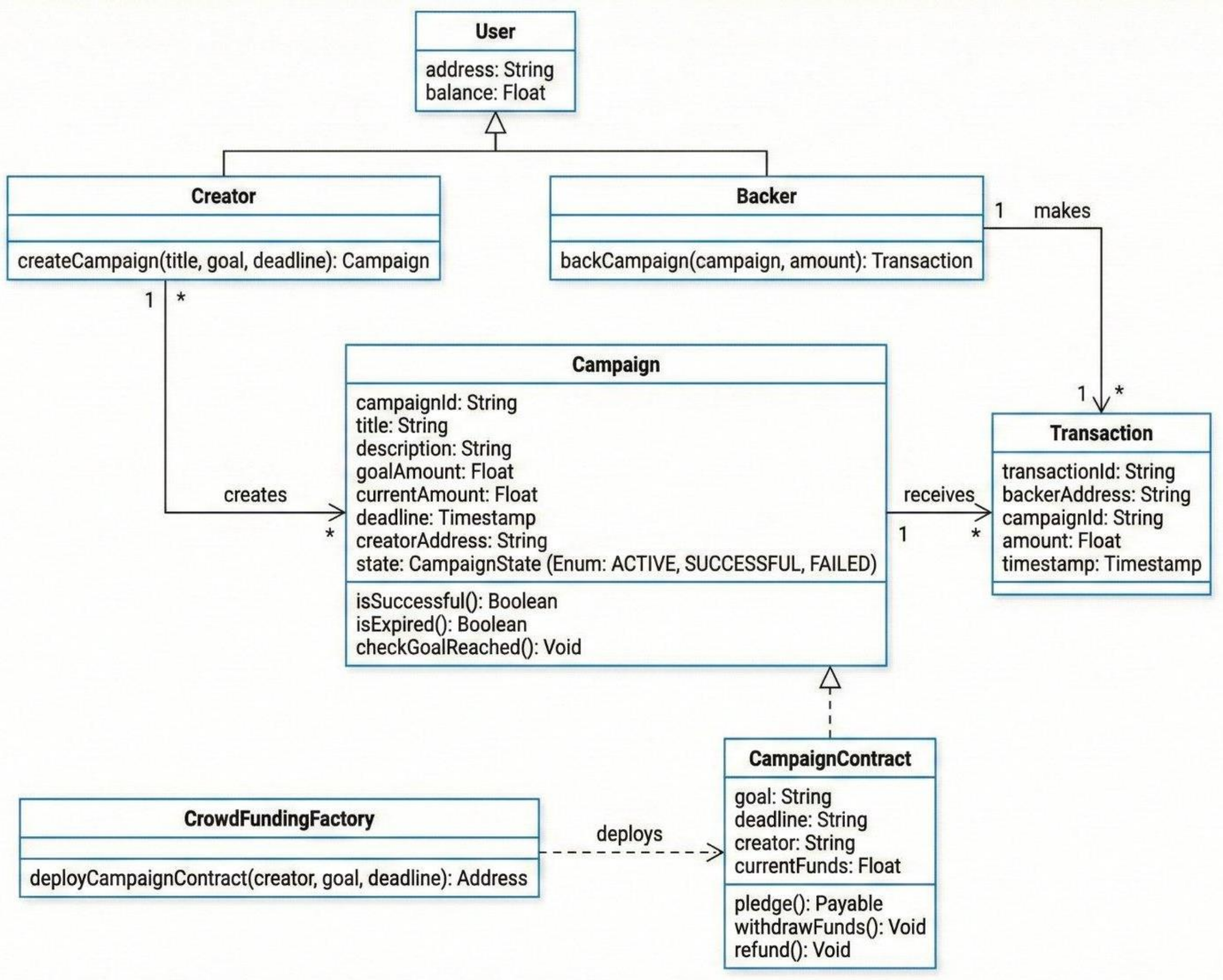

Figure 2: Functional block diagram - system object model

Each time it is called, a fresh instance of the Campaign Contract is generated from a reliable design blueprint. The same criteria are adhered to in every deployment. This way, any malicious changes are not incorporated in the process.

3.3. System Workflow

Firstly, the key specifics include the total amount needed, the total amount raised, the end date for this fundraising, and the status: ongoing, accomplished, or failed. Based on the criteria mentioned, it can be determined whether the target has been reached and whether the fundraising time limit has expired. Every pledge is recorded as a transaction between a supporter and the campaign they contributed to. These transactions ensure the actual transfer of funds, record the transaction time, and identify the sender's wallet. In blockchain, transactions serve as proof of one's contribution, thereby enabling participation in decision-making processes. To start using the proposed web application, a user needs to log in to their Web3 account using MetaMask, not via email. Thus, the system verifies one's digital identity as an Ethereum address. In the meantime, depending on the user's intention, they may either launch a new campaign or contribute to the ongoing one.

If one chooses Creator mode, they will be prompted to enter key campaign information, such as the paper name, total funding goal, and time limit. Data minimization will reduce the need to pay gas fees. Large files (images and text) are stored offline in the IPFS service and assigned a CID. Afterwards, the CID will be submitted to the CrowdFundingFactory contract, and a new Campaign Contract will be deployed. The main idea here is that the Ethereum network stores minimal data, and large files are located outside the blockchain. Consequently, it is cost-effective. From the supporters' perspective, the operation algorithm will proceed as follows. Users transfer their ETH directly to an active contract, and the system automatically registers the transfer. As soon as this process occurs, the supporter becomes an Approver, with voting power determined by the contribution size (Figure 3).

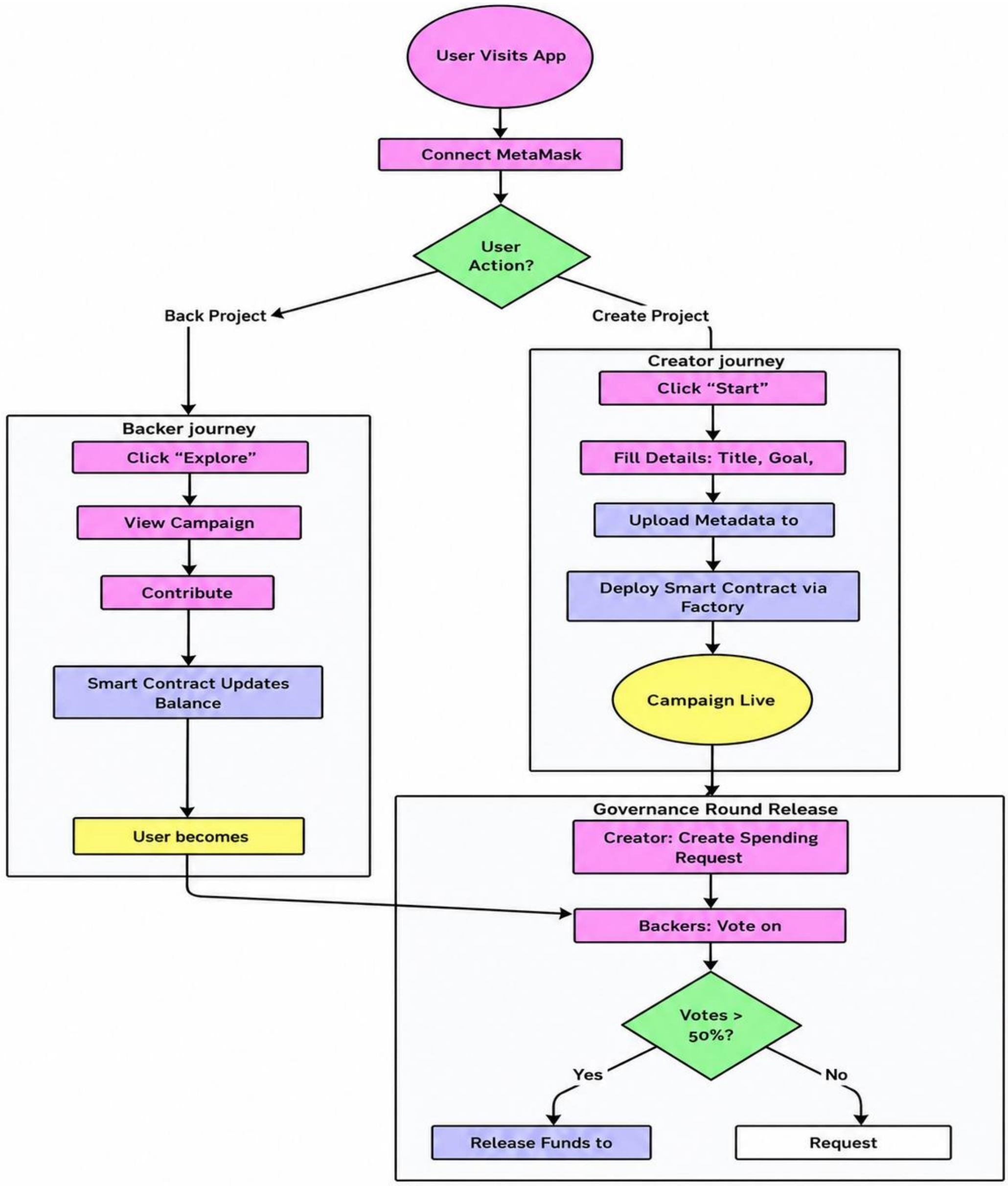

Figure 3: System workflow diagram

The main process is called the Fund Release Cycle. Referring to the provided diagram, the process involves several actions. Initially, one should submit a Spending Request from the proponents' side, for example, to purchase necessary campaign goods. Then users will vote, and once enough positive votes are received, the system releases the funds needed to pay suppliers.

3.4. Algorithmic Implementation

CrowdChain's functionality depends on three main elements that guarantee both openness and credibility. One of the major elements is the Contribution Algorithm, which analyses contributions to ensure they exceed a minimum level. If this requirement is met, the person contributing funds is marked as an approver and added to the Approvers group. In such a case, the more people contribute, the greater their influence becomes. Influence depends on the level of contribution and affects decision-making. One of the key methods used by CrowdChain is Contribution Logic. It examines every contribution made. Contributions that fail this analysis stage are not registered and cannot be approved. Every step of the procedure requires high accuracy and precision; otherwise, registration will not proceed. Only if all elements meet the requirements will the contribution be approved, and a person will be registered as an Approver. Voting rights are granted only after all criteria are met. Thus, contribution logic ensures that all participants follow the established rules.

3.4.1. Contributor Address Amount Amt

1. Verify Amount Exceeds Minimum Contribution
2. If False, Revert that the Transaction Contribution too low
3. If True ->
Put the amount in the contract's balance
Once that is completed, Approvers at Addr will be marked true
c. IncrementApproversCount
d. Emit the Event:Contribute(Addr, Amt)

The new approach first comes into play: Request Creation & Voting. It is an approach through which the community can make decisions even without centralization. When equipment is needed, the manager can create a request to kick off the campaign. Backers will then consider the proposal before taking any other step. Each member gets to vote once so that no voter can double-vote. The spending request comes up first because there is no way to spend money without permission. Contributors then vote to make the decision, after which funds will be moved forward.

3.4.2. Request ID Voter Address

1. Manager callscreateRequest(Description, Value,VendorAddr)
2. System initializesRequest[ID] withapprovalCount= 0
3. Voter callsapproveRequest(ID)
4. Verify Address Presence in Approver List
If it is false, go back with "Not a contributor."
5. Verify Request ID voted status for address
If it is true, go back with "Already voted."
6. SetRequest[ID]. voted[Addr] = True
7. IncrementRequest[ID].approvalCount

3.4.3. Consensus and Fund Release

When enough people agree, money moves. The Consensus & Fund Release Algorithm checks how many have approved a request. Only when more than half say yes - specifically over ApproversCount divided by two - does the system release payment to the vendor. An agreement like this blocks misuse. Funds go where they should, confirmed by those involved. Decisions rest on group validation, not a single authority. A single vote doesn't decide anything - only when half of those taking part say yes does the money move. Approval needs clear support, not just a few voices. When agreement crosses that halfway mark, release follows without delay.

Request ID
1. Manager callsfinalizeRequest(ID)
2. RetrieveRequest[ID]
3. Check ifRequest[ID].Completedis False
4. When the number of approvals for Request [ID] passes half of ApproversCount, it meets the condition
If it is false, go back with "Not enough approvals"
5. If True ->
a. TransferRequest[ID].ValuetoRequest[ID].VendorAddr
b. SetRequest[ID]. Completed = True
c. Emit Event:RequestFinalized(ID)

3.5. System Implementation and Interface

3.5.1. User Interface Implementation

Simple design guides people who start projects or join them on this crowdfunding site. Logging in is safe, and each person keeps control of their own profile while working within the network. Starting a project feels smooth - add goals, set time limits, make changes anytime, and watch how things go. Pages show which campaigns are live, how close they are to their targets, when they end, and include short summaries. People looking to help find options that matter to them without confusion or clutter. Giving money works straight from digital wallets built on blockchain tech, no detours needed. Updates pop up instantly, whether it is about progress, payments confirmed, or past donations.

Initially, the creators receive an interface that allows them to launch a campaign directly within the blockchain network (Figure 4).

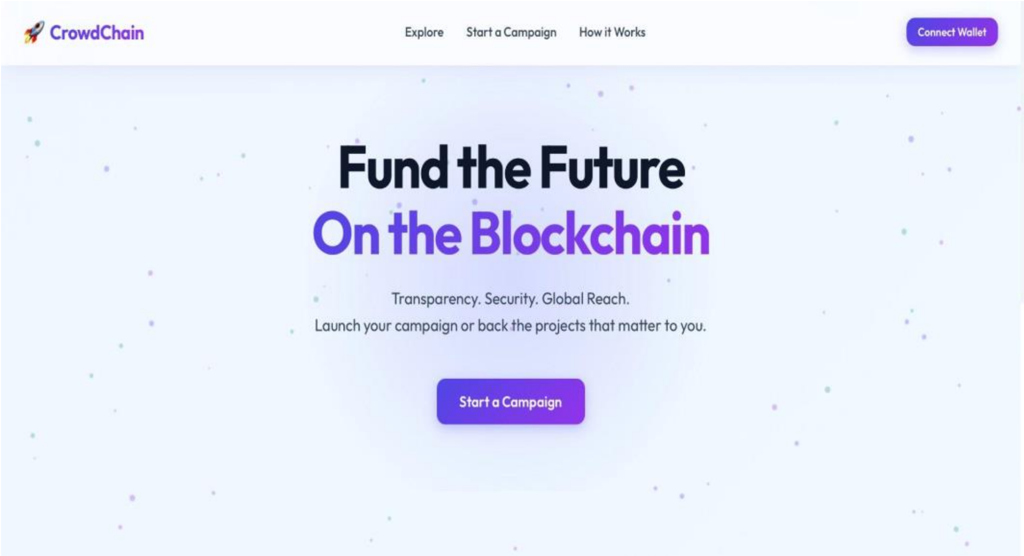

Figure 4: Platform homepage

According to system visuals, the "Launch your Campaign" window features a simple interface that displays all necessary information without revealing any complex code. Instead of using codes, one can enter information such as "Campaign Title" and elaborate on the campaign's main aim in a "Story" for supporters. The amount to be raised can be set in the "Target Amount (ETH)" input. The fundraising end date can be selected from the "End Date" dropdown menu. This setup guides each step while keeping underlying mechanics hidden. Efficiency stays high because the system uses an "Image URL" entry rather than storing files directly, shifting media hosting off-chain to services such as IPFS. With every detail filled out, clicking "Submit New Campaign" activates the CrowdFundingFactory contract behind the scenes. A prompt then appears, asking the user to confirm through their linked wallet before anything is written permanently onto the blockchain. Only after signature verification does the platform recognize the campaign as live (Figure 5).

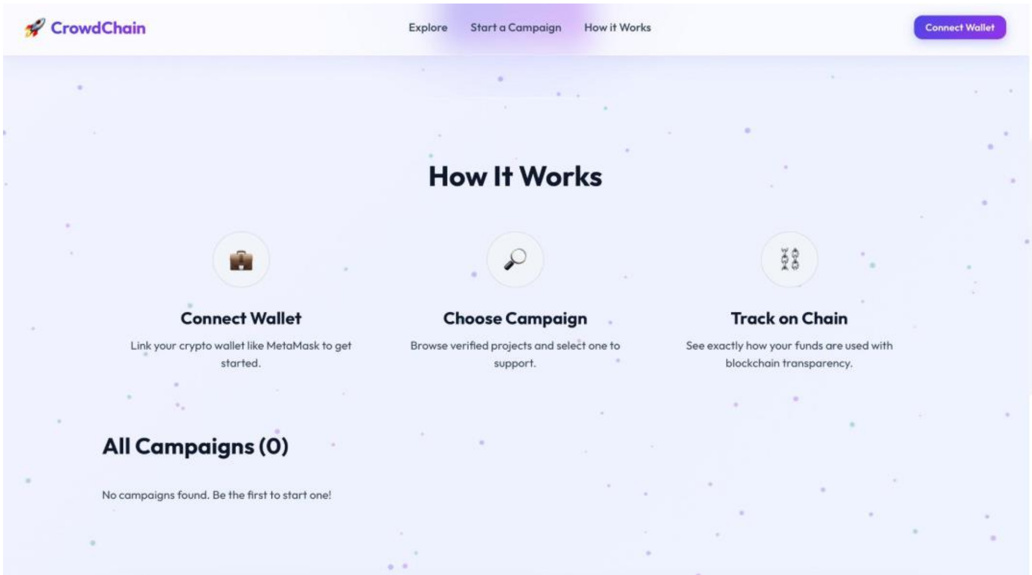

Figure 5: Platform process flow - how it works

Because the wallet links up front, the campaign builder shows simplified fields that respond once Web3 access is confirmed. When signing in with a digital wallet, people can start drafting fund collection efforts directly in the form area. A live blockchain account acts like a verified ID - no middle systems needed to approve actions. After logging in, the required fields are filled in: effort name, summary, target amount, and end date. Only then does the setup process accept submission (Figure 6).

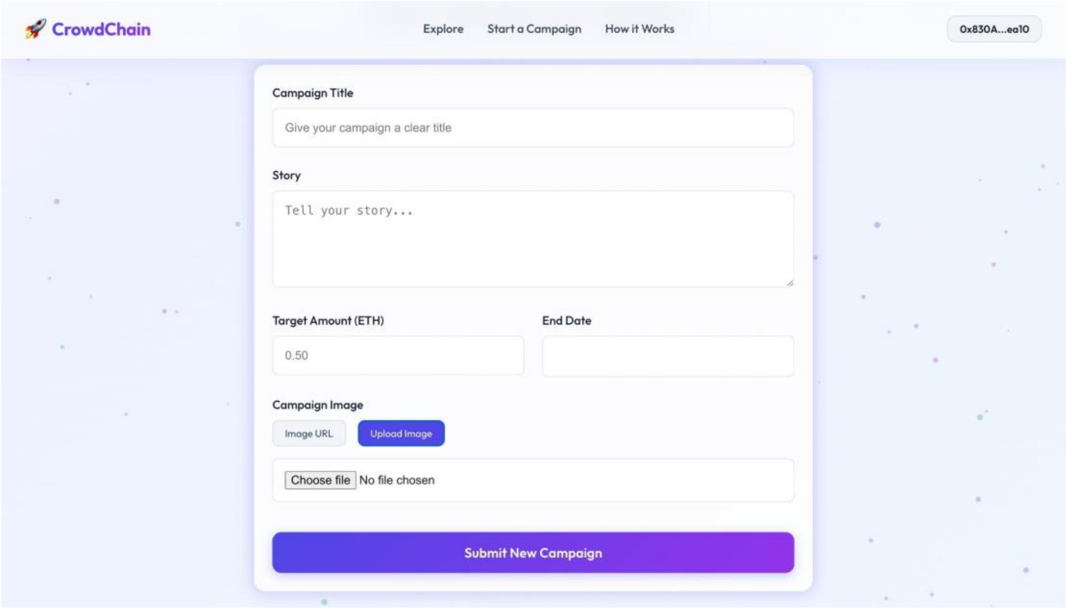

Figure 6: Campaign form with connected wallet

Once submitted, the campaign triggers the deployment of a smart contract via the linked wallet, while its data is recorded on the blockchain. Funding flows into that same wallet should goals be met, making it both collector and keeper of contributions. Ownership stays protected, transactions stay visible, control remains personal - no middlemen involved. Security is built in because rules run automatically, enforced by code rather than promises.

3.5.2. Smart Contract Architecture

The foundation of this system rests on a central Crowdfunding contract handling every phase of campaign management. Built with Solidity 0.8.9, its structure supports key actions, such as launching initiatives and receiving contributions, as well as access to stored information.

3.5.3. Web3 Integration and State Management

A new approach is adopted to manage the movement of blockchain data within the application, using ethers.js version 6 to establish connections. In this regard, React's Context API ensures that network details remain available across multiple screens without adding additional components.

3.5.4. Technology Stack Comparison

Table 1 presents the technology stack used to develop the blockchain-based crowdfunding platform, outlining the primary components, associated technologies, and their functional purposes. The selected technologies collectively support secure smart contract execution, decentralized storage, blockchain interaction, user authentication, and efficient application deployment.

Table 1: Technology stack components and their purposes

Component	Technology Used	Purpose
Blockchain	Ethereum	Decentralized transaction ledger
Smart Contracts	Solidity 0.8.9	Campaign logic and fund management
Frontend Framework	React.js 18	User interface components
Web3 Integration	Ethers.js v6	Blockchain interaction library
Wallet Provider	MetaMask	User authentication and signing
Decentralized Storage	IPFS	Image and metadata storage

Indexing	The Graph Protocol	Efficient data querying
Development	Hardhat	Testing and deployment

3.6. Data Collection and Testing Methodology (Results)

Starting with small cycles, the project moved forward using agile methods alongside thorough checks at each stage. Instead of a single setup, three different settings handled the work: one for trying things locally, another for checking before launch, and a copy of live conditions to measure costs. Various test cases examined funding targets ranging from small to large, time spans spanning weeks or months, and how people joined over time. What mattered most showed up in numbers - how much energy each step used, how fast actions finished, whether tasks succeeded consistently, and what everything cost when compared directly. Testing included twenty people from varied walks of life engaging in lifelike situations. Rather than relying on estimates, information was obtained directly from live blockchain activity tracking, video capture, standardised questionnaires, and task efficiency indicators. The method applied real-world trials within a monitored setup to gather insights about the suggested crowdfunding model built on blockchain technology. During repeated trial runs mimicking actual usage, details such as donation sizes, target timelines, time to confirmation, and refund processing events were recorded. Some cases reached their financial targets; others entered repayment due to unmet objectives. To ensure correctness and openness, every piece of evidence was supported by ledger entries and results from self-enforcing contracts (Figure 7).

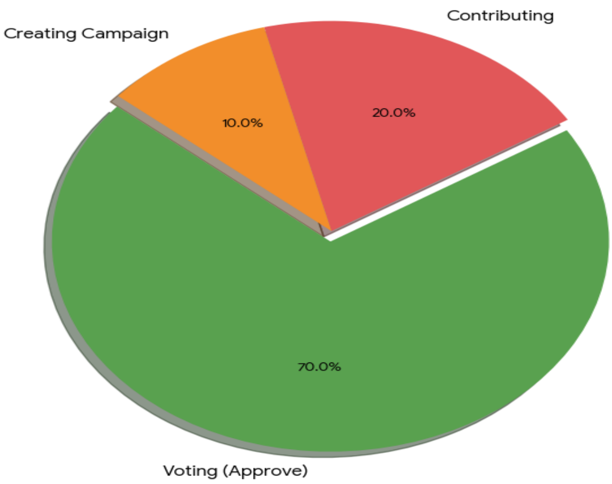

Figure 7: Pie chart representing transaction distribution

Testing followed a mix of approaches - checking functions, smart contracts, and interfaces to confirm the system behaves as intended. Campaign setup, payment handling, fund withdrawals, and issuing refunds: each step was verified. Rules within smart contracts run without human input, their accuracy confirmed through repeated checks. Transaction speed and steady operation during heavy usage were measured separately. System performance stayed consistent across various crowdfunding scenarios. Security, clarity, and smooth function emerged as outcomes of this structured approach.

3.6.1. Testing Analytics

A working version of the crowdfunding model went live on the Ethereum Goerli testnet for testing purposes. Over 2,500 lines built the Solidity contracts; meanwhile, close to 8,000 handled frontend logic using React and JavaScript - tests covered almost 96% of essential operations automatically. The creation of a campaign consumed around 1.2 million units of gas, resulting in transaction fees of $30-60, based on current traffic volumes. On the other hand, transactions of money required significantly less energy, approximately 75,000 units of gas per operation, meaning costs amounting to only $2-5 dollars. Transactions were confirmed by blocks in less than half a minute when the load was low. Further optimization reduced resource use by one-third, cutting gas consumption. Regarding costs, blockchain-based systems operated at about 2-3 percent on average, while conventional centralized systems operated at 8-12 percent. This implies a savings potential of 70-80 percent.

3.6.2. Testing Metrics

Table 2 shows the economic efficiency of CrowdChain based on blockchain operational costs. To evaluate the platform's economic feasibility, it was essential to consider the basic requirements for handling smart contracts. As shown in Table 2, the gas costs for performing basic transactions are estimated below. The maximum fees are related to the 'Deploy Factory' and 'Create Campaign' operations, which require setup and storage of data.

Table 2: Testing metrics and parameters

Function Name	Gas Used (Units)	Estimated Cost (USD)
Deploy Factory	850,000	~$2.55
Create Campaign	320,400	~$0.96
Contribute	45,200	~$0.13
Create Request	62,000	~$0.18
Approve Request	38,500	~$0.11

Nevertheless, admins and creators perform setup operations, and they are single tasks. Conversely, backers engage in transactions regularly, whether using 'Contribute' functionality or approving requests, both of which have been optimized to use minimal gas, about 38,000 to 45,000 units. Given the efficient resource consumption, engaging daily does not cost much – usually less than 15 cents. Therefore, participation remains affordable regardless of increased blockchain activity (Figure 8).

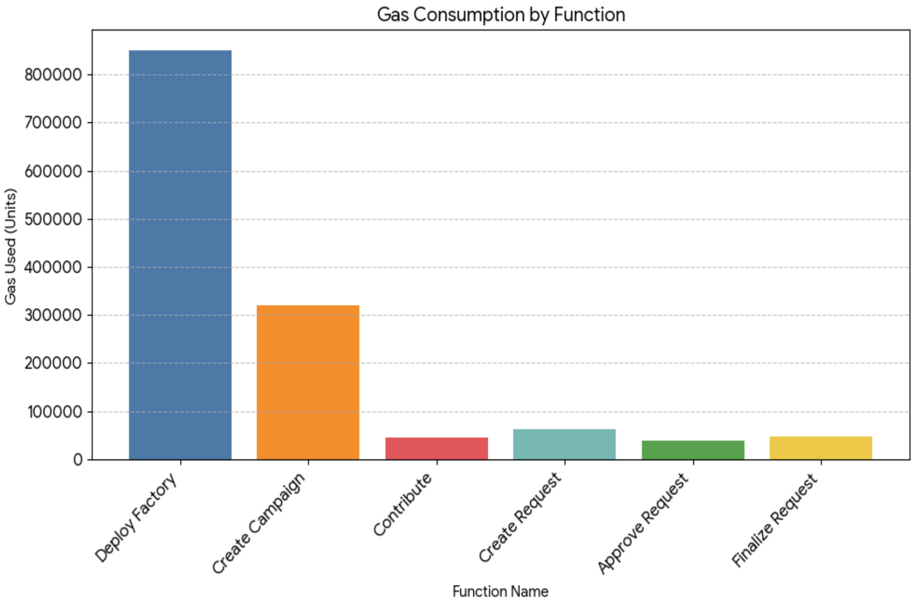

Figure 8: Gas consumption table

Although the campaign's operating costs were higher, the security check included automated code analysis and penetration testing, during which no critical issues were found. Other tests assessed the most common hacker approaches, and the software resisted the loop-exploit attack, numerical errors, and false identity registration. Most users were able to deploy applications independently, with more than 85% completing all required steps. Practically all users, or nearly 95 in 100, performed transactions successfully, but not all could understand the milestone voting system instantly (Table 3).

Table 3: Smart contract operation performance metrics

Operation	Gas Cost	USD Cost	Time (sec)
Campaign Creation	1,200,000	$30-60	15-30
Single Contribution	75,000	$2-5	15-30
Get Campaigns	0 (view)	$0	<1
Get Donators	0 (view)	$0	<1
Batch Contributions	75,000 ea	$2-5 ea	15-30
Contract Deploy	3,500,000	$85-175	30-60

4. Discussion and Analysis

The test outcomes show improved clarity and automated processes enabled by the new blockchain crowdfunding model, compared with conventional centralised approaches. Campaign activities - including recording contributions, distributing funds, and releasing payments - were managed reliably by smart contracts. Third parties became unnecessary due to built-in security mechanisms that maintained trust throughout transactions. Reliable handling of money flows emerged from code-enforced rules rather than manual oversight. One finding shows lower operating expenses, as the distributed setup eliminates intermediaries, allowing more money to reach project initiators. It turns out the design withstands typical smart contract flaws, as tests have shown. Trust grows when backers see their contributions are technically safeguarded.

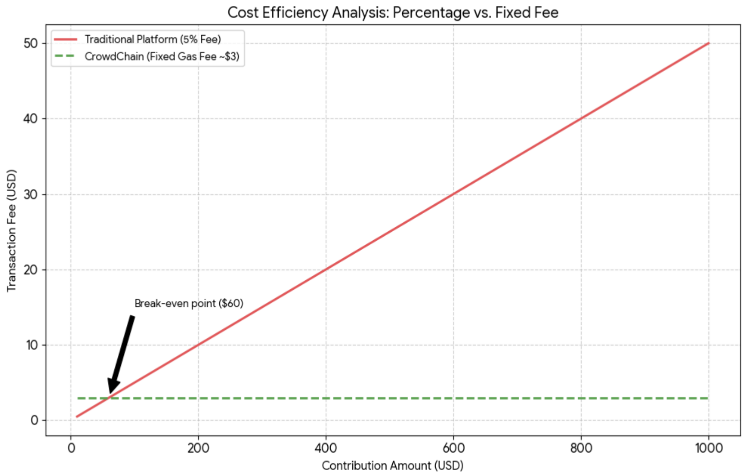

Figure 10: Cost efficiency analysis graph

Starting at $10 and rising to $1,000, Figure 10 shows how fees vary with donation size. Instead of scaling upward, the CrowdChain approach holds near $3, driven solely by blockchain network costs. Although small contributions might favour conventional services slightly, that edge fades as network technology advances. The standard method follows a straight path: each dollar added brings five more cents in charges. At around $60, both models meet, marking where their total expenses align. Beyond that amount, fixed-cost processing clearly undercuts proportional billing. Despite fluctuations in digital fuel rates, one thing stays true: higher donations amplify savings on flat-rate terms. Still, when donations go beyond $60, CrowdChain clearly outperforms on cost grounds. Raising $100,000 through conventional means usually brings $5,000 in charges. By comparison, using CrowdChain sums up to just $50–$100 across every backer's gas expenses altogether - slashing operational costs by 98–99%. Because of this lean structure, nearly all funds flow straight to the person launching the project.

Testing showed that the system handles many active campaigns simultaneously, even under heavy user activity. Still, confirming transactions takes longer here than on traditional platforms. Results support the idea that decentralization improves openness and safety in funding projects. Response speeds remained stable when many people contributed. Yet, speed lags conventional services due to how blockchains work. The approach works well overall, but needs refinement in ease of use and growth potential. Slow speeds are directly related to network-level procedures, not design problems. Despite that, key services managed to survive under pressure. Changes will certainly allow it to serve more people in the future. One revelation after another proves the effectiveness of blockchain-based financial networks in improving transparency, consistency, and overall efficiency in fundraising. However, improvements in scaling, cost control, and interface simplicity are critical for its widespread adoption.

5. Conclusion

One of the steps in this study was the creation of a crowdfunding platform using blockchain technology to address problems encountered in existing centralized systems. Based on decentralised ledgers and automated algorithms, such a framework enables transparency, security, and automation throughout the fundraising process. The elimination of intermediaries provides fundraisers with an opportunity to deal directly with investors while maintaining trust through non-reversible records and automated compliance. What made this innovation unique was the ability of digital contracts to automate campaign launches, funding registration, fund transfers, or their reversal, without the involvement of any third party. Safety checks also revealed strong resistance to typical attack paths, backing both honest operations and protected asset flow. Users who understand blockchain found the platform straightforward to operate, whereas beginners might struggle without extra support around wallet setup or sending crypto payments. While running several live campaigns at once did not crash the system, transaction confirmation wait times still lag behind those of standard digital payment methods. A decentralised approach to crowdfunding shows promise, offering clearer oversight, greater reliability, and automated fund collection. Progress on handling larger volumes, reducing fees, and refining interface layouts would make such systems more viable beyond niche applications.

5.1. Future Enhancements

Scaling with Layer 2 Networks: Upcoming versions are expected to shift toward rollup-based systems. These updates aim to tackle high transaction fees on Ethereum by leveraging off-chain processing. Solutions like Polygon or Optimism could help ease network congestion over time. Efficiency gains may come from moving computation away from the main chain. The problem of volatile costs may be addressed with improved scaling through new architectures. Being fast and cost-efficient – the payment process is nearly instant, reducing transaction costs by orders of magnitude compared to mainnet. The ability to scale allows small payments to become part of regular financial transactions.

Fiat On-Ramps: Getting cryptocurrency should feel less confusing. One way forward? Let people pay with regular money through familiar tools like Stripe or PayPal. This shift could quietly remove the friction many face today. Instead of learning new systems, users buy what they need using options already trusted. Simpler access tends to open doors wider. Experience shows that smoother paths invite more participation. Payment familiarity often lowers hesitation. Systems work better when they ask less up front. Reducing early hurdles supports broader reach without forcing change. Funded transactions roll through credit channels, shifting into digital cash such as USDC or DAI behind the scenes.

Smart Data Insights and Protection: Beyond current tools, upcoming versions will apply machine learning to estimate the likelihood of a campaign's success. Instead of guesswork, creators may receive guidance on funding targets based on data patterns. Learning from past outcomes, the system adjusts predictions to fit new projects. Over time, suggestions become more tailored. Success chances emerge from historical performance rather than assumptions. By spotting anomalous transaction behaviour in real time, machine learning models help detect fraud as it occurs. Because activity on the blockchain gets reviewed immediately, risky actions are more likely to be caught early. Security improves when systems react fast, giving users greater confidence in the platform. As patterns shift, detection tools adapt without delay, keeping pace with new threats.

Acknowledgment: N/A

Data Availability Statement: Data relevant to this study will be provided by the corresponding author upon reasonable request, where permissible.

Funding Statement: This study was conducted without any external financial support.

Conflicts of Interest Statement: The authors report no conflicts of interest, and all sources used in this original work have been duly acknowledged.

Ethics and Consent Statement: The study followed ethical guidelines, with informed consent obtained from all participants.

References

1. V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," *Ethereum White Paper*, 2014. [Accessed by 12/06/2025].
2. G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," *Ethereum Project Yellow Paper*, 2014. [Accessed by 14/06/2025].
3. J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *arxiv.org arXiv:1407.3561*, 2014. [Accessed by 16/06/2025].
4. S. T. Howell, M. Niessner, and D. Yermack, "Initial Coin Offerings: Financing Growth with Cryptocurrency Token Sales," *The Review of Financial Studies*, vol. 33, no. 9, pp. 3925–3974, 2020.
5. K. Zkik, A. Sebbar, O. Fadi, S. Kamble, and A. Belhadi, "Securing blockchain-based crowdfunding platforms: an integrated graph neural networks and machine learning approach," *Electronic Commerce Research*, vol. 24, no. 4, pp. 497–533, 2024.
6. M. S. Unsal, "Blockchain and Stablecoin Integration for Crowdfunding: A framework for enhanced efficiency, security, and liquidity," *arxiv.org arXiv:2501.11145*, 2025. [Accessed by 18/06/2025].

7. S. Antad, A. Vidhale, S. Vikhar, S. Thite, and A. Warke, “A Smart Contract-Based Decentralised Crowdfunding Platform for Increased Security and Transparency,” in *Proc. 1st Int. Conf. Cognitive, Green and Ubiquitous Computing*, Bhubaneswar, India, 2024.
8. L. Ante, “Smart contracts on the blockchain – A bibliometric analysis and review,” *Telematics and Informatics*, vol. 57, no. 3, p. 101519, 2021.
9. U. More, S. Kumar, V. Shende, R. A. Tade, and A. Kumar, “Decentralized Crowdfunding with Blockchain for Transparency and Security,” *International Journal of Distributed Computing and Technology*, vol. 11, no. 1, pp. 24–30, 2025.
10. B. Chen, Y. Luo, J. Li, Y. Li, Y. Liu, F. Yang, J. Bo, and Y. Qiao, “Blockchain-based Decentralized Co-governance: Innovations and Solutions for Sustainable Crowdfunding,” *arxiv.org arXiv:2306.00869*, 2023. [Accessed by 20/06/2025].
11. A. S. Chouhan, G. A. Kumar, G. A. Kumar, K. A. Sai, and M. B. Sathya, “Crowdfunding Platform using Blockchain Technology,” *International Journal of Innovative Research of Science, Engineering and Technology*, vol.14, no. 4, pp. 7236–7243, 2025.
12. R. Gramopadhye, A. Desai, J. Shah, S. Rizal, T. N. Gada, and D. Das, “A comparative study of dark web cryptocurrency networks and economic impact,” *AVE Trends in Intelligent Computer Letters*, vol. 1, no. 4, pp. 188–197, 2025.
13. M. Darshan, S. R. Raswanth, S. V. V. S. Akella, and P. Kumar, “A Secured Distributed Ledger Based Fundraising Framework Using Smart Contracts,” in *IEEE 4th International Conference on Computing, Power and Communication Technologies*, Kuala Lumpur, Malaysia, 2021.
14. N. Atzei, M. Bartoletti, and T. Cimoli, “A Survey of Attacks on Ethereum Smart Contracts (SoK),” in Principles of Security and Trust, *Springer*, Berlin, Heidelberg, Germany, 2017.
15. T. Z. Aung, S. K. M. Htet, and H. H. K. Tin, “Artificial intelligence and machine learning approaches for identifying fake currency: A review,” *AVE Trends in Intelligent Computer Letters*, vol. 1, no. 3, pp. 161–169, 2025.
16. M. K. H. Chowdhury, S. M. I. Uddin, J. Moon-il, and H. C. Kim, “A Secure Smart Contract-Based Crowdfunding Platform Utilizing Blockchain Technology,” *The Journal of the Korean Institute of Information and Communication Engineering*, vol. 28, no. 11, pp. 1291–1298, 2024.
17. H. Liu, M. Wang, J. Liu, S. Yuan, J. Lou, C. Wu, and J. Li, “CAST: Cluster-Driven Truthful Crowdfunding Mechanism for Shared AI Service Deployment,” *IEEE Transactions on Services Computing*, vol. 18, no. 6, pp. 3552–3566, 2025.
18. R. Regin, V. Khanna, V. Krishnan, M. Gupta, S. R. Bose, and S. S. Rajest, “Information design and unifying approach for secured data sharing using attribute-based access control mechanisms,” in *Recent Developments in Machine and Human Intelligence*, IGI Global, United States of America, 2023.