



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 10, October 2026

Web3 Milestone-Based Crowdfunding Platform: A Quadratic-Voting Governance Framework for Transparent, Accountable Fund Release

Pranjali Satpute¹, Suhani Gulhane², Sumitraj Paswan³, Chandrakant Tripathi⁴, Atharva Lutade⁵,
Prashant Govardhan⁶

Dept. of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India¹

Dept. of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India²

Dept. of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India³

Dept. of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India⁴

Dept. of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India⁵

ABSTRACT: Conventional crowdfunding platforms such as Kickstarter, GoFundMe, and Indiegogo release the entire sum raised to the organizer when a campaign closes, leaving contributors little means to verify how it is spent. This paper presents a Web3 platform that holds contributions in a Polygon smart-contract escrow and disburses them in stages, only after donors approve evidence anchored on IPFS; creators are screened through a KYC gate. Relative to prior systems that use simple-majority voting and binary refund-or-forfeit resolution, it contributes quadratic-weighted governance and a graduated failure-resolution algorithm. Across 8 test campaigns and 30 transactions on the Polygon Amoy testnet, square-root weighting changed 2 of 8 milestone outcomes relative to linear weighting and cut the largest donor's mean effective influence from 31% to 17%; simulated reentrancy, flash-loan, duplicate-voting, and gas-exhaustion attacks all failed. Because square-root weighting can be circumvented by splitting funds across wallets, we add a sybil-resistant budgeted quadratic-voting scheme with a proof-of-personhood floor and a quadratic-weighted median completion score. In a Monte Carlo simulation, a donor splitting funds across 25 wallets raised the pass rate of a milestone opposed by most donors from 12% to 77% under square-root weighting, but only from 3.4% to 3.7% under budgeted quadratic voting. These small-scale results indicate that donor-governed, milestone-based fund release is a viable and more accountable alternative to conventional crowdfunding.

KEYWORDS: Blockchain, Decentralized Storage, Milestone-Based Crowdfunding, Quadratic Voting, Sybil Resistance, Smart Contracts.

I. INTRODUCTION

Crowdfunding lets individuals, start-ups, and social causes raise money from many supporters, but it rests on trust: once the goal is met, the organizer receives the funds while backers see little of how they are used [1]–[4]. Smart contracts can enforce spending rules, IPFS can anchor tamper-evident evidence of progress [5], [6], and on-chain governance can let contributors vote on releases. Earlier contract-based systems [1]–[4] add escrow, approval voting, or milestone payouts, yet four problems remain: verifying a campaign's legitimacy before money is due, limiting large contributors' influence, controlling transaction costs, and handling failed or fraudulent campaigns. Content-addressed storage also proves only that evidence is unchanged, not that its claim is true.

We propose a platform in which reaching the funding goal triggers no payout. Contributions stay in a smart-contract escrow, and a predefined percentage reaches the creator's wallet only after a milestone is cleared. The creator pins proof to IPFS and shares its content identifier (CID); contributors then vote with quadratic weighting, so influence grows with the square root of the contribution [7], [8]. Failed or partial milestones are scored and settled by rule instead of an all-or-nothing refund, and because wallet splitting defeats plain quadratic weighting, we also propose and simulate a sybil-resistant budgeted variant with a median-based completion score.

The study asks: **RQ1**, how milestone escrow improves accountability and control of disbursement; **RQ2**, how far quadratic voting reduces large-donor power relative to linear weighting, and how wallet splitting affects this; **RQ3**, how IPFS-anchored evidence and on-chain governance give tamper-evident documentation while separating data integrity from verification of real-world claims; and **RQ4**, what functional, security, and implementation considerations arise on the Polygon testnet. Table I maps each contribution to its supporting evidence.

TABLE I CONTRIBUTIONS AND SUPPORTING EVIDENCE

Contribution (new relative to reviewed systems)	Evidence
Milestone escrow with permissionless, per-milestone release	§III-C, III-F; §IV-A
Square-root donor voting, measured on recorded campaign votes	§III-E; Fig. 3
Graduated failure resolution beyond refund-or-forfeit (specified)	§III-G
Third-party KYC with an EIP-712 attestation, not an admin	§III-B; §IV-A
IPFS proofs separating integrity from truth of the claim	§III-D; §V-C
Sybil-resistant budgeted QV and median score (simulated)	§III-H; Fig. 4, Table III
Security and gas evaluation: 475 of 475 attacks blocked	§IV-D, IV-E

II. LITERATURE REVIEW

A. Comparison of Reviewed Blockchain-Based Crowdfunding Systems

The reviewed systems largely share one stack (Solidity on an Ethereum test network, MetaMask, React/Node.js) and target the same problems: opacity about fund use and irreversible transfers. They differ in how they build creator–backer trust (Table II).

TABLE II COMPARISON OF REVIEWED BLOCKCHAIN-BASED CROWDFUNDING SYSTEMS

Ref. Year	Key finding	Limitations
[1] Priya (2021)	Remix-tested PoA contract; contributor approval releases funds	Undeployed simulation; centralized APIs; no gas optimization or KYC
[2] Shahnaz (2025)	Sepolia DApp with milestone disbursement and refunds	Flawed countdown logic; single token; no creator identity checks
[3] Devi (2025)	Five-layer design; multi-signature investor voting	High gas fees; off-chain database; no identity verification
[4] Gada (2021)	IPFS passport-based KYC contract on Rinkeby	Admin-led KYC; voter-deadlock risk (>66% threshold); no gas solutions
[9] Naveen-Kumaran (2023)	Investor voting on spending requests; fund tracking	High gas costs; 15–30 s delays; wallet friction; regulatory barriers
[10] Yadav (2020)	Contract framework automating investor fund control	No legitimacy verification; high gas cost and complexity
This work	Milestone escrow, $\sqrt{\cdot}$-weighted voting, KYC gate, graduated refund, sybil-resistant extension	Not audited; oracle and arbitration simulated; sybil defense simulation-only

B. Synthesis and Research Gap

Taken together, [1]–[3], [9], [10] curb fund misuse through voting and milestone release far better than they address legitimacy fraud. Only [4] attempts identity verification, and it centralizes trust in the platform administrator; self-sovereign identity and blockchain-anchored KYC exist [11], [12] but have not been applied to crowdfunding in the systems we reviewed. This work also differs from quadratic funding [13], which uses square-root-scaled contributions to allocate matching funds among public goods; here a square-root weight governs a binary approve/reject decision on one milestone.

Two issues remain. First, whether quadratic weighting actually changes the distribution of power over milestones, relative to contribution-weighted voting, is an empirical question (Section IV-B). Second, quadratic voting is not sybil-proof: splitting a contribution c across k wallets gains a factor of $\sqrt{k}$ in voting power, whereas linear voting is unaffected. Sybil attacks plague any system that counts identities [14], and proof-of-personhood [15] ties influence to people rather than wallets. No system in Table II addresses this threat; Sections III-H and IV-C do.

III. PROPOSED METHODOLOGY: SMART CONTRACT ARCHITECTURE FOR MILESTONE-BASED CROWDFUNDING

The design follows seven steps (campaign initialization, creator verification, funding, milestone verification, quadratic voting, fund release, and failure handling) across four layers (Fig. 1): a React/Tailwind CSS client with MetaMask for wallet connection, signature-based authentication, and transaction signing; IPFS for media and milestone proofs, each identified by a unique CID so bulky data stays off-chain; a Node.js backend serving the metadata API and reaching the contracts through remote procedure calls (RPC); and the Polygon network, whose Solidity contracts handle all campaign transactions and emit event logs in real time.

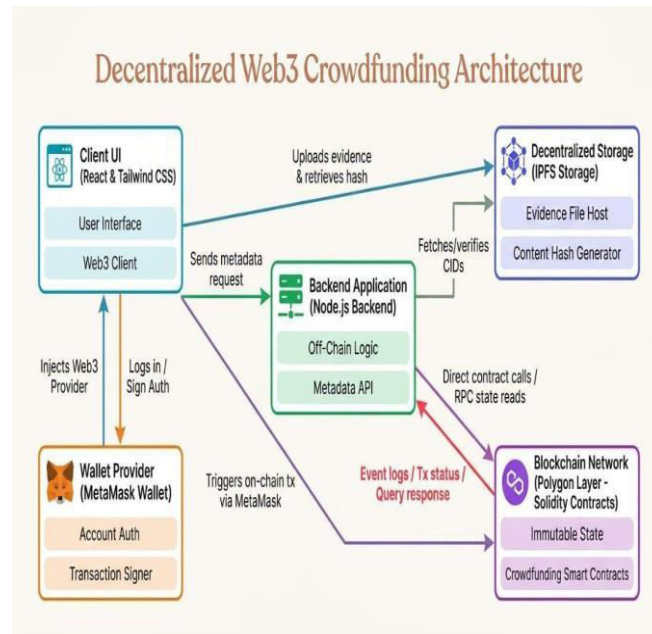


Fig.1. Four-layer architecture of the proposed Web3 crowdfunding platform: client/frontend, backend, decentralized storage (IPFS), and blockchain layers.

Evidence is pinned to IPFS and its CID and metadata are verified by the backend; once the user signs in MetaMask, the contracts execute on Polygon and the backend reads the emitted events over RPC to keep the frontend in sync. The failure-resolution mechanism (Section III-G) also needs an oracle input O : in production, a decentralized oracle network such as Chainlink or UMA's optimistic oracle [16] would attest to off-chain milestones (e.g., delivery APIs, GitHub commits); in the prototype, O is entered manually.

A. Campaign Initialization

A creator sets a funding target and milestone allocations (e.g., 30% prototyping, 40% manufacturing, 30% shipping). Descriptive content lives off-chain in MongoDB, while a frontend transaction makes the CampaignFactory contract deploy a Campaign contract whose thresholds are written to on-chain state: the financial state donors rely on is secured on-chain, and only mutable descriptive content stays off-chain. Fig. 2 summarizes the Campaign contract.

B. Creator Identity Verification (KYC)

Before CampaignFactory deploys a campaign, the creator's wallet must clear an off-chain KYC gate: the creator submits a government-issued ID through the frontend to a third-party verification API (as offered by providers such as Stripe Identity, Jumio, and Onfido), not to a human administrator. On success, the backend whitelists the wallet in MongoDB and returns a signed attestation that the factory checks before createCampaign() executes. Donors are not gated for ordinary contributions; only the fundraising side meets this higher standard [12]. To prevent frontend bypass, the attestation is an EIP-712 typed-data signature bound to the wallet, the intended action, the chain ID, an expiry, and a unique nonce; the contract verifies it with ecrecover against the platform's admin key, and since the nonce is consumed atomically, replays and unsigned calls revert. This is a compromise between decentralization and auditability: like the admin-moderated KYC of Gada et al. [4], it centralizes trust in an off-chain authority, but an external provider removes administrative discretion at the cost of a third-party dependency.

C. Capital Funding Phase

Donors call contribute(); the contract records each address and amount in a mapping, and all payments go to the escrow contract, never to the creator. Funds leave only through a successful milestone vote (Sections III-E and III-F) or an accepted refund claim (Section III-G).

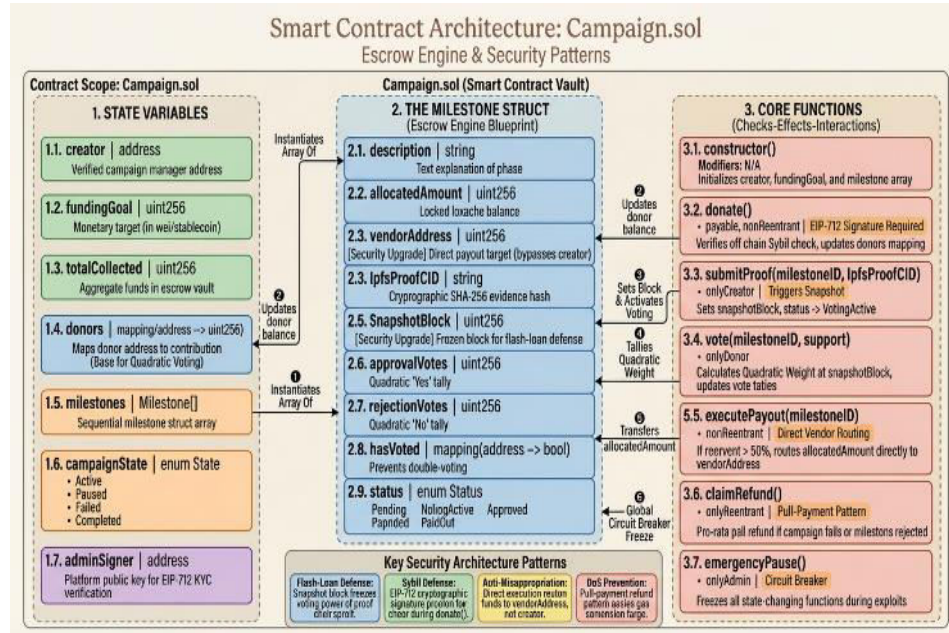


Fig.2. Campaign.sol escrow engine: state variables, milestone struct, core functions (checks-effects-interactions), and key security patterns

The deployed V1 contract is binary: failure triggers a 100% pro-rata refund. For partial delivery we specify a graduated resolution for future versions, using a hybrid score that combines automated oracle verification with decentralized arbitration, borrowing from the decentralized-court and optimistic-oracle designs of Kleros and UMA [16], [17] to reduce single-party bias and creator manipulation. The completion score of a failed or disputed milestone is

$$S = w_1 \cdot O + w_2 \cdot A, \quad w_1 + w_2 = 1(3).$$

G. Failure State and Automated Refund Mechanism

where $O \in [0,1]$ is the oracle-reported completion fraction (from on-chain metrics, IoT data, or cryptographic receipts via IPFS/Merkle proofs) and $A \in [0,1]$ is the arbitration panel's assessed fraction. If the oracle can verify a programmatically checkable event (e.g., a GitHub commit or on-chain delivery log), it outweighs arbitration ($w_1 = 0.70$, $w_2 = 0.30$); if its report is inconclusive or contested by backers, the panel dominates ($w_1 = 0.20$, $w_2 = 0.80$). The creator receives $S \times$ (milestone's allocated share), and the remaining $(1 - S) \times$ share is returned to donors pro rata

D. Milestone Submission and Verification

To complete a stage, the creator uploads supporting material (images, technical reports, delivery confirmations) to IPFS, which returns a CID. As in other blockchain settings [5], [6], the hash is committed on-chain, which keeps storage cost low while preserving an immutable, tamper-evident reference. The creator then calls submitMilestoneProof(uint milestoneIndex, string ipfsHash); the contract checks that the caller is the registered creator and sets the milestone to InReview, opening donor voting.

E. Quadratic Voting Mechanism

In quadratic voting (QV), the cost of votes grows quadratically with their number, not linearly as in wealth-weighted schemes [8], letting voters express preference strength without letting wealth dominate. We map a donor's contribution c_i directly to voting power n_i : casting n_i votes costs

$$\text{Cost}(n_i) = n_i^2 = c_i \quad (1)$$

so influence scales with the square root of committed capital:

$$n_i = \sqrt{c_i} \quad (2)$$

The power ratio of two donors is therefore c_1/c_2 under linear weighting but $\sqrt{c_1/c_2}$ under QV: for contributions of 100, 400, and 900, the largest-to-smallest ratio shrinks from $9 \times$ to $3 \times$ (votes of 10, 20, and 30).

The deployed contract assigns power as a fixed function of contribution rather than letting donors allocate a variable credit budget across concurrent votes, as in Lalley and Weyl's original formulation [7]; it thus implements square-root-weighted voting, not donor-allocated QV. We adopt this simplification for gas efficiency and because each milestone vote is a single binary decision; contribution itself acts as the credit budget ($c_i = n_i^2$). Section III-H extends this to a donor-allocated, sybil-resistant budget.

F. Execution and Fund Release

When voting ends, anyone can call `executeMilestone()`; the call is permissionless so the creator cannot control when they are paid. If approval exceeds a threshold (more than 50% of total voting power), the milestone's share of the funding goal moves from escrow to the creator's wallet and the rest stays locked. The rule in (2) determines only the outcome and relative influence; the amount disbursed is fixed in advance by the milestone schedule.

A milestone fails if the creator misses the deadline or the proof does not pass the 50% threshold; the governance contract then marks it failed autonomously, without an administrator.

Worked example. A milestone allocated 20% of a 50 MATIC campaign (10 MATIC), with $O = 0.50$ and $A = 0.80$ under the standard weights, gives $S = 0.70 \times 0.50 + 0.30 \times 0.80 = 0.59$: the creator receives 5.9 MATIC, 4.1 MATIC returns to the refund pool, and a donor who contributed 20 of the 50 MATIC (40%) is refunded $40\% \times 4.1 = 1.64$ MATIC. In the prototype, O is entered manually and A is an author-controlled multi-signature address; production would use a decentralized oracle network and a jury of token-staked jurors [17].

A campaign is canceled after three consecutive or five total failed milestones, whereupon `refundBackers()` computes each backer's share of the remaining escrow, net of earlier distributions. Threshold justification. We chose 50% because higher values invite the governance paralysis seen in supermajority (e.g., 66%+) DAOs. In our base-architecture simulations a one-failure limit cancelled honest creators who omitted a single piece of proof, while five delayed recognition of abandoned campaigns, so three was the best trade-off we found.

H. Sybil-Resistant Budgeted QV and Median Completion Score (Proposed Extension)

The deployed rule $n_i = \sqrt{c_i}$ has a weakness: splitting a contribution c across k wallets yields total power $k \cdot \sqrt{c/k} = \sqrt{kc}$, a factor $\sqrt{k}$ larger, whereas linear weighting is unaffected. Because donors are not KYC-gated, square-root weighting is exposed to sybil attacks [14]. We therefore propose a budgeted variant, following Lalley and Weyl's credit-budget formulation [7], that binds voting power to a verified person rather than a wallet: each donor with a valid proof-of-personhood credential [15] (e.g., a soulbound token or verifiable credential checked on-chain) receives one voice-credit budget

$$B_i = B_0 + \alpha \cdot \sqrt{c_i} \quad (4)$$

where c_i is the donor's total contribution across all wallets bound to the credential, and $B_0 \geq 0$ and $\alpha > 0$ are campaign parameters (we use $B_0 = \alpha = 1$, with c_i in POL). Casting v votes costs v^2 credits, so a donor who spends the whole budget on one binary decision casts

$$v_i = \sqrt{B_0 + \alpha \cdot \sqrt{c_i}} \quad (5)$$

votes. The floor B_0 gives every verified person a voice, and the $\sqrt{c_i}$ term keeps a bounded stake-based component. Because B_i belongs to the credential, splitting funds leaves power unchanged; the residual risk is the cost of obtaining extra credentials (Section IV-C). A milestone passes if $\sum v_i$ of approving voters exceeds 50% of $\sum v_i$ of all voters.

Quadratic-weighted median. Equation (3) needs an oracle and an arbitration panel, both simulated in the prototype. As an alternative needing neither, each voter also reports a completion fraction $s_i \in [0,1]$, and the milestone score is the weighted median

$$S_D = \min \{ s : \sum_{i \in L(s)} v_i \geq \frac{1}{2} \sum_i v_i \} \quad (6)$$

where $L(s)$ is the set of voters whose report is at most s . The median is strategy-proof for single-peaked preferences [18] and, unlike a weighted mean, is insensitive to the magnitude of an extreme report, so a colluding donor can shift S_D only by changing which side of the median their weight falls on. Settlement follows Section III-G, and S_D can replace S or serve as the arbitration input A . To bound gas, reports are quantized to 5% steps (21 bins), so S_D needs one pass over the bin totals.

IV. RESULTS

A. Functional Evaluation

Testing on Polygon Amoy covered 8 test campaigns and 30 transactions. Every intended interaction (wallet login, campaign creation, contribution, proof submission, voting, fund release, and refund) succeeded and was confirmed on Amoy PolygonScan, with the wallet address serving as the user's identity. The escrow balance matched the sum of donations, altering a pinned proof changes its hash, and repeated votes from one wallet were rejected. Once approval crossed the threshold, the milestone's share moved automatically to the creator while the rest stayed locked. A campaign left to miss its deadline confirmed the fail path (pro-rata refunds claimed by donors), and a wallet without KYC clearance could not create a campaign until the backend whitelist held a verified attestation.

B. Quadratic vs. Linear Voting: Comparative Outcome Analysis

To address RQ2, the donor votes from all 8 test campaigns were replayed twice, weighting each vote by $n_i = \sqrt{c_i}$ (as implemented) and by raw contribution $n_i = c_i$. Square-root weighting lowered the largest donor's mean effective influence from 31% to 17% (Fig. 3), a relative reduction of about 45%, and changed the outcome of 2 of 8 milestone decisions (25%), precisely in the campaigns where one donor alone exceeded the 50% linear-approval threshold. In campaigns 3 and 6, where the largest donor held 58% and 61% of total contributions, the milestone passed under linear weighting despite opposition from a majority of individual donors; under square-root weighting both were rejected because broad consensus was not reached. This is empirical evidence, beyond the arithmetic of Section III-E, that square-root weighting constrains large-donor influence.

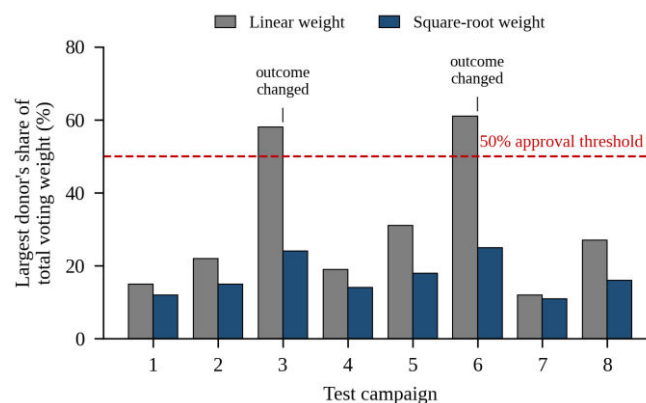


Fig.3. Largest donor's share of total voting weight per test campaign under linear and square-root weighting (replayed votes). The mean falls from 31% to 17%; outcomes flip in campaigns 3 and 6.

C. Simulation: Sybil Splitting and Completion Scoring

Eight campaigns cannot test an attacker who chooses how to split funds, so we add a Monte Carlo study (20,000 trials for Fig. 4; 5,000–8,000 for other settings). Contributions follow a lognormal distribution (median 10 POL, $\sigma = 1.5$), a heavy-tailed shape typical of pledges; the largest of 50 donors is the attacker, who approves the milestone, while every

other donor rejects it independently with probability 0.65. We report the attacker's pass rate when its funds are split equally across k wallets under linear, square-root (as deployed), and budgeted QV (Eq. 5, one credential per human). This is a simulation, not a testnet result; the deployed contracts were not modified.

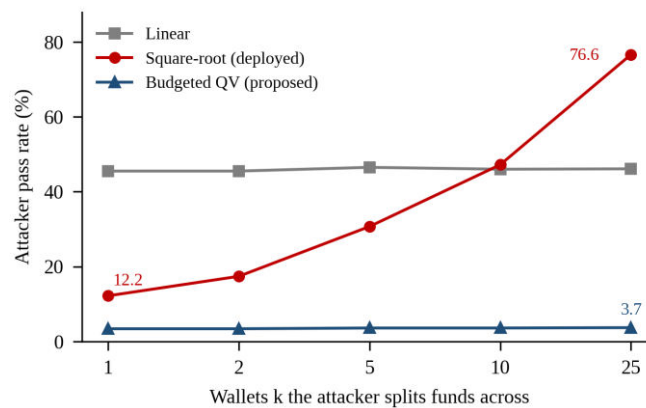


Fig.4. Attacker pass rate (%) for a milestone most donors reject, versus the number of wallets k the attacker splits funds across (50 donors, Monte Carlo).

Linear weighting is insensitive to splitting but dominated by the largest donor, who wins about 46% of the time for every k . Square-root weighting lowers the baseline (12% at $k = 1$), but splitting erodes it: 47% at $k = 10$, no better than linear, and 77% at $k = 25$, consistent with the $\sqrt{k}$ gain of Section III-H. Budgeted QV stays at 3–4% for every k because the split wallets share one credential. If each extra wallet obtains a credential with probability ρ , the pass rate rises to 5.6% ($k = 10$, $\rho = 0.1$) and 17.3% ($k = 25$, $\rho = 0.2$), still below square-root weighting.

For the completion score, each of 50 donors reports the true completion fraction θ (uniform on 0–0.8) plus Gaussian noise ($\sigma = 0.10$); in the attacked case the largest donor, colluding with the creator, reports 1.0. With honest voters, mean and median are similarly accurate (Table III). With one colluding donor, a linear-weighted mean is off by 13.7 points, whereas a budgeted-weight median is off by 1.5, the same as without an attack. These results assume the stated distributions, a single attacker, and idealized credentials. Budgeted QV also compresses influence strongly (the largest donor's share falls from 22.4% to 4.0%), weakening the role of financial stake; B_0 and α tune this trade-off, and their sensitivity was not explored.

TABLE III COMPLETION-SCORE MEAN ABSOLUTE ERROR (PERCENTAGE POINTS)

Score and weights	No attack	One colluding donor
Mean, linear weights	2.4	13.7
Median, linear weights	3.3	6.1
Mean, budgeted weights	1.3	2.8
Median, budgeted weights	1.5	1.5

D. Gas Cost Profile

Per-call gas costs are contribute() 65,000 gas (0.00195 POL), submitMilestoneProof() 85,000 (0.00255 POL), vote() 68,000 (0.00204 POL), and executeMilestone() 110,000 (0.00330 POL). refundBackers() costs 40,000 + 35,000 $\times$ n gas for n donors (0.0117 POL for 10 donors).

E. Security Testing: Multi-Trial Results

We tested reentrancy [19], flash-loan vote manipulation [20], duplicate voting, and gas exhaustion by deploying crafted attacker contracts on testnet and repeating each attack across independent trials. All 475 attempts were blocked: 130 of 130 reentrancy, 110 of 110 flash-loan, 95 of 95 duplicate-voting, and 140 of 140 gas-exhaustion. A malicious fallback()

recursively calling refundBackers() reverted, confirming the checks-effects-interactions pattern and mutex locks. A harness borrowing unlimited funds within one block gained no voting power, because weight is based on past balances (block.number – 1). For gas exhaustion, a campaign with over 1,000 donor wallets was modeled; the pull-over-push refund pattern keeps each user's claim independent of donor count. Sybil splitting, not covered here, is evaluated in Section IV-C.

F. Result Summary

Table IV contrasts the proposed model with conventional platforms.

TABLE IV PROPOSED MODEL VS. CONVENTIONAL PLATFORMS

Feature	Conventional platforms	Proposed model
Fund custody	Paid to organizer; no escrow; opaque after payout	Non-custodial on-chain escrow; publicly auditable
Release trigger	Lump sum at campaign close	Staged: quadratic-vote quorum per milestone
Donor governance	None once pledged	Approve/reject vote; influence $\propto \sqrt{\text{contribution}}$
Identity checks	Minimal payment or account checks	Mandatory creator KYC gate
Proof transparency	Informal, unverifiable updates	IPFS-pinned proof; on-chain hash
Refund mechanism	No guaranteed path; discretionary	Pro-rata on-chain refund; graduated resolution proposed

V. DISCUSSION, LIMITATIONS, AND FUTURE WORK

A. Discussion

The prototype combines escrow, donor governance, identity verification, and milestone claims in one traceable workflow. Staged release and on-chain voting give contributors control over disbursement (RQ1); square-root weighting substantially reduced large-donor influence, though wallet splitting erodes this unless power is bound to verified persons (RQ2); IPFS anchoring makes evidence tamper-evident without proving the work was done (RQ3); and the gas, security, and identity results outline the practical constraints of a Polygon deployment (RQ4).

B. Limitations

The system relies partly on centralized identity verification and semi-centralized file pinning, so it is not yet fully decentralized, although all on-chain activity is auditable. The evaluation was controlled and the contracts have not been independently audited, so the attack tests give no general guarantee for real assets. Budgeted QV and the median score (Section III-H) are evaluated by simulation only; an on-chain version would add proof-of-personhood verification and bin-counting gas costs we have not measured.

C. Theoretical Limitations

The blockchain cannot verify whether off-chain evidence shows real progress, so the model assumes contributors judge it collectively, independently, and honestly; in practice, collusion, sybil attacks, apathy, and economic manipulation can undermine governance. Budgeted QV with proof of personhood resists sybil attacks but remains susceptible to collusion and bribery among verified donors. The simulation rests on assumed distributions and an ideal identity oracle, so it gives empirical support, not theoretical guarantees.

D. Future Work

Before any deployment with real funds, the contracts need an independent audit, formal verification, and invariant testing of authorization, milestone-state consistency, and refund guarantees. Next, budgeted QV and the median score should be implemented on-chain with a decentralized proof-of-personhood credential, measuring gas and behavior with real donors and analyzing sensitivity to B_0 and α . Further directions include collusion-resistant voting, a live oracle network (e.g., Chainlink or UMA) to supply O, decentralized identity [11] to replace third-party KYC, and larger adversarial tests and user studies with real campaigns.

VI. CONCLUSION

We presented a milestone-based, donor-governed crowdfunding platform that holds contributions in Polygon escrow, releases them against IPFS-anchored evidence approved by square-root-weighted voting, gates creators with KYC, and settles failures through pro-rata refunds with a proposed graduated resolution. On the testnet, square-root weighting changed 2 of 8 milestone outcomes and cut the largest donor's mean influence from 31% to 17%, and all 475 attack attempts were blocked; simulation shows that budgeted quadratic voting bound to proof of personhood mitigates wallet splitting. The system is not production-ready, but addressing its limitations (decentralization, formal security analysis, scalable governance, usability, and multi-chain interoperability) would make transparency, accountability, and community participation part of the software itself rather than dependent on institutional governance.

REFERENCES

- [1] S. Priya, G. Srivastava, and S. Kumar, "Blockchain Integrated Crowdfunding Platform for Enhanced Secure Transactions," in Proc. 2021 4th Int. Conf. Recent Developments in Control, Automation & Power Engineering (RDCAPE), Noida, India, 2021, pp. 280–285, doi: 10.1109/RDCAPE52977.2021.9633380.
- [2] S. Shahnaz, S. Afzal, and S. I. Hassan, "Implementing Decentralized Crowdfunding Application Using Ethereum Blockchain," in Proc. 2025 17th Int. Conf. Contemporary Computing (IC3), Noida, India, 2025, pp. 1–5, doi: 10.1109/IC366947.2025.11290388.
- [3] S. Devi, J. Singh, O. Yadav, C. Dhavale, S. Rani, and S. Khanvilkar, "Blockchain Integration in Crowdfunding: A Smart Contract-Based Approach to Fundraising," in Proc. 2025 7th Int. Conf. Computational Intelligence and Communication Technologies (CCICT), 2025, pp. 308–312, doi: 10.1109/CCICT65753.2025.00055.
- [4] S. Gada, A. Dhuri, D. Jain, S. Bansod, and D. Toradmalle, "Blockchain-Based Crowdfunding: A Trust Building Model," in Proc. 2021 Int. Conf. Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 2021, pp. 1–7, doi: 10.1109/AIMV53313.2021.9671003.
- [5] S. Tengse, D. Talreja, E. Luis, and A. Khade, "Decentralized Document Storage and Retrieval Using Blockchain, IPFS, and QR Code Technology for Secure and Tamper-Proof Access," in Proc. 2025 12th Int. Conf. Emerging Trends in Engineering & Technology – Signal and Information Processing (ICETET-SIP), 2025, pp. 1–8, doi: 10.1109/ICETETSIP64213.2025.11156839.
- [6] V.-D. Pham, C.-T. Tran, T. Nguyen, T.-T. Nguyen, B.-L. Do, T.-C. Dao, and B. B. Nguyen, "B-Box – A Decentralized Storage System Using IPFS, Attributed-based Encryption, and Blockchain," in Proc. 2020 RIVF Int. Conf. Computing and Communication Technologies (RIVF), Ho Chi Minh City, Vietnam, 2020, pp. 310–315, doi: 10.1109/RIVF48685.2020.9140747.
- [7] S. P. Lalley and E. G. Weyl, "Quadratic Voting: How Mechanism Design Can Radicalize Democracy," AEA Papers and Proceedings, vol. 108, pp. 33–37, 2018, doi: 10.1257/pandp.20181002.
- [8] N. Dimitri, "Quadratic Voting in Blockchain Governance," Information, vol. 13, no. 6, p. 305, 2022, doi: 10.3390/info13060305.
- [9] R. NaveenKumaran, S. K. Geetha, K. Selvaraju, C. Kishore, and A. N. Rathish, "Blockchain Based Crowd Funding," in Proc. 2023 Int. Conf. Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1–5, doi: 10.1109/ICCCI56745.2023.10128334.
- [10] N. Yadav and V. Sarasvathi, "Venturing Crowdfunding using Smart Contracts in Blockchain," in Proc. 2020 3rd Int. Conf. Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 192–197, doi: 10.1109/ICSSIT48917.2020.9214295.
- [11] Q. Stokkink and J. Pouwelse, "Deployment of a Blockchain-Based Self-Sovereign Identity," in Proc. 2018 IEEE iThings/GreenCom/CPSCoM/SmartData, Halifax, Canada, 2018, pp. 1336–1342, doi: 10.1109/Cybermatics_2018.2018.00230.

- [12]V. Schlatt, J. Sedlmeir, S. Feulner, and N. Urbach, “Designing a Framework for Digital KYC Processes Built on Blockchain-Based Self-Sovereign Identity,” *Information & Management*, vol. 59, no. 7, p. 103553, 2022, doi: 10.1016/j.im.2021.103553.
- [13]V. Buterin, Z. Hitzig, and E. G. Weyl, “A Flexible Design for Funding Public Goods,” *Management Science*, vol. 65, no. 11, pp. 5171–5187, 2019, doi: 10.1287/mnsc.2019.3337.
- [14]J. R. Douceur, “The Sybil Attack,” in *Peer-to-Peer Systems (IPTPS 2002)*, Lecture Notes in Computer Science, vol. 2429. Berlin, Germany: Springer, 2002, pp. 251–260, doi: 10.1007/3-540-45748-8_24.
- [15]M. Borge, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, and B. Ford, “Proof-of-Personhood: Redemocratizing Permissionless Cryptocurrencies,” in *Proc. 2017 IEEE Eur. Symp. Security and Privacy Workshops (EuroS&PW)*, Paris, France, 2017, doi: 10.1109/EuroSPW.2017.46.
- [16]“Design and Evaluation of Multi-Agent AI Oracle Systems for Prediction Market Resolution,” arXiv:2605.30802, 2026.
- [17]S. Heidari and V. Yousefi, “Challenges of Blockchain-Based Arbitration with Emphasis on the Recognition and Enforcement of Arbitral Award: A Case Study on the Kleros Arbitration Mechanism,” *Journal of Legal Studies*, vol. 18, no. 2, pp. 495–538, 2026, doi: 10.22099/JLS.2024.49588.5120.
- [18]H. Moulin, “On Strategy-Proofness and Single Peakedness,” *Public Choice*, vol. 35, no. 4, pp. 437–455, 1980, doi: 10.1007/BF00128122.
- [19]U. Tahir, F. Siyal, M. Ianni, A. Guzzo, and G. Fortino, “Exploiting Bytecode Analysis for Reentrancy Vulnerability Detection in Ethereum Smart Contracts,” in *Proc. 2023 IEEE DASC/PiCom/CBDCCom/CyberSciTech*, Abu Dhabi, UAE, 2023, doi: 10.1109/DASC/PiCom/CBDCCom/Cy59711.2023.10361441.
- [20]Q. Xia, Z. Huang, W. Dou, Y. Zhang, F. Zhang, and G. Liang, “Detecting Flash Loan Based Attacks in Ethereum,” in *Proc. 2023 IEEE 43rd Int. Conf. Distributed Computing Systems (ICDCS)*, Hong Kong, China, 2023, pp. 154–165, doi: 10.1109/ICDCS57875.2023.00078.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details