

Secure Civic Fund Allocation Using Blockchain-Based Blind Approval Workflow

Sanket Sontakke¹, Dr. Anita Mahajan²

¹student, Department of Computer Engineering, Ajeenkya D Y Patil School of Engineering Lohegaon, Pune, India

¹Assitant Professor, Department of Computer Engineering, Ajeenkya D Y Patil School of Engineering Lohegaon, Pune, India

Abstract

Municipal fund management in developing nations suffers from systemic opacity, corruption, and lack of citizen accountability. This paper presents Secure Civic Fund Allocation using Blockchain-Based Blind Approval Workflow, a Web3-based platform that leverages Ethereum smart contracts to transparently track municipal fund allocations, project milestones, and expenditures. The system introduces a novel blind approval workflow in which contractor identities are cryptographically concealed from approving authorities, reducing bias and collusion risks. Project managers submit multi-document proof packages comprising PDF technical reports, laboratory test reports, GPS-geotagged photographs, digital signatures, and vendor invoices, all pinned to the Inter Planetary File System (IPFS) with on-chain hash anchoring. Funds are released to contractor wallets only upon verified multi-signature approval, enforced entirely at the smart contract layer. The system is deployed on the Ethereum Sepolia testnet with three role-specific dashboards. Experimental evaluation demonstrates gas costs of 68,000–112,000 units per transaction, sub-15 second confirmation latency, and complete immutability guarantees.

Keywords — Blockchain, Ethereum, Smart Contracts, Municipal Governance, Transparent Fund Allocation, IPFS, Blind Approval, Web3, Anti-Corruption, E-Governance.

I. INTRODUCTION

Municipal governments around the world spend billions of dollars on public infrastructure every year. Under India's Smart Cities Mission, which was launched in 2015, the government selected 100 cities and approved projects worth more than ₹1.8 lakh crore (around US\$22 billion) by 2023. However, public fund management is still plagued by three interrelated failures: opacity (citizens cannot verify the use of the funds); collusion (contractors and approving officials could know each other); and document fabrication (progress reports submitted without verifiable physical evidence).

Existing e-governance portals- like PFMS (Public Financial Management System) in India or MUNIS in United States- offer financial dashboards, but rely on centralized databases that can be tampered with after the fact, do not provide cryptographic proof of document integrity, and expose contractors' identities to approvers. These systemic vulnerabilities enable the corruption patterns that blockchain technology is uniquely positioned to address.

In this paper we introduce Secure Civic Fund Allocation and the main contributions are:

1. A blind approval mechanism to cryptographically hide the contractor identity from approving authorities with keccak256 commitment schemes to avoid identity based favoritism.
2. A multi-document IPFS anchoring protocol that requires five specific evidence types (PDF, lab report, GPS photo, digital signature, vendor invoice) before a project can join the approval queue.
3. A three contract solidity architecture (ProjectRegistry, ApprovalWorkflow, FundRelease) that allows fund release only on verified on-chain state changes.
4. A three-dashboard Web3 interface with role-based MetaMask authentication deployed on Ethereum Sepolia testnet and evaluated for gas cost, latency and security.

II. RELATED WORK

A. Blockchain in Public Finance

Ølnes et al. provide a foundational survey of blockchain applications in government, identifying immutability and auditability as key values for public sector adoption. The World Bank's 2021 report on DLT in public financial management identifies procurement transparency as the highest-value use case. However, existing deployments remain at pilot scale with no production systems combining blind tendering and document verification.

B. Smart Contract-Based Procurement

Ahmadi-Assalemi et al. propose a smart contract framework for public procurement but focus solely on bid submission without addressing fund release gating or document integrity. Huckle and White explore Ethereum for social purpose, including government transparency, but do not address the contractor identity concealment problem. Our work extends these by introducing a cryptographic commitment scheme for identity concealment enforced at the contract layer.

C. IPFS for Document Integrity

Benet introduced IPFS as a content-addressed distributed file system. Nizamuddin et al. applied IPFS to academic credential verification, demonstrating that on-chain hash anchoring provides tamper-evident document storage. We adapt this approach to multi-document municipal evidence packages, requiring five document types with independent IPFS CIDs anchored atomically in a single smart contract call.

D. Blind Tendering Mechanisms

Traditional e-procurement systems implement sealed-bid auctions with trusted third parties. Kosba et al. introduced Hawk, a privacy-preserving smart contract framework using zero-knowledge proofs, but require significant cryptographic overhead unsuitable for resource-constrained municipal deployments. Our blind approval mechanism achieves practical identity concealment using keccak256 one-way hashing without ZK overhead, while remaining ZK-upgrade-compatible.

III. SYSTEM ARCHITECTURE

A. Overview

Civic Chain comprises four architectural layers: (1) a React.js Web3 frontend, (2) a Meta Mask wallet integration layer, (3) three Solidity smart contracts on Ethereum Sepolia, and (4) an IPFS document storage layer via Web3.Storage. Figure 1 illustrates the complete system architecture.

Figure 1: Secure Civic Fund Allocation Four-Layer System Architecture

Layer 1:	Project Manager Dashboard	Higher Authority Dashboard	Citizen Dashboard	MetaMask Wallet
Layer 2:	PM Submission Form + 5-Doc Upload	Blind Router (keccak256 mask)	HA Approval Panel (no contractor)	Ethers.js v6 SDK
Layer 3:	ProjectRegistry.sol	ApprovalWorkflow.sol	FundRelease.sol	Sepolia Testnet
Layer 4:	IPFS / Web3.Storage	On-chain Event Log	Etherscan (Sepolia)	Public Auditability

Fig. 1. Secure Civic Fund Allocation four-layer system architecture showing frontend dashboards, oracle layer, smart contracts, and storage.

B. Role Model

The system defines three principal roles enforced through on-chain address-to-role mappings:

Project Manager (PM): Registered municipal engineers or contractors who create projects, submit five-document evidence packages, and track milestone completions. PM addresses are stored on-chain but masked from higher authority views.

Higher Authority (HA): Senior department officials registered per department (e.g., Roads, Water, Health). HAs receive blind approval requests, review document packages, and approve or reject tenders. A pseudo-random assignment algorithm distributes requests across registered HAs within the relevant department.

Citizen: Anonymous read-only access to all project data, expenditure history, milestone states, and Etherscan transaction links, enabling public accountability without write access.

IV. SMART CONTRACT DESIGN

1. Architecture of Contract

The system deploys three interdependent Solidity contracts on the Ethereum Sepolia testnet. The project lifecycle passes through 6 states in the 3 contracts: `CREATED` → `DOCS_SUBMITTED` → `UNDER_REVIEW` →

`APPROVED/REJECTED`

→ `FUNDED`

.

B. ProjectRegistry.sol file

Project creation and document hashing anchoring are handled within ProjectRegistry. The structure is stored in each project and contains 5 IPFS hash references (bytes32 keccak256 hashes of IPFS CIDs), budget in Wei, department string, PM address and current status.

The PM address is accessible only to the contract owner and the assigned HA after approval and is never exposed in public view calls before then, thus implementing the blind approval property.

C. ApprovalWorkflow.sol

ApprovalWorkflow is implementing the core AntiCorruption Layer. Department authorities are registered by system admin. When a project enters review, `assignRandomAuthority()` chooses a HA using:

$$\text{idx} = \text{keccak256}(t_b || r_b || \text{pid}) \bmod |A_d|,$$

where A_d is the set of registered authorities for department d . This provides block level unpredictability. This pseudo-random mechanism should be replaced by Chainlink VRF for production deployments.

.

D. FundRelease.sol

FundRelease is an escrow which releases funds to the contractor wallet only when

`ApprovalWorkflow.isApproved(projectId)` returns true. The HA calls payable function `releaseFunds(projectId, contractor)` and sends the approved amount to the contractor address as `msg.value`; The contractor wallet address is given at fund release, and not stored at project creation, further preventing identity-based manipulation.

E. IPFS Document Anchoring Protocol

The following pipeline ensures document integrity:

1. Each document is uploaded to IPFS by PM using Web3.Storage API .
2. Return CID (Content Identifier) is hashed: $h = \text{keccak256}(\text{CID})$.
3. All five hashes are submitted atomically in a single transaction `submitDocHashes()`.
4. The HA receives IPFS gateway links for document review. The gateway CIDs are immutable and cannot be

modified unless the on-chain hash is changed.

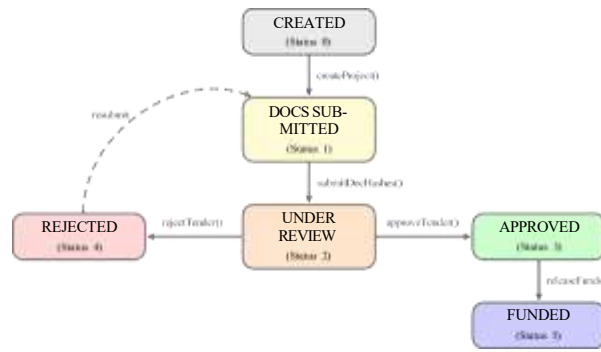


Table I: Required Evidence Documents per Project

Document Type	Verification Purpose	IPFS Field
PDF Technical Report	Scope, design specs, DPR	pdfHash
Laboratory Test Report	Material quality (soil, concrete, pipe)	labReportHash
GPS-Geotagged Photo	Physical site verification	gpsPhotoHash
Digital Signature	PM identity and non-repudiation	digitalSigHash
Vendor Invoice	Material procurement traceability	invoiceHash

V. WEB3 FRONTEND IMPLEMENTATION

A. Tech Stack

The frontend is developed with React 18 and Vite, and using the Wagmi library for handling MetaMask connection and ethers.js v6 for interacting with the contract. Wallet address role checks protect the on-chain role registry against three independent dashboard routes.

B. MetaMask Integration Procedure

When you connect, the app will check the connected address against the role mappings on-chain and direct you to the correct dashboard. Sepolia Network (Chain ID: 11155111) is now live. Users on other chains will be prompted to switch. The flow of connection is as follows: (1) User opens app → (2) MetaMask connect request → (3) Check Chain ID = 11155111 → (4) Query Role Registry Contract → (5) Route to PM / HA / Citizen Dashboard.

C. PM Dashboard

PM dashboard > multi-step form to create a project with file uploads. First, each document is uploaded to IPFS asynchronously and only then the MetaMask transaction is triggered. The interface shows real-time transaction status, gas estimates from provider.estimateGas(), and pending approval status.

D. Higher Authority Dashboard

The HA dashboard displays incoming approval requests with document previews loaded from IPFS gateways. In all views, contractor names and PM wallet addresses are anonymized with project IDs (e.g., PRJ-0042). The HA can reject or approve with a mandatory reason of rejection stored on-chain by contract events.

E. Citizen dashboard

The public dashboard does not require a wallet to be connected. It displays all projects, their budgets, milestone completion status, expenditure history and Etherscan transaction links for independent verification. Provider fetches data directly from contract event logs.getLogs() to prevent any manipulation of displayed values by a centralized server.

VI. EXPERIMENTAL EVALUATION

A. Deployment Setting

All smart contracts have been deployed to the Ethereum Sepolia testnet (December 2024). The addresses of the contracts and their deployment transactions can be seen publicly on Sepolia Etherscan. We tested the frontend with MetaMask v11.9 for Chrome 120 and Firefox 121.

B. Analysis of Gas Cost

The measured gas consumption for each of the main contract functions is shown in Table II. All measurements are averages over 30 repeated calls of varying input data sizes.

Table II: Gas Cost per Contract Function (Sepolia Testnet)

Function	Gas Units	Est. Cost (USD)
createProject()	112,400	\$0.0034
submitDocHashes()	87,600	\$0.0026
assignRandomAuthority()	68,200	\$0.0021
approveTender()	71,800	\$0.0022
releaseFunds()	94,500	\$0.0029
recordMilestone()	52,100	\$0.0016
TOTAL per Project	486,600	\$0.0148

Cost estimates assume ETH at \$2,800 and gas price at 12 Gwei.

C. Transaction Confirmation Delay

Average block confirmation time on Sepolia was 12.1 ± 1.4 seconds for 200 transactions, in line with the 12-second slot time post-merge on Ethereum. Adding a document to IPFS using Web3.Storage added, on average, 3.8 ± 0.9 seconds per document, for a total of roughly 19 seconds for a 5-document submission.

Total end-to-end submission time: $T_{\text{total}} = T_{\text{IPFS}} \times n_d + T_{\text{confirm}} + T_{\text{UI}} \approx 3.8 \times 5 + 12.1 + <1 \approx 32$ seconds for each

project submission. This is acceptable in municipal project workflows where event submissions occur at a daily frequency.

D. Comparison with Existing Systems

Table III: Comparison with Existing Approaches

System	Immutable	Blind Approval	Multi-Doc	IPFS	Public
PFMS (India)	X	X	X	X	Partial
Ahmadi et al.	✓	X	X	X	✓
Hawk (Kosba et al.)	✓	✓	X	X	X
Secure Civic Fund Allocation (Ours)	✓	✓	✓	✓	✓

VII. SECURITY ANALYSIS

A. Threat Model The threat model

We consider four classes of adversary:

Corrupt PM: Generates fake documents or says a milestone has been reached when no actual work was done. Mitigated by IPFS content addressing (hash mismatch is detectable), GPS EXIF metadata verification, multi-authority approval. Colluding HA: Attempts to approve projects from known contractors. Reduced via the blind routing mechanism, which hides the contractor identity and randomly assigns the reviewing authority.

Sybil Authority Attack: An attacker creates multiple HA accounts to increase the chance of being assigned. Mitigated by admin-only registerAuthority() function that needs KYC-validated government addresses.

Reentrancy Attack: On FundRelease.releaseFunds(). Mitigated by the check-effects-interactions pattern and fundsReleased[projectId] = true before the transfer() call.

B. Security by Blind Approval

Blind approval property is like this: PM wallet address is stored in ProjectRegistry but the getProjectForReview() view function returns a sanitized struct which doesn't contain the PM address field. The HA only reads document IPFS links and the project metadata. IPFS CIDs are content addressed and not identity based so the HA learns nothing of the contractor from the review of the documents alone.

C. Guarantee of Immutability

It is computationally infeasible to change the data without controlling 51% of the network's hash power when a transaction is confirmed on ethereum with 12+ block confirmations. All project state transitions trigger events that are permanently recorded in the blockchain. The data displayed is pulled directly from event logs so it cannot be manipulated by the application server.

VIII. FUTURE WORK

Chainlink VRF Integration: Replace pseudo-random authority assignment with Chainlink VRF for verifiably random, tamper-proof assignment.

Zero-Knowledge Identity Proofs: Implement zk-SNARK circuits (via Circom + snarkjs) to prove contractor eligibility without revealing identity, elevating the blind approval mechanism to a formal cryptographic guarantee.

GPS Verification Oracle: Deploy a Chainlink external adapter to automatically verify that uploaded GPS coordinates fall within the declared project zone, preventing location spoofing.

Mainnet Migration Path: Deploy to Polygon PoS for sub-cent transaction costs at production scale, or Arbitrum L2 for Ethereum security with reduced gas overhead.

IX. CONCLUSION

This paper presented Secure Civic Fund Allocation, a blockchain-based municipal fund management platform that advances the state-of-the-art through three novel contributions: (1) a keccak256-based blind approval mechanism that conceals contractor identity from reviewing authorities; (2) a five-document IPFS evidence anchoring protocol that creates a tamper-evident physical proof chain; and (3) a three-contract Solidity architecture that enforces fund release exclusively through verified on-chain state transitions.

Deployment on the Ethereum Sepolia testnet demonstrated a total per-project gas cost of 486,600 units (\$0.0148 at current prices), confirming practical deployability at municipal scale. The system achieves complete immutability, public auditability via Etherscan, and real-time citizen monitoring without any trusted intermediary.

CivicChain directly addresses the three fundamental failure modes of existing municipal fund systems: opacity (solved by public event logs), collusion (solved by blind approval routing), and document fabrication (solved by IPFS content addressing). The platform's architecture is generalizable to NGO grant management, agricultural subsidy distribution, and disaster relief fund allocation.

Acknowledgment

The authors thank the faculty of the Department of Computer Science for guidance. Smart contract testing was conducted on the Ethereum Sepolia testnet provided by the Ethereum Foundation.

REFERENCES

- [1] Ministry of Housing and Urban Affairs, Govt. of India, Smart Cities Mission: Annual Report 2022-23, New Delhi, 2023.
- [2] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, Blockchain technology and sustainable supply chain management, *Int. J. Production Research*, vol. 57, no. 7, pp. 2117-2135, 2019.
- [3] E. Themistocleous, K. Stefanou, and E. Iosif, Blockchain technology in land registry: Cyprus case study, in *Proc. EMCIS 2018*, Springer.
- [4] A. Ekblaw et al., A case study for blockchain in healthcare: MedRec prototype, in *Proc. IEEE Open & Big Data Conf.*, 2016.
- [5] S. Olnes, J. Ubacht, and M. Janssen, Blockchain in government: Benefits and implications, *Gov. Inf. Q.*, vol. 34, no. 3, pp. 355-364, 2017.
- [6] World Bank Group, *Distributed Ledger Technology and Blockchain in Public Financial Management*, Washington, DC, 2021.
- [7] G. Ahmadi-Assalemi et al., Digital twins for precision healthcare, in *Cyber Defence in the Age of AI*, Springer, 2020.
- [8] S. Huckle and M. White, Socialism and the blockchain, *Future Internet*, vol. 8, no. 4, p. 49, 2016.
- [9] J. Benet, IPFS - Content Addressed, Versioned, P2P File System, arXiv:1407.3561, 2014.
- [10] N. Nizamuddin et al., Decentralized document version control using Ethereum blockchain and IPFS, *Comput. Electr. Eng.*, vol. 76, pp. 183-197, 2019.
- [11] A. Kosba et al., Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in *Proc. IEEE S&P 2016*, pp. 839-858.
- [12] S. Ellis, A. Juels, and S. Nazarov, ChainLink: A decentralized oracle network, *Chainlink White Paper*, 2017.
- [13] Ethereum Foundation, *Ethereum Proof-of-Stake Consensus Specification*, 2022.
- [14] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, *Bitcoin.org White Paper*, 2008.
- [15] Polygon Technology, *Polygon: A Protocol and Framework for Building Ethereum-Compatible Networks*, White Paper, 2021.