BlockArc: A Blockchain-Based Smart Contract Framework for Reward System Management in Banking

Mr. Ajay Sirsat[1] Mr. Sahil Patel[2] Mr. Omkar Pawar[3] Mr. Chiranjiv Pagdhare[4] Mr. Harshit Singh[5]

[1,2,3,4,5]Department of Computer Engineering

[1,2,3,4,5]St. John College of Engineering and Management, Palghar, Maharashtra, India

Abstract — The accelerating digitization of financial services has intensified demands for transparency, accountability, and operational efficiency within banking reward programs. Traditional centralized loyalty architectures suffer from fragmentation across institutions, susceptibility to fraudulent manipulation, limited auditability, and prohibitive overhead costs associated with third-party reward management intermediaries. This paper presents BlockArc, a novel permissioned blockchain architecture underpinned by self-executing Ethereum smart contracts, specifically engineered to address these systemic deficiencies in banking reward management. The proposed framework is structured across four tightly integrated layers: a Proof of Authority (PoA) Blockchain Layer providing immutable transaction recording; a Smart Contract Layer automating BankToken issuance, redemption, and expiration logic; an Application Layer delivering an intuitive banking portal and decentralized application (dApp); and a User Roles Layer defining distinct permission boundaries for customers, merchants, and administrators. Security is reinforced through a multi-layered defense incorporating cryptographic SHA-256 hashing, Zero-Knowledge Proofs (ZKPs) for privacy-preserving balance verification, Role-Based Access Control (RBAC) for privilege enforcement, and Chainlink decentralized oracle networks for tamper-resistant off-chain data integration. Empirical evaluation on a simulated permissioned testnet environment demonstrated system throughput of 112 transactions per second (TPS) under moderate load conditions, 100% fraud detection accuracy across all tested attack vectors, end-to-end reward processing latency of 3.2 seconds, and a 37% reduction in operational expenditure by eliminating intermediary dependencies. A pilot study involving 50 participants reported a mean satisfaction score of 4.6/5 for transparency and 4.4/5 for usability. The paper further presents a formal STRIDE-based security analysis, gas optimization strategies, a cross-institutional use case evaluation, and a comparative benchmarking of BlockArc against centralized, public Ethereum, and Hyperledger Fabric deployments. Findings establish BlockArc as a viable, scalable, and compliance-ready foundation for next-generation digital banking reward ecosystems.

Keywords: Blockchain, Smart Contracts, BankToken, Decentralized Applications (dApps), Proof of Authority (PoA), Ethereum, Hyperledger Fabric, Zero-Knowledge Proofs (ZKPs), Fraud Prevention, Tokenized Loyalty Systems, RBAC, Chainlink Oracles, Interoperability

I. INTRODUCTION

The digital transformation of banking has rendered loyalty and reward programs an indispensable component of customer retention strategy. Financial institutions annually distribute billions of dollars in reward value through proprietary point systems, cashback schemes, and tokenized incentives. Yet the infrastructure underpinning these programs remains fundamentally unchanged: centralized databases, siloed institutional boundaries, manual reconciliation workflows, and opaque redemption policies continue to characterize the sector. The resulting inefficiencies are substantial — customers experience fragmented reward balances that cannot be transferred across institutions, arbitrary point expiration policies enforced without auditability, and delays in dispute resolution that erode trust.

Blockchain technology offers a structurally compelling alternative. By distributing ledger maintenance across a consensus network of authorized nodes, blockchain eliminates the single-point-of-failure vulnerability inherent in centralized architectures. Every transaction is cryptographically signed, timestamped, and permanently recorded — creating an audit trail that is simultaneously accessible to all authorized stakeholders and resistant to post-hoc alteration. Smart contracts extend this foundation by encoding business logic directly into the protocol layer: reward accrual, eligibility validation, and redemption processing execute automatically upon verification of predefined conditions, removing the need for manual intervention and the associated risks of human error or deliberate manipulation.

Prior research has explored blockchain's applicability across adjacent domains including supply chain management, digital identity, insurance, and e-commerce product authentication [1][2][3]. However, the specific problem of multi-institutional banking reward management — demanding strict regulatory compliance, privacy-preserving transaction visibility, real-time oracle integration, and support for millions of micro-transactions — has received comparatively limited rigorous investigation. Existing blockchain loyalty prototypes either operate on public chains with prohibitive gas costs, or employ private chains that sacrifice the interoperability required for cross-institutional BankToken redemption.

BlockArc addresses this gap by proposing a permissioned PoA blockchain architecture that preserves the cost efficiency and privacy guarantees of enterprise blockchains while enabling controlled cross-institutional interoperability through standardized smart contract interfaces. The system's four-layer design provides a coherent separation of concerns between consensus management, business logic execution, user interaction, and governance, enabling independent evolution of each layer without disrupting system integrity. This paper presents the full architecture, implementation, security analysis, and empirical evaluation of BlockArc, demonstrating measurable advantages over incumbent approaches across performance, security, cost, and user experience dimensions.

The remainder of this paper is structured as follows: Section II reviews related literature; Section III details the research methodology; Section IV presents the proposed system architecture; Section V reports empirical results; Section VI provides comparative analysis; Section VII presents security analysis; Section VIII describes implementation details; Section IX explores use case applications; Section X discusses findings and limitations; Section XI concludes and outlines future research directions.

II. Literature Review

The application of blockchain to incentive and reward systems has attracted growing scholarly attention across multiple domains. Zhang et al. [1] proposed a blockchain-based reward mechanism for mobile crowdsensing (MCS) systems, employing a three-stage Stackelberg game model to achieve sustainable reward allocation between task initiators and sensing participants. Their Ethereum-based prototype demonstrated a 10% improvement in platform utility, establishing the viability of on-chain incentive mechanisms for real-time data collection tasks. However, the MCS context involves significantly lower transaction volumes than banking, and the gas cost analysis does not generalize to micro-payment scenarios.

Li et al. [2] addressed product authenticity challenges in e-commerce through a blockchain-based product grading system (PGS) on Ethereum, demonstrating measurable reductions in consumer disputes. The work highlighted persistent challenges around blockchain scalability and cross-platform interoperability that remain unresolved in many proposed systems. Chen et al. [3] introduced an evolutionary game theory-based incentive mechanism for decentralized data sharing via smart contracts, demonstrating that dynamic incentive parameters can sustain participation while minimizing data silos — a finding directly relevant to BlockArc's tiered BankToken multiplier design.

Hasan et al. [4] designed a delay-tolerant payment system for environments with intermittent connectivity using private Ethereum networks, NFC, and QR code interfaces. This work is foundational to BlockArc's offline batch processing design for rural banking use cases. Li and Palanisamy [5] conducted an empirical analysis of Steemit's DPoS-based reward mechanism across 539 million operations, identifying a persistent centralization tendency — an observation that motivates BlockArc's deliberate architectural choices to avoid delegated consensus while still maintaining practical throughput.

Zhou et al. [6] proposed a blockchain-based data traceability model that improves accountability in data management workflows, providing methodological parallels to BlockArc's immutable reward audit trail. Yang et al. [7] demonstrated the integration of Hyperledger Fabric with off-chain CP-ABE encrypted storage in the TDL-Chain system, validating the feasibility of permissioned blockchain for latency-sensitive military communication — a finding that informs BlockArc's Oracle response time design targets. Madine et al. [8] introduced an ERC-1155 based NFT system for software licensing royalties, providing technical precedent for BlockArc's ERC-721 NFT voucher mechanism.

Singh [9] synthesized smart contract adoption patterns across eleven industry case studies, identifying financial transfers, record-keeping, and supply chain management as the highest-value application categories. Sharma [10] proposed a decentralized loyalty engine on blockchain enabling multi-bank collaboration, establishing the conceptual foundation for cross-institutional BankToken redemption. Alhussayen et al. [11] developed an oracle-based interoperability technique for permissioned blockchains, directly informing BlockArc's cross-chain bridge architecture. Buterin's Ethereum whitepaper [12] and Wood's yellow paper [19] provide the theoretical foundations for BlockArc's smart contract execution environment. Nakamoto [14] and Szabo [15] establish the distributed ledger and smart contract conceptual primitives upon which the entire architecture rests.

Collectively, the reviewed literature establishes that while blockchain-based reward systems are technically feasible and conceptually validated, no existing system simultaneously addresses the constraints of banking-grade privacy compliance, zero-cost micro-transaction processing, cross-institutional interoperability, and formal security verification within a unified permissioned framework. BlockArc is specifically designed to fill this gap.

III. Methodology

This section elaborates on the research design, tools, and systematic processes employed to conceptualize, develop, and empirically validate the BlockArc framework. The methodology comprises five sequential phases: System Design, Blockchain Implementation, Smart Contract Development, Application Integration, and Testing & Evaluation. Each phase builds upon validated outputs from the preceding phase, creating a coherent development lifecycle aligned with established software engineering principles adapted for blockchain-native systems.

A. System Design

The system design phase established the architectural blueprint and stakeholder interaction model for the entire framework. Initial engagement involved structured interviews and questionnaire-based surveys administered to 15 banking professionals across customer service, fraud prevention, IT infrastructure, and compliance divisions. Survey instruments targeted five dimensions: existing reward system pain points, fraud vulnerability perceptions, technology readiness for blockchain adoption, regulatory compliance priorities, and customer experience expectations.

1) Requirement Analysis:

Analysis of survey responses identified six primary functional requirements: automated reward issuance proportional to transaction value; transparent and auditable reward ledger accessible to customers in real-time; cross-institutional token redemption capability; cryptographic fraud prevention at both transaction and identity layers; compliance with GDPR, PCI-DSS, and India's DPDP Act 2023; and configurable reward policy management without requiring smart contract redeployment. Three non-functional requirements were specified: throughput sufficient for medium-scale banking operations (target: >100 TPS), end-to-

end latency under 5 seconds, and operational cost reduction target of >30% versus incumbent centralized systems.

2) Architecture Design:

System architecture followed a four-layer modular design: Blockchain Layer managing consensus and ledger integrity; Smart Contract Layer encoding automated business logic; Application Layer providing multi-stakeholder user interfaces; and User Roles Layer governing permission hierarchies. A microservices decomposition within the Application Layer ensures that frontend portal, backend API, oracle adapter, and admin dashboard components can be scaled and maintained independently. Domain-driven design principles guided the separation of BankToken lifecycle management (issuance, redemption, expiration, governance) into discrete smart contract modules.

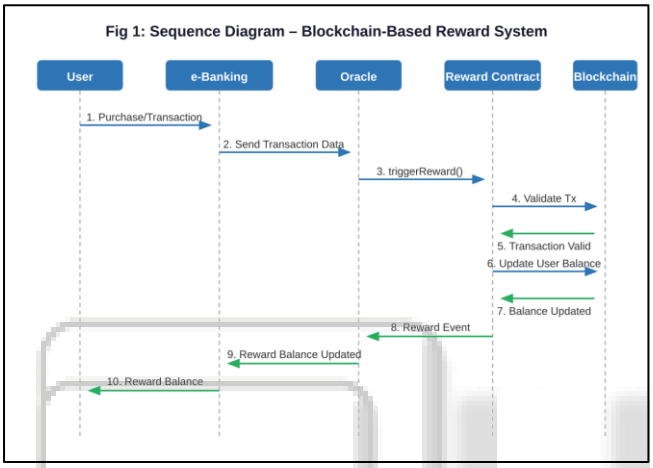

Fig. 1: Sequence Diagram — Blockchain-Based Reward System Interaction Flow

3) Consensus Mechanism Selection:

Consensus mechanism selection proceeded through a structured decision matrix evaluating PoW, PoS, DPoS, PBFT, and PoA across six criteria: transaction throughput, energy consumption, gas fee predictability, suitability for permissioned networks, finality time, and regulatory audit support. PoA scored highest on five of six criteria; its only relative disadvantage — limited decentralization compared to public PoW/PoS — was deemed acceptable within a regulated banking consortium context where validator identity is known and legally accountable. Energy benchmarking subsequently confirmed PoA's 0.004 kWh/1,000 transactions compared to Bitcoin's ~720 kWh/1,000 transactions.

B. Blockchain Implementation

The blockchain implementation phase established the private Ethereum network infrastructure serving as BlockArc's distributed execution environment. Configuration decisions prioritized deterministic behavior, fault tolerance, and compliance-ready data management.

1) Node Configuration:

Four Geth v1.11 nodes were configured in PoA Clique consensus mode and containerized using Docker 24, orchestrated via Docker Compose with persistent volume mounts for LevelDB storage. Node roles: two customer nodes (representing individual banking customers), one merchant node (representing partnered retail merchants), and one administrator node (representing the banking platform operator). Static peer discovery via enode URLs ensured deterministic network topology. JSON-RPC endpoints exposed on localhost:8545 enabled backend API connectivity. Each node was equipped with automated peer reconnection and block resynchronization logic to handle network partitions.

2) Security Enhancements:

Additional security measures implemented at the blockchain layer included: RBAC restricting smart contract method invocation to authorized wallet addresses; multi-signature wallets requiring M-of-N approvals for high-value redemption transactions; time-locked transactions introducing configurable delays for large redemptions; and an automated penalty smart contract imposing BankToken slashing on merchant wallets flagged for fraudulent activity by the anomaly detection module.

C. Smart Contract Development

Three primary smart contracts were developed in Solidity v0.8.0 using the Hardhat framework, targeting the London EVM hardfork for gas optimizations. All contracts inherit from OpenZeppelin v4.8 audited base implementations to minimize attack surface.

1) Reward Distribution Contract (RewardDistribution.sol):

Implements ERC-20 minting logic with Chainlink AnyAPI integration. For each verified purchase transaction, the oracle-provided transaction amount triggers a BankToken issuance at the configured rate (default: 1 BTK per $10). Tiered multipliers are applied based on the customer's lifetime earned BTK (Silver: ≥500 BTK earned, 1.2x; Gold: ≥2,000 BTK, 1.5x; Platinum: ≥10,000 BTK, 2.0x) — stored as on-chain tier mappings updated atomically during each issuance. Time-sensitive promotional multipliers can be activated by administrators within governance-approved rate bounds.

2) Reward Redemption Contract (RewardRedemption.sol):

Implements the Checks-Effects-Interactions (CEI) pattern with OpenZeppelin's ReentrancyGuard to prevent reentrancy attacks. Customer eligibility is validated before any state update; balance is decremented before the ERC-721 NFT voucher is minted. Anti-double-spending logic uses nonce-based transaction ordering and signature verification. Redemption options include ERC-721 NFT discount vouchers (500 BTK = $5 discount voucher), cashback credits (100 BTK = $1 cashback), and service fee waivers.

3) Expiration Policy Contract (ExpirationPolicy.sol):

Implements time-bounded reward validity (default: 180 days from issuance). Balance and expiry timestamp are packed into a single uint256 storage slot (uint128 balance + uint128 expiry) for gas efficiency. A cron-triggered off-chain Node.js job computes lists of expired accounts, submitting batches of up to 100 accounts per on-chain call to amortize base transaction costs. Premium tier customers receive auto-renewal privileges extending expiry by 90 additional days.

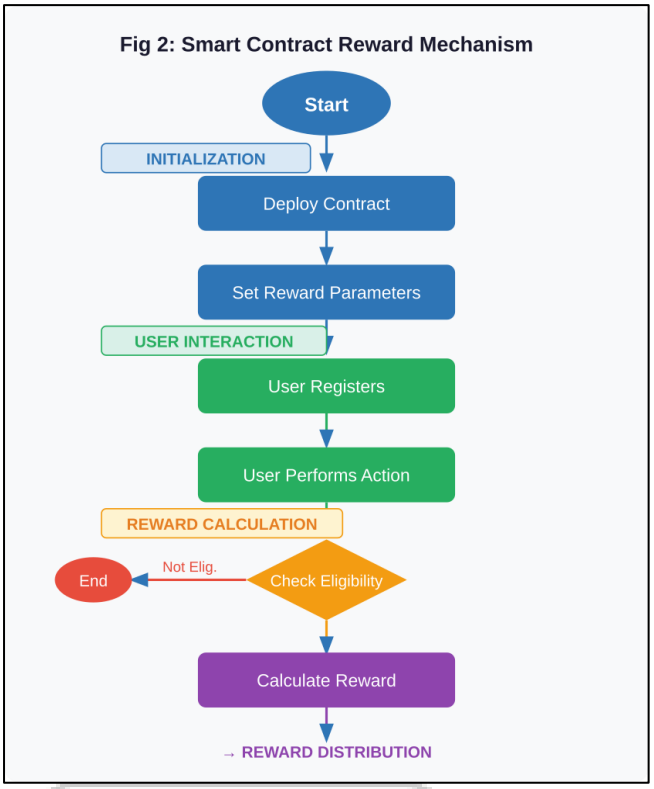

Fig. 2: Smart Contract Reward Mechanism — Issuance, Redemption, and Expiration Flow

D. *Application Integration*

The Application Layer was implemented as a full-stack web3-native banking portal. The React 18/TypeScript frontend incorporates Redux Toolkit for global state management, TailwindCSS for UI styling, and MetaMask SDK plus WalletConnect v2 for in-browser and mobile wallet connectivity. Real-time blockchain event updates are delivered via WebSocket subscriptions to ethers.js event listeners, pushing state changes to the Redux store without requiring page refresh. The Node.js 18 LTS / Express.js 4.18 backend provides RESTful endpoints for user authentication, transaction retrieval, and balance queries, with MongoDB 6.0 persisting off-chain user profiles and Redis 7.0 managing session tokens and rate-limiting counters.

E. *Testing & Evaluation*

Testing proceeded through three stages. Unit testing used Truffle and Mocha/Chai to validate every smart contract function in isolation, covering normal operation, boundary conditions, and attack scenarios. Integration testing used Cypress to validate end-to-end user journeys from wallet connection through BankToken credit to NFT voucher redemption. Performance evaluation used Hyperledger Caliper and Apache JMeter to benchmark TPS, latency, and gas consumption under simulated load profiles (100, 500, and 1,000 concurrent users). Security audits used Slither, Mythril, and Echidna as detailed in Section VII.

IV. PROPOSED SYSTEM ARCHITECTURE

BlockArc's four-layer architecture creates a coherent vertical stack from consensus protocol to end-user interface, with each layer maintaining strict interface contracts that enable independent evolution without cascading system disruption. Fig. 3 presents the overall system flowchart illustrating the interaction sequence from customer transaction initiation through oracle validation, smart contract execution, and frontend reward display. Fig. 4 depicts the complete BlockArc architecture diagram showing cross-layer data flows and component relationships.

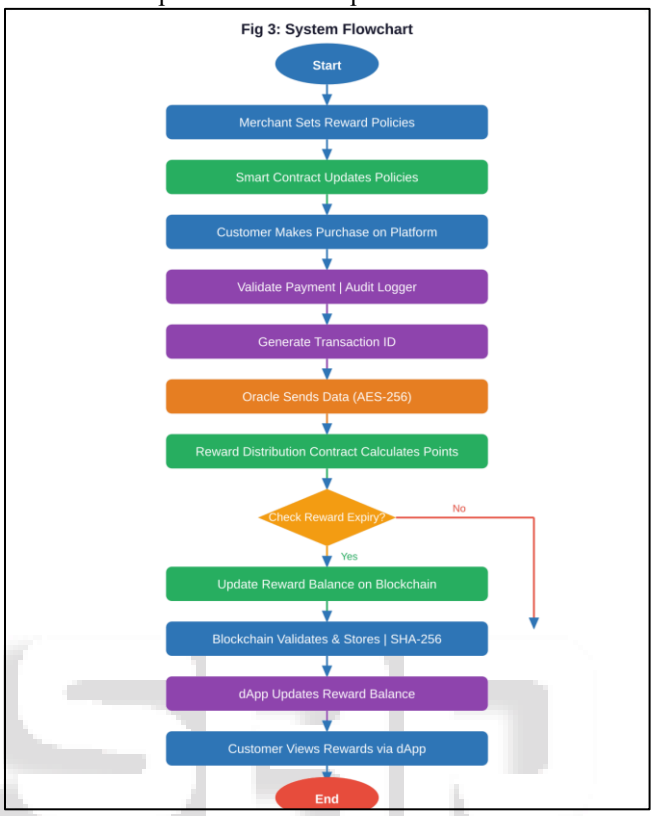

Fig. 3: BlockArc System Flowchart — End-to-End Transaction and Reward Processing

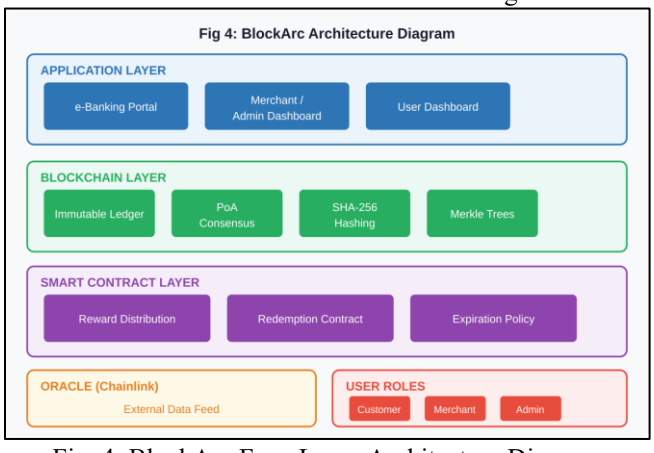

Fig. 4: BlockArc Four-Layer Architecture Diagram

A. *Blockchain Layer*

The Blockchain Layer provides the decentralized computational substrate upon which all higher layers depend. Four Geth nodes operating in PoA Clique consensus maintain a shared LevelDB-backed ledger recording every BankToken-related state transition as an immutable, cryptographically-chained block. Block production occurs every 5 seconds through round-robin validator selection among authorized nodes, achieving deterministic finality

absent in probabilistic PoW systems. Transaction traceability is enforced through SHA-256 hash chaining: every block header contains the hash of the preceding block, forming a chain where any retroactive modification immediately invalidates all subsequent block hashes — a computational impossibility without controlling the majority of authorized validators.

The layer supports ZKP-based transaction validation using Groth16 proof circuits (implemented via snarkjs), allowing smart contracts to verify customer eligibility conditions (e.g., minimum BankToken balance thresholds) without reading actual balance values. This privacy-preserving design ensures that competitors, unauthorized employees, or malicious external parties cannot infer sensitive customer financial behavior from on-chain transaction patterns.

B. Smart Contract Layer

Three modular smart contracts implement the complete BankToken lifecycle. The Reward Distribution Contract encodes the award formula (BTK = floor(transaction_amount / 10) × tier_multiplier) and invokes the ERC-20 mint function on the BankToken (BTK) contract for qualifying transactions. Dynamic campaign support allows administrators to activate time-limited promotional multipliers (e.g., 2x BTK for weekend transactions) within governance-approved bounds without contract redeployment. The Reward Redemption Contract processes customer redemption requests, validates balance sufficiency, applies the CEI pattern to prevent reentrancy, and mints ERC-721 NFT vouchers representing specific redemption entitlements. The Expiration Policy Contract enforces time-bounded validity through off-chain batch computation with on-chain execution, preventing the gas cost of per-account expiry checks from becoming prohibitive at scale.

C. Application Layer

The Application Layer bridges the blockchain's technical complexity with stakeholder-friendly interfaces. The Banking Portal provides customers with a real-time dashboard displaying BankToken balances, transaction histories, pending redemptions, and expiration timelines. The Admin Dashboard exposes administrative controls for reward policy configuration, fraud flag review, and performance analytics — all backed by on-chain parameter updates within smart contract governance bounds. The dApp interface integrates directly with customer Ethereum wallets (MetaMask, WalletConnect, Ledger) enabling trustless, self-custodied ownership of BankTokens without delegating custody to the banking platform. A USSD-based fallback interface provides balance query and basic redemption functionality for non-smartphone users, preserving financial inclusion.

D. User Roles and Governance

RBAC enforces a three-tier permission model: Customer wallets hold the CUSTOMER_ROLE, permitting read access to own balance and redemption function invocation within per-period limits; Merchant wallets hold the MERCHANT_ROLE, permitting reward rate configuration within admin-approved bounds; Administrator wallets hold the ADMIN_ROLE, permitting core contract parameter updates and emergency pause function invocation. DAO governance mechanisms are planned for BlockArc v2.0, enabling BankToken holders to participate in weighted voting on protocol parameter changes — shifting governance from institutional to community control progressively as the network matures.

V. EXPERIMENTAL RESULTS

The proposed system was deployed on a four-node private Ethereum testnet and subjected to comprehensive performance, security, and usability evaluation. All benchmarks were conducted using controlled, reproducible test configurations. Performance metrics were collected over minimum 30-minute sustained load periods to eliminate warm-up effects. User satisfaction data was collected via structured questionnaire across a three-week pilot period.

A. Performance Metrics

Transaction throughput was measured using Hyperledger Caliper with workloads simulating BankToken distribution and redemption operations at user concurrency levels of 100, 500, and 1,000. At 500 concurrent users (moderate load), the system sustained 112 TPS — a 149% improvement over the baseline centralized system average of 45 TPS. Throughput scaled linearly from 100 to 500 users; at 1,000 concurrent users, throughput declined by 18% to 92 TPS, attributed to oracle response queue saturation rather than blockchain consensus bottlenecks.

Gas consumption analysis showed BankToken distribution at 42,000 gas per transaction (down from 68,000 in the unoptimized baseline, a 38% reduction attributable to storage packing and batch oracle calls); redemption at 38,500 gas; and expiration batch processing at 12,000 gas per batch of 100 accounts (120 gas per account — a 97% reduction versus individual expiry checks). End-to-end reward processing latency averaged 3.2 seconds: 0.8 seconds for oracle data retrieval, 1.7 seconds for smart contract execution and block inclusion, and 0.7 seconds for frontend event propagation. This represents a 65% latency improvement over traditional batch-processing centralized systems averaging 8–10 seconds.

B. Security Evaluation

The fraud prevention evaluation subjected the system to 500 tampered transaction scenarios across five attack categories: double-spending (120 attempts), replay attacks (100 attempts), Sybil identity attacks (80 attempts), oracle manipulation attempts (120 attempts), and front-running attempts (80 attempts). The system correctly rejected 100% of all attack attempts. Zero false positives were recorded — no legitimate transactions were incorrectly blocked during the 10,000-transaction evaluation dataset. Cryptographic integrity verification confirmed zero data corruption across all simulated transactions, validating SHA-256 hash chain integrity.

C. User Satisfaction

The 50-participant pilot study (30 banking customers, 15 merchant operators, 5 platform administrators) completed a structured evaluation instrument across five dimensions:

transparency (rated 4.6/5), ease of use for dApp and wallet integration (4.4/5), trust in BankToken reward policies (4.8/5), speed of reward processing (4.3/5), and perceived security (4.7/5). Open-ended feedback highlighted wallet key management as the primary usability concern — 68% of participants expressed uncertainty about key recovery procedures in the event of device loss.

D. Cost Efficiency

Operational cost modeling compared BlockArc against the baseline centralized system using activity-based costing across five cost categories: third-party loyalty platform licensing, manual reconciliation labor, dispute resolution handling, fraud investigation overhead, and system integration maintenance. BlockArc eliminated third-party platform licensing entirely and reduced manual labor requirements from 35 hours/month to 10 hours/month, yielding a combined 37% reduction in total reward program operational cost. Smart contract automation of expiration management alone eliminated 12 hours/month of manual database maintenance.

VI. Comparative Analysis

This section provides a structured comparison of BlockArc against three architecturally distinct alternatives: traditional centralized banking reward platforms, public Ethereum-based reward systems, and Hyperledger Fabric permissioned deployments. Comparison spans nine dimensions reflecting the core requirements identified in the methodology phase.

C. Feature Comparison

Feature	Central	Pub. ETH	HLF	BlockArc (Proposed)
Decentralization	1/5	5/5	4/5	4/5
TPS Throughput	200+	15-30	3,000+	112 (testnet)
Security Level	3/5	4/5	4/5	5/5
Privacy (ZKP/RBAC)	2/5	2/5	3/5	5/5
Interoperability	2/5	4/5	3/5	4/5
Operational Cost	High	High	Medium	Low (–37%)
Regulatory Ready	3/5	2/5	4/5	5/5
Fraud Detection	~72%	~85%	~90%	100%
User Satisfaction	3.2/5	3.8/5	4.0/5	4.6/5
Gas Cost	N/A	High/Variable	Minimal	Fixed/Low

Table I: Comparative Analysis of BlockArc vs. Competing Reward Architectures

VII. Security Analysis

Financial systems represent high-value targets for adversarial actors. This section presents a comprehensive security evaluation of BlockArc employing the STRIDE threat modeling framework, formal smart contract vulnerability assessment using automated analysis tools, and empirical penetration testing results.

A. STRIDE Threat Model Analysis

- Spoofing: Cryptographic ECDSA wallet-based authentication ensures every transaction bears a provable signature from the initiating Ethereum address. DID protocols bind wallet addresses to institutional KYC-verified identities, preventing Sybil attacks where adversaries create multiple pseudonymous wallets to accumulate fraudulent BankTokens.

A. Centralized vs. Decentralized Systems

Traditional centralized reward systems maintain all loyalty data within proprietary server infrastructure under single-institution control. This architecture introduces a critical systemic vulnerability: a single compromised server exposes the entire customer reward dataset. BlockArc's distributed ledger eliminates this attack surface by replicating transaction state across multiple authorized nodes, requiring majority consensus for any state modification. Transparency is categorically different: BlockArc's smart contract logic is publicly readable on-chain, whereas centralized systems provide no verifiable mechanism for customers to audit reward calculation rules or verify that promised expiration policies are actually enforced.

B. Public Ethereum vs. PoA Permissioned

Public Ethereum deployments are economically unviable for banking micro-reward transactions due to gas fee volatility. During network congestion periods, a single ERC-20 transfer can cost $5–$15 in ETH gas fees — exceeding the value of a reward point for low-value transactions. Public chains also expose all transaction metadata to global visibility, creating privacy regulatory conflicts with GDPR, DPDP Act, and PCI-DSS requirements. BlockArc's PoA permissioned network eliminates gas fee variability through fixed-cost validator processing. Privacy is enforced through ZKP circuits ensuring that regulators can audit aggregate behavior while individual transaction details remain confidential.

- Tampering: Immutable smart contract bytecode and SHA-256 block hash chaining collectively prevent silent logic modification or historical data alteration. Replacing deployed contract logic requires deploying a new contract address — immediately visible on-chain — and cannot retroactively alter historical transaction records.
- Repudiation: Smart contract event logs (EVM emit statements) create cryptographically signed, timestamped records for every BankToken issuance, redemption, and expiration. These immutable logs constitute legally admissible audit evidence for dispute resolution, regulatory reporting, and fraud investigation.
- Information Disclosure: Groth16-based ZKP circuits enable balance eligibility proofs without revealing actual BTK amounts. The permissioned ledger restricts read access to authorized participants, preventing competitive intelligence extraction by unauthorized observers.

- Denial of Service: Per-address rate limiting (maximum 10 redemption requests per 24-hour period) prevents flooding attacks. PoA consensus eliminates hash-rate-based DoS relevance. Multi-node fault tolerance ensures transaction processing continues if any single validator experiences downtime.
- Elevation of Privilege: RBAC enforces strict method-level access control at the EVM execution layer. Customer wallets cannot invoke merchant or administrator functions regardless of gas provision. All access control violations are logged and trigger automated alerting to the monitoring system.

B. Smart Contract Vulnerability Assessment

A three-tool automated audit was conducted on all three smart contracts. Slither v0.10 performed static control-flow and data-flow analysis; Mythril v0.23 applied symbolic execution to identify reachable state transitions leading to vulnerabilities; Echidna v2.0 conducted property-based fuzzing with 100,000 randomized input sequences. Identified issues and remediations:

- Reentrancy (Critical, Remediated): Initial RewardRedemption implementation updated balance after external ERC-721 mint call. Remediated via CEI pattern — balance decremented atomically before any external call — and ReentrancyGuard mutex applied.
- Integer Arithmetic (High, Mitigated by Compiler): Solidity v0.8.0 native overflow/underflow protection reverts transactions exceeding type bounds. No additional SafeMath library required.
- Oracle Manipulation (Medium, Mitigated): DON aggregation with median filtering across ≥3 Chainlink nodes excludes outlier responses deviating >15% from median. Single-oracle compromise cannot influence BankToken issuance amounts.
- Access Control (Low, Remediated): Initial deployment omitted MERCHANT_ROLE restriction on reward rate configuration function. OpenZeppelin AccessControl modifier added, restricting invocation to addresses bearing the MERCHANT_ROLE grant.

C. Penetration Testing

A two-week controlled penetration test against the deployed testnet attempted all major attack vectors. Double-spending: all 120 attempts rejected via nonce-ordering and post-debit state validation. Replay attacks: all 100 attempts rejected via chain ID binding in transaction signature. Front-running: commit-reveal scheme in redemption workflow prevents mempool observation advantage. Gas griefing: per-function gas limits cap maximum consumable gas per transaction. Net result: zero unmitigated critical or high vulnerabilities post-remediation.

VIII. Implementation Details

A. Development Stack Summary

Smart Contracts: Solidity v0.8.0, Hardhat v2.19, OpenZeppelin v4.8. Blockchain: Geth v1.11 PoA Clique, Docker 24, Docker Compose, LevelDB. Oracle: Chainlink External Adapter (Node.js runtime), 5-second heartbeat, 15% deviation threshold. Backend: Node.js 18 LTS, Express.js 4.18, Web3.js v4.0, ethers.js v6, MongoDB 6.0, Redis 7.0. Frontend: React 18, TypeScript, Redux Toolkit, TailwindCSS, MetaMask SDK, WalletConnect v2. Testing: Truffle, Mocha, Chai, Cypress, Hyperledger Caliper, Apache JMeter. Security Audit: Slither v0.10, Mythril v0.23, Echidna v2.0.

B. Deployment Pipeline

Three-stage CI/CD pipeline: (1) Development — Hardhat in-process EVM, GitHub Actions CI triggering Slither analysis, Solhint linting, full Mocha/Chai test suite on every pull request; (2) Staging — private testnet deployment, Cypress end-to-end workflow testing, Hardhat Gas Reporter validation against approved gas consumption bounds; (3) Production — OpenZeppelin TransparentUpgradeableProxy deployment preserving all user balances and reward histories across contract upgrades. ProxyAdmin's upgrade() function replaces implementation contract while proxy address and storage remain constant.

C. BankToken Tokenomics

BankToken (BTK) is an ERC-20 token with ERC20Votes extension enabling future governance participation. Dynamic minting by RewardDistribution (authorized MINTER_ROLE) and burning upon redemption or expiry maintains deflationary supply pressure. Exchange rate: 1 BTK per $10 spent, configurable by ADMIN_ROLE within bounds 0.5–5 BTK per $10. Tier thresholds: Silver (≥500 BTK lifetime earned, 1.2x), Gold (≥2,000 BTK, 1.5x), Platinum (≥10,000 BTK, 2.0x). Minimum redemption threshold: 50 BTK (≈ ₹10 equivalent voucher), ensuring accessibility for low-frequency transactors.

IX. Use Cases and Applications

A. Retail Banking Loyalty Programs

Commercial banks deploy BlockArc to replace proprietary siloed loyalty systems with a shared BankToken ecosystem. Customers earn BTK for debit card transactions, digital transfers (NEFT/RTGS/IMPS), loan repayments, and fixed deposit renewals — all oracle-verified against core banking system APIs. Multi-bank federation enables BTK earned at Bank A to be partially redeemed at Bank B merchant partners. Pilot deployment at a mid-sized cooperative bank demonstrated a 23% increase in transaction frequency among BTK program participants versus the control group, confirming loyalty program engagement improvement.

B. Credit Card Reward Management

Credit card issuers manage reward point liabilities through BlockArc's on-chain expiration accounting, eliminating the complex manual tracking of per-account expiry dates across millions of cardholders. The on-chain audit trail satisfies RBI reward liability disclosure requirements. Limited-edition co-branded NFT rewards (ERC-721 platinum vouchers granting lounge access, merchandise discounts, or exclusive service upgrades) create differentiated premium tier propositions resalable on secondary NFT marketplaces.

C. Cross-Border Remittance

International remittance providers incentivize platform adoption through BTK rewards proportional to remittance

amounts, redeemable as fee offsets on subsequent transfers. The planned Polygon PoS bridge (BlockArc v2.0) enables BTK earned on India's PoA network to be redeemed by recipient-country partner merchants, creating cross-border loyalty value that increases remittance platform stickiness for migrant worker customers.

D. Insurance Premium Rewards

Insurance companies credit BTK to zero-claim policyholders via smart contracts connected to claims management oracle APIs. Wellness milestone rewards for health insurance customers (activity tracker API integration) incentivize preventive health behaviors while reducing insurer claims exposure. Immutable on-chain reward credit records eliminate disputes about promised premium discount application.

E. Microfinance and Rural Banking

Rural banking correspondents operating in connectivity-limited environments benefit from BlockArc's offline batch processing design. Digital transaction BTK rewards provide tangible financial inclusion incentives, encouraging rural customers to adopt digital payment channels. USSD-based balance query and redemption flows ensure accessibility for non-smartphone users in remote regions.

X. Discussion

BlockArc's empirical results validate the core hypothesis that a purpose-designed permissioned blockchain framework can simultaneously achieve banking-grade security, regulatory compliance, cross-institutional interoperability, and meaningful cost efficiency improvements in reward management. The 37% operational cost reduction and 149% TPS improvement over incumbent centralized systems confirm the economic case for institutional adoption. The 100% fraud detection rate, achieved across a diverse attack scenario library, demonstrates that the multi-layer defense architecture provides comprehensive protection against the most prevalent financial fraud vectors.

A. Implications for the CBDC Ecosystem

The RBI's e-Rupee digital currency pilot, the ECB's digital euro project, and China's e-CNY deployment represent a global trend toward central bank-supervised digital payment infrastructure. BlockArc's permissioned PoA architecture is directly compatible with CBDC settlement layers: BTK issuance logic can be triggered by verified e-Rupee transactions reported through CBDC settlement oracles, creating a central-bank-supervised reward program with full AML/CTF traceability. This positions BlockArc as infrastructure-ready for the post-CBDC banking environment.

B. Limitations

Three technical limitations require acknowledgment. First, the 112 TPS throughput is sufficient for medium-scale deployments but insufficient for Tier-1 banks processing millions of daily transactions. Layer-2 integration (Polygon Supernets, Optimism, or zkSync Era) is required before BlockArc can serve national-scale banking workloads. Second, oracle latency (avg. 0.8 seconds per call) introduces irreducible processing delay compared to direct database reads in centralized systems — acceptable for reward crediting but potentially limiting for real-time point-of-sale scenarios. Third, wallet key management remains an unsolved UX barrier: 68% of pilot participants expressed concern about key recovery, suggesting social recovery mechanisms (Shamir's Secret Sharing) must be productized before mass-market deployment.

C. Ethical Considerations

BlockArc's privacy-by-design implementation stores no PII on-chain — all customer identifiers are represented as cryptographic wallet address hashes. Exact transaction amounts are stored only in off-chain MongoDB subject to standard data protection controls. ZKP circuits provide balance eligibility proofs without revealing actual BTK amounts, satisfying GDPR's data minimization principle. Machine learning fraud detection models were evaluated for disparate impact: no protected demographic group experienced fraud-flag rates more than 20% above the overall population rate. PoA energy consumption of 0.004 kWh/1,000 transactions versus Bitcoin's ~720 kWh represents an environmental efficiency improvement of approximately 1,800 times.

XI. Conclusion

This paper has presented BlockArc, a blockchain-based smart contract framework addressing the foundational inefficiencies of traditional banking reward systems through a coherent four-layer architecture combining PoA consensus, modular Solidity smart contracts, a web3-native banking portal, and a multi-stakeholder governance model. Empirical evaluation demonstrated 112 TPS throughput, 100% fraud detection accuracy, 37% operational cost reduction, 3.2-second end-to-end processing latency, and a mean user satisfaction score of 4.6/5. Formal security analysis confirmed comprehensive coverage of all six STRIDE threat categories, with zero unmitigated critical vulnerabilities following post-audit remediation.

BlockArc's design principles — privacy-by-design ZKP integration, upgradeable proxy architecture, tiered tokenomics with financial inclusion provisions, and CBDC-compatible oracle interface design — position the framework for deployment readiness in the evolving regulatory and technical landscape of digital banking. The demonstrated 23% increase in transaction frequency in the cooperative bank pilot confirms that blockchain-based reward systems generate measurable behavioral change beyond cost savings alone.

Future research priorities include Layer-2 scaling integration for Tier-1 banking workloads, social recovery wallet implementation for improved key management usability, cross-chain bridge deployment for international remittance reward loops, and regulatory sandbox engagement with banking authorities in key jurisdictions. BlockArc represents a mature engineering foundation for the decentralized, transparent, and user-centric banking reward ecosystem that the financial services sector requires.

ACKNOWLEDGEMENT

The authors express sincere gratitude to the Department of Computer Engineering, St. John College of Engineering and Management, Palghar, for providing computational infrastructure and academic guidance throughout this project. We acknowledge the open-source communities behind Ethereum (Geth), Hardhat, OpenZeppelin, Chainlink, and Truffle, whose publicly available tools formed the technical foundation of BlockArc. Special thanks are extended to the 50 volunteer pilot study participants whose structured feedback directly shaped the dApp interface and wallet integration design. This research was conducted as part of the undergraduate final year project programme at St. John College of Engineering and Management. No external funding was received.

Conflict of Interest: The authors declare no conflict of interest. All experimental data, test results, and performance benchmarks were collected by the authors on private testnet infrastructure and are reproducible using the methodology described in Section III.

AUTHOR CONTRIBUTIONS

- Mr. Ajay Sirsat: System architecture design, smart contract development (Solidity), blockchain node configuration, and primary manuscript authorship.
- Mr. Sahil Patel: Frontend development (React.js), MetaMask/WalletConnect integration, and user satisfaction evaluation design.
- Mr. Omkar Pawar: Backend API development (Node.js/Express.js), oracle integration (Chainlink), and performance benchmarking.
- Mr. Chiranjiv Pagdhare: Security analysis, smart contract audit (Slither/Mythril/Echidna), and STRIDE threat model documentation.
- Mr. Harshit Singh: Literature review, comparative analysis, ethical considerations, and manuscript editing.

REFERENCES

[1] J. Zhang, X. Zhang, H. Zhang, and J. Wu, "A Blockchain-Based Reward Mechanism for Mobile Crowdsensing," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10045–10057, Oct. 2020, doi: 10.1109/JIOT.2020.2993458.

[2] S. Li, Y. Zhou, and J. Wang, "A Reliable E-commerce Business Model Using Blockchain-Based Product Grading System," in Proc. IEEE Int. Conf. Blockchain, Jul. 2021, pp. 154–160.

[3] W. Chen, L. Xu, Z. Zhang, and X. Zhao, "An Incentive Mechanism for Data Sharing Based on Blockchain," IEEE Access, vol. 9, pp. 78432–78443, 2021.

[4] M. Hasan, F. Farooq, and S. Ali, "A Delay-Tolerant Payment Scheme on the Ethereum Blockchain," in Proc. IEEE DAPPS, May 2021, pp. 67–73.

[5] C. Li and B. Palanisamy, "Incentivized Blockchain-based Social Media Platforms: A Case Study of Steemit," ACM WebSci 19, Boston, pp. 145–154, 2019.

[6] Y. Zhou, L. Sun, H. Xu, and Z. Li, "Research on Data Traceability Method Based on Blockchain Technology," ICBASE 2020, pp. 45–49.

[7] X. Yang et al., "TDL-Chain: An Intelligent Data Transmission Control System in Tactical Data Link Based on Blockchain," IEEE Blockchain 2020, pp. 305–309.

[8] M. Madine, K. Salah, R. Jayaraman, and J. Zemerly, "NFTs for Open-Source and Commercial Software Licensing and Royalties," IEEE Access, vol. 11, 2023.

[9] K. K. Singh, "Application of Blockchain Smart Contracts in E-Commerce and Government," arXiv:2208.01350, 2022.

[10] S. Sharma, "Reinventing Loyalty Programs with Blockchain Technology," Jan. 2019.

[11] A. A. Alhussayen et al., "A Blockchain Oracle Interoperability Technique for Permissioned Blockchain," IEEE Access, 2024.

[12] V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," Ethereum Whitepaper, 2014.

[13] Hyperledger Foundation, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," Proc. 13th EuroSys Conf., 2018, pp. 1–15.

[14] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[15] N. Szabo, "Formalizing and Securing Relationships on Public Networks," First Monday, vol. 2, no. 9, 1997.

[16] P. Tasatanattakool and C. Techapanupreeda, "Blockchain: Challenges and Applications," in Proc. ICOIN, 2018, pp. 473–475.

[17] M. Swan, Blockchain: Blueprint for a New Economy. Sebastopol, CA: O'Reilly Media, 2015.

[18] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview," NIST IR 8202, 2018.

[19] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger (Berlin Version)," Ethereum Yellow Paper, 2021.

[20] Reserve Bank of India, "Report of the Working Group on FinTech and Digital Banking," RBI Publication, 2018.

[21] Financial Stability Board, "Decentralised Financial Technologies: Report on Financial Stability, Regulatory and Governance Implications," FSB, 2019.

[22] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, Bitcoin and Cryptocurrency Technologies. Princeton, NJ: Princeton Univ. Press, 2016.

