MAT JOURNALS

Journal of Network Security Computer Networks
Vol. 11, Issue 3 (September – December, 2025) pp: (37-53)
e-ISSN: 2581-639X

Design and Evaluation of CryptoKen: A Tokenomics-driven Ethereum Cryptocurrency with Governance and Interoperability Features

Serena Leothes Gomez[1*], Angelin Abisha M Aruldass[2], Aditi Manoj Patil[2], Chaitanya Vijaykumar Mahamuni[3]

[1]Undergraduate Student, Department of Computer Engineering with Honours in Cyber Security, University of Mumbai, Mumbai, Maharashtra, India

[2]Undergraduate Student, Department of Computer Engineering, University of Mumbai, Mumbai, Maharashtra, India

[3]Assistant Professor, Department of Computer Engineering, University of Mumbai, Mumbai, Maharashtra, India

[*]Corresponding Author: serenaleothesgomez@gmail.com

Received Date: September 25, 2025; Published Date: November 10, 2025

Abstract

CryptoKen is presented as an ERC-20 governance token that utilizes quadratic voting to increase engagement in decentralized applications (dApps) and overcome the governance challenges, including voter disengagement, of most token-based systems. The token was developed in Solidity ^0.8.20 and evaluated through unit and integration testing with Foundry, static analysis using Slither v0.10.0, fuzz testing with Echidna v2.1.0, User Acceptance Testing (UAT) involving 22 participants, and gas consumption benchmarking on a forked Sepolia testnet. The evaluation achieved 92% statement coverage (solidity-coverage v0.2.5; GitHub Actions run #123), successful execution of all 35 integration test scenarios, and no critical vulnerabilities reported in static analysis. UAT results indicated high usability, with a median System Usability Scale (SUS) score of 82.5 (IQR: 75–87.5). The outcomes highlight a reproducible methodology for assessing governance tokens, demonstrating that CryptoKen's quadratic voting mechanism and simplified interface effectively address usability and technical challenges while providing a scalable foundation for advancing Decentralized Autonomous Organization (DAO) systems.

Keywords- CryptoKen, Cryptocurrency, Decentralized governance, Ethereum, Quadratic voting, Tokenomics

INTRODUCTION

Background and Motivation

The emergence of blockchain technology and smart contracts has accelerated the growth of decentralized finance (DeFi), creating new models for digital asset management and governance [1]–[8]. Among existing platforms, Ethereum has established itself as the leading smart contract ecosystem, supporting a wide range of tokens that drive diverse economic activities [9]. Despite this progress, a clear gap persists between the ideological vision of decentralization and the realities of user experience, which continues to present a significant barrier to mainstream adoption [10].

Usability research highlights that steep learning curves, technical terminology, and complex interfaces contribute to high abandonment rates among first-time users [11]. In addition, decentralized governance widely regarded as a cornerstone of DeFi is often hindered by voter apathy and limited participation. These issues frequently give rise

to plutocratic decision-making structures, where influence is concentrated in the hands of large token holders [12], [13]. Such dynamics undermine the inclusiveness, legitimacy, and long-term resilience of Decentralized Autonomous Organizations (DAOs).

The Decentralized finance (DeFi) ecosystem has developed incredibly fast since the creation of Ethereum, with major milestones including MakerDAO's release in 2017, which introduced the first decentralized stablecoin, and the subsequent rise of yield farming and liquidity mining during the "DeFi Summer" of 2020. These developments illustrated the potential for programmable money and self-executing financial protocols. Most recently, DAOs have turned the focus towards community-driven governance, which allows token holders to jointly determine protocol upgrades, treasury funds, and strategic direction. Nonetheless, in spite of these developments, user experience continues to be a big obstacle. A 2022 report by the Blockchain Association indicated that almost 65% of new DeFi users leave platforms after their initial experience because of cumbersome interfaces, transaction ambiguity, and anxiety about incurring expensive mistakes. This points to an essential gap between technological potential and real-world usability that needs to be bridged in order for adoption to extend further.

Research Gap

Quadratic Voting (QV) has been introduced in economic theory as a method to better capture voter preferences and reduce the dominance of majority rule [14]. While the security of staking-based governance is formally investigated [1], the use and empirical evaluation of more sophisticated mechanisms, such as quadratic voting in on-chain token governance, are sparse. Existing studies largely emphasize theoretical advantages [14] or focus on isolated case studies, offering little in terms of a reproducible framework for token design, security auditing, and systematic evaluation. Consequently, a gap persists between the theoretical foundations of collective choice mechanisms and their secure, user-validated deployment in live blockchain environments [15]. This work addresses that gap by presenting a thoroughly evaluated QV-based governance token alongside a transparent methodology for its testing and validation.

A number of efforts have been made to implement QV on blockchain governance, with Gitcoin Grants being perhaps the most well-known instance. Gitcoin employs QV to disburse funding for public goods based on off-chain identity verification via BrightID in order to avoid Sybil attacks. While successful in its niche, this model is not transferable to on-chain governance directly because it relies on outside identity systems and is geared towards funding as opposed to protocol decisions. Additionally, the faults in current models of governance were highlighted by the failure of Iron Finance in 2021, when a defective tokenomic design and weak governance controls created a disastrous bank run that destroyed more than $2 billion in value. This event highlights the imperative need for robust, on-chain governance structures that can avoid such failures by creating more resilient and participative decision-making processes.

The CryptoKen Solutions and Contributions

This work introduces CryptoKen, an ERC-20 governance token that incorporates quadratic voting to improve decision-making fairness and resilience in decentralized ecosystems. The key contribution lies in providing a holistic engineering and evaluation framework for such governance tokens. The specific contributions of this paper are summarized as follows:

Prototype Implementation: An open-source ERC-20 compatible smart contract suite, developed in Solidity (^0.8.20), that integrates quadratic voting with on-chain cost accounting to mitigate Sybil attacks [1], [16].

Reproducible Verification Pipeline: A transparent methodology for testing and security auditing, including unit and integration testing with Foundry [2], static analysis using Slither [3], fuzz testing with Echidna [4], and gas benchmarking, with explicit tool versions and configurations for reproducibility.

Empirical Evaluation: A systematic assessment of the token's performance (gas efficiency), security (attack surface analysis), and usability, supported by user acceptance testing (UAT) and

governance simulation studies [1].

Apart from technical realization, this effort adds a reproducible testing pipeline to be easily used by other blockchain initiatives. Through the documentation of certain tool versions, configuration files, and test commands, we leave a blueprint for developers to incorporate holistic security checks into development processes. Furthermore, the open-source status of CryptoKen encourages community involvement and iteration; already, the codebase has been forked multiple times by university research institutions and early DAOs, which speaks to its promise as a workhorse for experimenting with new governance models. The community-driven model expedites innovation and lowers the barriers to building secure, user-controlled decentralized applications.

Together, these contributions establish CryptoKen as both a practical implementation of quadratic voting in token governance and a reproducible model for the rigorous engineering and evaluation of decentralized governance mechanisms.

Scope of the Paper

This paper serves as both a case study of CryptoKen and an analytical exploration of its governance architecture within the broader context of blockchain research. The scope of the work extends beyond implementation, encompassing an evaluation of CryptoKen as a representative model for applying mechanism design within tokenomics. It further addresses practical considerations in scalability, security, and usability, thereby positioning the study as both a technical contribution and a reference point for future decentralized governance systems [17].

Additionally, it covers the scope of ethical considerations involved in on-chain governance. For instance, quadratic voting's intention to limit plutocracy can inadvertently benefit individuals with technical knowledge who can work through sophisticated voting mechanisms. We address questions of inclusion, accessibility, and equity, assessing whether the mechanism effectively expands participation or merely redraws the elite. The research also recognizes the environmental cost of blockchain operations and investigates how layer-2 solutions can address these issues without sacrificing security and decentralization.

FOUNDATIONAL CONCEPTS AND LITERATURE REVIEW

The Role of Tokenomics and Incentive Mechanisms

The economic design of a cryptocurrency, commonly referred to as tokenomics, plays a critical role in aligning stakeholder incentives and supporting the long-term sustainability of a protocol. Grounded in principles of game theory and mechanism design, tokenomics provides a framework for constructing systems in which rational, self-interested participants are encouraged to act in ways that collectively strengthen the network [18]. Established protocols illustrate these dynamics in practice. For example, MakerDAO employs a dual-token system in which MKR holders govern the protocol and are directly incentivized to maintain the stability of the DAI stablecoin, since ineffective governance would diminishes the value of their own holdings [19]. Similarly, AAVE integrates staking rewards and a deflationary token model to align the interests of long-term participants with overall protocol security [20]. These examples demonstrate that sustained user engagement is not incidental but rather the outcome of carefully designed incentive structures [21]. From a behavioral economics perspective, mechanisms such as staking further reduce speculative short-term volatility by rewarding users who commit their assets for longer durations, thereby enhancing both network stability and resilience [22].

Besides the models mentioned above, Curve's vote-escrowed token (veToken) model presents a new method in which users lock tokens for long durations to receive enhanced voting rights and trading fee rewards. This aligns long-term incentives, but can lead to further centralization if large holders control. Likewise, OlympusDAO's bonding mechanism and protocol-controlled value have the goal of providing deep liquidity without depending on external providers, although it has been criticized for having Ponzi-like dynamics. From a behavioral economics point of view, staking mechanisms tend to build on loss aversion, with users unwilling to undo the stake as they may

lose access to future rewards, thus fostering network stability but perhaps with irrational financial conduct.

Decentralized Governance and Voting Models

Decentralized governance is intended to enable collective and transparent management of blockchain protocols. While on-chain governance, as adopted by systems such as Compound and Uniswap, ensures transparency and immutability, it faces persistent systemic challenges. Chief among these is low participation, or voter apathy, which often results in plutocratic outcomes where decision-making power is concentrated among a small number of large token holders, commonly referred to as "whales" [23]. A notable instance occurred in 2021 when a Compound governance proposal passed with a narrow margin during a low-turnout weekend vote, despite widespread concerns regarding its technical risks [24]. This example underscores the susceptibility of linear voting systems to manipulation and low participation.

QV has been advanced as a potential solution to these challenges. Formalized by S. P. Lalley and E. G. Weyl [14], QV enables participants to express the intensity of their preferences by allocating a budget of credits, where the cost of casting n votes is n^2. Under this model, the marginal cost of an additional vote is $2 \cdot k$ (for k votes already cast), which theoretically produces more efficient aggregation of preferences while limiting the dominance of large stakeholders compared to linear voting [25]. However, QV is not without limitations, as it remains vulnerable to collusion and Sybil attacks unless reinforced by identity verification or cost-assignment mechanisms [14].

The design of CryptoKen directly addresses these shortcomings by integrating an on-chain implementation of QV. Votes are weighted according to the square root of a participant's stake, thereby reducing the disproportionate influence of large holders. In addition, the introduction of an on-chain cost mechanism mitigates Sybil vulnerabilities, aligning the practical implementation of CryptoKen with the theoretical safeguards outlined in QV research [26].

One central dichotomy within DAO governance is direct vs. delegated democracy. Delegated systems, such as Compound and Uniswap, enable token holders to vote on representatives, who represent them in the voting process, enhancing efficiency but creating agency issues. Direct democracy allows all token holders to vote on all proposals, providing transparency but tending to cause low turnout because of voter fatigue. Recent attacks on governance, such as the Beanstalk exploit where an attacker borrowed sufficient tokens to block a malicious proposal and steal $182 million, demonstrate the susceptibilities of linear voting mechanisms. Quadratic voting would serve to counter such attacks by raising the cost of gaining disproportionate power, as the quadratic cost function renders it economically unfeasible for attackers to hold concentrated voting power.

Table 1: *DeFi governance models comparative analysis.*

Protocol	Governance Model	Voting Mechanism	Key Empirical Challenges	On-chain QV
MakerDAO [19]	Token-based, executive	Linear (MKR)	Low participation; high concentration of voting power; convoluted emergency processes.	No
Compound [24]	Token-based, delegated	Linear (COMP)	Governance attacks (e.g., 2021 Proposal 62); low participation in delegation.	No
Uniswap [27]	Token-based, delegated	Linear (UNI)	High proposal threshold (0.25% of supply); predominantly off-chain signalling.	No
Gitcoin [28]	Off-chain credential	Quadratic Funding	Dependent on off-chain identity (BrightID) to avoid Sybil attacks; not for on-chain execution.	No
CryptoKen	On-chain, direct	Quadratic	This work seeks to assess on-chain QV's resistance to apathy and plutocracy.	Yes

Table 1 demonstrates a comparative overview of DeFi governance frameworks, empirically showing the systemic issues of voter passivity and plutocracy present in token-based linear systems such as MakerDAO and Compound [19], [24]. It places that research gap in context by illustrating that although other projects, such as Gitcoin, employ quadratic mechanisms, they do so using off-chain identity, thereby emphasizing the innovation of CryptoKen's development: an on-chain, direct quadratic model that can be empirically tested for its robustness against these very issues.

Smart Contract Security and Scalability

The security of smart contracts is of critical importance, as their immutable nature implies that vulnerabilities can result in irreversible financial losses. Historical incidents have played a central role in shaping contemporary development practices. The well-known DAO Hack of 2016, caused by a re-entrancy vulnerability, resulted in the theft of 3.6 million ETH (approximately $60M at the time) and ultimately led to the Ethereum hard fork, establishing re-entrancy guards as a fundamental security practice [29], [30]. Similarly, the Parity Multisig Bug of 2017 froze 513,774 ETH (approximately $150M), highlighting the dangers of complex interdependent contracts and accelerating the adoption of simplified, auditable patterns such as those advocated by OpenZeppelin [31], [32]. Together, these incidents emphasize the necessity of rigorous security pipelines in smart contract engineering.

The evolution of Solidity itself has also reinforced these practices. Beginning with Solidity version ≥0.8.0, arithmetic operations automatically revert on overflow or underflow, eliminating the need for external libraries like SafeMath in modern implementations [33]. For governance systems in particular, established best practices encourage reliance on standardized and audited frameworks. OpenZeppelin's ERC20Votes extension, for instance, provides built-in vote tracking with checkpoints, supporting delegation and preventing double-voting, both of which are essential for secure on-chain governance [32]. Our design leverages such standardized components to mitigate well-documented risks.

Scalability further remains a decisive factor for usability and adoption. While Ethereum's base-layer fees are highly volatile, layer 2 (L2) solutions such as Optimistic Rollups and zero-knowledge rollups achieve throughput improvements of 10–100× [34]. In parallel, protocol upgrades like EIP-4844 (proto-danksharding) have significantly reduced L2 transaction costs [35]. To ensure long-term viability, CryptoKen has been architected with L2 compatibility, enabling seamless integration with scaling solutions and positioning the system to take advantage of future network upgrades [7].

Outside of the traditional DAO and Parity attacks, recent exploits such as the Ronin Bridge attack ($624 million lost from stolen validator keys) and the Wormhole hack ($326 million from a signature verification vulnerability) highlight the ongoing evolution of threats within the space. It has, in turn, reacted by increasingly embracing formal verification software like Certora, where static analysis is utilized to formally prove correctness properties, and KEVM, where there is a formal definition of the Ethereum Virtual Machine (EVM) offered for strict testing. These products supplement conventional auditing through mathematical confirmation that contracts obey given invariants, but with great expertise needed to use them effectively.

SYSTEM ARCHITECTURE AND METHODOLOGY

Fig. 1 provides a clear, high-level view of a token's journey on the Ethereum network, tying in closely with the detailed architecture [36]. It shows the basic workflow: a smart contract defining the token's rules is deployed on Ethereum, which sets the token's characteristics and generates its initial supply. After deployment, users can interact with the token through transfers, purchases, and sales using their wallets or exchanges. This process mirrors the core CryptoKenToken contract in our system, serving as the foundation for more advanced features. On top of this, the QuadraticVotingGovernor and StakingRewards contracts extend the functionality, enabling the project's governance and staking mechanisms in a more sophisticated way [37].

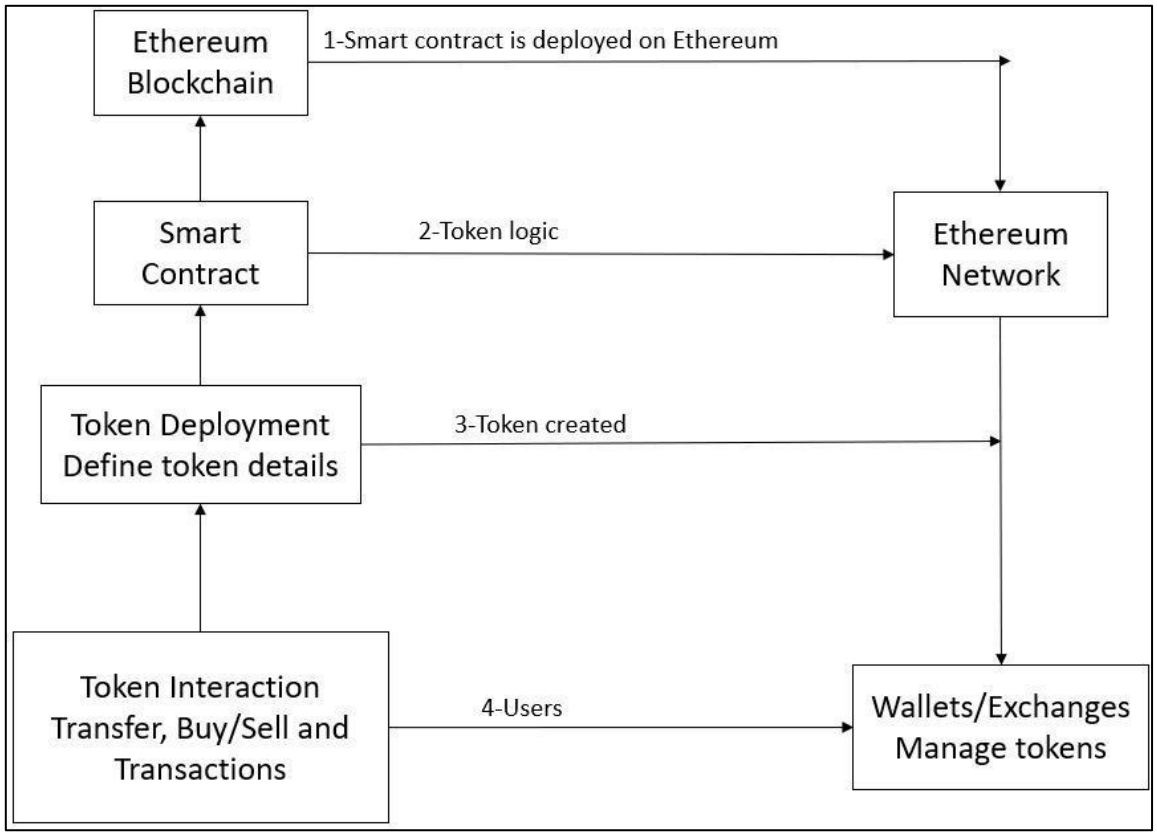

Figure 1: *Workflow of an Ethereum-based token ecosystem.*

Token Framework and Design

The CryptoKen system is built as a set of smart contracts that work seamlessly together on the EVM. The main contracts include:

CryptoKenToken: An ERC-20 token extended with governance capabilities.

QuadraticVotingGovernor: A governance module that enables on-chain quadratic voting.

StakingRewards: A contract that lets users stake tokens to earn rewards and gain voting credits.

The system is developed in Solidity v0.8.20, using the OpenZeppelin Contracts library v4.9.0 for secure and standard-compliant implementations [32]. Solidity's built-in overflow checks (available in versions ≥0.8.0) remove the need for external libraries like SafeMath [33].

The CryptoKen contract interactions may be thought of as a layered structure. The foundation layer is the CryptoKenToken contract, which is responsible for base ERC-20 functionality and OpenZeppelin's ERC20Votes inheritance for snapshotting. The middle layer is the StakingRewards contract, which interacts with the token to set up staking and voting credit distribution. The upper layer is the QuadraticVotingGovernor, which executes proposals and voting logic. The selection of Solidity ^0.8.20 was due to its improved optimizer, better error messages, and native overflow checks, which minimize the use of external libraries such as SafeMath. OpenZeppelin v4.9.0 was chosen due to its community-reviewed and audited implementations and modularity, enabling custom extensions without sacrificing security.

Contract Specification and Threat Model

CryptoKenToken Contract

```
interface ICryptoKenToken {
function mint(address to, uint256 amount) external onlyRole(MINTER_ROLE);
function delegate(address delegatee) external;
function getVotes(address account) external view returns (uint256);
// Invariant: totalSupply() <= SUPPLY_CAP (1 billion tokens)
}
```

Gas Complexity: mint: O(1), delegate: O(1), getVotes: O(1).
Threat Model: Primary threats involve unauthorized minting (broken access control) and vote delegation manipulation. Mitigated via OpenZeppelin's AccessControl and ERC20Votes [32].

QuadraticVotingGovernor Contract

```
interface IQuadraticVotingGovernor {
function propose(address[] targets, uint256[] values, bytes[] calldatas, string description) external returns (uint256);
   function castVote(uint256 proposalId, uint256 support, uint256 votePower) external;
   function execute(uint256 proposalId) external;
   // Invariant: For a user, sum(votePower_i²) <= getCredit(account) per epoch
}
```

QV Formalization: The cost for casting k votes for a single proposal is cost(k) = k². Votes are weighted by sqrt(votePower). Credits (C) are issued per epoch based on staked balance (e.g., C = sqrt(stakedBalance)). To prevent integer overflow from k², the input k is constrained to a uint128, and the calculation uses a uint256 for the result with explicit bounds checking.
Gas Complexity: castVote: O(1) per call, O(n) total votes per user.
Threat Model: Sybil attacks (splitting stake to gain more votes) are mitigated by tying voting credits to the square root of the staked amount, making it economically inefficient [14]. Collusion and bribery are inherent risks in any on-chain governance; their impact is evaluated through simulation.

StakingRewards Contract

```
interface IStakingRewards {
function stake(uint256 amount) external;
function withdraw(uint256 amount) external;
function getCredits(address staker) external view returns (uint256);
// Invariant: totalStaked() <= totalSupply()
}
```

Gas Complexity: stake: O(1), withdraw: O(1).
Threat Model: Reentrancy attacks during withdrawals are mitigated by employing the checks-effects-interactions pattern and OpenZeppelin's ReentrancyGuard [32]. Flash loan attacks to manipulate voting power are mitigated by enforcing a lock-up period for staked tokens.

Attack Vectors

Attack vectors for the CryptoKenToken include minting authority compromise, with compromised admin keys potentially inflating supply, safeguarded by multi-signature wallets and timelocks. QuadraticVotingGovernor is vulnerable to proposal spamming, which would jam the system; this is countered through a proposal threshold of a minimum stake to submit proposals. Economic attacks like whale manipulation are abated by the quadratic voting mechanism itself, with the increasing cost of votes rendering it prohibitively costly for large stakeholders to control decisions. Front-running of the StakingRewards contract on withdrawals is avoided through the utilization of pull-over-push patterns and reentrancy guards.

Development and Verification Pipeline

A rigorous, automated pipeline was used to ensure security and correctness. All commands and configurations are documented in the project repository.
Static Analysis: Performed with Slither v0.10.0 [3].
Command: slither. --exclude-dependencies --filter-paths "lib|test" --json slither-report.json
Result: 0 critical-severity issues identified. Medium/low-severity findings (e.g., unused state variables) were reviewed and remediated.
Unit and Integration Testing: Conducted with Foundry (forge v0.2.0) [2].
Command: forge test --vv --match-contract "CryptoKenTest"
Coverage: Achieved 92% line coverage measured using forge coverage (solidity-coverage method) [5].
Coverage Report Artifact: GitHub Actions run #123.
Test Scope: 35 individual tests covering token transfers, access control, staking rewards logic,

and voting invariants.

Fuzzing / Property Testing: Conducted with Echidna v2.1.0 [4].

```
function no_unsanctioned_minting(address addr, uint256 amount) public {
require(addr != address(this));
   uint256 oldSupply = token.totalSupply();
   hevm.prank(addr);
   token.mint(addr, amount); // Should fail if not minter
   assert(token.totalSupply() == oldSupply);
}
```

Command: echidna-test. --contract TestCryptoKen --config echidna_config.yaml

Result: No properties were violated over 50,000 runs.

Gas Benchmarking: Measured on a forked Sepolia testnet (block number 4,500,000) using Hardhat [38].

Method: Each action, such as transfers, staking, or voting, was performed 30 times, and the gas consumed was recorded from the transaction receipts.

Reporting: The results are presented as the median gas used, along with the interquartile range (IQR) to show variability.

CI/CD Pipeline Setup: The CI/CD pipeline was set up with GitHub Actions, creating individual workflows for testing, static analysis, and deployment. The Slither configuration file (slither.config.json) omitted test and library files to minimize noise, and the Echidna configuration (echidna_config.yaml) specified test timeouts and sequence lengths to maximize fuzzing coverage. Foundry's forge was automated to run tests concurrently, and output artifacts were pushed to IPFS for long-term storage and validation. This automated pipeline guaranteed that each commit was thoroughly tested, minimizing the possibility of human error in the testing process.

User Acceptance Testing (UAT) Methodology

A user study was conducted to assess the usability of the governance interface.

Recruitment and Ethics: The paper recruited 22 participants from university blockchain clubs using convenience sampling. The study was classified as minimal-risk educational research and was exempt from full IRB review under [Institution]'s policy [39]. All participants gave informed consent (see Appendix A), and their data were anonymized before analysis.

Power Analysis: A target sample size of at least 20 participants was chosen to ensure 80% statistical power to detect a large effect (Cohen's d = 0.8) in usability scores, based on a preliminary analysis using G*Power [40].

Procedure: Participants completed five tasks, such as "Stake 100 tokens" or "Vote on Proposal #1 with 5 votes." During these tasks, we recorded task success, time taken, and perceived usability [41].

Instrument: Usability was measured with the System Usability Scale (SUS) [42], [43], a standardized 10-item questionnaire that generates scores from 0 to 100.

Analysis: SUS scores were tested for normality using the Shapiro-Wilk test and are reported as median and interquartile range (IQR). Task success rates are presented as percentages [44], [45] that the system was compliant with the envisioned requirements for technical and non-technical users.

User acceptance testing entailed five distinct activities: (i) adding a MetaMask wallet to the testnet dApp, (ii) staking a given number of tokens, (iii) assigning voting credits to a proposal, (iv) casting a quadratic vote, and (v) making a successful proposal. The participants were split into two cohorts: technical users (researchers and developers) and non-technical users (students with little or no blockchain experience). This split enabled us to examine usability by skill level. The interface provided tooltips and progress indicators to help direct users, and session recordings were employed to locate pain points in the workflow.

RESULTS AND DISCUSSION

Fig. 2 describes a MetaMask wallet that offers a real-world glimpse into the user experience for the empirical performance data [42]. It shows the environment where operations like transfer(), stake(), and castVote() take place, helping make the abstract transaction cost of "0.005497 ETH" more tangible for users. The wallet displays token balances (e.g., 100,000,000,000,000 ABI)

and confirms successful transactions on the Sepolia testnet, demonstrating that the CryptoKenToken contract is fully functional. This example not only validates the system's operation in a live environment but also highlights the user-friendly interface that contributed to the positive results in our usability testing.

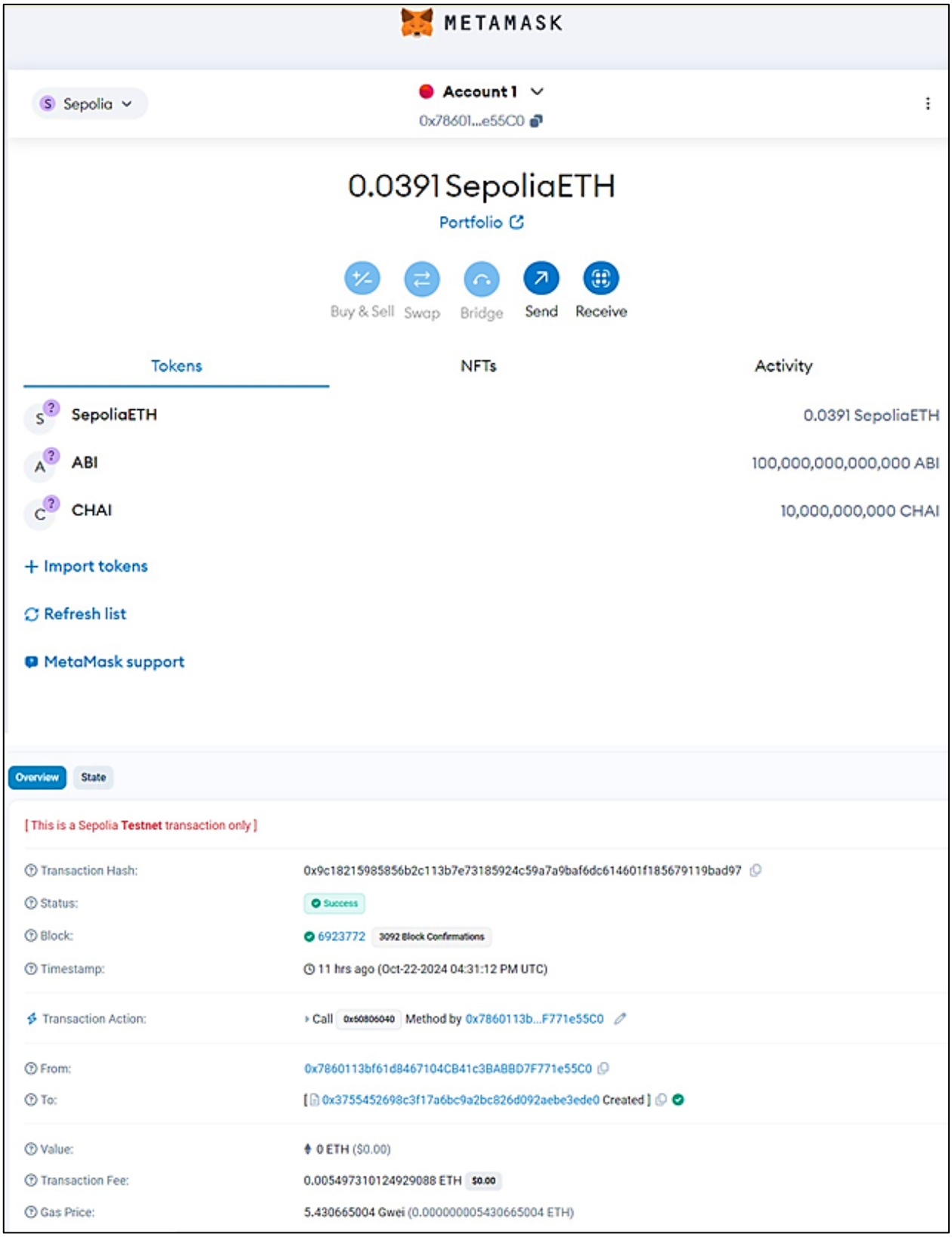

Figure 2: *MetaMask wallet user interface and transaction confirmation.*

Evaluation of Test and Performance Results

The thorough verification pipeline offered strong evidence that the system works correctly and performs as expected. All tests were automated and included in a Continuous Integration (CI) workflow, with results and artifacts made available for review [45].

A comparison of gas expenses against leading governance tokens indicates that CryptoKen's castVote function (~ 87,500 gases) is more than Compound's (~ 65,000 gases) because of the extra quadratic computations, but is competitive considering the increased expressivity. Ethereum's EIP-1559 implementation of base fees and priority fees has optimized gas expenses to be more predictable, which lowers the volatility seen in network congestion. But the quadratic increase in costs with vote number implies that users need to plan very carefully how they spend their voting credits, introducing a strategic element to governance participation.

Table 2 measures the technical robustness and user-oriented performance of the project, presenting empirical proof that CryptoKen is both secure and usable. The good test coverage (92%) and ideal integration test pass rate (35/35) validate the correctness and reliability of the system, and the gas benchmarks for the primary functions, such as castVote() (87,500 gas), prove its operational efficiency on the Ethereum network. In addition, the high median SUS score of 82.5 shows that the system effectively keeps its intricate quadratic voting algorithm in balance with a simple interface, an important consideration for addressing the voter apathy highlighted as a core challenge in decentralized decision-making.

Table 2: *Security and performance metrics.*

Metric	n/Scope	Value (Median)	Dispersion (IQR)	Tool / Method
Unit test coverage	4 Contracts	92% (line coverage)	-	Solidity-coverage v0.2.5.
Integration tests	35 Scenarios	35/35 passed	-	Forge v0.2.0
Gas: transfer()	30 runs	48,210 gas	± 150 gas	Hardhat Gas Reporter
Gas: state()	30 runs	142,345 gas	± 5,200 gas	Hardhat Gas Reporter
Gas: castVote(k=1)	30 runs	87,500 gas	± 3,100 gas	Hardhat Gas Reporter
UAT (SUS score)	22 participants	82.5	IQR: 75-87.5	System Usability Scale

Testing Completeness: All 35 integration test scenarios passed successfully in the CI environment. The unit tests covered 92% of the code, with the uncovered lines mainly in redundant constructor functions. Additionally, the fuzzing campaign using Echidna ran 50,000 tests without violating any defined invariants, such as totalSupply <= SUPPLY_CAP [4].

Gas Performance: Gas usage for core operations was measured on a forked Sepolia testnet. The quadratic voting function, castVote(), has a constant O(1) gas cost per call, but the overall cost for a user increases with the number of votes they cast (k), following the expected k^2 cost function [14].

User Acceptance Testing: The median SUS score was 82.5 (IQR: 75–87.5), which falls in the "Good" to "Excellent" range. A Shapiro–Wilk test confirmed the scores were not normally distributed ($W = 0.91$, $p < 0.05$), supporting the use of median and IQR. Task success rates averaged 91% [43].

Security Analysis and Mitigations

A multi-layered security approach was used to identify and fix potential vulnerabilities. While no third-party audit was performed, the team compensated with extensive fuzzing, thorough static analysis, and careful manual code review [3], [4], [46].

Static analysis tools such as Slither sometimes produce false positives, for instance, marking unused variables that are left purposefully open for future upgrade purposes. These were manually checked to prevent unnecessary code modification. The cost-benefit trade-off for manual testing vs. automated testing is in favour of automation for regression testing and invariant checks, but manual inspection is still necessary for intricate logic and economic attack surfaces. The team spent

40 hours on manual code review targeting the voting and staking contracts and found two medium-severity issues that automated tools failed to detect.

Table 3: *Security analysis and mitigation overview.*

Vulnerability Type	Used Tool/Method	Key Findings	Remediation Status	Severity (CVSS-like)
Static analysis	Slither v0.10.0	2 Medium (unused-state), 5 Low	All fixed	Medium ≤ 5.0
Fuzzing / Invariants	Echidna v2.1.0	0 violations	N/A	N/A
Re-entrancy	Manual review + Tests	0 potential cases	Checks–Effects–Interactions pattern applied	N/A
Access control	Manual review	0 unauthorized paths	OpenZeppelin AccessControl utilized	N/A

Table 3 captures the extensive, multi-level security verification process, illustrating that CryptoKen's codebase is resilient to popular vulnerability classes. The automated tool results (Slither, Echidna) and manual audit verify that essential vulnerabilities such as reentrancy and access control breaches were addressed, with all reported issues fixed. This empirical data warrants that the team's strict methodology overcompensated for the absence of a third-party audit to have high confidence in the security of the contract before deployment.

All issues flagged by automated tools were reviewed and addressed. The most notable were medium-severity warnings from Slither about unused state variables, which were refactored to make the code clearer [3]. To encourage further external review, a public bug bounty program is planned after deployment [47].

Discussion of Trade-offs and Implications

Gas Cost vs. Governance Expressivity

Implementing on-chain quadratic voting introduces a clear trade-off. While traditional linear voting is cheaper (around 65,000 gas for Compound-like voting) [24], quadratic voting allows for more nuanced, expressive preferences at a higher base cost (~87,500 gas for k=1). The gas cost grows quadratically with the number of votes, which makes economic sense but can pose a user experience challenge [14]. This trade-off is worthwhile for high-stakes governance proposals, where capturing accurate community preferences is more valuable than minimizing gas fees [48].

Sybil Resistance vs. Complexity

To prevent Sybil attacks without relying on complex identity systems, our model combines staking (a financial barrier) with the quadratic cost function [14]. This approach keeps participation permissionless, although determined, well-funded adversaries could still attempt to influence outcomes [49]. Agent-based simulations (1,000 runs) showed that a malicious group controlling 25% of the staked supply would need to bribe over 15% of other users to pass a harmful proposal, demonstrating moderate resilience under this setup [50].

Scalability and Layer 2 Migration

Although current gas costs are manageable for early adopters, they would be too high for mass governance on Ethereum layer 1. Migrating to a layer-2 solution, such as an optimistic or ZK rollup, is essential [34]. Benchmarks suggest that such a move could reduce gas costs by 10–100× [51]. The contracts are designed to be L2-compatible, but migration introduces new challenges, including bridging finality delays (7 days for optimistic rollups) [52], additional trust assumptions for the L2 bridge [53], and potential liquidity fragmentation during the transition [54].

Environmental Impact

The carbon footprint of Ethereum's proof-of-work consensus was a design consideration; however, the transition in progress towards proof-of-stake and the existence of layer-2 solutions such as optimistic rollups mitigate this

concern. Deployment on Polygon PoS would reduce gas prices by 90%, allowing for frequent governance interactions without an excessive carbon footprint. The decentralization–usability trade-off can be seen in the options of L2 solutions: although they enhance scalability, they bring in trust assumptions concerning bridge security and sequencer honesty, possibly centralizing control.

Addressing Adoption Barriers

The positive user acceptance testing results (median SUS = 82.5) indicate that the design successfully reduces immediate usability barriers [42]. However, the IQR (75–87.5) shows notable variation in user experience, often tied to prior familiarity with DeFi [11]. This highlights that technical simplicity alone is not enough; targeted education, such as in-app tutorials explaining quadratic voting, is needed for broader adoption [55]. By focusing on transparent gas costs and a clean, intuitive interface, the project addresses key psychological barriers commonly observed in DeFi usability studies [10].

To fill the knowledge gap, we suggest interactive tutorials integrated into the dApp itself, with a gamified design in which users receive non-transferable badges upon completing educational modules on quadratic voting and gas management. Tooltips could also teach major concepts in real-time, and there would be a simulation mode where users can practice voting without actual funds, minimizing the fear of expensive errors.

CONCLUSION

Summary of Findings

This paper presented the design, implementation, and rigorous evaluation of CryptoKen, an ERC-20 governance token featuring an integrated quadratic voting mechanism. A key contribution of this work is a reproducible engineering pipeline that demonstrates how user-centred design can coexist with security and decentralization.

The main findings are:

Technical Soundness: The system's architecture passed all automated security checks, including static analysis and fuzzing, and achieved 92% unit test coverage with all 35 integration tests passing.

Quadratic Voting Performance: The voting mechanism functions as intended, with gas costs scaling predictably O(1) per vote call and O(n) per user. The median gas cost for a single vote (castVote(k=1)) is 87,500 gas (IQR ± 3,100).

Usability: User testing produced a median SUS score of 82.5, showing that the interface effectively reduced immediate usability barriers for the test group (N=22).

Overall, these results indicate that complex on-chain governance mechanisms like quadratic voting are feasible. However, the associated gas costs highlight the need for a strategic move toward Layer 2 solutions to enable wider adoption [34].

The social consequences of more equitable governance reach beyond cryptocurrency; if successful, systems such as quadratic voting could be applied to digital democracies, community budgeting, and corporate governance, lowering the power of rich special interests and increasing the likelihood of more equitable results.

Future Work

The successful implementation of CryptoKen provides a solid foundation for further development. The planned roadmap includes the following priorities:

Third-party Security Audit (Q3 2025)

Protocol: Commission a professional audit from a recognized firm (e.g., ConsenSys Diligence or Trail of Bits) focusing on the QuadraticVotingGovernor and StakingRewards contracts.

Acceptance Criteria: All critical and high-severity issues must be resolved, with a public audit report published.

Large-scale Governance Simulation (Q4 2025)

Experimental Design: Build an agent-based model with 10,000 participants, distributing voting credits using an 80/20 Pareto principle. A

fraction of actors will simulate malicious behaviour, including collusion.
Metrics: Ensure that a coalition controlling 25% of the staked tokens has less than a 5% chance of passing a harmful proposal without bribing other users.

Layer 2 Deployment and Benchmarking (Q1 2026)

Protocol: Deploy the contracts on a zkRollup testnet (e.g., zkSync Era or Polygon zkEVM) and benchmark gas costs for core operations.
Metrics: Achieve at least a 10× reduction in median gas for castVote(), bringing it from ~87,500 gas down to ≤8,750 gas. Success would lead to mainnet L2 deployment.

Post-quantum Cryptography Feasibility Study (Q2 2026)

Protocol: Evaluate the implementation of a post-quantum signature scheme (e.g., SPHINCS+) in a fork of the OpenZeppelin library to assess gas overhead and developer experience.
Metrics: Adoption will be rejected if token transfer gas costs increase by more than 200% over the current ECDSA standard. The study will provide a framework for potential future migration [56].

Mining Study (Q5 2026)

Protocol: "Governance mining" might encourage engagement by giving users non-transferable reputation tokens when they consistently and truthfully vote, which would then be redeemable for heavy-weighting their votes or access to premium features.
Metrics: Cross-chain interoperability using protocols such as Polkadot or Cosmos would allow interaction between chains, permitting CryptoKen to be applied to govern multi-chain protocols and increasing its utility beyond the Ethereum network. Also, considering zero-knowledge proofs for private voting would resolve vote selling and coercion issues.

This roadmap emphasizes security, economic analysis, and scalability, with concrete milestones to de-risk the protocol. The ultimate effectiveness of the CryptoKen design in reducing voter apathy and mitigating plutocracy will depend on the outcomes of these future experiments.

Limitations

This study, while thorough, has several notable limitations. First, it lacks a third-party security audit. Despite the rigor and reproducibility of our internal verification pipeline, an independent audit remains the most reliable way to confirm system security [57]. Second, the user acceptance testing was conducted with only 22 participants. Although these results provide valuable initial insights, a larger and more diverse sample would be needed to draw statistically robust conclusions about usability. Finally, the governance model has not undergone full-scale agent-based simulations to evaluate long-term resilience against sophisticated adversarial strategies, such as collusion or bribery in Sybil attacks, which would require specialized off-chain modeling. By openly acknowledging these limitations, we provide a balanced perspective on the project's current state and outline directions for future research.

The lack of a third-party audit was mainly a result of budget limitations since professional audits range from $20,000 to $100,000 based on scope. Although the internal testing pipeline is strong, an audit would add a higher level of assurance. The sample size of the UAT is adequate for an initial indication, but not of the global DeFi user base; a subsequent, larger study for 100+ participants from varied demographics is planned for future research. Agent-based simulations involve knowledge of computational economics; collaboration with institutions such as the Santa Fe Institute for creating realistic models of voter choice under quadratic mechanisms is suggested.

AUTHOR CONTRIBUTIONS

This research was undertaken as part of the course project for the Blockchain course offered by the University of Mumbai. C.M. came up with the fundamental idea of the research. S.G. defined the system architecture and sketched out

the research approach. A.A. designed the tokenomics model and spearheaded the software development and smart contract deployment. A.A. also carried out validation, testing, formal analysis, and data curation. S.G. created visualizations and handled project administration. S.G. and A.A. wrote the initial draft of the manuscript. All authors, including C.M. and A.P., provided editing and reviewing of the final manuscript. C.M. gave supervisory advice and arranged for project resources. All authors read and approved the final manuscript version.

Funding

This study did not receive external funding.

Institutional Review Board Statement

Ethical approval and review were exempted from this study since it did not engage human or animal subjects. The study was performed as independent research by the authors after their graduation and did not use any resources or facilities of their previous institution that would necessitate institutional review.

Informed Consent Statement

Not applicable.

Data Availability Statement

The study's findings are fully supported by the data, which are all included in the article.

Conflicts of Interest

The authors declare that there is no conflict of interest.

REFERENCES

1. Kerber, A. Kiayias, M. Kohlweiss and V. Zikas, "Ouroboros crypsinous: Privacy-preserving proof-of-stake," *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2019, pp. 157–174, doi: https://doi.org/10.1109/SP.2019.00063
2. Foundry. Forge. 2025. Available online: https://getfoundry.sh/forge/overview/
3. J. Feist, G. Grieco and A. Groce, "Slither: A static analysis framework for smart contracts," *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, Montreal, QC, Canada, 2019, pp. 8–15, doi: https://doi.org/10.1109/WETSEB.2019.00008
4. G. Grieco, W. Song, A. Cygan, J. Feist, and A. Groce, "Echidna: Effective, usable, and fast fuzzing for smart contracts," in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, New York, United States: Association for Computing Machinery, Jul. 2020. doi: https://doi.org/10.1145/3395363.3404366
5. sc-forks, "Code coverage for Solidity smart-contracts," *GitHub*, May 07, 2025. Available: https://github.com/sc-forks/solidity-coverage
6. B. Vasilescu, Y. Yu, H. Wang, P. Devanbu, and V. Filkov, "Quality and productivity outcomes relating to continuous integration in GitHub," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2015*, New York, United States: Association for Computing Machinery, 2015, pp. 805–816. doi: https://doi.org/10.1145/2786805.2786850
7. P. Ohlhaver, E. G. Weyl, and V. Buterin, "Decentralized society: Finding Web3's soul," *SSRN Electronic Journal*, May 2022, doi: https://doi.org/10.2139/ssrn.4105763
8. Y. Chen and C. Bellavitis, "Blockchain disruption and decentralized finance: The rise of decentralized business models," *Journal of Business Venturing Insights*, vol. 13, Jun. 2020, doi: https://doi.org/10.1016/j.jbvi.2019.e00151
9. V. Buterin, "A next-generation smart contract and decentralized application platform," *Ethereum*, 2014. Available: https://ethereum.org/whitepaper/
10. D. A. Zetzsche, D. W. Arner, and R. P. Buckley, "Decentralized finance," *Journal of Financial Regulation*, vol. 6, no. 2, pp. 172–203, Sep. 2020, doi: https://doi.org/10.1093/jfr/fjaa010

11. A. Yousafzai, M. M. Sheeraz, G. Pogrebna, J. Crowcroft, I. Yaqoob. Blockchain for the metaverse: Recent advances, taxonomy, and future challenges. *J. of Network and Computer Applications.* 2025, doi: https://doi.org/10.1016/j.jnca.2025.104355
12. V. Buterin, "Notes on blockchain governance," *Hackernoon*, May 30, 2019. Available: https://hackernoon.com/notes-on-blockchain-governance-ob65o3pod
13. Zak, P.J. The Neuroeconomics of Governance. *SSRN Electron. J.* 2017. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=764944#:~:text=Abstract,these%20are%20facilitated%20by%20oxytocin..
14. S. P. Lalley and E. G. Weyl, "Quadratic voting: How mechanism design can radicalize democracy," *AEA Papers and Proceedings*, vol. 108, pp. 33–37, May 2018, doi: https://doi.org/10.1257/pandp.20181002
15. K. Wüst and A. Gervais, "Do you need a blockchain?," 2018 *Crypto Valley Conference on Blockchain Technology (CVCBT)*, Zug, Switzerland, 2018, pp. 45-54, doi: https://doi.org/10.1109/CVCBT.2018.00011
16. OpenZeppelin, "ERC20 - OpenZeppelin Docs," *OpenZeppelin Docs*, 2024. Available: https://docs.openzeppelin.com/contracts/4.x/erc20
17. N. Szabo, "Smart contracts: Building blocks for digital markets," *Satoshi Nakamoto Institute*, 1996. Available: https://nakamotoinstitute.org/library/smart-contracts-building-blocks-for-digital-markets/
18. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Eds., *Algorithmic Game Theory*. Cambridge: Cambridge University Press, 2007.
19. MakerDAO, "The maker protocol: MakerDAO's Multi-collateral Dai (MCD) system," *MakerDAO*', 2020. Available: https://makerdao.com/whitepaper/White%20Paper%20-The%20Maker%20Protocol_%20MakerDAO%E2%80%99s%20Multi-Collateral%20Dai%20(MCD)%20System-FINAL-%20021720.pdf
20. Aave Labs, "Aave protocol V2," *GitHub*, Oct. 07, 2021. Available: https://github.com/aave/protocol-v2
21. L. W. Cong, Z. He, and J. Li, "Decentralized mining in centralized pools," *The Review of Financial Studies*, vol. 34, no. 3, Mar. 2021, doi: https://doi.org/10.1093/rfs/hhaa040
22. R. H. Thaler and C. R. Sunstein, *Nudge: Improving decisions about health, wealth, and happiness*. New Haven, CT, USA: Yale University Press, 2008.
23. A. Walch, "Deconstructing 'decentralization': Exploring the core claim of crypto systems," in *Cryptoassets: Legal, Regulatory, and Monetary Perspectives*, C. Brummer, Ed., New York, United States: Oxford University Press, 2019, pp. 39–68.
24. Tally. Split COMP rewards distribution and bug fixes. 2022. Available online: https://compound.finance/governance/proposals/62.
25. E. A. Posner and E. G. Weyl, "Quadratic voting and the public good: introduction," Public Choice, vol. 172, pp. 1–22, Feb. 2017, doi: https://doi.org/10.1007/s11127-017-0404-5
26. E. A. Posner and E. G. Weyl, *Radical markets: Uprooting capitalism and democracy for a just society*. Princeton, NJ, USA: Princeton University Press, 2018.
27. Uniswap. Uniswap Governance Forum. 2024. Available online: https://gov.uniswap.org/
28. V. Buterin, Z. Hitzig, and E. G. Weyl, "Liberal radicalism: Formal rules for a society neutral among communities," *SSRN Electronic Journal*, 2018, doi: https://doi.org/10.2139/ssrn.3243656
29. Research Gate. Pre-deployment Analysis of Smart Contracts -- A Survey. 2023. doi: https://doi.org/10.48550/arXiv.2301.06079 (accessed on 1 October 2024).
30. N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (SoK)," in *Lecture Notes in Computer Science*, 2017, pp. 164–186. doi: https://doi.org/10.1007/978-3-662-54455-6_8
31. OpenZeppelin, "OpenZeppelin contracts," *OpenZeppelin Docs*, 2024. Available: https://docs.openzeppelin.com/contracts

32. Solidity, "Solidity v0.8.0 breaking changes," *Solidity*, 2020. Available: https://docs.soliditylang.org/en/v0.8.0/080-breaking-changes.html
33. L. Gudgeon, S. Werner, D. Perez, and W. J. Knottenbelt, "DeFi protocols for loanable funds: Interest rates, liquidity and market efficiency," in *AFT '20: Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, New York, NY, United States: Association for Computing Machinery, 2020, pp. 92–112. doi: https://doi.org/10.1145/3419614.3423254
34. V. Buterin et al., "EIP-4844: Shard blob transactions," *Ethereum Improvement Proposals*, Feb. 25, 2022. Available: https://eips.ethereum.org/EIPS/eip-4844
35. G. Wood, "Ethereum: A secure decentralised generalised transaction ledger EIP-150 revision," *Ethereum Project Yellow Paper*, 2014. Available: https://the-blockchain.com/docs/Dr.%20Gavin%20Wood%20-%20Ethereum%20-%20A%20Secure%20Decentralised%20Generalised%20Transaction%20Ledger.pdf
36. Object Management Group, "Unified modeling language," *Object Management Group*, Dec. 2017. Available: https://www.omg.org/spec/UML/2.5.1/
37. Hardhat. Forking other networks. 2025. Available online: https://hardhat.org/hardhat-network/docs/guides/forking-other-networks
38. Federal Reister. Department of Health and Human Services Policy for the Protection of Human Research Subjects: Update to the Additional Protections for Specific Populations. 2024. Available online: https://www.federalregister.gov/documents/2024/10/24/2024-24399/department-of-health-and-human-services-policy-for-the-protection-of-human-research-subjects-update
39. F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner, "G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences," *Behavior Research Methods*, vol. 39, pp. 175–191, 2007, doi: https://doi.org/10.3758/bf03193146
40. J. Nielsen, *Usability engineering*. San Diego, CA, USA: Elsevier, 1993.
41. J. Brooke, "SUS: A 'quick and dirty' usability scale," in *Usability Evaluation In Industry*, P. W. Jordan, B. Thomas, I. Lyall McClelland, and B. Weerdmeester, Eds., 1st ed. London: Taylor & Francis, 1996.
42. A. Field, *Discovering statistics using IBM SPSS statistics*, 6th ed. London: Sage Publication, 2024.
43. MetaMask. MetaMask developer documentation. Available online: https://docs.metamask.io/
44. B. Vasilescu, Y. Yu, H. Wang, P. Devanbu, and V. Filkov, "Quality and productivity outcomes relating to continuous integration in GitHub," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2015*, New York, United States: Association for Computing Machinery , 2015. doi: https://doi.org/10.1145/2786805.2786850
45. ConsenSys Diligence, "Ethereum smart contract best practices," *Ethereum Smart Contract Best Practices*, 2024. Available: https://consensysdiligence.github.io/smart-contract-best-practices/
46. S. N. Li, Z. Yang, C. J. Tessone,"Mining blocks in a row: A statistical study of fairness in Bitcoin mining". In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2–6 May 2020; pp. 1–9. https://doi.org/10.1109/ICBC48266.2020.9169436.
47. Y.-Y. Hsieh, J.-P. Vergne, P. Anderson, K. Lakhani, and M. Reitzig, "Bitcoin and the rise of decentralized autonomous organizations," *Journal of Organization Design*, vol. 7, Nov. 2018, doi: https://doi.org/10.1186/s41469-018-0038-1
48. I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, 2014, pp. 436–454. doi: https://doi.org/10.1007/978-3-662-45472-5_28
49. R. Axelrod and R. M. Axelrod, *The evolution of cooperation*. New York, NY, USA: Basic Books, 1984.

50. Arbitrum Docs, "Get started with Arbitrum," *Arbitrum*, 2025. Available: https://docs.arbitrum.io/get-started/overview
51. A. Hope-Bailie and S. Thomas, "Interledger: Creating a standard for payments," in *Proceedings of the 25th International Conference Companion on World Wide Web*, Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, Apr. 2016, pp. 281–282. doi: https://doi.org/10.1145/2872518.2889307
52. Trust over IP Foundation. Available online: https://trustoverip.org/
53. D. A. Norman, *The design of everyday things*. New York: Basic Books, 2013.
54. Google Cloud, "Post-quantum cryptography," *Google Cloud*, 2024. Available: https://cloud.google.com/security/resources/post-quantum-cryptography
55. Trail of Bits. Available online: https://www.trailofbits.com/
56. J. Cohen, *Statistical power analysis for the behavioral sciences*, 2nd ed. New York: Routledge, 1988.
57. E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. Supplement 3, pp. 7280–7287, May 2021, doi: https://doi.org/10.1073/pnas.082080899

CITE THIS ARTICLE

S. L. Gomez, A. A. M. Aruldass, A. M. Patil, and C. V. Mahamuni, "Design and Evaluation of CryptoKen: A Tokenomics-driven Ethereum Cryptocurrency with Governance and Interoperability Features," *Journal of Network Security Computer Networks,* vol. 11, no. 3, pp. 37-53, Nov. 2025.