JOURNAL OF Emerging Paradigms in Computing Systems JEPCS

Journal of Emerging Paradigms in Computing Systems (JEPCS) | Volume 03, Issue No. 01, 2026

ARTICLE TYPE: Research Article

Article ID: JEPCS-2026-0301-0024

Blockchain-Based Electronic Voting Framework for Enhancing Electoral Transparency, Security, and Efficiency in Zimbabwe

Marvellous Courage Tshuma[1,*], Edwin Farai Madzuwo[2], and Tatenda Benjamin Manyuchi[1]

[1] Department of ICT and Electronics, Chinhoyi University of Technology, Chinhoyi, Zimbabwe
[2] Department of Computer Science, University of Cumberlands, Williamsburg, Kentucky, USA

*Corresponding author: mtshuma@cut.ac.zw

Contributing authors: mtshuma@cut.ac.zw, tmanyuchi@cut.ac.zw, efmadzuwo@gmail.com

Received: May 29, 2026 **Accepted:** June 30, 2026

Abstract- This study proposes a blockchain-based electronic voting framework for Zimbabwe that addresses electoral auditability, voter eligibility, ballot secrecy, duplicate voting, administrative control, and voter usability. Using design science research, the study translates requirements drawn from the electronic-voting and blockchain literature into a hybrid off-chain/on-chain architecture and evaluates the design ex ante through requirements-to-architecture traceability and threat-control-residual-risk analysis. The framework separates civil identity from ballot processing, issues one election-specific voting credential to each eligible voter, and uses smart contracts on a permissioned multi-stakeholder blockchain to enforce election-state and duplicate-vote rules. Operational and identity data remain off-chain, while election-critical state is recorded on a tamper-evident distributed ledger. The proposed voter interface also hides wallet and cryptocurrency operations from ordinary voters. The analysis indicates that blockchain can strengthen auditability and rule enforcement when it is placed within a wider set of identity, privacy, governance, endpoint-security, and legal controls. It does not by itself guarantee anonymity, availability, coercion resistance, endpoint security, or legal legitimacy. The main contribution is therefore a Zimbabwe-focused design framework and a staged evaluation agenda for future prototyping, security assessment, usability testing, scalability experiments, governance review, and legally authorized pilot evaluation.

Keywords- blockchain; electronic voting; smart contracts; electoral security; Zimbabwe

1. Introduction

Electronic voting is not only a software problem. It is a high-assurance socio-technical problem in which the technology must support eligibility, uniqueness of the vote, ballot secrecy, integrity, auditability, availability, and accountable administration at the same time. This combination is important because improving one property can weaken another. For example, stronger identity checks can improve eligibility control while creating privacy risks if identity information is linked too closely to the ballot. Similarly, a fully centralized system may simplify administration but can concentrate technical power in a small number of privileged operators. We therefore treat the central design problem as the distribution of trust rather than the simple digitization of a ballot.

We investigated blockchain because some of its properties are relevant to this trust-distribution problem. Cryptographically linked records, replicated state, consensus, and programmable smart contracts can support tamper-evident audit trails and consistent enforcement of election rules [1]-[3]. However, our review also showed that blockchain does not solve several of the most difficult electoral problems. It cannot decide who is legally eligible to

vote, protect a compromised voter device, guarantee coercion resistance, or make an election legally valid. It can also preserve a badly designed identity-to-ballot linkage just as permanently as it preserves a valid state transition. For this reason, the framework does not treat blockchain as a complete voting system; it uses the ledger only where replicated, independently inspectable election state provides a clear design benefit.

A further reason for developing the framework was the weakness of the common assumption that one blockchain wallet can represent one voter. We reject that assumption because possession of a wallet does not establish citizenship, registration status, constituency, age, or any other legal qualification. A single person can also create many addresses. In our design, voting power therefore comes from an election-specific authorization issued after an authoritative eligibility check, not from ownership of a blockchain address. This decision brings together the legal definition of an eligible elector with the technical system, while avoiding the voter having their civil identity recorded on the ballot paper.

Zimbabwe is a suitable illustration within which to explore the design decisions as any legitimate electronic voting proposal would need to fit within existing electoral institutions, legal rules, infrastructure limitations and differential degrees of digital connectivity. What matters to us is not whether a blockchain application can actually record a vote, but whether the proposed architecture can be understood by ordinary people, clearly and audibly verified by trusted observers, and feasibly operated in a real-world setting where connectivity and backup capabilities are reasonable. Forcing voters to register and use a separate wallet, engage with seed phrases, deposit crypto, and comprehend transaction fees would put technical risk and tasks on the voter. We thus remove blockchain operations from the voter experience to provide greater abstraction.

Four related study questions are then presented: (1) What security and socio-technical requirements should a blockchain voting architecture for Zimbabwe address? (2) How should civil identity, operational information, and voting credentials be detached from one another, and how should smart-contract logic, voting information, and the state of the ledger be separated? (3) How can one-person-one-vote be ensured without associating a civil ID with a wallet ID? (4) What evidence would be necessary to warrant a controlled deployment? What is new and innovative about the proposed framework is its ability to intertwine these concerns into a single design but also to bring into plain sight the other assumptions required for the trust along with any risks that remain outside its scope.

The output of the present study is therefore a conceptual design artefact, not an implemented national voting system. No transaction benchmarks, reliability percentages, user-satisfaction results, or security guarantees are claimed.

2. Related Work

2.1. Electronic voting, verifiability, and trust

Even before blockchain, there was already research on electronic voting and some principles to adhere to when judging newer designs. Helios demonstrated the value of web-based open-audit voting [4] while Chaum, Ryan and Schneider [5] proved how voter-verifiable methods can enable the reassessment of election results when relying on a central authority is the only possibility. These methods stress that it is not only the institutions that assert the correctness, there must be evidence to support that correctness. End-to-end verifiability typically needs facilities for voters or observers to verify that ballots have been dealt with properly without having to reveal their preferences.

Cryptographic correctness is not the only requirement for trust in an electronic election. An election involves voter devices, authentication systems, interfaces, election officials, help desks, networks, and legal procedures. Even a mathematically correct tally protocol can fail operationally if credentials are issued incorrectly, endpoints are compromised, administrators misuse their privileges, voters do not understand the interface, or the system becomes unavailable during voting. Electronic voting is therefore a socio-technical system whose security properties must be mapped to both technical and institutional controls.

2.2. Blockchain and smart contracts in electronic voting

Each node in the blockchain has access to the entire chain, and any changes to the past state are detectable once they have been made, which is achieved by being cryptographically linked and by making an agreement. The key attributes that have made it relevant to voting are these: replicated verification, tamper evidence, programmable validation, and an append-oriented audit history. Smart contracts can define election open or close times, disallow unauthorized or duplicate ballots, collect just valid ballots, and manage elections and tally states. McCorry et al. showed the feasibility of voting logic being implemented with smart contracts while granting a high level of privacy to voters [6].

The literature reveals significant differences in how potential blockchain voting systems are designed (architecture), who is able to manage the networks, how participants are authenticated, how privacy is achieved, and deployment models [2, 3]. We see this as a variation of the claim, "blockchain makes elections secure," which is too broad to guide the development of systems. Security depends on who sets voter eligibility, who maintains the validators, who holds the deployment and recovery keys, what information is recorded on-chain, how a ballot is authorized, and what evidence independent observers can verify. These questions were used as design inputs rather than being regarded merely as implementation details.

This is represented in the public/permissioned choice. A public blockchain can replicate widely, but at the same time it may also move some aspects of the governance, metadata exposure, confirmation behaviour, and in some cases the transaction cost, out of the institution of the election authority. A permissioned network can offer clearer accountability and membership, yet that doesn't mean the network is decentralized when one organization owns all the keys and all the validators. We don't therefore equate permissioning with decentralization. The purpose of the design is to ensure that the technical powers are shared fairly among the authorized actors and that there is separation of duties and independent auditing.

2.3. Privacy, coercion resistance, and ballot secrecy

Replacing a voter's name with a blockchain address does not assure that the voter's vote will be secret. Addresses are pseudonyms and the timing of transactions, network metadata, application logs or identity-registration records might enable correlation. Separation is a core component of privacy: voter authorization should be separated from civil identity, while ballot secrecy and integrity must be preserved. Depending on the threat model and the desired protocol, various techniques are applicable to this separation such as anonymous credentials, blind signatures, zero-knowledge proofs, mix networks, homomorphic encryption, and threshold cryptography.

Recent research illustrates the need for these additional mechanisms. Shi et al. [7] introduced anonymous and self-tallying voting via blockchain with time bounded ballot secrecy. Yousfi and Chaieb [8] explored scalable and coercion-resistant blockchain voting, and Kho et al. [9] formalized such desirable properties as confidentiality, anonymity, unforgeability and verifiability. These studies demonstrate that there is a difference between coercion resistance and ledger integrity. Even if the ledger is tamper-evident, a voter who can prove how they voted may still be susceptible to vote-buying or coercion.

This problem is exacerbated in the case of remote voting when the physical environment of the voter is not completely under the control of the election authority. Another party may observe or pressure the voter, ballot selection may be influenced, and credential-recovery mechanisms may provide an additional attack path. A secure blockchain architecture should therefore explicitly define the threats it protects against and those that fall outside the security boundary of the ledger.

2.4. Identity, credentials, and the research gap

Identity is a theme strongly linked to blockchain voting, yet often not properly defined. Voter eligibility requires verification by an authoritative process, whereas a blockchain verifies possession of a cryptographic key. These are distinct trust domains. All these must be appropriately integrated with the architecture in such a way that there is no public relation between a person's civil identification and ballot choice. A method is to establish identity off-chain, and then provide on-chain an election credential or piece of proof which can only be consumed once during the election. This can be used in support of one-person-one-vote without revealing the voter's civil identifier.

Based on this review, we noted that, although there are prototype blockchain voting systems, the lack of integration among their key security and operational concerns constitutes the problem addressed by this research. While existing studies address important individual properties, the point is that a Zimbabwe-based design should bridge the gap between these properties and the required voter eligibility, data handling operations, governance, ease of use, and realistic failure cases, among others. We have therefore consolidated these concerns into our framework and associated them with a responsible security mechanism, an explicit assumption, a residual risk, and a future validation method, identified and assigned for each claimed security property.

Another take from the literature is the notion that transparency in elections should not be equated with disclosure of private information. We believe a working audit tool should show evidence of compliance with electoral laws but not disclose the votes of an identifiable voter. That is why, distinct from keeping ledger identity records, the architecture keeps ledger state related to ballots. Observers should be able to audit the contract logic, transitions of election state, activity of validators, and paperwork of counts, and the system should not allow a direct mapping of a civil identity to a candidate.

Recovery and exception handling are also pertinent elements of our security design. In the real world, lost credentials, interrupted sessions, administrative corrections, changes of candidates and other unusual events are problems for actual electoral systems. A recovery protocol that allows one administrator to restore voting capacity with weak safeguards may undermine the properties of the main protocol. Therefore, credential lifecycle management, administrative authorization, audit logging, and credential recovery governance are included in the framework rather than treated merely as operational details.

3. Methodology

3.1. Design science approach

Design science research (DSR) was selected because the present research problem requires the construction and reasoned evaluation of an information-systems artefact [10], [11]. The objective at this stage is not to measure the performance of

an existing voting platform, because no such platform has been implemented in this study. Instead, the study develops a conceptual architecture that responds to identified electoral and security requirements and examines whether the proposed components, trust boundaries, and controls are internally coherent. DSR is therefore more appropriate to this phase than an experimental evaluation or a software-development lifecycle presented as if implementation had already occurred. Prototype development and empirical testing are reserved for subsequent work.

The design process followed four steps. First, we derived requirements from the electronic-voting and blockchain literature, concentrating on eligibility, one-person-one-vote, secrecy, integrity, auditability, usability, availability, and governance. Second, we mapped each requirement to the architectural component that could most directly enforce or support it. Third, we developed a threat model covering credential theft, multiple-address voting, identity-ballot linkage, smart-contract defects, insider abuse, endpoint compromise, coercion, denial of service, validator failure, and privileged collusion. Fourth, we used requirements-to-architecture traceability and threat-control-residual-risk analysis to examine whether each major claim in the design had a responsible mechanism and whether important residual risks remained visible.

3.2. Requirements and design assumptions

The design is guided by seven core requirements. A main principle for ensuring the right to vote is that only people who are allowed to vote for a certain election can vote for that election. To ensure uniqueness, no instance of a valid entitlement can generate more ballots than the mechanisms of the voting procedure allow. Ballot secrecy means that the ballot should not show a direct link between the voter's civil identity and the choice of candidate. To ensure integrity and auditability, the state of the election should be auditable and fault-tolerant. Usability means that common voters shouldn't be expected to use cryptocurrency wallets or to manipulate blockchain keys. Availability requires tolerance of the failure of an individual service or validator. Separation of duties in governance means that no single technical actor takes unchecked control of any functions related to the elections.

The threat model takes a deliberate approach to the failure of one of the components or actors. It handles external attackers, compromised voter endpoints, creation of multiple blockchain addresses, malicious or negligent administrators, validator outages for a limited period of time, and collusion by trusted parties. We do not presume that infrastructure operators are trustworthy merely because they are members of an approved institution. Meanwhile, the model does not offer protection against an unlimited adversary or a fully compromised authoritative identity service. Parameters like validator count, fault threshold, credential protocol, consensus configuration and key ceremonies are then implementation-specific and cannot be validated through reasoning alone and must be assessed in a future prototype.

Table 1: *Requirements-to-architecture traceability.*

Requirement	Architectural mechanism	Future evidence
Eligibility	Authorized eligibility service + election credential	Eligibility/integration tests
Uniqueness	Single-use credential/nullifier + smart contract	Replay and duplicate-vote tests
Ballot secrecy	Identity-ballot separation + privacy proof	Linkability/metadata analysis
Integrity	Smart-contract validation + replicated ledger	Contract/ledger consistency tests
Usability	Web/mobile interface abstracts wallets	Usability/accessibility study
Availability	Redundant services and validators	Load/failover/recovery tests
Governance	Role separation + multi-stakeholder validators	Governance/key-compromise exercises

One rule we used consistently for the conceptual evaluation was that a requirement would not be considered satisfied just because a blockchain component was present. We characterized the mechanism that is responsible for each requirement, the trust assumptions needed, the primary failure mode, and the types of evidence that a future prototype should generate for each one. Integrity can be supported by ledger replication, but ballot secrecy cannot. Similarly, a login screen can be used to gain access to an application, but without being linked to a formal eligibility process, it does not guarantee voter eligibility. This separation means that the design doesn't ascribe other unrelated security properties to the ledger. Therefore, the present evaluation is not based on statistics. It questions if the architecture is logical in relation to the requirements, whether trust is unnecessarily concentrated, and if there are still important residual risks that are visible. This approach cannot be used to show operational effectiveness, performance or voter acceptance. Its worth is in the fact that it sets forth claims that can be subsequently substantiated with measurable data. For a future implementation, the traceability matrix should be kept and test results, failed cases, changed assumptions, and measured performance for each requirement should be added to the matrix.

4. Evaluation

The framework is not assessed through an experiment, but rather ex ante. The evaluation is performed based on the requirements to architecture traceability matrix and threat-control-residual-risk analysis outlined in the methodology. At this stage there are no prototype benchmarks, user-study results or deployment measurements reported.

5. Results and Discussion

5.1. Proposed system architecture

The final result is a layered architecture similar to that in Figure 1. The user layer has interfaces for voters and election officials. The presentation layer offers ballot viewing, ballot confirmation, results viewing, accessibility options, ballot authentication, and candidate informationrmation. The Application layer handles voter eligibility, election management, credential orchestration, coordinated interaction with the database and blockchain, notifications, monitoring, audit logging, controller interaction and voter roles.

Persistent operational data is stored off-chain. These logs contain authentication information, eligibility data, election configuration, candidate information, user roles and permissions, audit info and system logs. The data should be encrypted at rest and in motion, have least-privilege access controls, be backed up and be subject to proper retention policies—at a minimum. Importantly, there should not be a direct National ID-to-candidate relationship in the operational database. An access failure should not result in disclosure of ballot selection.

The blockchain layer includes the smart contracts and a multi-stakeholder, permissioned ledger. Smart-contract logic is deliberately limited to election-critical functions that may be essential to the integrity of the election, such as determining elections state, accepting authorized voting credentials, rejecting duplicate ballots, recording election-related state, generating relevant events, and managing controlled transitions to the tally state. A smaller contract surface might simplify auditing and limit the number of functions that could fail or be misused in the contract.

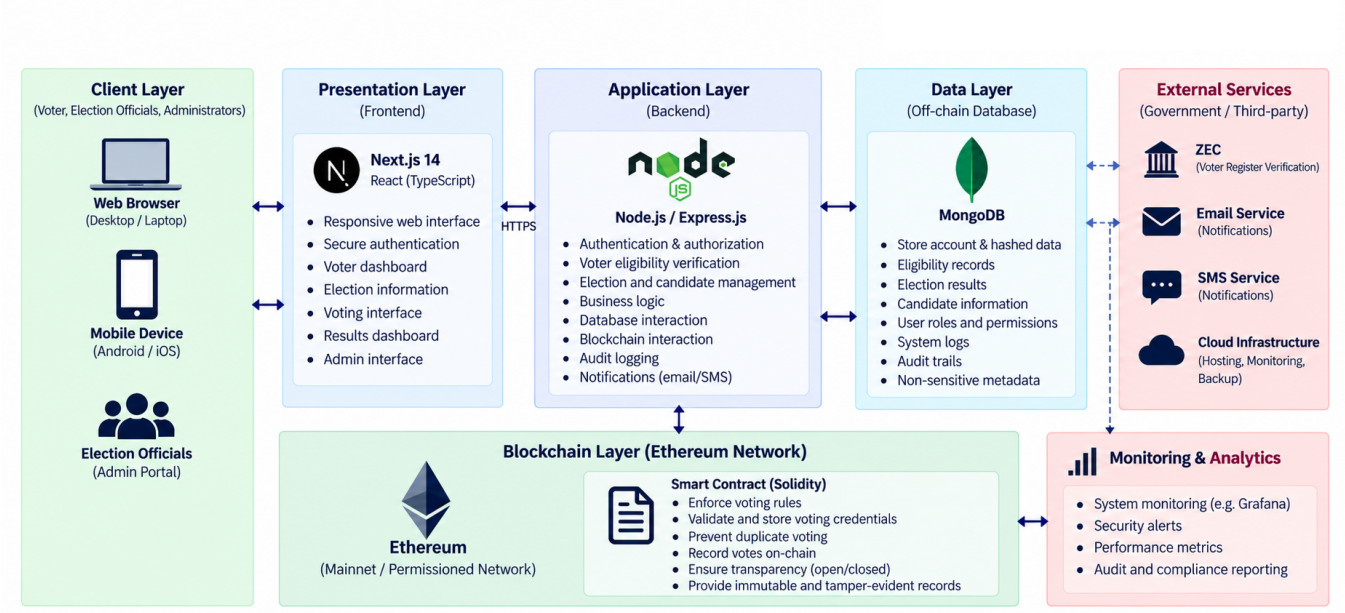

Figure 1: *Proposed blockchain-based electronic voting architecture.*

5.2. Identity, eligibility, and one-person-one-vote

The framework differentiates between the legal nature of identity and the technical aspects of blockchain authentication. Legal identity and a voter-registration program are essential for ensuring that eligible individuals can take part in a particular election. When verified successfully, each voter is issued a single voting entitlement, specific to the election. Rather than being based on the voter's civil ID, it is represented by a privacy-preserving credential, or derived proof. The blockchain is not then the only voter register.

The smart contract verifies the credential proof and validates a one-time marker, such as a nullifier, when a ballot is submitted. Once the marker has been consumed, any attempt to reuse it is rejected. Creating another wallet or blockchain address does not provide an additional voting right because voting power is derived from the verified election credential

rather than from the creation of addresses. This directly mitigates the one-wallet-one-vote weakness found in simplified designs.

The specific credential mechanism is left to the implementation stage. Anonymous credentials, blind signatures, or zero-knowledge proofs should be compared on the basis of unlinkability, revocation, recoverability, computational cost, integration complexity, and metadata correlation. This deliberate openness has advantages over selecting a cryptographic scheme before the implementation constraints and proposed validator model are fully understood.

5.3. Permissioned blockchain and governance

We selected a permissioned blockchain based on the need for identifiable responsibility for blockchain operation, software updates, deployment, key management, incident response, and audit. A public network would put parts of this operational environment under the control of others and would not eliminate the requirement for an electoral-registration system, which is currently reliant on an authoritative process. But a centralized network maintained by a single institution would cause the same trust issues that the architecture is meant to alleviate. The suggested model thus assumes multiple, authoritatively approved and independently accountable stakeholders running validator nodes. This institutional composition needs to be established through Zimbabwe's legal and electoral governance mechanisms, and not by software design. This does not imply that permissioning automatically gives rise to decentralization. Even with multiple nodes in operation, this network is operationally centralized if one organization holds the power to generate and deploy all of the validator credentials, implement the recovery system and release network software updates. All these features were, therefore, an essential part of the framework, including separation of duties, independent auditability, controlled key custody, visibility of software versions deployed, and documented recovery procedures. For an eventual deployment, the consensus mechanism and validator configuration should be selected based on specific requirements for fault tolerance, finality, throughput, network conditions and tolerance for collusion within the governance model.

5.4. Voter interface and blockchain abstraction

The envisaged voter's experience is similar to the one that is usually experienced in a secure digital service. A voter authenticates, verifies eligibility, views current elections, reviews candidates, selects a candidate, confirms the ballot, and receives a privacy-preserving verification reference. The conceptual screens presented are as depicted in Figure 2. The interface deliberately does not require voters to connect MetaMask, manage seed phrases, or pay transaction fees manually in order to keep each step as simple as possible. If complexity is hidden, this can make a project more usable, but it also means the project will have additional architectural responsibilities. Abstracting blockchain operations should not make the application operator a custodial single point of control. The design of cryptographic authorization, signing and handling of credentials must ensure the application operator is unable to generate legitimate voting authorizations without the proper cryptographic authority, or alter the state of the ledger without proper authorization. Future usability studies should consider how well tasks are completed, errors made, the clarity of confirmation information, accessibility and user confidence. While a straightforward interface can improve usability, simplicity alone does not guarantee security.

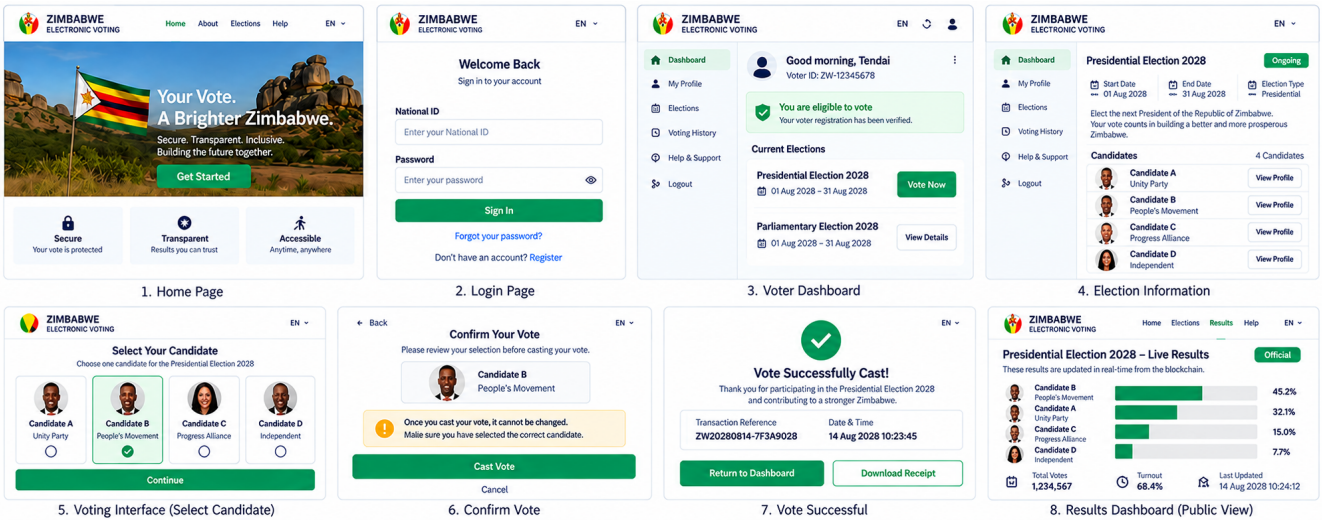

Figure 2: *Proposed user-interface mock-ups. The screens are conceptual and do not represent an implemented system.*

5.5. Threat analysis and residual risk

The threat analysis in Table 2 shows that blockchain should be seen as part of a larger electoral system. A single-use credential will help ensure that the same legitimate credential cannot be used to cast multiple votes, but will not help

prevent issuance of a fraudulent credential if the identity authority is compromised. There are privacy concerns, as the direct linkability can be diminished by identity-ballot separation while timing, network, device, and privileged metadata correlation still poses risks. Although smart contracts can be used to enforce accepted state transitions, possible implementation flaws may exist. While distributed validation may reduce a single validator's ability to control the system, it can be undermined by collusion or compromise of the keys.

Table 2: *Threat-control-residual-risk analysis.*

Threat	Proposed control	Residual risk
Multiple-wallet voting	One verified single-use credential	Fraudulent issuance/credential theft
Identity-ballot linkage	Off-chain identity + unlinkable credential	Timing/network/device correlation
Smart-contract defect	Minimal logic + independent audit	Unknown flaw/deployment-key compromise
Insider abuse	Least privilege + multi-party custody	Privileged collusion
Endpoint compromise	Hardened client + explicit confirmation	Malware may alter voter intent
Coercion/vote buying	Non-revealing receipt; evaluate re-voting	Remote physical/social coercion
Denial of service	Redundancy, monitoring, continuity plan	Large-scale upstream failure

5.6. Interpretation and testable propositions

The analysis leads to the formulation of four propositions that are intended to be tested empirically in the future. First, each valid election credential should generate only one usable electoral nullifier. Second, multiple blockchain addresses should not give rise to multiple voting entitlements. Third, separating identity from ballot state and using privacy-preserving credential proofs should prevent ordinary ledger inspection from revealing a direct link between a voter and a candidate. Fourth, when election-critical state is replicated across independently governed validators, unauthorized changes to an accepted vote should be difficult to perform without detection. Finally, voter interaction with the system should impose less technical burden than a MetaMask-based application. The propositions can be transformed into quantitative research questions. Replay, race condition and multi-address tests should be used in uniqueness testing. There are various ways to measure how well privacy is preserved—formal protocol analysis, linkability experiments, timing correlation experiments, log analysis and metadata analysis. Understanding governance can be achieved by considering validator-failure, compromised-key, and collusion scenarios. Task completion, error rates, accessibility checks, and validated usability instruments are measures of usability. Confirmation latency, throughput of confirming under concurrent workloads, database load, node resource utilization, and recovery following a network or service failure are metrics for evaluating performance.

5.7. Implications for Zimbabwe

The key implication for Zimbabwe is that blockchain should be used to enhance selected functions of the electoral system rather than being treated as the electoral system itself. The eligibility of voters, acceptable types of identity papers, the accreditation of observers, the resolution of disputes and whether certain types of electronic or remote voting are legal continue to be institutional and legal concerns. The ledger can facilitate more visible and verifiable state transitions, but it cannot make an otherwise unauthorized electoral process legitimate.

The same distinction applies to possible future remote and diaspora voting. Also, the architecture may be technically able to provide remote access, but that does not necessarily mean it is legally authorized or election-ready. If such an extension were implemented, it would necessitate an authoritative way of determining who could vote, secure and efficient ways to deliver and access voting credentials and ballots, secure tools to protect the endpoints, easily understandable ballot user interfaces, voter privacy controls, and explicit treatment of situations in which a vote may be coerced away from a controlled polling site. Remote participation is not an automatic consequence of making the system web-based; we therefore treat it as a future use case requiring separate legal and security analysis.

The viability of the framework is also related to infrastructure and inclusion. A well-functioning design for people with secure broadband access, modern devices, and some knowledge of technology and privacy issues could hinder rather than help those who do not have such access. Prototype testing should thus take into account poor connectivity conditions and the most prevalent mobile devices, as well as recovery of interrupted sessions, accessibility, language clarity, processes for assistance and support, and the continued effectiveness of the available alternative ways to vote. These issues are not stand-alone concerns to be resolved after the technology is completed; they are security and usability issues within the technology itself.

Public transaction hashes alone are not sufficient evidence of electoral transparency. Observers will require information about the audits that can be understood and that can be checked independently. Therefore, an implementation in the future should include records of election-state changes, validators who have participated in the process, rejected and repeated election-state changes, software versions, and tally commitments without revealing information about a voter or revealing

ballot choice. Ideally, these checks should be repeatable using independent tools without requiring privileged access to the application database.

Another area where technical design and governance interface is key management. It is preferable not to have validator keys, contract-deployment keys, credential-issuer keys, and administrative authentication secrets owned by the same entity or service. A future implementation should assess the following features: hardware-backed key storage, threshold authorization, documented key ceremonies, key rotation, key revocation, and emergency recovery procedures. Most importantly, when a high-value key is compromised, a predefined technical and institutional response should be followed rather than leaving the decision to one administrator.

Finally, the proposed architecture needs to be evaluated in comparison with simpler architectures. However, blockchain isn't necessarily the ideal solution just because it uses a distributed ledger. Similar benefits could be achieved with a conventional architecture that relies on signed append-only logs, replicated databases, or other cryptographically verifiable means, with lower operational complexity. For a future prototype, it should be tested with the same criteria to compare alternatives such as integrity, audit independence, privacy, latency, operating cost, recovery, administrative complexity and governance. If a non-blockchain design can meet the requirements more effectively, this would be a valid and important result.

5.8. Limitations and future work

The main limitation is that the framework is conceptual. No working election platform, smart-contract benchmark, validator deployment, penetration test, user study, or national-scale simulation is reported. The architecture therefore cannot provide guarantees about transaction latency, uptime, reliability, voter satisfaction, national throughput, or proven resistance to attack. The privacy-preserving credential protocol, consensus configuration, number of validators, key ceremony, and recovery procedure also remain to be determined.

Future implementation should proceed in stages. More specifically, the initial step should involve creating the smallest possible prototype to validate credential issuance, ballot authorization, the identification of duplicate ballots, transition to and from election states, and ballot tally consistency. The second phase should involve a comprehensive security audit of the credential protocol and smart contracts, including static, property-based, and formal analysis and testing by independent researchers. The third stage should evaluate throughput, latency, node failure, network partition, recovery, and resource consumption under realistic loads. Only after these technical phases should usability and accessibility, governance and legal requirements, and a fully non-binding or legally authorized pilot be considered.

Future research should also compare the proposed permissioned approach with alternative architectures. In certain contexts a traditional non-chain-based cryptographically verifiable system could offer similar security and lower complexity of operation. The future challenge is not simply to show that the proposed blockchain system is correct, but to determine which properties are strengthened by the ledger and how alternative architectures compare.

Any future pilot would also require a public communication strategy. Voters should not be expected to trust the system merely because it uses blockchain. Explanations should emphasize observable protections, including how eligibility is determined, why a second wallet does not create an additional vote, what data is and is not recorded on the ledger, how vote inclusion can be verified without revealing ballot content, who operates validation nodes, and what happens when components fail. Communication can reduce the gap between technical security and public understanding, but it cannot compensate for inadequate security.

Finally, the framework should be evaluated incrementally rather than through a single deployment. Contract and protocol defects can be identified in laboratory testing; capacity limits can be examined through simulation; usability problems can be identified in controlled studies; operational defenses can be tested through red-team exercises; and, only after these stages, tightly governed legal pilots can examine governance and support challenges. Progression to each later stage should depend on explicit criteria established in the preceding stage. This creates an evidence trail and prevents a successful demonstration from being treated as proof of readiness for national deployment.

Explicit stopping criteria should also be included in these evaluation stages. A prototype should not progress to a pilot while unresolved defects could endanger ballot secrecy, permit duplicate voting, undermine independent verification, or create unacceptable administrative control. Performance targets should be based on realistic election workloads rather than arbitrary values. Security issues, test failures, and operational failures should be documented alongside positive results so that residual risk can be evaluated. For electoral technology, evidence-based progression is particularly important because technical failures can affect both service quality and confidence in procedural legitimacy.

6. Conclusion

This study developed a blockchain-based electronic voting framework for Zimbabwe by starting from electoral requirements rather than from blockchain technology itself. The resulting design separates civil identity from ballot processing, derives voting rights from a single-use election credential rather than a wallet address, restricts smart-contract

logic to election-critical rules, and distributes ledger responsibilities across multiple accountable stakeholders. The voter-facing design also removes the need for ordinary users to manage MetaMask, cryptocurrency, or blockchain keys.

Our analysis suggests that the strongest justification for using blockchain in this framework is not that it makes an election automatically secure, but that it can provide replicated, tamper-evident election state and consistent enforcement of selected rules across independently governed validators. The same analysis also identifies clear boundaries. Ballot secrecy, coercion resistance, endpoint security, availability, legal acceptability, public trust, and correct voter registration depend on controls outside the ledger. These limitations are central to the framework because overstating what blockchain can guarantee would weaken rather than strengthen the design.

The framework should therefore be treated as a testable design artefact. Before any real electoral use is considered, future work must demonstrate the credential protocol, test one-person-one-vote under adversarial conditions, audit the smart contracts, evaluate privacy and linkability, measure performance and recovery under realistic loads, assess usability and accessibility, and examine governance and legal requirements. Only evidence from these stages can show whether the additional complexity of a permissioned blockchain is justified for Zimbabwe. The contribution of this study is to make those design choices, trust assumptions, and evaluation requirements explicit.

Author Declarations

Author Contributions (CRediT): Marvellous Courage Tshuma: Conceptualization, Methodology, Formal Analysis, Investigation, Visualization, Writing - Original Draft, Writing - Review & Editing. Edwin Farai Madzuwo: Conceptualization, Methodology, Validation, Writing - Original Draft, Writing - Review & Editing. Tatenda Benjamin Manyuchi: Conceptualization, Methodology, Validation, Writing - Original Draft, Writing - Review & Editing. All authors have read and approved the submitted version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No datasets were generated or analyzed during this conceptual study. The proposed framework and analytical artefacts are described within the manuscript.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Ethics Statement: Not applicable. This conceptual study did not involve human participants, animals, or identifiable personal data.

Declaration of Generative AI: During the preparation and revision of this manuscript, the authors used ChatGPT to assist with language improvement and the revision of selected explanatory passages. The authors reviewed, verified, and take full responsibility for the research concepts, methodological decisions, arguments, interpretations, conclusions, accuracy, and integrity of the final manuscript.

References

[1] N. Kshetri and J. Voas, "Blockchain-enabled e-voting," IEEE Software, vol. 35, no. 4, pp. 95-99, 2018, doi: https://doi.org/10.1109/MS.2018.2801546.

[2] H. O. Ohize et al., "Blockchain for securing electronic voting systems: A survey of architectures, trends, solutions, and challenges," Cluster Computing, vol. 28, art. 132, 2025, doi: https://doi.org/10.1007/s10586-024-04709-8.

[3] M. Alown, M. S. Kiraz, and M. A. Bingöl, "Enhancing democratic processes: A survey of DRE, Internet, and blockchain in electronic voting systems," IEEE Access, vol. 13, pp. 20512-20545, 2025, doi: https://doi.org/10.1109/ACCESS.2025.3531349.

[4] B. Adida, "Helios: Web-based open-audit voting," in Proc. 17th USENIX Security Symposium, 2008, pp. 335-348.

[5] D. Chaum, P. Y. A. Ryan, and S. Schneider, "A practical voter-verifiable election scheme," in Computer Security - ESORICS 2005, 2005, pp. 118-139, doi: https://doi.org/10.1007/11555827_8.

[6] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in Financial Cryptography and Data Security, 2017, pp. 357-375, doi: https://doi.org/10.1007/978-3-319-70972-7_20.

[7] Y. Shi, K. Fan, Y. Bai, C. Zhang, K. Zhang, H. Li, and Y. Yang, "Blockchain-based anonymous and self-tallying voting with time-bounded ballot secrecy," Computer Networks, vol. 263, art. 111217, 2025, doi: https://doi.org/10.1016/j.comnet.2025.111217.

[8] S. Yousfi and M. Chaieb, "Building a scalable and coercion-resistant e-voting system using blockchain," SN Computer Science, vol. 6, art. 561, 2025, doi: https://doi.org/10.1007/s42979-025-04088-w.

[9] Y.-X. Kho, S.-H. Heng, S.-Y. Tan, and J.-J. Chin, "A provably secure coercion-resistant e-voting scheme with confidentiality, anonymity, unforgeability, and CAI verifiability," PLOS ONE, vol. 20, no. 6, e0324182, 2025, doi: https://doi.org/10.1371/journal.pone.0324182.

[10] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," MIS Quarterly, vol. 28, no. 1, pp. 75-105, 2004, doi: https://doi.org/10.2307/25148625.

[11] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," Journal of Management Information Systems, vol. 24, no. 3, pp. 45-77, 2007, doi: https://doi.org/10.2753/MIS0742-1222240302.

[12] M.-V. Vladucu, Z. Dong, J. Medina, and R. Rojas-Cessa, "E-voting meets blockchain: A survey," IEEE Access, vol. 11, pp. 23293-23308, 2023, doi: https://doi.org/10.1109/ACCESS.2023.3253682.

[13] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," International Journal of Web and Grid Services, vol. 14, no. 4, pp. 352-375, 2018, doi: https://doi.org/10.1504/IJWGS.2018.095647.