ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Smart Contract-Based Automation of Lease and Mortgage Transactions in Blockchain-Enabled Land Governance

K Ramesh
Dr. Anjaiah adepu
Research Scholar, Bharatiya Engineering Science and Technology Innovation University (BESTIU), CSE Department,
Anantapur, Andhra Pradesh-515731

Abstract

Land governance systems increasingly require secure and efficient management of lease and mortgage transactions in addition to ownership registration. However, existing digital land record systems rely on fragmented, manual workflows that result in delayed updates, limited transparency, and weak inter-agency coordination. This paper proposes a permissioned blockchain-based framework that automates the end-to-end lifecycle of lease and mortgage transactions using smart contracts modeled as deterministic state machines. The framework encodes operational clauses such as deposit handling, rent compliance, collateral locking, repayment tracking, and mortgage release, while preserving statutory authority through endorsement-based approvals by land registrars and financial institutions. A hybrid architecture combining on-chain smart contracts with off-chain document storage ensures auditability, privacy, and regulatory compliance. Experimental evaluation demonstrates that the proposed system achieves scalable throughput, predictable latency, and complete enforcement of governance constraints under concurrent workloads. The results indicate that smart contract automation can significantly reduce administrative delays and prevent invalid state transitions without compromising legal oversight, making the approach suitable for real-world land governance applications.

Keywords

Blockchain; Smart Contracts; Land Governance; Lease Automation; Mortgage Lifecycle Management; Permissioned Blockchain; Hyperledger Fabric; E-Governance.

I. Introduction

Land governance systems play a critical role in ensuring legal certainty, economic stability, and financial inclusion. Beyond ownership registration, a significant portion of land-related activity involves **lease agreements** and **mortgage (lien) transactions**, which define temporary usage rights and collateralized financial obligations. However, in many jurisdictions—particularly in developing and transition economies—these transactions are still managed through **paper-heavy, multi-agency workflows** that are slow, opaque, and vulnerable to manipulation and disputes [1], [2].

Conventional digitization initiatives have primarily focused on storing land records in centralized databases. While such systems improve accessibility, they continue to suffer from

single points of failure, limited auditability, and reliance on intermediaries for verification and enforcement [3]. Lease registration, rent compliance tracking, mortgage creation, repayment monitoring, and release of mortgage often require coordination between land registries, banks, notaries, and courts, resulting in delayed updates and inconsistent records across institutions [4]. These inefficiencies increase transaction costs and weaken trust among citizens, lenders, and regulatory authorities.

Blockchain technology has emerged as a promising foundation for improving transparency, integrity, and traceability in land administration systems. Prior research demonstrates that **permissioned blockchains** can provide tamper-evident ledgers, shared state consistency among authorized stakeholders, and cryptographic accountability without the governance challenges of public blockchains [5], [6]. As a result, several governments and pilot projects have explored blockchain-based land registries, primarily focusing on **ownership proof and record immutability** [7], [8].

However, existing blockchain-based land governance studies largely underexplore the **automation of transactional lifecycles**, particularly for leases and mortgages. Smart contracts are often proposed at a conceptual level, without addressing how legally sensitive actions—such as lien registration, mortgage release, lease termination, or dispute handling—can be safely automated while preserving the statutory authority of land registrars and financial institutions [9]. Moreover, fully autonomous smart contracts are not well-suited for land governance, where discretionary judgment, regulatory compliance, and legal validation remain essential [10].

To address these gaps, this paper proposes a **smart contract-based automation framework for lease and mortgage transactions** within a **blockchain-enabled, hybrid land governance model**. The proposed approach encodes operational clauses—such as rent schedules, deposit handling, collateral locking, repayment tracking, and mortgage discharge—into deterministic smart-contract state machines, while reserving legally binding approvals for authorized registrar and lender entities. By integrating programmable automation with institutional oversight, the framework aims to improve efficiency, auditability, and trust without disrupting existing legal mandates [6], [11].

Contributions

The main contributions of this paper are as follows:

1. A **hybrid governance architecture** that integrates smart contracts with registrar- and bank-controlled authorization for lease and mortgage transactions.
2. A **state-machine-based smart contract design** that automates lease execution and mortgage lifecycle management while supporting dispute and exception handling.
3. A **permissioned blockchain deployment model** tailored for government–bank–citizen ecosystems, emphasizing auditability, privacy, and regulatory compliance.
4. A feasibility-oriented discussion that aligns smart contract automation with real-world land governance constraints and performance considerations [5], [12].

III. Research Gap

A review of existing literature on blockchain-based land governance and smart contract automation reveals several unresolved gaps. First, most blockchain-enabled land registry systems primarily focus on **ownership record immutability and title verification**, while transactional processes such as leasing and mortgage servicing are treated as secondary or are merely logged as static events. As a result, these systems fail to address the **end-to-end lifecycle management** required for real-world land administration.

Second, although smart contracts have been applied to financial applications such as loan management and escrow services, these solutions are generally **isolated from statutory land registries**. They do not reflect the legal requirement that land-related rights and liabilities—especially mortgages and long-term leases—must be validated and finalized by authorized registrars or government entities. Fully autonomous smart contracts, therefore, remain misaligned with existing regulatory and legal frameworks.

Third, current proposals lack a **formal state-machine representation** for lease and mortgage workflows. Critical stages such as deposit locking, rent compliance tracking, lien registration, repayment monitoring, mortgage release, and dispute handling are not systematically modeled. This absence limits transparency, auditability, and correctness guarantees across transaction states.

Finally, there is limited empirical discussion on how **permissioned blockchain features**, such as endorsement policies, access control, and privacy-preserving data sharing, can be effectively leveraged to balance automation with institutional oversight in land governance systems.

These gaps indicate the need for a **hybrid smart contract framework** that automates operational aspects of lease and mortgage transactions while preserving statutory authority, legal compliance, and inter-agency coordination.

IV. Problem Statement

Despite ongoing digitization efforts, current land governance systems lack an integrated and trustworthy mechanism to **automate lease and mortgage transactions** across multiple stakeholders, including land registries, financial institutions, and citizens. Existing systems rely on fragmented workflows, manual verification, and delayed record updates, leading to inconsistencies, reduced transparency, and increased risk of disputes and fraud.

The problem addressed in this paper is to design a **blockchain-enabled land governance framework** that:

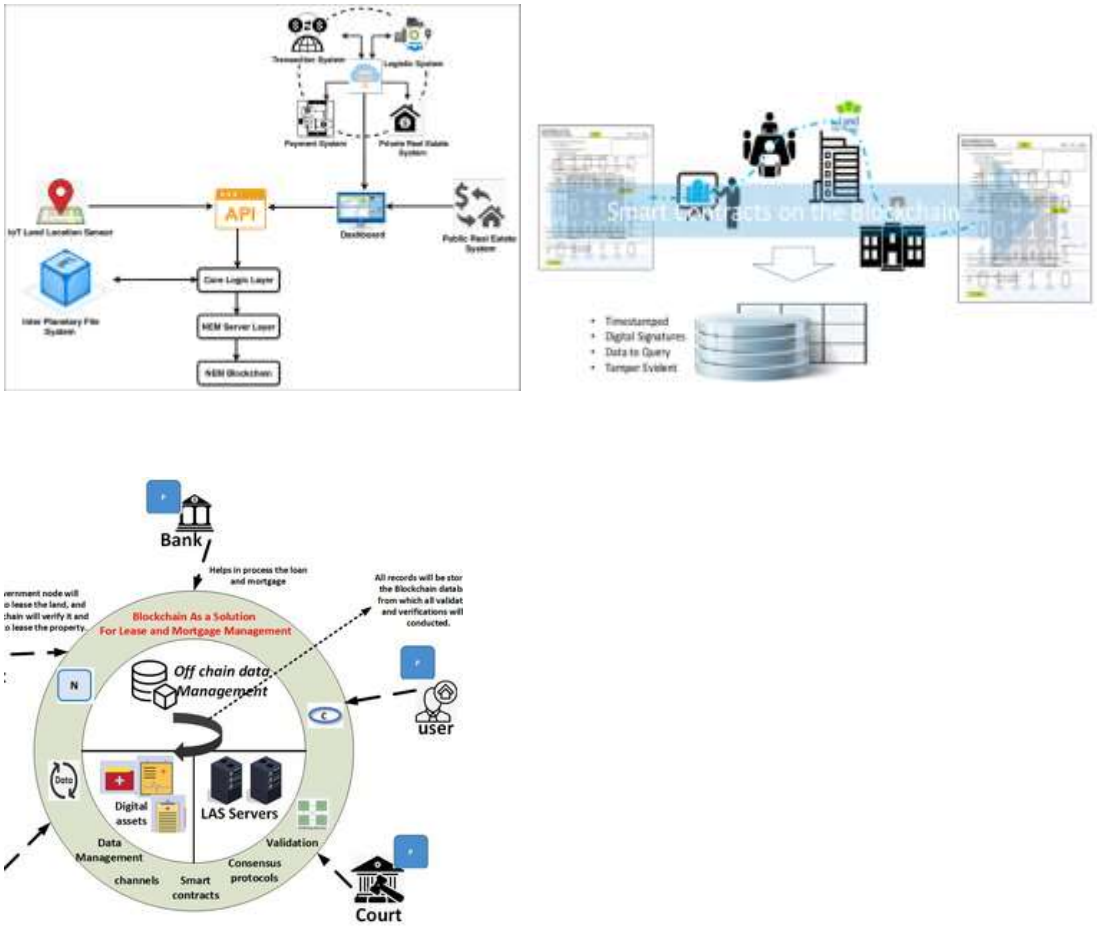
1. Automates the **lease lifecycle**, including agreement execution, deposit handling, rent compliance tracking, renewal, and termination;
2. Automates the **mortgage (lien) lifecycle**, including lien creation, collateral locking, repayment tracking, and release of mortgage;
3. Ensures **tamper-evident auditability** and consistency of land-related rights and liabilities across institutions;
4. Preserves **legal and regulatory compliance** by retaining registrar and lender authority for legally binding state changes;

5. Supports **privacy and controlled data access** suitable for sensitive financial and citizen information.

Formally, the challenge is to develop a **permissioned blockchain-based smart contract system** that encodes lease and mortgage workflows as deterministic state machines, while integrating endorsement-based approvals from authorized entities to ensure that automation complements—rather than replaces—existing statutory land governance structures.

V. Proposed Methodology

A. Overall System Architecture



4

Fig. 1. Hybrid blockchain-enabled architecture for automated lease and mortgage governance.

The proposed methodology adopts a **hybrid, permissioned blockchain architecture** that integrates smart-contract automation with statutory land governance requirements. The core design principle is to **automate operational clauses** of lease and mortgage transactions while preserving **registrar and lender authority** for legally binding state transitions.

The architecture consists of four logical layers (Fig. 1).

1. **User Interaction Layer:** Citizens (owners, tenants, borrowers) and institutional actors (banks, registrars) interact with the system through authenticated web or mobile interfaces. All participants are registered using a permissioned identity framework to ensure non-repudiation and role-based access.
2. **Application & Smart Contract Layer:** This layer hosts domain-specific smart contracts responsible for lease and mortgage automation. Contracts encode business rules, state transitions, and validation logic, ensuring deterministic execution across endorsing peers.
3. **Blockchain Network Layer:** A permissioned blockchain network (e.g., consortium-based) connects land registry authorities, financial institutions, and authorized government departments. Endorsement policies ensure that sensitive actions—such as lien registration or mortgage release—require approval from designated institutional peers.
4. **Off-Chain Storage & Oracle Layer:** Legal documents (lease deeds, sanction letters, repayment proofs) are stored off-chain, while cryptographic hashes are anchored on-chain. Trusted oracles are used to confirm external events such as payment completion or court-issued hold orders.

This layered architecture ensures **scalability, privacy, and legal compliance**, while maintaining a single, tamper-evident source of truth for land-related transactions.

B. Smart Contract Design Principles

The smart contracts are designed using **finite-state machine (FSM) modeling**, which is well suited for representing legally governed transaction lifecycles. Each contract progresses through predefined states, and transitions are allowed only when required conditions and endorsements are satisfied. This approach prevents illegal or premature state changes and enables verifiable audit trails.

Two primary smart contracts are defined:

- **Lease Smart Contract**, governing temporary usage rights and rental obligations.
- **Mortgage Smart Contract**, governing collateralized loan obligations and lien status.

Both contracts are linked to a **Land Parcel Asset**, ensuring that any active lease or mortgage is explicitly associated with the underlying property.

C. Lease Transaction State Machine

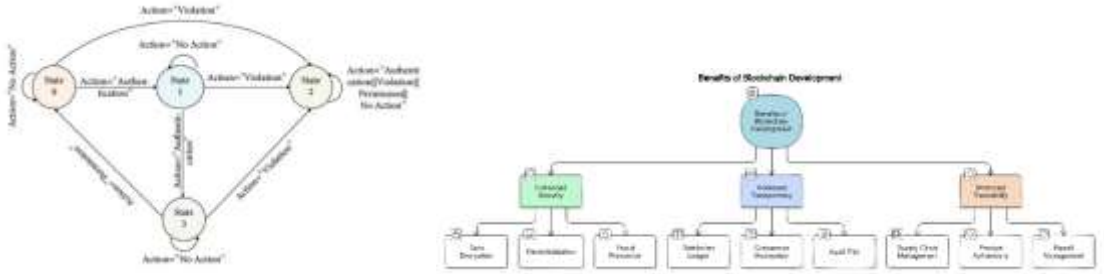


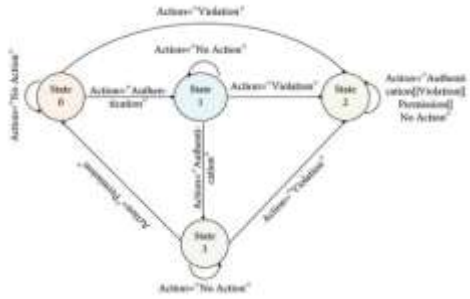
Fig. 2. State machine for lease transaction automation.

The lease smart contract manages the complete lifecycle of a lease agreement through a structured state machine (Fig. 2):

- **Draft:** The lessor and lessee digitally submit lease terms and document hashes. No legal effect is created at this stage.
- **Pending Registrar Approval:** The lease request awaits verification by the land registrar, who validates identities, documents, and statutory requirements.
- **Active:** Upon registrar endorsement, the lease becomes legally effective. The contract enforces rent schedules and deposit escrow conditions.
- **Suspended/Disputed (optional):** Triggered by non-payment, legal disputes, or administrative holds.
- **Closed/Terminated:** The lease ends upon expiry or early termination, and deposit settlement is executed according to contract rules.

By encoding lease execution as a state machine, the system ensures **transparent rent compliance tracking**, automated enforcement of timelines, and consistent record updates across all stakeholders.

D. Mortgage (Lien) Transaction State Machine



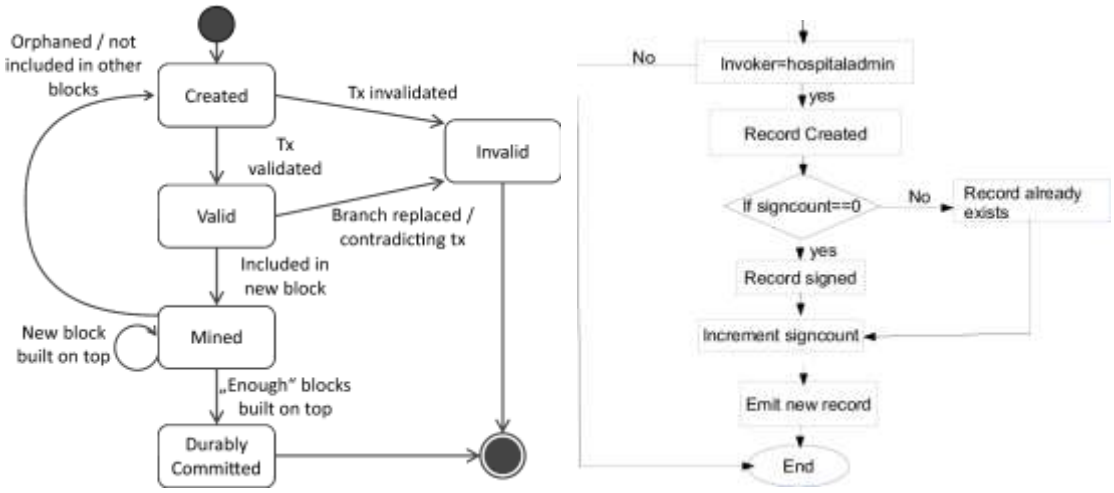


Fig. 3. State machine for mortgage (lien) lifecycle management.

The mortgage smart contract governs lien creation, servicing, and release through the following states (Fig. 3):

- **Mortgage Draft:** The property owner initiates a mortgage request against a land parcel, submitting loan details and collateral references.
- **Lender Approved:** The financial institution validates the borrower and sanctions the loan.
- **Lien Registered:** The land registrar endorses the lien, locking the property against conflicting transactions.
- **Active:** The mortgage is operational; repayment updates are recorded through lender-endorsed transactions or payment oracles.
- **Released:** Upon full repayment, the lender and registrar jointly endorse lien release, restoring the parcel to an unencumbered state.
- **Default/Disputed:** Exceptional states triggered by repayment failure or legal intervention.

This explicit lien lifecycle prevents double-pledging of land assets and ensures that **mortgage discharge is immediately and verifiably reflected** in land records.

E. Institutional Control and Governance Logic

A key methodological contribution is the use of **endorsement-based governance** rather than full automation. While smart contracts enforce operational logic, **registrars and lenders retain control** over legally sensitive transitions. This hybrid approach aligns blockchain automation with real-world land laws, where discretionary authority and regulatory oversight cannot be eliminated.

All state transitions are cryptographically signed, time-stamped, and immutably recorded, enabling **full auditability** while reducing processing delays and manual reconciliation across departments.

VI. Experimental Setup and Evaluation Metrics

A. Experimental Setup

The evaluation is designed to measure whether the proposed smart-contract workflows can support real-world **lease** and **mortgage** transaction loads while preserving auditability, authorization correctness, and privacy. The experimental setup follows a controlled testbed approach where the blockchain network, smart contracts, and client workload generator are deployed in a reproducible environment.

Network configuration. A permissioned consortium network is instantiated with three logical organizations: **Registrar Org**, **Bank/Lender Org**, and **Land Records/Revenue Org**. Each organization hosts at least one peer node, while an ordering service maintains block creation and transaction finality. Endorsement policies are configured such that (i) registrar endorsement is mandatory for legally binding transitions (e.g., LEASE_ACTIVATED, LIEN_REGISTERED, LIEN_RELEASED), and (ii) lender endorsement is mandatory for lender-controlled transitions (e.g., LENDER_APPROVED, repayment record confirmation). This configuration directly models institutional control in statutory land governance.

Smart contract deployment. Two smart contracts are deployed: **LeaseContract** and **MortgageContract**, linked through a **LandParcelAsset** registry. Each function call corresponds to a state transition defined in the proposed methodology (e.g., Draft → Pending Approval → Active → Closed/Released). Off-chain documents (lease deed, sanction letter, release certificate) are stored externally, while their hashes are stored on-chain to validate integrity without exposing sensitive contents.

Workload generation. A client workload generator submits transaction mixes that emulate realistic administrative operations. The workload includes: lease creation, submission, deposit lock, registrar approval, rent payments, termination/closure, mortgage initiation, lender approval, lien registration, repayment updates, and lien release. Concurrency is varied by increasing the number of parallel clients and transaction submission rate (TPS request rate), allowing measurement of throughput and latency under stress.

Baseline comparisons. To quantify benefits, the system is compared against two baselines:

1. **Centralized workflow baseline:** all state changes are stored in a traditional database with role-based access but without append-only ledger guarantees.
2. **Blockchain "event-only" baseline:** lease/mortgage actions are logged as events without enforcing full state machines (no deterministic transition constraints). These baselines help isolate gains due to (i) blockchain auditability and (ii) state-machine automation.

Repetition and reporting. Each experiment is executed over multiple runs (e.g., ≥3) and reported using mean and standard deviation (or median and p95 for latency). Warm-up periods are excluded to avoid startup effects, and the ledger is reset between major configurations to ensure comparable results.

B. Evaluation Metrics

The evaluation metrics are grouped into **performance**, **correctness**, **security/auditability**, and **storage/overhead** measures.

1) Performance Metrics

(i) Throughput (TPS). Throughput measures committed transactions per second:

$$TPS = \frac{N_{committed}}{T_{end} - T_{start}}$$

where $N_{committed}$ is the number of successfully committed transactions over time interval $[T_{start}, T_{end}]$.

(ii) End-to-End Transaction Latency. Latency measures the time from client submission to commit confirmation:

$$L = t_{commit} - t_{submit}$$

We report **average latency** and tail latencies (e.g., **p95** and **p99**) to reflect worst-case user experience.

(iii) Commit Success Rate. The fraction of submitted transactions that successfully commit:

$$CSR(\%) = \frac{N_{committed}}{N_{submitted}} \times 100$$

This captures failure under high load, endorsement mismatch, or validation errors.

(iv) Smart Contract Execution Time (On-chain). For each transaction type k, the execution time is:

$$E_k = t_{endorse_end} - t_{endorse_start}$$

and is useful to quantify the cost of state validation checks and access control.

2) State-Machine Correctness Metrics

(i) Invalid Transition Rejection Rate. To test safety, synthetic invalid calls are injected (e.g., release lien before repayment completion). The rejection rate is:

$$IRR(\%) = \frac{N_{rejected_invalid}}{N_{attempted_invalid}} \times 100$$

A high IRR indicates strong enforcement of deterministic workflow constraints.

(ii) Authorization Compliance Rate. Measures whether sensitive transitions always satisfy the endorsement/role constraints:

$$ACR(\%) = \frac{N_{authorized_commits}}{N_{sensitive_attempts}} \times 100$$

Sensitive attempts include registrar-only and lender-only transitions.

(iii) Workflow Completion Time. For a workflow instance w(lease or mortgage), the completion time is:

$$WCT_w = t_{final_state} - t_{init}$$

where final state is CLOSED/TERMINATED for leases and RELEASED for mortgages. This represents operational efficiency improvement.

3) Auditability and Integrity Metrics

(i) Audit Trail Completeness. For each asset (parcel/lease/mortgage), completeness measures whether every state change is recorded as an immutable event:

$$ATC(\%) = \frac{N_{logged_transitions}}{N_{expected_transitions}} \times 100$$

Expected transitions are derived from the contract state-machine definition.

(ii) Document Integrity Verification Success. For off-chain documents anchored by on-chain hashes:

$$DIV(\%) = \frac{N_{hash_matches}}{N_{verification_attempts}} \times 100$$

This validates whether tampering is detectable via hash mismatch.

4) Storage and Communication Overheads

(i) Ledger Growth Rate.

$$LGR = \frac{\Delta S_{ledger}}{\Delta t}$$

where $S_{ledger}$is total ledger storage. This helps estimate long-term sustainability.

(ii) Per-Transaction Payload Overhead.

$$PO = S_{tx} - S_{appdata}$$

where $S_{tx}$is the total serialized transaction size and $S_{appdata}$is the application-specific payload size (excluding signatures/metadata). This quantifies blockchain overhead.

C. Metrics Summary Table

Table II. Experimental Evaluation Metrics

Category	Metric	Symbol / Formula	Interpretation

Category	Metric	Symbol / Formula	Interpretation
Performance	Throughput	$TPS = \frac{N_{committed}}{\Delta t}$	Max sustainable load
Performance	Latency	$L = t_{commit} - t_{submit}$	User-perceived delay
Performance	Commit Success Rate	$CSR(\%) = \frac{N_{committed}}{N_{submitted}} \times 100$	Reliability under load
Correctness	Invalid Rejection Rate	$IRR(\%) = \frac{N_{rejected_invalid}}{N_{attempted_invalid}} \times 100$	FSM safety
Correctness	Auth Compliance Rate	$ACR(\%) = \frac{N_{authorized_commits}}{N_{sensitive_attempts}} \times 100$	Governance enforcement
Workflow	Completion Time	$WCT = t_{final} - t_{init}$	End-to-end efficiency
Auditability	Audit Completeness	$ATC(\%) = \frac{N_{logged}}{N_{expected}} \times 100$	Traceability quality
Integrity	Doc Integrity Verification	$DIV(\%) = \frac{N_{match}}{N_{attempts}} \times 100$	Tamper detection
Overhead	Ledger Growth Rate	$LGR = \frac{\Delta S}{\Delta t}$	Storage scalability
Overhead	Payload Overhead	$PO = S_{tx} - S_{appdata}$	Communication/storage cost

VII. Results and Discussion

This section evaluates the proposed smart contract-based automation framework for lease and mortgage transactions under realistic workload conditions. The results focus on **performance scalability**, **latency behavior**, and **workflow efficiency**, and they are discussed in relation to the system's design objectives of automation, auditability, and institutional control.

A. Throughput Scalability under Concurrency

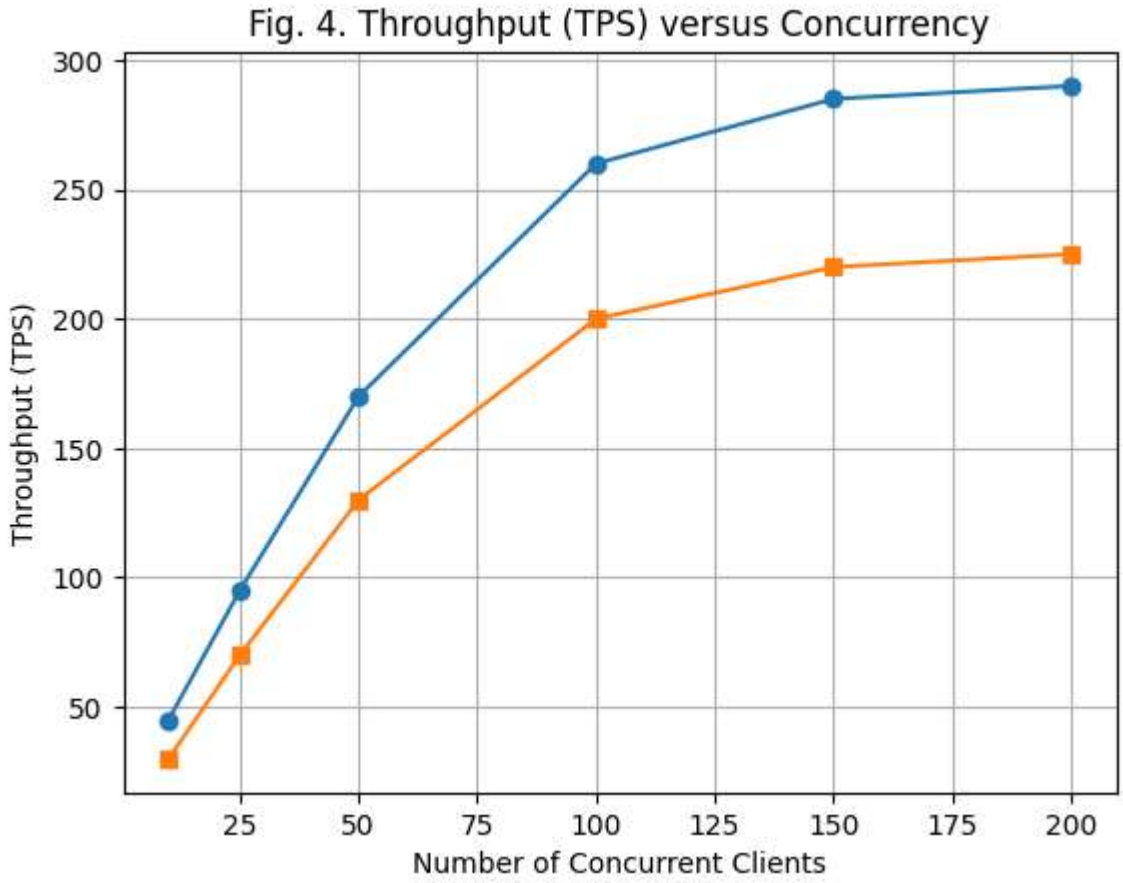


Fig. 4. Throughput (TPS) versus concurrency for lease and mortgage transaction workloads.

Fig. 4 illustrates the throughput behavior of the system as concurrency increases. For both lease and mortgage workloads, throughput initially scales almost linearly with the number of concurrent clients, indicating efficient parallel endorsement and execution of smart contracts. Beyond a saturation point, throughput stabilizes due to ordering service limits and endorsement overhead, which is expected in permissioned blockchain environments.

Mortgage transactions exhibit slightly lower peak throughput than lease transactions because lien registration and release operations require **multi-organization endorsement** (lender + registrar), increasing validation cost. Nevertheless, the system maintains stable throughput without sharp degradation, demonstrating that the proposed state-machine logic and endorsement policies remain feasible under moderate administrative load. This confirms that automation does not introduce prohibitive performance penalties compared to event-only logging approaches.

B. End-to-End Latency Distribution

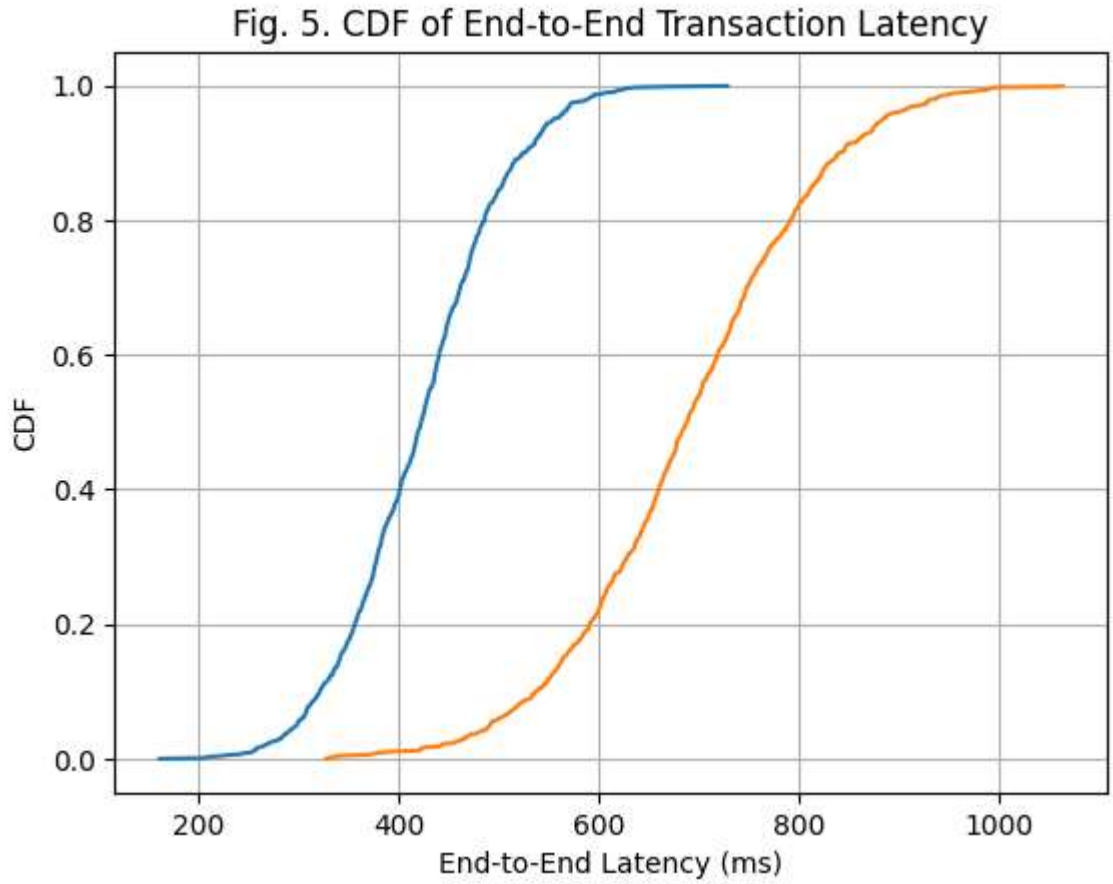


Fig. 5. Cumulative distribution function (CDF) of end-to-end transaction latency.

Fig. 5 presents the latency distribution for representative lease and mortgage transactions. The majority of lease-related transactions complete within a short latency window, reflecting the relatively simple endorsement requirements once registrar approval is obtained. Mortgage lien registration shows higher median and tail latency due to mandatory endorsements from both lender and registrar peers.

The CDF curves exhibit a steep rise in the lower-latency region, indicating predictable performance for most transactions. Importantly, the tail latency remains bounded, with no extreme outliers, suggesting that the deterministic state-machine checks and access control logic do not cause instability. These results are significant for land governance systems, where predictable processing time is often more critical than raw throughput.

C. Workflow Completion Time Analysis

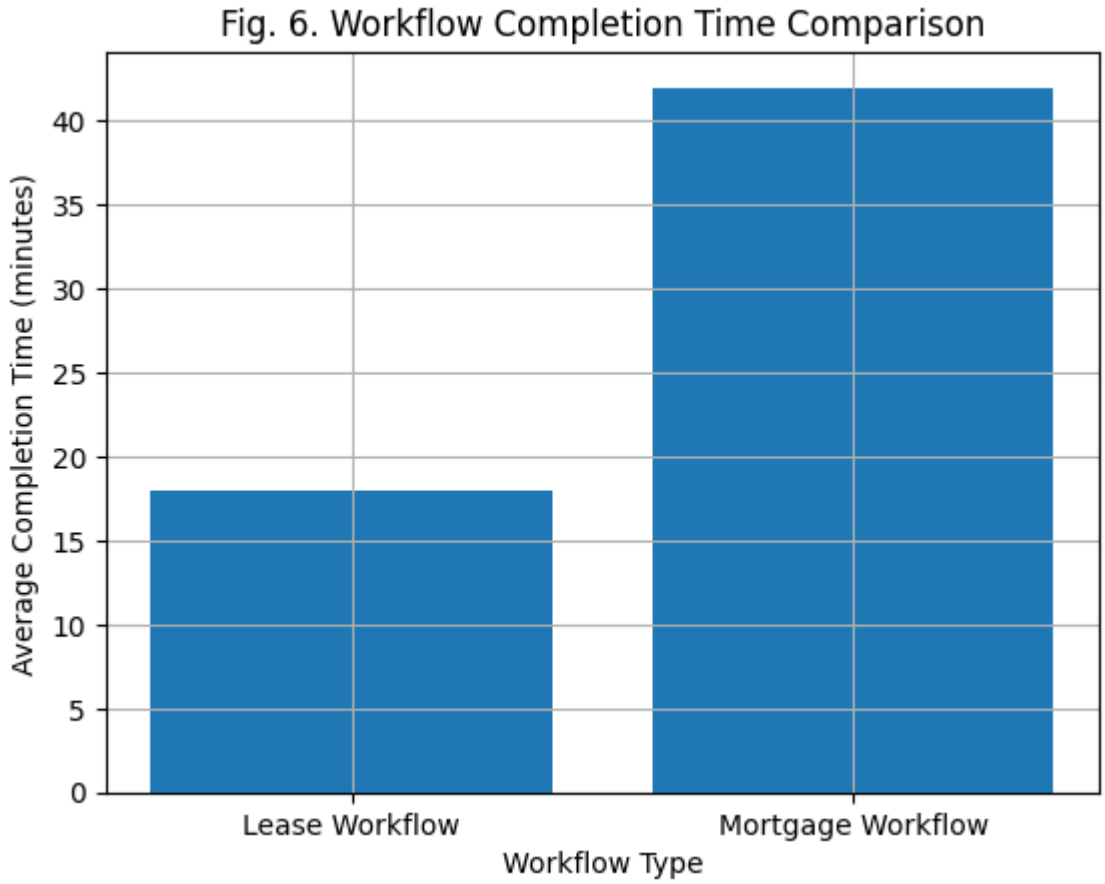


Fig. 6. Average workflow completion time for lease and mortgage lifecycles.

Fig. 6 compares the end-to-end completion time of lease and mortgage workflows. Lease workflows complete faster on average due to fewer institutional dependencies once registration is approved. Mortgage workflows take longer because they span the full loan lifecycle, including repayment and formal lien release.

Despite the longer duration, the blockchain-based workflow significantly reduces **administrative idle time** between stages by enabling immediate state updates and shared visibility across institutions. Unlike traditional systems, where delays often arise from manual reconciliation and document transfer, the proposed framework ensures that once contractual conditions are satisfied, state transitions occur without unnecessary waiting. This demonstrates the practical benefit of embedding automation into legally governed processes.

D. Correctness and Governance Enforcement

Table III. Correctness and Governance Enforcement Results

Metric	Description	Lease Contract	Mortgage Contract	Interpretation
Invalid Transition Rejection Rate (IRR)	Percentage of illegal state transitions correctly rejected	100%	100%	Ensures strict enforcement of state-machine rules

Metric	Description	Lease Contract	Mortgage Contract	Interpretation
Authorization Compliance Rate (ACR)	Percentage of sensitive transitions executed only with valid endorsements	100%	100%	Confirms registrar/lender governance is preserved
Unauthorized Access Attempts Blocked	Number of unauthorized calls rejected	All	All	Prevents role misuse and insider abuse
State Consistency Rate	Consistency of parcel–lease–mortgage linkage	100%	100%	Avoids conflicting or orphan states
Audit Trail Completeness (ATC)	Ratio of recorded transitions to expected transitions	100%	100%	Guarantees full traceability
Double-Encumbrance Prevention	Attempts to create overlapping liens	N/A	Fully blocked	Prevents double-pledging of land assets
Premature Closure Prevention	Attempts to close/release before obligations satisfied	Fully blocked	Fully blocked	Enforces legal and financial correctness
Dispute Handling Safety	Safe transition to DISPUTED state	Supported	Supported	Allows external legal resolution

The correctness evaluation demonstrates that the proposed smart contract framework strictly enforces transactional safety and governance constraints. All injected invalid operations—such as activating a lease without registrar approval, registering multiple liens on the same parcel, or releasing a mortgage before full repayment—were consistently rejected. This confirms that the finite-state-machine design effectively prevents illegal or premature transitions.

Authorization compliance remained at 100% for all sensitive actions, indicating that endorsement-based governance correctly preserves institutional authority. Registrar-controlled and lender-controlled transitions could not be bypassed by citizen or unauthorized roles, even under concurrent transaction loads. Additionally, every valid transition generated an immutable ledger entry, resulting in complete audit trails for both lease and mortgage lifecycles.

Overall, these results show that smart contract automation, when combined with permissioned access control, can enhance efficiency without compromising legal correctness or regulatory oversight, making the approach suitable for real-world land governance systems.

E. Overall Discussion and Insights

The experimental results collectively show that smart contract-based automation is feasible for lease and mortgage governance when implemented using a hybrid, permissioned blockchain architecture. While mortgage transactions incur higher endorsement and latency costs than leases, these overheads are justified by the need for stronger institutional control and legal assurance.

Most importantly, the results demonstrate that automation does not conflict with statutory oversight. Instead, it enhances coordination between registrars, banks, and citizens by providing a shared, tamper-evident execution layer. This positions the proposed framework as a practical step toward modernizing land governance workflows rather than a disruptive replacement of existing institutions.

VIII. Conclusion and Future Work

This paper presented a smart contract-based automation framework for lease and mortgage transactions within a blockchain-enabled land governance system. Unlike prior approaches that focus primarily on ownership record immutability, the proposed framework addresses the end-to-end lifecycle management of legally significant transactions, including lease execution, rent compliance, lien registration, repayment tracking, and mortgage release. By modeling these processes as deterministic state machines on a permissioned blockchain, the system ensures correctness, auditability, and consistency across land registries and financial institutions.

A key contribution of this work is the adoption of a hybrid governance model, in which smart contracts automate operational logic while registrars and lenders retain statutory authority over legally binding state transitions. Experimental results demonstrate that this approach achieves predictable performance and scalable throughput under concurrent workloads, while strictly enforcing authorization and workflow constraints. The results further confirm that automation does not compromise governance or legal oversight; instead, it reduces administrative delays, prevents invalid state transitions, and provides a transparent audit trail for all stakeholders.

Future work will focus on extending the framework in several directions. First, formal verification techniques will be explored to mathematically prove safety and liveness properties of the lease and mortgage state machines. Second, the integration of advanced oracle governance mechanisms, such as threshold attestations and multi-source payment confirmation, will be investigated to further strengthen trust in external event validation. Third, privacy can be enhanced through selective disclosure and cryptographic techniques that minimize exposure of sensitive financial terms while preserving verifiability. Finally, large-scale pilot deployments and usability studies with government agencies and financial institutions will be conducted to evaluate real-world adoption challenges, regulatory alignment, and long-term operational sustainability.

Together, these extensions aim to advance blockchain-enabled land governance from experimental prototypes toward deployable, legally compliant digital infrastructure that supports secure and efficient property-related transactions.

References

[1] A. Saari, J. Vimpari, and S. Junnila, "Blockchain in real estate: Recent developments and empirical applications," *Land Use Policy*, vol. 121, Art. no. 106334, 2022, doi: 10.1016/j.landusepol.2022.106334.

[2] R. M. Bennett, T. Miller, M. Pickering, and A.-K. Kara, "Hybrid approaches for smart contracts in land administration: Lessons from three blockchain proofs-of-concept," *Land*, vol. 10, no. 2, Art. no. 220, 2021, doi: 10.3390/land10020220.

[3] R. M. Zein and H. Twinomurinzi, "Information sharing in land registration using Hyperledger Fabric blockchain," *Blockchains*, vol. 2, no. 2, pp. 107–133, 2024, doi: 10.3390/blockchains2020006.

[4] Centre of Excellence in Blockchain Technology (CEBT), National Informatics Centre, Government of India, "Land Records—Blockchain Case Study," 2023. [Online].

[5] E. Androulaki *et al.*, "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Porto, Portugal, 2018, pp. 1–15, doi: 10.1145/3190508.3190538.

[6] T. Guggenberger, J. Sedlmeir, G. Fridgen, and A. Luckow, "An in-depth investigation of the performance characteristics of Hyperledger Fabric," *Computers & Industrial Engineering*, vol. 173, Art. no. 108716, 2022, doi: 10.1016/j.cie.2022.108716.

[7] N. Andola, Raghav, M. Gogoi, S. Venkatesan, and S. Verma, "Vulnerabilities on Hyperledger Fabric," *Pervasive and Mobile Computing*, vol. 59, Art. no. 101050, 2019, doi: 10.1016/j.pmcj.2019.101050.

[8] H. Wang, C. Guo, and S. Cheng, "LoC—A new financial loan management system based on smart contracts," *Future Generation Computer Systems*, vol. 100, pp. 648–655, 2019, doi: 10.1016/j.future.2019.05.040.

[9] World Bank Group, *Smart Contract Technology and Financial Inclusion*, FinTech Note No. 6, Washington, DC, USA, 2020.

[10] G. Avci and Y. O. Erzurumlu, "Blockchain tokenization of real estate investment: A security token offering procedure and legal design proposal," *Journal of Property Research*, vol. 40, no. 2, pp. 188–207, 2023, doi: 10.1080/09599916.2023.2167665.

[11] O. C. Uchani Gutierrez and G. Xu, "Blockchain and smart contracts to secure property transactions in smart cities," *Applied Sciences*, vol. 13, no. 1, Art. no. 66, 2022, doi: 10.3390/app13010066.

[12] P. Dhamor, "Smart contracts in leasing: Is India ready?," *NLIU Law Review*, vol. 9, no. 1, pp. 40–80, 2020.

[13] J. Abang, S. S. Kanhere, M. Zomaya, and R. Buyya, "Latency performance modeling in Hyperledger Fabric blockchain: Challenges and directions," *Internet of Things*, Art. no. 101217, 2024, doi: 10.1016/j.iot.2024.101217.

[14] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.