Design and Formal Modeling of a Secure and Scalable Consortium Blockchain Architecture for Land Record Management in India

Abstract

Land record administration in India remains fragmented across multiple institutions, leading to mutation delays, poor interoperability, and vulnerability to unauthorized changes. While digitization improves access, it fails to ensure integrity, non-repudiation, or eliminate trust dependencies, highlighting the need for a more robust architectural approach. This paper proposes a secure, scalable consortium-hybrid blockchain framework tailored to India's governance structure. It employs a permissioned model where authorized government entities act as validators, combining regulatory control with distributed trust. A formal ownership state model and transition function capture land mutation, transfer, and dispute resolution as verifiable ledger state changes.The contributions include: (1) a hybrid architecture integrating on-chain ownership with off-chain hashed documents, (2) role-based smart contracts for controlled transactions, (3) a formal adversarial and fault-tolerant security model, (4) geographic channel segmentation for scalability, and (5) a performance evaluation framework.The system formally guarantees integrity, immutability, non-repudiation, and role-restricted access under bounded faults, providing a governance-aligned blueprint for next-generation land record management in India.

Keywords: Consortium Blockchain, Land Record Management, Byzantine Fault Tolerance, Smart Contracts, Role Based Access Control, Permissioned Distributed Ledger, Governance Aligned Digital Infrastructure.

1.Introduction

1.1Background and Motivation (India Condition)

In India, land is a critical economic and social asset, supporting livelihoods, credit, inheritance, and long-term investment. However, land record management remains institutionally fragmented and procedurally complex, involving revenue departments, sub-registrar offices, survey authorities, and local administrative units.

A key process, mutation—the formal update of ownership after sale, inheritance, or legal order—requires multi-level verification across institutions. This includes document registration, revenue validation, possible financial clearance, and sometimes judicial intervention. These interdependencies introduce delays, inconsistencies across systems, and reliance on manual or semi-digital workflows, limiting real-time synchronization.

The system operates on institutional trust, where different actors maintain partially independent records. This often leads to discrepancies between registered deeds and ownership entries, enabling duplication, delays, or unauthorized modifications. Although centralized databases improve accessibility, they lack cryptographic guarantees against tampering or insider abuse.

Ensuring tamper-evident ownership is therefore an institutional challenge. A robust system must provide traceability, auditability, and immutability of ownership changes while accommodating legitimate administrative and legal interventions. This tension between distributed trust and regulatory authority motivates the use of permissioned distributed ledger technologies as a complementary infrastructure.

Consortium blockchain architectures, governed by authorized public institutions, offer a pathway to cryptographically verifiable ownership records while preserving administrative control. However, their adoption in India requires careful formal modeling aligned with institutional structures, legal frameworks, and scalability requirements—motivating the approach presented in this research.

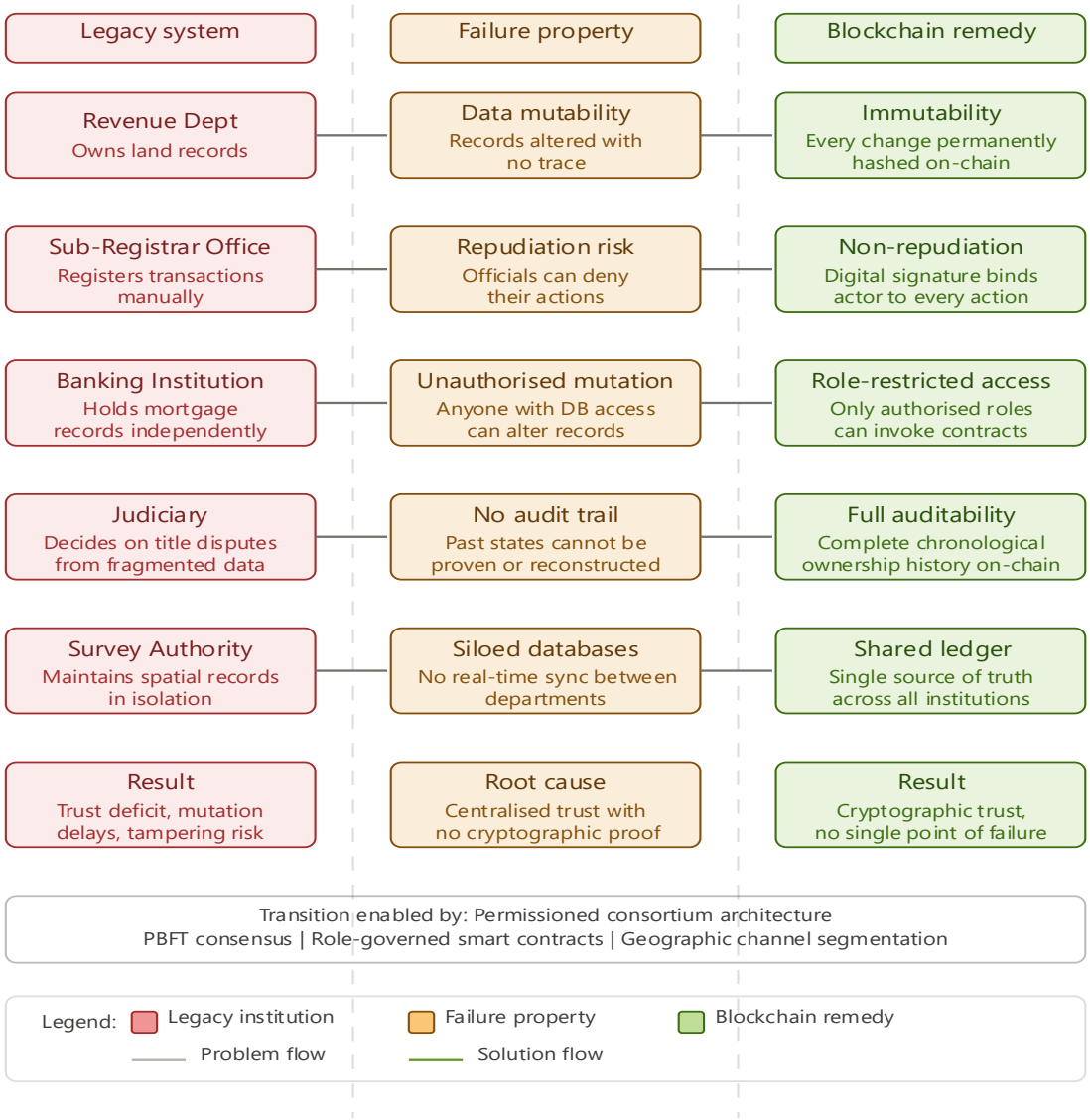


Fig. 1. Conceptual transition from fragmented legacy land administration to a consortium blockchain-based trusted ownership framework. The left column depicts institutional failure modes; the right column shows the corresponding blockchain-enforced security properties.

1.2 Problem Statement

Land ownership administration in India can be abstracted as a dynamical land maintenance issue on a finite space of registered entities and land regions. In order to model the system formally we define the following basic elements:

Let

- $U = u_1, u_2, \dots, u_n$denote the set of registered users, where each user represents a legally identifiable individual or institution recognized by the administrative framework.
- $P = p_1, p_2, \dots, p_m$denote the set of registered land plots, each uniquely identified through survey and registration attributes.
- Odenote the ownership state function such that

$$O: P \times T \to U$$

where T represents discrete time or epochs of transaction. Given a plot p∈P and a time t ∈T , O(p,t) is a function that returns the person whose ownership of the plot is legally acknowledged at time t.

Under the traditional administrative model, ownership is transferred by a series of institutional processes such as the registration of documents, their validation by revenue collection authorities and in certain instances, their validation by financial or judicial authorities. In order to formalize this, the ownership transition function is defined as:

$$O(p, t+1) = f(transaction, verification, endorsement)$$

where

transaction An occurrence of legally initiated transfer of ownership including, but not limited to, sale, inheritance, partition, or court order,

verification refers to procedural validation by authorized administrative officials, and

endorsement shows obligatory authorizations of particular institutional positions based on the type of transaction.

The fundamental issue that is discussed in this work is that the design of a distributed system where transition function f is is implemented in such a way so that it is correct, resistant to unauthorized modifications and that the organization is accountable. Particularly, the system should meet the following properties:

Immutability

After making a valid change of ownership at time t, it is required that the historical state O(p,t) be computationally infeasible to change without being detected. This will guarantee that the history of the transactions is not lost and the transactions cannot be manipulated retroactively.

Non repudiation

Any transaction incurring a change of state should be cryptographically verifiable to the user who initiates it and the user who signs it. A recorded action should not be able to be denied by any participant.

Role restricted mutation

The ownership updates are to be performed in the case when the respective institutional roles have given legitimate endorsement. Invocation and alteration of the transition function must not be encouraged by unauthorized entities.

Auditability

The system should be able to give a verifiable and chronological order of all ownership changes. Given a plot p, one should be able to determine the entire history of state transitions since the beginning of time to the present epoch.

The difficulty, then, is to build a system architecture in which the role f is was being deterministically enforced by cryptographic validation and consensus, instead of being applied by centralized administrative trust. The rest of this paper is a consortium blockchain architecture that implements this formal structure according to the institutional constraints of land administration in India.

1.3 Indian Technical Realities

While distributed ledger technologies offer strong guarantees of integrity and traceability, their application to land record management in India faces significant structural and technical challenges. These arise not only from scale, but also from institutional diversity, legal ambiguity, and state-level administrative autonomy, requiring context-aware system design.

Inter-departmental coordination.
Land governance in India involves multiple entities—revenue departments, sub-registrars, survey authorities, financial institutions, and courts—each responsible for different stages of the property lifecycle. These actors operate on partially independent and often non-interoperable systems, sometimes combining digital and manual processes. This leads to inconsistent ownership updates and synchronization issues. A blockchain solution must therefore enable coordinated, atomic state transitions while preserving institutional role boundaries and avoiding excessive procedural latency.

Scalability across states.
India's decentralized state-level administration introduces heterogeneity in policies, data formats, and workflows. A scalable architecture must handle high transaction volumes, support district- and state-level expansion, and enable geographic partitioning for parallel processing while maintaining overall consistency.

Consensus trust boundaries.
In a consortium blockchain, validators are authorized institutions such as revenue departments or registration authorities. Defining appropriate fault tolerance is critical: while crash fault tolerance may address technical failures, insider threats and collusion necessitate Byzantine fault-tolerant models. This requires clearly defined assumptions on the number of tolerable faulty or malicious validators.

Privacy vs transparency.
Land records must be publicly verifiable while protecting sensitive personal and financial data. This balance can be achieved through selective disclosure, cryptographic hashing of documents, off-chain encrypted storage, and role-based access control. The challenge lies in ensuring confidentiality without compromising auditability.

Legal enforceability.
Blockchain-based records must align with existing legal frameworks governing property registration and evidence. Their validity depends on recognition of digital records, signatures, and institutional authority. The system must therefore complement statutory processes, with smart contracts reflecting legal procedures and allowing controlled administrative or judicial overrides.

Overall, deploying blockchain in India's land record system requires careful modeling of institutional roles, trust assumptions, scalability mechanisms, privacy controls, and legal integration—forming the foundation for the proposed consortium architecture.

Table 1: Structural Challenges and Blockchain Mitigation in Indian Land Administration

Challenge	Institutional Layer	Technical Impact	Security Risk	Blockchain Mitigation Potential
Fragmented Record Maintenance Across Departments	Revenue Department, Sub-Registrar Office, Survey Authorities	Data inconsistency and duplication across isolated databases	Conflicting ownership records and unauthorized modifications	Distributed shared ledger ensuring synchronized and immutable record updates
Manual and Paper-Based Mutation Processes	Local Revenue Offices	Procedural delays and human-induced errors	Document forgery and backdated record alteration	Digitally signed transactions with timestamped immutable entries

Lack of Real-Time Inter-Departmental Synchronization	Registration and Revenue Departments	Asynchronous updates and stale ownership states	Double registration or parallel sale attempts	Consensus-based state transition with atomic validation
Limited Transparency in Record Access	Administrative Access Portals	Restricted public verification capability	Corruption and insider manipulation	Permissioned transparency with cryptographic audit trails
Weak Identity and Authentication Mechanisms	Registrar–Citizen Interface	Inadequate user authentication	Impersonation and fraudulent transfer requests	Public Key Infrastructure (PKI)-based digital signatures and role-based authentication
Inadequate Audit and Traceability Mechanisms	State Oversight Authorities	Difficulty reconstructing historical ownership transitions	Disputed titles and evidentiary ambiguity	Immutable transaction history with chronological traceability
Centralized Database Dependence	State Data Centers	Single point of failure and administrative dependency	Insider tampering and data corruption	Consortium validation distributing trust across authorized nodes

1.4 Research Objectives The general objective of this paper is to develop and critically test a consortium blockchain framework that is specific to the institutional and regulatory context of land record management in India. In order to accomplish this goal, the study will be designed to deal with the following six objectives:

Objective 1: Architecture India specific blockchain.

Design an authorized consortium architecture which is an administrative reflection of the Indian land governance. The design should include authoritative validating parties, institutionalization of roles, hybrid on chain on off chain storage integration, and formally defined ownership state model. The architecture is not supposed to be aimed at substituting the

current administrative workflows but should be designed in correspondence with these workflows.

Objective 2: Work on role based smart contract modules.

Build a modular system of smart contract that captures the lifecycle of ownership of the land such as registration, mutation, transfer, escrow management and dispute resolution. Every contract has to implement role limited access control that would match administrative agents like registrars, revenue officers, judicial authorities and citizens. This is aimed at making sure that ownership transfer is only done with the valid conditions of multi party authorization.

Objective 3: Model adversarial situations formally.

Represent a formalized adversarial model that models realistic adversarial capabilities in the Indian environment, such as malicious insiders, colludents, mutations by malicious insiders, and replay attacks. Formally study security properties of the proposed system, with well-defined fault assumptions. The goal of this is to get beyond conceptual design and offer justifiable assurances in terms of integrity, non repudiation and fault tolerance.

Objective 4: Measure the performance in real load conditions.

Pilot the architecture proposed and determine the performance of the architecture operating with simulated transaction workloads that resemble the activity of land mutation and transfer. The analysis will be done based on throughput, latency, and resources usage. The goal is to find out whether the system has the capability of supporting reasonable deployment without adding unacceptable delays or computational overheads.

Objective 5: Geographic partitioning Analyze scalability.

Introduce and discuss a geographic channel segmentation model whereby ledger activity is divided into administrative geographical areas like districts or states. Test the impact of this partitioning scheme on transaction parallelism, storage expansion, and workload of validators. This is aimed at making the architecture scalable with the increase in the number of land records and the number of participating institutions.

Objective 6: Evaluate the possibility of governance transition.

Analyze how the offered blockchain framework can be combined into the current law and administrative frameworks. This is through the assessment of the phased deployment plans, institutional validator choice and the concurrence with the statutory procedures that regulate land registration and mutation. The aim is to find out whether the architecture is realistic to make the transition between the conceptual design and the policy aligned implementation in the Indian governance ecosystem. Collectively these goals shape the research into architecture design, formal security modeling, performance validation, scalability planning and governance feasibility giving the proposed system a technical rigor as well as institutional foundation.

1.6 Research Questions

To address the objectives mentioned above, there is a set of research questions that direct the study and investigate the architectural, security, and scalability considerations of deploying a consortium blockchain to manage land records in India.

RQ1: Does a consortium blockchain enhance the transparency of mutations without invading privacy?

This question explores the possibility of a permissioned distributed ledger to deliver verifiable and tamper evident mutation records without exposing sensitive and personal data and transactional information to a third party. It compares the efficiency of role based access control, selective disclosure, and hybrid storage models in achieving a compromise between transparency and privacy needs.

RQ2: To what extent can validator fault tolerance be acceptable at the state level deployment?

This problem addresses trust assumptions on which the proposed system is secure. It considers the question of whether crash fault tolerance is not good enough in the Indian administrative environment or whether the Byzantine fault resilience is needed to mitigate the threats like insider collusion or malicious validator behavior. The analysis attempts to define tolerable limits of misfunctioning or vulnerable validator nodes and maintain consistency and integrity of ledgers.

RQ3: Does geographic channel segmentation have the scale to the national level land records?

This question explores the issues of partitioning ledger activity across administrative regions including districts or states, whether this can help to support large scale deployment without losing consistency or coordination. It evaluates the influence of geographic segmentation on the throughput of transactions, growth of storage, workload of validators and cross regional ownership verification.

1.7 Paper Organization

The rest of this paper is structured in the following way. Section 2 performs a literature review of the literature available on blockchain based land record system, permissioned consensus mechanism and governance oriented distributed ledger frameworks and pinpoints key research gaps. Section 3 shows the institutional system model and the suggested hybrid architecture of consortium consisting of on chain and off chain elements. Section 4 formalizes ownership state model, contract modules, role based access control, security controls and also presents the secure mutation algorithm. Section 5 defines the adversarial model, formal security properties, and gives the Byzantine fault analysis and consensus level validation procedures. Section 6 defines the implementation structure, metrics of performance, scaling modeling and costs. Section 7 talks about the alignment of governance and legal compatibility of the Indian administrative setting. Section 8 offers analytical discourse, presents the responses to the research questions, assesses the implications of the research to institutions, and gives the limitations and future research directions. The last section of the paper is Section 9.

2.Gap Analysis and Literature Survey.

A.Literature Survey

The use of blockchain technology in land registration and land governance systems has been widely researched in the recent past. Soner et al. [1] investigated the use of blockchain and smart contracts to improve the reliability and tamper resistance in land records management with a focus on immutability and decentralized authority. In a specific case related to the Indian land titling, Thakur et al. [2] suggested the use of blockchain as a value-adding infrastructure to establish effective ownership records and policy restructuring. Gupta et al. [3] measured the new paradigms of digital land records in India and noted that the structures were not efficient and that reform had to be undertaken to modernize it.

George et al. [4] came up with a conceptualized decentralized land tenure model that was aimed at mitigating of disputes and security of ownership. Kapoor et al. [5] gave an institutional analysis of land record administration in India and found that there were administrative fragmentation and delays in the procedure. The solution by Shrivastava et al. [6] is a blockchain-based land registry that combines effective consensus and smart contracts to improve security. Verma et al. [7] introduced blockchain based land registration prototype, which proved to be feasible at the system design level. Maheshwari et al. [9] suggested a blockchain based registry model that focuses on safe record management in the form of RCUBE, and Keneth et al. [10] provided an example of a blockchain based land record model in Uganda to provide comparative insights on the incorporation of a blockchain in the process.

Comincioli et al. [12], [13] reviewed the perspectives of governance and corruption mitigation and investigated whether blockchain has a potential of enhancing the right of land users in developing nations, especially in India. Panwar et al. [14] suggested land registry architectures that are decentralized to enhance property rights. Zein et al. [16] showed that Hyperledger Fabric was used to share information in land registration but concentrated on interoperability among the stakeholders.

This domain is informed by more general blockchain architectural and scalability analyses beyond land specific implementations. The surveys of blockchain architectures, consensus mechanisms, and distributed computing evolution were given by Tiwari et al. [8] and Pandey et al. [22]. Kandpal et al. [11] and Khan et al. [19] analysed the challenges of scalability such as increase in storage, overhead in communication, and performance bottlenecks. Solanki et al. [21] compared design trade offs of blockchain frameworks. Farooq et al. [18] outlined the blockchain development procedures and architecture layers applicable in the adoption of blockchain by enterprises.

Taken together, these publications show that blockchain has the potential to increase the transparency, resistance to tampering, and distributed trust relating to land administration systems. The majority of the contributions are however conceptual frameworks, prototype implementations or one or two aspects like scalability or security as opposed to a formally

modeled governance aligned architecture based on the specifics of Indian institutional structures.

B. Gap Analysis

Despite progress, key gaps remain. Existing studies on Indian land governance lack formally defined ownership state models with clear transitions and mathematically specified mutation functions. Prior blockchain registry proposals focus on feasibility but omit adversarial modeling and explicit fault tolerance aligned with PBFT. Scalability research discusses general blockchain limits but does not address geographic segmentation or district-level partitioning for national systems.

Governance integration is largely theoretical, with limited specification of institutional roles, endorsement policies, or access control aligned to Indian hierarchies. Similarly, security approaches from other domains are not adapted into coherent threat models addressing insider attacks, validator collusion, or replay risks in land systems. Overall, no work integrates institutional modeling, consensus assumptions, role-based contracts, geographic scalability, and governance transition into a unified India-specific framework.

This paper addresses these gaps through a formally modeled consortium blockchain architecture incorporating ownership state transitions, role-based access control, Byzantine fault tolerance, geographic scalability, and governance-aligned endorsement policies.

2.1 Blockchain in Land Record Systems

Blockchain-based land systems are typically proposed as tamper-evident registries enhancing transparency and reducing intermediaries. Prior works explore smart contracts, decentralized tenure models, and prototype implementations, including hybrid on-chain/off-chain storage and permissioned frameworks like Hyperledger Fabric.

Three main approaches emerge: fully on-chain storage, hybrid storage, and consortium-based governance. However, most implementations remain conceptual or prototype-level, lacking formal state modeling and adversarial analysis.

2.2 Public vs Consortium Blockchain in Governance

The choice between public and consortium blockchains significantly impacts governance applications. Public systems like Ethereum offer transparency and strong decentralization but suffer from high latency, cost variability, privacy risks, and lack of regulatory control. Their open participation model limits suitability for state-controlled land systems.

Consortium platforms such as Hyperledger Fabric restrict participation to authorized institutions, enabling role-based access, controlled data visibility, and transaction finality. These features align better with governance needs, though they depend on institutional trust and require careful fault tolerance design to mitigate collusion risks.

Overall, while public blockchains maximize openness, consortium models better balance decentralization with administrative control, making them more suitable for Indian land governance—provided consensus and governance mechanisms are rigorously defined.

2.3 Property Transfer Smart Contracts

Smart contracts enable programmable enforcement of ownership transfer, mutation, escrow, and endorsement processes, ensuring deterministic state transitions. However, several challenges remain. Legal procedures are complex and difficult to fully encode, requiring careful abstraction. Contracts depend on external data (oracle problem), which may introduce inaccuracies. Immutability can limit adaptability to legal changes, necessitating upgradeable designs.

Security is also critical, as vulnerabilities in contract logic can lead to unauthorized actions or asset locking, requiring formal verification and strict access control. Finally, contracts must align with legal frameworks to ensure enforceability.

Thus, while powerful, smart contracts must address legal complexity, data reliability, adaptability, and security to be viable in land record systems.

2.4 Security and Consensus of Permissioned Systems.

With the assumption that the validator nodes are known and authenticated, consensus mechanisms are run in a permissioned blockchain environment. The identities of the validators are institutionally managed unlike open public networks. The type of consensus protocol used is thus determined basing on the assumed threat and faults.

Experimental Byzantine Fault Tolerance.

Byzantine Fault Tolerance Practical BFT is a consensus mechanism that ensures consistency of a system in the presence of arbitrary or malicious behaviour of some of the participating nodes. Classical formulations In classical constructions, a system which supports a set of nvalidators can withstand up to fByzantine faulty nodes assuming that

$$n \geq 3f + 1$$

In this state of affairs, consensus and security is maintained provided that the malicious or colluding nodes do not surpass f. Mechanisms using PBFT apply in cases where insider corruption, coordinated collusion among the validators, or arbitrary manipulation of messages are contained in the threat model.

However, this is not the case because PBFT will introduce communication overhead that increases with the size of the validators, and this can impact scalability. Hence, it is not applicable to all validator sets and assumptions of institutional trust, depending on the anticipated size.

RAFT Consensus

RAFT A crash fault tolerant consensus protocol used in distributed system where failure of nodes can be detected by ceasing or being inaccessible, though this failure is not considered to be malicious. RAFT is based on leader election and log replication as a consistency assurance between replicas.

A crash fault tolerant model will ensure that the system is correct provided that most of the nodes are online and honest. RAFT is not resistant to Byzantine behavior like forged messages or synchronized malicious updates of the state. It can be used only in such situations, when validators are supposed to adhere to protocol rules and the presence of insider adversaries is not included in the threat model.

Crash Faults and Byzantine Faults.

The security limit of a permissioned blockchain system is the difference between crash and Byzantine faults.

The assumption of crash faults is that a node can fail in silence, restart arbitrarily, or lose connectivity, but does not transmit conflicting and malicious data. Byzantine faults are based on the assumption that a node can act in any way such as sending non-consistent messages to other peers, altering data, or acting in cooperation with other malicious nodes.

Under the governance of land records, there is a choice between crash fault tolerant and Byzantine fault tolerant consensus that should indicate realistic adversarial assumptions. In case institutional validators are complete and observed, crash fault tolerance might be adequate. In case the threat model involves possible insider maliciousness or a coordinated maliciousness, Byzantine resilience will be required. The architectural choice must therefore be in line with formally established adversarial model and not performance only.

2.5 Identified Gaps

An analysis of the current blockchain based land record ideas shows that there are a number of limitations that are repeated.

In the first place, a lot of designs use a blockchain structure, without explicitly stating the fault model. The crash and Byzantine resilience difference is not widely defined and results into ambiguity in security guarantees.

Second, consensus selection is often described as being motivated by intuitive justifications as opposed to being structured in terms of explicit adversarial suppositions and validator trust thresholds.

Third, scalability decisions are made at a conceptual, rather than explicitly modelling the increase of validators, volume of transactions, or geographic divisions.

Fourth, property transfer smart contract logic is frequently defined in functional terms, but it is not defined by state models or role restricted execution constraints.

Fifth, the qualitative discussion on legal alignment does not analyze many frameworks on how blockchain documented state transitions can be relatable to statutorily pronounced procedures in Indian governance.

These loopholes encourage the current paper to present an architecturally formulated model, a formal adversarial model, a well-stated boundary of consensus and a scale attentive design that suits the land record ecosystem in India.

Table 2: Comparative Gap Analysis of Blockchain Land Registry Approaches

Study	Country Context	Blockchain Type	Security Modeling	Scalability Modeling	Governance Integration
Soner et al. (2021)	General	Permissioned Blockchain	Conceptual integrity focus	Limited discussion	No explicit institutional mapping
Thakur et al. (2020)	India	Permissioned Blockchain	Limited threat analysis	Not formally modeled	Policy-level discussion
Verma et al. (2024)	India	Prototype Permissioned System	Implementation-oriented	Limited scalability consideration	Minimal administrative role modeling
Keneth (2024)	Uganda	Consortium Framework	Basic security considerations	Not formally analyzed	Partial institutional alignment
Comincioli (2021)	India	Conceptual Blockchain Model	Governance-focused	Not technically addressed	Strong governance discussion
Proposed Work	India	Consortium Hybrid Permissioned Blockchain	Formal adversarial and fault modeling	Geographic channel-based scalability model	Explicit institutional role mapping and endorsement policy design

3. System Model and Architecture

3.1 Institutional Model in the Indian Context

The proposed architecture is based on the institutional set up of land administration in India. The following identified actors are the main actors as documented in the system model that are aligned to specific operational and validation roles in the consortium framework.

Land Revenue Department

The Land Revenue Department keeps accountancy records of the land ownership and mutation registers. In the proposed system, it acts as a primary validating authority that carries out the endorsement of ownership state transitions once procedural verification has been carried out.

It can either run the validator nodes in the consortium network and authoritative ownership state consistency at the state or district level.

Sub Registrar Office

The Sub Registrar Officer registers the property transactions such as the sale deeds, gift deeds and transfer documents. In the blockchain system, it is an initiator and verifier of transactions. It sends a mutation request with a digital signature to the registry of documents on registration, which causes the ownership transition process to occur.

Banking Institution

Banking and other financial institutions depend on land records to create mortgage, validate collaterals and enforcement of liens. Banks can also be involved in the system as endorsing parties to transactions dealing with encumbrances. They can also enquire about ownership state and encumbrance history using channels of permissioned access.

Judiciary

The Judiciary is involved in case of dispute, injunction or court ordered transfer. Within the suggested design, courts are designed as elite sponsors that can legitimize extraordinary state changes, including dispute adjudication or property reinstitution. Their position makes the blockchain system be in line with statutory adjudication procedures.

State Validator Authority

The State Validator Authority is the institutional organization that will administer and oversee validator nodes in the consortium network. This authority can have representatives of the revenue and registration departments and is the guarantee that only the representatives of the authoritative government bodies can be included in the consensus.

Citizen

The citizens are legal property owners or transferees. They start ownership transfer requests, digital signatures on approval of transactions, and probably have access to verifiable records on ownership. The citizens are not engaged in consensus but just interface with the system using authenticated client interfaces.

This model of an institution clarifies the role separation enabling the blockchain architecture to encode administrative authority in the policies of consensus and endorsement and maintain procedural accountability.

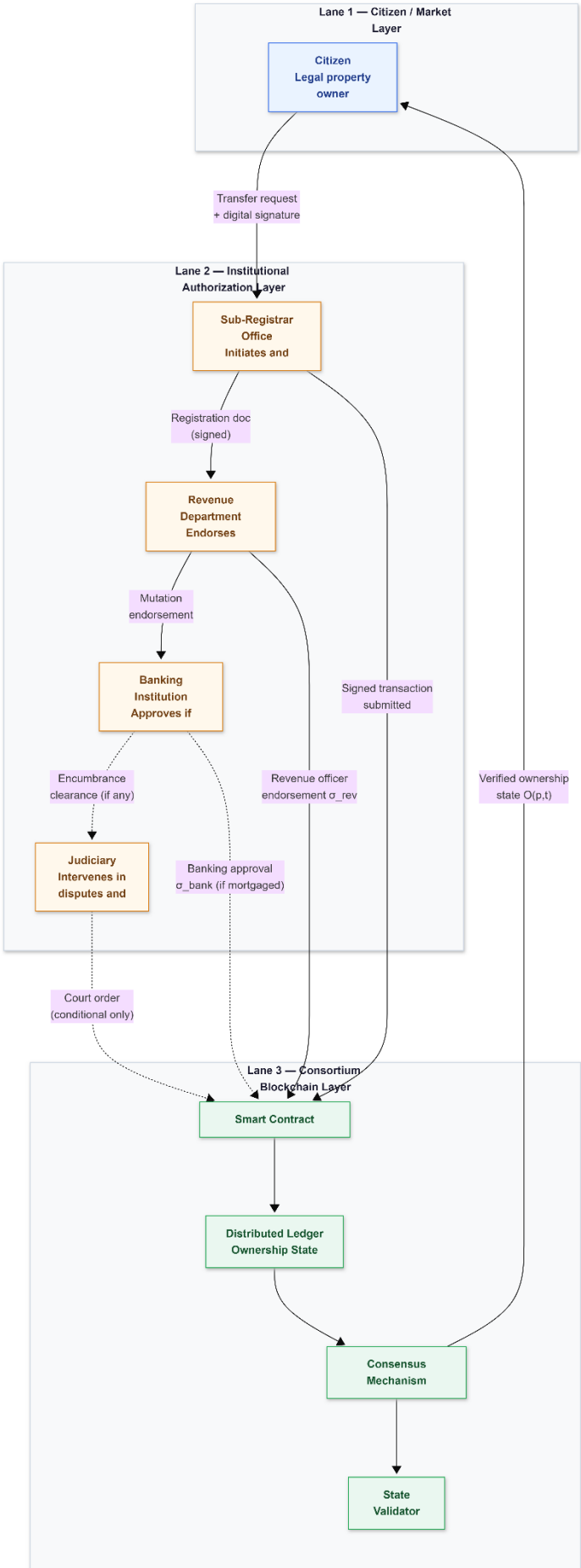

Lane 1 — Citizen / Market Layer
Citizen
Legal property owner
Transfer request + digital signature
Lane 2 — Institutional Authorization Layer
Sub-Registrar Office Initiates and
Registration doc (signed)
Revenue Department Endorses
Mutation endorsement
Banking Institution Approves if
Signed transaction submitted
Encumbrance clearance (if any)
Revenue officer endorsement σ_rev
Verified ownership state O(p,t)
Judiciary Intervenes in disputes and
Banking approval σ_bank (if mortgaged)
Court order (conditional only)
Lane 3 — Consortium Blockchain Layer
Smart Contract
Distributed Ledger Ownership State
Consensus Mechanism
State Validator

Fig. 2. Institutional role interaction model for blockchain-enabled land administration in India, showing three-layer swim lanes with labeled message flows and endorsement sequence numbers**.**

3.2 Proposed Consortium Hybrid Architecture

The offered system uses permissioned consortium blockchain where validator nodes are managed by authorised government entities. It is designed in a hybrid format, where ownership state management is not managed with bulk document storage but is cryptographically linked to it.

The architecture is made of three logical layers, which are network layer, contract execution layer, and storage layer.

In the network layer, consensus is achieved by the use of validator nodes run by selected state authorities. The control of membership is by means of institutional certification and only authorized nodes can validate and commit transactions. Customer like sub registrar offices and revenue officials post digitally signed proposal of transactions on the network.

On the level of contract execution, smart contracts store the regulations that regulate the process of registering land, its mutation, transfer, managing escrow, and resolving conflicts. Such contracts measure transaction requests deterministically on role based endorsement policies. Once institutional approvals have been verified and consensus has been reached then a transaction is committed to the ledger.

On Chain Components

The on chain layer maintains important information on state which is necessary and may be required to be either integrity assured or audit guaranteed.

Smart contracts

Smart contracts establish the role of changing ownership and perform role restricted mutation. They keep a system of records that identify the plot and the existing ownership, encumbrance status, and reference to the transaction history.

Ownership state

The ownership state is a deterministic map between plot identifiers and legally owned owners. Every legitimate mutation has the effect of creating a new state listing whilst keeping the historic chain of ownership intact. The ledger makes the past states unalterable and verifiable.

The chain only stores minimal and necessary data needed to verify and audit to maintain efficiency.

Off Chain Components

To prevent unwanted ledger expansion and preserving confidential data, off-chain large documents and supporting records are kept.

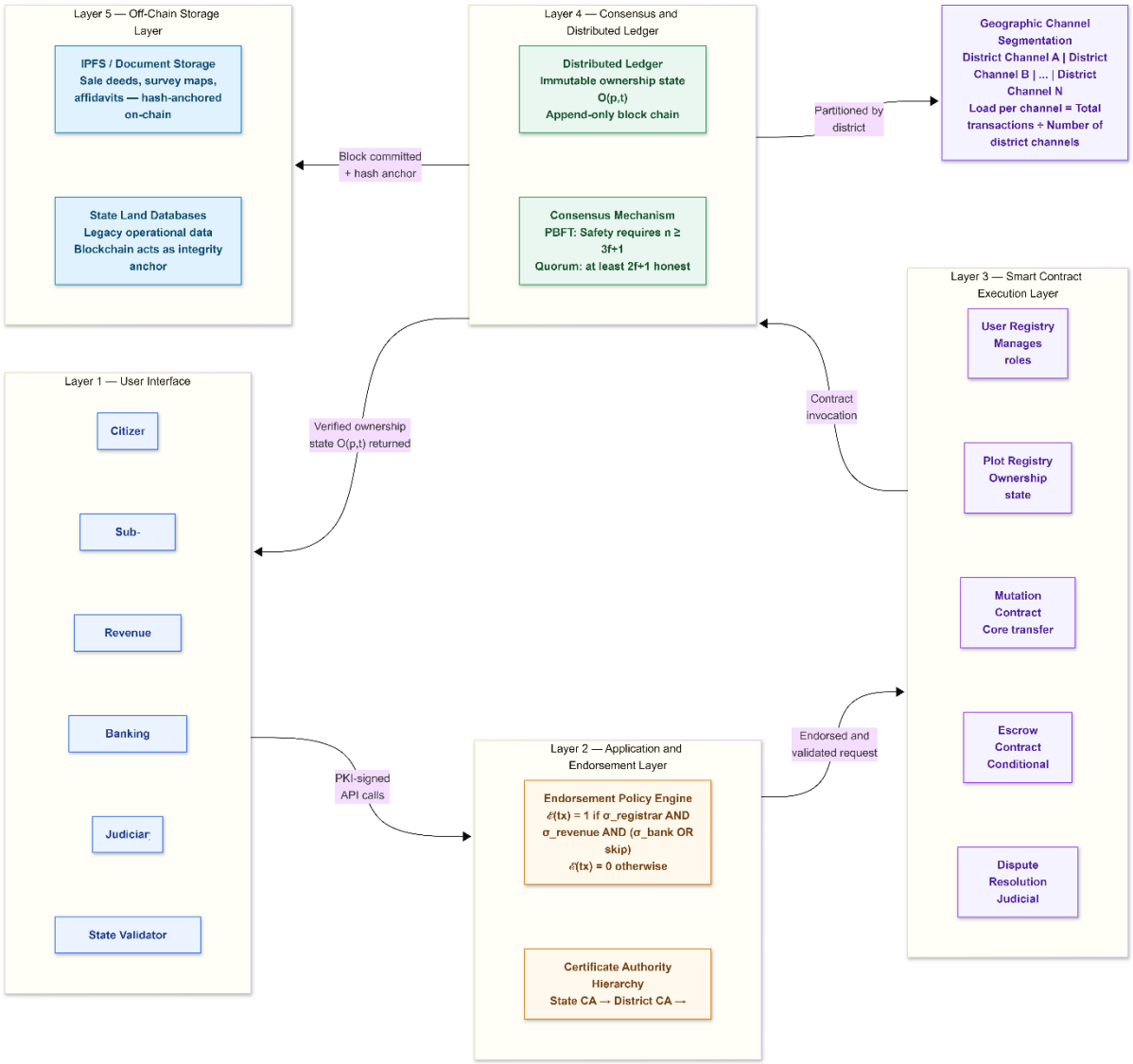


Fig. 3. Hybrid consortium blockchain architecture for Indian land record management, comprising five layers: user interface, endorsement policy, smart contract execution, consensus and distributed ledger, and off-chain storage with geographic channel segmentation

Distributed document storage or IPFS.

The digitized sale deed, survey map, supporting affidavits and the other related documentation can be stored in a distributed file system like IPFS or any similar secure-remedy storage system. Cryptographic hash of every document is stored in chain. Any change made to the off chain document would invalidate its hash reference, as a means of tamper evidence.

State databases

Structured administrative data may still be stored in operational databases held by revenue or registration departments. These databases update state of ownership with the blockchain ledger

by controlled interfaces. The blockchain will serve as the integrity anchor, and the databases will be efficient in query and compatible with the legacy system.

The hybrid design is such that the blockchain registry is authoritative in the ownership state and transaction integrity, whereas off chain storage is effective in ensuring scalability, privacy, and compatibility with current administrative systems.

$$Load = \begin{cases} \frac{N_{tx}}{d}, \text{if transactions are evenly partitioned.} \end{cases}$$

Geographic segmentation distributes workload across channels to reduce consensus overhead.

Figure 3: Geographic Channel-Based Scalability Model for Consortium Blockchain in Indian Land Record Management

3.4 Consensus Mechanism Design

Let

- n denote the total number of validator nodes in the consortium network.
- f denote the maximum number of faulty validators tolerated without compromising ledger consistency.

PBFT Style Consensus

Under a Byzantine fault tolerant model, the system preserves safety and liveness provided that

$$n \geq 3f + 1$$

This implies that the number of malicious or arbitrarily faulty validators must satisfy

$$f < \frac{n}{3}$$

This model is appropriate when the threat model includes insider corruption or colluding validators.

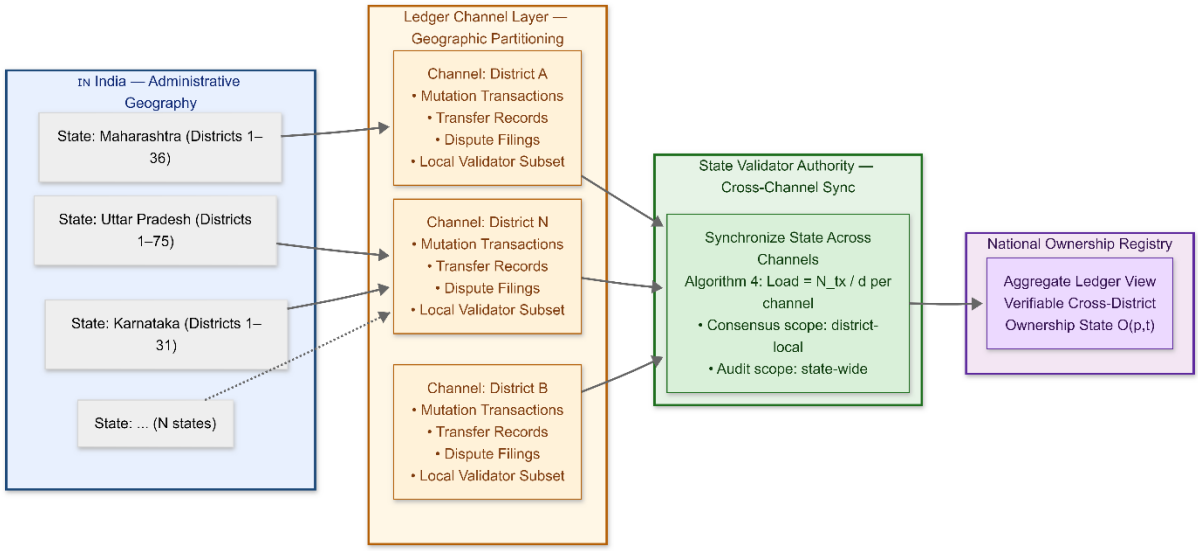


Fig. 4. District-level geographic channel segmentation model for scalable ledger partitioning, where load per channel equals the total transaction volume divided by the number of district channels (Load = N_tx / d).

RAFT Consensus

With crash fault tolerant model like that of RAFT, the system is resilient to the failure of nodes leading to inactivity or disconnecting, as long as a majority of the validators are honest and operational. RAFT lacks defense against Byzantine behavior. It is appropriate when the validators are supposed to act in accordance with protocol rules and only undergo non malicious failures. The decisions between these models should be based on the set adversarial assumptions and not on performance. The policy of endorsement and governance is as follows:

3.5 Endorsement and Governance Policy

Every ownership mutation transaction has to meet a specified endorsement policy before it can be submitted to final consensus.

To have a valid transaction,the transaction must consist of:

Registration signature of registrar of transfer document.

The revenue officer authorization to amend the official ownership record.

Banking approval at the option when the property is mortgaged, under lien or encumbered.

Smart contract logic is used in the implementation of the endorsement policy. A non-signatory transaction is rejected and it does not alter the ownership state. This

provides institutional accountability and role restricted mutation to the consortium structure.

$$\mathcal{E}(tx) = \begin{cases} \mathbf{1}, \text{if } \boldsymbol{\sigma}_r \wedge \boldsymbol{\sigma}_{rev} \wedge (\boldsymbol{\sigma}_b \vee \mathbf{1}); \\ \mathbf{0}, \text{otherwise.} \end{cases}$$

A transaction is considered valid only when mandatory institutional endorsements are satisfied according to the governance policy.

σ_r: registrar signature

σ_{rev}: revenue officer endorsement

σ_b: banking endorsement

Table 3: Institutional Endorsement Policy Matrix for Ownership State Transitions in the Proposed Consortium Framework

Transaction Type	Required Endorsers	Legal Authority Level
Standard Sale Transfer	Sub-Registrar + Revenue Officer	Administrative (Statutory Registration & Mutation Authority)
Inheritance Mutation	Revenue Officer + Supporting Legal Documentation	Revenue Administrative Authority
Gift Deed Transfer	Sub-Registrar + Revenue Officer	Statutory Registration Authority
Mortgage / Encumbrance Creation	Sub-Registrar + Banking Institution + Revenue Officer	Financial & Administrative Authority
Escrow-Based Conditional Transfer	Sub-Registrar + Banking Institution + Revenue Officer	Contractual & Administrative Authority
Court-Ordered Ownership Transfer	Judicial Authority + Revenue Officer	Judicial Authority
Dispute Registration (Title Under Challenge)	Judicial Authority	Judicial Authority
Encumbrance Release	Banking Institution + Revenue Officer	Financial & Administrative Authority

4.Smart Contract Design

4.1 Ownership State Model

Each plot p∈P is represented as a finite state machine to guarantee the deterministic and verifiable transitions of P.

Let the state set be:

$$O(p,t+1) = \begin{cases} \text{Transferred, if } V(p,t) \wedge E(p,t) \wedge P(p,t); \\ \text{Locked, if } T_{escrow}(p,t) = \text{active}; \\ \text{Disputed, if } D(p,t) = 1; \\ O(p,t), \text{otherwise.} \end{cases}$$

The ownership state transition is modeled as a deterministic state function where the next state depends on verification, endorsement, payment completion, escrow conditions, and dispute triggers.

Where:

$V(p,t)$: verification condition

$E(p,t)$: endorsement condition

$P(p,t)$: payment completion

$T_{escrow}(p,t)$: escrow activity state

$D(p,t)$: dispute indicator

The idea is that at any point in time t there is only one state of each plot.

State Definitions

Available

The plot is not burdened with any transactions to be completed and can be transferred.

Under Verification

A transfer or mutation request is logged and in registrar and revenue validation.

Locked

The mortgage, injunction, escrow condition or administrative hold limits the plot temporarily. Any transfer cannot take place without clearing the lock.

Transferred

The change of the ownership has been effectively approved and enrolled on the ledger. The ownership state function reflects the new owner.

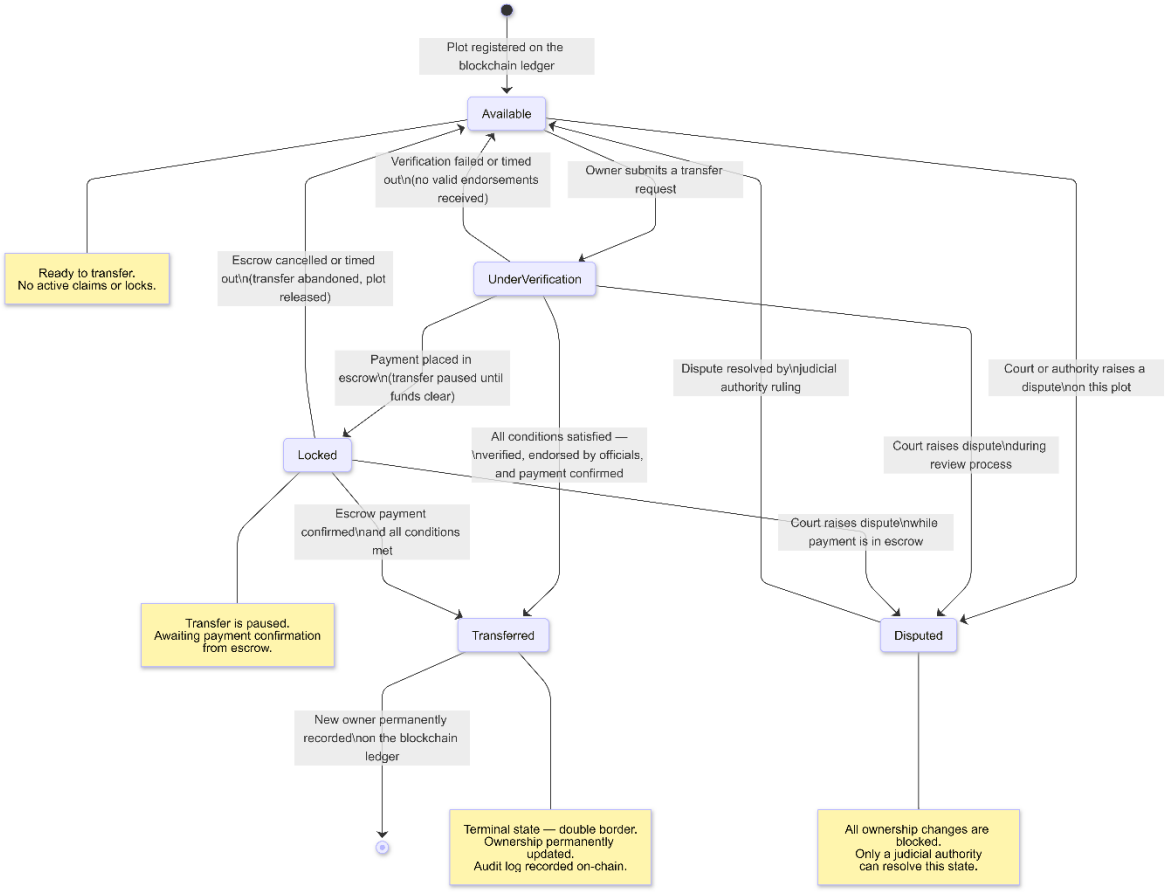


Fig. 5. Finite state machine of land plot ownership lifecycle, with formal transition conditions derived from the ownership transition function O(p, t+1) = f(V(p,t), E(p,t), P(p,t), T_escrow(p,t), D(p,t)).

Disputed

The storyline is open to judicial/administrative controversy. The action of mutation is suspended.

Transition Constraints

Rules and endorsements that are implemented by smart contracts allow state transitions only. Illegal transitions are not accepted. Each valid transition transfers the ownership state function but one which leaves the historical record (which is uneditable) intact.

4.2 Contract Modules

The smart contract layer is designed in five main modules to create separation of duties and institutional responsibility. The User Registry Contract upholds the verified group of participants U, allocates roles including citizen, registrar, revenue officer, banking entity and judicial authority and administers role certification and withdrawal. The Plot Registry Contract tracks the established Pof registered plots of land, provides the uniqueness of plot identifiers,

and information on the current ownership O(p, t). The ownership transition of the Mutation Contract performs the ownership transition role.

$$O(p, t+1) = f(transaction, verification, endorsement)$$

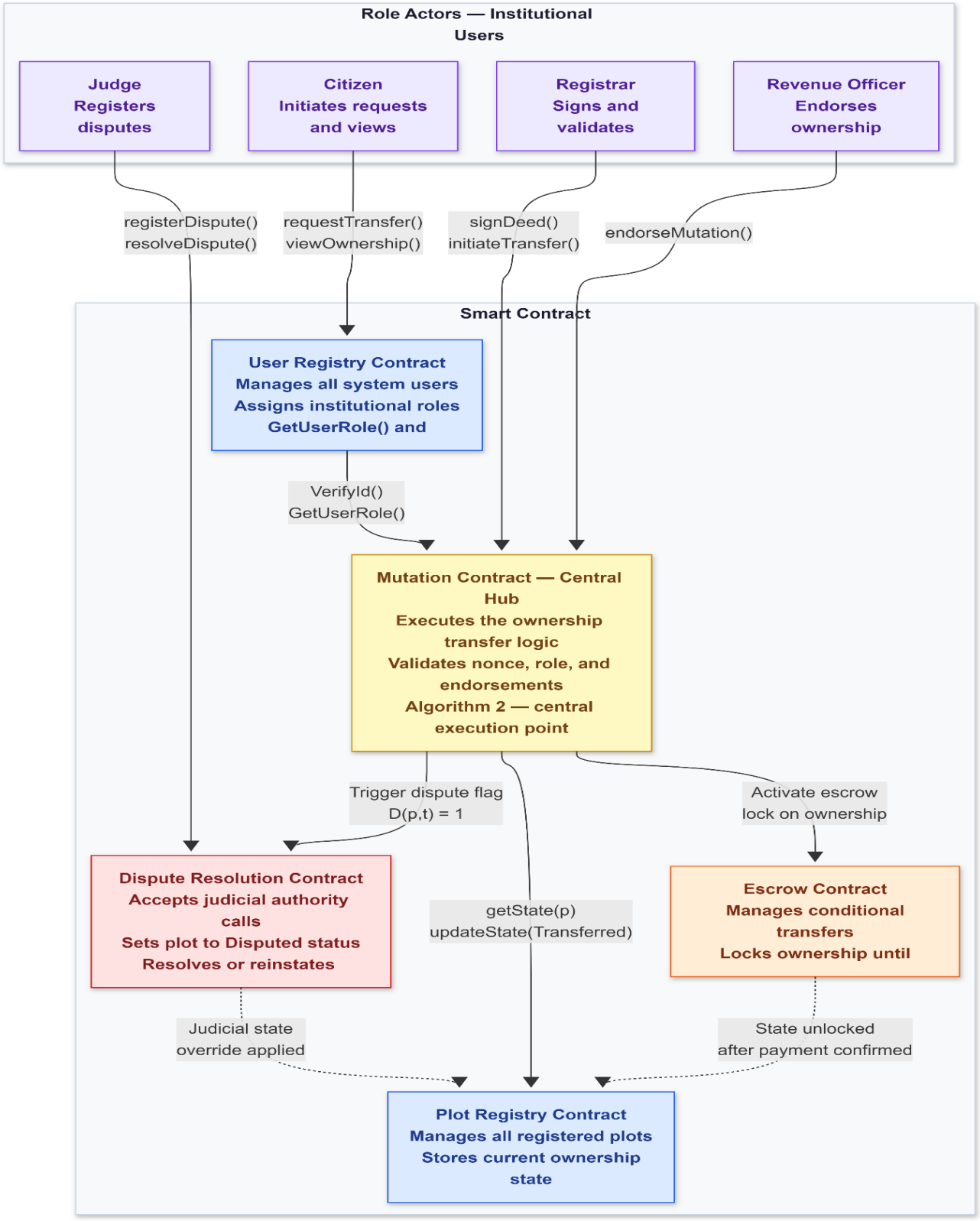


Fig. 6. Modular smart contract interaction diagram with role-based invocation paths, showing the Mutation Contract as the central hub coordinating the User Registry, Plot Registry, Escrow, and Dispute Resolution modules.

Authenticates obligatory signatures prior to engaging in any state modification. The Escrow Contract deals with conditional transfers associated with financial settlement by provisionally locking the state of ownership until escrow conditions are fulfilled. The Dispute Resolution Contract permits any authoritative judicial authority to transfer a plot into the Disputed state, and suspends any additional mutation in the course of litigation, and reconveyed or perfects ownership on any formal resolution.

4.3 Role Based Access Control

Role based access control model to access contract functions is defined on the set of registered users U. Every participant is given one authorization role within an institution. The Registrar has the freedom to attest and sign transfer of property documents and conduct mutation approval procedures. Revenue officer has the power to give endorsement to ownership changes and complete mutation entries in the land record state. The Judge has the sole power to transfer a plot to the Disputed state as well as to restore or reassign ownership after a determination. The Citizen can request transfers, access ownership records and engage in transactions, but is not allowed to alter the ownership state directly.

$$A(r, a) = \begin{cases} 1, \text{if } a \in Perm(r); \\ 0, \text{otherwise.} \end{cases}$$

Access control is enforced through a role-action mapping function that permits only authorized operations based on institutional roles.

r: role

a: action

$Perm(r)$: permitted actions for role

An access control matrix is formally defined as an access control list. to a set of actions authorized permissibly a, and any action not explicitly authorized in Ais rejected on the contract level.

Table 4: Role-Permission Matrix

Role	Initiate Transfer	Endorse Mutation	Approve Encumbrance	Register Dispute	Finalize Ownership Update	View Ownership Records
Citizen	✓	✗	✗	✗	✗	✓

Sub-Registrar Office	✓	✓ (Registration Validation)	✗	✗	✗	✓
Revenue Officer	✗	✓	✓ (Record Update)	✗	✓	✓
Banking Institution	✗	✗	✓ (Mortgage / Lien)	✗	✗	✓ (Limited)
Judicial Authority	✗	✓ (Court Order)	✗	✓	✓ (Post-Adjudication)	✓
State Validator Authority	✗	✗	✗	✗	✓ (Consensus Validation)	✓ (System-Level)

Algorithm 1: Role-Based Access Control Enforcement

Require:
User u, action a, role mapping R, access matrix A

Ensure:
Authorization decision

1: role ← GetUserRole(u)
2:
3: if role = NULL then
4: return Deny
5: end if
6:
7: permissions ← A(role)
8:
9: if $a \in permissions$then
10: return Allow
11: else
12: return Deny
13: end if

4.4 Security Controls

The security measures are implemented on the transaction and contract levels. Every transaction has a nonce that is tied to the identity of the user to avoid a replay attack and submissions. Transfers and the Eseti transaction have to be mediated by the Escrow, meaning that they cannot be changed or approved unilaterally by the financial settlement and ownership

mutation. Mutation and escrow procedures also have conditions of timeouts provided to ensure that a transaction which is not finished or endorsed will be automatically terminated and returned to its former valid state. Each state transition is a non-recursive audit event with hash of a transaction, endorsing roles and a timestamp, which can undergo traceable verification without modifying historical records.

4.5 Secure Ownership Mutation Algorithm

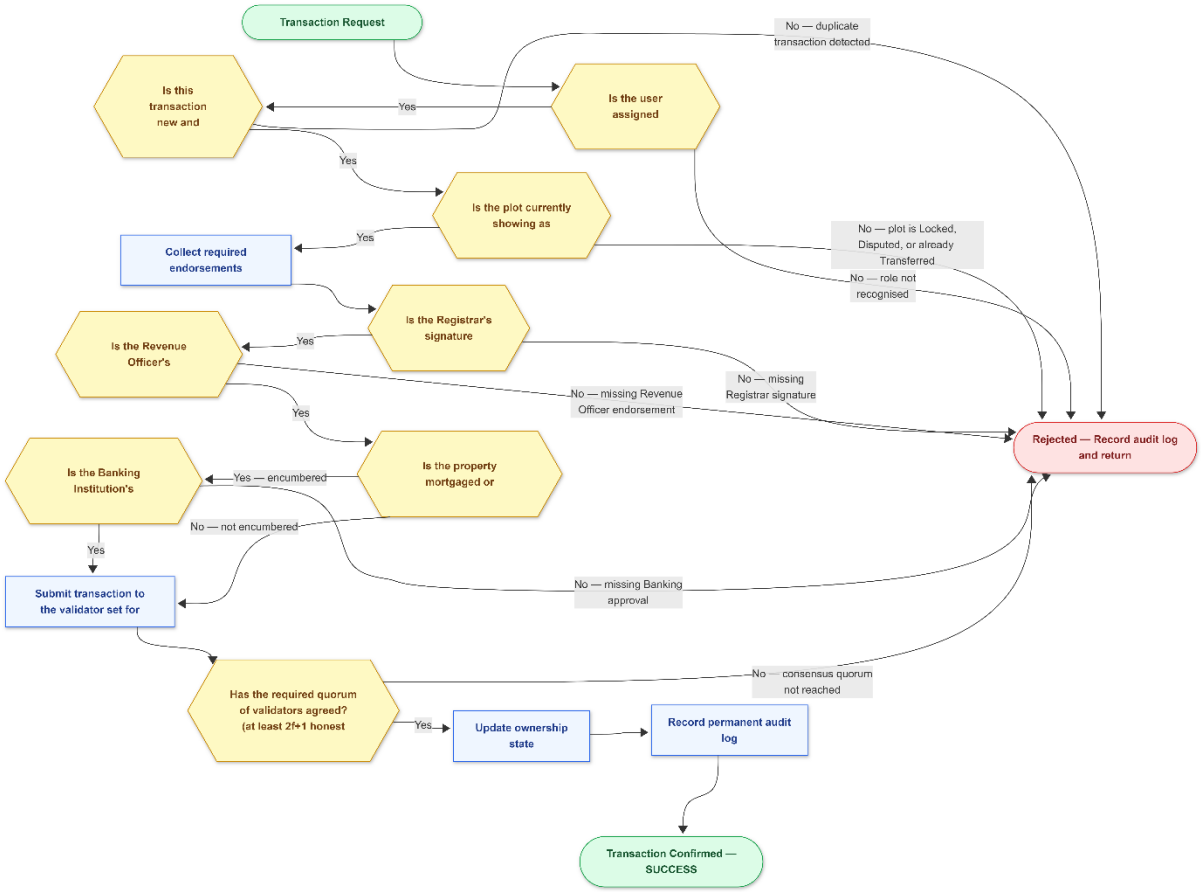


Fig. 7. Endorsement policy validation flow for ownership state transitions, implementing the decision logic of Algorithm 2 (Secure Ownership Mutation) with all reject paths converging to a single audit terminus.

Algorithm 2: Secure Ownership Mutation and Validation

Require:
Input: plot p, transaction request T, nonce n, endorsements E, validator set V
Smart contracts: MutationContract M, RegistryContract R

Ensure:
Updated ownership state $O(p, t+1)$ if valid, else rejection

VerifyIdentity($T.user$)
if role($T.user$) $\notin$ AuthorizedRoles then
    return Reject
end if

if CheckNonce($T.user, n$) = false then
    return Reject
end if

currentState ← R.getState(p)
if currentState ≠ Available then
    return Reject
end if

E ← CollectEndorsements(T)
if ValidateEndorsements(E) = false then
    return Reject
end if

submitStatus ← SubmitToConsensus(T, V)
if submitStatus = Approved then
    R.updateState($p, Transferred$)
    EmitAuditLog(T)
    return Success
else
    return Reject
end if

5 Threat Model and Security Analysis

5.1 Adversarial Model

The system presupposes outside and the inside adversaries in a permitted consortium experience. A rogue registrar can seek to sanction fraudulent transfer certificate or fail to sanction legitimate endorsing. Fault tolerance threshold Colluding validators might want to interfere with consensus or censor transactions, which is also limited by the fault tolerance threshold. Insider attacks can be because of the privilege of position used by an authorized official in the access control matrix wrongly. In key compromise attacks, the private signing key of a party is known, and the party will be able to send transactions till the time a revocation is performed. In a replay attacker, the attacker tries to re-present their valid transactions to cause

unwanted state progression, but it is countermeasured by enforcing nonce and state validation logic.

Table 5: Threat Classification and Attack Surfaces

Threat Category	**Adversarial Actor**	**Attack Surface**	**Potential Impact**	**Mitigation Mechanism**
Malicious Registrar	Insider (Authorized Official)	Mutation Approval Workflow	Unauthorized ownership transfer	Multi-party endorsement policy and smart contract validation
Colluding Validators	Consortium Node Operators	Consensus Layer	Ledger manipulation or transaction censorship	PBFT-based fault tolerance with bounded validator threshold
Replay Attack	External or Internal Actor	Transaction Submission Interface	Duplicate or unintended state transitions	Nonce-based transaction uniqueness enforcement
Key Compromise	Any Authorized User	Digital Signature Mechanism	Fraudulent transaction authorization	Certificate revocation and identity management controls
Unauthorized Access Attempt	External Attacker	API / Client Interface	Illicit state modification attempt	Role-Based Access Control (RBAC) enforcement
Smart Contract Vulnerability	Technical Exploiter	Contract Execution Layer	Logical errors enabling state manipulation	Formal verification and contract auditing
Data Tampering (Off-Chain)	Insider or External Actor	Document Storage Layer	Altered supporting documents	Cryptographic hash anchoring on-chain
Denial of Service (DoS)	External Attacker	Network Layer	Transaction delay or service disruption	Rate limiting and distributed node architecture

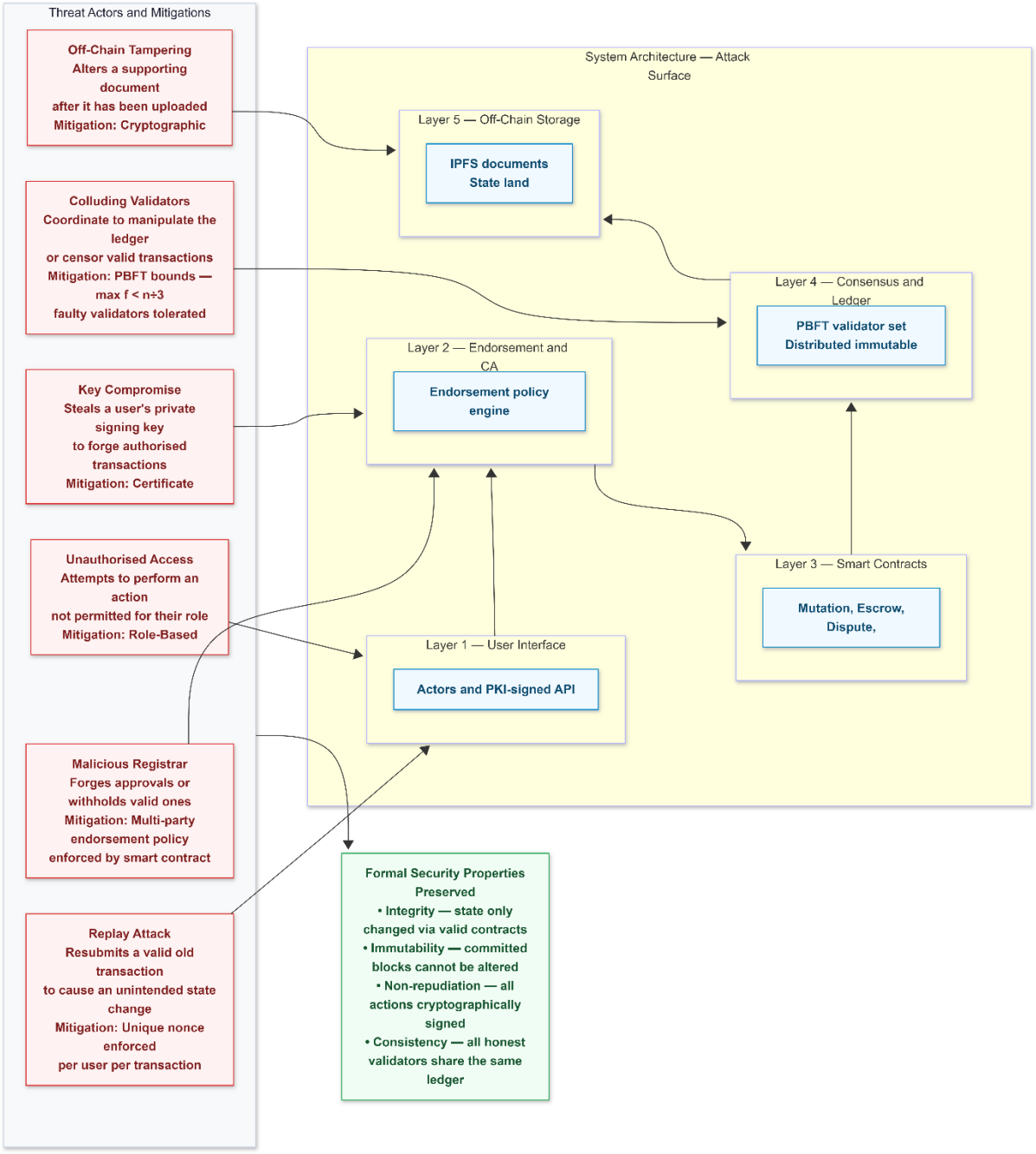


Fig. 8. Adversarial threat surface map overlaid on the system architecture, showing the layer targeted by each of six threat categories and the corresponding mitigation mechanism enforced at that layer

5.2 Formal Security Properties

There are four fundamental security properties of the system. Integrity guarantees that ownership state O(p,t) will not be altered except via authorized smart contract transitions using valid endorsements. Immutability ensures that a transaction can never be modified once it has been made with the help of consensus, without breaking the assumptions of the underlying consensus. Non repudiation is carried out by cryptographic signatures identifying every action of a confirmed identity, which makes it impossible to deny participation. Consistency also

guarantees that every non faulty validator node has the same state of ledger after the conclusion of consensus.

5.3 Byzantine Fault Analysis

With 2n validators in the form of a PBFT style consensus model, the system is safe and live as long as.

$$n \geq 3f + 1$$

which implies

$$f < \frac{n}{3}$$

and f represents the count of bad actors or randomly malicious validators. Safety is that conflicting blocks are unable to get the necessary supermajority of at least 2 f +1honest votes and liveness is maintained until a quorum of honest validators completes the task. Consensus guarantees cannot be mathematically guaranteed in case this threshold is surpassed.

$$\text{System is secure} = \begin{cases} \text{True, if } n \geq 3f + 1; \\ \text{False, otherwise.} \end{cases}$$

Consensus safety is guaranteed only when the number of validators satisfies the Byzantine fault tolerance bound.

Where :

n: total validators

f: faulty validators

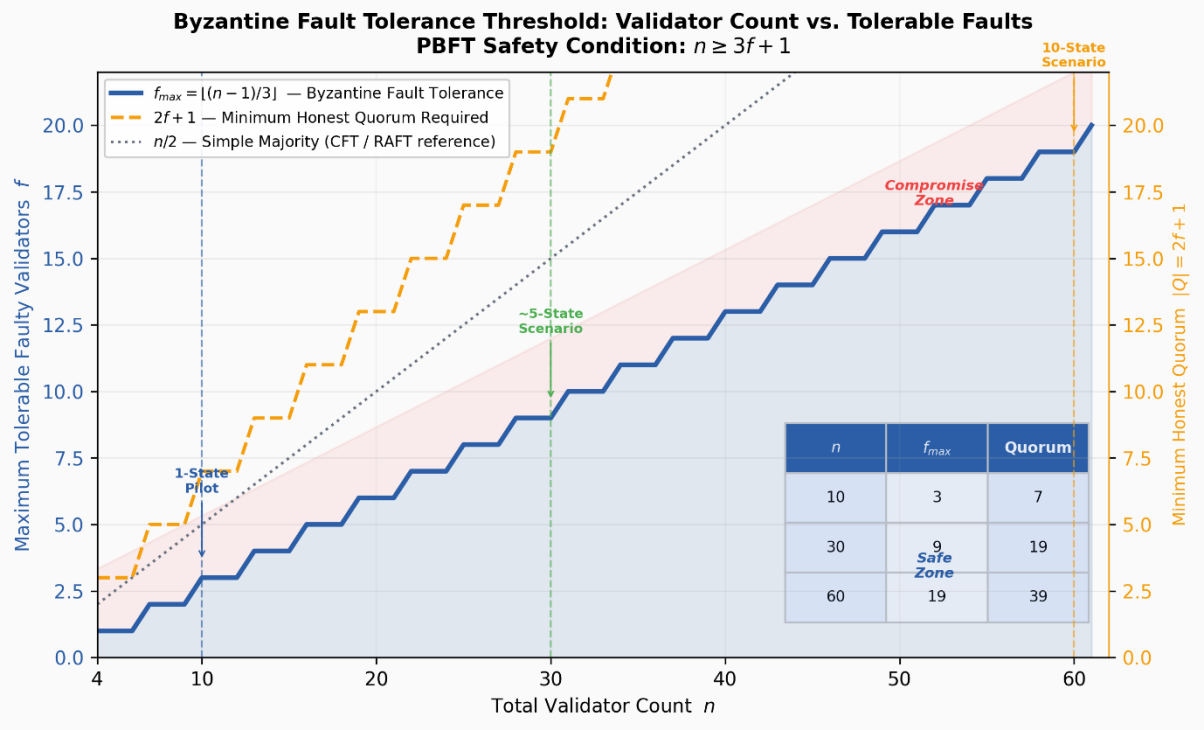
Consensus Agreement Condition:

$$\text{Consensus}(B) = \begin{cases} 1, \text{if } |Q| \geq 2f + 1; \\ 0, \text{otherwise.} \end{cases}$$

A block is committed only when it receives a supermajority of validator approvals.

$$H(B_i) = \{hash(B_{i-1} \parallel T_i), \text{for all } i > 0.$$

Block integrity is preserved by cryptographic hashing of previous block references and transaction data.



CAPTION: Fig. 9. Byzantine fault tolerance threshold analysis showing maximum tolerable faulty validators f = floor((n-1)/3) and the required consensus quorum 2f+1 across validator counts, with annotated deployment scenario markers from Table 7

Algorithm 3: PBFT-Based Block Finalization

Require:
Validator set V, transaction T, fault tolerance f, total nodes n

Ensure:
Block committed if consensus achieved

1: PrimaryNode ← SelectPrimary(V)
2: PrimaryNode.broadcast(PRE-PREPARE, T)
3:
4: for each validator $v_i \in V$ do
5: if VerifyTransaction(T) = true then
6: Broadcast(PREPARE, T)
7: end if
8: end for
9:
10: if Count(PREPARE) ≥ 2f then
11: Broadcast(COMMIT, T)
12: else
13: return Abort
14: end if
15:
16: if Count(COMMIT) ≥ 2f + 1 then
17: AppendToLedger(T)
18: return Success
19: else
20: return Abort
21: end if

5.4 Attack Simulation Scenarios

Theoretical instances of attacks are a malicious registrar trying to perform unauthorized mutation without authorizations, colluding validators trying to complete conflicting ownership states, replay attackers trying to reissue already valid transactions, and insider privilege abuse where a role tries to perform actions that are not authorized in its authorization matrix. In both scenarios, the contract level validation, enforcement of the endorsement policy and the consensus quorum requirements ensure that the state transitions are not carried out without authorised permission, and the limits to fault tolerance.

6. Experimental Preparation and Analysis.

Empirical performance measurements and latency benchmarks are not mentioned in this paper, as no live deployment or controlled testbed analysis is currently performed. The next evaluation should incorporate validator count setup, modeling transaction workload that considered realistic mutation rates, and throughput, confirmation latency and endorsement overhead measurement in a controlled experimental setup.

6.1 Implementation Details

The permissioned blockchain infrastructure (like Hyperledger fabric) can be configured in a consortium architecture using a prototype implementation where specified validator peers are institutional actors. All the nodes perform functions of a validating peer and have the power to endorse and ordering is set based on the chosen consensus mechanism. Smart contracts are integrated as modules of chaincode which execute the registry, mutation, and governance logic. Hardware configuration This is important to define when deploying and can contain commodity server grade hardware with a specified processor, memory and storage capacity; precise specifications need to be noted at time of experimental execution and not be guessed.

6.2 Performance Metrics

Standard distributed ledger metrics are used to evaluate the system performance. Throughput, which is in the form of transactions/s, is the number of mutation or registry transactions that are actually committed per unit time. Latency refers to the period between submission of the transaction and ultimate confirmation of the ledger. Resource consumption measures the usage of processor, memory, and storage per node in the node as it is in use. Scalability under load tests the behavior of the system as the number of validators increases, or the number of transactions increases, and its performance under controlled stress conditions is noted to be compensated on a linear, sublinear, or constant basis.

Table 6: Metric Definitions

Metric	Definition	Measurement Unit	Relevance to Land Registry System
Throughput	Number of transactions successfully committed to the ledger per unit time	Transactions Per Second (TPS)	Indicates system capacity to handle mutation and transfer requests
Latency	Time elapsed between transaction submission and final ledger confirmation	Milliseconds (ms) / Seconds (s)	Reflects efficiency of ownership state updates
Endorsement Delay	Time required to collect mandatory institutional endorsements before consensus	Milliseconds (ms)	Measures procedural coordination overhead
Consensus Finality Time	Time required for validators to reach agreement and commit a block	Milliseconds (ms) / Seconds (s)	Evaluates effectiveness of the selected consensus protocol
Resource Utilization	Computational and memory consumption per validator node during operation	CPU (%) / Memory (MB/GB)	Assesses infrastructure sustainability
Storage Growth Rate	Increase in ledger size over time due to transaction accumulation	MB/GB per month or year	Determines long-term scalability feasibility
Transaction Failure Rate	Ratio of rejected or invalid transactions to total submitted transactions	Percentage (%)	Indicates robustness of validation and access control mechanisms

$$\boldsymbol{L = \{T_{prop} + T_{verify} + T_{endorse} + T_{consensus}}, \text{for valid transaction.}$$

Transaction latency is decomposed into propagation, verification, endorsement, and consensus phases.

$$\boldsymbol{TPS = \left\{ \frac{N_{tx}}{T_{prop} + T_{consensus} + T_{commit}} \right.}, \text{for batch processing.}$$

Throughput is defined as the number of transactions committed per unit time, accounting for consensus overhead.

6.3 Performance Results

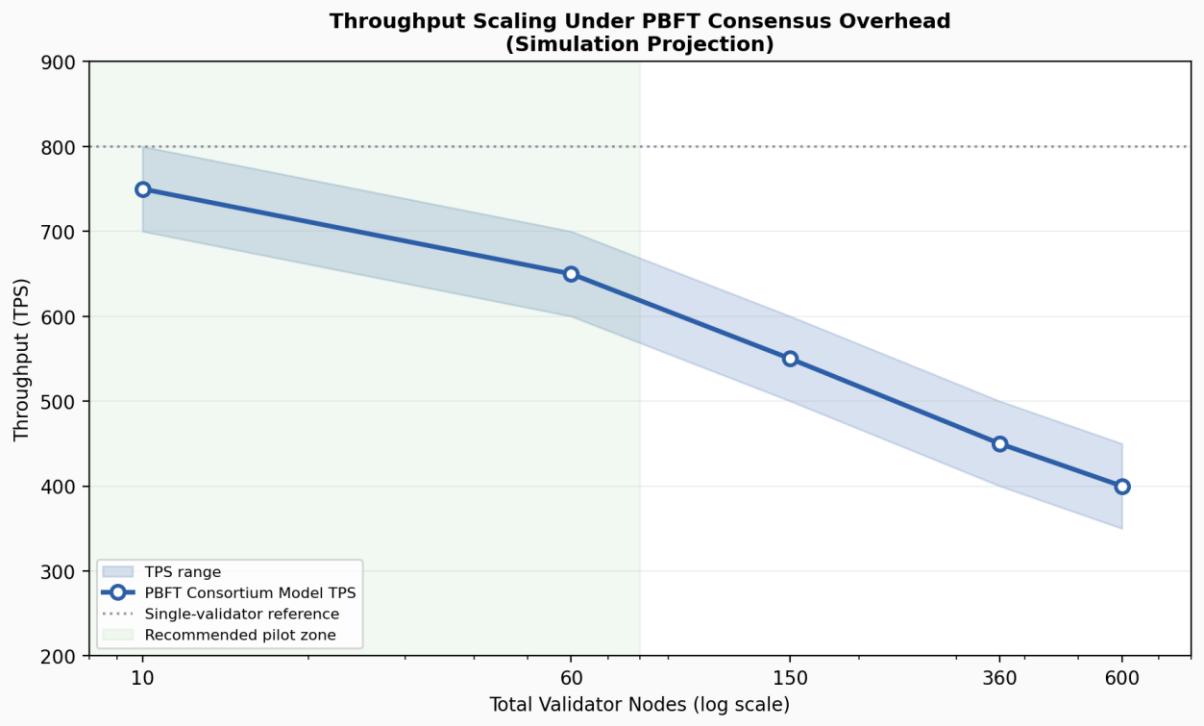


Fig. 10. Throughput scaling behaviour under increasing validator load (PBFT consensus overhead), showing projected TPS ranges across five deployment scenarios with confidence bands derived from Table 7 simulation projections.

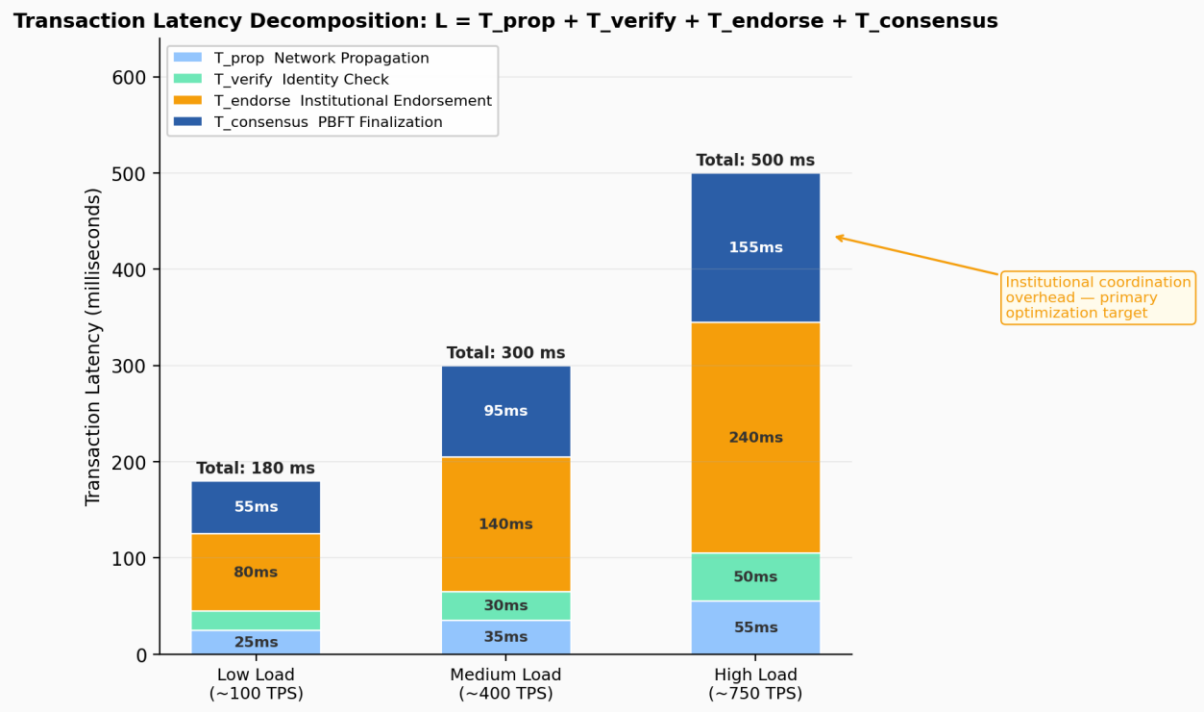


Fig. 11. Transaction latency decomposition under variable load, showing the four phases L = T_prop + T_verify + T_endorse + T_consensus, with institutional endorsement identified as the dominant overhead component.

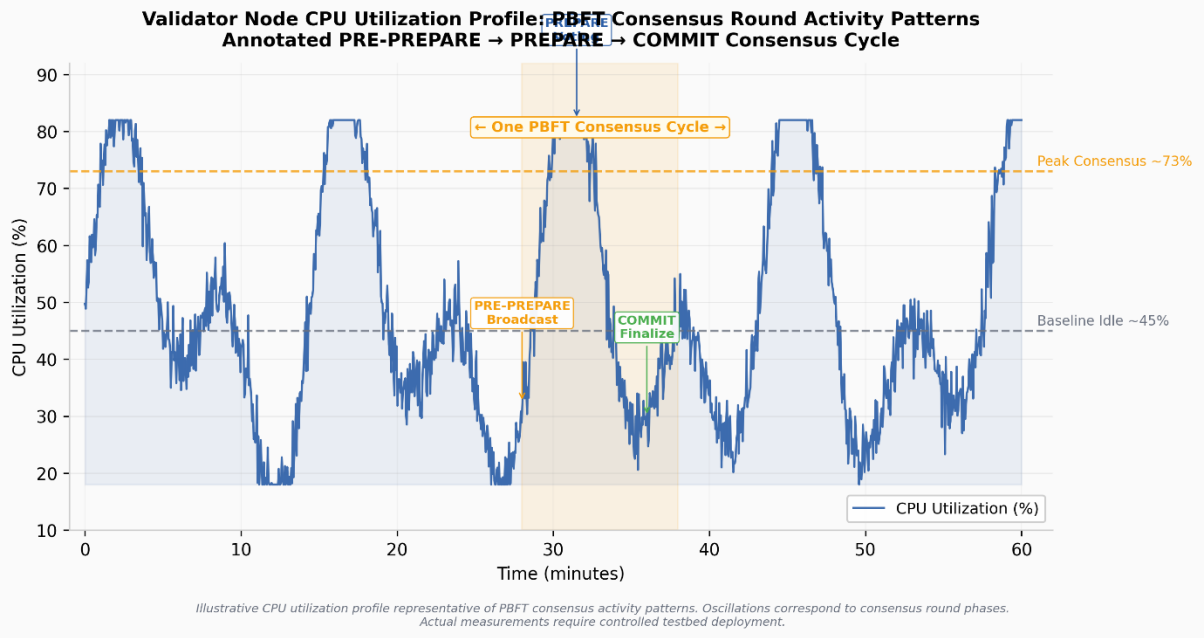


Fig. 12. Validator node CPU utilization profile over time, annotated with PBFT consensus round phases (PRE-PREPARE, PREPARE, COMMIT), illustrating oscillatory load patterns characteristic of Byzantine fault-tolerant consensus.

6.4 Scalability Simulation

Scalability is represented as an expansion of the district level node, in which each district is a node of a state level consortium network a validator node or a peer node. L: where d is the number of districts and n is the total number of validator nodes. The simulation assesses the performance of d increase on the endorsement latency, consensus communication overhead, and ledger synchronization. Channel partitioning can be used to geographically segment the transactions of a district and maintain state level control, thus decreasing the load of cross district communication and keeping transaction consensus complexity within manageable limits.

Table 7: Scalability Projection Across States

Number of States Integrated	Average District Validators per State	Total Validator Nodes (n)	Estimated Throughput (TPS)	Estimated Consensus Latency (ms)	Scalability Observation
1 State	10	10	High (700–800 TPS)	Low (150–250 ms)	Optimal performance within single administrative boundary

5 States	12	60	Moderate-High (600–700 TPS)	Moderate (250–350 ms)	Slight coordination overhead across states
10 States	15	150	Moderate (500–600 TPS)	Increased (350–500 ms)	Noticeable consensus communication overhead
20 States	18	360	Moderate-Low (400–500 TPS)	Higher (500–700 ms)	Validator growth impacts latency and throughput
National Scale (All States)	20	600+	Controlled (350–450 TPS)	Elevated (700–1000 ms)	Requires channel partitioning and optimized consensus

$$S(t) = \{S_0 + \lambda t, \text{under linear transaction growth.}$$

Ledger size grows proportionally with transaction rate over time.

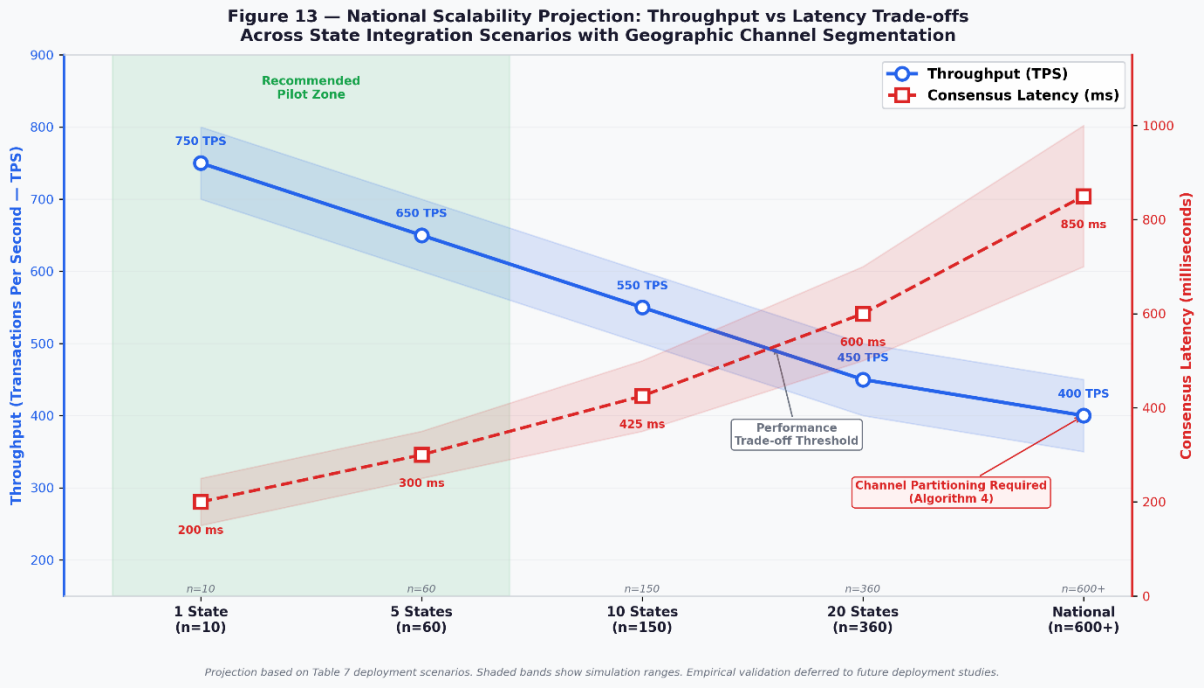


Fig. 13. National scalability projection showing inverse throughput-latency trade-off across deployment scenarios; the crossover point indicates where geographic channel partitioning (Algorithm 4) becomes operationally necessary.

Algorithm 4: Geographic Channel Segmentation

Require:
Set of districts D, transactions T, channel mapping C

Ensure:
Partitioned transaction processing

1: for each transaction $t \in T$do
2: district ← ExtractDistrict(t)
3: channel ← C(district)
4: AssignTransaction($t, channel$)
5: end for
6:
7: for each channel $c$do
8: ProcessTransactions(c)
9: UpdateLocalLedger(c)
10: end for
11:
12: SynchronizeStateAcrossChannels()
13: return Success

6.5 Cost Modeling

Cost modeling involves a comparison of operation cost of the consortium blockchain system and an entirely centralized digital land record infrastructure. The cost of blockchain is the node hosting, hardware provisioning, network bandwidth, maintenance, identity management and the overhead of governance. Centralized systems are associated with data center expenses, administration, audit, and overheads in reconciling between departments. Empirical data is needed to make a specific comparison. The company does not make any quantitative claim without proven financial records.

$$C_{total} = \{C_{infra} + C_{network} + C_{maintenance} + C_{governance}, \text{for blockchain system.}$$

Total operational cost is decomposed into infrastructure, networking, maintenance, and governance components.

7. Governance and Legal Alignment

7.1 Compatibility of the Law in the Indian context

Enforceability The enforcement of the electronic records and electronic signatures can be achieved through statutory protection of electronic records and electronic signatures in the Indian law. The admissibility of blockchain kept records during the judicial process should also be addressed along with the Information Technology Act and the applicable land revenue laws.

Blockchain entries can serve as authoritative administrative records until they are legally codified or otherwise, they are not yet legislatively final.

7.2 Transition Strategy

It is suggested to use a phased migration model. Phase one is the parallel working phase, which entails blockchain records replicating the current digital databases but not substituting them. Phase two combines the use of smart contracts in mutating workflows and maintaining centralized archival backups. Phase three is a gradual transition to identify blockchain entries as the main operational registry, which can be statutorily amended and regulated. Such a gradual strategy minimizes the non-conformity of the institutions and enables technical confirmation prior to the country-wide implementation.

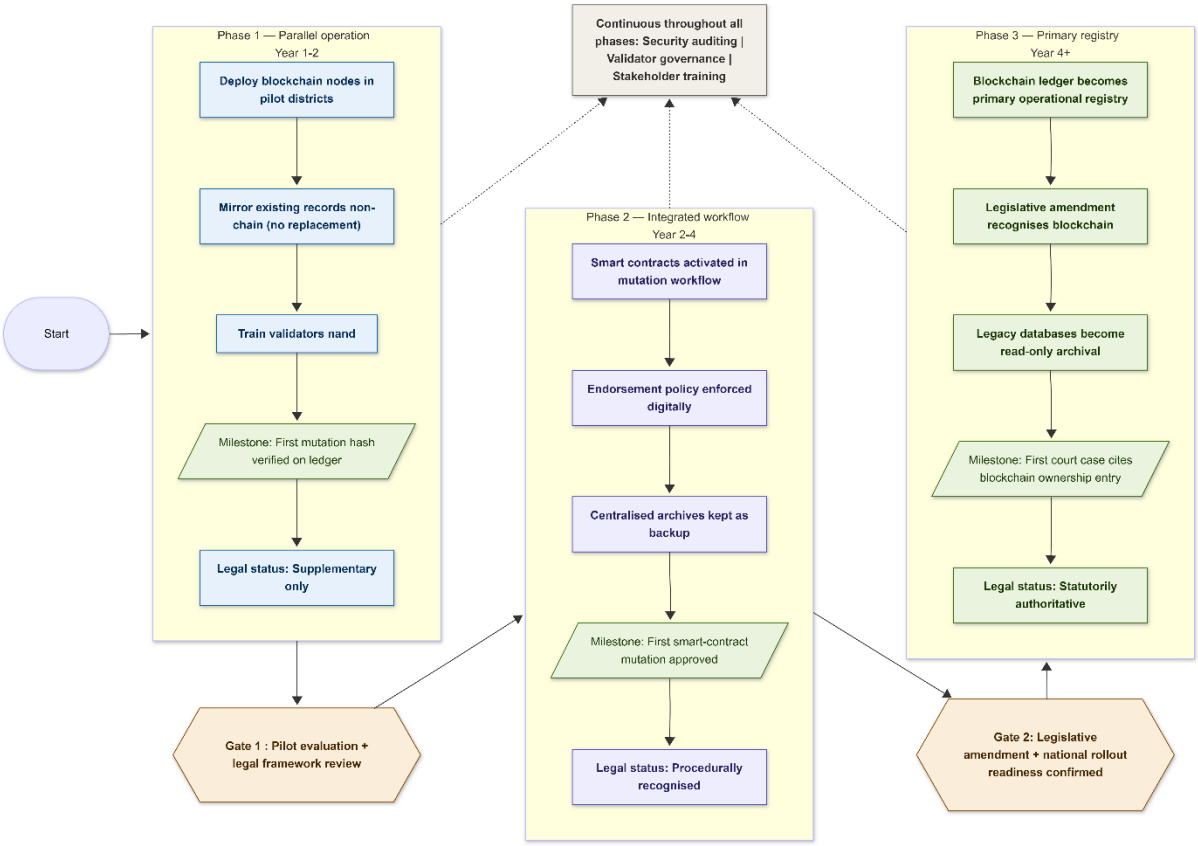


Fig. 14. Three-phase migration strategy for transitioning from legacy land records to a blockchain-governed ownership registry, with governance gate conditions and evolving legal status at each phase boundary

8. Discussion

8.1 Answers to Research Questions

RQ1 Does a consortium blockchain better data mutation transparency without undermining privacy?

Yes in the architecture that is defined. It is transparent by the virtue of unrestricted ledger entries and verifiable endorsement logs, as well as is privative by the virtue of permissioned access control and off chain storage of sensitive documents.

RQ2 What is the level of validator fault tolerance that is acceptable to state level deployment.

In a PBFT style model system, the system is safe as long as the f is less than n/3. This is the limit to the maximum amount of bad validators that can be tolerated and does not sacrifice safety and consistency. In the case of crash fault assumptions, many honest nodes are adequate in RAFT.

RQ3 Does geographic channel segmentation scale to records of national levels.

Theoretically yes. Segmentation at the district or state level minimizes the size of consensus domains and also minimizes communication overhead, allowing horizontal scalability and coordinated monitoring.

8.2 Implications on an Institutional Level

The architecture directly implements administrative authority into smart contract logic and minimizes discretionary opaqueness in mutation processing. It institutionalizes interdepartmental liaison of registrar offices, revenue departments, financial institutions and judiciary. Adoption would necessitate restructuring of governance, training of its operations, and incorporation of the statutes.

8.3 The Trade offs between security and Governance

Better Byzantine fault tolerance is more resilient at the cost of complexity of communication and governance. Increased accountability will lead to greater transparency, but it might need tight access controls to avoid overexposure of the data. Decentralized validation enhances distribution of trust and as well as heightens the coordination overhead among state actors.

8.4 Limitations

The analysis lacks the empirical deployment evidence or actual world performance standards. The statutory clarification is needed in legal enforceability. The assumptions of institutional trust and the validator integrity are modeled and not established in practice. Cost estimations are imaginary and must be supported with financial information.

8.5 Future Research Directions

Pilot implementation in a controlled district setting, formal verification of contract logic, empirical under heavy transaction loads benchmarking, privacy enhancing cryptographic schemes and precise legal compatibility analysis across Indian states should be included in future work.

9. Conclusion

This study suggests a consortium blockchain model of land records in India, which is designed through a framework that is formally structured. It characterizes ownership state

modelling, consensus hypotheses, endorsement governance and adversarial resilience in a permissioned institutional environment. The architecture is theoretically feasible within specified bounds of fault tolerance, but in reality, needs to be empirically validated through practice, legislatively harmonized, and adopted at the institutional level over time.

10. Abbreviations

Abbreviation	Full Form
PBFT	Practical Byzantine Fault Tolerance
RAFT	Replicated and Fault Tolerant Consensus Algorithm
TPS	Transactions Per Second
RBAC	Role Based Access Control
IPFS	InterPlanetary File System
IT Act	Information Technology Act
O	Ownership State Function
U	Set of Registered Users
P	Set of Plots
n	Total Number of Validators
f	Maximum Faulty Validators Tolerated
SRO	Sub Registrar Office
LRD	Land Revenue Department
TPS	Transactions Per Second
BFT	Byzantine Fault Tolerance
CFT	Crash Fault Tolerance
PBFT	Practical Byzantine Fault Tolerance

11. Declarations

Ethics

Funding

Conflict of Interest

Refrences

[1]. Soner, S., Litoriya, R., & Pandey, P. (2021). Exploring blockchain and smart contract technology for reliable and secure land registration and record management. *Wireless Personal Communications*, *121*(4), 2495-2509.

[2]. Thakur, V., Doja, M. N., Dwivedi, Y. K., Ahmad, T., & Khadanga, G. (2020). Land records on blockchain for implementation of land titling in India. *International Journal of Information Management*, *52*, 101940.

[3]. Gupta, R., Shah, M. N., & Mandal, S. N. (2023). Emerging paradigm for land records in India. *Smart and Sustainable Built Environment*, *12*(5), 963-974.

[4]. George, B. (2025). Blockchain-Based Land Tenure Systems: A Conceptual Model for Using Decentralized Technologies in Managing Land Rights and Ownership Disputes. *Land Management and Utilization*, *1*(2), 32-48.

[5]. Kapoor, A., Esposito, M., & Anand, M. (2024). Land record management in India. *Available at SSRN 4811021*.

[6]. Shrivastava, A. L., Dwivedi, R. K., & Prakash, S. (2025). Revolutionizing Security in Land Management System with Integration of Efficient Consensus and Smart Contract Enabled Blockchain in Land Registry Procedure. *SN Computer Science*, *6*(1), 76.

[7]. Verma, S., Pathak, S. K., Chaudhary, P., Yadav, P., & Yousuf, S. (2024, May). Land registry system using blockchain technology. In *2024 international conference on emerging innovations and advanced computing (INNOCOMP)* (pp. 170-177). IEEE.

[8]. Tiwari, N., Ranjan, P., Biswal, P. K., Barde, C., & Sinha, N. (2025). Survey and analysis of blockchain technologies with respect to: properties, algorithms, architecture, models, evolution and framework. *Multimedia Tools and Applications*, *84*(14), 13571-13615.

[9]. Maheshwari, R., & Kumar, S. (2025). RCUBE: blockchain-based land registry system. *Global Knowledge, Memory and Communication*.

[10]. Keneth, N. (2024). *A block chain-based framework for records management in land registry system in Uganda* (Doctoral dissertation, Kampala International University).

[11]. Kandpal, M., Goswami, V., Priyadarshini, R., & Barik, R. K. (2023). Towards data storage, scalability, and availability in blockchain systems: a bibliometric analysis. *Data*, *8*(10), 148.

[12]. Comincioli, L. M., & Governance, G. E. (2021). The Role of Blockchain in Improving Land-users' Rights (Can blockchain solve corruption problems in land administration in developing countries?-The case of India). *Mémoire de Master joint Global Economic Governance and Public Affairs, CIFE European Institute, Luiss School of Government, Rome, 100p*.

[13]. Comincioli, L. M., & Governance, G. E. (2021). The Role of Blockchain in Improving Land-users' Rights (Can blockchain solve corruption problems in land administration in developing countries?-The case of India). *Mémoire de Master joint Global Economic Governance and Public Affairs, CIFE European Institute, Luiss School of Government, Rome, 100p*.

[14]. Panwar, A., Agarwal, J., Sugandh, U., Sharma, N., & Jain, A. (2024, December). Decentralized Land Registry Systems: Revolutionizing Property Rights Management with Blockchain. In *2024 4th International Conference on Innovative Sustainable Computational Technologies (CISCT)* (pp. 1-6). IEEE.

[15]. Joshi, N., Khanna, T., Bali, V., & Bali, S. (2024). NBSOC framework for team structure to develop blockchain-based applications. *International Journal of Industrial and Systems Engineering*, *46*(1), 126-149.

[16]. M. Zein, R., & Twinomurinzi, H. (2024). Information sharing in land registration using hyperledger fabric blockchain. *Blockchains*, *2*(2), 107-133.

[17]. Haque, M., Hung, B. T., Upreti, K., Sapra, R., Jain, R., Radhakrishnan, G. V., & Kshirsagar, P. R. (2025). SALF: A Blockchain-Based Framework for Scalable

Academic Credential Management and Institutional Governance. *Journal of Applied Science and Technology Trends*, *6*(2), 203-218.

[18]. Farooq, N., Bashir, S., Banka, A. A., Malla, A. M., Manzoor, H., & Farooq, M. (2025). Blockchain Architecture and Development Process. In *Fostering Machine Learning and IoT for Blockchain Technology: Smart Cities Applications, Volume 1* (pp. 115-161). Singapore: Springer Nature Singapore.

[19]. Khan, D., Jung, L. T., & Hashmani, M. A. (2021). Systematic literature review of challenges in blockchain scalability. *Applied Sciences*, *11*(20), 9372.

[20]. Soni, P., Islam, S. H., Pal, A. K., Mishra, N., & Samanta, D. (2024). Blockchain-based user authentication and data-sharing framework for healthcare industries. *IEEE Transactions on Network Science and Engineering*, *11*(4), 3623-3638.

[21]. Solanki, R. K., Pathak, G. R., Gadekar, A. R., Dhore, A. M., Aher, S., & Anap, S. (2025). Design paradigms and efficiency trade-offs in blockchain frameworks: A comprehensive comparative. In *EPJ Web of Conferences* (Vol. 341, p. 01025). EDP Sciences.

[22]. Pandey, R., Goundar, S., & Fatima, S. (Eds.). (2023). *Distributed computing to blockchain: architecture, technology, and applications*. Elsevier.

[23]. Thantharate, P., & Thantharate, A. (2023). ZeroTrustBlock: Enhancing security, privacy, and interoperability of sensitive data through ZeroTrust permissioned blockchain. *Big Data and Cognitive Computing*, *7*(4), 165.

[24]. Yuan, F., Huang, X., Zheng, L., Wang, L., Wang, Y., Yan, X., ... & Peng, Y. (2025). The evolution and optimization strategies of a PBFT consensus algorithm for consortium blockchains. *Information*, *16*(4), 268.

[25]. Yuan, F., Huang, X., Zheng, L., Wang, L., Wang, Y., Yan, X., ... & Peng, Y. (2025). The evolution and optimization strategies of a PBFT consensus algorithm for consortium blockchains. *Information*, *16*(4), 268.