Peer-to-Peer Networking and Applications (2026) 19:90
https://doi.org/10.1007/s12083-026-02227-5

RESEARCH

BERBLOM: Blockchain-enabled reliable land ownership management system using ERC-4907 smart contract

Odinachi Udemezuo Nwankwo[1] · Muhammad Rasyid Redha Ansori[2] · Esmot Ara Tuli[1] · Dong-Seong Kim[1] · Jae-Min Lee[1]

Received: 19 September 2025 / Accepted: 16 March 2026

Abstract

Land administration in Nigeria continues to face critical challenges, including document inconsistencies, opaque verification procedures, slow processing times, and persistent administrative malpractice associated with centralized and conventional, non-automated record-management systems. These systemic weaknesses create opportunities for double allocation, unauthorized alterations, and hidden registry manipulation, highlighting the need for more secure and transparent digital approaches. This paper proposes BERBLOM, a blockchain-enabled land ownership management system tailored to Nigerian land-registry workflows. The system tokenizes land parcels as ERC-4907 assets and stores supporting documents in the InterPlanetary File System (IPFS) to enhance data integrity and reduce single points of failure. BERBLOM supports practical use cases such as government plot allocation, peer-to-peer property transfers, and an on-chain leasing mechanism that automatically reverts ownership upon lease expiration. The framework operates on a permissioned Hyperledger Besu network using the Quorum Byzantine Fault Tolerance (QBFT) consensus protocol, selected for its Byzantine fault tolerance, predictable gas-free operation, and scalability under controlled conditions. Experimental evaluation indicates that QBFT maintains stable throughput and manageable latency as the number of validator nodes increases from 4 to 32. A Flutter-based decentralized application (DApp) enables real-time interaction through event-driven synchronization. While human-driven corruption and malpractice cannot be fully eliminated, the proposed prototype demonstrates how blockchain-based registries can reduce opportunities for hidden technical manipulation and improve on-chain accountability. It is important to note that the primary performance benefits of BERBLOM are realized after tokenization. Initial land registration remains dependent on legally mandated, human-driven governmental approvals; however, post-registration operations such as peer-to-peer transfers and on-chain leasing can be executed without intermediaries and complete within seconds. Consequently, the performance analysis focuses on the lifecycle stages where blockchain automation provides the greatest impact.

Keywords Blockchain · Land · Smart contract · IPFS · ERC-4907 token

✉ Jae-Min Lee
ljmpaul@kumoh.ac.kr

Odinachi Udemezuo Nwankwo
odinachifoot@gmail.com

Muhammad Rasyid Redha Ansori
rasyidred@nslab.tech

Esmot Ara Tuli
esmot@kumoh.ac.kr

Dong-Seong Kim
dskim@kumoh.ac.kr

[1] Kumoh National Institute of Technology, Gumi 39177, Gyeongsangbuk-do, South Korea

[2] NS Lab Inc., Gumi 39177, Gyeongsangbuk-do, South Korea

1 Introduction

Since the enactment of the Land Use Act (LUA) in 1978, the Federal Capital Territory (FCT) administration has been vested with authority over all lands and properties in Nigeria, subject to compensation for prior claims as specified in the Act. In the late 1990s, the government introduced privatization policies that facilitated the transfer of properties and land to private individuals, contributing to the growth of a commercial real estate sector in Abuja [22, 40]. In 2003, the Abuja Geographic Information Systems (AGIS) was established to manage real estate matters, providing computerized records and data services for the Federal Capital

Published online: 11 May 2026

Territory. AGIS issues a Certificate of Occupancy (C of O) for each property and records Deeds of Assignment to document ownership transfers. The C of O functions as the primary legal proof of ownership. Typically, AGIS records consist of a C of O along with one or more Deeds of Assignment, depending on the number of times the land has changed hands and been registered with AGIS [28].

Figure 1 is a visual summary of the land transfer process in Nigeria. It displays each step in the transaction, from property advertisement to legal transfer of ownership, making the procedure easier to understand at a glance. The diagram illustrates the real estate transaction process in Abuja, Nigeria, highlighting the sequence of legal and administrative steps involved in property transfer. The process begins with a property being advertised by the seller through agents, followed by the buyer expressing a declaration of interest. A lawyer is then engaged to facilitate acquisition, providing the buyer with essential property documents, such as the Certificate of Occupancy or Deed of Assignment. The lawyer conducts a legal search to ensure the property title is valid, which involves verifying the authenticity of documents with the Abuja Geographic Information System (AGIS) and confirming that the property is unencumbered. Transfer documents, including a new deed of assignment and sale agreement, are then drafted by the lawyer. Both parties sign the new deed in the presence of two witnesses, after which the documents are formally registered with AGIS. Finally, ownership is legally transferred, and payment is completed. This workflow emphasizes the multiple layers of verification and legal oversight necessary to ensure secure and legitimate property transactions in Abuja.

In the current property transaction system, both buyers and sellers face numerous fees at every step. Sellers pay real estate agents a percentage of the sale price, while buyers incur expenses like lawyer fees, legal searches, and agent fees. This lucrative industry has attracted many, including some with dishonest intentions, leading to frequent property fraud cases. Buyers, especially those unfamiliar with Abuja's property transfer laws, are often victims. Land asset property transaction involves buyers, sellers, and witnesses signing a purchase agreement. However, this simple process is highly susceptible to fraud, as ownership records may not be updated, leading to confusion. Verification through the existing AGIS system is cumbersome, requiring physical visits, paperwork, and coordination with multiple departments. This lack of efficiency in the land registration system can lead to delays and errors, as outdated records may prevent the timely identification of fraudulent transactions [56]. The summary of the challenges of the conventional land record system is depicted in Fig. 2 [29].

A promising solution to the aforementioned challenges lies in the adoption of blockchain technology. This technology has been widely embraced across various sectors, including information management and security protocols [2, 34, 45, 54, 55, 57], healthcare systems [16, 23, 38, 49, 51, 52, 58], auctions [1, 9, 21, 26, 48, 50], and loyalty programs [20, 31–33, 41]. Each piece of data is securely stored within a chain of blocks, with each block cryptographically linked to its preceding blocks. As a result of this decentralized architecture, the system is fortified with robust security measures and does not require control by a singular authority [12], effectively safeguarding it against tampering and ensuring the integrity of the ledger, as modifications to it are inherently prohibited [18].

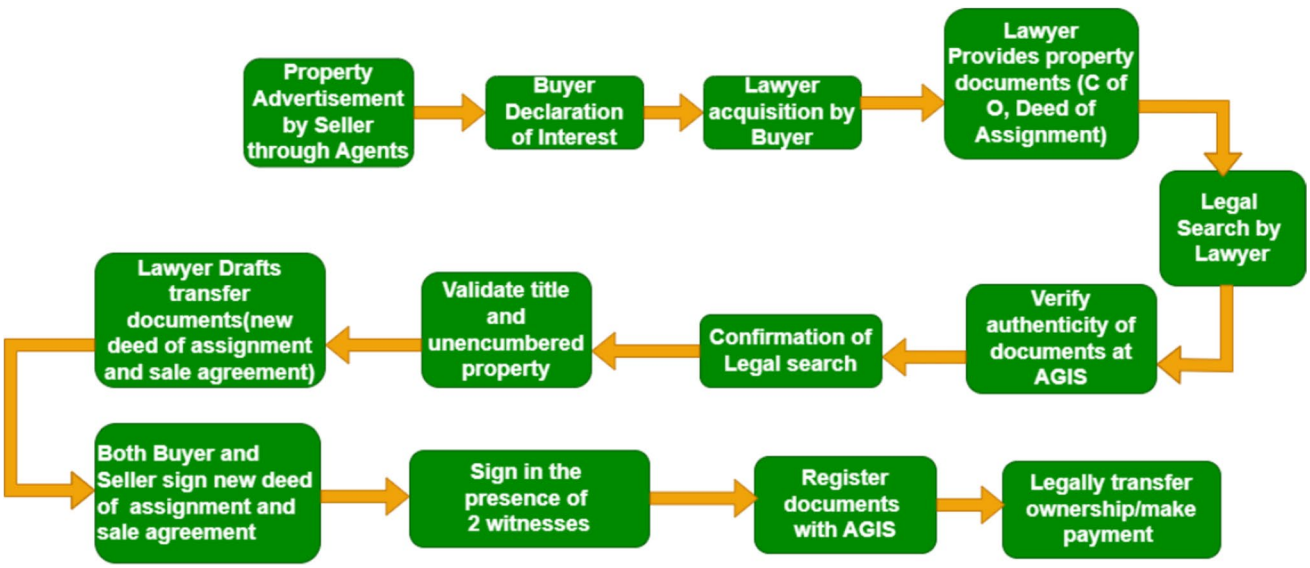


Fig. 1 Real estate transaction process in Abuja, Nigeria, AGIS : Abuja Geographic Information System

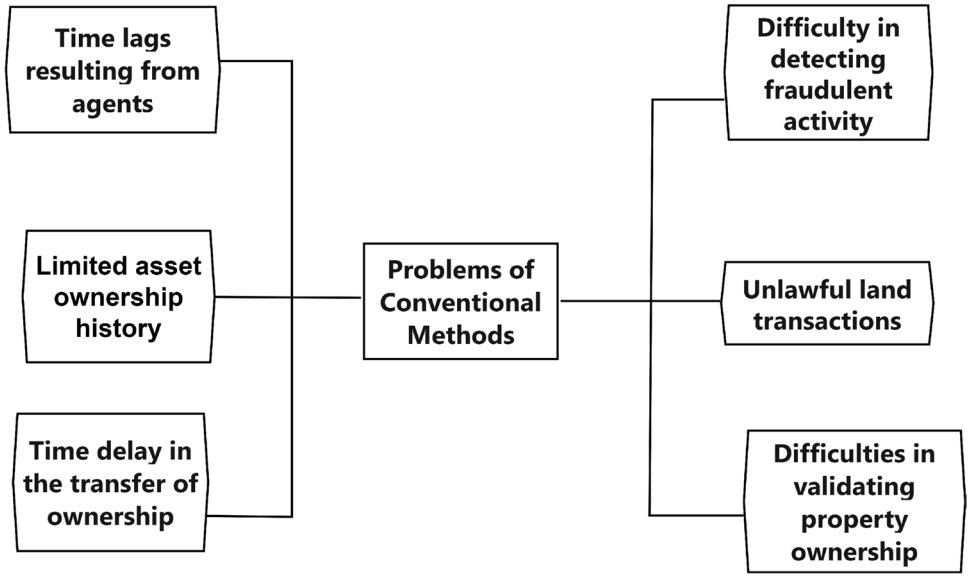


Fig. 2 Problems associated with conventional land record system

The summary of main contributions of this paper are as follows.

- A blockchain-based land registration system is introduced, utilizing the ERC-4907 standard and IPFS. This system is proposed as a better alternative to paper-based and centralized online-based land management systems, which are vulnerable to single points of failure and unauthorized alterations.
- To reflect practical land administration dynamics, the smart contract architecture was meticulously engineered around four comprehensive transaction paradigms—land registration, governmental allocation, peer-to-peer ownership transfer, and temporal leasing. Each operational scenario was rigorously modeled through formalized algorithms and interaction diagrams to ensure robust on-chain execution. Notably, the inclusion of a rental mechanism facilitated ephemeral ownership delegation via tokenId, with automated reversion upon expiration, thereby preserving asset integrity while supporting flexible tenure arrangements.
- A decentralized approach is implemented, allowing transactions to be conducted directly between buyers, sellers, and renters, without the involvement of brokers or middlemen.
- To further enhance privacy and efficiency, the system is designed and built on Hyperledger Besu, a permissioned blockchain. This ensures the privacy of network activities and avoids the high costs associated with gas fees on public blockchain networks such as Ethereum.

Note that the impact of the proposed system is most significant after a land parcel has been registered and tokenized on the blockchain. The initial registration process in Nigeria inherently requires human actions such as physical land inspection, document verification, and governmental approval which remain unavoidable and dominate overall processing time regardless of the underlying digital system. BERBLOM does not attempt to remove these legal and administrative procedures. Instead, the framework provides substantial improvements in the post-registration phase, where peer-to-peer land sales and temporary leasing traditionally require in-person visits, paperwork, and intermediaries. These post-tokenization operations are fully automated through smart contracts in BERBLOM, allowing them to complete in seconds with full auditability.

The remainder of this article is organized as follows: Section 2 reviews existing blockchain-based land registry systems in the Related Works section, highlighting their limitations, particularly in scalability, privacy, and the lack of support for temporary land-use rights. Section 3 introduces the Proposed System, which leverages ERC-4907 tokenization, IPFS for document storage, and Hyperledger Besu for a permissioned, gas-free network. Section 3.1 details the System Architecture, while Section 3.3 outlines the Operational Workflows, describing the algorithms for land registration, transfer, and leasing. Section 4 evaluates the system's efficiency in the Performance Metrics, Experimental Setup, and Results section, followed by Section 4.6, which discusses how the system ensures transparency and accountability in the Fraud Prevention and Official Accountability Model. Section 4.7 contrasts BERBLOM with other land registry systems in the Comparison with Existing Solutions, and Section 4.8 addresses the challenges faced during implementation in the Limitations section. Finally, Section 5 summarizes the potential of the system and suggests areas for future research in the Conclusion, with Appendix A providing technical details on the IPFS storage mechanism.

2 Related works

Blockchain has emerged as a promising foundation for improving land administration by mitigating fraud, enhancing transparency, and accelerating property transactions. Existing research in this domain can be broadly categorized into solutions built on public blockchain networks, primarily Ethereum, and those developed using permissioned frameworks such as Hyperledger Fabric. As summarized in Table 1 , these studies demonstrate meaningful progress in securing ownership records and reducing opportunities for administrative manipulation. Nevertheless, the comparison reveals several persistent limitations across both categories. Public network–based approaches often benefit from stronger interoperability but suffer from volatility in transaction costs and expose sensitive registry operations to privacy risks. Conversely, permissioned solutions provide controlled governance and lower operational costs but typically rely on non-standardized asset models and lack decentralized off-chain storage, resulting in limited extensibility and scalability. Moreover, despite the widespread recognition of blockchain's potential, most existing frameworks focus exclusively on permanent ownership transfers and provide no support for temporary usage rights such as automated renting or leasing. Collectively, these observations indicate a critical research gap in designing a cost-efficient, standards-compliant, and fully decentralized architecture capable of supporting both permanent and temporary land-use transactions at national scale. To the best of our knowledge, this is the first work to integrate a gas-free private blockchain network, ERC-4907 tokenization, and IPFS in a framework designed specifically for the Nigerian land registry system.

2.1 Background: ERC-4907 and decentralized storage

The Ethereum Request for Comment 721 (ERC-721) Non-Fungible Token (NFT) standard serves as the foundational framework for representing unique, indivisible digital assets. By enforcing a strict correspondence between a

Table 1 Comparison of state-of-the-art approaches with proposed system

Study	Year	Key Findings/Focus	Limitations
Mendi et al. [30]	2020 [30]	Hyperledger Fabric for tax verification	No tokenization, absence of performance metrics
Gollapalli et al. [15]	2020 [15]	Hyperledger Fabric and CFT consensus for anti-tampering	State-based asset modeling
Jahan et al. [24]	2020 [24]	Satellite Chain + Hashing for string data storage	Hashed string data storage, absence of asset tokenization
Mintah et al. [27]	2020 [27]	Public Ethereum proposal for Ghana land transparency	Privacy risks and costs concerns, conceptual framework (no prototype)
Sladić et al. [42]	2021 [42]	Public Ethereum architecture for Serbian land records	Privacy risks and costs concerns, conceptual framework (no prototype)
Panda et al. [35]	2021 [35]	Public network solution for general land management	Unspecified consensus mechanism, no tokenization
Aborujilah et al. [6]	2021 [6]	Public Ethereum with PoW	Requires high energy, no tokenization
Alam et al. [3]	2022 [3]	Hybrid Ethereum for inter-dept coordination	Undefined consensus, non-standardized asset model
Stefanovic et al. [43]	2022 [43]	ERC-721 on Ropsten testnet for land trading	On-chain storage scalability constraints, lack of decentralized off-chain integration
Dubey et al. [11]	2023 [11]	Permissioned network using PoW	High computational overhead for private network, latency degradation under load
Hari et al. [19]	2023 [19]	T-PASS property exchange using ERC-721 standard	Centralized database for storage, validation limited to local simulation
Alyounis and Yasin [5]	2023 [5]	Hyperledger Fabric to prevent double spending	Raft (CFT) lacks malicious protection, misinterpretation of token standards
Sharma et al. [44]	2024 [44]	Public Ethereum registry using ERC-721 + IPFS	Public network privacy risks, lack of sovereign governance, no renting provision
Rashid et al. [39]	2025 [39]	AI/IoT + Blockchain framework using Ganache simulation	High estimated transaction costs, validation limited to simulation, no leasing
Gupta et al. [17]	2025 [17]	Ethereum prototype for reducing manual records	Unspecified consensus configuration, non-standardized data structures
Paavo et al. [36]	2025 [36]	Design science study for Namibian land registry	Simulated environment (Ganache), on-chain storage scalability constraints
Proposed	2025	Hyperledger Besu + IPFS + ERC-4907 for Renting/Sales	—

digital token and a specific asset, it provides the immutability and provenance lineage essential for digitizing land parcels and property deeds. However, the traditional ERC-721 model primarily addresses permanent ownership, which creates limitations for scenarios involving temporary property rights such as renting or leasing. To resolve this, the ERC-4907 extension was introduced to separate the concept of *ownership* from *usage*. This architectural improvement allows a digital land title to hold two distinct states simultaneously: the permanent legal owner and a temporary user. This distinction enables the secure delegation of property rights for a fixed duration, allowing usage to expire automatically without requiring manual intervention or transaction fees to reclaim the asset [4, 7, 53].

Complementing ERC-4907 is the private InterPlanetary File System (IPFS), a decentralized content addressed file storage protocol designed to store large external data objects outside the blockchain. IPFS assigns each file a unique Content Identifier (CID) based on its cryptographic hash, ensuring tamper evident persistence and enabling lightweight on chain references. This separation allows high volume documents, such as survey plans, Certificates of Occupancy, or geospatial metadata, to be stored efficiently while keeping only essential identifiers on the blockchain. Prior studies have demonstrated that combining NFTs with decentralized storage mechanisms improves interoperability, auditability, and scalability for land registry applications [43, 44]. Together, NFTs and IPFS provide a standards compliant and tamper resistant foundation for modeling land titles as unique digital assets while ensuring secure and decentralized access to associated documentation.

2.2 Comparative analysis of existing blockchain-based land registry solutions

As summarized in Table 1, contemporary efforts toward blockchain-enabled land management reveal significant architectural, functional, and performance constraints that limit their applicability for national-scale deployment. The comparison illustrates how prior approaches vary in token models, storage architectures, consensus mechanisms, and operational scope, including their ability (or inability) to support advanced features such as decentralized document storage or temporary land-use rights. To contextualize these gaps, the following studies illustrate how current approaches, while influential, exhibit architectural, functional, and performance constraints that limit their applicability for modern land administration.

Mendi et al. [30] proposed a blockchain-based framework for the Turkish land registry, primarily targeting tax evasion caused by the under-reporting of property values. They deployed a solution on Hyperledger Fabric using a Proof of Authority (PoA) consensus to involve municipalities and banks in the verification loop. However, from an architectural standpoint, their reliance on Hyperledger Fabric's standard chaincode introduces a limitation in asset liquidity and interoperability. It means land parcels are recorded as static ledger entries rather than unique, tradable digital assets. Consequently, the study omits critical performance benchmarking, such as transaction throughput and latency, making it impossible to assess its scalability for real-world deployment.

Gollapalli et al. [15] addressed the risk of administrative tampering by implementing Hyperledger Fabric with Crash Fault Tolerance (CFT) consensus. While this architecture provides robust security for static registry records, the asset modeling approach presents certain limitations. By relying on Fabric's native chaincode rather than a standardized token framework, the system treats land ownership as mutable ledger states. Consequently, this design constrains the system's flexibility regarding complex property rights, making it less adaptable for scenarios involving temporary usage or leasing compared to programmable digital asset models.

Jahan et al. [24] introduced a parallel blockchain architecture utilizing the Satellite Chain Formation algorithm, which focuses on securing string-based land data on a public network. Although this method effectively ensures data integrity, the decision to store records as hashed strings limits the system's functional scope. By not implementing a token standard such as ERC-4907, the system represents land as static data entries rather than unique, programmable assets. Consequently, this design precludes the native integration of advanced functionalities, such as automated ownership transfers or smart-contract-based leasing mechanisms.

Similarly, Mintah et al. [27] and Sladić et al. [42] proposed public Ethereum-based architectures to mitigate transparency issues in the land administration systems of Ghana and Serbia, respectively. While these studies correctly identify blockchain's immutability as a solution for data tampering, their reliance on the public Ethereum mainnet presents a significant barrier to practical adoption. Public networks impose volatile gas fees and expose transaction history to the open network, raising serious economic and privacy concerns that these theoretical frameworks failed to address. Furthermore, as these contributions were restricted to architectural proposals without functional prototypes, they offer no empirical data regarding system throughput or latency under load.

In similar Ethereum-based studies, Panda et al. [35] proposed a public network solution but did not specify the underlying consensus mechanism, rendering the system's security model and throughput potential difficult to verify. Conversely, Aborujilah et al. [6] implemented the

Proof-of-Work (PoW) consensus algorithm. However, PoW presents scalability challenges due to significant energy consumption and probabilistic transaction finality, which can cause high latency during network congestion. Furthermore, neither study adopted the ERC-4907 standard, relying instead on custom data structures. This approach results in a limited asset model, restricting interoperability with external wallets and marketplaces.

Similarly, Alam et al. [3] proposed a hybrid Ethereum-based framework to improve inter-departmental coordination in Bangladesh. However, the study does not specify the underlying consensus mechanism. This omission complicates the assessment of the system's transaction finality and resistance to collusion—critical attributes for government-level infrastructure. Furthermore, the architecture does not incorporate the ERC-4907 standard for unique asset differentiation. The use of a non-standardized data model constrains the system's ability to support unique asset differentiation and reduces its compatibility with the liquidity and programmability requirements of emerging digital real estate markets.

Stefanovic et al. [43] utilized the ERC-721 standard on the Ethereum Ropsten testnet, demonstrating a standards-compliant approach to land trading. However, the system design lacks an integrated decentralized off-chain storage solution, creating a potential bottleneck for scalability. Without off-chain storage, handling heavy documentation imposes significant resource requirements if stored on-chain, or reliance on centralized databases, which may compromise the decentralized nature of the system and introduce vulnerability to single points of failure.

In an effort to disintermediate land registration, Dubey et al. [11] deployed a permissioned blockchain network. However, their architectural decision to utilize the PoW consensus algorithm within a permissioned environment may introduce computational overhead and scalability bottlenecks. This limitation was reflected in the study's results, which indicated increased system latency as transaction volumes rose. Additionally, the study did not report gas usage metrics, which limits a full assessment of the smart contracts' economic feasibility.

Hari et al. [19] introduced 'T-PASS,' a property exchange framework that utilizes the ERC-721 standard to facilitate real estate bidding. However, their architecture relies on a hybrid model where critical audit trails and user data are maintained in a centralized database. This design choice creates a single point of failure and potential data synchronization discrepancies between the off-chain database and the on-chain ledger, undermining the trustless nature of the system. Furthermore, the technical evaluation was limited to a local Ganache simulation, which fails to account for the network latency and consensus overhead present in a production-grade distributed network.

Alyounis and Yasin [5] proposed a framework for land record management based on Hyperledger Fabric, focusing on mitigating double-spending and unauthorized modifications. While the implementation utilizes the Fabric Token SDK to ensure transaction integrity, the reliance on the Raft consensus protocol introduces certain security trade-offs. As a Crash Fault Tolerant (CFT) algorithm, Raft offers resilience against node failure but does not support Byzantine Fault Tolerance (BFT), leaving the system potentially exposed to malicious actors within the network—a relevant concern for administrative systems requiring high trust. Furthermore, the adaptation of fungible token standards for non-fungible assets may complicate unique asset tracking and reduce interoperability with external token standards.

Sharma et al. [44] presented a land registry platform that integrates the ERC-721 standard with IPFS to securely manage property documentation and ownership. Although their implementation use the standardized library for asset tokenization, the system relies on the public Ethereum network. This architectural choice exposes the registry to data privacy risks regarding transaction history and lacks the sovereign governance over consensus nodes that is essential for national land administration. Additionally, the scope of their smart contract logic is restricted to the permanent transfer of ownership between buyers and sellers.

Rashid et al. [39] proposed a multi-layered framework integrating Blockchain, Artificial Intelligence (AI), and Internet of Things (IoT) to modernize land management in Bangladesh. While the incorporation of AI for predictive analytics represents a noteworthy advancement, the blockchain component's evaluation is limited by its reliance on a simulated environment. The study utilized Ganache, a local development simulator, rather than a live production network. Consequently, the reported performance metrics may not fully reflect real-world factors such as network latency, peer propagation delays, or consensus overhead. Additionally, the economic model, based on public Ethereum gas pricing, which presents potential adoption challenges for developing countries.

Gupta et al. [17] highlighted the efficiency of Ethereum smart contracts for land registry, projecting a 99% reduction in manual recordkeeping and accelerated processing times. However, the study adopts a high-level architectural view, discussing Proof-of-Work (PoW) and Proof-of-Stake (PoS) as options without specifying a definitive consensus configuration. This generalization limits the ability to benchmark the throughput and security guarantees

necessary for a national-scale deployment. Additionally, the framework utilizes custom data structures instead of the ERC-4907 standard. While functional for internal registration, this design choice constrains interoperability with external digital asset ecosystems. Furthermore, the specific mechanisms for storing high-volume land documentation are not detailed.

Paavo et al. [36] employed a Design Science approach to develop an Ethereum-based land registry tailored for Namibia. Although the study outlines a comprehensive roadmap for user interface design and role-based access control, the technical evaluation was conducted within a local simulation environment using Ganache. As a result, the performance metrics reflect an idealized, single-node environment and do not account for the consensus latency inherent in a live distributed network. Additionally, the architecture stores land records directly on-chain without a decentralized off-chain storage solution (e.g., IPFS), which presents potential scalability challenges and increased storage costs for national-scale implementation. The reliance on custom data structures rather than the ERC-4907 standard further limits the potential for asset interoperability within the broader blockchain ecosystem.

Based on a review of the state-of-the-art, the following research gaps are identified:

1. **Reliance on Simulated Environments:** Numerous studies utilize theoretical models or local simulators, which may not fully account for real-world production factors such as network latency, peer propagation delays, and consensus overhead.
2. **Storage and Scalability Constraints:** A significant number of frameworks store voluminous land documentation directly on-chain without integrating decentralized off-chain storage. This design choice contributes to rapid ledger growth and potential scalability bottlenecks.
3. **Economic Viability Challenges:** Architectures utilizing public Ethereum networks are subject to fluctuating gas fees for transaction execution, presenting significant cost barriers for government-scale land administration in developing economies.
4. **Consensus Protocol Limitations:** Several permissioned implementations employ PoW, which entails high computational overhead, or CFT algorithms, which do not provide BFT protection against malicious internal actors.
5. **Non-Standardized Asset Representation:** Many studies rely on custom data structures or fungible token standards (ERC-20) rather than the industry-standard Non-Fungible Tokens (NFTs) such as ERC-721 and ERC-4907. This approach restricts asset interoperability and limits the potential for integration with broader digital asset ecosystems.
6. **Restricted Functional Scope (Absence of Leasing):** Existing literature focuses primarily on permanent ownership transfers (sales). There is limited research addressing temporary usage rights, such as automated renting or leasing logic, where ownership is delegated for a fixed period.

The BERBLOM framework was specifically designed to address the trade-offs identified in earlier research, particularly regarding security, speed, and cost. While previous studies often relied on expensive public networks or less secure protocols like Raft, the selection of Hyperledger Besu with QBFT creates a predictable and gas-free environment that is suitable for government administration. This approach also ensures the high throughput needed for national operations while maintaining data privacy and protection against malicious actors, which is a critical security feature missing in many previous implementations. Additionally, by connecting this private network with the IPFS, the storage limitations that earlier prototypes have are resolved. The use of IPFS ensures that heavy land documents are stored securely in a decentralized manner without overloading the blockchain ledger or incurring high storage costs.

Functionally, the adoption of the ERC-4907 standard distinguishes this framework from earlier studies that treated land records as static database entries. By defining land parcels as as standardized NFTs, the system enables automated leasing as a core feature, extend the simple buying and selling mechanism found in previous registries. This approach allows the blockchain to enforce time limits on property access, ensuring that tenant rights expire automatically without requiring manual intervention. Consequently, the framework better reflects the practical realities of land administration, where temporary tenancy is just as important as permanent ownership.

3 Proposed system

The system must satisfy several general requirements before any technology is chosen. It must ensure privacy of user information and documents, maintain data integrity so that records cannot be changed incorrectly, and guarantee the availability of records whenever they are needed. It should also provide auditability and traceability, allowing every action to be tracked over time. Each land parcel must have unique identifiers to avoid duplication, and the system must

offer strong resistance to unauthorized alteration of any stored information. In addition, it should support both temporary and permanent rights, such as renting and ownership, while ensuring accountability of officials by recording who performs each action. These requirements guide the overall design, and the following sections explain how the selected technologies meet them.

3.1 System architecture

Figure 3 outlines a blockchain-based architecture designed to streamline land administration and transaction processes. This system engages multiple participants, including the government body, landowners/sellers, buyers, renters, and other stakeholders. It employs a hybrid decentralized storage model, integrating the IPFS for off-chain storage with smart contracts for on-chain operations.

The underlying network infrastructure is built as a permissioned private blockchain using Hyperledger Besu. To ensure high throughput and zero-gas transaction costs for end-users, the system utilizes Quorum Byzantine Fault Tolerance (QBFT), a consensus protocol for private or consortium blockchains, designed to achieve agreement in a distributed system that may contain faulty or malicious nodes consensus mechanism. Unlike public networks where any node can participate in validation, this architecture restricts block validation to authorized government nodes. This design choice safeguards the integrity of the ledger while maintaining the speed required for national-scale land administration.

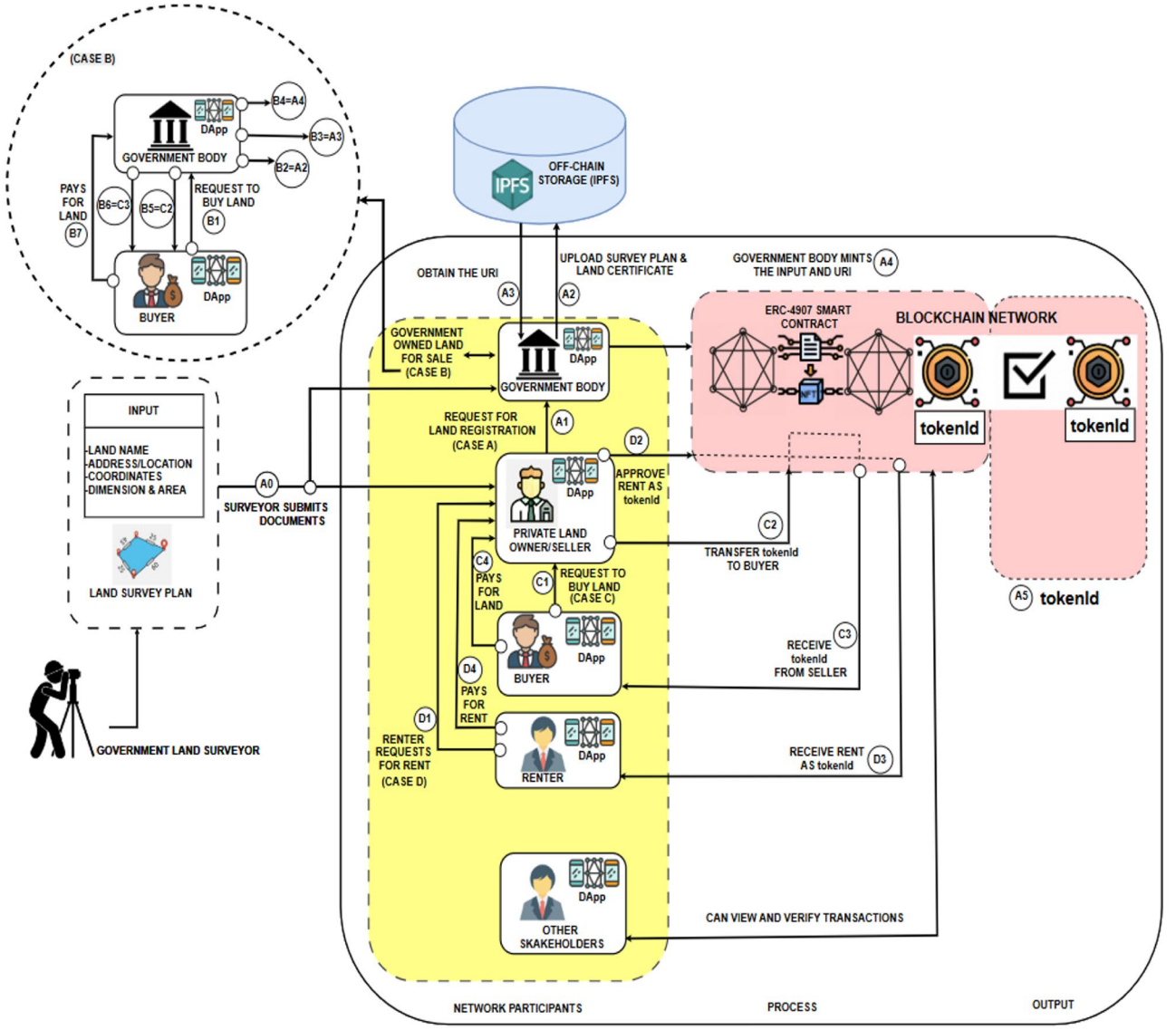


Fig. 3 Proposed system architecture

The use of the ERC-4907 standard with IPFS offers advantages over the custom methods used in earlier research. Industrially, this approach ensures the system works interoperably with standard wallets and external auditing tools, allowing for the prevention of double-selling fraud through the protocol's native indivisibility [25]. Academically, this design establishes a hybrid architecture that models land as distinct non-fungible entities rather than generic ledger entries. This structure enables efficient on-chain inheritance execution while using the TokenURI standard to immutably link lightweight tokens to heavy Geographic Information System (GIS) data on IPFS [46].

The system interfaces with participants through a flutter-based Decentralized Application (DApp) layer, as depicted by the user interfaces in Fig. 3. This layer serves as the bridge between the external stakeholders and the blockchain network. The architecture enforces strict role-based access control within the smart contracts. The 'Government Body' retains exclusive rights to mint new tokens and validate surveyor inputs (A0-A4), ensuring the legitimacy of the land data entering the system. Case B represents the governmental allocation and tokenized transfer of unregistered or unallocated land within the BERBLOM framework, where ownership is established through controlled blockchain minting. The process begins at B1, where the buyer submits a formal request for an unallocated land parcel through the DApp interface to the government authority. At B2, Government uploads the land's supporting documents, including the survey plan and land certificate, to IPFS to ensure decentralized, tamper-evident storage. The system then proceeds to B3, where a unique URI generated from IPFS is combined with land metadata to mint a tokenId representing the parcel within the government's blockchain account in B4. Next, in B5, the government transfers the tokenId to the buyer's blockchain address, enabling transparent and immutable ownership assignment. At B6, the buyer verifies the token transfer on-chain through the DApp interface, ensuring authenticity and correctness before financial commitment. Finally, B7 represents the payment completion and government acknowledgment, which finalizes the allocation and establishes the buyer as the legally recognized owner through ERC-4907 tokenized ownership recorded on the permissioned blockchain. This workflow demonstrates how BERBLOM integrates traditional governmental authority with automated blockchain processes to ensure transparency, traceability, and fraud-resistant land allocation.Conversely, 'Private Owners', 'Buyers', and 'Renters' are granted permissioned access to initiate transfers (C1-C4) and rental agreements (D1-D4), allowing for peer-to-peer interactions without centralized intermediaries for secondary transactions.

3.2 Rationale for ERC-4907 in land-registry tokenization

While the industry-standard ERC-721 token effectively models permanent land ownership through unique tokenized assets, it is insufficient for the dynamic requirements of a modern rental market. In a standard ERC-721 framework, leasing a property would typically require transferring the token itself to the tenant. This approach is fundamentally flawed for land administration because it conflates usage with ownership, forcing the landlord to relinquish custody of the title to the tenant and rely on trust or complex escrow mechanisms to ensure its return.

To address this limitation, the proposed system adopts the ERC-4907 standard, which extends ERC-721 by introducing a dual-role architecture that directly models the landlord–tenant relationship [4, 7, 53]. ERC-4907 distinguishes between the *Owner* role (the legal title-holder) and the *User* role (the active tenant), enabling a clear separation of ownership and usage rights. Importantly, the standard incorporates an automatic `expires` timestamp associated with the User role. This mechanism allows temporary usage rights to be granted and automatically revoked on-chain upon expiration of the lease period, eliminating the need for additional gas-intensive transactions to reclaim the asset. By preserving the proven security framework of ERC-721 while introducing native temporal access control, ERC-4907 provides a robust technical foundation for secure, automated, and trustless leasing systems.

3.3 Operational workflows

The land data registration process, as outlined in Algorithm 1, provides a structured approach to securely tokenize land ownership using blockchain technology. The process begins with the land owner submitting essential documents $land\_data$, $survey\_plan$, and $land\_certificate$ to a government office for verification. Upon acceptance, the government uploads the $survey\_plan$ and $land\_certificate$ to the IPFS, generating a unique URI that serves as a pointer to these files. This URI, combined with the $land\_data$, is then used to create a tokenId, a digital token representing ownership, which is registered on a blockchain and assigned to the land owner's account. The process concludes with the owner confirming the presence of the tokenId in their blockchain address, ensuring a tamper-proof and transparent record of ownership. This algorithm leverages blockchain's immutability and IPFS's decentralized storage to enhance the security and accessibility of land registration systems.

Input : $land\_data$, $survey\_plan$, $land\_certificate$
Output: $tokenId$ representing land ownership
Owner visits government office with $land\_data$, $survey\_plan$ and $land\_certificate$;
Government accepts the documents;
Government uploads the $survey\_plan$ and $land\_certificate$ to IPFS;
Government generates URI from IPFS;
Government converts $land\_data$, and URI to $tokenId$ into current land owner's blockchain account;
Land Owner confirms existence of the $tokenId$ in his blockchain $address$;
Output: land ownership represented as $tokenId$;

Fig. e (Case A) Land data registration process.

Algorithm 2 outlines the process of land allocation and token transfer (Case B), where unallocated land is securely assigned to a buyer using blockchain technology. The process begins with a buyer requesting an unallocated plot of land from the government, which accepts the request and proceeds to upload the associated $survey\_plan$ and $land\_certificate$ to the IPFS. IPFS generates a unique URI, which the government uses, along with the $Unallocated\_land\_data$, to mint a *tokenId* representing the land in the government's blockchain account. This *tokenId* is then transferred to the buyer's blockchain account, allowing the buyer to view and verify the transfer. Following verification, the buyer completes the payment for the land, and the government acknowledges receipt, finalizing the ownership transfer. The output is the land ownership securely transferred to the buyer as a *tokenId*.

Input : $Unallocated\_land\_data$, $survey\_plan$ and $land\_certificate$
Output: Land ownership transferred to buyer
Buyer requests unallocated plot of land from government;
Government accepts the request;
Government uploads $survey\_plan$ and $land\_certificate$ to IPFS;
Government generates URI from IPFS;
Government mints $Unallocated\_land\_data$ and URI into government's blockchain account as $tokenId$;
Government transfers $tokenId$ to buyer's blockchain account;
Buyer views and verifies the transfer of $tokenId$;
Buyer proceeds to pay for the land;
Government acknowledges receipt of payment;
Output: Land ownership transferred to buyer as $tokenId$;

Fig. f (Case B) Lands allocation and *tokenid* transfer process.

Algorithm 3 details the process of buying and selling already registered land (Case C), focusing on the transfer of tokenized land ownership between a buyer and a seller using blockchain technology. The process begins with the buyer requesting to purchase a plot of land from the seller, and both parties reaching an agreement on the terms of the sale. The seller then transfers the *tokenId*, which represents the land ownership, from their blockchain address to the $buyer's\_address$. The buyer confirms the successful transfer of the *tokenId* into their blockchain account, ensuring the ownership has been updated. Following this confirmation, the buyer pays the agreed-upon amount to the seller, finalizing the transaction. The output is a confirmation of the land ownership transfer to the buyer.

Input : $buyer's\_address$, $tokenId$
Output: Confirmation of land ownership transferred to buyer
Buyer request to purchase plot of land from seller;
Agreement reached between the two parties;
Seller transfers land $tokenId$ from seller's address to $buyer's\_address$;
Buyer confirms ownership of the $tokenId$;
Buyer pays the seller amount agreed upon;
Output: Confirmation of land ownership transferred to buyer;

Fig. g (Case C) Buying and selling of already registered land.

Algorithm 4 describes the process of renting land through a temporary assignment of usage rights (Case D). The process begins with the land owner and renter agreeing on the rental terms, including a fixed expiration time *rentExpiresTime* expressed as a Unix timestamp. The owner then assigns the renter as the temporary user of the *tokenId* until *rentExpiresTime*, and the renter confirms receipt of these usage rights before completing the agreed payment. During execution, the system compares the current block time $current\_time$ with *rentExpiresTime*: if $current\_time < rentExpiresTime$, the rental remains active and the renter retains usage rights. Once *rentExpiresTime* is reached, the renter's usage rights automatically expire and full control remains with the owner, completing the rental cycle without requiring a manual token return.

Input : $renter\_address$, $rentExpiresTime$, $tokenId$
Output: Expiration of renter usage rights after rent period ends
Both land owner and renter meets and agree on renting terms;
Property owner assigns renting rights of $tokenId$ to $renter\_address$ with $rentExpiresTime$;
Renter confirms receipt of usage rights for $tokenId$ with $rentExpiresTime$;
Renter pays the amount agreed for renting;
if $current\_time < rentExpiresTime$ **then**
  Renting period is not yet over;
  Output: Usage rights of $tokenId$ remain assigned to the renter;
else
  Renting period has expired;
  Output: Renter usage rights for $tokenId$ automatically expire;
end

Fig. h (Case D) Renting of land and automatic expiration of usage rights

3.4 Data models and smart contract specification

This section defines the system's data models and formalizes the smart contract logic underlying its execution. It

outlines the schema for off-chain data representation and the algorithmic procedures enforced on-chain.

3.4.1 Off-chain data models

To maintain ledger efficiency, the system utilizes IPFS for decentralized content storage. Because blockchain storage is prohibitively expensive for heavy files, BERBLOM offloads high-resolution survey plans and legal certificates to IPFS. To guarantee data availability and prevent the eviction of critical records by garbage collection processes, the system enforces a mandatory pinning protocol on the IPFS nodes.

Input : $file$ (Binary), $data$ (land metadata)
Output : $metaURI$ (Metadata URI)
Require: Authenticated IPFS Gateway

```
// Upload & Pin Certificate
docHash ← IPFS.add(file, pin=true);
docURI ← "ipfs://" + docHash;

// Construct Metadata
meta ← {
  "name" : data.name,
  "description" : "Land certificate for" ∥ D.name,
  "uri" : docURI,
  "attributes" : {
    "City" : data.city,
    "Country" : data.country,
    "Coordinates" : {data.y1, data.y2, data.x1, data.x2},
    "Dimensions" : data.dimensions}
};
// Upload & Pin Metadata
metaHash ← IPFS.add(meta, pin=true);
metaURI ← "ipfs://" + metaHash;
return metaURI;
```

Fig. i Store land certificate and metadata to IPFS.

Algorithm 5 defines the off-chain preservation workflow as a two-stage IPFS storage process. The binary land certificate and associated metadata are submitted to an authenticated private IPFS gateway, where the document is uploaded with pinning enabled to ensure persistent availability across participating nodes. This operation generates a unique document URI (*docURI*), which serves as the reference to the off-chain artifact. Furthermore, it is embedded directly within the metadata object to establish a verifiable linkage between structured metadata and the underlying certificate.

The algorithm then constructs a standardized JSON metadata structure encoding descriptive and spatial attributes of the land asset. The identifier is stored as *data.name*, and the description concatenates a static label with this identifier, while the document reference is preserved through *docURI*. Jurisdictional attributes include city and country, spatial boundaries are defined as coordinate tuples (*data.y*1 , *data.y*2, *data.x*1, *data.x*2), and physical characteristics are recorded. The metadata object is subsequently uploaded and pinned to IPFS, producing a second unique URI (*metaURI*) that represents the off-chain record and is stored on-chain as the token's persistent reference.

Conversely, Algorithm 6 defines the process for reconstructing off-chain records from an on-chain reference. Given a valid *tokenId*, the framework queries the *landData* structure to obtain the stored *metaURI*, retrieves the corresponding JSON metadata from IPFS, and reconstructs the asset's descriptive and structural context. The algorithm then extracts the document reference *docURI* from *meta.uri* and performs a second IPFS retrieval to obtain the associated binary certificate file. The final output consists of the metadata object and the certificate file, enabling the on-chain token identifier to be resolved into verifiable off-chain evidence.

Input : *tokenId* (tokenized land ID)
Output: $meta$ (metadata struct), $file$ (certificate file)
Require: Accessible IPFS gateway
$owner[tokenId] \neq address(0)$ // token ID must exists

```
metaURI ← landData[tokenId].uri; // Get URI from landData struct
meta ← IPFS.get(metaURI); // Retrieve metadata from IPFS
docURI ← meta.uri; // Get file URI from metadata
file ← IPFS.get(docURI); // Retrieve file from IPFS
return ⟨meta, file⟩;
```

Fig. j Fetch land certificate and metadata from IPFS.

3.4.2 On-chain contract logic

The on-chain component is governed by the smart contract, which inherits from the ERC-4907 standards. The designated government authority operates and administers the smart contract functions as the deploying entity responsible for enforcing access control, validating land registration transactions, and maintaining the integrity of ownership and usage rights. Through privileged administrative roles, the authority ensures that only authorized updates are committed on-chain, providing regulatory oversight while preserving transparent and auditable state transitions across the system.

Each smart contract function emits an event to indicate that an operation has been completed successfully, such as a new land registration, ownership updates, or changes in usage rights. The event logs are stored with the transaction and provide a clear record of contract activity without changing the contract state. As a result, it allows external

systems, such as DApps, to track contract actions and stay synchronized with on-chain updates.

Algorithm 7 specifies the procedure for registering a new land asset as a tokenized record. The process is restricted to the designated government authority, which invokes the function to create an official land entry. Upon execution, the algorithm assigns the next available identifier ($tokenId_{next}$) as the unique token ID and mints a new NFT to the recipient address using the *safeMint* interface, thereby establishing verifiable ownership on-chain. Following token creation, the algorithm records the associated land information by storing the recipient address, structured metadata (*data*), and the corresponding IPFS metadata URI within the *landData* mapping. The token counter is then incremented to maintain sequential registration, and a *LandRegistered* event is emitted to provide an auditable transaction record. The function concludes by returning the newly generated *tokenId*, enabling external systems to reference the registered land asset.

Input : to (recipient address), $data$ (land metadata), URI (IPFS URI)
Output : $tokenId$ (newly minted token ID)
Require: $caller = GovernmentBody$

$tokenId \leftarrow tokenId_{next}$; // Get current ID
$ERC4907.safeMint(to, tokenId)$; // Mint NFT via ERC-4907 Interface
$landData[id] \leftarrow \{to, data, URI\}$; // Store inputs to $landData$ struct
$tokenId_{next} \leftarrow tokenId_{next} + 1$; // Prepare ID for next registration
emit $LandRegistered(to, data, URI)$;
return $tokenId$;

Fig. k Land registration.

Input : $owner$ (land owner address), to (recipient address), $tokenId$ (token ID)
Output : Event $Transfer(owner, to, tokenId)$
Require: $caller = ERC4907.ownerOf(tokenId)$
// Caller is land owner
Lease is inactive

$ERC4907.safeTransferFrom(caller, to, tokenId)$;
// Transfer ownership
emit $Transfer(caller, to, tokenId)$;

Fig. l Land transfer.

Algorithm 8 describes the transferring land ownership from one user to another. The function can only be executed by the current owner of the token, which ensures that unauthorized parties cannot initiate a transfer. Before the transfer occurs, the system checks that no active lease is associated with the land, preventing ownership changes while usage rights are still assigned. Once these conditions are met, the contract transfers the token to the recipient using the ERC-4907 interface and emits a Transfer event to record the ownership change on-chain.

Input : $renter$ (renter address), $tokenId$ (token ID), t_{exp} (expiration time)
Output : Event $UpdateUser(id, renter, t_{exp})$
Require: $caller = ERC4907.ownerOf(tokenId)$
// caller is land owner
Lease is inactive
$ERC4907.setUser(tokenId).renter \leftarrow \{renter\}$;
// Assign new renter
$ERC4907.setUser(tokenId).expires \leftarrow \{t_{exp}\}$; // Assign expiration time
emit $UpdateUser(tokenId, renter, t_{exp})$;

Fig. m Land renting.

Algorithm 9 defines the process for assigning temporary usage rights of a land token to a renter. The function can only be executed by the current owner of the token, which ensures that only authorized parties can grant rental access. Before assigning a renter, the system checks that no active lease already exists for the land. Once validated, the contract records the renter address and the expiration time using the ERC-4907 interface and emits an *UpdateUser* event to log the rental assignment and duration on-chain.

Input : $tokenId$ (token ID)
Output: $renter$ (renter address), t_{exp} (expiration time)
$t_{now} \leftarrow t_{block}$; // Retrieve Current Block Time
$renterInfo \leftarrow ERC4907.userOf(tokenId)$; // Fetch current rent info

// Check Lease Validity
if $renterInfo.expires \geq t_{now}$ **then**
  return $\langle renterInfo.renter, renterInfo.expires \rangle$;
  // Lease active
else
  return $\langle address(0), 0 \rangle$; // Lease inactive
end

Fig. n Lease information query.

Algorithm 10 describes the operation for checking the current rental status of a land token. Given a *tokenId*, the contract first retrieves the current block time and queries the ERC-4907 interface to obtain the stored renter information and expiration time. The system then compares the expiration value with the current time to determine whether the lease is still active. If the lease is valid, the function returns the renter address and expiration time. Otherwise, it returns address zero and zero value to indicate that no active rental exists.

4 Performance metrics, experimental setup, and results

4.1 Performance metrics

To ensure a rigorous and interpretable evaluation of the proposed blockchain framework, it is essential to introduce performance metrics that capture both the temporal efficiency

and operational capacity of the system. Blockchain platforms inherently involve sequential processes such as transaction submission, validation, block formation, and finality that collectively shape their responsiveness and scalability. Consequently, two fundamental metrics, latency and throughput, are employed to quantify these characteristics. Latency reflects the end-to-end delay experienced by users during transaction confirmation, while throughput measures the system's ability to process and finalize transactions over time. Together, these metrics provide a comprehensive performance assessment and enable meaningful comparisons across blockchain designs, consensus mechanisms, and network configurations.

4.1.1 Latency

Latency in a blockchain system is the time delay between submitting a transaction and receiving its confirmation [8, 13]. Mathematically, it can be expressed as the following Eq. 1.

$$Latency = TCT - TST, \tag{1}$$

Transaction Submission Time (TST) marks the moment when the client application sends the transaction request to the blockchain network. Transaction Confirmation Time (TCT) occurs when the transaction is successfully included in a block, validated by the consensus mechanism, and confirmed to the client. This measure accounts for all processing delays, network transmission times, and consensus verification periods. For example, if a transaction has a TST of 10.5 seconds and a TCT of 12.7 seconds (from an arbitrary starting point), the latency is 12.7 - 10.5 = 2.2 seconds.

4.1.2 Throughput

Throughput represents the speed at which a blockchain network processes and confirms transactions within a specific timeframe [8, 13]. Typically measured in transactions per second (TPS), it quantifies the network's capacity to handle transactional demand. The simplest way to express throughput is through the following formula:

$$Throughput = CT/t \tag{2}$$

where CT denotes the number of confirmed transactions—those successfully added to blocks and validated by the network's consensus mechanism. t is the time period (in seconds) during which these transactions are confirmed. Alternatively, throughput can also be expressed in terms of proposed transactions (PT) and confirmed transactions (CT), where:

$$Throughput = \frac{CT}{PT} \times \frac{1}{TCT} \tag{3}$$

CT is the number of confirmed transactions, as defined above. PT represents the total number of proposed transactions submitted to the network, including those that may fail or be rejected. Lastly, TCT is the average time (in seconds) required to confirm a single transaction.

It is important to note that these metrics can vary based on multiple factors including network load, block size, consensus algorithm parameters, and network topology. Therefore, we report average values along with standard deviations across multiple experimental runs.

4.2 Experimental setup

The experimental computer is made up of ASUS TUF Gaming A17FA707RR_FA707RR system model with AMD Ryzen 7 6800H processor. It has 16 central processing unit cores and operates at 3.2GHz processing speed. The random access memory (RAM) is rated 32,768MB. The computer hosts a Windows 11 Pro 64-bit (10.0, Build 22631) operating System. The smart contract was written in Solidity programming language using Remix integrated development environment. The user interface for the DApp was developed using the Flutter framework to ensure cross-platform compatibility and responsive performance. We deployed the NFT smart contract using Metamask hosting Hyperledger Besu free gas network. Using Hyperledger caliper, we were able to measure the performance of the network based on latency and througput. The Hyperledger Besu network was created based on the PoA consensus algorithm, specifically utilizing Clique, a variant tailored for Ethereum-based networks [13]. Furthermore, the network was configured with three peers. The network operates with a chain identity (*chainId*) of 1337, a block period of 1 second, and the system's default epoch length of 30,000.

4.3 Network performance

The performance evaluation of the Clique Proof-of-Authority (PoA) blockchain network, as detailed in Table 2, was conducted using Hyperledger Caliper to assess its efficiency across varying transaction loads, ranging from 100 to 1,000 requests per second (Req/s). The table presents comprehensive metrics for three distinct smart contract function categories, *open*, *query*, and *transfer*, including success and failure rates, average latency (in seconds), and throughput (in transactions per second, TPS), highlighting the network's operational behavior under increasing demand. According to Pradhan et al. [37], smart contract functions are categorized based on *open*, *query*, and *transfer*. Notably,

Table 2 Clique POA blockchain network performance

Send Rate	Success(%)	Failure(%)	Avg Latency (s)			Throughput (TPS)		
(Req/s)			Open	Query	Transfer	Open	Query	Transfer
100	100	0	0.564	0.001	0.560	98.500	100.100	93.400
200	100	0	0.619	0.001	0.728	191.300	191.500	181.800
400	100	0	1.730	0.001	0.962	194.300	187.200	287.500
600	100	0	1.729	0.001	1.501	194.700	176.200	247.600
800	100	0	1.672	0.001	1.611	194.400	174.000	252.100
1000	100	0	1.760	0.001	1.270	200.100	185.100	296.800

Table 3 QFBT blockchain network performance

Send Rate	Success(%)	Failure(%)	Avg Latency (s)			Throughput (TPS)		
(Req/s)			Open	Query	Transfer	Open	Query	Transfer
100	100	0	0.720	0.001	0.650	90.800	100.000	93.400
200	100	0	2.910	0.001	1.630	98.700	188.900	134.300
400	100	0	4.130	0.001	2.850	99.700	205.500	127.300
600	100	0	4.260	0.001	3.140	101.800	201.100	129.400
800	100	0	4.300	0.001	3.330	100.500	204.100	128.500
1000	100	0	4.310	0.001	3.410	98.700	193.100	132.600

the *query* function exhibited satisfactory performance with a consistently low latency of 0.001 seconds across all send rates, underscoring its optimization for rapid data retrieval. In contrast, the *open* and *transfer* functions demonstrated a gradual increase in latency as request rates rose, with maximum values of 1.760 and 1.611 seconds, respectively, at 1,000 Req/s, while throughput peaked at 296.800 TPS for the *transfer* function, reflecting the network's capacity to handle high transaction volumes effectively.

These results confirm the inherent trade-off between latency and throughput in Byzantine fault-tolerant consensus, where at 1,000 Req/s the QBFT network incurs 2.45 and 2.68 times higher latency for the *open* and *transfer* functions, respectively, compared to the Clique baseline (Table 3, while delivering approximately half the peak write throughput (98.700 vs. 200.100 TPS for *open*; 132.600 vs. 296.800 TPS for *transfer*). Nevertheless, QBFT demonstrates a more stable throughput profile, with *open* TPS varying by only 11 TPS (90.800 to 101.800) across all load levels, compared to over 100 TPS variation in the Clique configuration (98.500 to 200.100). Both configurations achieved a 100% success rate throughout the evaluation. These characteristics make QBFT a suitable choice for a permissioned land administration system, where consistency and security guarantees take precedence over raw throughput.

Figure 4 shows blockchain network performance based on average latency for different consensus algorithms. The graph compares the average latency (in percentage) against the send rate (in requests per second) for various consensus algorithms: Quorum Byzantine Fault Tolerance (QBFT), Clique, and Byzantine Fault Tolerance 2.0 (BFT2.0). The results demonstrate how each algorithm performs as the send rate increases, showing distinct latency behaviors for each consensus method under the OPEN configuration.

Figure 5 shows the blockchain network performance based on average latency for different consensus algorithms for smart contract TRANSFER functions. The graph compares the average latency (in seconds) against the send rate (in requests per second) for various consensus algorithms: QBFT, Clique, and BFT2.0. Based on the graph, Clique demonstrates the best performance, maintaining the lowest latency across different send rates. This suggests that Clique is particularly well-suited for environments requiring consistent and low-latency transaction processing.

While Fig. 5 shows that the Clique consensus algorithm achieves the lowest latency for TRANSFER transactions in our initial evaluation, these measurements were obtained under benign conditions and do not capture behaviour in the presence of faulty or adversarial validators. However, for a government grade land administration system, we require a consensus mechanism with strong byzantine fault tolerance, immediate finality, and stable performance as the network grows. To assess alternative options, we conducted an additional review of recent studies on Hyperledger Besu proof of authority protocols in permissioned environments. Tkachuk et al. [47] analyse Clique, IBFT 2.0, and QBFT in a privacy enabled renewable energy marketplace and report that QBFT consistently offers the best combination of throughput, latency, and scalability, sustaining about 190–200 transactions per second with up to twelve validators while avoiding the forks and non finality issues observed with Clique. Delladetsimas et al. [10] provide a broader comparison of byzantine fault tolerant protocols and show, both through simulations and synthesis of prior Besu based

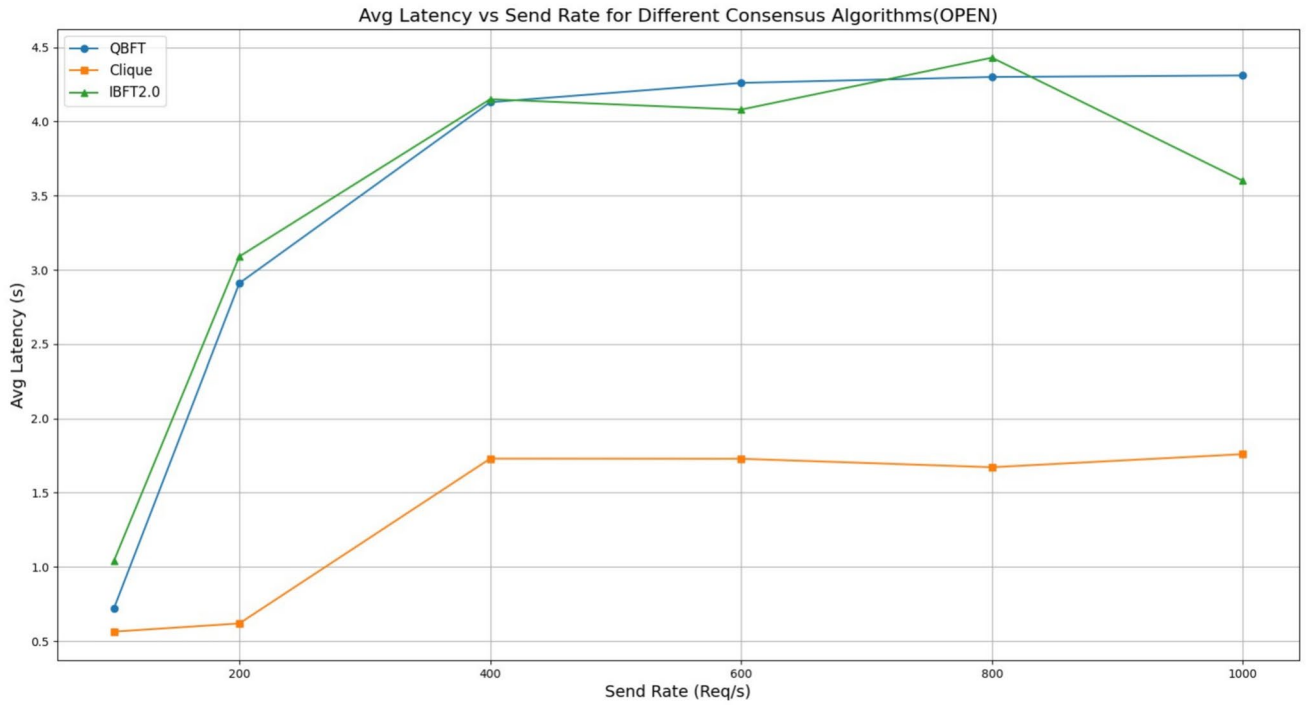


Fig. 4 Blockchain network performance based on average latency for (OPEN)

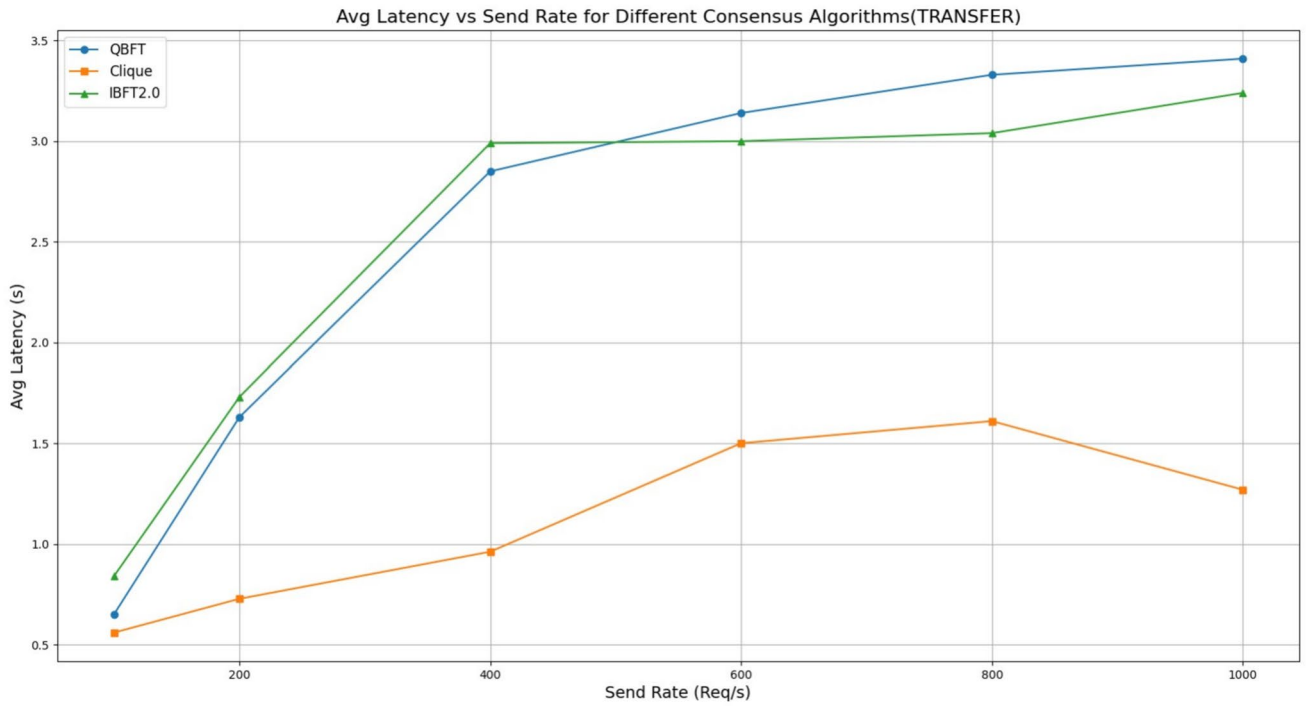


Fig. 5 Blockchain network performance based on average latency for (TRANSFER)

evaluations, that QBFT scales more gracefully than IBFT 2.0 and Clique and delivers more uniform leader rotation and consensus fairness in permissioned networks. Ferone and Verrilli [14] benchmark Clique, IBFT 2.0, and QBFT in a digital twin security backbone built on Hyperledger Besu and conclude that, although Clique can show slightly lower latency, QBFT maintains higher and more stable throughput under increasing transaction rates and is therefore better suited for fault tolerant industrial deployments. Based on these findings and our own requirements for stronger

security guarantees, we reconfigured our prototype from a Clique based network to a QBFT based network and carried out a second set of experiments to evaluate the latency, throughput, and scalability of the QBFT configuration, which are reported in the following subsection.

4.4 QBFT performance and scalability

The implementation of the QBFT (Quorum-Based Fault Tolerance) consensus mechanism in this solution is carefully designed to provide high throughput, low latency, and strong fault tolerance, which are crucial for applications such as industrial IoT and blockchain-based systems. Configured with Besu version 25.11.0, the QBFT protocol is set with a block period of 1 second, an epoch length of 30, 000, and a request timeout of 4 seconds, ensuring rapid block finality and quick transaction processing. Besu's "FULL" sync mode, "FOREST" data storage format, and other specific settings (such as a minimum gas price of 0, a sequenced transaction pool, and 300 maximum RPC HTTP active connections) ensure that the network operates efficiently. Additionally, Besu's handling of transactions with specific limits, such as tx-pool limits by account percentage and non-local priority in the transaction pool, further optimizes performance. The QBFT protocol, combined with these Besu settings, helps protect the system against Byzantine faults, maintaining consensus even when faced with faulty or malicious nodes. In terms of benchmarking, the system is optimized with 3 to 10 workers and handles 2000 transactions, supported by send rates of 150, 300, and 500 transactions per second (Tx/s). The system's scalability is enhanced by supporting node configurations ranging from 4 to 32 nodes, and the integration of a load balancer (haproxy) ensures efficient workload distribution. This setup guarantees high resilience, fault tolerance, and performance, making it ideal for blockchain solutions where real-time processing, system scalability, and security are critical, such as in IoT and blockchain-based applications.

Table 4 Simulation parameters

Parameter	Value
Besu Version	25.11.0
Sync Mode	FULL
Data Storage Format	FOREST
Minimum Peers for Sync	1
Remote Connections Limit	Disabled (remote-connections-limit-enabled=false)
Transaction Pool	SEQUENCED
Transaction Pool Local Priority	Disabled (tx-pool-no-local-priority=true)
Transaction Pool Limit (per account)	15% (tx-pool-limit-by-account-percentage=0.15)
RPC HTTP Max Active Connections	300
Minimum Gas Price	0
Bonsai Trie Logs	Disabled (bonsai-limit-trie-logs-enabled=false)
QBFT Block Period	1 second
QBFT Epoch Length	30, 000
QBFT Request Timeout	4 seconds
Benchmark Workers	3 to 10 workers
Transactions (Tx)	2000
Send Rates (Tx/s)	150, 300, 500
Node Configurations	4, 8, 12, 16, 20, 24, 28, 32 nodes
Load Balancer	Enabled (haproxy)

Table 4 summarizes the key simulation parameters, including the Besu configuration, QBFT settings, and benchmarking details used to optimize the network's performance. Additionally, Fig. 16 illustrates the QBFT protocol in action, showing the logs of the QBFT nodes as they process empty blocks, reflecting the real-time operations of the consensus mechanism during network execution.

Table 5 presents the system latency and throughput results for varying node counts in a blockchain network, highlighting the performance of different transaction types (Open, Query, Transfer) across different node configurations (4, 8, 12, 16, 20, 24, 28, 32 nodes). The table provides key performance metrics, including average latency (in seconds) at different send rates (150, 300, and 500 transactions per second), and throughput (in transactions per second) for each configuration. As the number of nodes increases, the system experiences higher latencies and decreased throughput across all transaction types, particularly for the Open and Transfer operations, which show a more significant rise in latency. Query operations maintain relatively stable latencies, with minimal impact from increasing node counts. For example, with 32 nodes, the latency for the Open transaction reaches up to 8.30 seconds at 500 transactions per second, and the throughput drops to 108 Tx/s. These results suggest that as the network scales with more nodes, the system faces greater challenges in maintaining high performance, indicating potential bottlenecks that may require optimization strategies in large-scale blockchain networks.

Figure 7 provides a clear depiction of the latency performance of the QBFT consensus mechanism across varying node counts and transaction types. As the number of nodes increases, both Open and Transfer transactions experience a significant rise in average latency, particularly at higher send rates (300 and 500 Tx/s), suggesting that these transactions are more sensitive to network size and load. Query transactions, on the other hand, exhibit relatively stable latency up to 20 nodes, but after that, a sharp increase is observed, especially at 500 Tx/s, indicating that query operations are less affected by network scaling until a threshold is reached. These trends align with the findings in Table 5, where latency and throughput performance degrade as the number of nodes increases. The results from Fig. 7 emphasize

```
2025-11-21 09:13:58.031+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)

2025-11-21 09:13:58.032+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)

2025-11-21 09:13:58.032+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)

2025-11-21 09:13:58.034+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)

2025-11-21 09:13:58.035+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)

2025-11-21 09:13:58.036+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)

2025-11-21 09:13:58.037+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)

2025-11-21 09:13:58.038+0000 | BftProcessorExecutor-QBFT-0 | INFO  | QbftBesuControllerBuilder |
Imported empty block #5,809 / 0 tx / 0 pending / 0 (0.0%) gas /
(0x64bf4a49c43981e2bde063c5f6131fc93616cef83880b169fc1fb78993c1bc3e)
```

Fig. 6 QBFT network running

the scalability challenges of the QBFT protocol in larger blockchain networks. Specifically, transaction types that involve more complex operations (Open and Transfer) are notably impacted by network growth, while simpler operations like Query transactions show more resilience up to a certain scale. These insights point to the need for optimization strategies, such as load balancing or hybrid consensus mechanisms, to mitigate performance bottlenecks and enhance system scalability in real-time applications.

Figure 8 presents the throughput results of the QBFT consensus mechanism under different node counts and transaction types. The results show that throughput for Open and Transfer transactions drops quickly as the number of nodes increases, especially at higher transaction loads of 300 and 500 Tx/s. This pattern suggests that these transaction types are sensitive to network scaling, as more nodes introduce higher coordination and communication overhead during consensus. As a result, system performance declines more noticeably when processing these more complex operations in larger networks.

In contrast, Query transactions maintain relatively stable throughput across different node sizes, with only small changes as the network grows. This indicates that Query operations place less strain on consensus and are less affected by transaction load compared to Open and Transfer transactions. The consistent performance of Query operations highlights their efficiency and lower execution complexity. Overall, the findings point to scalability limits within QBFT for certain transaction types, suggesting that improvements such as better load distribution, transaction prioritization, or more efficient consensus mechanisms may be required to sustain performance at larger scales.

Table 5 System latency and throughput results for varying node counts

Nodes	Name	Txs	Succ(%)	Avg Lat @ 150(s)	Avg Lat @ 300(s)	Avg Lat @ 500(s)	Thru @150	Thru @300	Thru @500
4	Open	2000	100	0.62	0.85	0.76	147	274	459
	Query	2000	100	0.001	0.001	0.001	150	300	500
	Transfer	2000	100	0.64	0.81	1.30	146	272	397
8	Open	2000	100	0.90	1.40	2.10	145	250	340
	Query	2000	100	0.001	0.001	0.002	150	300	500
	Transfer	2000	100	0.95	1.90	3.10	144	230	310
12	Open	2000	100	1.30	2.90	4.40	142	200	250
	Query	2000	100	0.002	0.002	0.002	150	300	500
	Transfer	2000	100	1.40	3.20	4.90	140	185	220
16	Open	2000	100	1.80	5.40	6.90	140	160	200
	Query	2000	100	0.002	0.002	0.003	150	300	500
	Transfer	2000	100	2.10	6.10	7.40	138	150	190
20	Open	2000	100	2.60	12.50	14.80	130	110	150
	Query	2000	100	0.003	0.002	0.003	150	300	490
	Transfer	2000	100	3.50	11.80	13.90	120	105	140
24	Open	2000	100	3.50	18.90	22.40	125	80	100
	Query	2000	100	0.003	0.003	0.003	150	300	480
	Transfer	2000	100	4.60	16.70	20.80	114	90	95
28	Open	2000	100	5.00	26.30	31.00	120	65	80
	Query	2000	100	0.003	0.003	0.003	150	300	450
	Transfer	2000	100	6.20	24.90	28.20	110	70	75
32	Open	2000	100	6.80	35.40	42.80	115	55	70
	Query	2000	100	0.003	0.003	0.003	150	300	430
	Transfer	2000	100	8.30	33.10	39.50	108	60	65

4.5 Smart contract functionality

This subsection presents the results of smart contract testing which shows the corresponding outputs results of functions and events. Events are inbuilt function from the smart contract which helps maintain data provenance and information traceability and can be used as transactions logs. Therefore, all actions performed by the respective participants can be easily tracked. For testing functionality scenarios, the following participants interact with the smart contract: landowner/seller, renter, government, buyer and other stakeholders.

Several functions are similar in execution and results, hence only results from testing crucial functions are presented in this section. Inputs and outputs logs resulting from the network participants' interactions with the functions of the smart contract are demonstrated as follows.

Initially, the process of registering new land via a smart contract involves a landowner submitting detailed information about their property. This includes the landowner's blockchain public address, the specific location and coordinates of the land, its dimensions and overall area, and a scanned copy of the official land certificate. The scanned certificate, illustrated by Fig. 9, is uploaded to IPFS, generating a URI that is then registered in the smart contract by government staff, as reflected in Fig. 10.

Land registration is finalized when the smart contract executes the *safeMint*() function, creating a new tokenized representation of the parcel as illustrated in Fig. 11. This operation assigns a unique *tokenId*, shown as *tokenId* 4, thereby formally recording the land asset on-chain. The figure captures the corresponding registration event emitted by the authorized registry account, which in this prototype represents the operational address of the government land registry authority. While the current implementation reflects a simplified approval model, a production deployment would incorporate role-based verification and multi-signature authorization to ensure that *safeMint*() is executed only after confirmation from all designated government officials, thereby enforcing structured approval workflows and strengthening institutional oversight. Unless otherwise specified, all subsequent log-output figures follow a consistent structure in which the contract address (*from*), event signature hash (*topic*), event name (*event*), and input parameters (*args*) are recorded by the emitted event.

Following registration, the landowner can verify the stored information by calling the *getInfo* function with the corresponding *tokenId* as the input parameter. This query retrieves the on-chain metadata reference and confirms that the ownership and land details have been correctly recorded. Figure 12 presents the resulting log output for land information associated with *tokenId* 4, showing the returned values and demonstrating how users can independently confirm the

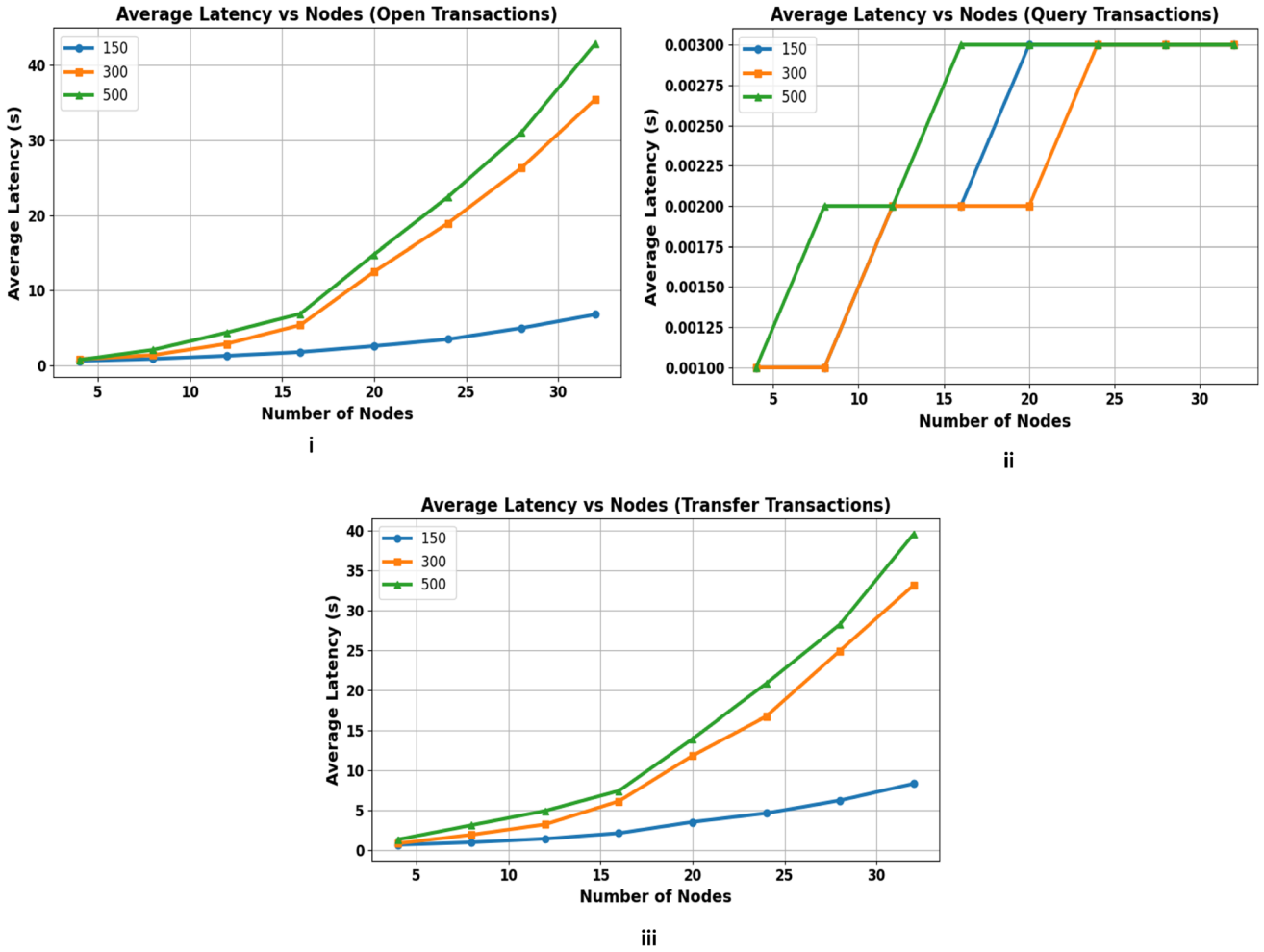


Fig. 7 QBFT Latency Performance for send rates of 150, 300 and 500

state of a registered asset. This verification ensures that the tokenized land entry remains accessible and verifiable after registration.

Once a land parcel is listed on the marketplace, a prospective buyer may proceed with acquisition after selecting the desired property. The transaction is executed by invoking the *transfer*() function within the smart contract. This operation permanently reassigns the rights from the current landowner to the buyer. The successful conclusion of this procedure can be verified by the new owner through the *getInfo* function by querying the unique *tokenId*. In this scenario, the landowner sells the parcel associated with *tokenId* 5. The execution log, visualized in Fig. 13, serves as confirmation of this transfer.

Alternatively, for temporary tenure, a tenant may secure a lease for an available property. The lease is initialized using the *setUser*() function in the smart contract. Unlike a sale, this function does not transfer the permanent title. Instead, it assigns the `User` role to the tenant for a specified duration, while the original landlord retains underlying ownership. The ledger records this usage assignment, ensuring the tenant's temporary rights are transparent and verifiable. The tenant can confirm the validity of their lease by calling the *getInfo* function with the relevant *tokenId*. In this case, the occupant rents the property with *tokenId* 4 until the *rentExpiresTime* of 1802277768. The transaction log is depicted in Fig. 14. Furthermore, Fig. 15 displays the updated rental status, demonstrating an efficient digital management of tenancy. Lastly, while the lease is active, the smart contract prevents the tenant from subleasing and the landowner from selling the asset.

When the rental period ends, the smart contract automatically resets the land status to available. This is reflected by the renter address reverting to the zero address (0*x*000000 0000000000000000000000000000000000) and the expiration timestamp being set to 0. The state of the respective tokenized land after the rental period concludes is identical to that shown in Fig. 12, thereby restoring the landowner's

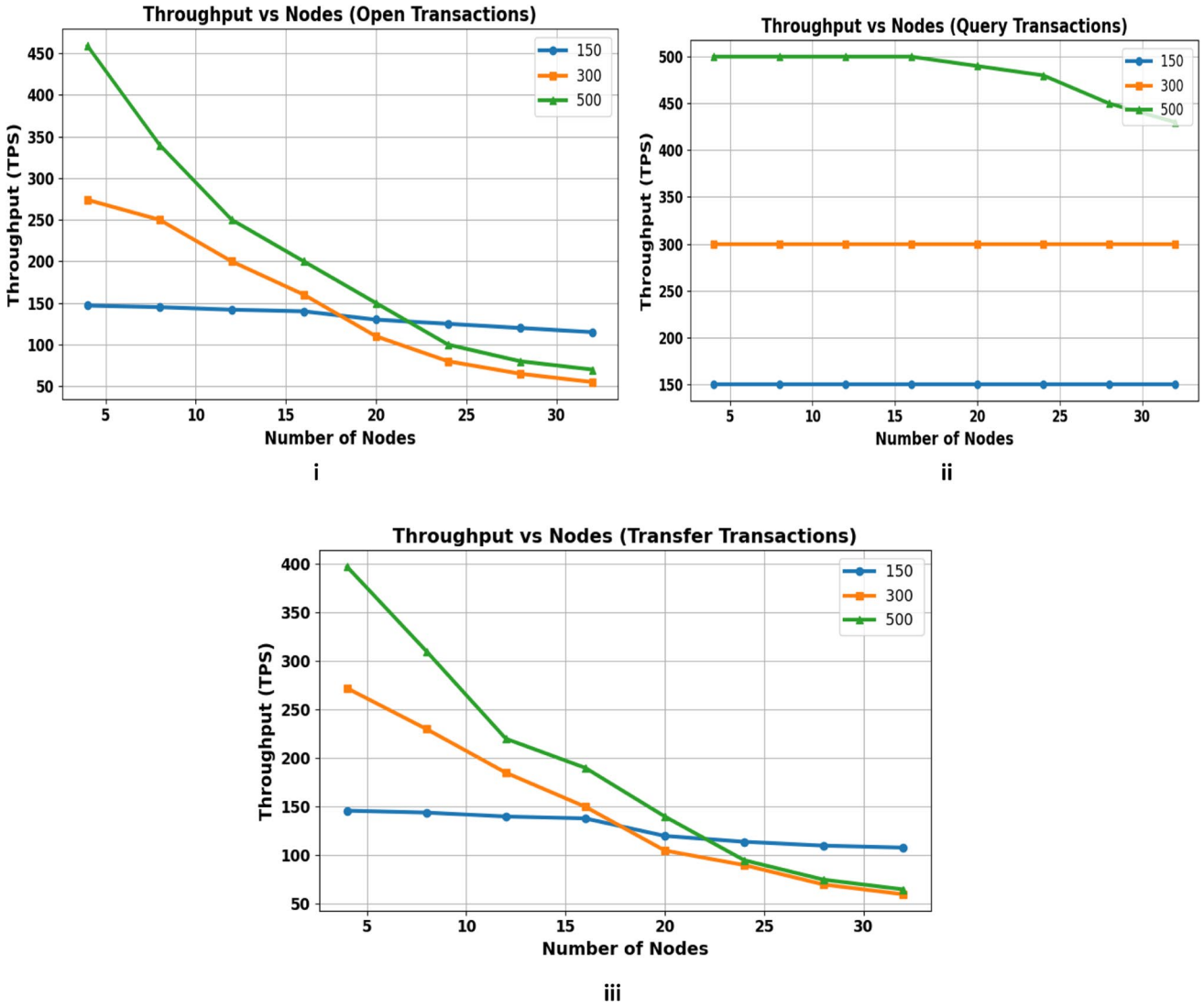


Fig. 8 QBFT Throughput Performance for send rates of 150, 300 and 500

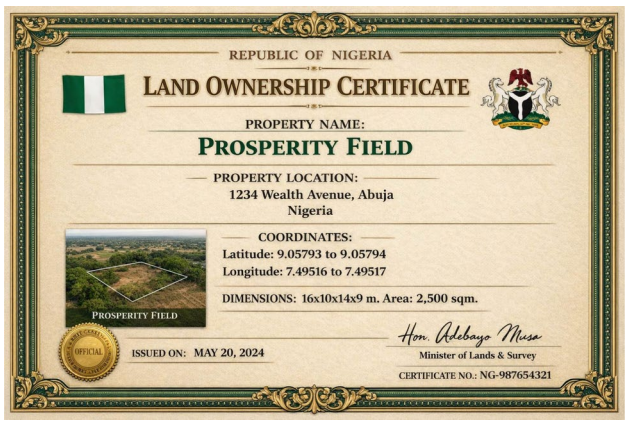


Fig. 9 Land certificate illustration

full control and re-establishing the token in its original pre-rental state.

Figure 16 illustrates the interface design of DApp for the proposed system. The left screen, Fig. 16(a), shows the government staff registration portal with a structured form that captures crucial land parameters, including geo-spatial coordinates (X_1, X_2, Y_1, Y_2), precise dimensional measurements, and IPFS-based land certificate URI for decentralized document storage. The right screen, Fig. 16(b), shows the land details interface displaying data of land ID 4, such as the property name "Prosperity Field", address "1234 Wealth Avenue, Abuja" and information like rental duration, alongside actionable buttons for rental and sale transactions.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS

```
[2026-02-11 17:12:38] [INFO   ] Uploading and pinning certificate to IPFS...
[2026-02-11 17:12:38] [INFO   ] File: prosperity_field.png (300.9 KB)
[2026-02-11 17:12:41] [SUCCESS] Certificate uploaded
[2026-02-11 17:12:41] [INFO   ] Document URI: ipfs://bafybeicvssbyewr5fkyttezn2d35ufaziasvv3h4547tfhscp3wspcrk2m
[2026-02-11 17:12:41] [INFO   ] Loading metadata...
[2026-02-11 17:12:41] [INFO   ] Land: Prosperity Field
[2026-02-11 17:12:41] [INFO   ] Uploading and pinning metadata to IPFS...
[2026-02-11 17:12:42] [SUCCESS] Registration complete
[2026-02-11 17:12:42] [INFO   ] Metadata URI: ipfs://bafybezkfjzdjiou2twl2vwyqkhk33p57gpkx44stzdfyjnjrx4is32wy
```

Fig. 10 Log output from terminal console when pinning to IPFS


```
"from": "0x8207D032322052AfB9Bf1463aF87fd0c0097EDDE",
"topic": "0x8486e68c6ebfccf90e1d345317c35520ebdc371c0cbb13534a9b0f098289da94",
"event": "landRegistered",
"args": {
        "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
        "1": "Prosperity Field",
        "2": "1234 Wealth Avenue, Abuja",
        "3": "Nigeria",
        "4": "9.05793",
        "5": "9.05794",
        "6": "7.49516",
        "7": "7.49517",
        "8": "16x10x14x9 m. Area: 2,500 sqm.",
        "9": "ipfs://bafybezkfjzdjiou2twl2vwyqkhk33p57gpkx44stzdfyjnjrx4is32wy",
        "10": "other",
        "11": "other2",
        "12": "other3",
        "13": "other4"
}
```


Fig. 11 Log showing a land is registered successfully after calling *safeMint*()


```
{
        "0": "uint256: id 4",
        "1":
"tuple(address,string,string,string,string,string,string,string,string,string,string,string,string,s
tring): data 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4,Prosperity Field,1234 Wealth Avenue,
Abuja,Nigeria,9.05793,9.05794,7.49516,7.49517,16x10x14x9 m. Area: 2,500
sqm.,ipfs://bafybezkfjzdjiou2twl2vwyqkhk33p57gpkx44stzdfyjnjrx4is32wy,other,other2,other3,other4",
        "2": "address: renter 0x0000000000000000000000000000000000000000",
        "3": "uint256: rentExpiresTime 0"
}
```


Fig. 12 Log output from smart contract function *getInfo*() showing data for registered land certificate with *tokenId* 4

```
"from": "0x8207D032322052AfB9Bf1463aF87fd0c0097EDDE",
"topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
"event": "Transfer",
"args": {
        "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
        "1": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
        "2": "5"
}
```


Fig. 13 Event log generated by the *transferLand*() function confirming the successful ownership transfer of land with *tokenId* 5. The emitted parameters record the previous owner address (Entry *0*), the new owner address (Entry *1*), and the *tokenId* (Entry *2*)


```
"from": "0x8207D032322052AfB9Bf1463aF87fd0c0097EDDE",
"topic": "0x4e06b4e7000e659094299b3533b47b6aa8ad048e95e872d23d1f4ee55af89cfe",
"event": "UpdateUser",
"args": {
        "0": "4",
        "1": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
        "2": "1802277768"
}
```


Fig. 14 Event log emitted after successful execution of the *setUser* function, indicating the establishment of a rental agreement for a tokenized land asset. The logged parameters include the asset identifier (Entry *0*), the renter address assigned temporary usage rights (Entry *1*), and the rental expiration timestamp (Entry *2*)

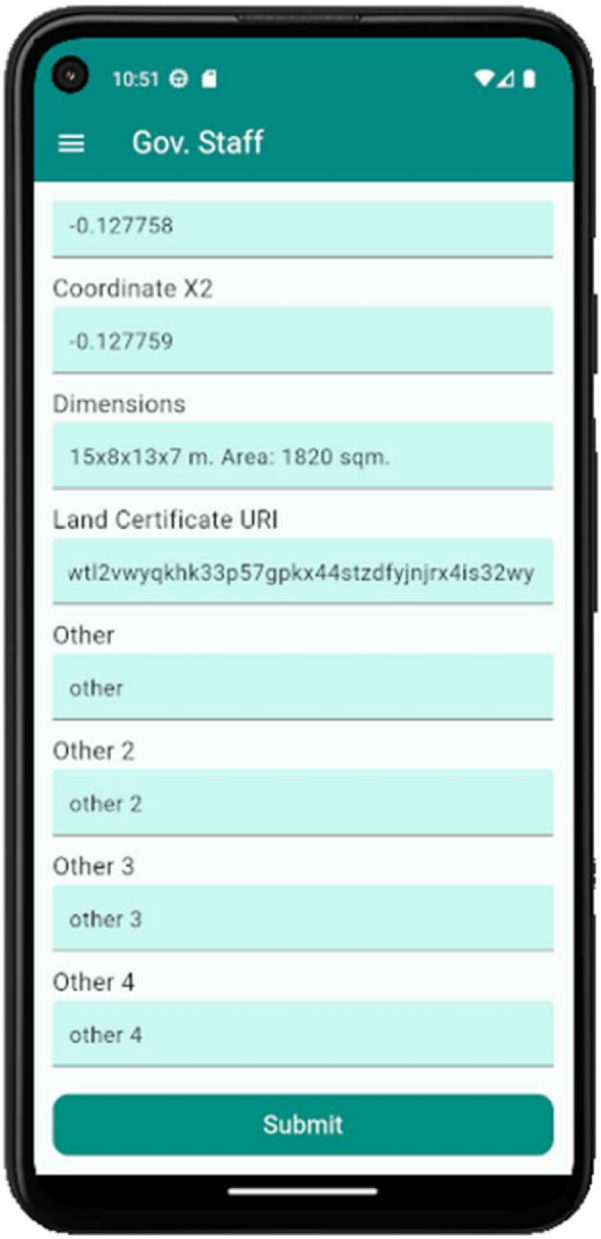

```
{
        "0": "uint256: id 4",
        "1":
"tuple(address,string,string,string,string,string,string,string,string,string,string,string,s
tring): data 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4,Prosperity Field,1234 Wealth Avenue,
Abuja,Nigeria,9.05793,9.05794,7.49516,7.49517,16x10x14x9 m. Area: 2,500
sqm.,ipfs://bafybezkfjzdjiou2twl2vwyqkhk33p57gpkx44stzdfyjnjrx4is32wy,other,other2,other3,other4",
        "2": "address: renter 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
        "3": "uint256: rentExpiresTime 1802277768"
}
```


Fig. 15 Log output from smart contract function *getInfo*() showing data for rented land with *tokenId* 4

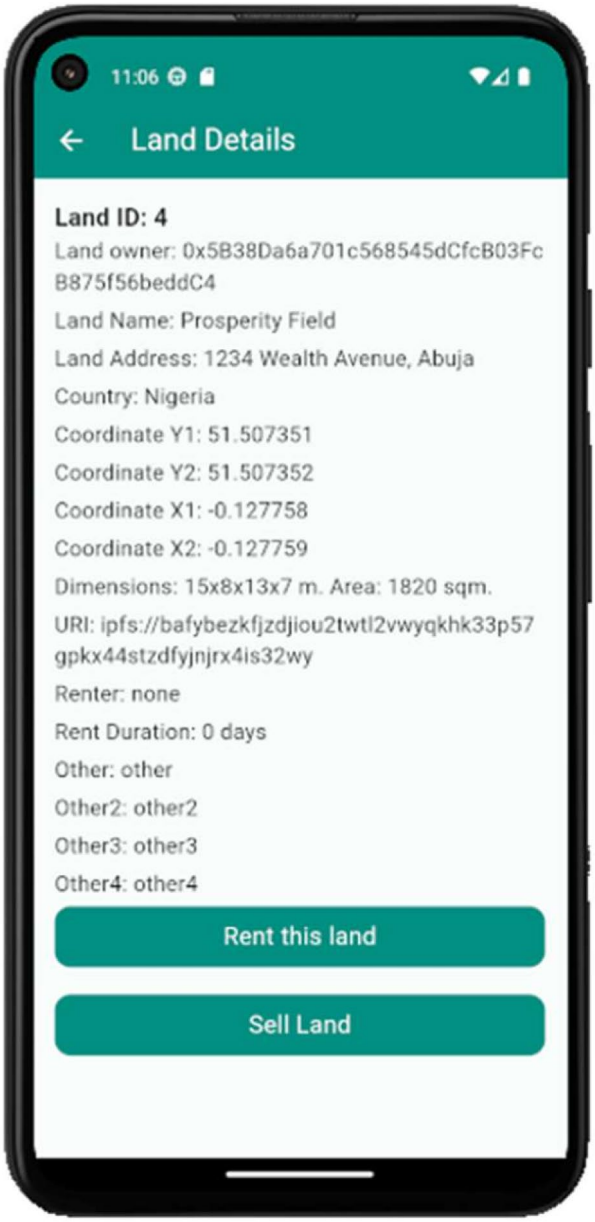
4.6 Fraud prevention and official accountability model using ERC-4907

Tokenizing land records does not eliminate all forms of corruption, it does not address the types of fraud that occur outside the blockchain or before the asset is registered in the blockchain. In our system, once a land parcel is minted as an ERC-4907 token, every subsequent operation, allocation, transfer, or modification must occur through a smart contract that enforces transparent, auditable, and tamper-proof rules. This ensures that no official can secretly alter ownership records, back-date approvals, or issue duplicate allocations, as all actions generate immutable on-chain traces. The cryptographic guarantees of ERC-4907 make silent or undetected record manipulation technically impossible.

Bribery is fundamentally a human-layer issue rather than a blockchain-layer vulnerability. Tokenization cannot fully prevent individuals from attempting to bribe an official to accelerate processing, however, it removes the official's ability to perform fraudulent technical actions in exchange for the bribe. Even if bribery occurs, the official cannot secretly approve an allocation, modify metadata, or alter the ledger without leaving a cryptographically verifiable record. Thus, the incentive and impact of bribery are significantly reduced because the system no longer allows invisible manipulation or off-ledger favors.



(a) Land Registration page for government staff.



(b) Land Details Page shows the full information about selected land (land ID 4).

Fig. 16 Decentralized App (DApp) user interface

To ensure full accountability, all government officials interacting with the system are required to use role-based, institution-controlled on-chain identities. Each official is assigned a dedicated wallet address. Every land allocation or approval action automatically records the transaction caller address field, binding the responsible official to the transaction. This design ensures that the individual (or institutional role) behind each approval is permanently visible, auditable, and traceable. Any abnormal pattern such as unusually fast approvals for a specific individual can be detected through on-chain analysis, thereby enabling governance bodies or auditors to identify potential misconduct. Tokenization prevents technical forms of fraud such as record tampering, duplicate land allocations, registry manipulation, and undocumented backdating by ensuring all actions are recorded immutably on the blockchain. However, it cannot fully eliminate human-driven issues like bribery to speed up approvals, coercion of officials, or political influence during the initial verification process.

In summary, while tokenization cannot eliminate bribery as a human behavior, it removes the possibility of secret fraudulent actions, enforces verifiable accountability, and provides strong transparency mechanisms that discourage and expose attempts to manipulate the land registry.

4.7 Comparison with existing solutions

In the current Nigerian land-administration workflow, the registration of a new property is inherently dominated by human-driven procedures such as physical site inspection, survey validation, file vetting, and multiple levels of governmental approval. These steps typically span days, weeks or months and introduce delays that cannot be removed by any digital platform, including blockchain-based systems. For this reason, a direct latency comparison between BERBLOM and the full legacy registration cycle would be misleading, as the bottleneck lies in manual verification rather than computational processing. The meaningful comparison instead occurs after a parcel has been registered and tokenized, where ownership changes and lease agreements traditionally require physical visits, intermediaries, and extensive paperwork. In BERBLOM, these post-registration processes peer-to-peer transfers and temporary leasing are executed entirely through smart contracts without middlemen, enabling near-instant settlement with full auditability and tamper-proof history. Thus, while the proposed system does not shorten the legally required registration procedures, it significantly accelerates and secures the majority of property lifecycle operations that currently suffer from administrative delays.

Table 6 highlights architectural gaps in the existing literature across three primary dimensions: economic viability, data scalability, and functional scope. First, regarding economic viability, frameworks built on public networks (e.g., [27, 44]) subject users to volatile gas fees, presenting a significant barrier to adoption in government-led initiatives. While permissioned solutions using Hyperledger Fabric (e.g., [30]) avoid these costs, they often sacrifice the interoperability provided by standard EVM token interfaces.

Second, regarding scalability, the integration of decentralized storage remains inconsistent. The majority of reviewed studies either omit heavy document storage entirely or rely on centralized databases, reintroducing the single points of failure that blockchain aims to eliminate. Only a minority of solutions (e.g., [35, 44]) leverage IPFS, yet they lack the accompanying private network infrastructure required for sensitive land data.

Finally, regarding functional scope, the comparison reveals a distinct absence of support for temporary land tenure. While traceability and permanent ownership transfer are widely explored, no prior study utilizes the ERC-4907 standard to facilitate automated, trustless leasing. BERBLOM uniquely synthesizes these requirements—combining the gas-free efficiency of Hyperledger Besu, the tamper-evident scalability of IPFS, and the dual-role rental logic of ERC-4907—to provide a holistic land administration framework that addresses the specific limitations of previous approaches.

Table 7 shows the experimental results against recent baselines established in similar Hyperledger Besu (QBFT) environments to contextualize the performance metrics of the proposed framework. Across the compared studies, validator configurations up to 12 nodes consistently exhibit latency levels that remain within a tolerable range.

Table 6 Comparison of earlier research on land records and transactions with proposed solution

Study	Network	IPFS	ERC-4907 Provision	Traceability	Renting Provision	Gas Free Network
Mendi et al. [30]	Hyperledger Fabric	✗	✗	✗	✗	✓
Gollapalli et al. [15]	Hyperledger Fabric	✗	✗	✗	✗	✓
Jahan et al. [24]	Public Network	✗	✗	✗	✗	✗
Mintah et al. [27]	Public Ethereum	✗	✗	✗	✗	✗
Sladić et al. [42]	Public Ethereum	✗	✗	✗	✗	✗
Panda et al. [35]	Public Ethereum	✓	✗	✗	✗	✗
Aborujilah et al. [6]	Permissioned Blockchain	✗	✗	✗	✗	✓
Alam et al. [3]	Ethereum Hybrid Blockchain	✗	✗	✗	✗	✗
Stefanovic et al. [43]	Public Ethereum	✗	✗	✓	✗	✗
Dubey et al. [11]	Private Ethereum	✗	✗	✓	✗	✗
Hari et al. [19]	Private Ethereum	✗	✗	✓	✗	✗
Alyounis and Yasin [5]	Hyperledger Fabric	✗	✗	✓	✗	✓
Sharma et al. [44]	Public Ethereum	✓	✗	✓	✗	✗
Rashid et al. [39]	Public Ethereum	✗	✗	✓	✗	✗
Gupta et al. [17]	Public Ethereum	✗	✗	✓	✗	✗
Paavo et al. [36]	Public Ethereum	✗	✗	✓	✗	✗
Proposed Solution	Hyperledger Besu (QBFT)	✓	✓	✓	✓	✓

Table 7 Performance comparison of QBFT-based blockchain systems by validator node count

	Throughput (TPS)			Avg Latency (s)		
Nodes	Ours	Tkachuk et al. [47]	Ferone and Verrilli [14]	Ours	Tkachuk et al. [47]	Ferone and Verrilli [14]
Write (Transfer)						
4	272	∼200	∼350	0.81	∼1.4	∼1
8	230	∼195	∼320	1.90	∼1.7	∼10
12	185	∼190	∼280	3.20	∼2	∼24
Read (Query)						
Any†	300	∼1440	—	0.001	< 0.5	—

Write throughput and latency are reported for our QFBT system (send rate: 300 req/s), Tkachuk et al. [47] (send rate: 200 req/s), and Ferone and Verrilli [14] (send rate: 400 req/s). Read performance is reported where available. Values marked with ∼ are approximated from published figures. Tkachuk et al. [47] tested write transactions at a send rate of 200 req/s (up to a maximum of 300 req/s) and read transactions at send rates ranging from 100 to 3000 req/s. Ferone and Verrilli [14] tested write transactions at 400 TPS, and no read benchmark was reported.

† Read throughput from Tkachuk et al. [47] is independent of validator count. Proposed framework's Query throughput matches the 300 req/s send rate with minuscule latency across all node counts

The proposed system achieves a write throughput of 272 TPS with an average latency of 0.81 seconds for transfer operations under a 4-node configuration. This performance exceeds the 200 TPS baseline observed by [47], likely due to the optimized nature of the ERC-4907 logic compared to the heavier settlement contracts used in energy trading. However, the throughput trails the peak of 350 TPS reported by [14].

Regarding scalability, all three studies exhibit a consistent trend: as the validator set expands from 4 to 12 nodes, throughput decreases and latency rises. Specifically, the throughput of the proposed system declines to 185 TPS at 12 nodes, a trajectory that closely mirrors the degradation curve reported by [47] (∼190 TPS). Notably, while [14] report a steeper latency spike at 12 nodes (∼10 seconds), the proposed framework maintains a manageable 3.20 seconds. This comparison confirms that the known characteristics of the QBFT consensus mechanism, specifically the communication overhead required to reach a two-thirds majority in larger networks, dictate these performance limits rather than inefficiencies in the smart contract architecture. Finally, for read operations (Query), the system consistently delivers near-instant responses (0.001s) independent of the validator count, validating its suitability for high-frequency public verification of land titles.

4.8 Limitations

This study demonstrates feasibility but has several important limitations:

- Governance: The prototype relies on a single deployer key, multi-party governance or multi-sig schemes may be required for production.
- Legal validity: ERC-4907 tokens are not yet legally recognized as land titles; integration with land registries and regulators remains open.
- Visible wallet identifiers: users' public addresses are visible on-chain.

5 Conclusion

The adoption of blockchain technology for land registration, verification, and transaction management offers promising opportunities for improving land-administration processes in developing countries. In this study, a blockchain-based prototype was designed and implemented to explore how tokenized land records can support more transparent and auditable ownership management. The results indicate that representing land parcels as ERC-4907 tokens can facilitate faster ownership verification through token identifiers and reduce reliance on conventional, non-automated administrative workflows for post-registration transactions.

It is important to emphasize that the proposed system does not replace legally mandated administrative procedures involved in initial land registration, which remain dependent on human-driven verification and governmental approval. Rather, the primary benefits of the framework are realized after tokenization, where ownership transfers and temporary leasing operations can be executed through smart contracts with improved traceability and reduced operational overhead.

While the prototype demonstrates the feasibility of blockchain-enabled land registries and suggests potential improvements in transparency and resistance to technical manipulation, several challenges remain. These include governance design, long-term data persistence, regulatory

integration, and the legal recognition of tokenized land titles. Addressing these limitations will be essential for transitioning from a proof-of-concept implementation to a production-ready system suitable for real-world deployment.

Future research will also include developing a customized NFT marketplace to improve user experience, as well as exploring the integration of Large Language Models (LLMs) to automate transaction reporting and generate accessible insights for stakeholders. To strengthen security in a real deployment, a Multi-signature (Multisig) approval scheme will be introduced so that sensitive operations require authorization from multiple officials. In addition, to reduce the risks associated with self-managed private keys, future iterations will explore Account Abstraction and custodial wallet solutions to enhance usability for non-technical landowners. We also plan to examine compliance with relevant privacy regulations such as the General Data Protection Regulation (GDPR) and the Nigeria Data Protection Regulation (NDPR), particularly for handling personal data and document metadata. Zero-Knowledge Proofs (ZKPs) may be incorporated to hide participant identities without compromising auditability. Interoperability with existing government land registry systems will also be investigated to support institutional integration and potential real-world deployment.

Appendix A: URI (Uniform Resource Identifier) and CID (Content Identifier) in IPFS

In the context of IPFS (InterPlanetary File System), the relationship between a URI (Uniform Resource Identifier) and CID (Content Identifier) is crucial for accessing and referencing content in a distributed network. The URI provides a means of referencing content, while the CID uniquely identifies that content based on its cryptographic hash. This section explores the mathematical relationship between the two and how they function within the IPFS ecosystem.

CID: Content Identifier

The CID is a cryptographic hash of the content stored in IPFS. It is the core identifier that uniquely represents a piece of content. When content C (such as a file or data object) is added to the IPFS network, it is processed through a cryptographic hash function H, such as SHA-256. This function produces a unique identifier, the CID, which serves as a reference to that specific content.

Mathematically, the relationship between the content and its CID is expressed as:

$$\text{CID} = H(C) \tag{A1}$$

Where:

- C represents the content to be stored in the IPFS network.
- H is the cryptographic hash function (e.g., SHA-256).
- CID is the resulting content identifier, a unique and fixed-length string derived from the content C.The CID ensures that each piece of content is uniquely identified based on its data. If the content changes, even slightly, the CID will change, ensuring content immutability and data integrity.

URI: Uniform Resource Identifier

A URI is used to reference and access content within the IPFS network. Specifically, an IPFS URI includes the CID as part of its structure, allowing the content to be retrieved via the IPFS protocol or through a public gateway.

Mathematically, a URI can be represented as a function f that maps a CID to a specific URI:

$$\text{URI} = f(\text{CID}) \tag{A2}$$

Where the function f provides a mapping between the CID and the URI that points to the content. The URI is typically expressed as:

$$\text{URI} = \text{scheme} + \text{“}://\text{”} + \text{CID} \tag{A3}$$

For example:

$$\text{URI} = \text{ipfs://QmZGhXc9w5iU2T7p7eW8A9cwvq8Xq5kPbKPnm4U5YbnkDC} \tag{A4}$$

Where:

- scheme is typically “ipfs” or a public gateway URL like “https://ipfs.io/ipfs/”.
- CID is the content identifier that uniquely references the content in the IPFS network.The URI provides a convenient way to access the content identified by the CID, allowing users to retrieve the data either through the IPFS network or through a publicly accessible gateway.

Acknowledgements This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT)

(IITP-2026-RS-2020-II201612, 20%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 20%) and by the MSIT, Korea, under the ITRC support program (IITP-2026-RS-2024-00438430, 20%) and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(RS-2025-25431637, 20%) and the regional innovation system & education (RISE)-(Regional Growth Innovation LAB) program through the Gyeongbuk RISE Center, funded by the Ministry of Education (MOE) and the Gyeongsangbuk-do, Republic of Korea(2026-rise-15-105, 20%)

Author Contributions O.U.N. contributed to the research, methodology, design, software development, and manuscript writing. M.R.R.A. contributed to the research, software development, and writing. E.A.T. contributed to editing and writing. D.-S.K. secured funding and provided supervision. J.-M.L. also secured funding and provided supervision. All authors reviewed and approved the final manuscript.

Funding This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201612, 20%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 20%) and by the MSIT, Korea, under the ITRC support program (IITP-2026-RS-2024-00438430, 20%) and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(RS-2025-25431637, 20%) and the regional innovation system & education (RISE)-(Regional Growth Innovation LAB)program through the Gyeongbuk RISE Center, funded by the Ministry of Education (MOE) and the Gyeongsangbuk-do, Republic of Korea(2026-rise-15-105, 20%)

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Allwinnaldo, Ansori MRR, Alief RN, Lee JM, Kim D-S et al (2023) Efficient blockchain-based energy auction platform for secured energy trading on industrial applications. Korean Instit Commun Inf Sci 48(7):885–895
2. Ansori MRR, Allwinnaldo, Alief RN, Igboanusi IS, Lee JM, Kim D-S (2023) Hades: Hash-based audio copy detection system for copyright protection in decentralized music sharing. IEEE Trans Netw Serv Manag 20(3):2845–2853. https://doi.org/10.1109/TNSM.2023.3241610
3. Alam KM, Ashfiqur Rahman JM, Tasnim A, Akther A (2022) A blockchain-based land title management system for bangladesh. Journal of King Saud University - Computer and Information Sciences 34(6, Part A), 3096–3110. https://doi.org/10.1016/j.jksuci.2020.10.011
4. Anders (@LanceSnow) L (2022) Shrug: ERC-4907: Rental NFT, an Extension of EIP-721 [DRAFT]. Ethereum Improvement Proposals (EIP-4907). [Online]. Available: https://eips.ethereum.org/EIPS/eip-4907
5. Alyounis S, Yasin M (2023) Secure framework for land record management using blockchain technology. J Cyber Secur Risk Audit 2023:19–48. https://doi.org/10.63180/jcsra.thestap.2023.1.3
6. Aborujilah A, Yatim MNBM, Al-Othmani A (2021) Blockchain-based adoption framework for authentic land registry system in malaysia. TELKOMNIKA (Telecommun Comput Electron Control) 19(6):2038–2049. https://doi.org/10.12928/telkomnika.v19i6.19276
7. Ben Fekih R, Lahami M, Bradai S, Jmaiel M (2025) Formal verification of erc-based smart contracts: A systematic literature review. IEEE Access 13:11396–11422. https://doi.org/10.1109/ACCESS.2025.3527158
8. Bhawana, Kumar S, Dohare U, Kaiwartya O (2023) Flame: Trusted fire brigade service and insurance claim system using blockchain for enterprises. IEEE Trans Industr Inf 19(6):7517–7527. https://doi.org/10.1109/TII.2022.3212172
9. Barbhaya UR, Vishwakarma L, Das D (2024) Etradechain: Blockchain-based energy trading in local energy market (lem) using modified double auction protocol. IEEE Trans Green Commun Netw 8(1):559–571. https://doi.org/10.1109/TGCN.2023.3307360
10. Delladetsimas AP, Papangelou S, Iosif E, Giaglis G (2025) Leadership uniformity in timeout-based quorum byzantine fault tolerance (qbft) consensus. Big Data Cogn Comput 9(8):196
11. Dubey S, Vinod D, Gupta A, Dwivedi RK (2023) Secure land registration management via ethereum blockchain. In: 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp 185–191. https://doi.org/10.1109/IDCIoT56793.2023.10053494
12. Farshidi S, Jansen S, España S, Verkleij J (2020) Decision support for blockchain platform selection: Three industry case studies. IEEE Trans Eng Manage 67(4):1109–1128. https://doi.org/10.1109/TEM.2019.2956897
13. Fan C, Lin C, Khazaei H, Musilek P (2022) Performance analysis of hyperledger besu in private blockchain. In: 2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), pp 64–73. https://doi.org/10.1109/DAPPS55202.2022.00016
14. Ferone A, Verrilli S (2025) Exploiting blockchain technology for enhancing digital twins' security and transparency. Future Internet 17(1):31
15. Gollapalli SA, Krishnamoorthy G, Jagtap NS, Shaikh R (2020) Land registration system using block-chain. In: 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), pp 242–247. https://doi.org/10.1109/ICSIDEMPC49020.2020.9299606
16. Garcia RD, Ramachandran GS, Jurdak R, Ueyama J (2022) Blockchain-aided and privacy-preserving data governance in multi-stakeholder applications. IEEE Trans Netw Serv Manage 19(4):3781–3793. https://doi.org/10.1109/TNSM.2022.3225254
17. Gupta R, Shah MN, Mandal SN (2025) Land registry using blockchain. Intern J Res Appl Sci Eng Technol 13:1377–1380. https://doi.org/10.22214/ijraset.2025.68367

18. Geneiatakis D, Soupionis Y, Steri G, Kounelis I, Neisse R, Nai-Fovino I (2020) Blockchain performance analysis for supporting cross-border e-government services. IEEE Trans Eng Manage 67(4):1310–1322. https://doi.org/10.1109/TEM.2020.2979325
19. Hari MK, Agrawal A, Bhatia R, Bhatia A, Tiwari K (2023) T-pass: A blockchain-based nft enabled property management and exchange system. In: 2023 International Conference on Information Networking (ICOIN), pp 140–145. https://doi.org/10.1109/ICOIN56518.2023.10048973
20. Haryadi GA, Ansori MRR, Lee J-M, Kim D-S (2023) Efficient voucher distribution and non-transferability in nft loyalty programs. Korean Instit Commun Inf Sci 858–859
21. Hassan MU, Rehmani MH, Chen J (2022) Optimizing blockchain based smart grid auctions: A green revolution. IEEE Trans Green Commun Netw 6(1):462–471. https://doi.org/10.1109/TGCN.2021.3095424
22. Ibrahim I, Daud D, Azmi FAM, Noor NAM, Yusoff NSM (2021) Improvement of land administration system in nigeria: A blockchain technology review. Intern J Scientif Technol Res 10(08)
23. Igboanusi IS, Kim D-S (2023) Blockchain enabled medical record and organ matching integrated system. Korean Instit Commun Inf Sci 870–870
24. Jahan F, Mostafa M, Chowdhury S (2020) Sha-256 in parallel blockchain technology: Storing land related documents. Intern J Comput Appl 175:33–38. https://doi.org/10.5120/ijca2020920911
25. Konagari A, Kusuma HP, Chetharasi S, Kuchipudi R, Babu PR, Murthy TS (2023) Nft marketplace for blockchain based digital assets using erc-721 token standard. In: 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), pp 1394–1398. https://doi.org/10.1109/ICSCSS57650.2023.10169350
26. Luo L, Feng J, Yu H, Sun G (2022) Blockchain-enabled two-way auction mechanism for electricity trading in internet of electric vehicles. IEEE Internet Things J 9(11):8105–8118. https://doi.org/10.1109/JIOT.2021.3082769
27. Mintah K, Baako K, Kavaarpuo G, Otchere GK (2020) Skin lands in ghana and application of blockchain technology for acquisition and title registration. J Propert Plann Environm Law ahead-of-print. https://doi.org/10.1108/JPPEL-12-2019-0062
28. Muhammad NB, Mustapha Z, Iro ZS (2022) Towards adopting a blockchain-based real estate transaction in federal capital territory (abuja). Nigerian J Comput Eng Technol
29. Mohaghegh M, Panikkar A (2020) A decentralised land sale and ownership tracking system using blockchain technology. In: 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), pp 1–8. https://doi.org/10.1109/CITISIA50690.2020.9371777
30. Mendi AF, Sakaklı KK, Çabuk A (2020) A blockchain based land registration system proposal for turkey. In: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp 1–6. https://doi.org/10.1109/ISMSIT50672.2020.9255078
31. Mastilak L, Suchy R, Kostal K, Kotuliak I (2023) Blockchain-based sustainable retail loyalty program. IEEE Access 11:75063–75075. https://doi.org/10.1109/ACCESS.2023.3296914
32. Morsidi M, Tajuddin S, Patchmuthu RK, Newaz SHS (2021) Blockchain-based reward system: a means for providing incentive to students for teaching feedback. In: 2021 International Conference on Electronics, Communications and Information Technology (ICECIT), pp 1–5. https://doi.org/10.1109/ICECIT54077.2021.9641311
33. Nguyen CT, Hoang DT, Nguyen DN, Pham H-A, Tuong NH, Dutkiewicz E (2021) Blockchain-based secure platform for coalition loyalty program management. In: 2021 IEEE Wireless Communications and Networking Conference (WCNC), pp 1–6. https://doi.org/10.1109/WCNC49053.2021.9417501
34. Njoku JN, Nwakanma CI, Lee J-M, Kim D-S (2023) Enhancing security and accountability in autonomous vehicles through robust speaker identification and blockchain-based event recording. MDPI Electronics 12(24). https://doi.org/10.3390/electronics12244998
35. Panda SK, Mohammad GB, Nandan Mohanty S, Sahoo S (2021) Smart contract-based land registry system to reduce frauds and time delay. Secur Priv 4(5):172. https://doi.org/10.1002/spy2.172
36. Paavo JP, Rodríguez-Puentes R, Chigbu UE (2025) Practicality of blockchain technology for land registration: A namibian case study. Land 14(8). https://doi.org/10.3390/land14081626
37. Pradhan NR, Singh AP, Kumar N, Hassan MM, Roy DS (2022) A flexible permission ascription (fpa)-based blockchain framework for peer-to-peer energy trading with performance evaluation. IEEE Trans Industr Inf 18(4):2465–2475. https://doi.org/10.1109/TII.2021.3096832
38. Rahaman MF, Golam M, Putra MAP, Haryadi GA, Kim D-S, Lee J-M (2023) Blockchain empowered secure medical appointment for the patients using smart contract. Korean Instit Commun Inf Sci 868–869
39. Rashid MRA, Rafi AA, Islam MA, Sharkar SU, Rafi ZH, Hasan M, Ali MS, Khan MSH (2025) Enhancing land management policy in bangladesh: A blockchain-based framework for transparent and efficient land management. Land Use Policy 150:107436. https://doi.org/10.1016/j.landusepol.2024.107436
40. Saeda S, Barau AS (2009) Nigeria's land reform from fronts of multidisciplinary approach. In: 2009, 5th School of Arts and Social Sciences - Federal College of Education Kano - National Confab. https://doi.org/10.13140/2.1.3190.5603
41. Shaikh AA, Dahmani N, Khan S, Sharma R (2023) Blockchain-enabled platform for a meta customer loyalty program. In: 2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD), pp 1–9. https://doi.org/10.1109/ITIKD56332.2023.10099907
42. Sladić G, Milosavljević B, Nikolić S, Sladić D, Radulović A (2021) A blockchain solution for securing real property transactions: A case study for serbia. ISPRS Intern J Geo-Inf 10(1). https://doi.org/10.3390/ijgi10010035
43. Stefanovic M, Przulj D, Ristic S, Stefanovic D, Nikolics D (2022) Smart contract application for managing land administration system transactions. IEEE Access 10:39154–39176. https://doi.org/10.1109/ACCESS.2022.3164444
44. Sharma A, Sharma A, Tripathi A, Chaudhary A (2024) Real estate registry platform through nft tokenization using blockchain. In: 2024 2nd International Conference on Disruptive Technologies (ICDT), pp 335–340. https://doi.org/10.1109/ICDT61202.2024.10488945
45. Tran-Dang H, Kim D-S (2023) Online learning based matching for decentralized task offloading in fog-enabled iot systems. In: 2023 28th Asia Pacific Conference on Communications (APCC), pp 231–236. https://doi.org/10.1109/APCC60132.2023.10460738
46. Tariq MM, Ihsan U, Alamsyah Z (2025) Land registration and inheritance automation system using blockchain. Eng Proceed 107(1). https://doi.org/10.3390/engproc2025107097
47. Tkachuk R-V, Ilie D, Robert R, Kebande V, Tutschku K (2024) On the performance and scalability of consensus mechanisms in privacy-enabled decentralized renewable energy marketplace. Ann Telecommun 79(3):271–288
48. Thi Kim OT, Le THT, Shin MJ, Nguyen V, Han Z, Hong CS (2022) Distributed auction-based incentive mechanism for energy trading between electric vehicles and mobile charging stations.

IEEE Access 10:56331–56347. https://doi.org/10.1109/ACCESS.2022.3170709
49. Taylor A, Kugler A, Marella PB, Dagher GG (2022) Vigilrx: A scalable and interoperable prescription management system using blockchain. IEEE Access 10:25973–25986. https://doi.org/10.1109/ACCESS.2022.3156015
50. Thompson MJ, Sun H, Jiang J (2022) Blockchain-based peer-to-peer energy trading method. CSEE J Power Energ Syst 8(5):1318–1326. https://doi.org/10.17775/CSEEJPES.2021.00010
51. Wu G, Wang S, Ning Z, Zhu B (2022) Privacy-preserved electronic medical record exchanging and sharing: A blockchain-based smart healthcare system. IEEE J Biomed Health Inform 26(5):1917–1927. https://doi.org/10.1109/JBHI.2021.3123643
52. Wang Y, Zhang A, Zhang P, Qu Y, Yu S (2022) Security-aware and privacy-preserving personal health record sharing using consortium blockchain. IEEE Internet Things J 9(14):12014–12028. https://doi.org/10.1109/JIOT.2021.3132780
53. Ye L, Chen B, Shivanshu S, Sun C, Wang S, Feng S, Zhang S (2024) Dynamic spectrum sharing based on the rentable nft standard erc4907. In: 2024 13th International Conference on Communications, Circuits and Systems (ICCCAS), pp 503–507. https://doi.org/10.1109/ICCCAS62034.2024.10652863
54. Zainudin A, Alief RN, Adi Paramartha Putra M, Akter R, Lee J-M, Kim D-S (2023) Blockchain-aided decentralized collaborative automatic modulation classification for next-generation wireless networks. In: 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), pp 1058–1063. https://doi.org/10.1109/ICTC58733.2023.10393633
55. Zainudin A, Alief RN, Putra MAP, Akter R, Kim D-S, Lee J-M (2023) Blockchain-based decentralized trust aggregation for federated cyber-attacks classification in sdn-enabled maritime transportation systems. In: 2023 IEEE International Conference on Communications Workshops (ICC Workshops), pp 182–187. https://doi.org/10.1109/ICCWorkshops57953.2023.10283507
56. Zahuruddin MM, Gupta DS, Akram SW (2021) Land registration using blockchain technology. J Emerg Technol Innov Res (JETIR) 8(6):657–667
57. Zainudin A, Putra MAP, Alief RN, Akter R, Kim D-S, Lee J-M (2024) Blockchain-inspired collaborative cyber-attacks detection for securing metaverse. IEEE Internet Things J 11(10):18221–18236. https://doi.org/10.1109/JIOT.2024.3364247
58. Zhuang Y, Sheets LR, Chen Y-W, Shae Z-Y, Tsai JJP, Shyu C-R (2020) A patient-centric health information exchange framework using blockchain technology. IEEE J Biomed Health Inform 24(8):2169–2176. https://doi.org/10.1109/JBHI.2020.2993072

Odinachi Udemezuo Nwankwo Odinachi earned his undergraduate degree in Electrical and Electronic Engineering from the Federal University of Technology, Owerri, in 2014. He completed his Master's Degree in IT Convergence Engineering at Kumoh National Institute of Technology, South Korea in 2024. His research interests include networked embedded systems, real-time systems, network security, IoT, AI, and blockchain. [ORCID: 0009-0008-9595-921X]

Muhammad Rasyid Redha Ansori Muhammad Rasyid Redha Ansori is currently working as a researcher at NSLab Inc., South Korea. He holds a Master's degree in IT Convergence Engineering from Kumoh National Institute of Technology, South Korea in 2023. He received his Bachelor's degree inTelecommunication Engineering from Telkom University, Indonesia in 2020. His research interests include blockchain, informationsecurity, and real-time systems. [ORCID:0000-0002-5704-9134]

Esmot Ara Tuli She received her Ph.D. degree in IT Convergence Engineering from Kumoh National Institute of Technology (KIT), South Korea, in 2024. She pursued her B.Sc. Eng. degree in Computer Science and Engineering from Jatiya Kabi Kazi Nazrul Islam University, Bangladesh. Currently, she is working as a Post-doctoral Research Fellow in the ICT Convergence Research Center, KIT, South Korea. She received the Outstanding Academic Paper Award in the Doctoral Degree Program from the IT Convergence Engineering Department at KIT. Her major research interests include metaverse, digital twins, signal processing, and quantum computing. [ORCID: 0000-0002-7099-398X]

Dong-Seong Kim Dong-Seong Kim (Senior Member, IEEE) received his Ph.D. degree in Electrical and Computer Engineering from Seoul National University, Seoul, Korea, in 2003. From 1994 to 2003, he worked as a full-time researcher at ERC-ACI at Seoul National University. From March 2003 to February 2005, he served as a postdoctoral researcher at the Wireless Network Laboratory in the School of Electrical and Computer Engineering at Cornell University, NY. Between 2007 and 2009, he was a visiting professor in the Department of Computer Science at the University of California, Davis, CA. He served as Dean of IACF from 2019 to 2022. He is currently a Professor with the Department of IT Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, South Korea. He is also the director of the KIT Convergence Research Institute and the ICT Convergence Research Center (ITRC and NRF advanced research center program), supported by the Korean government at Kumoh National Institute of Technology, and the director of NSLab Co. Ltd. He is a senior member of IEEE and ACM. His primary research interests include realtime IoT and smart platforms, industrial wireless control networks, networked embedded systems, fieldbus, metaverse, and blockchain. [ORCID:0000-0002-2977-5964]

Jae-Min Lee Jae-Min Lee received his Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2005. From 2005 to 2014, he was a senior engineer at Samsung Electronics in Suwon, Korea. From 2015 to 2016, he was a Principal Engineer at Samsung Electronics, Suwon, Korea. Since 2017, he has been an Associate Professor in the School of Electronic Engineering and the Department of IT Convergence Engineering at Kumoh National Institute of Technology, Gyeongbuk, Korea. Since 2024, he has been the Director of the Smart Defense Logistics Innovation Convergence Research Center (ITRC Program). He is a member of IEEE. He is the Executive Director of the Korean Institute of Communications and Information Sciences (KICS). His current main research interests are smart IoT convergence application, industrial wireless control network, UAV, Metaverse and Blockchain. [ORCID:0000-0001-6885-5185]