

BlockBricks: A Blockchain-Based Decentralized Rental Ecosystem

Kartikey Gupta

Dept. of Computer Science & Engineering
Acropolis Institute of Technology and Research
Indore, India

kartikeygupta220679@acropolis.in

Kartik Sisodiya

Dept. of Computer Science & Engineering
Acropolis Institute of Technology and Research
Indore, India

kartiksisodiya220889@acropolis.in

Lakshya Raghuwanshi Kand

Dept. of Computer Science & Engineering
Acropolis Institute of Technology and Research
Indore, India

lakshyaraghuwanshi220268@acropolis.in

Khushdil Ansari

Dept. of Computer Science & Engineering
Acropolis Institute of Technology and Research
Indore, India

khushdilansari220928@acropolis.in

Kunal Singh Jadon

Dept. of Computer Science & Engineering
Acropolis Institute of Technology and Research
Indore, India

kunalsingh220500@acropolis.in

Dr. Krupi Saraf

Dept. of Computer Science & Engineering
Acropolis Institute of Technology and Research
Indore, India

narendrasingh@acropolis.in

Abstract—Property rental systems continue to suffer from opacity, high brokerage fees, and limited contractual transparency. Centralized platforms such as Airbnb, NoBroker, and MagicBricks digitize operations but impose hidden charges and retain centralized control, leading to trust and data security concerns. This paper introduces BlockBricks, a blockchain-based decentralized rental ecosystem that automates lease creation, and ensures immutable recordkeeping through Ethereum smart contracts, IPFS storage, MongoDB integration, and Razorpay-based hybrid escrow. Experimental evaluation on Ganache and the Ethereum Sepolia testnet demonstrates average contract execution times of approximately 15 seconds and transaction costs reduced from 10–15% (centralized models) to around 1–2%. By combining blockchain transparency with practical escrow compliance, BlockBricks achieves a balance of security and cost efficiency, offering a sustainable and scalable alternative to conventional rental management platforms.

Keywords—Blockchain, Decentralized Applications, Hybrid Escrow, IPFS, Rental Ecosystem, Smart Contracts

I. INTRODUCTION

The property rental sector faces persistent inefficiencies, opaque agreements, and dependence on intermediaries that inflate costs and increase fraud risks. Centralized platforms such as Airbnb and MagicBricks have digitized rental processes but continue to charge high commissions, control user data, and remain prone to disputes and breaches. Li et al. [1]. As introduced by Nakamoto [2] and later advanced by Buterin [3], blockchain offers immutability, transparency, and automation through smart contracts, enabling trustless digital agreements without intermediaries. Its decentralized architecture eliminates ensures tamper-proof, self-enforcing lease transactions [4].

To overcome these limitations, this paper proposes BlockBricks, a blockchain-based rental management ecosystem designed to enhance transparency, security, and cost efficiency. BlockBricks employs smart contracts for automated lease enforcement, the Inter Planetary File System (IPFS) for decentralized document storage, and Razorpay integration for a hybrid escrow mechanism that links on-chain contracts with off-chain transactions. The contributions of this research include:

1. The design and implementation of BlockBricks, a decentralized framework for property rental management.
2. The integration of a hybrid escrow model combining blockchain-based contracts with Razorpay payment workflows.
3. The evaluation of gas consumption, transaction latency, and operational costs compared with centralized rental platforms.

II. LITERATURE REVIEW

Centralized rental platforms such as Airbnb, NoBroker, and MagicBricks currently dominate the property rental market. Although these systems simplify listing, booking, and payment, they remain constrained by high service charges, dependence on intermediaries, and limited contractual transparency. For instance, Airbnb imposes commissions of 10–15%, while subscription-based models like MagicBricks retain centralized control over transactions and data, perpetuating issues of trust, enforceability, and cost.

Several studies have explored blockchain applications in the real estate domain to mitigate these challenges. Alvarenga et al. [5] proposed the use of smart contracts for automating rental agreements, improving trust and reducing human intervention. Qi-Long et al. [6] demonstrated a blockchain-enabled housing management system in China that enhanced transaction transparency. Meijer and Ubacht [7] analyzed the practical and governance challenges in integrating blockchain into rental

markets. Additionally, Li et al [1] and the LockChain Project [8] emphasized decentralized architectures to eliminate intermediaries and lower transaction costs in property management systems.

These studies collectively validate blockchain's potential to reduce fraud, disputes, and operational costs in rental ecosystems. However, existing implementations primarily focus on on-chain agreements, leaving gaps in integrating fiat-based systems and regulatory compliance. To bridge this gap, BlockBricks adopts a hybrid architecture that combines Ethereum smart contracts for automation, IPFS for decentralized document storage, and Razorpay integration for secure, escrow linking between blockchain verification and real-world payment gateways [1][5]-[8].

III. SYSTEM DESIGN

BlockBricks has a layered architecture integrating blockchain, decentralized storage, and regulated financial gateways. The system comprises five layers:

1. **Actors:** Landlords, tenants, and administrators operate through role-specific interfaces.
2. **Frontend:** Developed using React.js, it provides dashboards for property listing, lease management, and transaction tracking.
3. **Backend:** Implemented with Node.js and Express, it manages business logic, authentication, and communication between blockchain and off-chain services.
4. **Storage and Services:** MongoDB indexes metadata and state information, while IPFS stores rental agreements, KYC proofs, and images. Razorpay provides fiat-based escrow functionality.
5. **Blockchain:** Ethereum smart contracts automate lease creation, enforce terms, and record immutable events for auditability [9]-[11].

A. Workflow

The rental process begins when a landlord lists a property through the frontend interface. The backend validates and

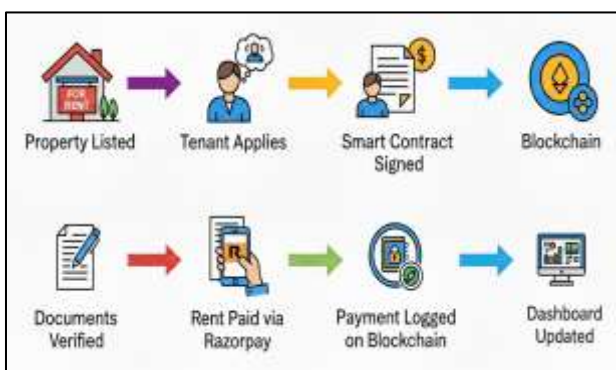


Fig. 1. System Flow.

uploads associated files to IPFS, generating cryptographic content identifiers (CIDs). These CIDs are recorded on the



Fig. 4. User Interface (Tenant Page).

Ethereum blockchain for tamper-evidence. A smart contract is then deployed to define lease terms and initiate escrow through Razorpay. Tenants apply for properties, submit deposits to escrow, and interact with the contract during the lease period. The contract autonomously enforces terms such as payments and penalties, logging all transactions and events on-chain. Upon lease completion or termination, escrow funds are released or refunded based on contract-defined outcomes, ensuring transparency and fairness throughout the rental lifecycle.

B. Hybrid Storage, Escrow and Blockchain integration

To optimize scalability, cost, and regulatory compliance, BlockBricks employs a hybrid architecture linking on-chain verification with off-chain resources. Documents and media files are stored on IPFS, which generates immutable CIDs anchored to Ethereum smart contracts. MongoDB indexes these CIDs for quick retrieval within the user interface. Razorpay serves as a regulated escrow provider, where each fiat transaction is hashed and linked on-chain, establishing traceability between blockchain events and real-world payments. This model reduces gas costs, prevents data redundancy and enhances user trust through transparent, tamper-proof storage and payment verification [8]-[11].

IV. IMPLEMENTATION AND RESULTS

A. Implementation

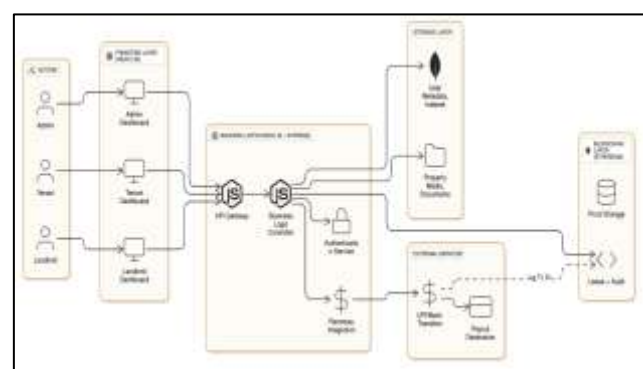


Fig. 2. System Architecture.

The BlockBricks prototype is designed using a multi-layer architecture that integrates smart contracts, decentralized storage, and regulated payment infrastructure. The frontend is built in React.js, providing dedicated interfaces for landlords, tenants, and administrators. The backend, implemented with Node.js and Express, manages business logic, RESTful API communication, IPFS uploads, MongoDB operations, and interactions with Ethereum smart contracts.

The blockchain layer consists of Solidity-based contracts deployed on the Sepolia testnet after preliminary validation on Ganache. MongoDB is used for metadata indexing, while IPFS stores property documents and lease files to ensure decentralized, tamper-proof access. The payment layer incorporates Razorpay's API to support INR-based escrow operations; each payment generates a unique transaction ID that is hashed using SHA-256 and anchored on-chain to provide immutable proof of payment [12]-[13].

The smart contract suite contains three core modules—Listing Management, Lease Agreement, and Escrow Logging—automating key rental lifecycle events, including property registration, agreement creation, and fund logging. To optimize gas usage, only content hashes are stored on-chain, while full documents remain off-chain in IPFS. Razorpay's sandbox environment successfully validated the escrow workflow, confirming reliable payment creation and consistent linkage between off-chain transaction data and on-chain event logs. Deployment tests on Sepolia consumed approximately 160,000–180,000 gas units per contract, with average confirmation times of 14–16 seconds, demonstrating efficient and stable transaction performance.

B. Experimental Evaluation

1) *Smart Contract Gas Usage:* Gas usage for core smart contract functions was measured using Hardhat test scripts and averaged over multiple iterations ($n = 10$). The cost of each transaction was calculated using the following formulas:

$$\text{USD}_{\text{cost}} = \text{ETH}_{\text{cost}} \times \text{ETH}_{(\text{USD})} \quad (2)$$

$$\text{INR}_{\text{cost}} = \text{USD}_{\text{cost}} \times \text{USD}_{(\text{INR})} \quad (3)$$

Assumptions: Gas Price = 3 gwei, ETH = \$4,526.82, USD → INR = ₹88.12, Test Date = 15 September 2025.

Smart contract deployment is the most gas-intensive operation, while recurring interactions such as payment logging and CID storage incur minimal costs (below ₹100). When tested on Layer-2 networks like Polygon, gas consumption decreased by approximately 35–40%, enabling near-negligible transaction fees with significantly faster confirmation times, calculations shown in Table 1.



Fig.3. User Interface (Home Page)

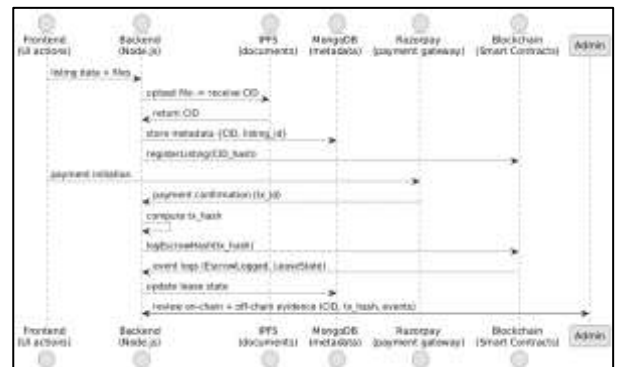


Fig. 5. Data Flow Diagram.

$$\text{ETH}_{\text{cost}} = \text{Gas Used} \times \text{Gas Price (gwei)} \times 10^{-9} \quad (1)$$

TABLE III. COMPARATIVE COST AND LATENCY ANALYSIS: CENTRALIZED PLATFORMS VS BLOCKBRICKS

Operation	Gas Used (avg.)	ETH Used (Mainnet)	Approx. USD Cost	Approx. Cost in ₹ (Ethereum)	Approx. Cost in ₹ (Layer-2)
Smart Contract Deployment	350k – 500k	~0.0012–0.0014	\$5–7	₹440 – ₹580	₹25 – ₹40
Store IPFS CID (hash)	40k – 60k	~0.00013–0.00017	\$0.6–0.8	₹50 – ₹70	₹5 – ₹10
Log Monthly Rent Payment	25k – 40k	~0.00008–0.00011	\$0.4–0.55	₹35 – ₹45	₹4 – ₹8
Record Escrow Tx ID	30k – 45k	~0.00010–0.00014	\$0.45–0.7	₹38 – ₹60	₹6 – ₹10
Final Lease Settlement	50k – 70k	~0.00016–0.00020	\$0.75–0.95	₹65 – ₹80	₹10 – ₹15

TABLE II. COMPARISON OF EXISTING PLATFORMS AND BLOCKBRICKS

Feature	Airbnb	NoBroker	MagicBricks	BlockBricks (Proposed)
Platform Type	Centralized rental platform	Broker-free centralized app	Centralized property portal	Decentralized blockchain DApp
Fee Structure	10–15 % per booking	Subscription / service fee	Subscription / listing fee	≈ 1–2 % (gas + escrow)
Trust Model	Platform-mediated	Platform-based	Broker/platform-based	Immutable smart contracts, blockchain audit
Contract Handling	Digital, off-chain	Mostly offline	Offline / third-party docs	On-chain self-enforcing smart lease
Payment System	Airbnb wallet	Centralized gateway	Centralized	Hybrid escrow (Razorpay + on-chain log)
Data Storage	Airbnb servers	Company servers	Company servers	IPFS (docs), blockchain (hashes)
Dispute Resolution	Platform mediation	Company/legal	Broker/legal	Smart-contract rules + admin arbitration
Transparency	Moderate (platform logs)	Moderate	Low	High — immutable, verifiable records
Geographic Reach	Global	India-focused	India-focused	India-first, globally scalable
Scalability	High (centralized infra)	Limited	Limited	High — blockchain + Layer-2 support

2) *Transaction Latency*: Controlled simulations on Ganache were conducted to benchmark transaction times, with artificial delays added to emulate realistic network

Experimental results show that BlockBricks operates at transaction costs of approximately 1–2%, compared to 10–15% charged by centralized platforms like Airbnb and

Operation	Centralized Platform (12–18%)	BlockBricks (On-Chain + Razorpay)	Improvement (Approx.)
Deploy Listing	Platform-controlled (no user access)	180k Gas → 0.00054 ETH ≈ \$2.45 (₹216)	Full user ownership
Sign Lease	Offline / Legal Handling	200k Gas → 0.00060 ETH ≈ \$2.72 (₹239)	Automated Smart Lease
Rent Payment (e.g. \$1,000)	\$120–\$180 (₹10,575–₹15,862)	Gas: 120k Gas → 0.00036 ETH ≈ \$1.63 (₹144); Razorpay: 1.5–2% → \$15–\$20 (₹1,322–₹1,762); Total: \$16.63–\$21.63 (₹1,465–₹1,906)	≈ 82–91% cost reduction
Average Latency	~5 s	~15–20 s	Slightly higher

conditions. Measured latencies from Sepolia ranged between 12–18 seconds, while comparative data from Layer-2 networks such as Polygon (~2s) and Arbitrum (~4s) indicate the potential for near real-time processing. These metrics provide practical performance estimates without necessitating mainnet deployment.

3) *IPFS Retrieval Time*: Tests involving small document uploads (≤1 MB) through public IPFS gateways recorded retrieval times averaging 2–3 seconds, aligning with benchmarks from existing studies (1.8–3s). These results demonstrate that IPFS retrieval speed for lightweight property media and lease documents is comparable to traditional centralized storage, confirming its suitability for production-scale deployment within BlockBricks.

4) *Escrow Validation*: Razorpay’s sandbox APIs were utilized to simulate escrow lifecycle events. Each tenant payment generated a verifiable off-chain transaction ID, which was hashed using SHA-256 and mapped to smart contract logs, creating an immutable payment reference. Settlement timelines, verified against Razorpay’s documentation, confirmed T+2 day clearance, validating the feasibility of hybrid escrow within regulatory and technical constraints

V. DISCUSSION

BlockBricks demonstrates that blockchain technology can streamline rental agreements by reducing operational costs while enhancing transparency and trust. Smart contracts automatically enforce lease terms without intermediaries, and the hybrid escrow model integrated with Razorpay enables secure, compliant fiat transactions. Each lease, payment, and contract event is immutably recorded on Ethereum, minimizing disputes and preventing data tampering.

MagicBricks. This cost efficiency results from eliminating intermediaries and automating core processes through smart contracts. Decentralized document storage via IPFS ensures tamper-proof data management, while hybrid escrow provides verifiable linkage between blockchain records and real-world payments [5]–[8].

Key advantages include:

1. Automated lease management via self-executing smart contracts.
2. Decentralized storage of property and identity documents on IPFS.
3. Hybrid escrow integration linking on-chain transactions with verified fiat payments.

However, challenges remain regarding transaction latency, gas price volatility, and user onboarding. Future enhancements will explore Layer-2 scaling solutions such as Polygon and Arbitrum to reduce latency and incorporate privacy-preserving mechanisms like Decentralized Identity (DID) and Zero-Knowledge Proofs (ZKPs) for improved confidentiality.

Overall, BlockBricks offers a transparent, low-cost, and practical alternative to centralized rental systems, demonstrating the viability of blockchain-based property management in real-world use.

VI. CONCLUSION AND FUTURE WORK

This paper presented BlockBricks, a blockchain-based decentralized rental ecosystem that integrates Ethereum smart contracts, IPFS storage, and Razorpay escrow to automate lease management and facilitate transparent, low-cost transactions. Experimental evaluation demonstrates that BlockBricks significantly reduces brokerage fees while

maintaining acceptable transaction latency, establishing it as a practical alternative to centralized rental platforms.

A. Key Contributions

1. Elimination of intermediary commissions through automated smart contracts.
2. Integration of a hybrid escrow model with INR support for financial compliance.
3. Decentralized and immutable document storage via IPFS, ensuring transparency and tamper resistance.

B. Future Work

1. Layer-2 deployment (e.g., Polygon, Arbitrum) to further lower fees and improve scalability.
2. Adoption of multi-currency payments options for deposits and rent, global adaptation
3. Implement an advanced dispute arbitration system for unresolved conflicts.
4. Enhance analytics and dashboards for landlords to monitor multiple properties.
5. Cloud-native deployment (e.g., AWS, Azure, GCP) to ensure scalability, high availability, and seamless integration with enterprise-grade services.
6. Integration of decentralized identity (DID) and privacy-preserving mechanisms for secure user verification.

REFERENCES

- [1] J. Li et al., "Blockchain in the Built Environment," Automation in Construction, 2019.
- [2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
- [3] V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," 2014.
- [4] Z. Zheng et al., "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," IEEE Big Data, 2017.
- [5] J. Alvarenga et al., "Smart Contracts for Real Estate Rental Agreements," IEEE iThings, 2018.
- [6] C. Qi-Long et al., "Application of Blockchain in Rental Housing," J. Phys.: Conf. Ser., 2019.
- [7] D. Meijer and J. Ubacht, "Toward a Blockchain-Based Rental Market," ICE/ITMC, 2018.
- [8] LockChain Project, "A Decentralized Open Source Booking Ecosystem." Available: <https://lockchain.co>. Accessed: Sep 2025.
- [9] X. Xu et al., Architecture for Blockchain Applications, Springer, 2019.
- [10] E. B. Hamida et al., "Blockchain for Enterprise," IEEE CloudCom, 2017.
- [11] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the IoT," IEEE Access, 2016.
- [12] W. Li et al., "Towards Scalable and Private Industrial Blockchains," IEEE Trans. Ind. Informatics, 2018.
- [13] T. Puschmann, "Fintech," Bus. Inf. Syst. Eng., 2017.