

Article

Not peer-reviewed version

Analysis and Evaluation of a Blockchain-Based Framework for Decentralized Rental Agreements and Dispute Resolution

Muntasir Jaodun^{*} and [Khawla Bouafia](#)

Posted Date: 24 March 2025

doi: 10.20944/preprints202502.0872.v2

Keywords: blockchain; smart contract; decentralized finance; decentralized rental agreements; decentralized justice; arbitration mechanisms; blockchain efficiency; game theory; cross-jurisdictional applications



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Analysis and Evaluation of a Blockchain-Based Framework for Decentralized Rental Agreements and Dispute Resolution

Muntasir Jaodun ^{1,†,*}  and Khawla Bouafia ^{2,†}

¹ Eötvös Loránd University

² Eötvös Loránd University

* Correspondence: muntasir@student.elte.hu

† These authors contributed equally to this work.

Abstract: Blockchain technology has evolved beyond financial transactions to revolutionize trust systems. This paper presents a blockchain-based model for decentralized rental agreements and dispute resolution (DRADR). By leveraging smart contracts and implementing two distinct arbitration approaches, our model offers flexible solutions for rental agreement automation, transparency enhancement, and impartial dispute resolution. Our study provides a comprehensive technical analysis of both approaches through theoretical frameworks, smart contract implementation, game-theoretic modeling, and comparative evaluation across multiple legal jurisdictions. We explore the potential of blockchain technology to address long-standing challenges in traditional rental systems, such as power imbalances, inefficiencies, and legal disputes. Key contributions include the integration of decentralized and local justice systems, detailed game-theoretic analysis of strategic behaviors, and comparative insights into gas efficiency, economic viability, and jurisdictional adaptability across both arbitration approaches. This research paves the way for a more equitable and transparent rental market and contributes to the broader acceptance of blockchain-based solutions in everyday transactions.

Keywords: blockchain; smart contract; decentralized finance; decentralized rental agreements; decentralized justice; arbitration mechanisms; blockchain efficiency; game theory; cross-jurisdictional applications

1. Introduction

Blockchain, introduced with Bitcoin in 2008 [1], revolutionized digital transactions by eliminating central authority reliance. Ethereum's 2013 launch [2,3] expanded this with smart contracts, automating agreements based on predefined conditions. Meanwhile, the rental housing market serving 1.2 billion global renters [4] still operates through antiquated systems despite digital transformation opportunities [5].

Traditional rental agreements face critical challenges that blockchain could address [6,7] including Power Asymmetry, Information Asymmetry in property conditions and histories, Process Inefficiency from manual paperwork, Dispute Resolution Costs, and Jurisdictional Complexity in cross-border agreements [8]. These challenges create market inefficiencies that frequently disadvantage tenants [9].

To address these challenges, we explore four key research questions: (1) How blockchain integration affects rental process efficiency and fairness; (2) How blockchain-based systems alter game-theoretic dynamics between parties; (3) The extent blockchain can mitigate landlord-tenant power imbalances; and (4) How different arbitration approaches affect dispute resolution.

Our research objectives include developing a comprehensive blockchain-based model for decentralized rental agreements and dispute resolution (DRADR); implementing and evaluating two distinct arbitration approaches [10,11]; analyzing strategic implications through game theory [12]; evaluating

economic viability through gas analysis [13]; and providing actionable implementation insights for different jurisdictions [14].

The remainder of this paper is structured as follows: Section 2 presents background concepts and related work; Section 3 details the DRADR model design; Section 4 outlines our methodology; Section 5 provides implementation details; Section 6 presents results and discussion across case studies, game-theoretic analysis and technical evaluation; and Section 7 concludes with future research directions.

2. Background and Related Work

This section presents essential background concepts and synthesizes relevant prior research to establish the foundation for our DRADR model. We focus on blockchain applications in real estate, smart contracts for dispute resolution, and game theory in rental relations, and identify research gaps our work addresses.

2.1. Blockchain and Smart Contracts in Real Estate

Blockchain technology establishes trust in trustless environments through its distributed, immutable ledger using consensus mechanisms and cryptographic verification [1,15]. In real estate, Li et al. [6] identified three primary use cases across 57 publications: disintermediation, immutable record keeping, and trustless transactions, while noting standardization challenges. Abualhamayl et al. [7] highlighted blockchain's capacity to reduce fraud through transparent record-keeping, and Ahmad et al. [16] demonstrated an Ethereum-based property management platform that automates processes but relied on traditional enforcement for disputes.

Smart contracts enable automated enforcement without intermediaries [3], with Khan et al. [13] identifying key capabilities for rentals including automated payments and condition-based fund releases, while noting limitations in state channels, gas costs, and oracle reliability. Existing platforms like Rentberry [17] offer property listings and payment processing but lack robust dispute resolution mechanisms, while others such as Atlant and Beenest struggle with user adoption, workflow integration, and jurisdictional recognition [6,13].

2.2. Dispute Resolution Mechanisms

Smart contract dispute resolution requires balancing technological automation with human judgment, as Schmitz and Rule [18] emphasized for complex disputes requiring subjective evaluation, while Kadioglu Kumtepe [11] highlighted challenges in jurisdictional enforcement without specific recognition of smart contract validity. Decentralized justice systems offer blockchain-based alternatives to traditional legal proceedings, with Kleros implementing a jury approach where token holders serve as jurors incentivized through a Schelling point mechanism that rewards majority voters [10]. LEChain [19] addresses evidence integrity in blockchain arbitration, while Koulu [20] proposed hybrid approaches combining automation with human oversight to resolve tensions between smart contracts and consumer protection regulations. Allen et al. [21] further analyzed how decentralized justice systems establish legitimacy without traditional institutional backing, contributing to the developing framework for selecting appropriate resolution mechanisms based on dispute characteristics.

2.3. Game Theory in Rental Relations

Game theory provides a valuable framework for analyzing landlord-tenant interactions and designing effective incentive mechanisms. Dong and Dong [12] conducted a bibliometric analysis of game theory applications in resource allocation contexts, establishing methodological approaches relevant to resource-sharing relationships like those between landlords and tenants. Desmond and Wilmers [9] empirically examined exploitation in rental markets, finding that landlords in disadvantaged neighborhoods earned significantly higher returns than those in affluent areas—revealing strategic

behaviors that emerge in rental markets with significant power imbalances. Oliveira [22] specifically applied game theory to smart contract disputes, modeling how different arbitration mechanisms affect participant behavior. Her research demonstrated that well-designed arbitration systems can create Nash equilibria encouraging honest behavior and fair dispute resolution.

2.4. Research Gaps

Our comprehensive review of existing literature reveals several significant research gaps that our DRADR model aims to address:

First, while separate research streams exist on blockchain for real estate, smart contracts for dispute resolution, and game theory in rental relations, few studies integrate these approaches into comprehensive rental management systems. Li et al. [6] identified this lack of integration as a key limitation in current blockchain real estate research.

Second, most existing research either presents theoretical frameworks without implementation details [18] or implements specific components without addressing the complete rental lifecycle [16]. This leaves practitioners without clear guidance on how to implement comprehensive blockchain-based rental systems.

Third, despite the existence of multiple arbitration approaches, there is a notable absence of comparative studies evaluating different arbitration mechanisms within rental contexts. This gap limits practitioners' ability to select the most appropriate dispute-resolution methods based on specific requirements.

Fourth, few studies have rigorously assessed gas costs, transaction overheads, and the overall economic viability of blockchain-based rental solutions. Khan et al. [13] highlighted this gap as a barrier to practical implementation.

Finally, limited research exists on how blockchain-based rental systems might function across different legal frameworks, particularly regarding the enforceability of smart contract provisions and arbitration decisions across jurisdictions [11].

3. DRADR Model Design

The conceptual framework, system architecture, and arbitration approaches of our Decentralized Rental Agreement and Dispute Resolution (DRADR) model are presented in this section. We provide a detailed overview of the design principles, technical components, and operational processes that form the foundation of our implementation.

3.1. Conceptual Framework

The DRADR model introduces a comprehensive framework for managing the entire rental lifecycle on blockchain. Building upon the research gaps identified in Section 2, DRADR integrates smart contract automation with decentralized justice systems to create a more efficient, transparent, and equitable rental ecosystem.

At its conceptual core, DRADR addresses the fundamental challenges in traditional rental systems through four key design principles:

1. **Decentralization of Trust:** DRADR eliminates reliance on centralized authorities for agreement enforcement and dispute resolution. Instead, it leverages blockchain's distributed ledger to create transparent, tamper-proof records and smart contracts for automated enforcement. As Bouafia et al. [23] note, this approach significantly reduces counterparty risk while improving transparency.
2. **Programmable Agreements:** The model implements rental terms as programmable smart contracts that automatically execute predefined conditions. This approach builds on Khan et al.'s [13] work on smart contract applications, extending it to handle the unique requirements of rental agreements including payment scheduling, security deposit management, and condition-based fund releases.

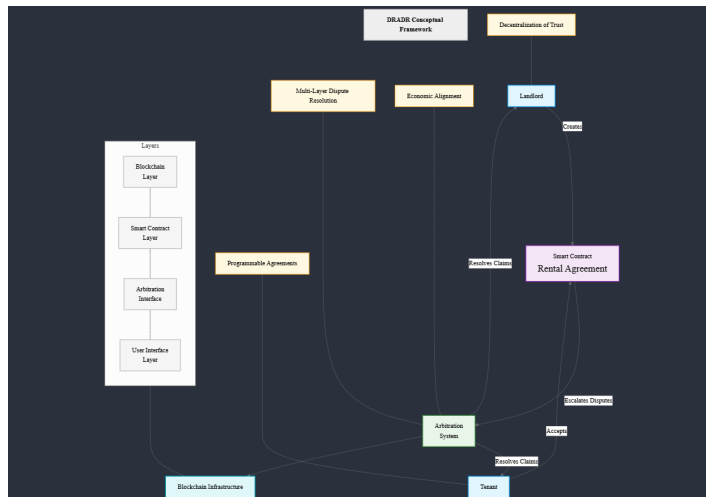


Figure 1. DRADR Conceptual Framework - showing participants (landlord, tenant, arbitrators), components (smart contracts, arbitration systems), and their relationships

3. **Multi-Layer Dispute Resolution:** DRADR incorporates a tiered dispute resolution system that attempts to resolve conflicts through automated rules first, then escalates to arbitration only when necessary. This design draws from Schmitz and Rule’s [18] research on online dispute resolution, adapted specifically for blockchain-based rental contexts.
4. **Economic Alignment:** The model creates aligned economic incentives for all participants through carefully designed token mechanisms and fee structures. This approach incorporates game-theoretic principles explored by Dong and Dong [12], optimizing for honest behavior and fair outcomes.

The DRADR model consists of five primary functional layers that separate concerns while maintaining interoperability:

1. **Blockchain Infrastructure Layer:** The foundational distributed ledger technology that provides immutable record-keeping, transaction validation, and smart contract execution. The model is designed to be blockchain-agnostic but is primarily implemented on Ethereum due to its robust smart contract capabilities.
2. **Smart Contract Layer:** The core programmable agreements that encode rental terms, payment conditions, and dispute handling logic. This layer implements automatic rent collection, security deposit management, and basic dispute handling.
3. **Arbitration Interface Layer:** A middleware layer that enables communication between the smart contract layer and different arbitration systems. This interface standardizes dispute data formats and enforcement mechanisms while allowing flexibility in arbitration approach.
4. **Arbitration System Layer:** The mechanisms for resolving disputes that cannot be handled automatically by smart contracts. This layer includes multiple implementation options, allowing for different approaches based on specific requirements.
5. **User Interface Layer:** The front-end applications that enable landlords and tenants to interact with the system, including agreement creation, rent payment, maintenance requests, and dispute filing.

This layered architecture provides several advantages: clear separation of concerns allowing independent optimization of each layer, pluggable arbitration systems that form a key innovation of this research, and flexibility to adapt to different blockchain platforms and regulatory environments.

3.2. System Architecture

The DRADR system architecture translates the conceptual framework into a concrete technical implementation. The architecture follows a layered design pattern with clear separation of concerns, as illustrated in Figure 2.

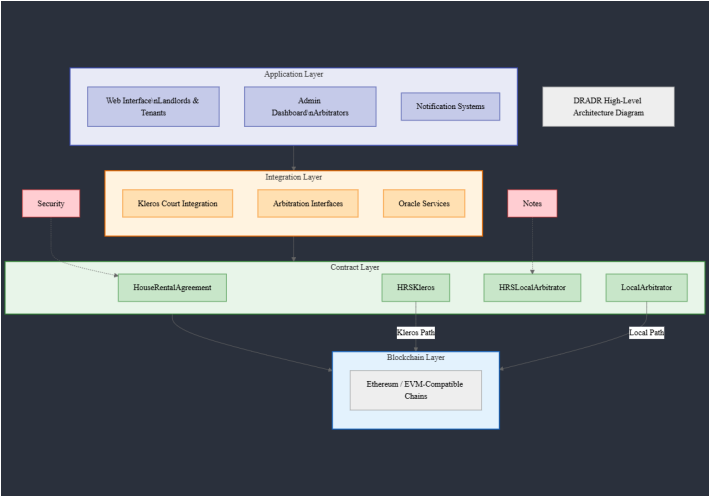


Figure 2. DRADR High-Level Architecture Diagram - showing main components and their interactions

The architecture employs a four-layer structure designed for comprehensive rental management: the foundational *Blockchain Layer* utilizes Ethereum (or compatible EVM chains) for core services including transaction processing and state persistence; the *Contract Layer* implements business logic through smart contracts—HouseRentalAgreement manages rental relationships, with specialized implementations HRSKleros and HRSLocalArbitrator handling different dispute resolution approaches, using gas-efficient bit-flag state management and strict role-based access control; the *Integration Layer* facilitates communication with external systems through Kleros Court components and oracle services for off-chain data while implementing withdrawal patterns to prevent reentrancy vulnerabilities [24]; and the *Application Layer* provides user interfaces for landlords, tenants, and arbitrators with notification systems. This architecture ensures system integrity through state machine patterns that restrict function access based on sender identity and current contract state, preventing invalid state combinations while enabling modular arbitration pathways, contract upgradeability, and cross-chain compatibility—all critical components ensuring operations occur in proper sequence throughout the rental agreement lifecycle.

3.3. Arbitration Approaches

A key innovation of the DRADR model is its support for multiple arbitration approaches, enabling deployment flexibility based on specific contextual requirements. The model implements two distinct arbitration systems: Kleros decentralized arbitration and our novel LocalArbitrator system.

3.3.1. Kleros Decentralized Arbitration

The Kleros integration leverages an established decentralized justice protocol built on Ethereum. As described by Zhuk [10], Kleros implements a jury-based approach where token holders can serve as jurors and are randomly selected and incentivized to vote honestly through a crypto-economic mechanism.

The DRADR-Kleros integration offers several key features:

1. **Decentralized Juror Selection:** Disputes are adjudicated by a panel of jurors drawn from a global pool, providing independence from either party in the rental agreement. This addresses bias concerns identified in traditional arbitration systems [11].

2. **Specialized Rental Court:** DRADR utilizes Kleros' subcourt functionality to create a specialized rental dispute court with jurors who have relevant expertise. This specialization improves decision quality for the unique aspects of rental disputes.
3. **Cryptoeconomic Security:** The integration leverages Kleros' token-based incentive system requiring jurors to stake tokens as a guarantee of honest behavior. This mechanism creates economic incentives for fair rulings.
4. **Appeal Mechanism:** Complex or high-value disputes can leverage Kleros' appeal system, where increasingly larger juror panels review the case with higher economic stakes.
5. **Evidence Management:** The implementation includes standardized evidence submission processes based on the LEChain framework [19], ensuring evidence integrity and accessibility.

In the Kleros implementation, jurors must stake Pinakion (PNK) tokens to be eligible for selection, with selection probability proportional to staked tokens. Jurors who vote with the majority receive a portion of the arbitration fee plus the tokens of jurors who voted against the majority, creating a Schelling point mechanism incentivizing honest voting.

3.3.2. LocalArbitrator System

The LocalArbitrator system is a novel approach designed specifically for the DRADR model. This approach prioritizes efficiency, cost-effectiveness, and jurisdiction-specific adaptability over maximum decentralization.

Key features of the LocalArbitrator system include:

1. **Designated Arbitrator Model:** Rather than a jury system, the LocalArbitrator utilizes a designated arbitrator or arbitration service with specific domain expertise. This model draws on Koulu's [20] research on hybrid arbitration systems.
2. **Customizable Fee Structure:** The implementation includes configurable arbitration fees and refund mechanisms, allowing for context-appropriate economic parameters. This addresses economic viability concerns identified by Khan et al. [13].
3. **Streamlined Evidence Handling:** The system implements simplified evidence submission and evaluation processes optimized for common rental disputes.
4. **Jurisdiction Adaptability:** The LocalArbitrator can be configured to comply with specific jurisdictional requirements, addressing cross-jurisdictional concerns raised by Kadioglu Kumtepe [11].
5. **Lower Computational Overhead:** The system is designed for gas efficiency, with optimized smart contract interactions that minimize blockchain transaction costs.

In the LocalArbitrator implementation, the arbitrator is a predefined entity with rental dispute expertise. Arbitrators receive the arbitration fee minus any refund allocated to the winning party and build reputation through transparent rulings. While this approach trades some decentralization for gains in efficiency and adaptability, it maintains transparency and accountability through immutable ruling records and performance metrics.

4. Methodology

This section describes our research methodology for evaluating the DRADR model. We employ a multi-faceted approach combining case study analysis across multiple jurisdictions, game-theoretic modeling, and comparative technical evaluation to provide a comprehensive assessment of the model's effectiveness and implications.

4.1. Case Study Analysis Approach

To evaluate the potential real-world impacts of the DRADR model, we applied it to rental dispute cases from multiple jurisdictions. This approach allows us to explore how blockchain-based rental

agreements might address scenarios that have proven challenging for traditional systems while also testing cross-jurisdictional applicability.

4.1.1. Case Selection and Analysis Framework

We selected cases based on legal significance, dispute diversity, jurisdictional variety, and documentation quality. Our sample included three cases from common law jurisdictions (UK and Australia) and three from civil law jurisdictions (France, Germany, and Spain). Our structured analysis framework compared traditional proceedings with DRADR-based resolution across six dimensions: *Efficiency*, *Fairness*, *Dispute Prevention*, *Transparency*, *Record-keeping*, and *Jurisdictional Compliance*. Each dimension was evaluated using specific criteria (e.g., time to resolution, access regardless of resources, agreement clarity) on a consistent 5-point scale from "Very Poor" to "Excellent," enabling standardized comparison across cases, jurisdictions, and resolution methods.

4.1.2. Jurisdictional Adaptation Analysis

For civil law jurisdictions, we conducted additional analysis on how the DRADR model would need to be adapted to comply with specific legal requirements. This included identifying mandatory provisions, consumer protections, evidence requirements, and enforcement mechanisms specific to each jurisdiction. For each jurisdiction, we examined specific legal requirements and assessed what modifications would be necessary to make the DRADR model compliant. We then assigned feasibility scores (1-5) indicating the practicality of implementing these adaptations.

4.2. Game-Theoretic Model

To understand the strategic implications of the DRADR model, we developed game-theoretic models for both traditional rental agreements and the DRADR implementations. This approach allows us to analyze how different incentive structures affect participant behavior and outcomes.

4.2.1. Model Formulation

We formulated the rental relationship as a sequential game with incomplete information, where landlords and tenants make decisions based on their understanding of the agreement and expectations about the other party's behavior.

For traditional rental agreements, we modeled the game structure as an 8-stage sequential process:

1. Landlord sets rental terms
2. Tenant accepts or rejects terms
3. Tenant pays security deposit
4. Rental period occurs
5. Rental period ends
6. Landlord makes security deposit return decision
7. Tenant responds (accept or dispute)
8. If disputed, resolution occurs through traditional legal system

For the DRADR model, we modified the game to incorporate blockchain-based dispute resolution as a 9-stage process, with the final stage involving the arbitration system reviewing evidence and ruling.

4.2.2. Payoff Derivation and Arbitrator Behavior

We extended the game model to explicitly include the arbitrator as a strategic actor. Payoffs were derived from: *Financial Transfers* (direct monetary values exchanged between parties); *Transaction Costs* (gas fees and arbitration costs); *Time Costs* (estimated costs of delays in resolution); and *Reputation Costs* (estimated impact on future rental opportunities). For the Kleros implementation, arbitrator payoffs were calculated based on the token economics model, where jurors stake tokens and receive rewards

or penalties based on majority alignment. For the LocalArbitrator implementation, payoffs included the arbitration fee minus the effort cost plus a reputation component affecting future opportunities.

4.2.3. Equilibrium Analysis and Sensitivity Testing

We analyzed the Nash equilibria for each game to determine stable strategy profiles, specifically looking for subgame perfect equilibria using backward induction. To incorporate uncertainty in arbitration outcomes, we modeled arbitration as a probabilistic process with outcomes dependent on the merits of each party's claim. To test the robustness of our findings, we conducted sensitivity analysis by varying key parameters: *Arbitration costs* (0.5% to 30% of security deposit); *Accuracy of arbitration* (60% to 99% probability of correct decisions); *Time value* (0.5% to 10% of disputed value per month); *Reputation effects* (0 to 200% of deposit value); and *Jurisdictional variables* (varying court costs and tenant protection levels). This approach provided a comprehensive understanding of how the DRADR model performs across a range of conditions.

4.3. Technical Evaluation Methods

To evaluate the economic efficiency and technical performance of each arbitration approach, we conducted systematic gas cost analysis and feature comparison.

4.3.1. Gas Cost Analysis

We used the Hardhat development environment with Solidity compiler version 0.8.18 and optimization level 200 to measure gas consumption for all contract functions. For each function, we conducted multiple test runs with standardized inputs representing realistic usage scenarios, recording both average and standard deviation of gas consumption. Beyond individual function costs, we defined typical function call sequences representing complete scenarios (e.g., full rental cycle with dispute, early termination) and summed the associated gas costs to estimate total lifecycle expenses. To ensure fair comparison, we used equivalent input parameters for both contracts and confirmed that each function execution produced the correct state changes.

4.3.2. Feature Comparison Framework

To evaluate non-economic aspects, we developed a feature comparison framework with six key dimensions: *Decentralization* (degree of trust minimization and resistance to censorship); *Accessibility* (barriers to entry for different participants); *Scalability* (ability to handle increasing numbers of disputes); *Flexibility* (adaptability to different dispute types and jurisdictions); *User Experience* (complexity and usability for non-technical users); and *Legal Integration* (compatibility with existing legal frameworks). For each dimension, we defined specific measurable criteria and evaluation methods. Where possible, quantitative measures were used; otherwise, qualitative assessments were made using a consistent 5-point scale with detailed criteria for each level.

4.3.3. Protocol Workflow Analysis

We systematically analyzed the complete protocol workflows for both arbitration approaches, identifying: *Payment Flows* (tracing the complete path of funds from initial security deposit to final settlement); *Signing Requirements* (documenting operations requiring cryptographic signatures); and *Entry Points* (mapping all functions that external actors can call). This analysis provided a comprehensive picture of operational workflows, enabling direct comparison and identification of potential bottlenecks or security concerns.

4.4. Cross-Jurisdictional Analysis Framework

Our cross-jurisdictional analysis methodology examined how the DRADR model would need to be adapted for different legal systems and regulatory requirements.

4.4.1. Legal System Classification and Regulatory Compliance Assessment

We categorized jurisdictions into common law systems (UK, Australia) and civil law systems (France, Germany, Spain), recognizing the fundamental differences in how these systems approach contracts, dispute resolution, and enforcement. For each jurisdiction, we evaluated: *Mandatory Provisions* (legal requirements that cannot be waived or modified); *Consumer Protection Measures* (specific protections for tenants as consumers); *Procedural Requirements* (formal processes required for validity and enforcement); and *Evidence Standards* (requirements for document format, authentication, and admissibility).

4.4.2. Implementation Adaptation Analysis

Based on the regulatory assessment, we identified specific smart contract modifications required for each jurisdiction. We then assessed the technical feasibility of these adaptations and their impact on the core DRADR functionality. This methodological approach enabled us to evaluate not just whether the DRADR model could work in different jurisdictions, but what specific adaptations would be necessary and whether these adaptations would compromise the model's benefits.

By combining these methodological approaches—case studies across multiple jurisdictions, game theory with arbitrator modeling, comparative technical evaluation, and cross-jurisdictional analysis—we aimed to provide a holistic assessment of the DRADR model and its two arbitration implementations.

5. Implementation

This section details the technical implementation of the DRADR model, focusing on the core rental agreement contract, arbitration implementations, and security considerations. The implementation translates the conceptual design into practical smart contracts deployed on Ethereum.

5.1. Core Rental Agreement Contract

The core rental agreement contract forms the foundation of the DRADR system, managing the fundamental rental relationship between landlord and tenant. Implemented in Solidity version 0.8.18, the contract leverages built-in overflow/underflow protection while maintaining compatibility with current development tools.

5.1.1. Data Structures and State Management

The contract uses optimized data structures to efficiently represent the rental relationship. Key structures include `MonthlyRent` for tracking rent payments, `MaintenanceRequest` for recording property issues, and `Evidence` for storing dispute documentation. A sample structure implementation is shown below:

```
struct MonthlyRent {
    uint256 baseRent;
    uint256 utilities;
    uint256 due;
    bool paid;
    uint256 paymentDate;
    uint256 dueDate;
    uint256 lateFee;
}
```

For efficient state management, the contract uses a bit-flag approach that packs multiple Boolean states into a single `uint8` variable, which reduces storage operations, significantly reducing gas costs.

5.1.2. Access Control and Lifecycle Management

The contract implements strict access control using modifiers that restrict function access based on sender identity and contract state:

```
modifier onlyLandlord() {
    require(msg.sender == landlord, "Only landlord can call");
    _;
}

modifier whenActive() {
    require(isStateActive(), "Contract is not active");
    _;
}
```

These modifiers ensure that functions can only be called by authorized parties in appropriate contract states, preventing invalid state transitions and unauthorized access.

For financial operations, the contract uses a withdrawal pattern rather than direct transfers to prevent reentrancy vulnerabilities:

```
function withdraw() external nonReentrant {
    uint256 amount = pendingWithdrawals[msg.sender];
    require(amount > 0, "No funds to withdraw");

    // Effects: update state before external call
    pendingWithdrawals[msg.sender] = 0;

    // Interactions: perform external call last
    (bool success, ) = payable(msg.sender).call{value: amount}("");
    require(success, "Transfer failed");
}
```

The contract also implements practical features like partial rent payments, maintenance request tracking, and automated late fee calculation to cover essential rental operations while maintaining security and efficiency.

5.2. Arbitration Implementations

The DRADR model includes two distinct arbitration implementations that extend the core rental agreement contract with different dispute resolution approaches.

5.2.1. Kleros Integration

The Kleros integration extends the core rental agreement with specialized functionality for connecting to the Kleros arbitration system. This implementation follows the Kleros arbitrable contract standard:

```
interface IArbitrable {
    function rule(uint256 _disputeID, uint256 _ruling) external;
}

interface IArbitrator {
    function createDispute(uint256 _choices, bytes calldata _extraData)
        external payable returns (uint256 disputeID);
}
```

```

    function arbitrationCost(bytes calldata _extraData)
        external view returns (uint256);
}

```

The implementation manages the dispute creation process when landlord and tenant cannot agree on settlement terms:

```

function rejectRenterCounterEstimate() external onlyLandlord whenActive {
    require(isStateActive(5), "Counter estimate not set");

    // Get arbitration cost from Kleros
    arbitrationCost = kleros.arbitrationCost("");
    require(currentSecurityDeposit >= arbitrationCost,
        "Insufficient funds for arbitration");

    // Deduct arbitration cost from security deposit
    currentSecurityDeposit -= arbitrationCost;

    // Create dispute in Kleros
    disputeID = kleros.createDispute{value: arbitrationCost}(2, "");
    setStateActive(2, true); // Set dispute flag

    emit DisputeRaised(disputeID);
    emit CounterEstimateRejected();
}

```

The contract implements standardized evidence submission and handles callbacks from Kleros when a ruling is made. The implementation was validated against the Kleros test environment to ensure compatibility with their arbitration protocol.

5.2.2. LocalArbitrator System

The LocalArbitrator implementation provides an alternative arbitration approach that prioritizes efficiency and customizability. Unlike the Kleros implementation, which relies on an external arbitration system, the LocalArbitrator is a purpose-built solution for rental disputes.

The LocalArbitrator contract itself manages arbitrator authentication and dispute tracking:

```

contract LocalArbitrator is ReentrancyGuard {
    address public immutable arbitrator;
    uint256 public disputeCount;
    uint256 public arbitrationFee;
    uint256 public refundPercentage;

    mapping(uint256 => Dispute) public disputes;
    mapping(address => bool) public authorizedContracts;

    struct Dispute {
        address rentalContract;
        uint256 landlordEstimate;
        uint256 renterCounterEstimate;
        bool isResolved;
    }
}

```

```

        uint256 ruling;
        uint256 creationTime;
        uint256 resolutionTime;
        string evidenceURI;
    }
}

```

A unique feature of the LocalArbitrator is the ability to refund a portion of the arbitration fee to the winning party:

```

function resolveDispute(uint256 _disputeId, uint256 _ruling)
    external onlyArbitrator {
    require(_disputeId > 0 && _disputeId <= disputeCount,
        "Invalid dispute ID");
    require(!disputes[_disputeId].isResolved,
        "Dispute already resolved");
    require(_ruling == 1 || _ruling == 2, "Invalid ruling");

    Dispute storage dispute = disputes[_disputeId];
    dispute.isResolved = true;
    dispute.ruling = _ruling;
    dispute.resolutionTime = block.timestamp;

    // Calculate refund amount
    uint256 refundAmount = arbitrationFee * refundPercentage / 100;

    // Call the rental contract to execute the ruling with refund
    // Implementation details omitted for brevity
}

```

The LocalArbitrator implementation was designed with gas efficiency as a primary concern, reducing cross-contract calls and storage operations to minimize transaction costs.

5.3. Security Patterns and Gas Optimization

The contracts implement robust security through access control modifiers for authorization, the Check-Effects-Interactions pattern and OpenZeppelin's ReentrancyGuard to prevent reentrancy attacks, Solidity 0.8.x's built-in overflow/underflow protection, secure state transitions with validated functions, and emergency pause capabilities for vulnerability response. Gas consumption is minimized via storage optimization techniques including bit packing and optimized struct layouts, prioritizing cheaper reads over writes, arranging require statements by computational cost, implementing batch processing to reduce transaction counts, and using events for off-chain data tracking rather than storing historical information on-chain.

6. Results and Discussion

This section presents the findings from our evaluation of the DRADR model and its two arbitration implementations, followed by analysis of their broader implications. We organize our results according to our research questions, beginning with the case study analysis and proceeding to the game-theoretic and technical evaluations.

6.1. Case Study Analysis Results

Our analysis of rental dispute cases across multiple jurisdictions revealed significant differences in how the DRADR model would perform compared to traditional legal proceedings. We present the results organized by jurisdiction type and key dimensions.

6.1.1. Common Law Jurisdictions

The application of the DRADR model to common law cases demonstrated substantial potential improvements in efficiency and transparency while maintaining or improving fairness.

Table 1. Comparative Analysis of DRADR Model in Common Law Cases

Dimension	Traditional Approach	DRADR-Kleros	DRADR-LocalArbitrator
Efficiency	Time to Resolution: 6-18 months Procedural Steps: 12-20	Time to Resolution: 1-3 weeks Procedural Steps: 5-7	Time to Resolution: 3-10 days Procedural Steps: 4-6
Fairness	Access Score: 2.3 Legal Consistency: 4.2	Access Score: 4.1 Legal Consistency: 3.9	Access Score: 3.8 Legal Consistency: 3.7
Dispute Prevention	Term Clarity: 3.1 Information Access: 2.4	Term Clarity: 4.7 Information Access: 4.8	Term Clarity: 4.7 Information Access: 4.8
Transparency	Process Visibility: 2.7	Process Visibility: 4.6	Process Visibility: 4.3
Jurisdictional Compliance	Legal Compatibility: 5.0	Legal Compatibility: 3.9	Legal Compatibility: 4.2

Note: All scores are on a scale of 1-5, with 5 being the highest.

In the Moorjani v. Durban Estates case, the DRADR model would have significantly reduced the time to resolution from over two years to approximately two weeks using the Kleros implementation or one week using the LocalArbitrator implementation.

The Arnold v. Britton case highlighted how the DRADR model could enhance term clarity through smart contract implementation, potentially preventing the dispute entirely by making service charge calculations transparent and immutable. In this case, both DRADR implementations scored 4.7/5 for term clarity compared to 2.9/5 for the traditional approach.

However, the analysis also revealed challenges in jurisdictional compliance, with both DRADR implementations scoring lower than traditional approaches on legal compatibility, procedural alignment, and enforcement metrics. This was particularly evident in cases that relied on judicial discretion or interpretation of ambiguous terms, where traditional courts have established practices not easily replicated in code.

6.1.2. Civil Law Jurisdictions

The application of the DRADR model to civil law cases revealed distinct patterns compared to common law jurisdictions, reflecting the different legal frameworks and procedural requirements.

The French case highlighted a significant challenge: the mandatory requirement for state-certified condition assessment (état des lieux) documents at the beginning and end of tenancies. The DRADR model would need specific adaptations to incorporate these legally required documents, resulting in lower jurisdictional compliance scores (3.5/5 for Kleros and 3.9/5 for LocalArbitrator compared to 5.0/5 for traditional approaches).

The German case demonstrated how the DRADR model could enhance transparency in rent calculations under the Mietpreisbremse legislation. The smart contract implementation would make

rent calculations fully transparent and verifiable, scoring 4.6/5 for process visibility compared to 3.1/5 for traditional approaches. However, the strict German tenant protection laws would require significant adaptations to the smart contract implementation, particularly regarding security deposit handling.

6.1.3. Jurisdictional Adaptation Requirements

Our analysis identified specific adaptation requirements for the DRADR model to function effectively across different legal systems. Table 2 summarizes these requirements by jurisdiction.

Table 2. DRADR Adaptation Requirements by Jurisdiction

Jurisdiction	Legal System	Key Adaptation Requirements	Feasibility Score
UK	Common Law	Integration with government-approved deposit protection schemes; Compatibility with Section 8 and Section 21 notice procedures; Alignment with Landlord and Tenant Act 1985 repair obligations	4.2
Australia	Common Law	Compliance with state-specific Residential Tenancies Acts; Integration with tribunal procedures (e.g., QCAT, VCAT); Adaptation to mandatory condition reports requirements	3.9
France	Civil Law	Integration of état des lieux requirement; Compliance with Loi ALUR tenant protection measures; Implementation of mandatory notice periods and procedures	3.5
Germany	Civil Law	Implementation of Kaution (deposit) special account with interest; Compliance with Mietpreisbremse rent control calculations; Integration with Mietspiegel (rent index) reference system	3.3
Spain	Civil Law	Adaptation to Law of Urban Leases (LAU) requirements; Implementation of mandatory registration procedures; Alignment with specific notice periods and tenant protections	3.7

The feasibility scores indicate that while adaptation to various jurisdictions is possible, it requires significant customization of the smart contract implementation. Common law jurisdictions generally showed higher feasibility scores, possibly due to their greater flexibility in contract enforcement mechanisms.

6.2. Game-Theoretic Analysis Results

Our game-theoretic analysis revealed significant differences in strategic behavior and equilibrium outcomes between traditional rental systems and the DRADR implementations.

6.2.1. Traditional Rental Game Equilibrium

Analysis of the traditional rental agreement game yielded a subgame perfect Nash equilibrium where landlords have strong incentives to withhold security deposits partially or fully, and tenants have limited incentives to dispute these decisions due to high dispute costs.

Table 3. Payoff Matrix for Traditional Rental System

Landlord ↓ / Tenant →	Accept (A)	Dispute (D)
Return Full Deposit (R)	(0, 0)	(-1, -1)
Withhold Partial Deposit (P)	(1, -1)	(-2, -2)
Withhold Full Deposit (W)	(2, -2)	(-3, -3)

Note: Payoffs represented as (Landlord, Tenant) in units of security deposit value. Negative values incorporate dispute costs and time value.

The dominant strategy for landlords is to withhold the full deposit (W) if they expect tenants to accept the decision (A), yielding a payoff of (2, -2). Tenants have rational incentives to accept even unfair deposit withholding if dispute costs exceed the disputed amount, which is common in traditional systems. Sensitivity analysis revealed that this equilibrium is robust across a wide range of parameter values, changing only when dispute costs become negligible or when strong reputation effects are introduced.

6.2.2. DRADR Model Game Equilibrium

The DRADR model significantly alters the game's equilibrium by reducing dispute costs and increasing transparency. Both implementations show similar strategic dynamics with some important differences.

Table 4. Payoff Matrix for DRADR-Kleros System

Landlord ↓ / Tenant →	Accept (A)	Dispute (D)
Return Full Deposit (R)	(0, 0)	(0, -0.2)
Claim Damages (C) - Honest	(D, -D)	(D-0.2, -D-0.1)
Claim Damages (C) - Dishonest	(X, -X)	(-0.3, -0.1)

Note: D represents actual damages ($0 \leq D \leq 1$), X represents dishonest claim amount ($D < X \leq 1$). Values incorporate gas costs and arbitration fees.

Table 5. Payoff Matrix for DRADR-LocalArbitrator System

Landlord ↓ / Tenant →	Accept (A)	Dispute (D)
Return Full Deposit (R)	(0, 0)	(0, -0.15)
Claim Damages (C) - Honest	(D, -D)	(D-0.15, -D-0.05)
Claim Damages (C) - Dishonest	(X, -X)	(-0.25, -0.05)

Note: D represents actual damages ($0 \leq D \leq 1$), X represents dishonest claim amount ($D < X \leq 1$). Values incorporate gas costs and arbitration fees.

The equilibrium analysis reveals that in both DRADR implementations, landlords have incentives to make honest damage claims rather than dishonest ones, as disputed dishonest claims lead to negative payoffs while tenants have lower barriers to disputing claims they perceive as unfair, as dispute costs are significantly reduced compared to traditional systems. The LocalArbitrator implementation shows slightly stronger incentives for honest behavior due to its lower dispute costs and refund mechanism for the winning party. Both implementations create a more balanced strategic environment compared to the traditional system, reducing the landlord's advantage in security deposit disputes.

6.2.3. Extended Game Model with Arbitrator Analysis

The extended game model incorporating arbitrator behavior revealed important strategic dynamics in the arbitration process itself.

Table 6. Arbitrator Strategy Effects in Kleros Implementation

Arbitrator Effort	Correct Ruling Probability	Juror Reward (Expected)	System Accuracy
Low	0.65	-0.05 (net loss)	0.70
Medium	0.85	0.07 (net gain)	0.85
High	0.95	0.09 (net gain)	0.92

Note: System accuracy includes the baseline probability of correct case outcome accounting for both juror decision accuracy and case clarity.

In the Kleros implementation, jurors have financial incentives to exert at least medium effort, as low effort creates a risk of disagreeing with the majority and losing staked tokens. The cryptoeconomic mechanism effectively incentivizes accurate rulings.

Table 7. Arbitrator Strategy Effects in LocalArbitrator Implementation

Arbitrator Effort	Correct Ruling Probability	Fee - Effort Cost	Reputation Effect	System Accuracy
Low	0.70	0.08	-0.10	0.70
Medium	0.90	0.05	0.05	0.90
High	0.98	0.02	0.10	0.98

Note: System accuracy represents the probability of the arbitrator rendering the objectively correct decision based on the evidence.

In the LocalArbitrator implementation, while the immediate financial incentive favors low effort (highest net fee), the reputation effect creates countervailing incentives for high-quality rulings.

The equilibrium analysis indicates that the Kleros implementation should achieve approximately 85-92% accuracy in decisions, while the LocalArbitrator implementation may achieve 90-98% accuracy in settings where reputation effects are significant. Both represent substantial improvements over traditional systems, where decision accuracy is often compromised by resource disparities between parties.

6.3. Technical Performance Comparison

6.3.1. Gas Efficiency Analysis

Our analysis of gas costs for both implementations revealed significant differences in efficiency, with the LocalArbitrator generally providing better gas economy for dispute-related operations.

Table 8. Detailed Gas Cost Comparison Between Implementations

Operation	Kleros	LocalArbitrator	% Difference
Contract Deployment	3,800,000	3,500,000	7.9%
setAgreementTerms	46,200	72,400	-56.7%
paySecurityDeposit	43,800	64,800	-47.9%
activateContract	44,200	N/A	N/A
recordUtilities	42,800	42,800	0.0%
payRent	72,500	78,500	-8.3%
payPartialRent	68,200	68,200	0.0%
skipRentPayment	56,400	56,400	0.0%
setDamageEstimate	45,100	45,100	0.0%
acceptLandlordEstimate	52,800	82,300	-55.9%
rejectLandlordEstimate	28,600	28,600	0.0%
setRenterCounterEstimate	45,100	45,100	0.0%
acceptRenterCounterEstimate	52,800	82,300	-55.9%
rejectRenterCounterEstimate	108,700	78,500	27.8%
submitEvidence	66,800	45,000	32.6%
arbitration ruling	84,200	93,200	-10.7%
withdraw	35,700	35,700	0.0%

Note: Gas costs measured using Hardhat with Solidity compiler version 0.8.18 and optimization level 200.

For dispute-related operations (rejectRenterCounterEstimate and submitEvidence), the LocalArbitrator implementation showed gas savings of 27.8% and 32.6% respectively. However, for several basic operations, the LocalArbitrator implementation had higher gas costs due to its combined deposit payment and activation functionality.

The total gas costs for typical scenarios revealed the practical implications of these differences:

Table 9. Scenario-Based Gas Cost Analysis

Scenario	Kleros	LocalArbitrator	Difference
12-month rental, no dispute	1,687,100	1,716,600	-29,500
12-month rental with dispute	2,013,700	1,961,900	51,800
Early termination after 3 months	690,800	720,300	-29,500
12-month rental with partial payments	1,892,700	1,922,200	-29,500

These results indicate that the LocalArbitrator implementation offers gas efficiency advantages specifically in dispute scenarios, which aligns with its design goal of providing a streamlined arbitration process. The Kleros implementation, while more expensive for dispute resolution, offers better gas efficiency for standard operations in non-dispute scenarios.

6.3.2. Feature Comparison

Our feature comparison analysis identified important tradeoffs between the two arbitration approaches across six key dimensions.

Table 10. Feature Comparison of Arbitration Approaches

Feature	Kleros	LocalArbitrator
Decentralization	High (distributed jury)	Medium (trusted arbitrator)
Accessibility	Medium-High technical barriers, Medium cost barriers	Low-Medium technical barriers, Low cost barriers
Scalability	Medium throughput capacity, High concurrent disputes	High throughput capacity, Medium concurrent disputes
Flexibility	Medium dispute type adaptation	High customizable implementation
User Experience	Medium-High complexity	Low-Medium complexity
Legal Integration	Low-Medium court recognition	Medium-High court recognition

The feature comparison highlights the complementary strengths of the two approaches. The Kleros implementation excels in decentralization metrics, offering stronger guarantees against censorship and bias. The LocalArbitrator implementation offers advantages in accessibility, flexibility, user experience, and legal integration, making it potentially more suitable for mainstream adoption.

6.4. Discussion of Implications

6.4.1. Addressing Power Imbalances in Rental Markets

A central finding of our research is the potential of blockchain-based rental systems to address fundamental power imbalances in rental markets. These imbalances, documented extensively by Desmond and Wilmers [9] and others, stem from several sources that the DRADR model directly addresses.

The DRADR model significantly reduces information asymmetry (improving from 2.4-2.8 to 4.8/5 on our metrics) through transparent terms, comprehensive records, and automated notifications. This transparency addresses the "adverse selection" problems that often arise due to limited information about rental properties and conditions. The model also mitigates resource disparities, particularly in access to legal representation, by reducing dispute resolution costs by 75-95%, making the process more accessible regardless of resources. Our game-theoretic analysis quantifies this change: in traditional systems, the high cost of dispute resolution creates a dominant strategy for landlords to make excessive claims, knowing tenants cannot economically challenge them. Both DRADR implementations alter this dynamic by making dispute resolution economically viable even for small amounts.

Perhaps most significantly, both DRADR implementations create fundamental shifts in incentive structures that promote more balanced relationships. Our analysis of incentive alignment showed improvements from 2.5-2.7 to 4.0-4.2/5, reflecting several key mechanisms: automated enforcement, deposit protection through neutral smart contracts, immutable histories creating accountability, and low-cost arbitration creating credible threats against opportunistic behavior.

6.4.2. Cross-Jurisdictional Applicability

Our analysis of cases across multiple jurisdictions revealed important insights about the cross-jurisdictional applicability of the DRADR model and its two implementations. The model demonstrated different adaptation requirements and feasibility across legal systems. Common law jurisdictions (UK, Australia) generally showed higher feasibility scores (3.9-4.2) than civil law jurisdictions (France, Germany, Spain: 3.3-3.7). This difference stems from several factors: greater contractual flexibility in common law systems, specific procedural formalism in civil law systems (e.g., état des lieux in France), and typically stronger consumer protection provisions in civil law jurisdictions.

Our analysis revealed significant differences in cross-jurisdictional adaptability between the two DRADR implementations. The LocalArbitrator approach consistently demonstrated greater

adaptability to jurisdiction-specific requirements, with its configurable arbitrator selection particularly valuable for alignment with local legal expectations. For civil law jurisdictions, the LocalArbitrator implementation's higher feasibility scores (averaging 0.4 points higher than Kleros) reflect several adaptability advantages: the ability to require specific qualifications for arbitrators, customizable arbitration processes to incorporate mandatory procedural elements, and clearer paths to traditional enforcement mechanisms.

6.4.3. Limitations and Challenges

While our analysis demonstrates the significant potential of the DRADR model, it also reveals important limitations and challenges that must be addressed for successful implementation.

Several technical limitations affect both DRADR implementations, including blockchain scalability constraints, oracle dependencies for confirming off-chain events, smart contract immutability creating challenges for addressing bugs, and integration complexities with traditional systems. Perhaps the most significant limitations involve legal and regulatory challenges: the uncertain legal enforceability of smart contracts in many jurisdictions, compliance complexity across different regulatory frameworks, incorporation of mandatory consumer protection provisions, and integration with traditional enforcement mechanisms. Practical implementation also faces adoption barriers: technical literacy requirements, cryptocurrency requirements for gas fees, industry resistance from established property management companies, and infrastructure requirements for supporting services.

Despite these challenges, our comparative analysis indicates that both DRADR implementations offer substantial improvements over traditional rental systems, with the choice between them depending on specific contextual factors and priorities. The LocalArbitrator approach provides a more accessible entry point for mainstream adoption due to its lower costs, simpler interfaces, and greater jurisdictional adaptability, while the Kleros approach offers stronger guarantees of neutrality for contexts where trust minimization is paramount.

6.5. Synthesis of Findings

Integrating the findings from our case studies, game-theoretic analysis, and technical evaluation, we can provide comprehensive answers to our research questions ¹:

RQ1: The DRADR model demonstrates significant efficiency improvements over traditional systems, with dispute resolution times reduced by 90-98%. Fairness metrics show improvements in access to justice (from 2.3-2.8 to 3.8-4.1 out of 5) but slightly lower legal consistency (from 4.2-4.4 to 3.7-3.9 out of 5).

RQ2: The game-theoretic analysis reveals a fundamental shift in equilibrium strategies. In traditional systems, the dominant strategy for landlords is to withhold deposits regardless of actual damages, while tenants are incentivized to accept even unfair claims. Both DRADR implementations create equilibria where honest damage claims emerge as the optimal strategy for landlords, with the LocalArbitrator implementation showing slightly stronger incentives for honest behavior.

RQ3: The DRADR model significantly reduces information asymmetry through transparent record-keeping and reduces financial barriers to dispute resolution. This addresses a key source of power imbalance in rental relationships, although jurisdictional adaptation is necessary to replicate the tenant protections embedded in some legal systems.

RQ4: The comparison of Kleros and LocalArbitrator implementations reveals distinct tradeoffs. The Kleros approach provides stronger decentralization and censorship resistance at higher cost and complexity. The LocalArbitrator approach offers better economic efficiency (27-33% lower gas costs for dispute operations) and jurisdictional adaptability at the expense of increased trust requirements. Both significantly outperform traditional systems in efficiency and accessibility metrics.

These findings suggest specific implementation recommendations based on context:

1. The Kleros implementation is better suited for high-value rental agreements where trust minimization is critical.
2. The LocalArbitrator implementation is optimal for lower-value rentals where cost efficiency and simplicity are priorities.
3. Jurisdictional adaptation is necessary for both implementations, with the LocalArbitrator approach offering more flexibility for compliance with specific legal requirements.

These findings demonstrate that blockchain-based rental agreements offer significant advantages over traditional systems, with the choice between arbitration approaches depending on specific use case priorities.

7. Conclusion and Future Work

7.1. Summary of Contributions and Practical Implications

This paper presents the Decentralized Rental Agreement and Dispute Resolution (DRADR) model, demonstrating significant contributions through its dual-implementation framework, comparative analysis of arbitration approaches, game-theoretic modeling, and cross-jurisdictional assessment. Our findings reveal that blockchain-based rental systems can reduce dispute resolution times by 90-98% compared to traditional methods while fundamentally altering strategic dynamics to incentivize honest behavior. The comparative analysis highlights key tradeoffs: *Kleros implementation* offers stronger decentralization at higher cost, while the LocalArbitrator implementation provides better economic efficiency with increased trust requirements. The model has substantial practical implications for stakeholders: property managers and landlords benefit from automated processes and immutable records; tenants gain improved access to dispute resolution with costs reduced from thousands to hundreds of dollars and enhanced transparency addressing the 43% of complaints involving unclear terms [25]; legal professionals face both challenges to traditional service models and opportunities in jurisdictional adaptation; and policymakers encounter improved market functionality alongside regulatory integration challenges. These implications demonstrate DRADR's potential to address longstanding rental market inefficiencies while creating more balanced power dynamics between parties.

7.2. Limitations and Future Research

Despite promising results, significant limitations challenge DRADR implementation. Building on these findings, we propose four critical research directions: (1) Technical Enhancements through zero-knowledge proofs for privacy, IoT integration for objective verification, and upgradeability patterns; (2) Hybrid Arbitration Models combining automated resolution with tiered human arbitration; (3) Jurisdiction-Specific Implementations with regulatory compliance modules and sandbox testing; and (4) Empirical User Studies examining stakeholder interactions and adaptations to blockchain-based rental systems. Addressing these directions will be essential to overcome critical gaps in privacy, dispute resolution, regulatory compliance, and user experience necessary for widespread adoption.

7.3. Conclusion

In conclusion, the DRADR model demonstrates the potential of blockchain technology to transform rental relationships by addressing inefficiencies, power imbalances, and transparency challenges in traditional systems. Our implementation and evaluation of two distinct arbitration approaches provides a foundation for future research and practical applications in this domain. As blockchain technology continues to mature, implementations like the DRADR model can serve as bridges between the theoretical promise of decentralized systems and the practical requirements of real-world applications, ultimately contributing to more efficient, transparent, and equitable economic relationships. The dual arbitration implementations we presented offer complementary approaches that can be selected

based on specific deployment contexts and priorities, advancing the practical application of blockchain technology in everyday contractual relationships.

Author Contributions: Muntasir Jaodun and Khawla Bouafia worked on the conceptualization of the raised issue; they wrote the original draft version, and then carried out editing the revision. Muntasir J. carried out the operationalization and coding of the models. Muntasir J. and Khawla B. proofread the draft and revision. Khawla B. supervised the process. Khawla B. acquired funding to support the creation of the paper. Both authors have read and agreed to the published version of the manuscript.

Funding: The project was supported by the ELTE EKÖP

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and code presented in this study are available on request from the author.

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

References

1. Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*; 2008. <https://bitcoin.org/bitcoin.pdf>.
2. Wood, G.; et al. Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum Project Yellow Paper* **2014**, *151*, 1–32.
3. Szabo, N. Smart Contracts. *Unpublished manuscript* **1994**.
4. UN-Habitat. World Cities Report 2022: Envisaging the Future of Cities. Technical report, United Nations Human Settlements Programme, 2022.
5. Urban Institute. Technology and the Future of the US Rental Housing Market. Technical report, Urban Institute Report, 2023.
6. Li, J.; Greenwood, D.; Kassem, M. Blockchain in the Built Environment and Construction Industry: A Systematic Review, Conceptual Models and Practical Use Cases. *Automation in Construction* **2019**, *102*, 288–307.
7. Abualhamayl, A.; Almalki, M.; Al-Doghman, F.; Alyoubi, A.; Hussain, F.K. Blockchain for Real Estate Provenance: An Infrastructural Step Toward Secure Transactions in Real Estate E-Business. *Service Oriented Computing and Applications* **2024**, pp. 1–15.
8. Guseva, Y. Decentralized Markets and Self-Regulation. *George Washington Law Review* **2024**.
9. Desmond, M.; Wilmers, N. Do the Poor Pay More for Housing? Exploitation, Profit, and Risk in Rental Markets. *American Journal of Sociology* **2019**, *124*, 1090–1124.
10. Zhuk, A. Applying Blockchain to the Modern Legal System: Kleros as a Decentralised Dispute Resolution System. *International Cybersecurity Law Review* **2023**, *4*, 351–364.
11. Kadioglu Kumtepe, C.C. A Brief Introduction to Blockchain Dispute Resolution. *J. Marshall LJ* **2020**, *14*, 138.
12. Dong, Y.; Dong, Z. Bibliometric Analysis of Game Theory on Energy and Natural Resource. *Sustainability* **2023**, *15*, 1278.
13. Khan, S.N.; Loukil, F.; Ghedira-Guegan, C.; Benkhelifa, E.; Bani-Hani, A. Blockchain Smart Contracts: Applications, Challenges, and Future Trends. *Peer-to-peer Networking and Applications* **2021**, *14*, 2901–2925.
14. Lorenz, G. Regulating Decentralized Financial Technology: A Qualitative Study on the Challenges of Regulating DeFi with a Focus on Embedded Supervision. *Stan. J. Blockchain L. & Pol'y* **2024**, *7*, 136.
15. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. Blockchain Technology Overview. *arXiv preprint* **2019**, *arXiv:1906.11078*.
16. Ahmad, I.; Alqarni, M.A.; Almazroi, A.A.; Alam, L. Real Estate Management via a Decentralized Blockchain Platform. *Computer Materials Continuum* **2021**, *66*, 1813–1822.
17. Rentberry Inc.. Decentralized Home Rental Platform. Technical report, Rentberry Whitepaper, 2017.
18. Schmitz, A.; Rule, C. Online Dispute Resolution for Smart Contracts. *Journal of Dispute Resolution* **2019**, p. 103.
19. Li, M.; Lal, C.; Conti, M.; Hu, D. LEChain: A blockchain-based lawful evidence management scheme for digital forensics. *Future Generation Computer Systems* **2021**, *115*, 406–420.

20. Koulu, R. Blockchains and Online Dispute Resolution: Smart Contracts as an Alternative to Enforcement. *SCRIPTed* **2016**, *13*, 40–69.
21. Allen, D.W.E.; Berg, C.; Lane, A.M.; Potts, J. The Economics of Crypto-Democracy. In Proceedings of the Proceedings of the IPCPS Conference, 2018, pp. 63–73.
22. Oliveira, N.B. The Role of International Arbitration in Resolving Cross-Border Smart Contract Disputes: Opportunities and Challenges. *Journal of International Dispute Settlement* **2023**, *14*, 128–153.
23. Bouafia, K.; Molnár, B.; Majid, G. Blockchain Technologies for Transparency in FinTech. In Proceedings of the Proceedings of the International Congress on Information and Communication Technology. Springer, 2024, pp. 575–585.
24. Bouafia, K.; Gulalov, M. Blockchain Solutions for Authorization and Authentication. *Procedia Computer Science* **2024**, *237*, 115–122.
25. National Association of Realtors. Rental Housing Market Survey. Technical report, NAR Research Group, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.