الجمهورية الجزائرية الديمقراطية الشعبية
The People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة محمد بوضياف بالمسيلة
University Mohamed Boudiaf of M'sila

كلية الرياضيات والإعلام الآلي
Faculty of Mathematics and Informatics

قسم الإعلام الآلي
Department of Computer Science

Domain: Mathematics and Computer Science

Thesis Presented to Fulfill the Partial Requirement
for **Master's Degree** in Computer Science

Specialty: Information Systems and Software Engineering

Prepared By: Ahmed-Amin Chellali

Supervised By:

Adel Moussaoui

ENTITLED

Real Estate Tokenization on Blockchain

Jury Members

Abdessattar Ghemougui	President
Adel Moussaoui	Supervisor
Makhlouf Benazi	Examiner

Academic Year 2024/2025

DEDICATION

If every letter speaks of effort, let every silence between them whisper thanks. When strength failed and certainty wavered, who remained but Him? I owe this to none but God Almighty.

And yes — for those He placed in my life as signs of His mercy. I shall be grateful to you.

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Adel Moussaoui, for their continuous support, invaluable guidance, and patience throughout the course of this research.

To all who contributed in ways big or small, spoken or silent — thank you.

ABSTRACT

Real estate represents one of the most substantial global asset classes, yet its accessibility and efficiency are hindered by structural challenges, including high capital entry requirements, protracted transaction processes, and dependence on costly intermediaries. This thesis investigates the technical viability of employing blockchain-based tokenization as a means to mitigate these limitations. The study examines the extent to which smart contracts can automate essential functions within real estate investment. The developed prototype illustrates that blockchain technology has the potential to streamline transactional workflows, reduce associated costs, and enhance transparency by limiting the reliance on third-party intermediaries. The study also identifies major barriers to adoption, particularly in the current legal and digital infrastructure, which lacks the maturity to fully support real-world implementation at scale. Through the empirical validation provided by this work, this study contributes to the growing body of knowledge on real estate tokenization.

Keywords: Real estate, Tokenization, Blockchain, Smart contracts, Fractional ownership, Compliance, Decentralized governance.

الملخص

يمثل قطاع العقارات أحد أكبر فئات الأصول على مستوى العالم، إلا أن الوصول إليه وكفاءته يعانيان من تحديات هيكلية، تشمل متطلبات رأس المال المرتفعة، وطول العمليات التعاقدية، والاعتماد الكبير على الوسطاء المكلفين. تهدف هذه الرسالة إلى دراسة الجدوى التقنية لاستخدام تقنيات البلوك تشين في ترميز الأصول العقارية كوسيلة لتجاوز هذه القيود. تبحث الدراسة في مدى قدرة العقود الذكية على أتمتة الوظائف الأساسية في استثمار العقارات. ويُظهر النموذج المطور أن تقنية البلوك تشين تمتلك إمكانيات واعدة في تبسيط سير العمليات التعاقدية، وتقليل التكاليف المرتبطة بها، وتعزيز الشفافية من خلال تقليل الاعتماد على الأطراف الوسيطة. كما تحدد الدراسة أبرز معوقات التبني، لا سيما في البنية التحتية القانونية والرقمية الحالية التي لا تزال تفتقر إلى النضج اللازم لدعم تطبيق هذه التقنية على نطاق واسع. ومن خلال التحقق *التجريبي الذي توفره هذه الدراسة، تُسهم الرسالة في إثراء المعرفة المتنامية حول ترميز العقارات باستخدام البلوك تشين.*

الكلمات المفتاحية: العقارات، الترميز، البلوك تشين، العقود الذكية، الملكية الجزئية، الامتثال، الحوكمة اللامركزية.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: GENERAL INTRODUCTION

The real estate industry represents one of the world's largest asset classes, yet it remains structurally constrained by inefficiencies that limit market participation and liquidity. Traditional property investment requires substantial capital commitments, involves lengthy transaction processes, and relies heavily on intermediary institutions that extract significant fees while introducing operational opacity. These structural barriers effectively exclude smaller investors from prime real estate opportunities while constraining property owners' ability to access liquidity from their holdings.

Blockchain technology has emerged as a transformative innovation capable of addressing fundamental inefficiencies across multiple industries, with real estate representing a particularly compelling application domain. The technology's capacity to enable fractional ownership through tokenization, automate compliance through smart contracts, and facilitate transparent transactions without centralized oversight aligns directly with the real estate sector's modernization needs. Most significantly, blockchain enables the representation of physical property rights as digital tokens that can be fractionally owned, securely transferred, and programmatically managed without traditional intermediary dependencies.

1.1. Problem Statement

The conventional real estate investment ecosystem operates through complex, intermediary-dependent processes that create systematic barriers to market participation and capital efficiency. This thesis investigates whether blockchain-based tokenization can provide a viable alternative framework that preserves essential market functions while reducing transactional friction and expanding access to real estate investment opportunities.

The research addresses critical questions regarding the technical feasibility of implementing compliant tokenization systems that maintain necessary regulatory oversight while leveraging blockchain's transparency and automation capabilities. Specifically, the study examines whether smart contract architectures can effectively coordinate property tokenization, investor onboarding, fractional ownership management, and revenue distribution within decentralized environments.

1.2. Research Objectives

This thesis aims to design and implement a comprehensive tokenization platform that demonstrates the viability of blockchain-based real estate investment systems. The primary objective involves constructing a modular smart contract architecture that enables property tokenization through Security Token Offerings (STOs).

The system encompasses essential investment lifecycle components including user registration and KYC verification, asset tokenization and fractional ownership distribution, automated revenue sharing through dividend mechanisms, and decentralized governance enabling token holder participation in asset management decisions. Additionally, the implementation integrates Special Purpose Vehicle (SPV) structures to maintain legal separation between individual assets and provide familiar investment frameworks for traditional market participants.

The research focuses specifically on creating an executable artifact that serves as both a technical reference implementation and a foundation for iterative refinement, rather than attempting comprehensive market simulation or full legal enforceability validation.

1.3. Scope and Limitations

This project concentrates exclusively on the smart contract implementation for real estate tokenization, deliberately excluding frontend interface development to maintain focus on core blockchain functionality. The implementation targets Ethereum-compatible networks to leverage established infrastructure and development tools while ensuring compatibility with existing DeFi ecosystems.

The prototype establishes a minimum viable transactional core operating under realistic technical and procedural assumptions while avoiding full market condition simulation. The implementation deliberately excludes comprehensive legal compliance mechanisms, cross-jurisdictional regulatory variations, and complex dispute resolution procedures, acknowledging these as external challenges requiring specialized legal and regulatory expertise beyond the technical scope.

Web platform integration, secondary market development, Real KYC integration and production-level security implementations remain designated for future research phases, allowing this work to focus on establishing fundamental tokenization mechanics and governance frameworks.

1.4. Thesis Structure Overview

The thesis organization reflects the progression from theoretical foundations through practical implementation to system evaluation. Chapters 2 and 3 cover the theoretical background of blockchain technology and tokenization. Chapter 4 examines related work in blockchain-based real estate applications and establishes the theoretical framework for tokenization systems. Chapter 5 presents the system architecture, design rationale, and the methodological approach adopted.

Chapter 6 provides detailed implementation analysis of individual smart contract components, including the tokenization platform, security token contracts, SPV mechanisms, and governance systems. And evaluates system functionality through systematic testing and analyzes the implementation's effectiveness in achieving core research objectives.

Chapter 7 consists of General conclusion that synthesizes key findings regarding the viability of blockchain-based real estate tokenization, acknowledges implementation limitations, and identifies priorities for future research including legal integration, scalability enhancement, and market adoption strategies.

CHAPTER 2: BLOCKCHAIN TECHNOLOGIE

2.1. Introduction

This chapter aims to provide a comprehensive understanding of blockchain technology. It defines blockchain and outlines its evolution. Furthermore, it explains key characteristics, alongside its architecture. Finally, it examines the challenges hindering its widespread adoption.

2.2. Definition of Blockchain Technology

Blockchain technology emerges from the integration of multiple disciplines, including software engineering, distributed computing, cryptographic science, and economic game theory. It combines established computer science mechanisms and merges them into a single solution, providing a stable and scalable software infrastructure, securing digital assets, and enabling global decentralization [1]. Over time, researchers and experts have defined blockchain from various perspectives:

Cryptographic Perspective: *"A sequentially linked chain of cryptographically secured records, ensuring data integrity and resistance to tampering* [2]*."*

Distributed Systems Perspective: *"A distributed ledger for recording transactions, maintained by many nodes without central authority through a distributed cryptographic protocol* [3]*."*

Economic and Financial Perspective: *"A peer-to-peer electronic cash system that eliminates intermediaries by allowing direct financial transactions between participants* [4]*."*

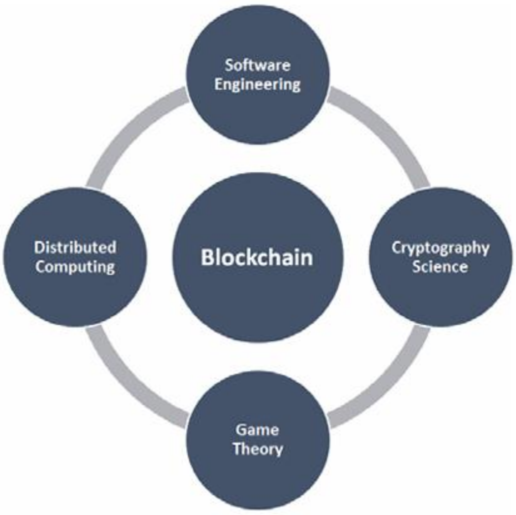


Figure 2.1 Multidisciplinary Foundations of Blockchain Technology [1]

2.3. Evolution of Blockchain Technology

2.3.1. Blockchain 1.0: Bitcoin

Bitcoin, introduced by Satoshi Nakamoto, was the first practical blockchain application; It aimed to create a decentralized, trustless electronic cash system. According to him, the primary motivations behind it were to eliminate reliance on trusted third parties and reduce transaction costs. The system relied on digital signatures and cryptographic proof rather than trust. To address the double-spending problem without a central authority, Bitcoin implemented a proof-of-work consensus mechanism [4].

2.3.2. Blockchain 2.0: Ethereum

Vitalik Buterin proposed Ethereum to expand blockchain use beyond financial transactions. Unlike Bitcoin, Ethereum supports general-purpose computation and smart contracts [5]. Additionally, Ethereum’s native cryptocurrency, Ether (ETH), is available for trading and serves as an incentive for network participants based on various requirements [6].

2.3.3. Blockchain 3.0:

The major setback of Blockchain 1.0 and 2.0 is that they face scalability limitations. Blockchain 3.0 was to make cryptocurrencies globally viable by improving scalability, interoperability, and sustainability. It involves Decentralized Apps (Dapps), (DeFi), Non-Fungible Tokens (NFTs), and Decentralized Autonomous Organizations (DAOs) that run on a blockchain network, promoting inter-chain transactions using techniques like sharding. Blockchain 3.0 uses Proof of Stake and Proof of Authority consensus mechanisms for faster and more efficient smart contract execution. Examples include ICON, IOTA, Cardano, and Aion [7].

2.3.4. Blockchain 4.0:

Blockchain 4.0, aims to deliver Blockchain Technology as a business-usable platform, integrating it with technologies like Artificial Intelligence. It enables seamless integration of different platforms to meet business and industry demands. Unibright and SEELE are examples of platforms that forward Blockchain 4.0 utilities [7].

2.4. Characteristics of Blockchain Technology

Blockchain characteristics are classified into functional characteristics and emergent characteristics. Functional characteristics are those which are mandatory for functioning, without which the system may not exist or function properly [8].

- **Decentralized:** Blockchain's decentralization is a fundamental characteristic. Instead of relying on a central authority, data is distributed across a network of nodes. reducing single points of failure [9].
- **Distributed Ledger:** Blockchain operates as a decentralized ledger, capturing a comprehensive record of all transactions within a peer-to-peer network [9]. All network nodes store the same data simultaneously adhering to DLT. The ledger is publicly accessible, and anyone can create a redundant copy of the database and verify it against other copies [10].
- **Immutability:** Once data is added to a block in the blockchain, it is very difficult to change or tamper with. Each transaction is timestamped and secured using cryptographic hash functions. The hash is irreversible, and any slight change in the input data will change the hash value [11].
- **Consensus Mechanisms:** Consensus is crucial to ensure data consistency across decentralized participants. The consensus protocol outlines the sequence in which nodes can introduce changes to the database, ensuring unanimous agreement among all network nodes regarding the database’s state [9].
- **Security:** Numerous hash functions, digital signatures, public key cryptography, and Merkle tree patterns are used to make blockchain a very secure cryptographic algorithm [10].

Emergent characteristics are derived and emerged as a result of functional characteristics. Thus include: Trust, achieved through transparent consensus rather than intermediaries; Auditability, ensured by timestamped, immutable records; Transparency, provided by shared access to the ledger; Tamper resistance, through the combined effects of immutability and cryptographic security; and Anonymity [8].

2.5. Understanding Blockchain: A Layered Architecture Approach

The term "Architecture" is broad and can be interpreted and applied from multiple perspectives. Indeed, many resources do not share a consistent definition under this term, some refer to network architecture, others to technological or physical architecture, and still others to organizational structures or software system components.

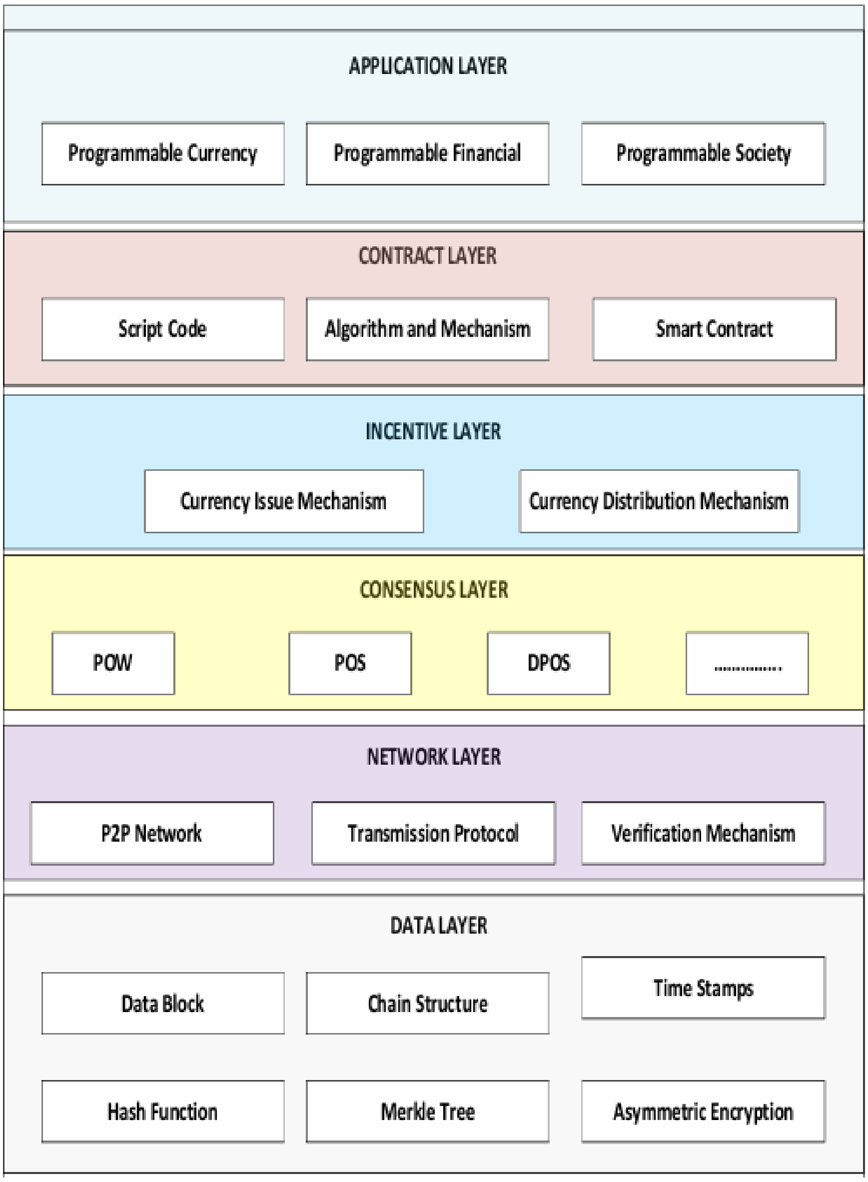


Figure 2.2 Blockchain layered Architecture [12]

Moreover, many resources do not share a common layered architecture. The structure and number of layers often differ depending on the author's perspective or the system's context. Figure [2.2] illustrates a general layered architecture for a generic blockchain application. However, this architecture may vary significantly based on specific use cases and implementation requirements.

2.5.1. Data Layer

The blockchain structure comprises linked data blocks, timestamp mechanisms, and cryptographic techniques such as hash functions, asymmetric encryption, digital signatures, and Merkle trees.

- **Blocks:** Each block includes a header and data section. The header contains metadata such as block number, hash of the previous block's header, hash of the current block's data, timestamp, block size, and nonce. Altering any previous block changes its hash. The Merkle root, stored in the block body, summarizes all transactions in the block via a Merkle tree. The data section lists verified transactions and ledger events [13].
- **Merkle Trees:** Merkle trees validate data integrity. Leaf nodes represent individual data blocks; non-leaf nodes contain the hash of their children. This structure enables efficient and secure verification of large datasets [15].
- **Transactions:** A transaction records an interaction, such as cryptocurrency transfer or asset registration. It includes the sender's address, public key, digital signature, transaction inputs, and outputs. The digital signature verifies the sender's ownership of the input resources. Each block may include multiple transactions or none [13].

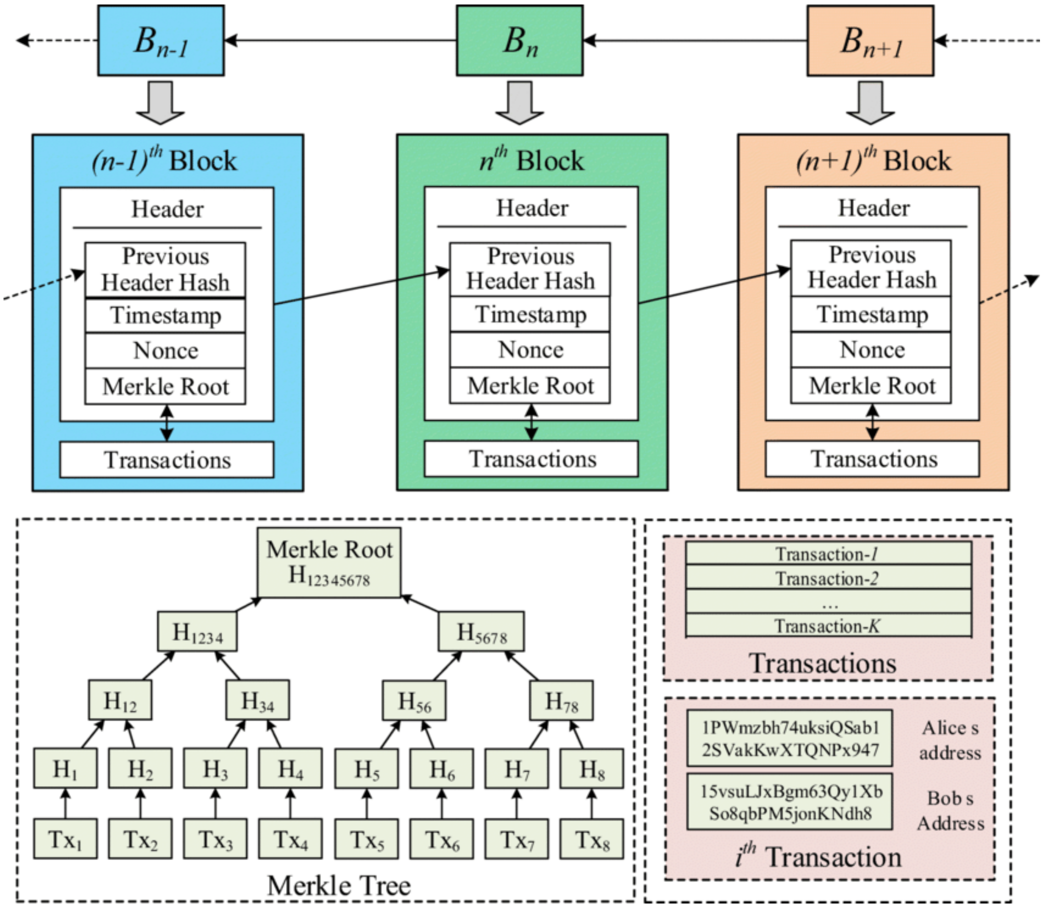


Figure 2.3 Blockchain Structure [14]

2.5.2. Network Layer

A Blockchain Network manages inter-node communication through a peer-to-peer (P2P) networking where each node maintains a full copy of the distributed ledger [9]. A peer-to-peer (P2P) network consists of nodes that operate cooperatively as equals, without central servers or client-server hierarchy. This architecture supports direct transactions between participants and ensures network resilience, as communication persists despite the failure of individual nodes [13].

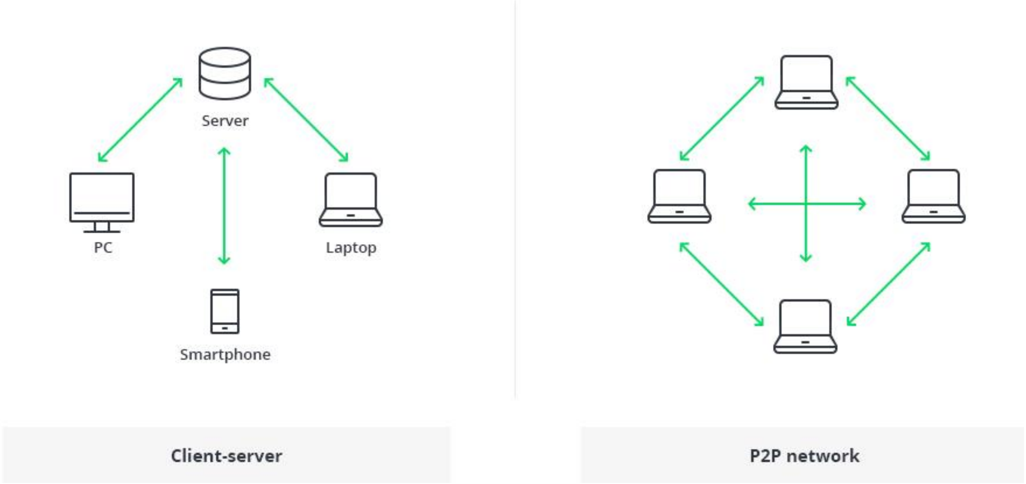


Figure 2.4 Client-Server vs. Peer-to-Peer Network Architectures [17]

2.5.2.1. Governance and Permissions:

The network architecture of blockchain depends on the governance models whether Blockchain is public or private, and on the permissions related to joining, reading, writing and commit operations [13].

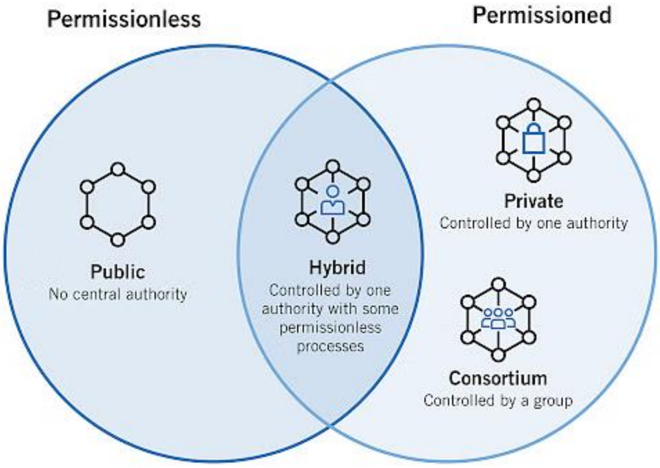


Figure 2.5 Types of Blockchain Networks [18]

- **Permissionless blockchain networks** Public blockchains function as open, decentralized ledgers. Any participant may publish blocks without prior approval, and all nodes share equal read and write permissions. Consensus mechanisms, which require resource expenditure or locking, protect against malicious actions [13]. A high number of validators strengthens security but also increases latency, leading to slower transactions [19].
- **Permissioned blockchain networks** are ones where node publishing blocks must be authorized by some authority. It is possible to restrict access rights and to restrict who can issue transactions [13]. Permissioned blockchains tend to be more efficient. Because access to the network is restricted, there are fewer nodes on the blockchain, resulting in less processing time per transaction [19].
- **Public Blockchain:** Public blockchains are open networks without access restrictions. Any user can join, validate transactions, and access the ledger. This ensures decentralization and transparency. Bitcoin represents an early example. While secure and trustless, public blockchains face challenges such as limited scalability, slow transaction rates, and high energy consumption. Common applications include voting and fundraising [20].
- **Private Blockchain:** Private blockchains operate within closed networks controlled by a single organization. Only authorized participants can access data or validate transactions. With fewer nodes, they offer higher speed and scalability but compromise on decentralization and security. Typical use cases include supply chain tracking, internal asset management, and enterprise voting [20].
- **Hybrid Blockchain:** Hybrid blockchains integrate elements of public and private models. They enable selective transparency, where access and visibility depend on predefined rules. This structure balances cost-efficiency and security but limits openness and user incentives. Suitable domains include real estate, retail, and regulated sectors such as finance [20].
- **Consortium Blockchain:** Consortium blockchains are governed by a group of stakeholders rather than a single entity. They support secure, collaborative data exchange across organizations. With better scalability and control than public chains, they still offer limited transparency. Use cases include interbank settlements, payments, and academic research [20].

2.5.3. Consensus Layer

In centralized systems, a single authority controls data and decisions. Decentralized systems rely on stakeholder consensus to validate transactions and agree on network state. Consensus mechanisms serve as protocols that ensure legitimate agreement, even in the presence of malicious actors. They secure the network by preventing fraud and preserving ledger integrity.

The general process of consensus involves selecting a leader node based on predefined rules. Each node maintains a transaction pool where incoming transactions are stored. The leader creates a new block by selecting legitimate transactions and broadcasting it to the network. Other nodes then verify the block's validity and, if accepted, add it to their local copy of the blockchain. A block is considered valid only when the majority of nodes approve and include it in their chain. To encourage participation, leaders may receive rewards for successfully creating new blocks. As a result, nodes often compete for leadership based on different criteria, depending on the consensus mechanism [21].

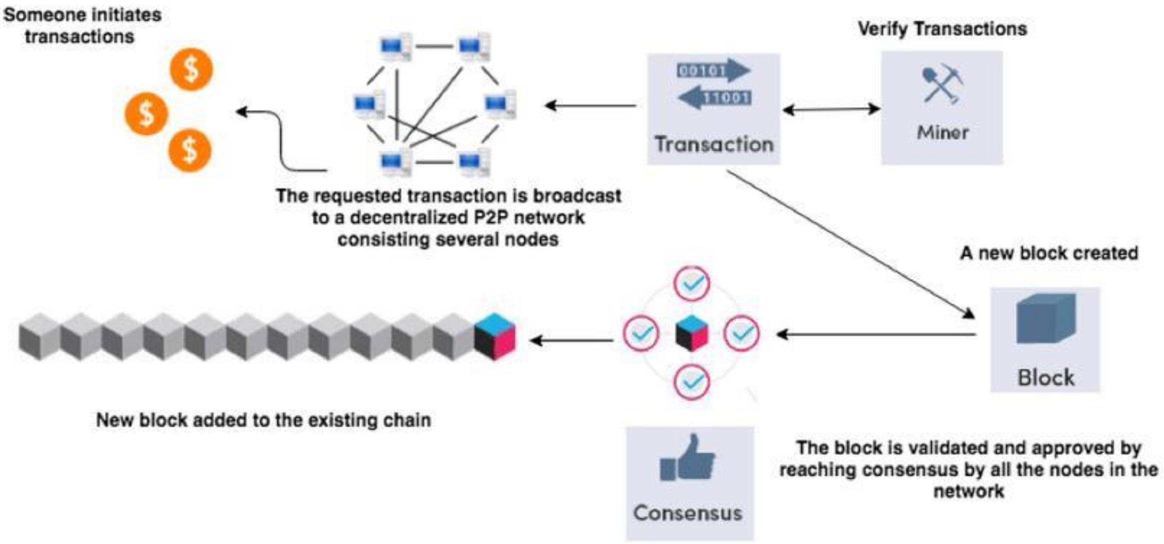


Figure 2.6 Overview of Blockchain Consensus Mechanism [22]

For a consensus mechanism to work effectively, it must meet key requirements: Agreement (all nodes must agree on the same value), Termination (the process ends when all nodes reach a conclusion), Validity (the agreed value must match the value from a legitimate node), Fault-Tolerance (despite malicious or faulty nodes), and Integrity [10].

2.5.3.1. Types:

Various research papers classify these mechanisms based on their design principles, such as proof-based, vote-based, and leader-based approaches. Some focus on energy efficiency, scalability, or security trade-offs [10]. Two of the most widely used:

- **Proof-of-Work:** Proof-of-Work (PoW) was the first blockchain consensus mechanism, introduced by Satoshi Nakamoto for Bitcoin. The second biggest crypto, Ethereum, also once used this consensus mechanism before transitioning to Proof-of-Stake [23]. In this model, the next block is published by the first user to solve a computationally intensive puzzle. The solution serves as "proof" of the work performed. A common puzzle involves finding a block header hash digest less than a target value by manipulating a nonce. While it makes denial-of-service attacks difficult and is open to anyone with the necessary hardware, PoW is computationally intensive, consumes significant power, and can lead to a "hardware arms race". It also has the potential for a 51% attack [13].

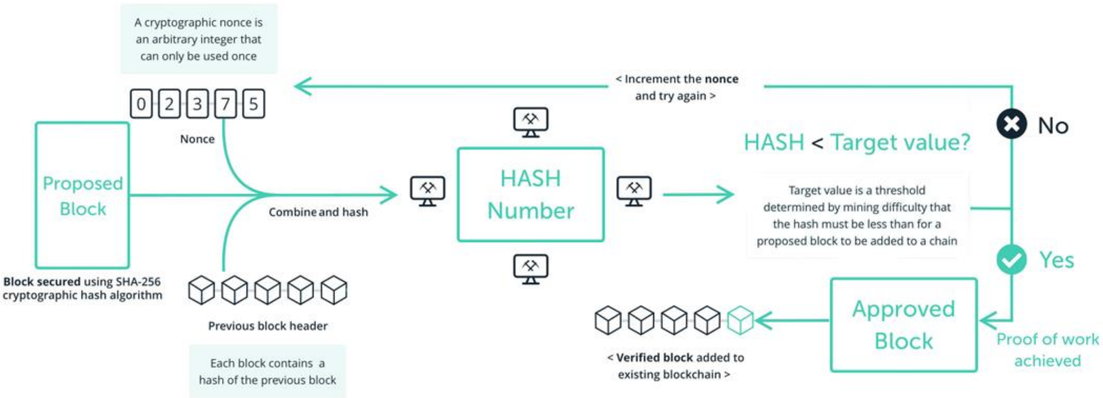


Figure 2.7 Overview of PoW Consensus Mechanism [23]

- **Proof-of-Stake:** Proof-of-Stake (PoS) is a blockchain consensus mechanism where randomly selected validators produce and approve blocks instead of miners. Validators stake the network’s native tokens by locking them into the blockchain, receiving rewards based on their total stake. To become a validator, a user must stake a minimum amount of the network’s cryptocurrency (e.g., 32 ETH for Ethereum). This stake serves as collateral that ensures honest behavior. Validators acting maliciously risk having their stake slashed, meaning a portion of their funds is forfeited as punishment. While validators are selected randomly, those with higher stakes have a greater probability of producing the next block. Once a block is proposed, other validators verify it before adding it to the blockchain [24].

Figure 2.8 Overview of PoS Consensus Mechanism [24]

This model relies on the idea that users with more stake (cryptocurrency invested in the system) are more likely to act in the system's best interest. The likelihood of publishing a new block is tied to the ratio of a user's stake to the overall staked cryptocurrency. PoS is less computationally intensive than PoW, but raises concerns over centralization and 51% attacks through financial dominance. Some PoS variants also face the *nothing-at-stake* problem, where validators can support multiple forks at no cost, delaying consensus on a single chain [13].

Consensus protocols vary by blockchain type. Public blockchains typically use permissionless mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS), to enable open participation. Private and consortium blockchains employ permissioned models, to restrict validation to authorized entities via voting or multi-party algorithms. Hybrid blockchains may combine elements from both to balance openness and control [25].

2.5.3.2. Ledger Conflicts

For some blockchain networks it is possible that multiple blocks will be published at approximately the same time. This can lead to temporary ledger conflicts at any given moment; and temporarily create different versions of the blockchain.

Most blockchain networks resolve these conflicts by waiting until the next block is published and then adopting the chain that is the longest as the "official" blockchain. Which is the longest chain rule, where the chain with the most accumulated proof-of-work (or equivalent consensus measure) is accepted as the official ledger, and transactions from orphaned blocks are returned to the pending pool.

The acceptance of a block is often probabilistic rather than deterministic since blocks can be superseded. To mitigate this, transactions are typically considered confirmed only after multiple subsequent blocks have been added. While a theoretically powerful adversary could rewrite the blockchain by outpacing the network, practical constraints in computational resources and security mechanisms, such as checkpointing, can prevent such attacks [13].

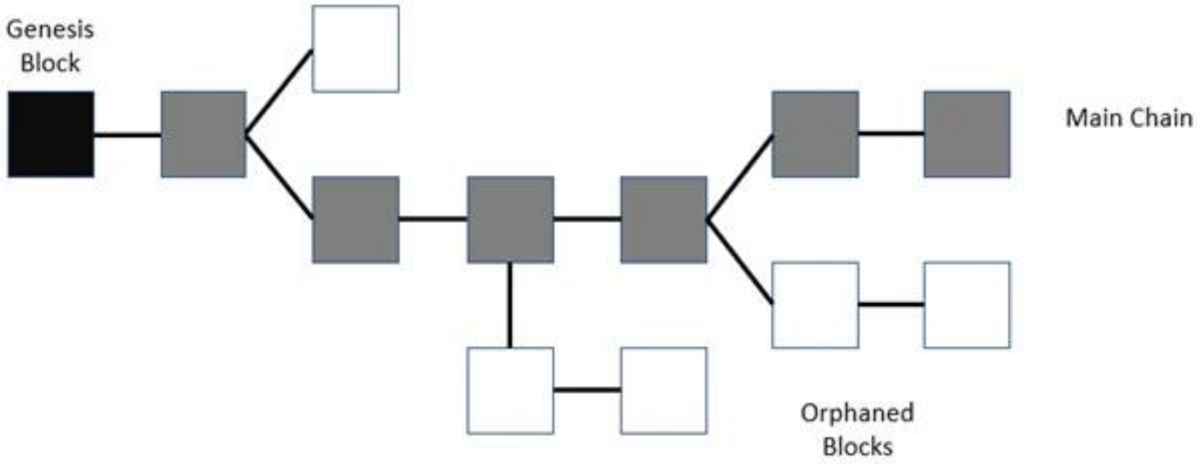


Figure 2.9 Ledger Conflict Resolution [26]

2.5.4. Incentive Layer

This layer strengthens the blockchain's security verification by offering specific incentives to nodes that participate in security verification. It incorporates economic rewards, the currency distribution system, and the currency issue mechanism to incentivize the generation of new blocks. Miners (the creators of new blocks) are rewarded with incentives to encourage the network to continue working on data verification and ensuring the chain's security [12].

2.5.5. Smart Contract Layer

A smart contract is a self-verifying, self-executing, and tamper-resistant computer program proposed by Nick Szabo in 1994. It allows transactions to be executed without third parties automatically when predefined conditions are met, taking transactions as input, changing states based on function logic, and triggering output events [27].

In the context of blockchain technology, a smart contract is "a collection of code and data (sometimes referred to as functions and state) that is deployed using cryptographically signed transactions on the blockchain network", such as Ethereum's smart contracts or Hyperledger Fabric's chaincode. Smart contracts have the ability to perform a range of actions, including calculations, storing information on the blockchain, exposing properties that reflect their current state, and automatically transferring digital assets to other accounts based on predefined conditions

[10]. They are event-driven programs [27]. Smart contracts gained prominence in the second generation of blockchain technology (Blockchain 2.0). Smart contracts are commonly implemented using the Solidity programming language on various blockchain platforms [27].

2.5.5.1. Execution of Smart Contracts:

These smart contracts are executed by nodes within the blockchain network, and it is crucial that all nodes executing the smart contract derive the same results, with these results then being recorded on the blockchain [13]. Leveraging a consensus protocol to manage and run a sequence of events. Furthermore, each smart contract is assigned a unique address by the blockchain technology, which can be used to trigger its execution through transactions [27].

The smart contract on a blockchain follows a structured process governed by consensus among network nodes:

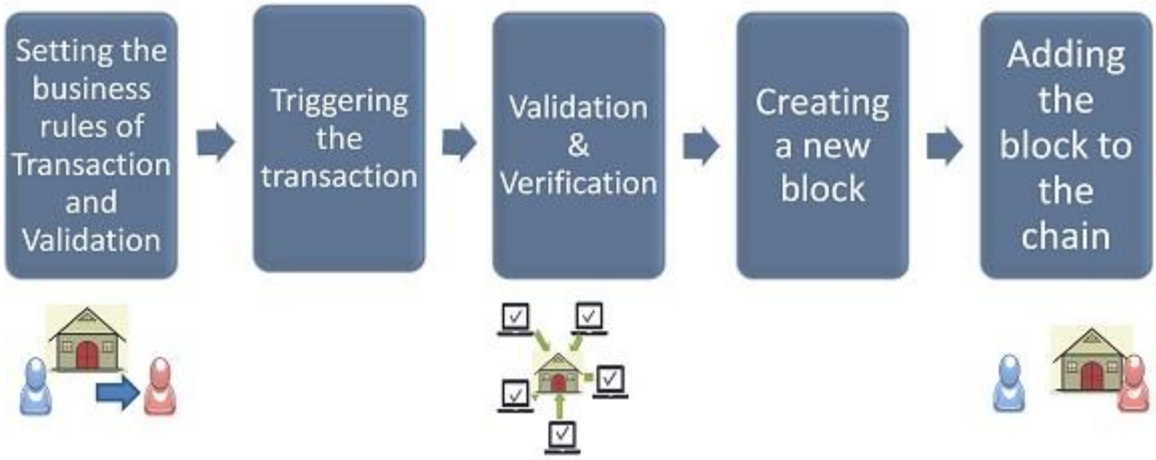


Figure 2.10 Execution of Smart Contracts [28]

Also, executing a smart contract requires transaction fees (gas) to compensate nodes for computational work, preventing spam and resource misuse [28].

2.5.6. Application Layer

This technology is being adopted across various industries, with spending projected to reach $16 billion by 2023 [20]. Initially, its applications were primarily in the banking and financial industries. However, other sectors have increasingly explored and adopted blockchain to meet their business objectives. These include: Cryptocurrency; Healthcare; Asset Management; Internet of Things (IoT); Voting; Improved Record Keeping;[20] and Supply chain management [29].

2.6. Challenges in Adopting Blockchain Technology

According to Namasudra (2020), adoption barriers can be grouped as follows:

- **Cybersecurity Risks**: Blockchain systems remain vulnerable to cyberattacks, including 51% attacks and breaches. Permissionless blockchains are particularly exposed [29].
- **Double Spending**: Exploiting the validation gap can allow malicious reuse of tokens, undermining trust in transaction finality [29].
- **Immutability vs. Flexibility**: Blockchain's resistance to data modification, while essential for integrity, complicates corrections or updates, often requiring forks [29].
- **GDPR Compliance**: The decentralized architecture conflicts with data protection laws that require identifiable data controllers, posing legal issues within the EU [29].
- **Energy Consumption**: Proof-of-Work systems demand excessive computational power, resulting in high energy usage and environmental concerns [29].
- **Data Scalability**: Growing on-chain data size necessitates compression solutions to prevent storage inefficiencies [29].
- **Transaction Throughput**: Low processing capacity (e.g., 3–7 TPS for Bitcoin) limits usability in high-volume applications.
- **Cost Barriers**: High development and operational costs, including energy use and skill scarcity, restrict adoption [29].
- **Regulatory Uncertainty**: Ambiguous legal frameworks deter institutional involvement, especially in jurisdictions lacking blockchain-specific regulation.
- **Interoperability**: The absence of standardized protocols hampers integration and cross-chain collaboration [29].

2.7. Conclusion

Blockchain technology is still in its early stages of development, with vast potential yet to be fully realized. While it has already made significant strides, particularly in the realm of cryptocurrencies like Bitcoin and Ethereum, the broader applications of blockchain are still being explored. With this foundational understanding in place of core components and operational mechanisms and both characteristics and challenges of blockchain systems. The next chapter will explore tokenization, a direct application of blockchain technology that involves converting real-world assets into digital tokens. This process holds the potential to democratize access to traditionally illiquid markets, such as real estate, while also introducing complex technical, legal, and regulatory considerations.

CHAPTER 3: BLOCKCHAIN-BASED TOKENIZATION

3.1. Introduction

Tokenization is a transformative technology that has revolutionized the way assets are represented, transferred, and managed, especially during the rise of smart contracts. This chapter will offer a comprehensive overview of blockchain-based tokenization. Its definition, classifications. It also reviews key standards and examines both the advantages and associated challenges.

3.2. Definition of Tokenization

The concept of a token is not exclusive to blockchain technology; rather it is a broader concept that likely exists independently of blockchain technology. In data security, tokenization is the process of converting sensitive data into a nonsensitive digital replacement that serves as unique identifiers, called a token, that maps back to the original. The token can then act as a secure replacement for the data. which retains essential data properties without exposing the original information. The token itself is nonsensitive and has no use or value without a connection to the data vault where the original data is securely stored [30].

Tokenization differs from encryption. While both aim to protect data, Encryption transforms plaintext data into an unreadable format (ciphertext) using a cryptographic algorithm and a key. Tokenization substitutes sensitive data with a surrogate value known as a token, which has no intrinsic value or meaning and bears no mathematical relationship to the original data [31].

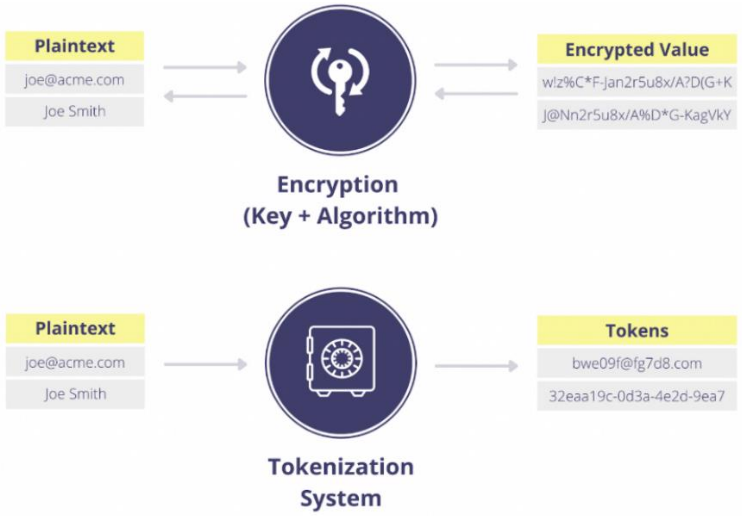


Figure 3.1 Tokenization vs Encryption [32]

Tokens have distinct meanings from different perspectives:

- Technical Perspective: Tokens are digital representations of value encoded as lines of computer code that specify what the token represents [33].
- Legal Perspective: Tokens can signify digital assets that hold agreed-upon value and are secured using cryptographic protocols, such as cryptocurrencies [33].

Beyond data security and compliance, tokens are versatile and can represent various rights, such as ownership rights, access rights to digital services. Additionally, they can represent assets from both the real world (e.g., real estate, collectibles, commodities, company shares) and the virtual world (e.g., cryptocurrencies, lottery tickets, in-game character skills). This can make assets safer, easier to trade, and can streamline operations by automating transactions and increasing liquidity [33].

Figure 3.2 Asset Tokenization in Different Industries [34]

In a general sense, tokenization refers to the process of converting a piece of information, asset, or right into a digital token that can be securely managed or transferred. This concept is widely applied in various industries, including finance, cybersecurity, and digital identity management.

3.3. Blockchain-Based Tokenization

In the blockchain domain Tokenization is defined as the transformation process of data/assets into a random digitized sequence of characters (a token). a token is a digital representation of an asset available in the physical or virtual worlds. Digital assets cannot be directly recorded on the blockchain, and tokenization is required to format these assets. A token serves as a reference to the original data or assets for blockchain applications but cannot be utilized to determine their values.

A token itself does not inherently include economic value; instead, this value is typically assigned by the market. Essentially, a token can be considered a symbol validated by smart contracts of the target blockchain system. Once validated, it can be used in various applications or traded [33].

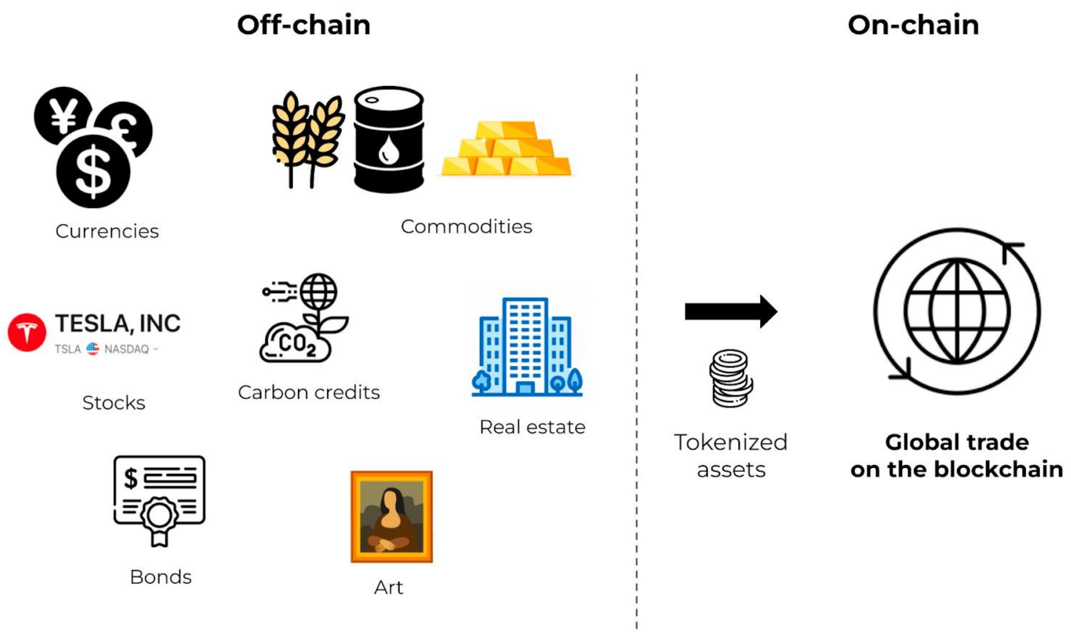


Figure 3.3 RWA Tokenization [35]

3.3.1. Types of Token Technology

Tokens in blockchain represent two types of information: who owns it, that is to which address it is recorded, and the chain of transactions. Because blockchain is public, immutable, and chronological, the history of transactions is natively available in the ledger, creating a traceable sequence of transactions. Fractional ownership is possible via decimal units of account and/or multi-signature schemes [36]. Blockchain-based tokens can be categorized into three primary types based on their underlying technology:

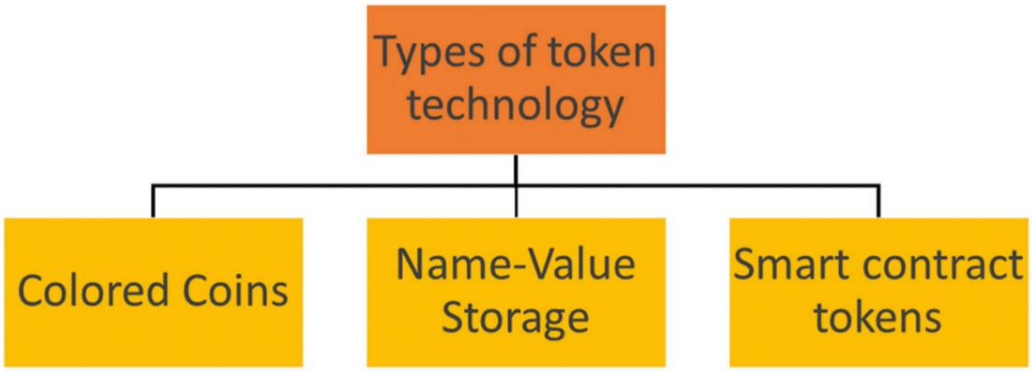


Figure 3.4 Types of Token Technology [36]

- **Colored Coins:** Colored Coins represent one of the earliest tokenization mechanisms, initially implemented on Bitcoin (Mizrahi, n.d.). This approach utilizes built-in Bitcoin protocol features, such as burning coins and embedding data in transactions, to create distinct digital assets. These assets, though recorded on the blockchain, require specialized software for interpretation, as standard Bitcoin wallets do not recognize them. Economic logic and ownership structures emerge through social consensus within communities that adopt Colored Coins [36].

- **Name-Value Storage (NVS):** Name-Value Storage (NVS), pioneered by Namecoin (2014) and enhanced by Emercoin (2015) (Loibl, 2014; Emercoin Community Documentation, n.d.), employs blockchain as a decentralized key-value database. Users insert immutable key-value pairs, where the key is unique, and the value represents associated data. Updates occur by reusing the same key with new values, ensuring a historical record of changes. While not inherently a unit of account, NVS enables decentralized data management and ownership tracking through cryptographic mechanisms [36].
- **Smart Contract-Based Tokens:** Ethereum introduced smart contract-based tokens, marking a significant advancement in blockchain tokenization (Ethereum Wiki, 2017). Smart contracts, self-executing pieces of code deployed on the blockchain, define token characteristics, including supply, transfer conditions, and associated rules. These tokens inherit blockchain's immutability and cryptographic ownership [36].

The two most common blockchain-based digital assets are cryptocurrencies and tokens. The biggest differentiation between the two is that cryptocurrencies have their own blockchains, whereas crypto tokens are built on an existing blockchain [37]. Usually, the token is based on cryptocurrency. To create the token, the user must spend ("burn") some coins and apply scripts depending on technology. Cryptocurrency is spent as well to make further transactions with tokens. For instance, in Ethereum, Ether coins needed to pay for "gas" to run a transaction with a smart contract (Ethereum Wiki, 2017). Hence, in such systems, tokens do not exist without cryptocurrency [36].

3.3.2. Token Management on the Blockchain

Users create, update, delete tokens and transfer them within the blockchain via mechanism of public-key cryptography. A token is attached to a user's address, where the address is a representation of a user's public key, and only the relevant private key can be applied to sign a transaction—tokens altered (transferred, updated, deleted, etc.) via blockchain transactions [36].

3.4. Classification of Digital Objects

The concept of tokens is deeply tied to digital objects, which serve as a bridge between the physical and digital worlds. Digital objects carry state information that allows them to be represented and interacted with in digital systems. Digital objects can be classified based on two primary characteristics: Tangibility and Fungibility [33].

Tangibility:

- **Tangible Objects:** These are assets with a physical existence, which are typically unique. When represented on a blockchain, they exist both in their physical form and as a digital token that verifies their authenticity and ownership. Examples include real estate, physical collectibles, and gold reserves [33].
- **Intangible Objects:** These do not have a physical form but exist as abstract representations within a digital system. Examples include digital services, software licenses, and in-game items [33].

Fungibility:

- **Fungible Objects:** These are interchangeable with one another and can be divided into smaller units without losing value. Common examples include currencies (e.g., USD, Bitcoin) and utility tokens (e.g., ERC-20 tokens) [33].
- **Non-Fungible Objects:** These are unique and indivisible, meaning each object has distinct characteristics and cannot be exchanged on a one-to-one basis. Examples include NFTs (ERC-721 tokens), digital artwork, and unique in-game assets [33].

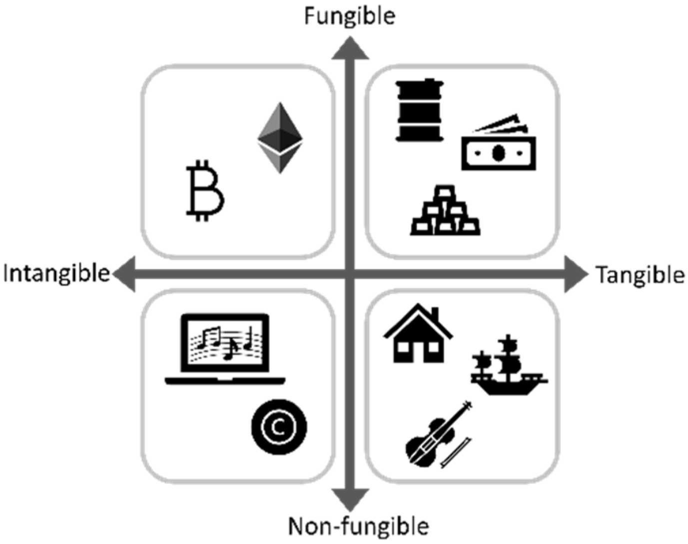


Figure 3.5 Classification of Digital Objects [38]

3.5. Classification of Tokens

A token acts as an abstraction layer that encodes ownership, rights, and properties of a digital object. From an application perspective, there is no universally recognized international classification of tokens, as different countries define and regulate them differently. However, in literature, tokens are often classified based on their **functionalities** [33]. Leading to four main types:

- **Payment Tokens:** Used primarily for digital payments, often in the form of cryptocurrencies. Bitcoin (BTC) is the most well-known payment token, and it is used to settle transactions on the Bitcoin network. Other examples of payment tokens include Ethereum (ETH), Litecoin (LTC) and Dash (DASH) [39].
- **Utility Tokens:** Primarily designed to provide access to a particular product or service. For instance, Ether (ETH) is the utility token of the Ethereum blockchain, which is used to pay for transaction fees and run smart contracts on the network [39].
- **Asset Tokens:** Asset tokens are backed by a physical asset such as gold or real estate. These tokens represent ownership of the underlying asset and can be bought or sold on blockchain-based marketplaces [39].
- **Security Tokens:** Security tokens represent ownership of an underlying asset, such as a company's stocks or real estate. Security tokens are subject to federal securities regulations, and their issuance and trading are governed by the securities and Exchange commission (SEC) [39].

Figure 3.6 Functional Classification of Tokens [39]

From a **technical perspective**, tokens can also be categorized based on their **fungibility:**

- **Fungible Tokens (FT):** These tokens are interchangeable and identical to one another in terms of type and value; They are generally divisible into smaller units without affecting their value; They Are not unique; The ERC-20 standard on Ethereum is a prime example of a fungible token standard [33].
- **Non-Fungible Tokens (NFT):** These tokens are not interchangeable and are indivisible; Each NFT contains distinctive information and attributes that make it unique; They are used to track the ownership of individual assets; The ERC-721 standard on Ethereum provides the technological framework for creating and tracking NFTs. Each ERC-721 token must have a universally unique identifier [33].
- **Semi-Fungible Tokens (SFT):** This is a newer class of tokens that combines features of both fungible and non-fungible tokens; They offer more flexible interfaces for representing complex assets or processes; The Ethereum ERC-1155 standard (Multi Token Standard) allows for the representation of both fungible and non-fungible assets within the same contract; SFTs can be more efficient for creating and bundling token transactions [33].

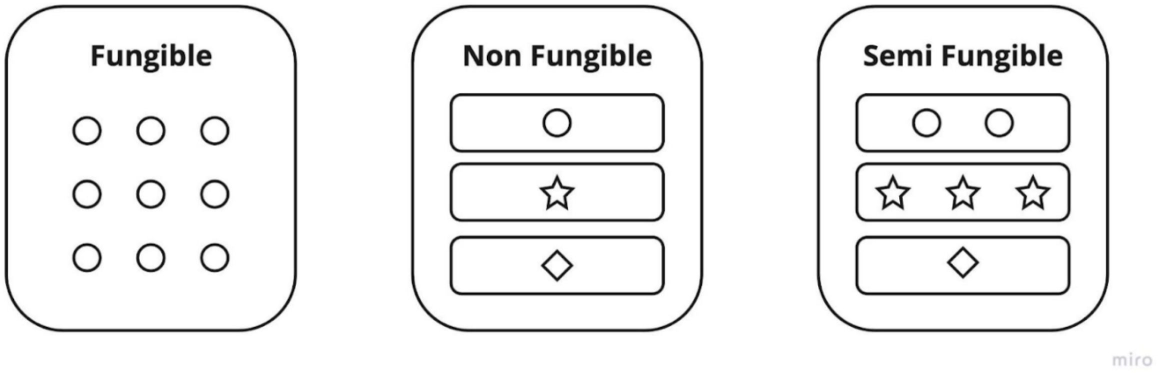


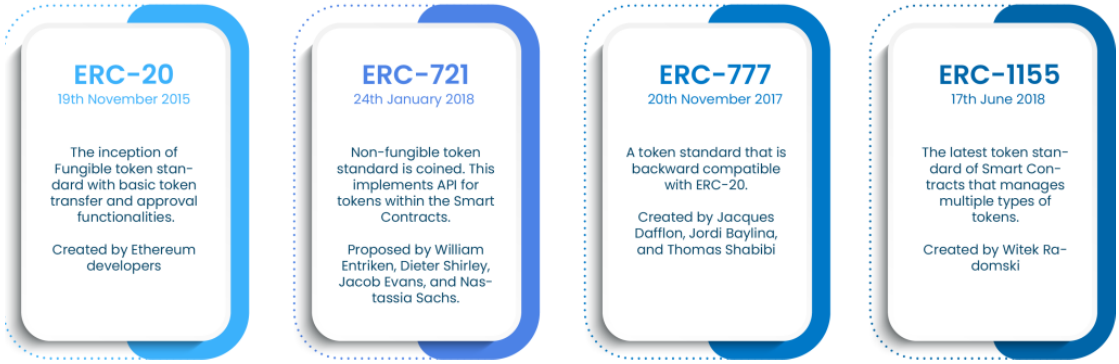
Figure 3.7 Technical Classification of Tokens [40]

These classifications are not always mutually exclusive, meaning that different digital objects can be mapped into different types of tokens, such as crypto coins, asset-tokens, or utility tokens. According to practical use cases and actual characteristics, one kind of digital asset can be classified into different categories. For instance, a token could function as a utility token while also representing a unique digital collectible, thus exhibiting non-fungible characteristics [33].

3.6. Token Standards

Token Standards are a set of regulations and technical specifications delineate the behavior of tokens on a given blockchain. By ensuring that tokens conform to a particular format, these standards facilitate their interoperability with a multitude of services and applications within the ecosystem, including wallets and exchanges. The implementation of this standard is vital in order to facilitate the smooth transfer and use of tokens across various platforms and applications [41]. There is currently no standard for establishing a compatible and universal token structure that suits most application platforms. Even though some standards are developed, they target specific application platforms [33]. which may produce some Interoperability Challenges. In general Token standards define the structure, behavior, and interactions of tokens on a blockchain by specifying their properties, functions and Smart contract rules.

In terms of smart contracts, Token standards are a subset of smart contract standards. For blockchains that support smart contracts, token standards represent a guide for the creation, issuance, and deployment of new tokens on them. Most blockchain smart contracts currently use Ethereum, and the most common token standards are ERC-20, ERC-721, ERC-777, and ERC-1155 [42].



The inception of Fungible token standard with basic token transfer and approval functionalities.

Created by Ethereum developers

Non-fungible token standard is coined. This implements API for tokens within the Smart Contracts.

Proposed by William Entriken, Dieter Shirley, Jacob Evans, and Nastassia Sachs.

A token standard that is backward compatible with ERC-20.

Created by Jacques Dafflon, Jordi Baylina, and Thomas Shabibi

The latest token standard of Smart Contracts that manages multiple types of tokens.

Created by Witek Radomski

Figure 3.8 Token Standards [43]

3.6.1. ERC-20 Token Standard

The ERC-20 token standard is a blueprint for creating fungible tokens on the Ethereum network. Like other digital assets, ERC-20 tokens are most commonly developed by organizations and tech-focused companies. These tokens allow the entity to customize their utility, such as granting voting rights or rewarding mechanisms [42]. The primary purpose of the ERC-20 standard is to ensure interoperability [44]. Each ERC-20 token has a standardized core functionality, enabling interoperability between ERC-20 tokens and compatibility with other products and services [42].

3.6.2. ERC-721 Token Standard

This standard serves as the framework for non-fungible tokens (NFTs) on the Ethereum blockchain. This type of Token is unique and can have different values than another Token from the same Smart Contract [45]. A key feature of ERC-721 is that each NFT token must have a universally unique identifier [33]. ERC-721 smart contracts keep a record of token ownership, allowing for the transfer of tokens between users. The contract also keeps track of the total token supply and the token balance held by individual addresses [46].

3.6.3. ERC-777 Token Standard

ERC-777 aims to address the limitations of ERC-20. This token standard makes it more efficient for smart contracts to send and receive tokens through a mechanism known as ‘Hooks’, a function that combines what would have been two messages sending tokens and notifying a contract into one. Furthermore, The ERC-777 standard uses the same underlying functions as ERC-20 and introduces additional functions to reject transactions from a blacklisted address, while remaining backward-compatible with ERC-20, allowing seamless interaction between tokens from both standards [42].

3.6.4. ERC-1155 Token Standard

This Multi Token Standard on Ethereum offers "semi-fungible" options and the potential to represent both fungible and non-fungible assets within the same smart contract [33]. The ERC-1155 token standard combines the best features of previous standards to create fungibility-independent and gas-efficient token contracts. In simple terms, it's a standard for managing multiple token types within a single smart contract, as explained by its developer, Enjin: "a single smart contract that can govern an infinite number of tokens." With ERC-721, sending multiple NFTs requires multiple transactions, which can overwhelm the network and lead to high transaction costs [42]. ERC-1155 on the other side operates by allowing multiple items to be stored in a single smart contract, allowing any number of items to be sent in a single transaction to one or more recipients (Batch transfers) [47].

3.7. Benefits of Asset Tokenization

- 1. **Increased liquidity:** One of the main benefits of asset tokenization is increased liquidity. Traditional assets such as real estate or artwork can be illiquid, meaning they cannot be easily bought or sold. Tokenization allows **fractional ownership** of these assets, which can be bought and sold in smaller units. This means that investors can access a wider range of investment opportunities that were previously out of reach.

- **2. Lower costs:** Asset tokenization eliminates the need for intermediaries, which can reduce costs and increase transparency. This can make investment opportunities more accessible to a wider range of investors.
- **3. Improved transparency:** The use of blockchain technology in asset tokenization provides greater transparency and accountability that can reduce the risk of fraud.
- **5. Global access:** Asset tokenization allows for global access to investment opportunities. Traditional assets may be limited to a specific geographic location, making it difficult for investors to access them. Tokenization allows these assets to be traded on a global scale, which can increase investment opportunities and diversification [48].

3.8. Challenges and Risks of Asset Tokenization

- **Regulatory Compliance:** Legal frameworks for tokenized assets remain under development. Most jurisdictions lack clear guidelines, creating uncertainty for issuers and investors. However, some countries, such as Switzerland, have taken early steps to regulate tokenized markets, offering a model for legal integration [48].
- **Security Concerns:** Digital assets face risks including cyberattacks, theft, and fraud. These threats can erode investor trust and result in substantial losses. Effective protection requires strong encryption, multi-factor authentication, and frequent security audits [48].
- **Liquidity Risks:** Many tokenized assets, especially those derived from illiquid markets, struggle to attract sufficient trading activity. Without an active secondary market, price discovery and exit opportunities remain limited. Accurate valuation and strategic market analysis and proper valuation methods are necessary to support liquidity [48].

3.9. Conclusion

Tokenization has emerged as a transformative approach to representing assets digitally, leveraging blockchain technology to enable secure, efficient, and transparent transfers. However, designing an effective tokenization model requires careful consideration of both technical and legal aspects. From a technical standpoint, selecting the appropriate token model is crucial to ensuring that digital tokens accurately reflect real-world assets, their ownership, and transactional rules. Understanding the nature of assets and choosing the right token standard, is a mandatory step in any tokenization process. On the legal front, tokenization introduces challenges related to property rights, regulatory compliance, and enforceability. Addressing these challenges is crucial for the broader adoption of tokenized assets across industries. The next chapters explores real estate tokenization and examines how blockchain can be integrated into that sector.

CHAPTER 4: RELATED WORK

This chapter provides a synthesis of Real Estate Tokenization, recent real-world initiatives, identifies key implementation challenges, and outlines academic contributions in this domain.

4.1. Real Estate Tokenization

Real estate assets can broadly be classified into residential, commercial, and industrial assets and all real estate assets fall into one of three categories [49]. The traditional real estate market, while historically considered a stable investment sector, faces numerous challenges that hinder accessibility, efficiency, and transparency.

A comprehensive systematic study [50] investigates the potential applications of blockchain technology across various real estate subsectors. This study aimed to understand why and where blockchain should be applied in the real estate sector, based on an extensive review of the literature.

The most prominent areas for blockchain adoption identified in this review include:

- Land Administration
- Property Transactions
- Real Estate Investment
- Leasing and Renting
- Real Estate Administration

Less attention was given to real estate development and maintenance, suggesting these areas are currently perceived as less suitable or are underexplored in blockchain research.

Figure 4.1 Most Attractive Real estate categories for Blockchain [50]

Based on this analysis, it becomes evident that the that challenges in traditional real estate markets are deeply interconnected, often forming a chain of cause and effect that drives structural and operational inefficiencies. Illiquidity, for example, results from high capital requirements and the lack of fractional ownership, which limit access to institutional investors and concentrate market power. High transaction costs arise from excessive intermediary involvement and inefficient administration, which also delay processes. Fragmented information systems and paper-based records further undermine transparency, increase fraud risk, and erode trust.

Figure 4.2 Problems with current Real Estate Ecosystem [49]

As noted in existing literature, blockchain technology addresses these root issues. Distributed ledgers enhance transparency by providing tamper-proof, verifiable records. Tokenization facilitates fractional ownership, lowers entry barriers, and increases liquidity through secondary market access. Smart contracts automate agreements and escrow functions, reduce reliance on intermediaries, and cut costs, time, and disputes. In rental markets, they enforce lease terms objectively, improving efficiency and reducing conflicts.

Figure 4.3 Benefits of Real Estate Tokenization [51]

4.2. Current Implementations and Case Studies

Several international initiatives have explored the application of blockchain in real estate. These include both public-sector pilots and private-sector ventures:

For instance, government-backed initiatives such as the UK's "Digital Street," Sweden's (Chromaway), and Georgia's (Bitfury). However, these projects were often early practical attempts rather than revolutionary transformations and many pilots did not progress far beyond initial stages [52]. Regulatory Sandboxes and Experimental Platforms. Multiple case studies have been explored in countries like Russia, the United States, and France. For instance, the FinTech firm Nivaura issued a tokenized bond within the UK Financial Conduct Authority's sandbox. Similarly, the Swiss Stock Exchange (SIX) is developing a digital asset infrastructure through its SDX project. Russia amended its legislation in 2019 to allow digital representation of rights and obligations, although full legislative support is still pending [53]. Several private companies have initiated tokenization of real estate assets: RealT (United States) has tokenized 58 residential properties—primarily in Detroit—between October 2019 and February 2021 [54]. BrickMark facilitated a partial acquisition of a commercial property in Zurich using its proprietary tokens. This project employed a hybrid structure and initially targeted only accredited institutional investors [55]. IPSX listed assets in London, while Templum Markets operates in the U.S. as an SEC-approved platform for digital real estate assets. Other European firms, including Blockimmo (Switzerland), Blocksquare (Slovenia), Brickblock and iEstate (Germany), and Tokeny (Luxembourg), are also developing Ethereum-based tokenization platforms. International initiatives like UPRETS, Leaseum Partners, and London & Oxford Properties focus on tokenized real estate investment funds aimed at global markets, particularly targeting investors in China and the U.S [56].

These initiatives demonstrate a growing interest in real estate tokenization but also reveal a lack of maturity and standardization in the market. But most projects are at early stages, and many pilots have not progressed significantly [52]. The outcomes of practical applications have often been unclear, and few were in widespread use, making it difficult to draw conclusions about theoretical benefits [55].

4.3. Academic Literature

Academic literature on real estate tokenization has significantly increased, particularly post-2020 [57]. Early research established foundational theories, focusing on blockchain's theoretical benefits, challenges, and concepts across four main categories: land administration, real estate transactions, tokenization, and real estate management [55].

4.3.1. Blockchain for Land Record Management

Initial studies explored replacing paper-based land registries with blockchain-based systems to enhance security, transparency, and efficiency. One proposal introduced a secure land record management system (LRMS) using asymmetric cryptography and a character-to-integer mapping scheme to reduce data overhead, alongside a trading module for ownership transfer [58]. Another project implemented Ethereum smart contracts in India to ensure constant transparency and immutable updates [59]. Other research employed SHA-256 and elliptic curve cryptography in a decentralized registry reliant on majority consensus to prevent fraud [60]. Prototypes of private blockchain-based real estate management systems have also emerged, automating registration, consent, and sale finality to avoid double-selling and fraud [61,62].

4.3.2. Smart-Contract Frameworks for Real Estate Transactions

A parallel research track focuses on smart contract architectures for automating sales, escrow, and title transfer. One Ethereum-based framework eliminates intermediaries through smart escrow contracts [63]. Another project developed a decentralized app (DApp) to manage property listings, offers, and transfers on a private Ethereum network [64]. Several studies propose design methodologies for property rental smart contracts, specifying actors, roles, and business logic [65]. One such framework integrates registration, payments, dispute resolution, and digital identity Integration using Ethereum's Proof-of-Authority model [66]. Another system leverages Hyperledger Fabric to codify property rules, generate deeds, and automate tax records [67].

Also several blockchain-based systems and approaches for rental and real estate have been proposed: One proposes a blockchain-based solution using smart contracts and IPFS (Inter Planetary File System) [68]. Another source detailed a similar blockchain housing rental system but with Oracle services [69]. Other third work has explored the tokenization of rental real estate assets and proposed smart contracts for automating rental processes. It outlines required functions and state transitions and serves as a design reference for implementing rental contracts via smart contracts and Dapps [70].

4.3.3. Tokenization Approaches and Frameworks

Recent scholarship has proposed various models for real estate tokenization, each emphasizing distinct legal, technical, and operational mechanisms:

One model emphasizes blockchain-first approach by leveraging ERC-1155 tokens for both fungible (fractional shares) and non-fungible (entire property) ownership. Properties and Stakeholders are registered and verified by an on-chain administrator. Legal documents are uploaded to the InterPlanetary File System (IPFS), with their content hashes embedded in the metadata of the corresponding tokens. Smart contracts oversee the minting, transfer, and validation processes [49]. A second model introduces Special Purpose Vehicles (SPVs) as legal custodians of tokenized assets. Upon property registration, an SPV is created to hold the title, and its equity is tokenized using ERC-777 standard security tokens. These tokens are distributed through a Security Token Offering (STO), subject to Know Your Customer (KYC) and Anti-Money Laundering (AML) requirements. Legal agreements are stored on IPFS and referenced on-chain to maintain verifiability [71]. A conceptual framework advances the notion of Title Tokens, digital surrogates for legal property titles directly tied to cadastral systems and validated by trusted entities such as notaries or registrars. This model advocates for interoperability through cross-chain protocols and proposes embedding legal logic into smart contracts via "smart laws." It also introduces the CHMOD model to digitally represent and manage the full bundle of property rights [52].

4.3.4. Our Work

In the course of evaluating multiple blockchain application domains within real estate, we have chosen to concentrate our efforts on the real estate investment sector, rather than other subdomains such as land administration, leasing, or development. This decision is grounded in both strategic and pragmatic reasoning:

- **Strategic Alignment**: Our current capabilities are primarily technical, and we lack access to comprehensive domain-specific studies or institutional collaboration that would be necessary for systemic reform in areas like land registry or legal title management.
- **Feasibility and Impact**: Compared to other blockchain use cases, investment models—particularly those involving tokenization—tend to be more modular and easier to prototype from a technical standpoint. They can be implemented without requiring direct involvement of legal or governmental frameworks.
- **Practical Research Potential**: Unlike land administration and transaction systems, which have already attracted significant academic and industry attention, the practical implementation of investment models—especially those based on tokenization through Special Purpose Vehicles (SPVs)—remains relatively underexplored. This presents a valuable opportunity.

Given these constraints, our work adopts the SPV-based tokenization investment model [71] —as a legally feasible and technically implementable approach under current conditions. Specifically, we prototype the smart contract infrastructure required to operationalize this model.

CHAPTER 5: METHODOLOGY

5.1. Research Design and Approach

This study follows a prototype-based exploratory research approach to investigate how smart contracts can structure and enforce core processes within real estate transactions, including fractional ownership representation, automated dividend distribution, and decentralized governance for investment decisions. The research trajectory commenced with the identification of an abstract conceptual model for real estate tokenization, which subsequently served as the theoretical foundation for practical implementation.

The methodological approach recognizes that blockchain-based financial systems require immediate engagement with technical constraints to validate theoretical propositions. Rather than pursuing traditional hypothesis-testing frameworks, this research generates knowledge through the iterative refinement of functional prototypes, where each implementation cycle reveals both technical feasibilities and conceptual limitations inherent in the abstract model.

The design process prioritizes empirical understanding through implementation. The research aims to expose both the potential and the limitations of smart contracts. This study does not attempt to simulate full market conditions or legal enforceability but focuses instead on modeling a minimally viable transactional core under realistic technical and procedural assumptions.

5.1.1. Artifact-Centric Validation

The research involves building a working artifact or system to explore ideas, validate concepts, or evaluate feasibility. The research treats the smart contract implementation as both a research instrument and primary deliverable. This approach acknowledges that in distributed ledger applications, theoretical constructs cannot be meaningfully evaluated without confronting the deterministic requirements of code execution.

5.1.2. Constructive Research Strategy

The methodology follows a constructive approach where knowledge emerges through the systematic translation of abstract concepts into executable code. This process revealed gaps between theoretical elegance and practical implementation, generating insights that pure conceptual modeling could not provide. The reference abstract model served as a starting point rather than a rigid template, allowing for flexible evolution based on technical discoveries.

5.2. Functional Scope and Assumptions

The system architecture establishes essential components necessary for demonstrating core tokenization mechanics while maintaining operational coherence under realistic technical constraints. This scoping strategy reflects the research objective of creating a technical reference implementation rather than developing comprehensive market solutions. The functional boundaries were determined through systematic distillation of the abstract model to its most critical transactional elements, ensuring the prototype remains sufficiently complex to demonstrate meaningful functionality while avoiding unnecessary implementation overhead.

5.2.1. Core System Components

- User identity verification and registration;
- Asset tokenization and fractional ownership representation;
- Ownership and financial Transactions processing;
- Decentralized governance for asset management decisions.

While the prototype is grounded in a conceptual framework found in existing literature [71], several design adaptations were introduced during implementation. For instance, although the original model recommended the ERC-777 token standard for its advanced features (e.g., hooks and composability), the prototype employs ERC-20 tokens. This change reflects the removal of ERC-777 from certain development repositories and recognizes ERC-20 as a widely supported and functionally adequate alternative.

Furthermore, several role definitions and process responsibilities were ambiguously defined in the conceptual model. For example:

- **Who initiates SPV creation?** The original model does not specify. In the prototype, the platform initiates SPV creation following successful off-chain verification of the property owner.

- **What is the asset owner's role post-SPV creation?** The model provides no clear guidance. In this implementation, the asset owner acts as the initial proposer and document provider but does not directly manage the SPV.

- **Who manages the SPV?** Management is abstracted within the smart contract, simplifying real-world corporate structures where such responsibility might lie with a legal entity, board, or third-party operator.

5.3. Workflow

The implementation assumes a structured interaction paradigm where participants engage with the system through defined procedural steps. These assumptions reflect both regulatory realities and the need for systematic validation of each system component.

Users progress through distinct phases: registration, verification, discovery, investment, and governance participation. This linearization simplifies implementation while preserving the essential characteristics of real-world investment processes.

5.3.1. Registration of Entities

Both property owners and investors must register on the platform. The platform requires all participants to undergo **Know Your Customer (KYC)** and **Anti-Money Laundering (AML)** checks through a third-party service provider. This assumption reflects current securities law requirements rather than idealized decentralized scenarios, acknowledging that practical blockchain applications must operate within existing legal frameworks.

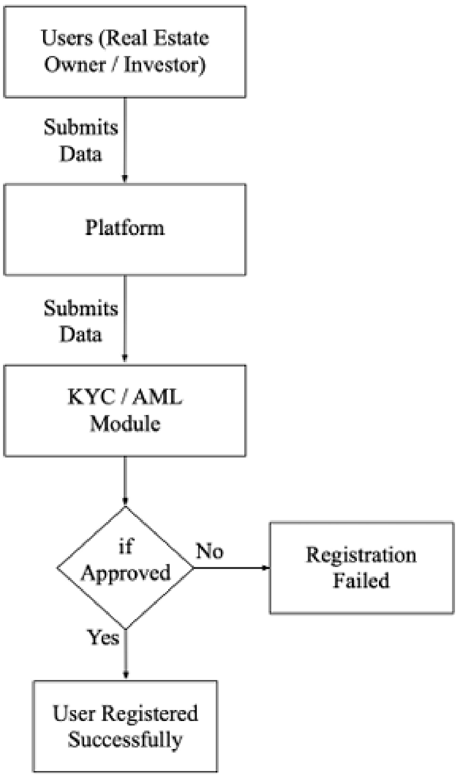


Figure 5.1 Regitration of Users on the platforme [71]

5.3.2. Creation of a Special Purpose Vehicle (SPV)

Each real estate asset is associated with a separate Special Purpose Vehicle (SPV)—a legal entity established to own and manage the asset. This mirrors traditional investment structures that separate asset risk and liability. Once the property owner is verified and submits required documentation, the platform initiates SPV creation and verification through jurisdictional authorities (mocked in the prototype for simulation).

The SPV becomes the legal owner of the asset and is responsible for operational tasks such as leasing, renting, or hotel management. Legal and practical constraints in most jurisdictions currently prohibit direct tokenization of physical property; hence, tokenization occurs at the SPV level, not the asset itself.

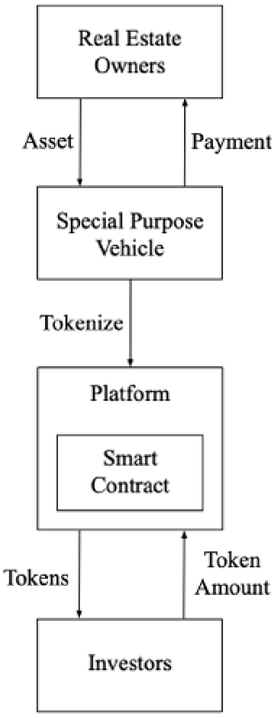


Figure 5.2 Tokenization of Real Estate [71]

5.3.3. Security Token Offering (STO)

Upon successful creation and verification of the SPV, the entity is tokenized. The issued ERC-20 tokens represent fractional ownership of the SPV and, by extension, indirect ownership of the underlying real estate asset. These tokens are embedded in a smart contract that encapsulates business logic for ownership transfers, KYC enforcement, and investment tracking.

The tokenized shares are issued via a Security Token Offering (STO). In contrast to ICOs, the STO is legally backed by a real-world asset, with each offering listing detailed investment information such as asset location, valuation, token supply, and expected yield.

If the STO meets its funding target, tokens are distributed to investors, and the SPV assumes legal title to the asset. If unsuccessful, investor funds are refunded, and the property owner retains the title. These tokens are considered securities and are subject to oversight under regulations such as the SEC's Howey Test or guidelines from the FCA.

5.3.4. Distribution of Dividends to Investors

Revenue generated by the SPV—such as rent, hotel income, or other operational profits—is periodically distributed to token holders. A profit distribution smart contract automates this process by:

- Calculating each holder's proportional share
- Transferring dividends directly to verified accounts

This automation ensures transparency, reduces manual errors, and mitigates fraud risks. The system supports extensibility to incorporate additional features such as automatic reinvestment, escrowed funds for maintenance, or tax reporting modules.

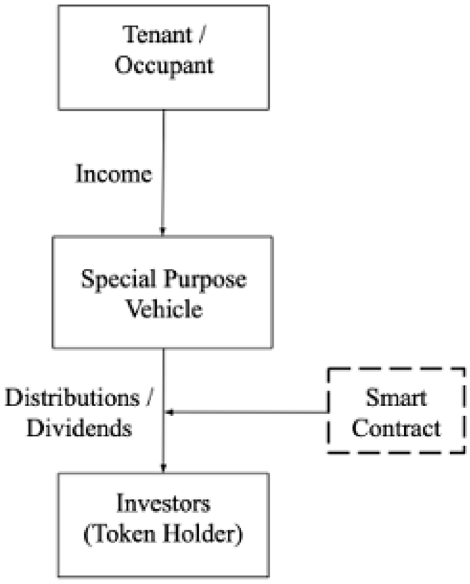


Figure 5.3 Distribution of Dividends to Investors [71]

5.3.5. Proportional Governance Rights

Governance mechanisms are embedded to allow token holders to influence key decisions, such as asset rental agreements or revenue reinvestment policies. Governance follows a proportional voting model, where influence is weighted by the number of tokens held. This reflects conventional shareholder structures and avoids the complexities of one-person-one-vote mechanisms typically found in DAOs.

5.4. Development Environment

5.4.1. IDE & Extensions

- **VS Code** with Solidity development plugins:
    - Solidity (Juan Blanco) - Syntax highlighting, compilation
    - Hardhat for Visual Studio Code - Integrated testing/debugging
    - Solidity Visual Developer - Contract visualization
    - Prettier Solidity - Code formatting

5.4.2. Smart Contract Development

- **Solidity 0.8.28** - Latest stable version with enhanced security features
- **Hardhat** - Development framework for:
    - Local blockchain deployment (chainId: 31337)
    - Automated testing with Chai/Mocha
    - Contract compilation with optimizer (200 runs)
    - Gas usage analysis

Solidity is a statically typed, contract-oriented programming language used to write smart contracts on Ethereum and other EVM-compatible blockchains. It allows developers to define complex logic, manage blockchain state, and interact with decentralized applications using a JavaScript-like syntax [72].

Hardhat is a versatile Ethereum development environment that streamlines smart contract development by providing built-in tools for compiling, testing, debugging, and deploying Solidity code. It supports a local blockchain for testing, integrates a powerful plugin system, and simplifies the development workflow with features like stack traces and automated error reporting [73].

CHAPTER 6: IMPLEMENTATION

6.1. Introduction

This chapter presents the practical implementation of a blockchain-based real estate tokenization platform designed to enable fractional ownership of real estate assets through security tokens. The implementation translates the theoretical framework established in previous chapters into a functional smart contract ecosystem built on the Ethereum blockchain. The development process prioritizes the technical validation of core tokenization concepts over regulatory and other external considerations.

This chapter outlines the system architecture, security measures, and testing results that demonstrate the technical viability of real estate tokenization.

6.2. System Architecture Overview

This implementation demonstrates a multi-contract ecosystem for real estate tokenization following the Security Token Offering (STO) paradigm. The system employs four primary smart contracts working in concert to create a complete real estate investment ecosystem:

- *RealEstateTokenizationPlatform* Contract
- *SpecialPurposeVehicle* (SPV) Contract
- *RealEstateSecurityToken* Contract
- *RealEstateDAO* Contract

And two mock contracts used to test a tokenized asset platform:

- *MockKYCProvider*
- *SecondaryMarket* Contract

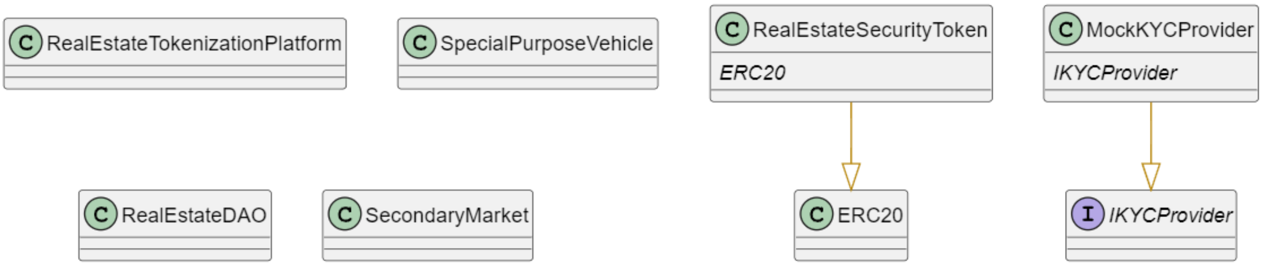


Figure 6.1 contract ecosystem overview.

6.2.1. RealEstateTokenizationPlatform Contract

This contract represents a central ecosystem coordinator managing user onboarding, KYC verification, asset registration, and STO lifecycle. Features robust access controls, reentrancy guards, and plausibility. Core components include:

1. **User Management:** Implements a two-stage identity verification process, progressing from user registration to KYC validation.
2. **Asset Registration:** Enables the deployment of property-specific Special Purpose Vehicle (SPV) contracts to ensure legal segregation and individualized asset governance.
3. **STO Management:** Facilitates the creation of tokenized offerings with supply determined by underlying asset valuations. Includes built-in compliance with offering structure and allocation logic.
4. **Token Purchase**: Enforces multi-layer validation for purchases, including KYC status verification, investment caps, and precision in wei-based arithmetic to maintain financial correctness.

6.2.2. SpecialPurposeVehicle (SPV) Contract

This contract represents Aan asset-level management module that handles revenue intake and dividend distribution for individual properties.

1. **Revenue Processing:** The contract accepts ETH through the receiveRevenue() function and maintains a record of distributions over time.
2. **Token Operations:** It supports token operations aligned with 18-decimal ERC-20 standards, ensuring accurate investor allocations.
3. **Dividend Claims:** Dividend calculations exclude tokens held by the SPV itself, preserving fair distribution based on circulating supply.
4. **Real-time Tracking:** The getPendingDividends() function allows real-time access to each holder’s claimable dividend amount.

6.2.3. RealEstateSecurityToken Contract

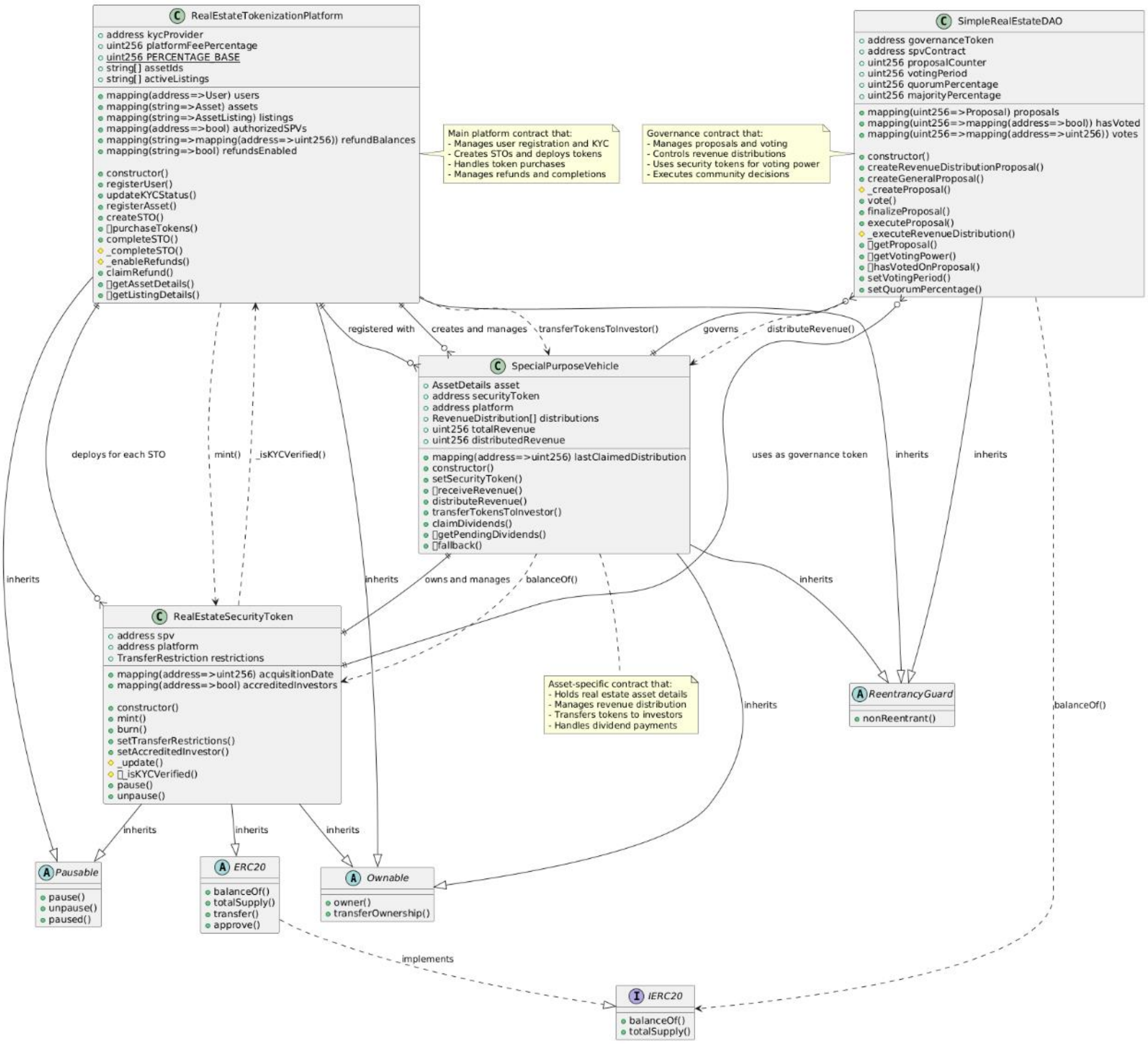
ERC-20 compliant security token with regulatory transfer restrictions and compliance enforcement. Key Features include:

1. **Transfer Controls:** Enforces regulatory requirements by validating KYC status, minimum holding periods, transfer limits, and accredited investor status prior to token transfers.
2. **Lifecycle Management:** Restricts minting and burning operations to pre-authorized contracts, preserving the integrity of supply control throughout the token lifecycle.
3. **Configuration:** Supports adjustable regulatory parameters, allowing simulation of evolving compliance rules and jurisdiction-specific constraints.

6.2.4. RealEstateDAO Contract

This contract represents a governance module that enables token holders to participate in decision-making related to revenue distribution and asset management;

1. **Proposal Types**: The contract supports two proposal types—revenue distribution proposals with executable data and general governance proposals.
2. **Voting System**: The voting mechanism uses token-weighted ballots with safeguards against duplicate submissions and includes real-time vote tracking.
3. **Execution Requirements**: Execution of approved proposals requires a minimum quorum of 30% and an approval threshold of 51%, both calculated based on circulating token supply. Upon meeting these conditions, proposals execute automatically with protection against reentrancy and integration with SPV structures.



Main platform contract that:
- Manages user registration and KYC
- Creates STOs and deploys tokens
- Handles token purchases
- Manages refunds and completions

Governance contract that:
- Manages proposals and voting
- Controls revenue distributions
- Uses security tokens for voting power
- Executes community decisions

Asset-specific contract that:
- Holds real estate asset details
- Manages revenue distribution
- Transfers tokens to investors
- Handles dividend payments

Figure 6.2 Smart Contract Architecture.

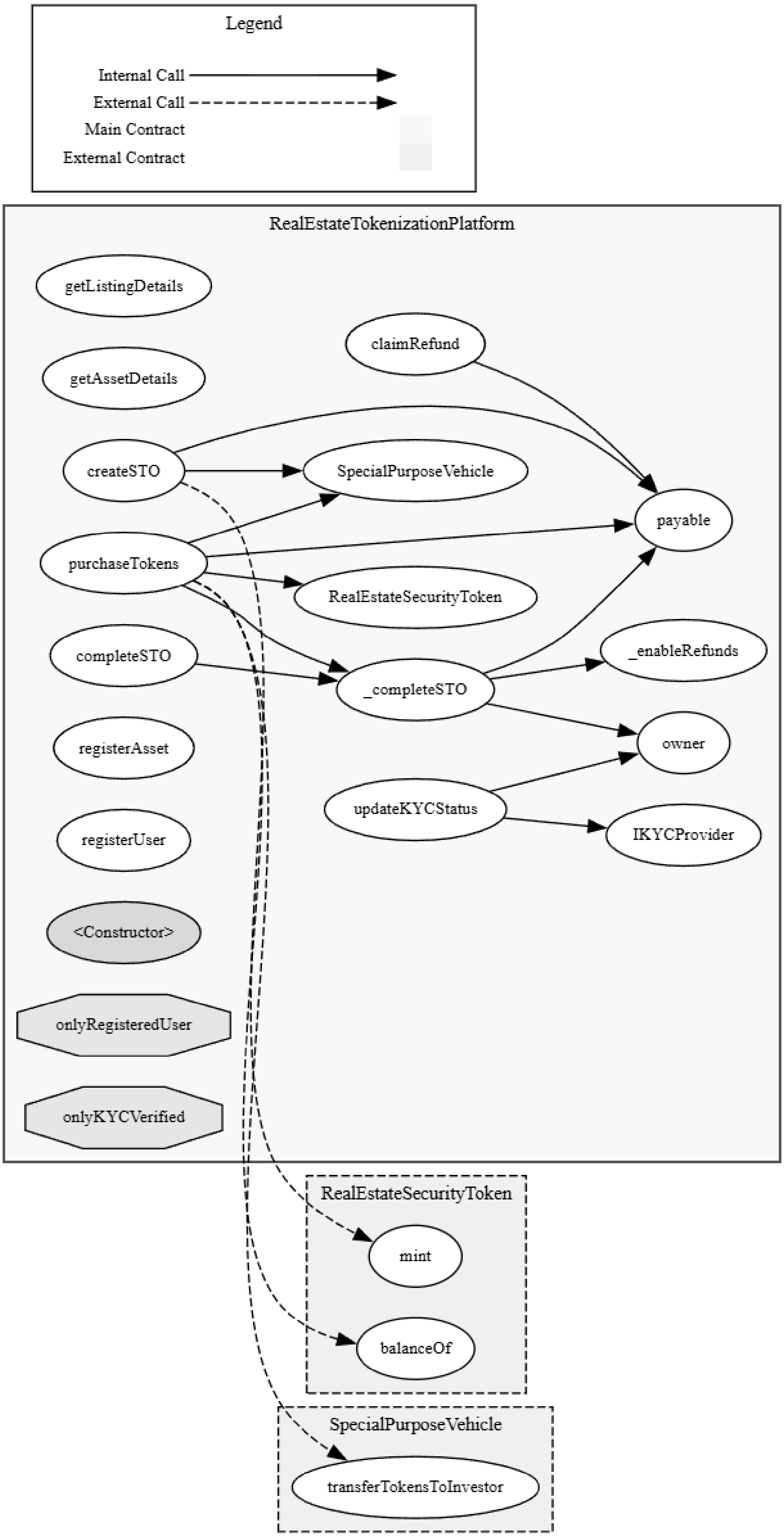


Figure 6.3 Smart Contract Interaction Graph.

6.3. Security Considerations

This section outlines the basic security measures implemented in the prototype. While sufficient for initial testing, these controls are limited in scope and not intended for production use.

6.3.1. Role-Based Access Control

The system defines specific roles with distinct permissions:

- **Platform Owner:** Asset registration, STO creation, system configuration
- **SPV Contracts:** Token minting, revenue distribution execution
- **KYC Provider:** User verification status updates
- **Token Holders:** Governance voting, dividend claiming

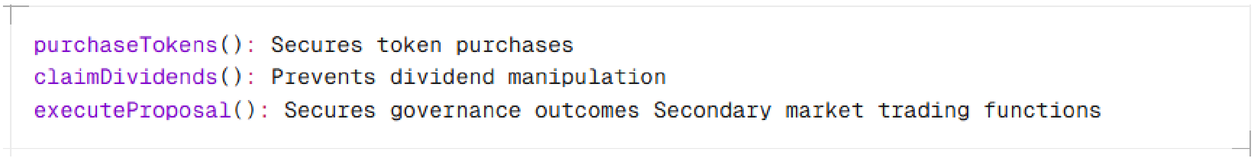
6.3.2. Modifier-Based Authorization

Custom modifiers restrict sensitive functions:

```
modifier onlyKYCVerified() {
    require(users[msg.sender].isKYCVerified, "KYC verification required");
    _;
}
modifier onlyAuthorized() {
    require(msg.sender == platform || msg.sender == spv, "Unauthorized");
    _;
}
```

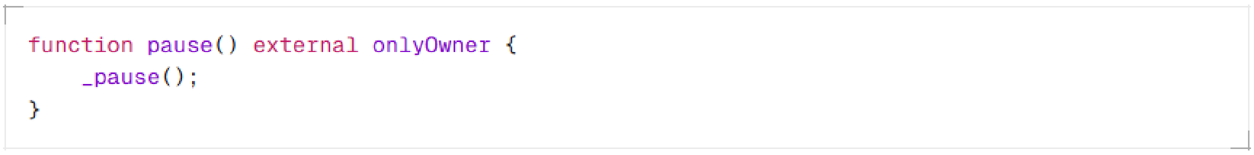
6.3.3. Reentrancy Protection

Functions handling ETH or external calls use OpenZeppelin's *ReentrancyGuard*. The *nonReentrant* modifier protects key functions:



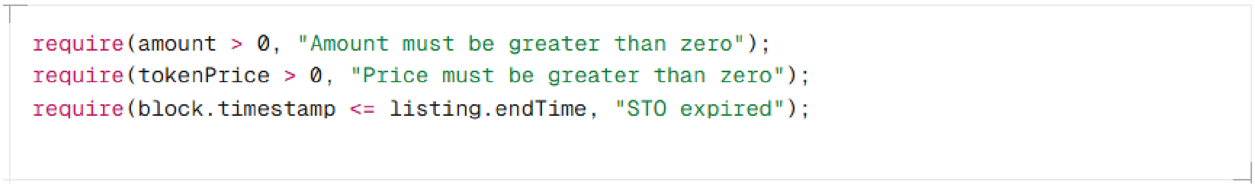
```
purchaseTokens(): Secures token purchases
claimDividends(): Prevents dividend manipulation
executeProposal(): Secures governance outcomes Secondary market trading functions
```

6.3.4. Emergency pause



```
function pause() external onlyOwner {
    _pause();
}
```

6.3.5. Input Validation and Sanitization

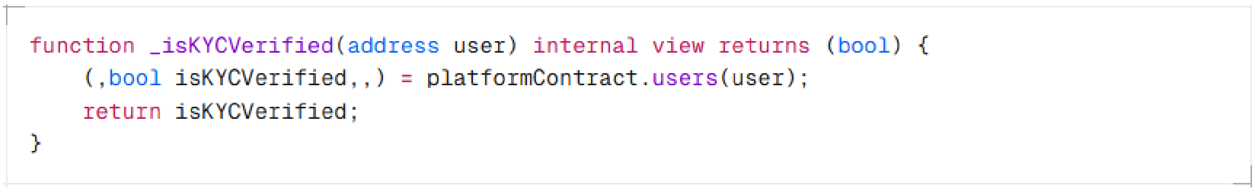


```
require(amount > 0, "Amount must be greater than zero");
require(tokenPrice > 0, "Price must be greater than zero");
require(block.timestamp <= listing.endTime, "STO expired");
```

6.3.6. Transfer Restriction Enforcement:

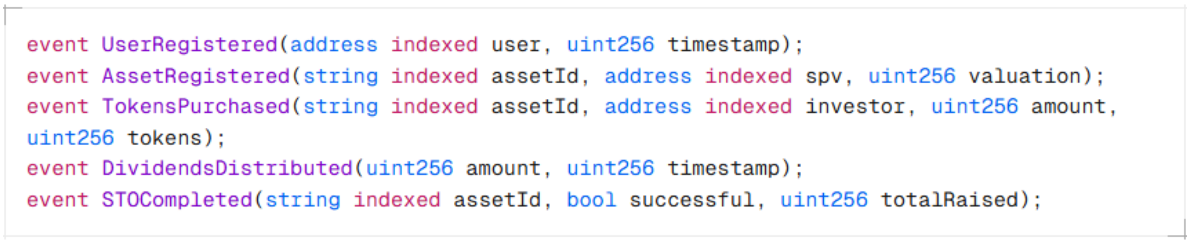
```
function setTransferRestrictions(
    bool _requiresKYC,
    uint256 _minHoldingPeriod,
    uint256 _maxTransferAmount,
    bool _accreditedOnly
) external onlyOwner {
    restrictions.requiresKYC = _requiresKYC;
    restrictions.minHoldingPeriod = _minHoldingPeriod;
    restrictions.maxTransferAmount = _maxTransferAmount;
    restrictions.accreditedInvestorsOnly = _accreditedOnly;

    emit TransferRestrictionUpdated();
}
```

6.3.7. KYC Enforcement:



```
function _isKYCVerified(address user) internal view returns (bool) {
    (,bool isKYCVerified,,) = platformContract.users(user);
    return isKYCVerified;
}
```

6.3.8. Event Logging: Key actions emit events for traceability:



```
event UserRegistered(address indexed user, uint256 timestamp);
event AssetRegistered(string indexed assetId, address indexed spv, uint256 valuation);
event TokensPurchased(string indexed assetId, address indexed investor, uint256 amount,
uint256 tokens);
event DividendsDistributed(uint256 amount, uint256 timestamp);
event STOCompleted(string indexed assetId, bool successful, uint256 totalRaised);
```

The current security measures are minimal and serve prototype purposes only. role-based access control, reentrancy protection, input validation, and overflow checks—these implementations are intentionally basic and serve primarily to support the prototype stage of development. This minimal security posture allows for faster iteration during early testing but should not be considered production-grade.

6.4. Testing

Deployment scripts were written in JavaScript using Hardhat's deployment framework, enabling named accounts, deterministic addresses for testing, and environment-specific configuration management. This approach mirrors production network deployment, supporting reproducibility and automation.

6.4.1. Contract Deployment Order

Table 6.1 Contract Deployment Order.

Step	Contract	Post-Deploy Configuration
1	MockKYCProvider	Set investor KYC status and levels
2	RealEstateTokenizationPlatform	Link to KYC provider
3	SpecialPurposeVehicle	Auto-deployed via platform during asset registration
4	RealEstateSecurityToken	Auto-deployed during STO creation, linked to SPV
5	SecondaryMarket	Authorize tokens for trading
6	SimpleRealEstateDAO	Link to token and SPV contracts

6.4.2. Configuration

The deployment process uses Hardhat's *ethers.getContractFactory()* and automated contract linking:

```javascript
// Deploy KYC Provider
const MockKYCProvider = await ethers.getContractFactory("MockKYCProvider");
kycProvider = await MockKYCProvider.deploy();

// Deploy Platform with KYC reference
const Platform = await ethers.getContractFactory("RealEstateTokenizationPlatform");
platform = await Platform.deploy(kycProvider.target);
```

Each deployment includes comprehensive validation through test suites covering. We conducted comprehensive automated testing using the Hardhat framework to verify the correctness, security, and regulatory compliance of the platform’s smart contracts. Tests covered all major functionalities, including KYC logic, token issuance, revenue distribution, governance execution, and compliance enforcement.

6.4.3. Key Test Cases

Test Case 1: Asset Tokenization and STO Creation

- **Scenario:** Register $1000 property, create STO with 100 token targets at $1 per token.
- **Expected Result:** Asset registered with SPV deployed, STO active with token contract deployed
- **Outcome:** Asset successfully tokenized with proper valuation and active STO listing

```
// Asset Tokenization
await platform.registerAsset(assetId,"New York, NY",
ethers.parseEther("1000"),"ipfs://metadata");
          const asset = await platform.getAssetDetails(assetId);
          expect(asset.active).to.be.true;
          expect(asset.spv).to.not.equal(ethers.ZeroAddress);
// STO Creation
await platform.createSTO(
assetId, ethers.parseEther("100"), ethers.parseEther("1"),   86400 * 30,
"Real Estate Token", "RET");
      const listing = await platform.getListingDetails(assetId);
      expect(listing.isActive).to.be.true;
      expect(listing.tokenAddress).to.not.equal(ethers.ZeroAddress);
```

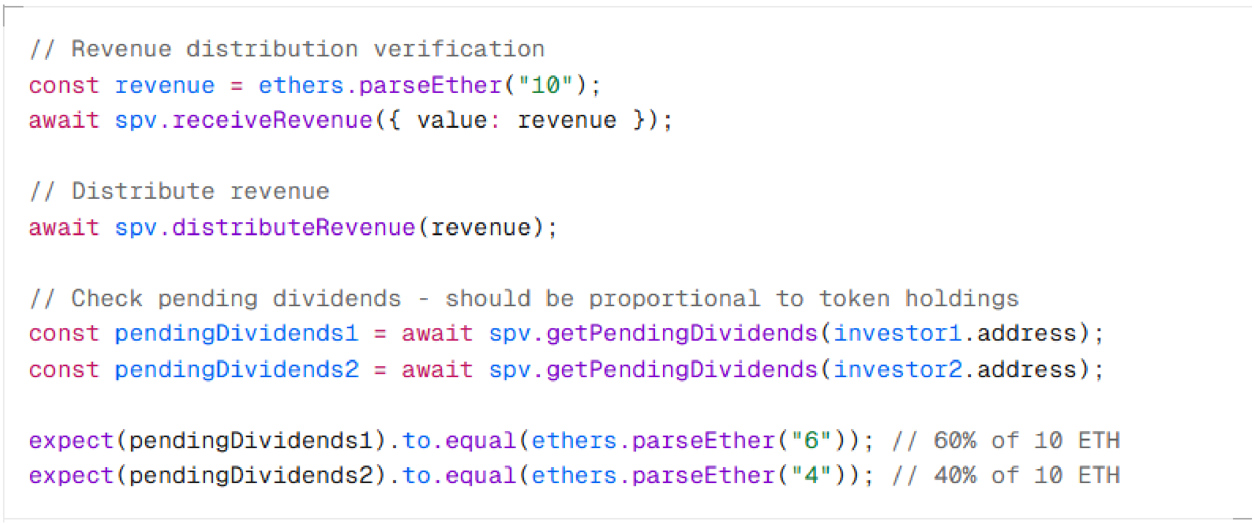
Test Case 2: Multi-Investor Token Purchase and STO Completion

- **Scenario:** Two KYC-verified investors purchase 60 and 40 tokens respectively to reach 100 token target Expected Result: Both investors receive proportional tokens, STO completes when target reached
- **Outcome:** Token purchases processed correctly, STO automatically completed at target

```
// First investor purchases 60 tokens
await platform.connect(investor1).purchaseTokens(assetId, {value: Amount1 });
// Second investor purchases 40 tokens
await platform.connect(investor2).purchaseTokens(assetId, {value: Amount2 });
// Token allocation and STO completion verification
expect(balance1).to.equal(ethers.parseEther("60")); // 60 tokens (60%)
expect(balance2).to.equal(ethers.parseEther("40")); // 40 tokens (40%)
expect(listing.isActive).to.be.false;
expect(listing.fundingSuccessful).to.be.true;
```

Test Case 3: Multi-Investor Revenue Distribution

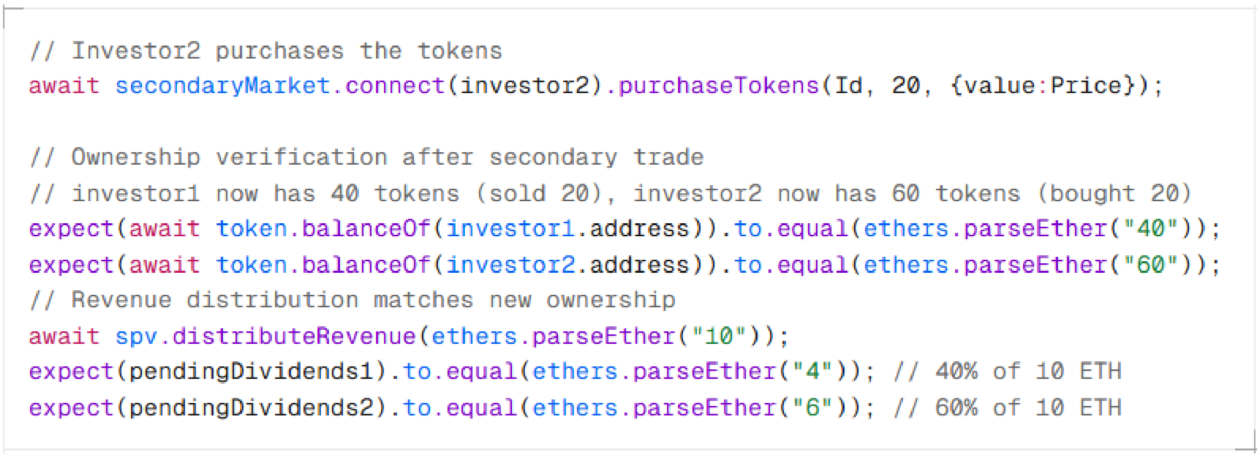
- **Scenario:** 10 ETH revenue distributed between investor1 (60% ownership) and investor2 (40% ownership)
- **Expected Result:** investor1 receives 6 ETH, investor2 receives 4 ETH
- **Outcome:** Dividends calculated and distributed accurately based on token holdings



```
// Revenue distribution verification
const revenue = ethers.parseEther("10");
await spv.receiveRevenue({ value: revenue });

// Distribute revenue
await spv.distributeRevenue(revenue);

// Check pending dividends - should be proportional to token holdings
const pendingDividends1 = await spv.getPendingDividends(investor1.address);
const pendingDividends2 = await spv.getPendingDividends(investor2.address);

expect(pendingDividends1).to.equal(ethers.parseEther("6")); // 60% of 10 ETH
expect(pendingDividends2).to.equal(ethers.parseEther("4")); // 40% of 10 ETH
```

Test Case 4: Secondary Trading Impact on Revenue Rights

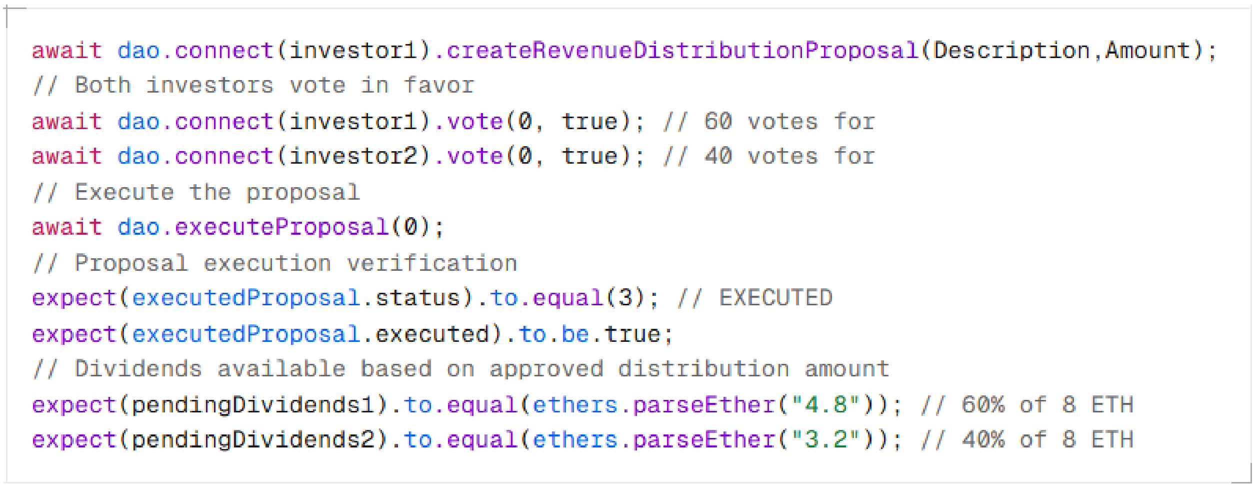
- **Scenario:** investor1 sells 20 tokens to investor2, then 10 ETH revenue is distributed
- **Expected Result:** Revenue distribution reflects new ownership (investor1: 40%, investor2: 60%)
- **Outcome:** Post-trade dividend calculations updated automatically based on current token balances



```
// Investor2 purchases the tokens
await secondaryMarket.connect(investor2).purchaseTokens(Id, 20, {value:Price});

// Ownership verification after secondary trade
// investor1 now has 40 tokens (sold 20), investor2 now has 60 tokens (bought 20)
expect(await token.balanceOf(investor1.address)).to.equal(ethers.parseEther("40"));
expect(await token.balanceOf(investor2.address)).to.equal(ethers.parseEther("60"));
// Revenue distribution matches new ownership
await spv.distributeRevenue(ethers.parseEther("10"));
expect(pendingDividends1).to.equal(ethers.parseEther("4")); // 40% of 10 ETH
expect(pendingDividends2).to.equal(ethers.parseEther("6")); // 60% of 10 ETH
```

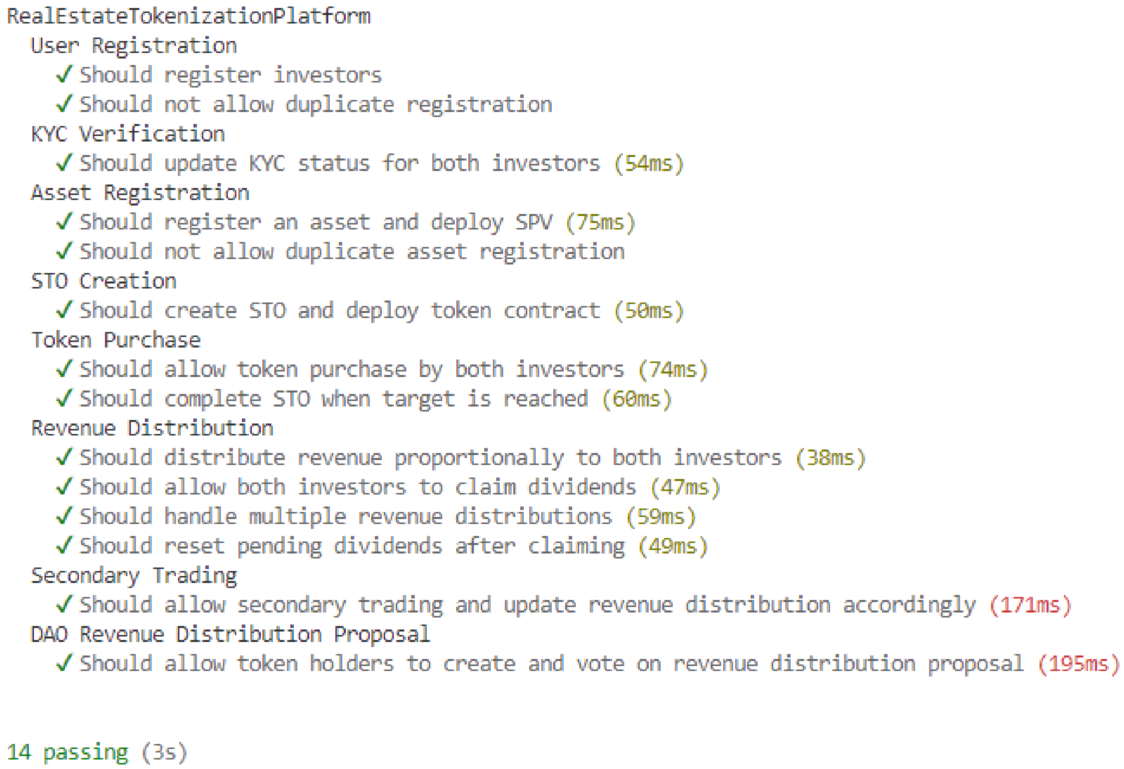
Test Case 5: DAO Governance Proposal Execution

- **Scenario:** Token holders propose and vote on 8 ETH revenue distribution from 10 ETH available
- **Expected Result:** Proposal passes with majority vote, executes successfully, dividends become claimable.
- **Outcome:** Complete governance cycle executed with proper quorum validation and dividend distribution



```
await dao.connect(investor1).createRevenueDistributionProposal(Description,Amount);
// Both investors vote in favor
await dao.connect(investor1).vote(0, true); // 60 votes for
await dao.connect(investor2).vote(0, true); // 40 votes for
// Execute the proposal
await dao.executeProposal(0);
// Proposal execution verification
expect(executedProposal.status).to.equal(3); // EXECUTED
expect(executedProposal.executed).to.be.true;
// Dividends available based on approved distribution amount
expect(pendingDividends1).to.equal(ethers.parseEther("4.8")); // 60% of 8 ETH
expect(pendingDividends2).to.equal(ethers.parseEther("3.2")); // 40% of 8 ETH
```

6.4.4. Tests execution results

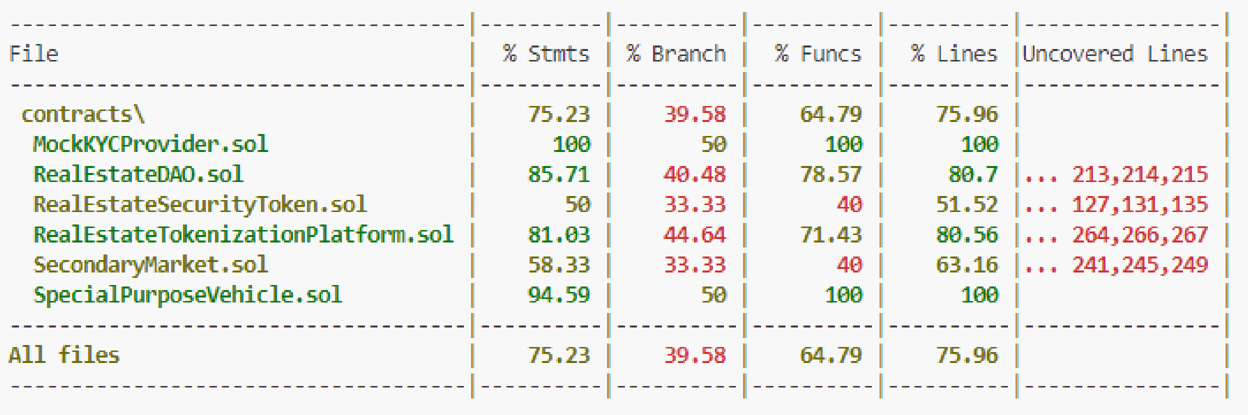


```
RealEstateTokenizationPlatform
  User Registration
    ✓ Should register investors
    ✓ Should not allow duplicate registration
  KYC Verification
    ✓ Should update KYC status for both investors (54ms)
  Asset Registration
    ✓ Should register an asset and deploy SPV (75ms)
    ✓ Should not allow duplicate asset registration
  STO Creation
    ✓ Should create STO and deploy token contract (50ms)
  Token Purchase
    ✓ Should allow token purchase by both investors (74ms)
    ✓ Should complete STO when target is reached (60ms)
  Revenue Distribution
    ✓ Should distribute revenue proportionally to both investors (38ms)
    ✓ Should allow both investors to claim dividends (47ms)
    ✓ Should handle multiple revenue distributions (59ms)
    ✓ Should reset pending dividends after claiming (49ms)
  Secondary Trading
    ✓ Should allow secondary trading and update revenue distribution accordingly (171ms)
  DAO Revenue Distribution Proposal
    ✓ Should allow token holders to create and vote on revenue distribution proposal (195ms)


14 passing (3s)
```

Figure 6.4 Tests execution results.

6.4.5. Coverage Summary

This report shows moderate overall coverage (75% statements, 76% lines), with key contracts well-tested. *RealEstateTokenizationPlatform* and *SpecialPurposeVehicle*, which handle core functionality, have over 80% coverage. *SecondaryMarket* has lower coverage, which is acceptable as it is a mock used to test post-tokenization trading. Coverage of other modules is partial, with branch testing needing improvement. For a prototype, test focus is appropriate, but broader coverage will be needed for production.

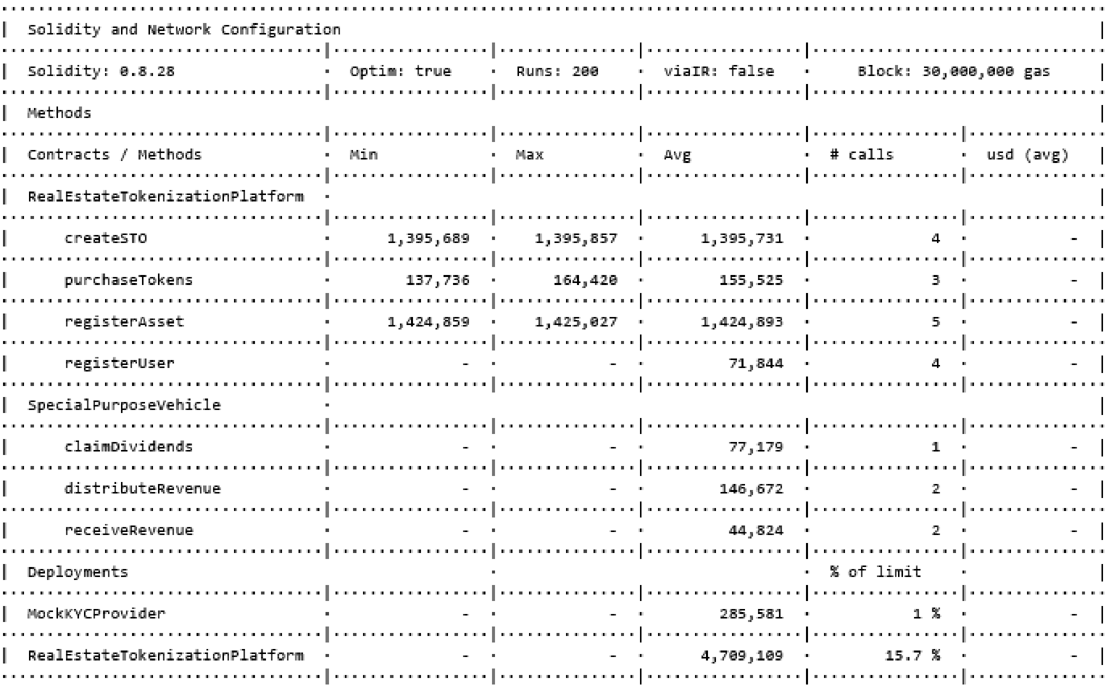


```
-------------------------------------|----------|----------|----------|----------|----------------|
File                                 |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
-------------------------------------|----------|----------|----------|----------|----------------|
 contracts\                          |    75.23 |    39.58 |    64.79 |    75.96 |                |
  MockKYCProvider.sol                |      100 |       50 |      100 |      100 |                |
  RealEstateDAO.sol                  |    85.71 |    40.48 |    78.57 |     80.7 |... 213,214,215 |
  RealEstateSecurityToken.sol        |       50 |    33.33 |       40 |    51.52 |... 127,131,135 |
  RealEstateTokenizationPlatform.sol |    81.03 |    44.64 |    71.43 |    80.56 |... 264,266,267 |
  SecondaryMarket.sol                |    58.33 |    33.33 |       40 |    63.16 |... 241,245,249 |
  SpecialPurposeVehicle.sol          |    94.59 |       50 |      100 |      100 |                |
-------------------------------------|----------|----------|----------|----------|----------------|
All files                            |    75.23 |    39.58 |    64.79 |    75.96 |                |
-------------------------------------|----------|----------|----------|----------|----------------|
```

Figure 6.5 Tests Coverage results.

6.4.6. Gas consumption

This gas report provides an overview of gas consumption across key functions within the prototype smart contracts, with optimizations enabled (200 runs, no IR), and block gas limit set to 30 million.



```
························································································
|  Solidity and Network Configuration                                                  |
·································|················|··············|················|·····················
|  Solidity: 0.8.28             ·  Optim: true   ·  Runs: 200   ·  viaIR: false  ·     Block: 30,000,000 gas  |
·································|················|··············|················|·····················
|  Methods                                                                             |
·································|················|··············|················|·············|··············
|  Contracts / Methods           ·  Min           ·  Max         ·  Avg           ·  # calls    ·  usd (avg)  |
·································|················|··············|················|·············|··············
|  RealEstateTokenizationPlatform ·                                                    |
·································|················|··············|················|·············|··············
|      createSTO                 ·     1,395,689  ·   1,395,857  ·     1,395,731  ·          4  ·           -  |
·································|················|··············|················|·············|··············
|      purchaseTokens            ·       137,736  ·     164,420  ·       155,525  ·          3  ·           -  |
·································|················|··············|················|·············|··············
|      registerAsset             ·     1,424,859  ·   1,425,027  ·     1,424,893  ·          5  ·           -  |
·································|················|··············|················|·············|··············
|      registerUser              ·             -  ·           -  ·        71,844  ·          4  ·           -  |
·································|················|··············|················|·············|··············
|  SpecialPurposeVehicle         ·                                                     |
·································|················|··············|················|·············|··············
|      claimDividends            ·             -  ·           -  ·        77,179  ·          1  ·           -  |
·································|················|··············|················|·············|··············
|      distributeRevenue         ·             -  ·           -  ·       146,672  ·          2  ·           -  |
·································|················|··············|················|·············|··············
|      receiveRevenue            ·             -  ·           -  ·        44,824  ·          2  ·           -  |
·································|················|··············|················|·············|··············
|  Deployments                                   ·                                ·  % of limit  ·             |
·································|················|··············|················|·············|··············
|  MockKYCProvider               ·             -  ·           -  ·       285,581  ·        1 %  ·           -  |
·································|················|··············|················|·············|··············
|  RealEstateTokenizationPlatform ·            -  ·           -  ·     4,709,109  ·     15.7 %  ·           -  |
·································|················|··············|················|·············|··············
```

Figure 6.6 Gas consumption.

Deployment of the main contract, RealEstateTokenizationPlatform, consumes around 4.7 million gas, or 15.7% of the block limit, which is acceptable for prototype deployment. The most expensive function calls are registerAsset and createSTO, each using approximately 1.4 million gas. These are setup operations and are expected to be executed infrequently.

Operational functions such as purchaseTokens, claimDividends, and distributeRevenue are more moderate in cost, ranging from roughly 77,000 to 155,000 gas. Lighter interactions like receiveRevenue and registerUser fall below 80,000 gas.

Overall, the gas usage is consistent with expectations for a prototype. While some functions may be optimized further, the current costs are acceptable for development and testing purposes.

6.5. Conclusion

This chapter presented the implementation of a blockchain-based real estate tokenization platform, demonstrating the technical feasibility of fractional property ownership through security tokens.

The platform's multi-contract architecture efficiently manages complex real estate investment workflows through four coordinated contracts. through coordinating user and asset onboarding, enabling asset-level management with automated dividend distribution, facilitating democratic governance among token holders, and ensuring compliance throughout the token lifecycle.

Testing validated system reliability through comprehensive coverage achieving 75% overall and exceeding 80% in critical modules. Functional requirements were verified using basic realistic scenarios including Security Token Offerings, multi-investor dividend calculations, and governance execution. Gas analysis shows acceptable costs, with main contract deployment at 4.7 million gas (15.7% of block limits) and routine operations consuming 77,000-155,000 gas, demonstrating operational efficiency appropriate for complex decentralized applications.

However, limitations remain. Current security measures are minimal and insufficient for production use. The platform lacks integration with external data sources, regulatory reporting, and real compliance mechanisms required for real-world deployment. These gaps highlight the distinction between technical feasibility and production readiness, reinforcing the need for further work to meet real-world deployment standards. Despite these constraints, the prototype demonstrates that a minimally viable transactional core can effectively support real estate tokenization under realistic technical and procedural assumptions.

CHAPTER 7: GENERAL CONCLUSION

7.1. Research Summary

This thesis investigated the technical feasibility of blockchain-based real estate tokenization as a solution to structural inefficiencies in traditional property investment markets. Through the design and implementation of a comprehensive smart contract ecosystem, the research demonstrated that blockchain technology can effectively address the critical challenges, that characterize conventional real estate investment.

The study followed an prototype-based exploratory research approach, prioritizing practical implementation over theoretical modeling to validate the viability of tokenization. The implementation successfully validates the core proposition that blockchain technology can preserve essential market functions.

7.2. Key Findings

7.2.1. Technical Feasibility

The smart contract suite demonstrates end-to-end functionality for property tokenization, covering user onboarding, STOs, automated dividends, and governance. Testing achieved 75% code coverage, with critical modules exceeding 80%. Gas analysis confirmed operational efficiency, with routine transactions consuming 77,000–155,000 gas and initial deployment requiring 4.7 million gas—well within acceptable limits for complex dApps. While focused on a minimal viable core, the framework is generalizable and can support broader use cases such as multi-asset portfolios, tiered investor rights, dynamic compliance logic, or integration with off-chain systems. This indicates strong potential for applying the same architectural principles to a wider class of real estate and financial applications.

7.2.2. Operational Efficiency

Automation reduces transaction costs, errors, and delays. Dividend distributions and governance are streamlined, providing transparency and investor control without intermediaries. The platform currently supports multi-investor scenarios, secondary trading, and governance via smart contracts. Beyond the implemented features, the framework is extensible to a wider range of real estate functions—such as rental income sharing, property management workflows, asset refinancing, and insurance claims processing. These could be similarly modeled, implemented, and validated, further demonstrating the capacity of smart contracts to encapsulate complex real estate operations with precision and auditability.

7.3. Limitations

7.3.1. Security

The current implementation incorporates basic security measures sufficient for prototype validation but requires substantial enhancement for production deployment. The minimal security framework, while adequate for demonstrating core functionality, lacks the comprehensive protection mechanisms necessary for handling real-world investment capital. Critical areas requiring enhancement include advanced access controls, multi-signature wallet integration, formal security auditing, and robust emergency response mechanisms.

The system's reliance on basic role-based access control and standard reentrancy protection represents a foundation that must be significantly expanded to meet institutional security requirements. The absence of comprehensive threat modeling and formal verification processes limits the current implementation's suitability for real-world deployment without substantial security enhancement.

7.3.2. Compliance Scope

This work operates within a simplified compliance framework that does not address the full complexity of securities regulation across multiple jurisdictions. The system lacks integration with external regulatory reporting mechanisms, cross-jurisdictional compliance variations, and sophisticated dispute resolution procedures that would be essential for regulatory approval in most markets.

The platform's current approach to Know Your Customer verification relies on mock implementations that simulate rather than integrate with actual regulatory infrastructure. Real-world deployment would require comprehensive integration with certified KYC providers, regulatory reporting systems, and jurisdiction-specific compliance frameworks that extend beyond the current technical scope.

7.3.3. Market Readiness

The system's current testing environment does not simulate real market conditions, including price volatility, liquidity constraints, and complex investor behavior patterns that would impact platform performance in actual deployment scenarios. These limitations prevent comprehensive evaluation of the platform's effectiveness under realistic market stress conditions.

7.4. Final Conclusions

This thesis successfully demonstrates that blockchain-based real estate tokenization represents a technically feasible and operationally viable approach to addressing structural inefficiencies in traditional property investment markets. The comprehensive smart contract implementation validates the core theoretical propositions while revealing important practical considerations for real-world deployment.

The research establishes that blockchain technology can effectively preserve essential market functions while reducing transactional friction, expanding access to investment opportunities, and enhancing operational transparency. A successful integration of regulatory requirements within smart contract architectures can prove that blockchain applications can operate within existing legal frameworks rather than requiring wholesale regulatory transformation.

While significant challenges remain in security enhancement, comprehensive regulatory integration, and market infrastructure development, the demonstrated technical feasibility provides a foundation for continued advancement in blockchain-based real estate applications. The prototype serves as both a validation of tokenization concepts and a reference artifact for future development efforts.

The implications of this research extend beyond technical validation to suggest that blockchain technology can play a transformative role in democratizing access to real estate investment while maintaining necessary regulatory oversight and operational integrity. As the technology matures and regulatory frameworks adapt, blockchain-based real estate tokenization has the potential to fundamentally reshape property investment markets by reducing barriers to entry, enhancing liquidity, and improving operational efficiency.

BIBLIOGRAPHY

[1] K. Sultan, U. Ruhi, and R. Lakhani, “Conceptualizing Blockchains: Characteristics & Applications,” *arXiv (Cornell University)*, Jan. 2018, doi: 10.48550/arxiv.1806.03693.

[2] S. Haber and W. S. Stornetta, “How to time-stamp a digital document,” *Journal of Cryptology*, vol. 3, no. 2, pp. 99–111, Jan. 1991, doi: 10.1007/bf00196791.

[3] C. Cachin and M. Vukolić, “Blockchain consensus protocols in the wild,” *arXiv (Cornell University)*, Jan. 2017, doi: 10.48550/arxiv.1707.01873.

[4] N. Satoshi, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. https://bitcoin.org/bitcoin.pdf [Online]

[5] B. Vitalik, “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.” https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[6] T. Hewa, M. Ylianttila, and M. Liyanage, “Survey on blockchain based smart contracts: Applications, opportunities and challenges,” *Journal of Network and Computer Applications*, vol. 177, p. 102857, Nov. 2020, doi: 10.1016/j.jnca.2020.102857.

[7] P. Mukherjee and C. Pradhan, “Blockchain 1.0 to Blockchain 4.0—The Evolutionary Transformation of Blockchain Technology,” in *Intelligent systems reference library*, 2021, pp. 29–49. doi: 10.1007/978-3-030-69395-4_3.

[8] S. Eswaran, “Blockchain Characteristics | Blockchain Technology Fundamentals,” *Witspry Witscad*, Dec. 07, 2019. https://witscad.com/course/blockchain-fundamentals/chapter/blockchain-characteristics (accessed Feb. 25, 2025).

[9] A. B. Baftijari and L. Nakov, “The architecture of Blockchain technology and Beyond,” in *IntechOpen eBooks*, 2024. doi: 10.5772/intechopen.1004138.

[10] M. N. Birje, G. R. H, R. C. M, and M. T. Tapale, “Blockchain Technology Review: Consensus Mechanisms and applications,” *International Journal of Engineering Trends and Technology*, vol. 71, no. 5, pp. 27–39, May 2023, doi: 10.14445/22315381/ijett-v71i5p204.

[11] G. Tripathi, M. A. Ahad, and G. Casalino, “A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges,” *Decision Analytics Journal*, vol. 9, p. 100344, Oct. 2023, doi: 10.1016/j.dajour.2023.100344.

[12] F. Anwar, B. U. I. Khan, L. B. M. Kiah, N. A. Abdullah, and K. W. Goh, “A Comprehensive Insight into Blockchain Technology: Past Development, Present Impact and Future Considerations,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 11, Jan. 2022, doi: 10.14569/ijacsa.2022.01311101.

[13] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain technology overview,” *Non*, Oct. 2018, doi: 10.6028/nist.ir.8202.

[14] P. Cui, U. Guin, A. Skjellum, and D. Umphress, “Blockchain in IoT: Current trends, challenges, and future roadmap,” *Journal of Hardware and Systems Security*, vol. 3, no. 4, pp. 338–364, Nov. 2019, doi: 10.1007/s41635-019-00079-5.

[15] S. Ahmad, S. K. Arya, S. Gupta, P. Singh, and S. K. Dwivedi, “Study of Cryptographic Techniques Adopted in Blockchain,” -, May 2023, doi: 10.1109/iciem59379.2023.10166591.

[16] H. R. Atiya and H. N. Nawaf, “Community structure-aware fairness and goodness algorithm for link weight prediction,” *Journal of Physics Conference Series*, vol. 1804, no. 1, p. 012080, Feb. 2021, doi: 10.1088/1742-6596/1804/1/012080.

[17] S. A and S. A, “A Brief Guide to Blockchain architecture - BLOCKGENI,” *BLOCKGENI*, Aug. 26, 2021. [Online]. Available: https://blockgeni.com/a-brief-guide-to-blockchain-architecture/

[18] S. Makani, R. Pittala, E. Alsayed, M. Aloqaily, and Y. Jararweh, “A survey of blockchain applications in sustainable and smart cities,” *Cluster Computing*, vol. 25, no. 6, pp. 3915–3936, May 2022, doi: 10.1007/s10586-022-03625-z.

[19] christina.reilly@829llc.com, “Types of blockchain: public, private, or something else | Foley & Lardner LLP,” *Foley & Lardner LLP*, Jul. 10, 2024. https://www.foley.com/insights/publications/2021/08/types-of-blockchain-public-private-between/ (accessed Mar. 03, 2025).

[20] F. Gedefaw, A. A. Dawit, J. Birara, and D. Andargachew Asmare, “Blockchain Technology: understanding its meaning, architecture, and diverse applications,” Master, Addis Ababa University, 2023. doi: 10.13140/RG.2.2.25588.32643/1.

[21] G. Baranwal, D. Kumar, and D. P. Vidyarthi, “Blockchain based resource allocation in cloud and distributed edge computing: A survey,” *Computer Communications*, vol. 209, pp. 469–498, Jul. 2023, doi: 10.1016/j.comcom.2023.07.023.

[22] “Blockchain: The biggest misconception about it - Phemex Academy,” *Phemex*. https://phemex.com/academy/what-is-blockchain-technology (accessed Mar. 10, 2025).

[23] Ledger, “What is Proof-of-Work (PoW)? | Ledger,” *Ledger*, Jul. 24, 2023. https://www.ledger.com/academy/blockchain/what-is-proof-of-work (accessed Mar. 13, 2025).

[24] Ledger, “What is Proof-of-Stake (PoS)?,” *Ledger*, Aug. 18, 2023. https://www.ledger.com/academy/blockchain/what-is-proof-of-stake (accessed Mar. 13, 2025).

[25] F. Anwar, B. U. I. Khan, L. B. M. Kiah, N. A. Abdullah, and K. W. Goh, “A Comprehensive Insight into Blockchain Technology: Past Development, Present Impact and Future Considerations,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 11, Jan. 2022, doi: 10.14569/ijacsa.2022.01311101.

[26] M. GmbH, “A walk through the evolution of DLTs: From Blockchain to Hashgraph and Tangle,” *Medium*, Apr. 26, 2018. [Online]. Available: https://micobo.medium.com/april-2018-3f575ef0cb31

[27] B. K. Mohanta, S. S. Panda, and D. Jena, “An overview of smart contract and use cases in blockchain technology,” *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–4, Jul. 2018, doi: 10.1109/icccnt.2018.8494045.

[28] S. Eswaran, “Smart Contract Concepts and Practice | Blockchain Technology Fundamentals,” *Witspry Witscad*, Dec. 07, 2019. https://witscad.com/course/blockchain-fundamentals/chapter/smart-contract-concepts (accessed Mar. 16, 2025).

[29] S. Namasudra and K. Akkaya, “Introduction to blockchain technology,” in *Studies in big data*, 2023, pp. 1–28. doi: 10.1007/978-981-19-8730-4_1.

[30] J. Holdsworth and M. Kosinski, “Tokenization,” *Think (IBM)*, Apr. 16, 2025. https://www.ibm.com/think/topics/tokenization (accessed Mar. 20, 2025).

[31] “Encryption vs. tokenization explained | Stripe,” Sep. 04, 2024. https://stripe.com/resources/more/encryption-vs-tokenization-how-they-are-different-and-how-they-work-together (accessed Mar. 16, 2025).

[32] “What is Tokenization? What Every Engineer Should Know - Skyflow.” https://www.skyflow.com/post/demystifying-tokenization-what-every-engineer-should-know (accessed Mar. 16, 2025).

[33] G. Wang and M. Nixon, “SoK: Tokenization on Blockchain,” -, pp. 1–9, Dec. 2021, doi: 10.1145/3492323.3495577.

[34] “Asset tokenization: how can you benefit from it? | Omertex Blog.” https://www.blog.omertex.com/asset-tokenization-business-benefits/ (accessed Mar. 20, 2025).

[35] R. Behnke, “Understanding tokenization Security (Part 1),” Jun. 26, 2023. https://www.halborn.com/blog/post/understanding-tokenization-security-part-1 (accessed Mar. 21, 2025).

[36] O. Konashevych, “General concept of real estate tokenization on Blockchain,” *European Property Law Journal*, vol. 9, no. 1, pp. 21–66, May 2020, doi: 10.1515/eplj-2020-0003.

[37] “Digital Assets: Cryptocurrencies vs. Crypto Tokens,” *Gemini*. https://www.gemini.com/cryptopedia/cryptocurrencies-vs-tokens-difference#section-what-is-a-digital-asset (accessed Mar. 22, 2025).

[38] A. B. Posavec, K. Aleksic-Maslac, and M. Tominac, “Non-Fungible tokens: Might learning about them be necessary?,” *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 700–705, May 2022, doi: 10.23919/mipro55190.2022.9803425.

[39] “Blockchain: Exploring the role of crypto tokens in blockchain technology - FasterCapital,” *FasterCapital*. https://fastercapital.com/content/Blockchain--Exploring-the-Role-of-Crypto-Tokens-in-Blockchain-Technology.html (accessed Mar. 26, 2025).

[40] “Tokens | CodeFI Assets,” Feb. 14, 2023. https://docs.assets.consensys.net/concepts/tokens (accessed Mar. 28, 2025).

[41] L. Team, “Brief introduction to token standards,” *LCX*, Aug. 13, 2024. https://www.lcx.com/brief-introduction-to-token-standards/ (accessed Mar. 28, 2025).

[42] “What are token standards? An overview,” *What Are Token Standards? An Overview*. https://crypto.com/en/university/what-are-token-standards (accessed Mar. 29, 2025).

[43] R. Chamria, “Introduction to token standards: ERC-20, ERC-721, ERC-777, and ERC-1155,” *Blockchain Deployment and Management Platform | Zeeve*, Jun. 26, 2024. https://www.zeeve.io/blog/introduction-to-token-standards-erc-20-erc-721-erc-777-and-erc-1155/ (accessed Mar. 29, 2025).

[44] “What is ERC-20?,” *Coinbase*. https://www.coinbase.com/learn/crypto-glossary/what-is-erc-20 (accessed Mar. 29, 2025).

[45] “ERC-721 Non-Fungible Token Standard,” *ethereum.org*. https://ethereum.org/en/developers/docs/standards/tokens/erc-721/ (accessed Mar. 29, 2025).

[46] “Qu’est-ce que l’ERC-721 ?,” *Coinbase*. https://www.coinbase.com/fr/learn/crypto-glossary/what-is-erc-721 (accessed Mar. 29, 2025).

[47] “What is ERC-1155?,” *Coinbase*. https://www.coinbase.com/learn/crypto-glossary/what-is-erc-1155 (accessed Mar. 29, 2025).

[48] “Asset Tokenization: Revolutionizing Ownership through Book Entry Systems - FasterCapital,” *FasterCapital*. https://fastercapital.com/content/Asset-Tokenization--Revolutionizing-Ownership-through-Book-Entry-Systems.html (accessed Mar. 29, 2025).

[49] S. Joshi and A. Choudhury, “Tokenization of real estate assets using blockchain,” *International Journal of Intelligent Information Technologies*, vol. 18, no. 3, pp. 1–12, Sep. 2022, doi: 10.4018/ijiit.309588.

[50] A. Saari, S. Junnila, and J. Vimpari, "Blockchain's grand promise for the real estate sector: A systematic review," *Applied Sciences*, vol. 12, no. 23, p. 11940, Nov. 2022, doi: 10.3390/app122311940.

[51] A. Prakash, "Real Estate Tokenization: How Blockchain is Transforming Property Investment," *Appventurez*, Apr. 30, 2025. https://www.appventurez.com/blog/real-estate-tokenization (accessed Jun. 05, 2025).

[52] O. Konashevych, "General concept of real estate tokenization on Blockchain," *European Property Law Journal*, vol. 9, no. 1, pp. 21–66, May 2020, doi: 10.1515/eplj-2020-0003.

[53] *The tokenisation of assets and potential implications for financial markets*. 2020. doi: 10.1787/83493d34-en.

[54] L. Swinkels, "Empirical evidence on the ownership and liquidity of real estate tokens," *Financial Innovation*, vol. 9, no. 1, Jan. 2023, doi: 10.1186/s40854-022-00427-5.

[55] A. Saari, J. Vimpari, and S. Junnila, "Blockchain in real estate: Recent developments and empirical applications," *Land Use Policy*, vol. 121, p. 106334, Aug. 2022, doi: 10.1016/j.landusepol.2022.106334.

[56] A. Baum, "Tokenisation – the future of real estate investment?," *Oxford FoRE*.

[57] A. A. I. Izadin and R. Yusof, "Democratizing Real estate Investment: A Systematic Review of tokenization in Real estate," Jan. 2024. doi: 10.2139/ssrn.5015941.

[58] Md. S. Shahariar, P. Banik, and Md. A. Habib, *A Secure Land Record Management System using Blockchain Technology*. 2022, pp. 557–562. doi: 10.1109/iccit57492.2022.10054925.

[59] K. Vayadande, R. Shaikh, S. Rothe, S. Patil, T. Baware, and S. Naik, "Blockchain-Based land Record System," *ITM Web of Conferences*, vol. 50, p. 01006, Jan. 2022, doi: 10.1051/itmconf/20225001006.

[60] K. S and G. Sarath, "Securing Land Registration using Blockchain," *Procedia Computer Science*, vol. 171, pp. 1708–1715, Jan. 2020, doi: 10.1016/j.procs.2020.04.183.

[61] D. D. Gaikwad, A. N. Hambir, H. S. Chavan, G. K. Khedkar, and S. V. Athawale, "Real estate Land Transaction System using Blockchain," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 3, pp. 307–311, Mar. 2022, doi: 10.22214/ijraset.2022.40633.

[62] I. Ahmad, M. A. Alqarni, A. A. Almazroi, and L. Alam, "Real estate management via a decentralized blockchain platform," *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 66, no. 2, pp. 1813–1822, Dec. 2020, doi: 10.32604/cmc.2020.013048.

[63] R. Akila, J. J. B. Merin, S. Subhashini, N. B. Behera, F. Varghese, and V. Jeswanth, "A trustable real estate transaction based on public blockchain: a smart contract-driven framework," *Salud Ciencia Y Tecnología - Serie De Conferencias*, vol. 3, p. 763, Jan. 2024, doi: 10.56294/sctconf2024763.

[64] N. V. Wanve, N. C. Sawant, N. R. Sonawane, and N. H. Pathak, "Decentralized Web Application for Real Estate Property Transaction using Blockchain and Smart Contract Technology," *International Journal of Advanced Research in Science Communication and Technology*, pp. 76–81, May 2022, doi: 10.48175/ijarsct-3710.

[65] I. Karamitsos, M. Papadaki, and N. B. A. Barghuthi, "Design of the Blockchain Smart Contract: a use case for real estate," *Journal of Information Security*, vol. 09, no. 03, pp. 177–190, Jan. 2018, doi: 10.4236/jis.2018.93013.

[66] H. Jamshed, U. Waheed, S. Iqbal, M. Faheem, M. W. Ashraf, and Y. Mansoor, "Dynamic smart contracts framework on Ethereum private blockchain for real estate management," *The Journal of Engineering*, vol. 2025, no. 1, Jan. 2025, doi: 10.1049/tje2.70063.

[67] N. R. Fernandes, N. A. Fernandes, and N. H. Jawale, "Blockchain based Real Estate using Smart Contracts," *International Journal of Engineering and Management Research*, vol. 10, no. 6, pp. 48–51, Dec. 2020, doi: 10.31033/ijemr.10.6.6.

[68] S. Chaudhari, P. Babhulgaonkar, R. Baheti, Y. Shah, R. Kulkarni, and A. Deshpande, "Blockchain Real-Estate Property using Smart Contracts," vol. 6, no. 11, 2023.

[69] C. Qi-Long, Y. Rong-Hua, and L. Fei-Long, “A Blockchain-based Housing Rental System:,” in Proceedings of the International Conference on Advances in Computer Technology, Information Science and Communications, Xiamen, China: SCITEPRESS - Science and Technology Publications, 2019, pp. 184–190. doi: 10.5220/0008097201840190.

[70] A. Rath, R. K. Pateriya, D. Tomar, and S. Singh, “Tokenization of rental real estate assets using blockchain technology,” *Research Square (Research Square)*, Jun. 2023, doi: 10.21203/rs.3.rs-3004275/v1.

[71] A. Gupta, J. Rathod, D. Patel, J. Bothra, S. Shanbhag, and T. Bhalerao, “Tokenization of real estate using blockchain technology,” in *Lecture notes in computer science*, 2020, pp. 77–90. doi: 10.1007/978-3-030-61638-0_5.

[72] “Solidity — Solidity 0.8.30 documentation.” https://docs.soliditylang.org/en/v0.8.30/

[73] GeeksforGeeks, “What is Hardhat?,” *GeeksforGeeks*, Jul. 04, 2024. https://www.geeksforgeeks.org/what-is-hardhat/ (accessed Jun. 05, 2025).