Dissertation/Report submitted to the

UNIVERSITY OF MOHAMED BOUDIAF - M'SILA, ALGERIA

1985

جامعة محمد بوضياف - المسيلة

University of Mohamed Boudiaf-Msila

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

in partial fulfillment of the requirements for the degree of

Master's/License in Computer Science

Specialization: Computer Science

Option: your option

Presented by

Tabi Achwaq

Entitled

Blockchain for the Secure Management of Digital Identities in Real Estate Platforms

---

Under the supervision of

Professor Dr. Dabba Ali

Jury Members

Prof. first Name Last Name	University of M'sila	President
Prof. first Name Last Name	University of M'sila	Reporter
Dr. first Name Last Name	University of M'sila	Examiner

09/June, 2026

الملخص

تتناول هذه المذكرة دراسة وبناء نظام رقمي يعتمد على تقنية البلوكشين والعقود الذكية من أجل اقتراح حل حديث يهدف إلى تحسين موثوقية المعاملات الرقمية ورفع مستوى الشفافية والأمان. وقد انطلق هذا العمل من الإشكالات التي تعاني منها الأنظمة التقليدية، مثل المركزية، وصعوبة التتبع، واحتمال التلاعب بالبيانات، مما يستدعي اعتماد مقاربات أكثر تطورًا وفعالية.

تم خلال هذه الدراسة عرض الأسس النظرية للتقنيات المعتمدة، ثم تصميم وتنفيذ نموذج تطبيقي يوضح كيفية استغلال البلوكشين في تأمين العمليات وتسجيلها بصورة لا مركزية. كما تم التركيز على دور العقود الذكية في أتمتة بعض الإجراءات وتقليل التدخل اليدوي، بما يساهم في تحسين الأداء العام للنظام.

وأظهرت النتائج أن الحل المقترح يوفر بنية أكثر أمانًا وشفافية وقابلية للتوسع مقارنة بالحلول التقليدية، مع إمكانية تطويره مستقبلًا ليدعم وظائف إضافية أكثر تقدمًا.

الكلمات المفتاحية: البلوكشين، العقود الذكية، اللامركزية، الشفافية، الأمان

Abstract

This dissertation studies and develops a digital system based on blockchain technology and smart contracts in order to propose a modern solution that improves the reliability of digital transactions and enhances transparency and security. The work is motivated by the limitations of traditional systems, such as centralization, difficulty in tracing operations, and the risk of data manipulation, which require more advanced and effective approaches.

The study first presents the theoretical foundations of the adopted technologies, then designs and implements a practical model showing how blockchain can be used to secure and record operations in a decentralized manner. Special attention is given to the role of smart contracts in automating procedures and reducing manual intervention, thereby improving the overall system performance.

The results show that the proposed solution offers a more secure, transparent, and scalable architecture than conventional approaches, with potential for future extensions and additional functionalities.

Keywords: Blockchain, Smart Contracts, Decentralization, Transparency, Security

Dedication

بسم الله الرحمن الرحيم

Every achievement in life is a moment to pause and reflect on those who stood by us through both happiness and hardship. I dedicate this humble work with sincere gratitude and deep respect to my beloved mother and father, to all my family, to everyone who walked with me on this journey, believed in me, encouraged me, and never ceased to support me. I also dedicate it to all my friends, whose presence, kindness, and encouragement gave me strength throughout this path.

Tabi Achwaq

Acknowledgements

First and foremost, I thank Allah, the Most Gracious and the Most Merciful, for granting me the strength, patience, and guidance to complete this work.

I would like to express my deepest gratitude to my supervisor, Professor Dr. Dabba Ali, for his guidance, patience, and valuable suggestions throughout the duration of this project.

I also extend my sincere thanks to all the instructors and colleagues who supported me with constructive feedback and encouragement.

Finally, I thank my family and friends for their unwavering support and motivation, which helped me remain focused and committed to completing this work.

Contents

List of Figures

List of Tables

Introduction

The real estate sector is one of the most significant and sensitive economic sectors due to its direct connection with property ownership, investment activities, and the legal transfer of assets. Despite its importance, traditional real estate management systems continue to face several challenges, including limited transparency, lengthy administrative procedures, bureaucratic complexity, and the risk of document forgery and manipulation. These issues often reduce efficiency, increase operational costs, and weaken trust among stakeholders.

With the rapid advancement of digital technologies, blockchain has emerged as a promising solution for addressing many of these challenges. Its decentralized architecture, immutability, and transparency provide new opportunities for securing records, automating transactions, and enhancing trust between participants without relying entirely on centralized authorities.

This dissertation proposes RESYS, a hybrid decentralized platform for real estate management and transaction processing. The proposed solution integrates blockchain technology, smart contracts, cryptographic authentication mechanisms, and Non-Fungible Tokens (NFTs) to improve transparency, traceability, security, and efficiency throughout the property lifecycle.

- **Key Research Aspects:**
  - Fundamental concepts of the real estate sector and its digital transformation
  - The role of blockchain technology in improving transparency, security, and trust
  - The use of smart contracts and NFTs for property registration and ownership transfer
- **Core Problems Identified:**
  - Lack of transparency in traditional property management systems
  - Administrative delays and bureaucratic complexity
  - Risks of document forgery, tampering, and centralized single points of failure
- **Study Objectives:**
  1. Design a secure hybrid architecture for real estate management
  2. Implement blockchain-based property registration and verification mechanisms
  3. Evaluate the feasibility of decentralized property listing and ownership transfer

- **Methodological Framework:**
  - Theoretical study of real estate systems and blockchain fundamentals
  - Design and implementation of the RESYS platform
  - Validation through user interactions, administrative review, and on-chain execution
- **Document Organization:**
  - Chapter 1 presents the real estate sector, its characteristics, and the main challenges affecting current systems.
  - Chapter 2 introduces blockchain technology and the concepts used in the proposed solution.
  - Chapter 3 presents the design, implementation, and practical demonstration of the RESYS platform.
  - The conclusion summarizes the main findings and discusses future work.

This work is motivated by the growing need for secure, transparent, and efficient real estate management systems. The proposed platform seeks to provide a reliable framework for property registration and transfer while establishing a foundation for future enhancements such as decentralized storage solutions, advanced interoperability mechanisms, and improved scalability.

Chapter 1: General Concepts

1.1 Introduction

The real estate sector is one of the most important economic sectors, representing a fundamental component of economic activity due to its role in regulating property ownership, supporting investment and meeting individuals' housing needs. In recent years, this sector has witnessed numerous changes and challenges as a result of economic and political factors, as well as rapid technological development.

In this context, it has become necessary to examine the various aspects associated with the real estate sector, particularly regarding its management methods and the systems employed within it. Accordingly, in this chapter we will address the key concepts relating to the real estate sector and its role in the economy, as well as the types of property and an overview of the property market and its significance.

We will also review the various traditional systems used in property management, highlighting the main problems they face, before moving on to present some modern technological solutions that can help improve this sector and address its shortcomings.

1.2 Basic Concepts of the Real Estate Sector

The real estate sector is considered one of the fundamental sectors underpinning various economic activities, encompassing a range of processes related to the ownership, use and trading of property. To understand this sector more accurately, it is necessary to address some of the basic concepts associated with it.

1.2.1 Definition of Real Estate

Real estate is defined as any immovable property that cannot be moved from one place to another, such as land, buildings and structures, and also includes the rights associated with it, such as the right of ownership and the right of usufruct. Real estate is considered an asset of high economic value, making it a focus of interest for both individuals and institutions.

From a legal perspective, the Algerian legislature defines real estate in Article 683 of the Civil Code as:

> "Anything that is fixed in a specific location and cannot be moved without being damaged; anything other than this is considered movable" [2].

This definition highlights the immovable nature of real estate and distinguishes it from movable property, which forms an important basis for regulating legal transactions relating to real estate ownership.

1.2.2 Types of Real Estate

Real estate is divided into several types according to its nature of use and purpose, the most important of which include:

- **Residential property:** This refers to property intended for habitation, such as houses, flats, apartment blocks and mansions.
- **Industrial property:** This refers to property used for industrial activities, such as factories and warehouses.
- **Agricultural property:** This refers to property used for agricultural activities, such as farms and orchards.
- **Commercial property:** This refers to property used for commercial activities, such as shops and shopping centres.
- **Publicly owned property:** This refers to public property owned by the state and used for specific purposes, such as government land, hospitals, schools and mosques.
- **Waqf property:** This refers to property designated for charitable or religious purposes, where the asset is set aside and its benefits allocated to charitable causes, and may not be disposed of by sale or transfer, as defined by Algerian legislation as follows:

  "Waqf properties are real estate properties which the owner freezes of his own volition, so that their benefits are permanently for the benefit of a charitable association or one serving the public interest, whether the benefit is immediate or deferred" [3].

1.2.3 The Property Market

The property market refers to all transactions relating to the sale, purchase and letting of property, and is considered one of the most important economic sectors due to its direct link to various economic activities.

This market is influenced by several factors, the most important of which are supply and demand, the geographical location of the property, as well as general economic conditions and the laws governing it. The property market is also characterised by its dynamic nature, as it is constantly changing in line with economic and technological developments, as well as price trends that directly affect property transactions.

The property market encompasses the dynamics of supply and demand, with investors and prospective investors constantly monitoring this market in order to make informed decisions regarding the purchase, sale or investment in property [4].

1.2.4 The Importance of the Real Estate Sector

The real estate sector is one of the vital sectors that plays a significant role in supporting the national economy, as it contributes directly to stimulating investment and creating employment opportunities [5].

The importance of this sector lies in its provision of individuals' basic needs, foremost among which is housing; it is also considered a key area for attracting capital, given its relative stability compared to some other sectors [6].

The real estate sector also contributes to the development of infrastructure and urban planning through the completion of residential and commercial projects, thereby promoting urban growth and local development [7].

In light of current digital transformations, the importance of this sector is increasing due to the need to adopt modern systems that help improve management efficiency and enhance transparency in transactions, particularly with the rising demand for property [5].

1.3 Users of the Real Estate Sector

The real estate sector comprises a range of users and stakeholders who contribute to the vitality of this market, with their roles varying according to the nature of their activities and interests.

Among the most important users of the real estate sector are:

- **Individuals:** These are buyers, sellers, or tenants; they represent the core stakeholders in this sector.
- **Property investors:** These are stakeholders who seek to make a profit by buying, selling or letting property.
- **Estate agents:** These act as intermediaries between sellers and buyers, facilitating transactions.
- **Financial institutions:** These include banks, which provide mortgage loans and support investment activities.

- **Government bodies:** These oversee the regulation of the sector through laws and legislation.

These parties interact with one another within a complex environment, necessitating the existence of effective systems to ensure the regulation of their relationships and transparency in transactions [7].

1.4 Digitalisation in the Real Estate Sector

With technological advancements, digitisation has become one of the most important means of improving performance in the real estate sector, as it involves converting traditional processes into digital ones using information technology.

Digitisation is defined as follows:

> "Digitisation refers to creating a digital representation of physical objects or attributes. For instance, we scan a paper document and save it as a digital document (e.g., PDF). In other words, digitisation is about converting something non-digital into a digital representation or artefact. Computerised systems can then use it for various use cases. An example from manufacturing would be when a measurement is converted from a manual or mechanical reading to an electronic one." [8]

Digitisation therefore helps facilitate transactions, speed up procedures and reduce errors, as well as improving transparency by providing accurate and easily accessible information at the touch of a button.

1.5 Digital Platforms in the Real Estate Sector

Digital platforms are among the most significant manifestations of digital transformation across various sectors, representing an electronic environment that allows users to interact and conduct transactions online in an easy and fast manner.

Digital platforms are defined as follows:

> "An online infrastructure based on software to facilitate interactions and transactions between users" [9].

Among the real estate platforms available in Algeria are Ouedkniss Immobilier, Mubawab Algerie and Lkeria, which enable the listing of properties and facilitate property searches.

1.6 Problems of Traditional Systems

Despite the importance of this sector to the national economy, the traditional systems used to manage it continue to suffer from several problems that limit their efficiency and affect their credibility.

Among the most prominent of these problems are:

- **Lack of transparency:** It is difficult for users to verify the accuracy of information relating to property ownership, which opens the door to ambiguity and mistrust.
- **Administrative complexity:** Traditional systems rely on lengthy and complex procedures requiring numerous documents, which slows down transactions.
- **Administrative bureaucracy:** Property transactions are characterised by numerous routine procedures and multiple intervening parties, which complicates the administrative process, delays the completion of operations, and increases the burden on users.
- **Slow transaction processing:** Due to reliance on manual processes and paper documents, sales or property transfers take a long time.
- **Risks of forgery and manipulation:** Documents can be tampered with or data altered, particularly in the absence of a unified and secure digital system.
- **Centralised systems:** These systems rely on centralised databases, making them vulnerable to hacking or data loss.
- **Difficulty accessing information:** In some cases, access to property data is limited or requires complex procedures.

1.7 Conclusion

In conclusion, this chapter has addressed the key concepts associated with the property sector, by defining property, outlining its types, analysing the property market and its significance, and identifying the various stakeholders in the sector.

The role of digitalisation and digital platforms in the development of the real estate sector has also been highlighted, whilst addressing the key problems faced by traditional systems, particularly regarding bureaucracy, lack of transparency, and slow transaction times.

Consequently, there is a clear need to adopt modern technological solutions capable of improving the management of this sector, a topic which will be addressed in the following chapter through an examination of blockchain technology and its role in addressing these challenges.

Chapter 2: Blockchain Technology

2.1 Introduction

The world has recently witnessed rapid transformations in the field of digitalisation, with traditional centralised systems proving unable to keep pace with growing demands for security, transparency and trust. With every transaction passing through a centralised intermediary, concerns and questions arise regarding the potential for data manipulation and the lack of independent oversight. Consequently, blockchain technology has emerged to represent a radical shift in the way information is stored, verified and transmitted.

The significance of blockchain extends from being the technical foundation for digital currencies such as Bitcoin, to improving vital sectors such as healthcare, supply chains, digital identity management, and even the property sector. The registration of property ownership, with its associated documents, contracts and legal powers, represents an ideal environment for applying blockchain principles.

This chapter addresses the theoretical and technical foundations of blockchain technology, starting with the concept and historical development, through to the technical architecture, smart contracts and digital assets, and finally the development environment used to build decentralised applications. This chapter forms the reference framework upon which the real estate transaction management system proposed in this research is based.

2.2 Blockchain Fundamentals

2.2.1 Definition of Blockchain: The Distributed Ledger

Technically speaking, blockchain is defined as a decentralised, distributed database consisting of a chain of blocks linked together using cryptographic techniques, where each block contains a set of pre-verified transactions, along with a unique digital fingerprint linking it to the previous block, ensuring data integrity and making any attempt to alter it detectable and unacceptable to the rest of the network participants.

It is defined as:

> "a distributed digital ledger consisting of cryptographically signed transactions grouped into blocks, operating without the need for a central authority or intermediary" [10].

This system relies on a network of interconnected nodes that verify and record transactions, ensuring a high level of security and transparency. Thanks to these characteristics, blockchain has become a reliable infrastructure for multiple applications, extending beyond digital currencies to other sectors such as real estate and healthcare.

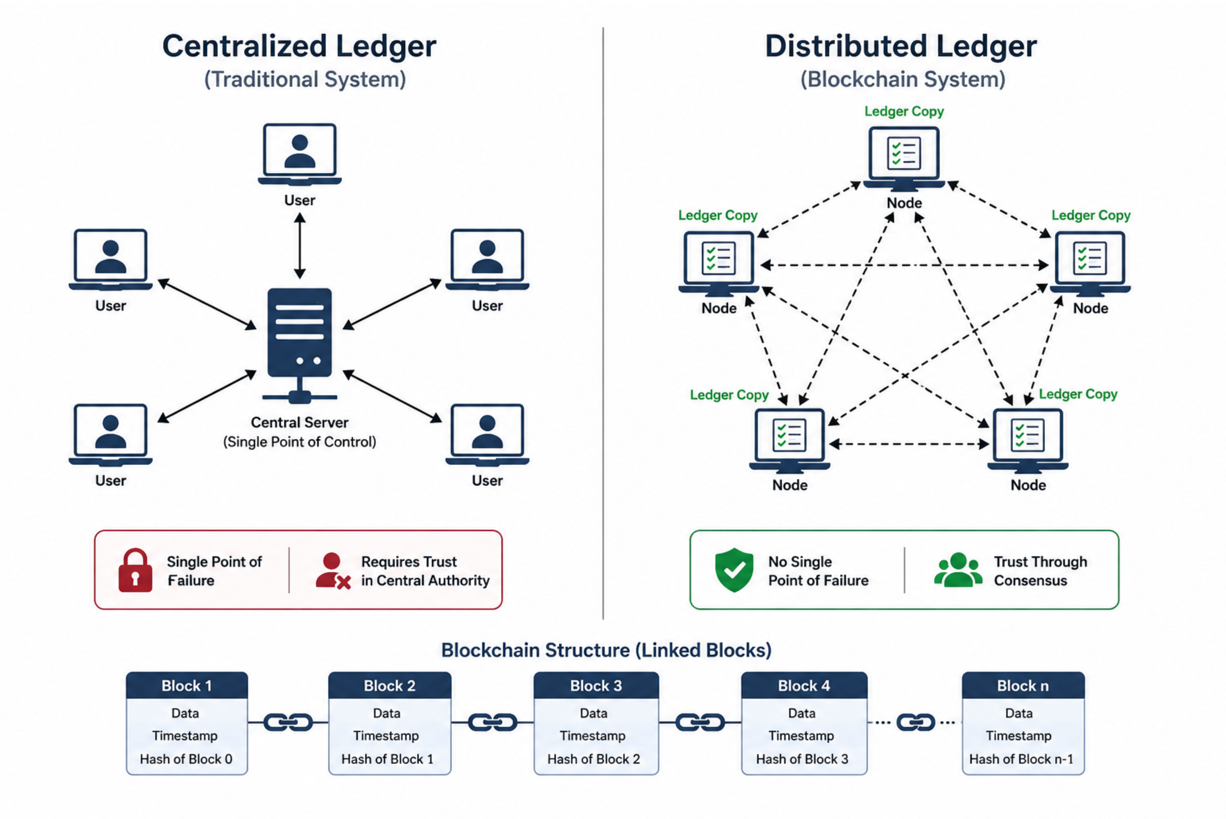

Figure 2.1: Comparison between a Centralised Ledger and a Distributed Ledger (Blockchain System)

2.2.2 History and Development of Blockchain Technology

A Brief History of Blockchain

The first prototype of blockchain was created in the early 1990s when computer scientist Stuart Haber and physicist Scott Stornetta used cryptographic techniques in a chain of blocks to secure digital documents against data tampering.

Haber and Stornetta inspired the work of many computer scientists and cryptography enthusiasts, which ultimately led to the creation of Bitcoin as the first digital currency powered by blockchain technology. Since then, the use of blockchain has grown significantly, and digital currencies are now a global phenomenon.

2.2.3 Fundamental Properties of Blockchain

1. **Immutability:** This characteristic ensures that once data is recorded within the blockchain, it becomes permanently unalterable. The absolute inability to modify or delete exe-

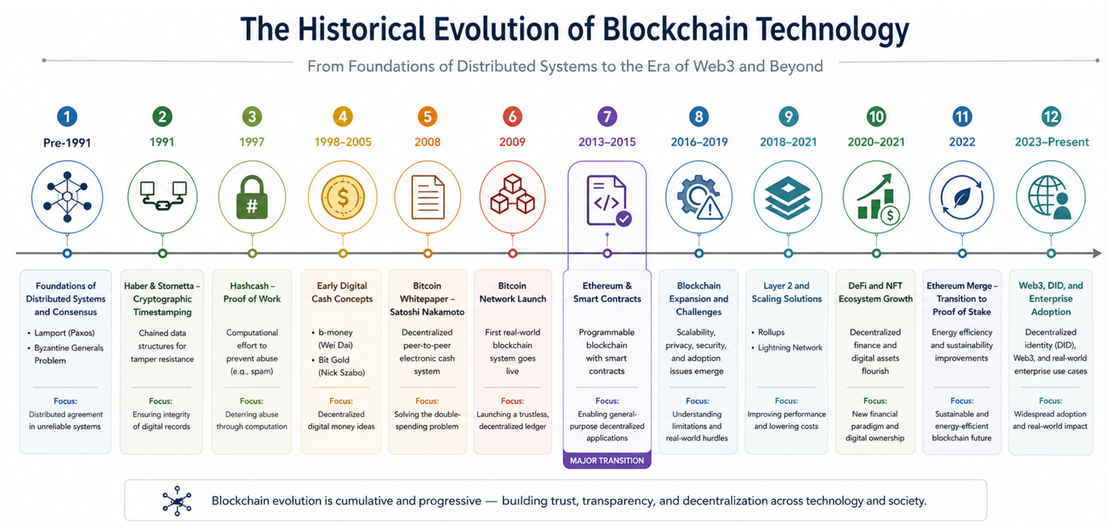

Figure 2.2: The Historical Evolution of Blockchain Technology

cuted transactions establishes a foundational layer of trust within the system.

2. **Distribution:** The architecture of blockchain relies on a distributed ledger replicated across all participating nodes in the network. Every participant maintains a synchronized copy of the entire ledger.

3. **Decentralisation:** Unlike traditional databases, blockchain operates entirely without a central governing authority. The responsibility for network control is distributed among a consensus of nodes.

4. **Security:** The system achieves an exceptionally high level of security through advanced cryptographic techniques and the sequential chaining of blocks.

5. **Consensus Mechanism:** To validate transactions prior to their permanent recording, the network utilizes robust consensus algorithms that allow all distributed nodes to reach a unified agreement.

6. **Transparency:** Blockchain inherently fosters transparency by permitting network participants to independently view and audit transactions.

7. **Traceability:** The chronological, chain-like structure of the blocks enables comprehensive tracking of all transactions back to their origin.

2.3 Technical Architecture and Network Components

The blockchain network consists of three interlocking structural layers: the network layer (P2P Network), the data layer (blocks and chain), and the consensus layer.

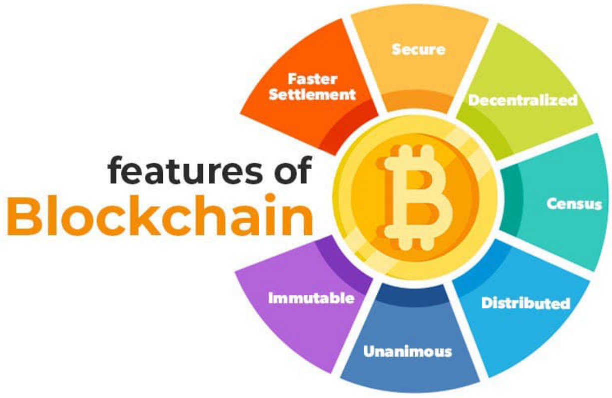

Figure 2.3: The Fundamental Features of Blockchain Technology

2.3.1 Nodes

Nodes are the fundamental building blocks of the blockchain network, representing computers that participate in the network by storing a copy of the transaction ledger and verifying its validity within a decentralised environment based on a peer-to-peer model [11].

> "It should be noted that interacting with blockchain applications does not necessarily require the user to become a full node; rather, their role is often limited to client interfaces..." [12]

2.3.2 Transactions

A transaction represents the transfer of value or data between parties participating in the network. Technically, a transaction is defined as an encrypted digital record documenting a specific event. A transaction consists of:

- **Addresses:** Unique digital identifiers for the sender and recipient.
- **Payload or data:** The part containing the substance of the transaction or contract data.
- **Digital signature:** Verifies identity and approval using the private key, preventing forgery.
- **Timestamp:** Precisely determines the time of the transaction's creation for sequential ordering.

Transaction lifecycle

A transaction begins with its creation, followed by cryptographic signing, and transmission over the network. Nodes verify the validity; if validated, it enters the "transaction pool" pending inclusion in the next block.

Node Type	Main Function	Data Stored	Resources Needed	Example Use
Full Node	Verifies and stores the entire blockchain	Entire ledger	High	Bitcoin, Ethereum
Light Node	Relies on full nodes for data verification	Block headers only	Low	Mobile wallets
Master Node	Provides extra services like governance or instant transactions	Full ledger	Medium-High	Dash, PIVX
Archival Node	Stores all historical blockchain states	Complete history	Very high	Ethereum research, auditing
Mining / Validator Node	Adds new blocks and secures consensus	Partial or full	High	Bitcoin miners, ETH validators

Figure 2.4: Types of Blockchain Nodes and their Main Functions [1]

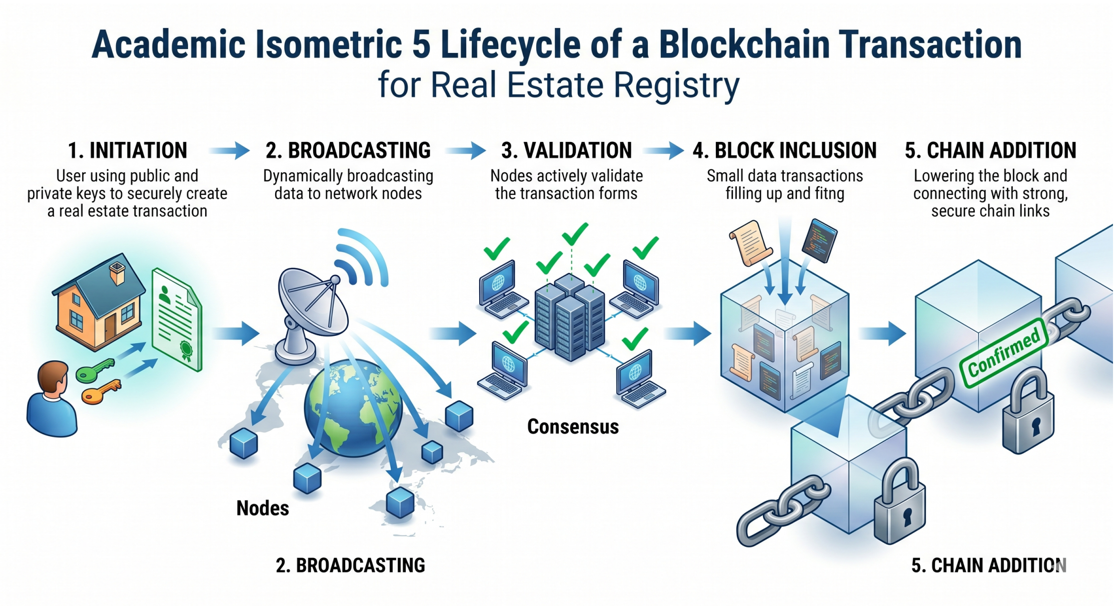

Figure 2.5: Lifecycle of a Blockchain Transaction for Real Estate Registry

2.3.3 The Block

The block is the basic unit of storage. As stated in :

> "Every block consists of two parts, block data which is a transaction record stored in each block and block header"[13]

The block header consists of:

- Block version (4 bytes): Validation rules.
- Merkle root hash (32 bytes): A hashing method to represent the hash of the block data.
- Previous block hash (32 bytes): The hash value of the preceding block.
- Timestamp (4 bytes): Creation time.
- nBits (4 bytes): Threshold for the hashing target value.
- Nonce (4 bytes): A unique number used to generate a hash for each block.

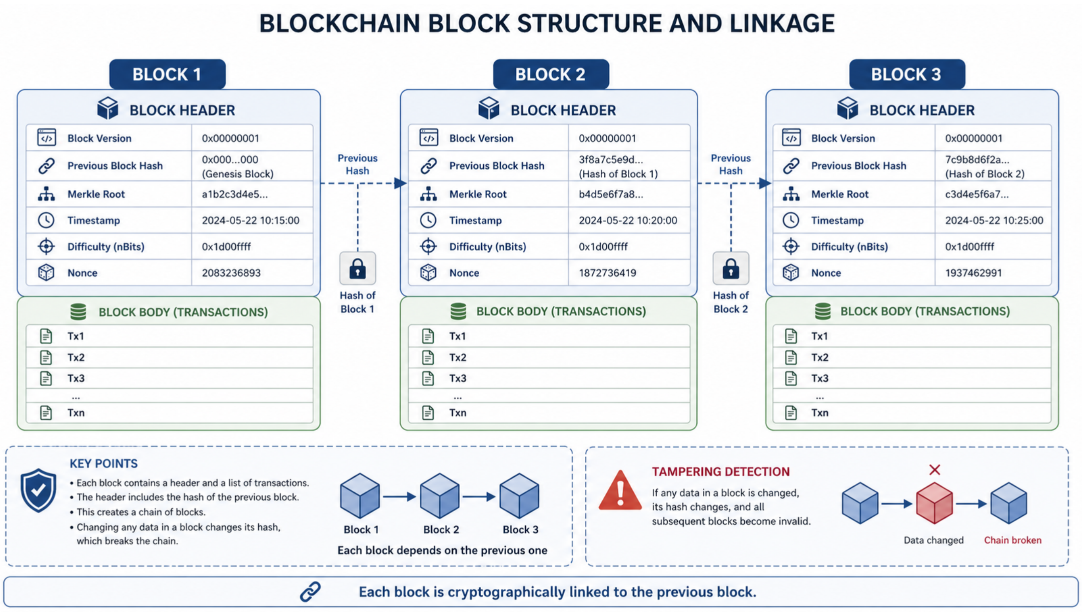

Figure 2.6: Blockchain Blocks Structure

2.3.4 Cryptography

The blockchain system relies primarily on two main cryptographic techniques:

Digital Signature

Used to verify the validity and source of a transaction. This ensures:

- **Authentication:** Issued by the actual address holder.
- **Non-repudiation:** The sender cannot deny the transaction.
- **Integrity:** Details have not been altered after signing [6].

Asymmetric Cryptography

Also known as "public-key" cryptography, this relies on a pair of mathematically linked keys [14]:

- **Public Key:** Acts as the digital address to receive transactions.
- **Private Key:** Acts as a signature known only to the user, essential for authorizing transfers.

Cryptographic Hash Functions

Functions such as the $SHA-256$ algorithm convert input into a fixed-length output known as a "digital fingerprint". Notable characteristics include [14]:

- **Pre-image Resistance:** Impossibility of deducing original data from the hash.
- **Avalanche Effect:** Slightest modification results in a drastic change in the hash value [1].
- **Uniqueness:** Each data set has a unique digital fingerprint.

2.3.5 Merkle Tree and Block Root

To organise multiple transactions efficiently, blockchain relies on a "Merkle tree" [10].

- **Leaves:** Individual transaction hashes.
- **Intermediary nodes:** Hierarchical combination of adjacent hashes [14].
- **Root:** The final "Merkle root" stored in the block header.

This allows nodes to verify any transaction quickly through the simplified verification protocol [10].

2.3.6 Consensus Mechanisms

A consensus mechanism is a decision-making algorithm designed to reach a unified agreement amongst all nodes [10]. Its core function is to prevent "double-spending" and maintain integrity [15].

Main types of consensus mechanisms

1. **Proof of Work (PoW):** Relies on solving complex mathematical problems (mining). High security but high energy consumption [16].

2. **Proof of Stake (PoS):** Validators are selected based on assets or "stakes" held. More energy-efficient [15].

3. **Proof of Authority (PoA):** Suitable for private networks, relies on pre-known trusted nodes [16].

Table 2.1: Comparison of the main consensus mechanisms

Standard	**PoW**	**PoS**	**PoA**
Energy consumption	Very high	Low	Very low
Security Level	High	High	Medium
Decentralisation	High	High	Low
Transaction Speed	Slow	Medium-Fast	Very Fast
Use case	Bitcoin	Ethereum 2022+	Private networks

2.4 Types of Blockchain Networks

- **Public Blockchain:** Open to anyone (e.g., Bitcoin, Ethereum). Highest decentralisation.

- **Private Blockchain:** Managed by a single organisation; restricted participation.

- **Consortium / Hybrid Blockchain:** Managed by a group of allied entities (e.g., Quorum).

Table 2.2: Comparison of blockchain network types

Type	**Public**	**Private**	**Hybrid**
Control	None / Everyone	Single entity	Group of entities
Transparency	Full	Limited	Partial
Privacy	Low	High	Medium
Scalability	Limited	High	High
Trust required	None	Full	Partial
Example	Ethereum	Hyperledger	Quorum

2.5 Smart Contracts

2.5.1 Concept and Mechanism

Nick Szabo coined the term in 1994, but practical application began with Ethereum in 2015 [13]. Smart contracts are defined as programs embedded within the blockchain to execute transactions automatically [6].

The mechanism is based on "If-This-Then-That" logic. Once pre-defined conditions are met, the contract fulfils obligations immediately [6].

2.5.2 Key features

- **Autonomy:** Reduces risks of human intervention.
- **Efficiency:** Maximises speed by eliminating intermediaries.
- **Flexible programming:** Uses specialized languages like Solidity [6].

2.6 Digital Assets and Tokens

2.6.1 Virtual Assets

Digital representations of value that can be traded or used for payment [17].

2.6.2 Tokens

Programmable digital data units managed via smart contracts.

1. **Fungible Tokens (FTs):** Identical in value (e.g., $ERC - 20$). Used for investment units [14].
2. **Non-Fungible Tokens (NFTs):** Unique and non-divisible (e.g., $ERC - 721$). Used for title deeds.

Table 2.3: Comparison between fungible and non-fungible tokens

Standard	**FT (ERC-20)**	**NFT (ERC-721)**
Distinctiveness	Each unit is identical	Each token is unique
Divisibility	Yes (0.5 units)	No (whole units)
Fungibility	Yes	No
Use case	Currencies, points	Art, real estate
Practical example	ETH, USDC	Apartment ownership

2.7 Real Estate Tokenisation

The process involves five main stages:

1. **Legal verification:** Off-chain Validation of documents.
2. **Minting:** Issuing a digital token (ERC-721) linking identity to metadata.
3. **Immutable Ledger Entry:** Permanent historical record on the blockchain [10].
4. **Atomic Swap:** Automation of trading where ownership and payment occur simultaneously.

2.8 Leading International Applications (Case Studies)

- **Republic of Georgia:** First to use blockchain to secure property deeds, reducing costs by 90% [14].
- **Sweden (Lantmäteriet):** Automated property buying/selling using smart contracts [18].
- **UAE (Dubai):** Launched a blockchain-based system to document all property transactions [13].

2.9 Conclusion

This chapter has addressed the conceptual and technical framework of blockchain, highlighting smart contracts,and NFTs. Chapter 3 will move to the practical framework, developing the architectural design of the proposed decentralised platform.

Note: *Figures 2.1, 2.2, 2.3, 2.5, and 2.6 were generated using artificial intelligence tools for illustrative and explanatory purposes only.*

Chapter 3: System Design and Implementation

3.1 Introduction

This chapter serves as the structural bridge from the theoretical and conceptual understanding of blockchain technology, as presented in the previous chapter, to the practical realisation of the proposed Real Estate System (RESYS). The primary objective of this chapter is to provide a detailed outline of the system architecture and to demonstrate how this architecture is successfully translated into a working decentralised application (DApp).

Structurally, this chapter is divided into two main phases:

- **The First Phase:** Dedicated to software engineering and system design using the Unified Modelling Language (UML) to formalise the architecture's core logic. We define the specific privileges of the system's main actors—the Citizen (Property Owner/Buyer) and the Authority Admin (Governor)—through detailed Use Case and Class Diagrams. These models capture the dynamic flow of data within our hybrid architecture, mapping interactions between the Web3 frontend, the off-chain Flask server (automating PDF deed generation and cryptographic hashing), and the on-chain Solidity smart contracts enforcing strict access control and absolute data immutability.

- **The Second Phase:** Dedicated to the practical implementation of these theoretical models, showcasing the engineered Graphical User Interfaces (GUIs) of the platform. This section demonstrates how the hybrid architecture operates in practice by providing real-world screenshots of the wallet connection via MetaMask, the property registration pipeline, administrative vetting, and automated digital deed retrieval. Ultimately, this chapter highlights the seamless integration of traditional web technologies with decentralised blockchain verification to deliver a secure, transparent, and trustless environment for real estate management.

3.2 UML-Based System Modelling

The Unified Modelling Language (UML) was adopted to build a solid, secure, and scalable architecture for the proposed real estate documentation and sales system (RESYS). This modelling approach provides precise structural and behavioural blueprints, serving as a vital link between abstract system requirements and production-ready software implementation. In this section, we dissect the functional behaviour and the static data structures of our platform using Use Case and Class diagrams.

3.2.1 Use Case Diagram

The Use Case diagram delineates the behavioural boundaries of the platform, formalising how different actors trigger operations across the hybrid layout. The system governs two primary human actors and an external decentralised protocol:

- **Citizen (Property Owner/Buyer):** Initiates interactions with the ecosystem. The citizen is granted privileges to submit new property registration applications, manage personal assets, list verified properties on the marketplace, cancel active listings, and execute immediate peer-to-peer purchases.

- **Authority Admin (Governor):** Represents the administrative vetting node responsible for validating legal documents before an asset is cleared for public trading. The admin's privileges are strictly bounded to inspecting the verification queue, reviewing application details, and cross-checking attached deeds (View Details & PDF). Following this review, the admin executes a decisive state change: either rejecting the request or approving and cryptographically minting it onto the blockchain network (Approve & Mint NFT).

- **Blockchain (≪External System≫):** Operates as a decentralised state machine. It acts as an immutable ledger that processes transaction requests, writes state variables (minting tokens, altering ownership via transfers), and fires events that ensure transparency, transaction security, and permanent storage of digital fingerprints.

- **Server Automation Logic:** One of the most critical architectural decisions in the RESYS ecosystem is the automated security processing of files to prevent manipulation during sales operations. As illustrated in the diagram, when the admin triggers "Approve & Mint NFT", the architecture automatically invokes the cryptographic calculation function (Calculate Hash) through an ≪include≫ relationship. Similarly, when a citizen triggers a "Purchase Property" action, the backend server is automatically stimulated to compile a fresh title deed (Auto-Generate PDF) reflecting the new owner, which immediately invokes a fresh hash calculation to keep the permanent registry perfectly synchronised.

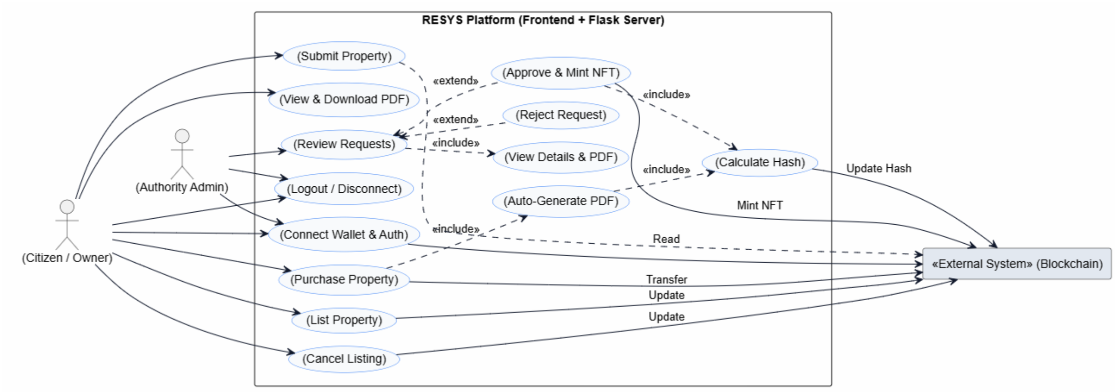

Figure 3.1: Use Case Diagram of the real estate documentation and sales platform (RESYS) illustrating user roles and automations

3.2.2 Class Diagram

The Class Diagram represents the static structural model of the system. It defines the fundamental data entities, their attributes, methods, and the logical relationships bridging the off-chain server environment with the on-chain smart contract storage for sales management.

Given the hybrid nature of the platform, relational database concepts have been adapted to fit the Web3 architecture paradigm:

- **User & Admin Entities:** In this decentralised environment, traditional login credentials (passwords) are entirely replaced by public-key cryptography. Consequently, walletAddress and adminWallet are designated as Primary Keys (≪PK≫) identifying sellers, buyers, and administrators across the distributed network.
- **Web3Frontend:** Represents the client-side presentation layer containing interactive methods such as connectWallet() and submitPropertyForm() to broadcast payloads.
- **PropertyRequest (≪JSON/Buffer≫):** Acts as a temporary off-chain data staging buffer managed by the Flask server. This entity captures the owner's personal info and property dimensions prior to validation, using the submitter's address (ownerAddress) as a Foreign Key (≪FK≫) to link the request to its legitimate author.
- **FlaskServer (≪Off-chain Backend≫):** Implements the backend automation logic. It controls request staging states and handles computationally heavy actions such as generatePDFDeed() and executing the SHA-256 hashing algorithms.
- **RealEstateRegistry (≪Smart Contract≫):** The immutable ledger containing the core commercial execution logic. It exposes state-changing execution pathways like registerProperty() and purchaseProperty(), heavily guarded by custom Modifiers to ensure that operations are restricted exclusively to authorised wallets.

- **PropertyStruct (≪struct≫):** The fundamental on-chain composite data type permanently written to Ethereum storage. It maps the asset metadata—including listing status (isForSale), sale price, and the document fingerprint (docHash). It incorporates adminAuthorizer as a Foreign Key (≪FK≫) to log the exact admin node that verified and authorised the asset.

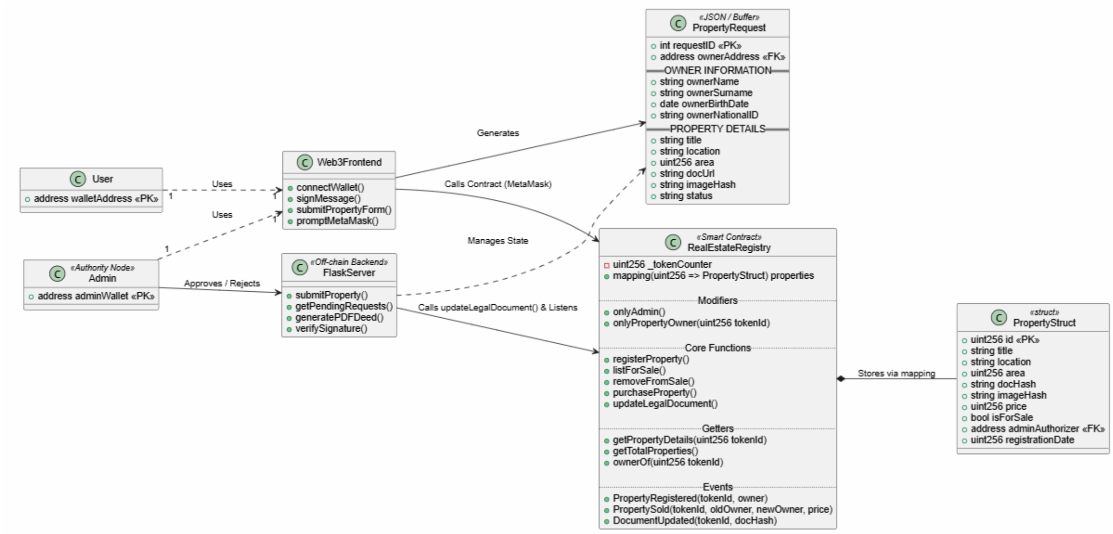

Figure 3.2: Class Diagram illustrating data entities and the structure of the hybrid sales platform

3.3 Tools and Technologies Used

To construct the hybrid architecture of the RESYS platform, a specific stack of development tools and frameworks was selected. Each technology was chosen based on its capacity to fulfil strict architectural requirements, balancing off-chain backend automation with on-chain Web3 capabilities.

3.3.1 Development Tools

1. **Visual Studio Code (VS Code):**
   The primary Integrated Development Environment (IDE) utilised for writing, managing, and debugging the multi-language codebase. According to its official documentation, VS Code is a lightweight yet powerful source code editor that supports multiple programming languages and extensions, making it ideal for full-stack Web3 development [19]. It provided a unified workspace for managing Python, Solidity, JavaScript, and HTML/CSS scripts simultaneously.

Figure 3.3: Visual Studio Code Logo [19]

2. **Python (Flask Framework & Web3.py):**
   Python serves as the core programming language for the off-chain backend infrastructure. The Flask framework, described in its official documentation as "a lightweight WSGI web application framework designed to make getting started quick and easy" [20], was employed to manage temporary data states (JSON buffers) and automate the generation of PDF deeds. Furthermore, the Web3.py library was integrated to allow the server to communicate cryptographically with the blockchain, listen to events, and execute the SHA-256 hashing algorithms. Web3.py is defined as "a Python library for interacting with Ethereum" [21], providing the essential bridge between the off-chain backend and the on-chain smart contracts.

Figure 3.4: Python, Flask, and Web3.py Logos [20], [21]

3. **Solidity:**
   The object-oriented, high-level programming language used to write the `RealEstateRegistry` smart contract. As defined in its official documentation, Solidity is "a statically-typed curly-braces programming language designed for developing smart contracts that run on the Ethereum Virtual Machine (EVM)" [22]. It defines the rigid logical constraints for property tokenisation, market listing, and atomic purchasing directly on the EVM. The smart contract also inherits from OpenZeppelin's battle-tested ERC-721 implementation to ensure security and standard compliance [23].

Figure 3.5: Solidity Logo [22]

4. **Ganache:**

   Utilised as a personal, local blockchain environment for rapid Ethereum distributed ledger testing. Ganache is officially described as "a personal blockchain for rapid Ethereum and Corda distributed application development" [24]. It provided a secure, isolated sandbox to deploy smart contracts and simulate transactional edge cases (such as ownership transfers and gas fee estimations) without incurring real financial costs.

Figure 3.6: Ganache Logo [24]

5. **MetaMask:**

   The critical Web3 gateway extension. MetaMask is defined as "a crypto wallet and gateway to blockchain applications" [25]. It functions as both the primary decentralised authentication mechanism (identifying users via cryptographic public keys) and the transaction signing provider for citizens and the Authority Admin. By leveraging MetaMask's cryptographic signature capabilities, RESYS eliminates the need for traditional username/password authentication entirely.

Figure 3.7: MetaMask Logo [25]

6. **Ethers.js:**

   The standard JavaScript library employed in the client-side presentation layer (Frontend). Ethers.js is described in its official documentation as "a complete and compact library for interacting with the Ethereum Blockchain and its ecosystem" [26]. It seamlessly connects the graphical user interface with the MetaMask extension and the local blockchain node, facilitating the reading of state variables and the broadcasting of new state-changing transactions.

Figure 3.8: Ethers.js Logo [26]

3.3.2 Technology Stack Summary

Table 3.3.2 summarises the complete technology stack adopted for the RESYS platform, organised by architectural layer.

Layer	**Technology**	**Role**
Frontend	HTML / CSS / JavaScript	User Interface
Frontend	Ethers.js	Blockchain Interaction
Frontend	MetaMask	Wallet & Auth
Backend	Python / Flask	Off-chain Processing
Backend	Web3.py	Blockchain Bridge
Smart Contract	Solidity	Contract Language
Smart Contract	OpenZeppelin ERC-721	NFT Standard
Dev / Testing	Ganache	Local Blockchain
Dev / Testing	VS Code	IDE

Table 3.1: RESYS Platform Technology Stack

3.4 Smart Contract Implementation

This section illustrates the complete implementation of the `RealEstateRegistry` smart contract using Solidity. This contract serves as the core of the decentralized system, leveraging the ERC-721 standard to tokenize real estate and convert it into unique digital assets (NFTs). It ensures the secure management of the initial registration processes by the administration, listing properties for sale, executing purchases, and transferring ownership automatically (Atomic Transactions). Furthermore, it implements advanced protection against Reentrancy Attacks to guarantee the security of all financial transactions.

3.4.1 Smart Contract Implementation

This section illustrates the complete implementation of the `RealEstateRegistry` smart contract using Solidity. The contract serves as the core of the decentralised registry, leveraging the ERC-721 standard to tokenise real estate. It ensures the secure execution of transactions and implements advanced protection against Reentrancy Attacks.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

/**
 * @title RealEstateRegistry
 * @dev A decentralised registry for securing real estate documentation and ownership transfers.
 * It includes reentrancy protection and strict administrative access control for initial registration.
 */

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";

contract RealEstateRegistry is ERC721, Ownable, ReentrancyGuard {

    // Token counter initialized to 1 for logical frontend indexing
    uint256 private _tokenCounter;

    struct Property {
        uint256 id;                  // Unique digital identifier
        string title;                // Property title
        string location;             // Geospatial location
        uint256 area;                // Area in square metres
        string docHash;              // SHA-256 fingerprint of the legal document
        string imageHash;            // Property image reference
        uint256 price;               // Listing price in Wei
        bool isForSale;              // Market listing status
        address adminAuthorizer;     // The administrative node that validated the deed
        uint256 registrationDate;    // On-chain registration timestamp
    }

    // Mapping token IDs to their respective Property structs
    mapping(uint256 => Property) public properties;

    // Off-chain telemetry events
    event PropertyRegistered(uint256 indexed tokenId, address indexed owner, string docHash, address indexed admin);
    event PropertyListed(uint256 indexed tokenId, uint256 price, string newImageHash);
```

```
    event PropertySold(uint256 indexed tokenId, address from, address to, uint256 price);
    event PropertyUnlisted(uint256 indexed tokenId);

    constructor() ERC721("Algerian Real Estate Registry", "ARER") {
        _tokenCounter = 1;
    }
```

Listing 3.1: Smart contract definition, library imports, and the core Property data structure

The initial state mutation is strictly controlled. The `registerProperty` function ensures that only the authorised administrative node can mint new digital deeds onto the blockchain after off-chain legal verification is completed.

```
    /**
     * @dev Registers a new property and mints a digital deed (NFT).
     * Restricted to the contract owner (Admin) after physical document verification.
     */
    function registerProperty(
        address _initialOwner,
        string memory _title,
        string memory _location,
        uint256 _area,
        string memory _docHash,
        string memory _imageHash
    ) public onlyOwner returns (uint256) {
        uint256 newTokenId = _tokenCounter;

        properties[newTokenId] = Property({
            id: newTokenId,
            title: _title,
            location: _location,
            area: _area,
            docHash: _docHash,
            imageHash: _imageHash,
            price: 0,
            isForSale: false,
            adminAuthorizer: msg.sender,
            registrationDate: block.timestamp
        });

        _safeMint(_initialOwner, newTokenId);
        _tokenCounter++;

        emit PropertyRegistered(newTokenId, _initialOwner, _docHash, msg.sender);
```

```
        return newTokenId;
    }
```

Listing 3.2: The `registerProperty` function executing the minting of the digital deed by the Authority Admin

Once registered, asset owners are granted autonomous control over their property's market status. The contract provides execution pathways to list or withdraw properties from the public marketplace, strictly enforcing ownership modifiers.

```
    /**
     * @dev Lists the property for sale in the marketplace. Restricted to current owner.
     */
    function listForSale(uint256 _tokenId, uint256 _price, string memory _newImageHash) public {
        require(ownerOf(_tokenId) == msg.sender, "Not the owner");
        require(_price > 0, "Price must be greater than zero");

        properties[_tokenId].price = _price;
        properties[_tokenId].imageHash = _newImageHash;
        properties[_tokenId].isForSale = true;

        emit PropertyListed(_tokenId, _price, _newImageHash);
    }

    /**
     * @dev Removes the property from the marketplace.
     */
    function removeFromSale(uint256 _tokenId) public {
        require(ownerOf(_tokenId) == msg.sender, "Not the owner");
        require(properties[_tokenId].isForSale == true, "Property is not listed for sale");

        properties[_tokenId].isForSale = false;
        properties[_tokenId].price = 0; // Price zeroed as a security measure

        emit PropertyUnlisted(_tokenId);
    }
```

Listing 3.3: Market control functions `listForSale` and `removeFromSale` governing market visibility

The core commercial logic of the platform is embedded within the `purchaseProperty` function. This function executes an atomic swap, ensuring that the transfer of property ownership and the transfer of financial value occur simultaneously and inextricably, guarded against reentrancy vulnerabilities.

```
/**
 * @dev Executes the purchase of a listed property.
 * Transfers ownership and routes funds to the seller in a single atomic transaction.
 */
function purchaseProperty(uint256 _tokenId) public payable nonReentrant {
    Property storage prop = properties[_tokenId];
    address seller = ownerOf(_tokenId);

    require(prop.isForSale, "Property is not listed for sale");
    require(msg.value >= prop.price, "Sent value is less than the asking price");
    require(msg.sender != seller, "Seller cannot buy their own property");

    // State mutation prior to external calls (Security Best Practice)
    prop.isForSale = false;
    uint256 salePrice = prop.price;

    // Execute ERC-721 ownership transfer
    _transfer(seller, msg.sender, _tokenId);

    // Route financial value to the seller
    (bool success, ) = payable(seller).call{value: msg.value}("");
    require(success, "Blockchain transfer to seller failed");

    emit PropertySold(_tokenId, seller, msg.sender, salePrice);
}
```

Listing 3.4: The secure `purchaseProperty` function demonstrating atomic transfer of funds and ownership

To maintain perfect synchronisation between the off-chain generated documents and the on-chain registry, the system architecture includes an automated update function. This ensures the permanent ledger reflects the cryptographic fingerprint of the newly generated deed immediately following a successful sale.

```
/**
 * @dev Automated synchronisation for updated legal document hashes.
 * Invoked automatically by the backend server after ownership transfer and new PDF generation.
 */
function updateLegalDocument(uint256 _tokenId, string memory _newDocHash) public onlyOwner {
```

```
        require(_exists(_tokenId), "Token ID does not exist in the registry");

        properties[_tokenId].docHash = _newDocHash;
        properties[_tokenId].registrationDate = block.timestamp;
    }

    /**
     * @dev Retrieves comprehensive property state data.
     */
    function getPropertyDetails(uint256 _tokenId) public view returns (Property memory) {
        require(_exists(_tokenId), "Token ID does not exist in the registry");
        return properties[_tokenId];
    }

    /**
     * @dev Helper function to query the total volume of documented properties.
     */
    function getTotalProperties() public view returns (uint256) {
        return _tokenCounter - 1;
    }
}
```

Listing 3.5: The `updateLegalDocument` synchronisation function and property state getters

3.5 User Interface (UI) Design and Implementation

The user interface (UI) represents the dynamic interactive layer connecting actors to the off-chain backend systems and the decentralised blockchain network. The platform's interfaces are engineered in compliance with modern UI/UX design standards to deliver a secure, frictionless, and intuitive user experience, effectively translating complex cryptographic workflows into clear, transparent, and sequential steps.

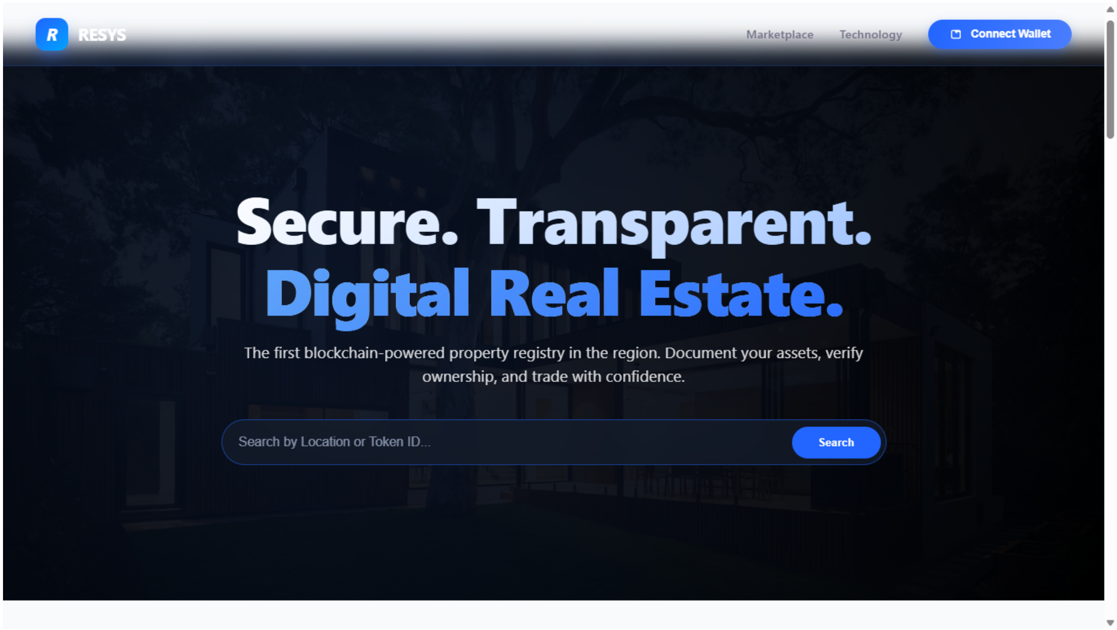

Figure 3.9: Platform Main Landing Page

3.5.1 Landing Page and Decentralised Authentication

The landing page serves as the primary interactive gateway to the platform, featuring a responsive design that enables users to explore verified listings. To gain access to the respective dashboards, the system completely bypasses traditional credentials (usernames and passwords) in favour of decentralised, cryptographic public-key authentication.

Upon clicking the "Connect Wallet" button, the frontend programmatically invokes the MetaMask extension. The platform then dispatches a secure cryptographic Signature Request to be signed locally by the user. This process unequivocally verifies wallet ownership and cryptographic identity without exposing any sensitive or private data outside the user's local environment.

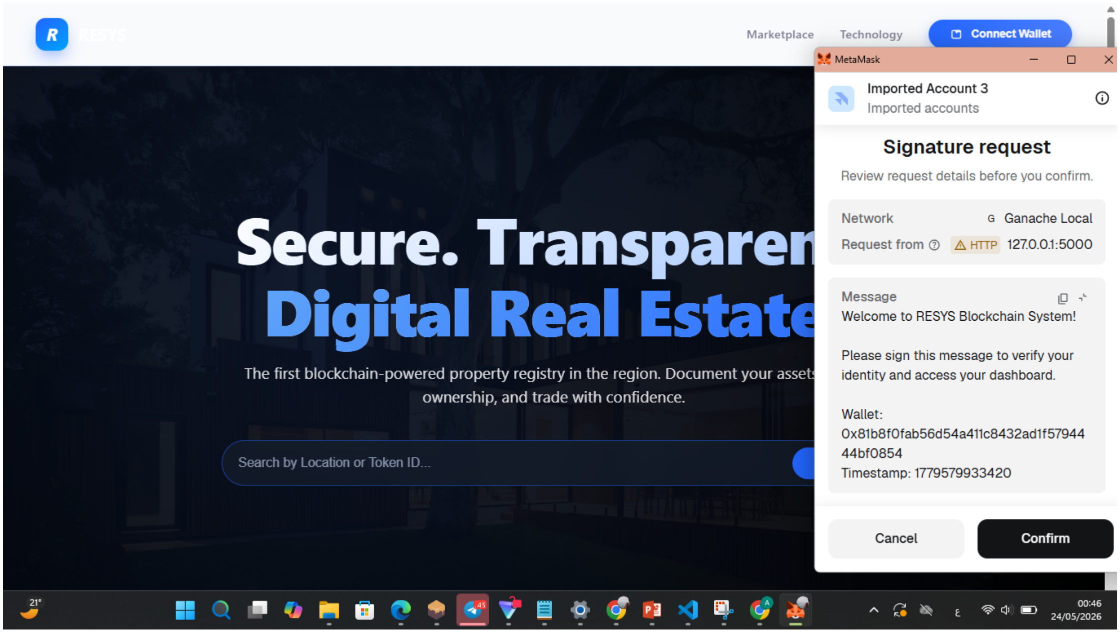

Figure 3.10: Cryptographic Signature Request via MetaMask Extension

3.5.2 User Dashboard and Asset Management

Following successful cryptographic authentication, users are automatically routed to their personalised asset management dashboard (My Assets). This interface provides a centralised, telemetry-driven environment to monitor real estate holdings, structurally split into two primary segments reflecting the hybrid nature of the architecture:

- **Pending Requests:** Showcases property applications currently residing within the off-chain temporary JSON storage, awaiting administrative evaluation.
- **Verified Blockchain Deeds:** Displays tokenised real estate assets permanently recorded on-chain as NFTs, augmented with real-time state indicators (e.g., "FOR SALE" badges).

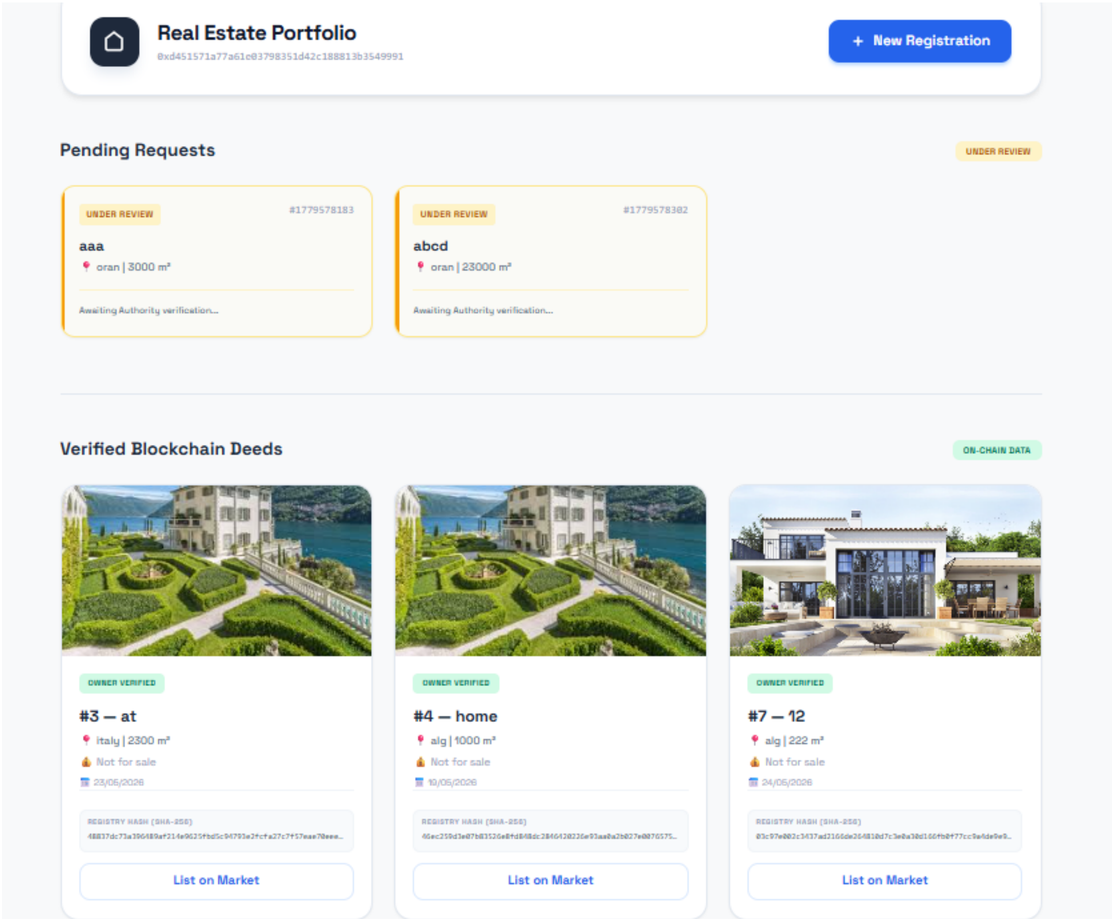

Figure 3.11: User Dashboard and Centralised Asset Management Interface

3.5.3 Property Registration and Submission

To register a new property, users are required to submit comprehensive metadata through the platform. The requested information includes:

- **Legal Identity Details:** Full name, surname, date of birth, and national identification number.
- **Geospatial and Engineering Attributes:** Title, exact geographical location, and total area dimensioned in square metres.
- **Physical Verification Artefacts:** Upload fields for a legal proof document in PDF format and a descriptive property image.

Once the user executes the "Send to Authority Node" command, the payload is transmitted to the Flask server for ingestion and hash pre-processing. The application instantly materialises within the citizen's dashboard under the pending queue, flagged with an amber "Under Review" state, awaiting the Authority Admin's decision.

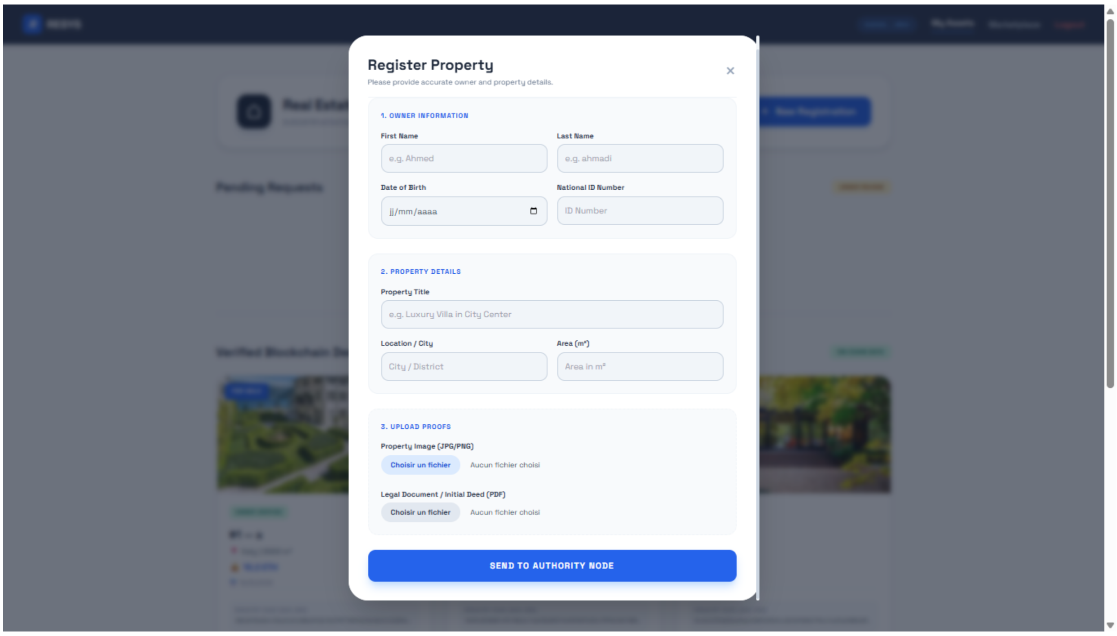

Figure 3.12: Property Registration Modal and Data Input Form

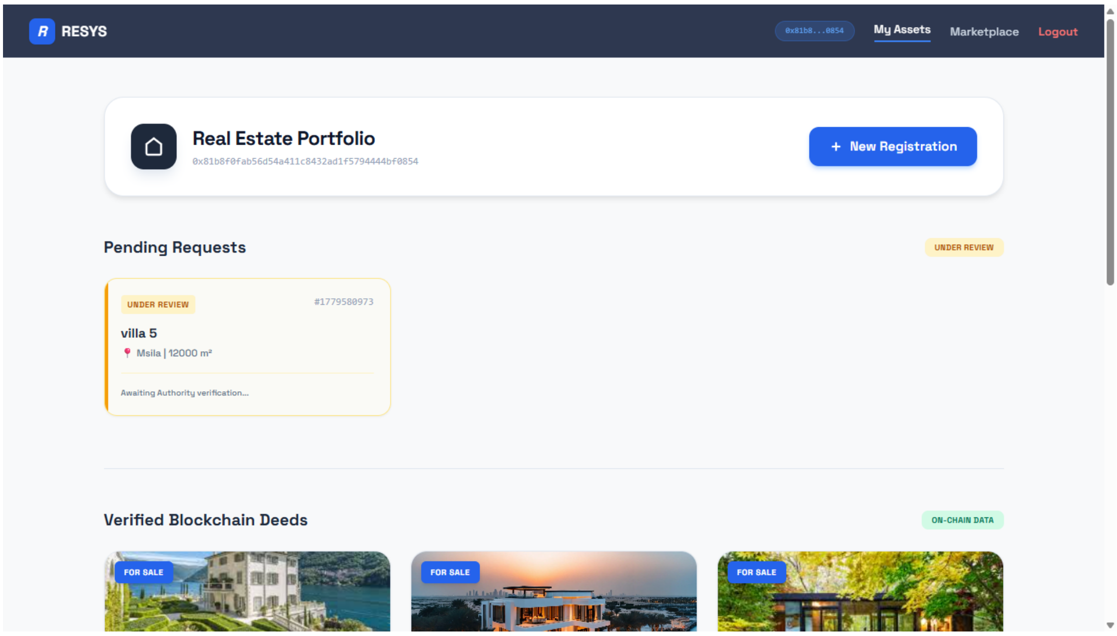

Figure 3.13: Newly Submitted Property Flagged as Under Review

3.5.4 Official Blockchain Registry

Upon receiving administrative approval and subsequent on-chain tokenisation, the owner can interact with their verified asset card to trigger the "Official Blockchain Registry" modal. This component pulls state parameters directly from the smart contract in real time (on-chain), presenting immutable metadata to the user: the distinct token ID, precise registration timestamp, the cryptographic address of the authorising administrative node (Authorizing Node), and the unique SHA-256 digital fingerprint of the legal document (DocHash). Additionally, an actionable "View Digital Property Deed" button is provided to fetch the physical deed.

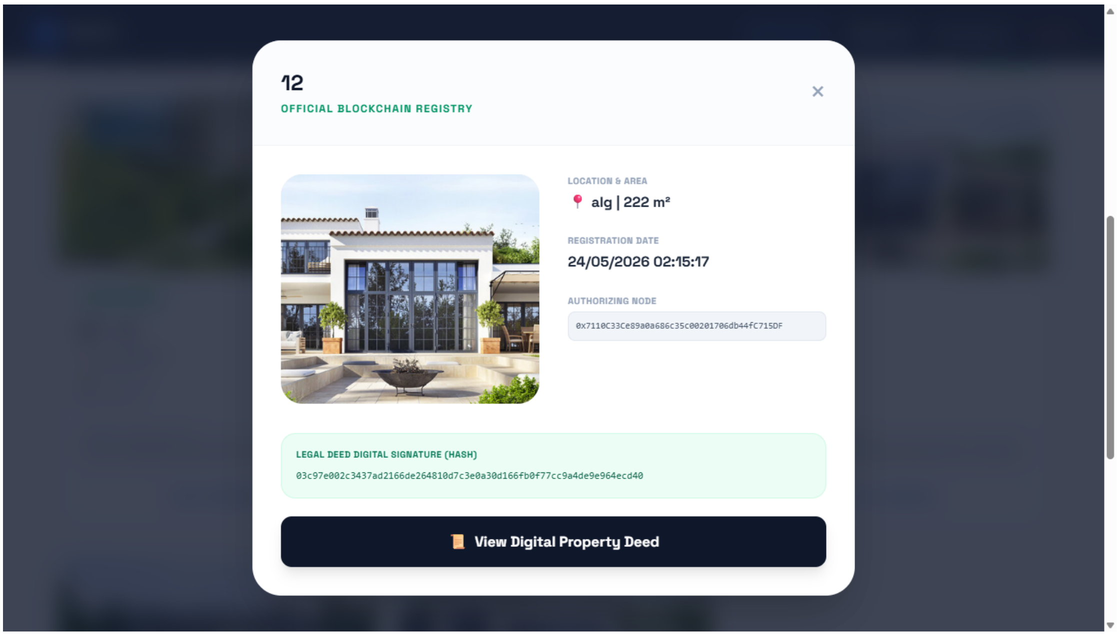

Figure 3.14: Official On-chain Blockchain Registry Card for a Verified Asset

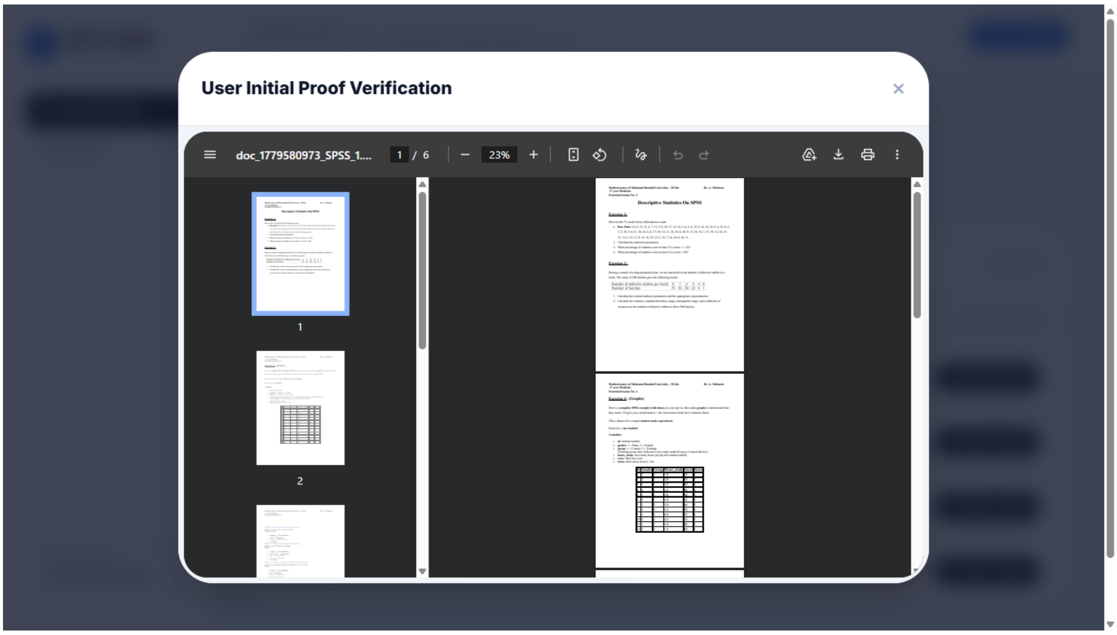

Figure 3.15: Integrated PDF Document Viewer

3.5.5 Market Listing and Transaction Confirmation

The ecosystem empowers validated owners to seamlessly liquidate their tokenised holdings by listing them on the decentralised marketplace. When electing to sell an asset, a "List on Market" modal prompts the owner to define their target liquidation price denominated in Ether (ETH).

Confirming the listing price initialises an interaction with the smart contract's state-changing functions, causing MetaMask to seamlessly intercept the workflow. The wallet presents a formal Transaction Request, itemising the estimated network gas fees required to update the asset's listing state on the blockchain network.

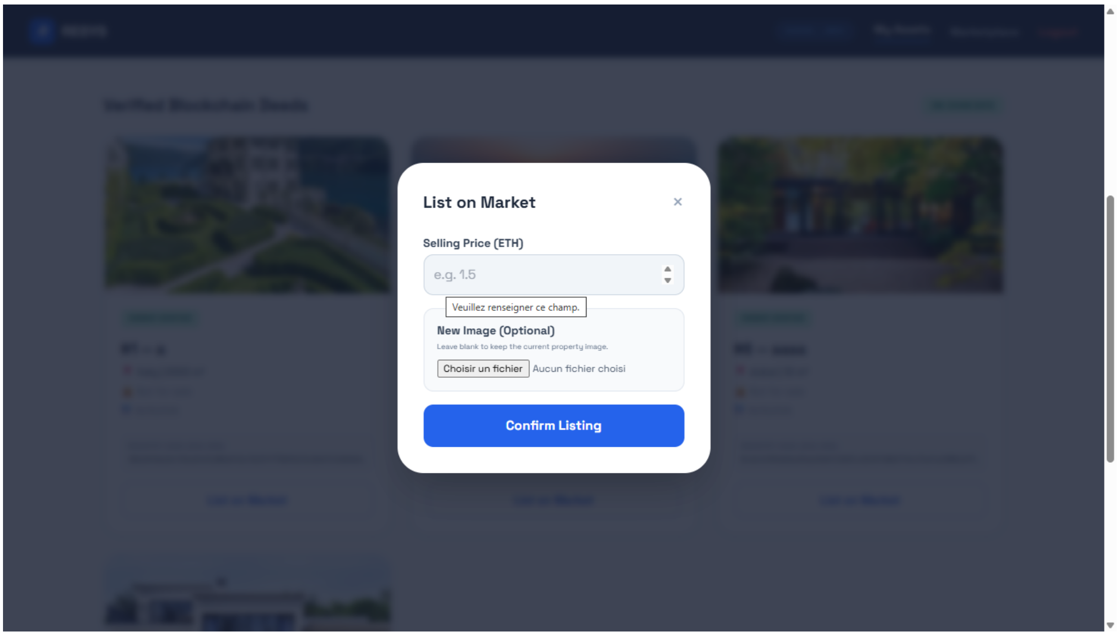

Figure 3.16: List the Property for Sale Option

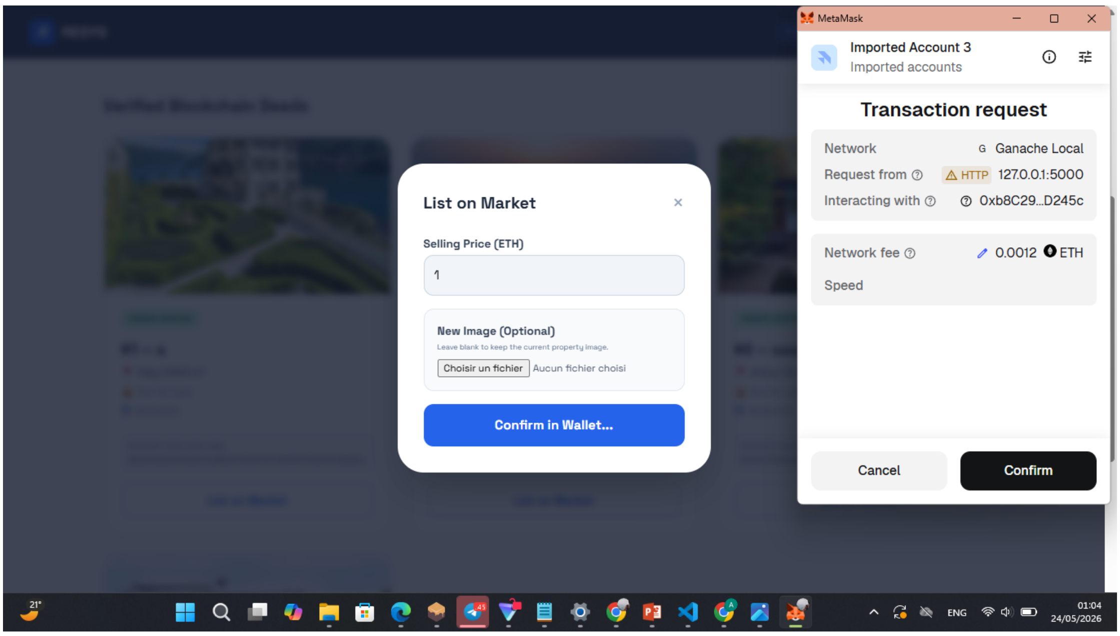

Figure 3.17: Marketplace Listing Window and Price Configuration Modal

3.5.6 Verified Property Marketplace

The platform features a public Marketplace interface where real estate assets are displayed for immediate trading (Live Listings). Properties are showcased with their specified ETH prices and core details, allowing users to browse and execute secure Peer-to-Peer (P2P) "Buy Now" transactions based on decentralised data that is completely immune to tampering.

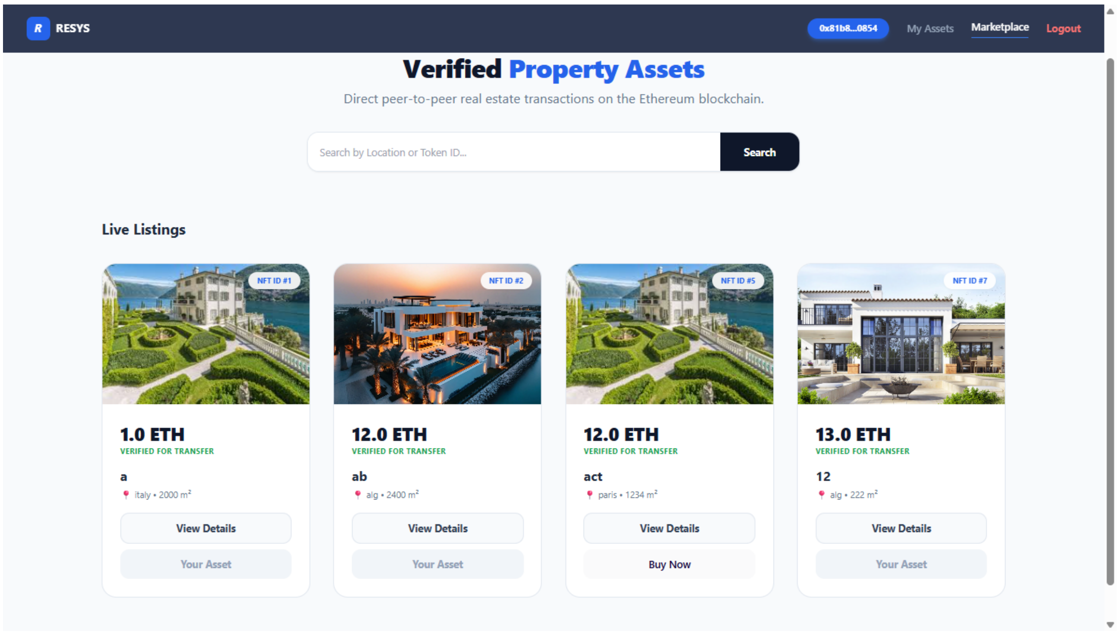

Figure 3.18: Property Marketplace Interface

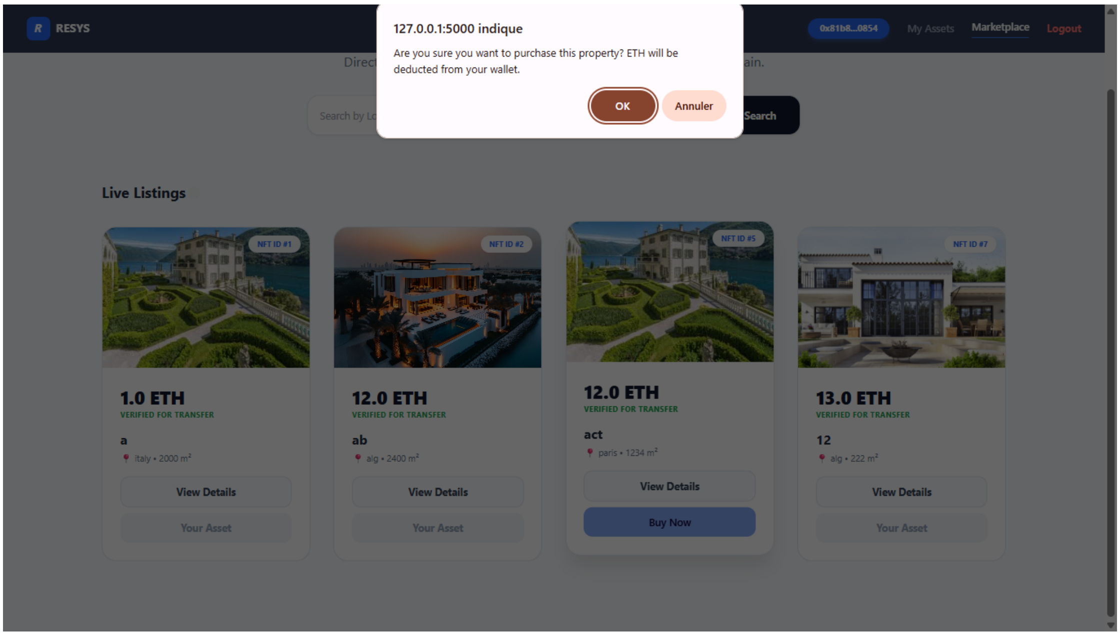

Figure 3.19: Buying a Property from the Marketplace

3.5.7 Property Purchase Process

Upon initiating the property purchase process, the system generates a cryptographic verification request displayed via the MetaMask extension. Once the buyer authenticates and digitally signs the transaction, the smart contract automatically and atomically executes the operations, whereby the ownership of the digital real estate asset is transferred from the seller to the buyer, while simultaneously, the corresponding financial value is securely and decentralizedly transferred from the buyer's wallet to the seller's wallet.

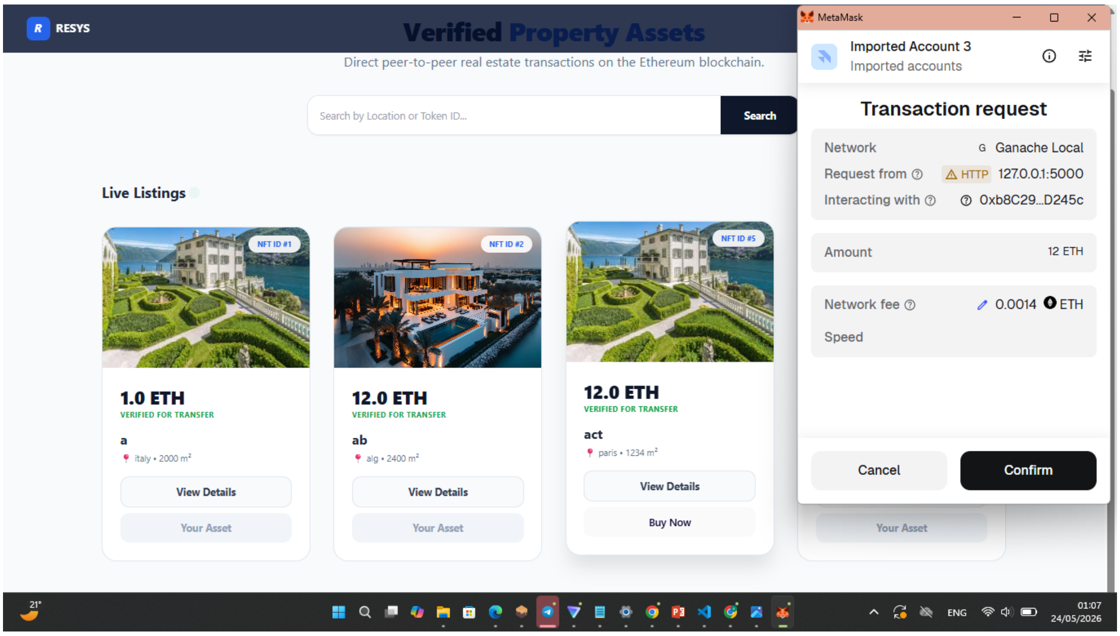

Figure 3.20: MetaMask Transaction Confirmation for Property Purchase

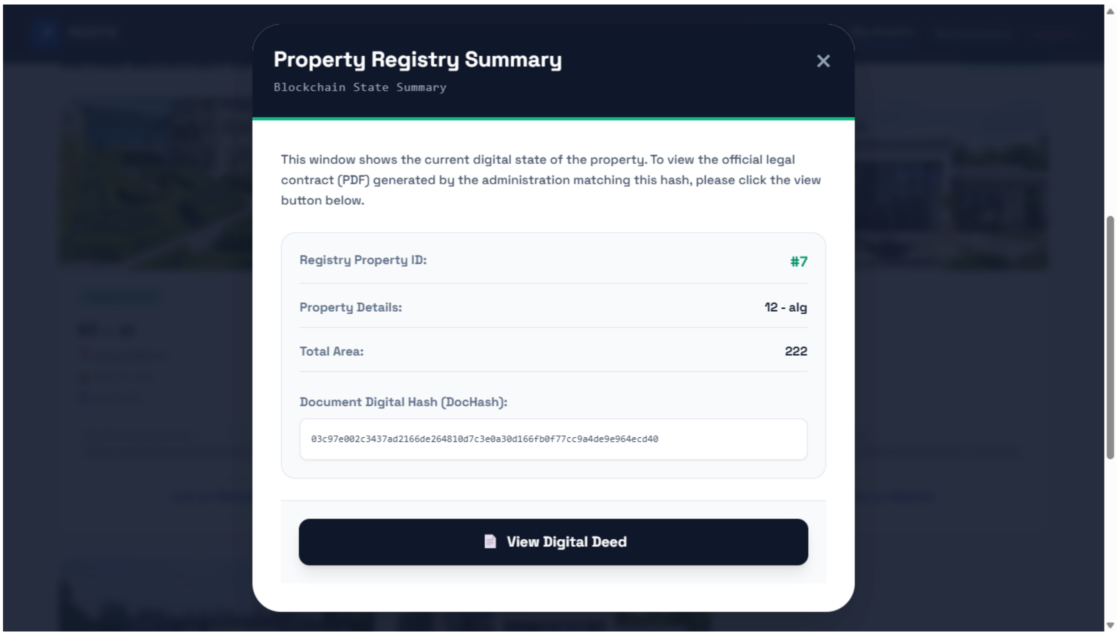

Figure 3.21: Viewing the Updated Registry Summary After Purchase

RESYS GOVERNMENTAL REGISTRY

UPDATED TITLE DEED (OWNERSHIP TRANSFER)

This certified document confirms that the legal ownership of the property registered under Blockchain Token ID [7] has been successfully transferred.

NEW LEGAL OWNER WALLET

0xD451571A77A61E03798351D42c188813B3549991

SMART CONTRACT AUTHENTICATION

AUTOMATED NODE SIGNATURE:

Executed via Smart Contract

Update Timestamp: 1779585317

Scan the QR Code to verify authenticity.

Figure 3.22: The Newly Generated PDF Title Deed

3.5.8 Administrative Verification

To guarantee uncompromising data integrity prior to final tokenisation, the Governor (Authority Admin) utilises an advanced, integrated document audit interface (User Initial Proof Verification) equipped with an embedded PDF viewer. This allows the administrator to cross-examine uploaded legal deeds against the entered metadata fields.

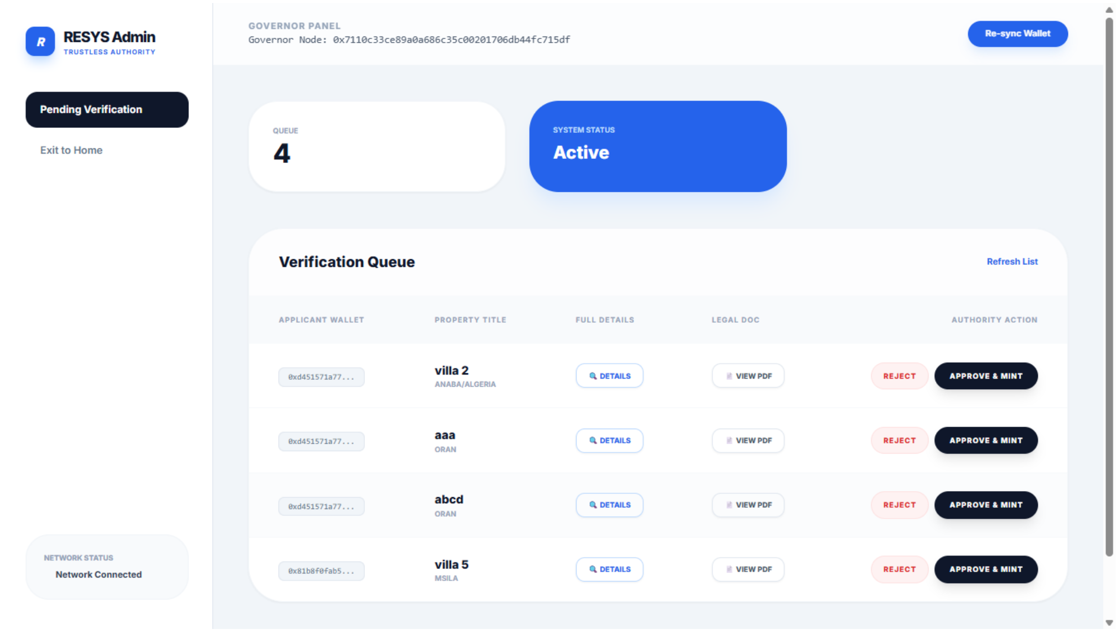

Figure 3.23: Administrator's Dashboard Showing the Pending Request Queue

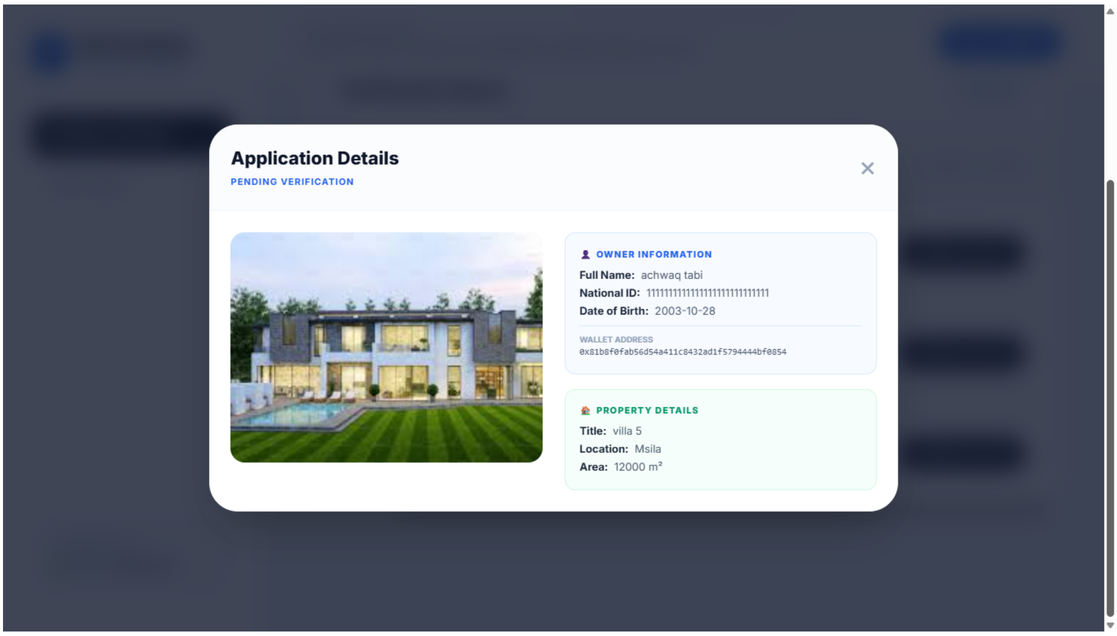

Figure 3.24: Verification Details of a Request

3.5.9 Decision Making: Approval or Decentralised Minting

Following deep verification, the Governor commits to one of two mutually exclusive state transactions:

- **Rejection:** Destroys the off-chain staging record and triggers an administrative confirmation message informing the owner of the application's denial.
- **Approval and Minting:** Triggers an explicit smart contract invocation via MetaMask. The Governor cryptographically signs the Approve and Mint payload. Upon block inclusion, the architecture executes an atomic minting sequence: it maps the verified metadata and explicitly registers the asset on the blockchain.

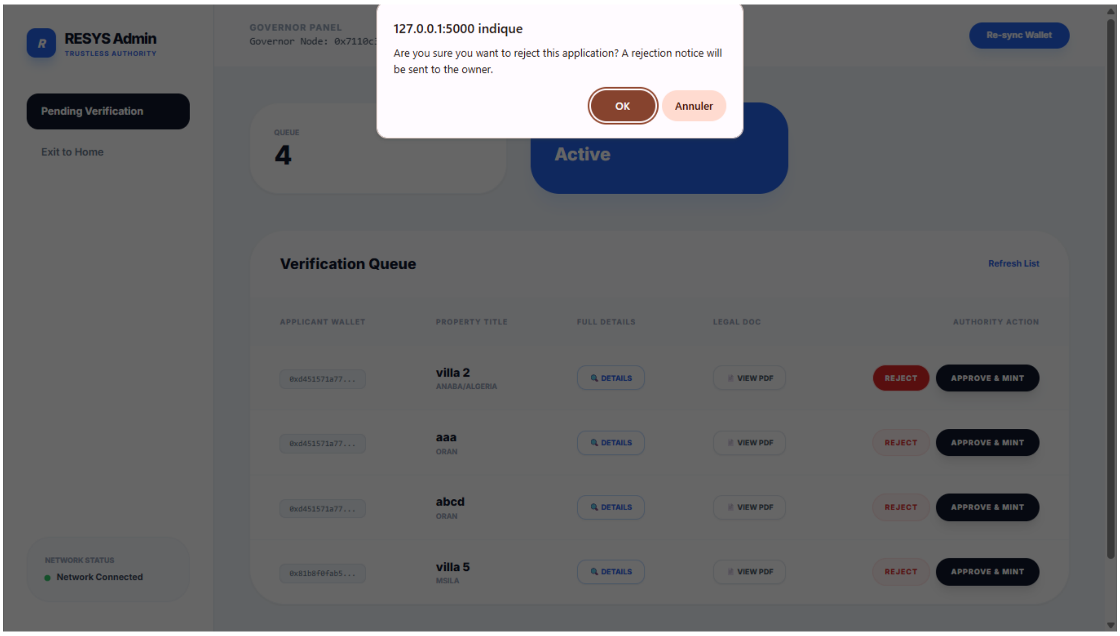

Figure 3.25: Rejecting a Non-compliant Request

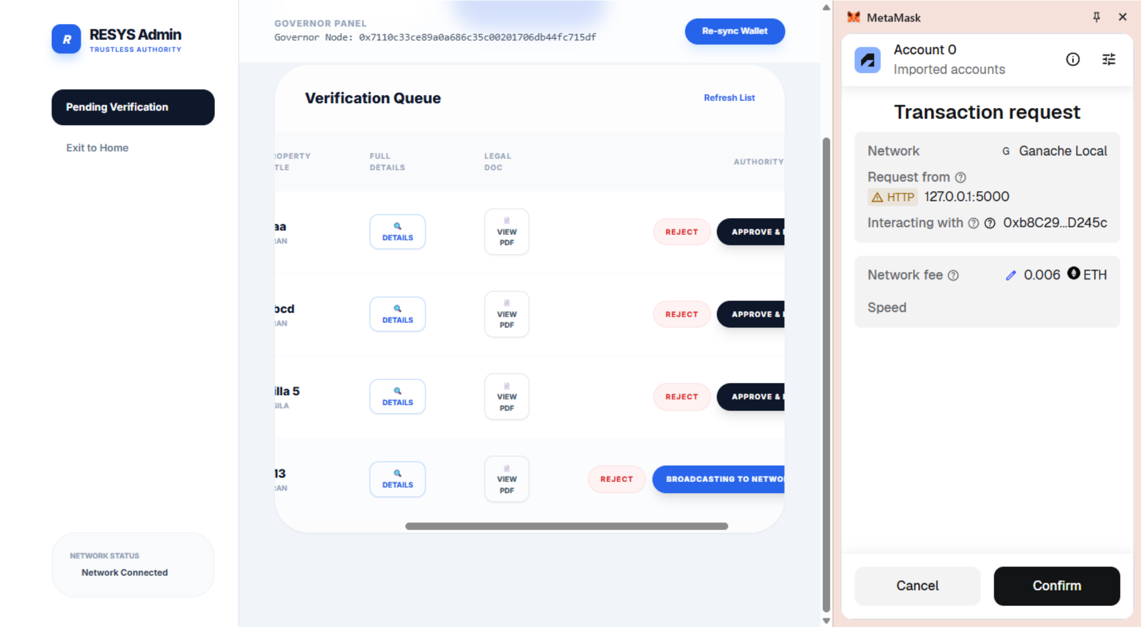

Figure 3.26: Approving a Request and Minting it on the Blockchain

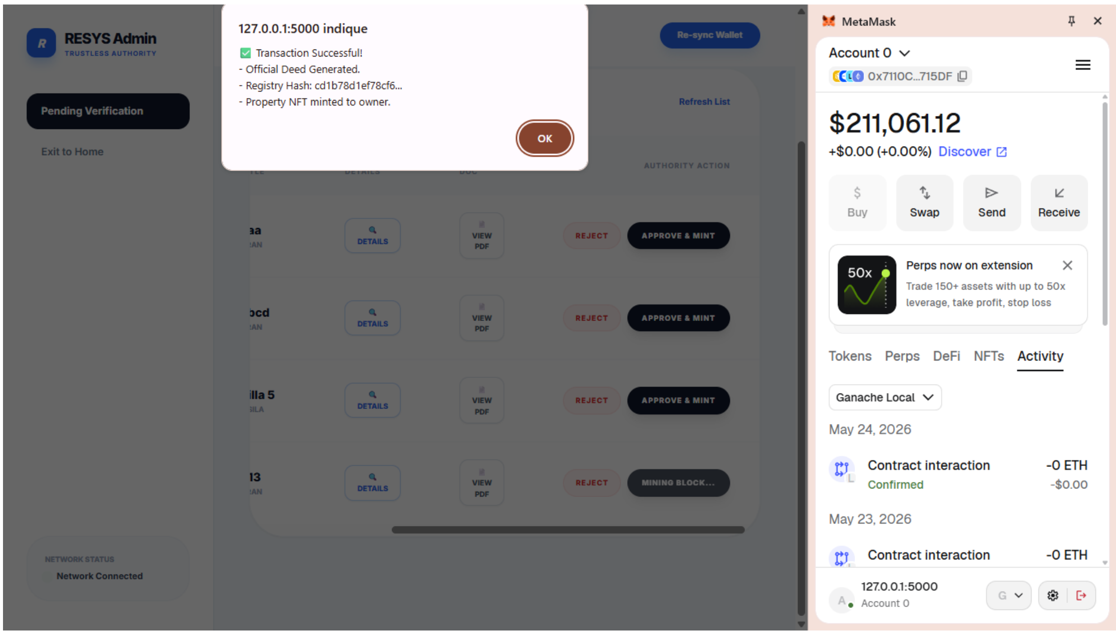

Figure 3.27: The Registration Successfully Completed

3.6 System Evaluation and Gas Consumption Analysis

In Ethereum-based smart contracts, execution efficiency is deeply tied to economic viability. Every state-modifying operation executed on the Ethereum Virtual Machine (EVM) consumes a deterministic amount of computational effort, measured in **Gas**. To evaluate the computational efficiency and practical feasibility of the proposed **RESYS** platform, empirical simulations were conducted using the **Ganache** local deterministic blockchain environment.

The evaluation focuses on benchmarking the gas consumption of core smart contract functions to validate our architectural choice of a hybrid (On-chain/Off-chain) model.

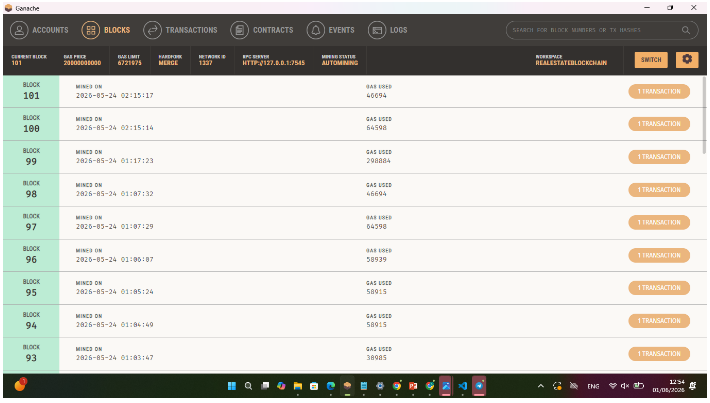

Figure 3.28: Terminal output from Ganache demonstrating transaction receipts and their associated gas usage.

3.6.1 Gas Consumption Results

As illustrated in Figure , the execution traces reveal deterministic gas costs. Table provides a comprehensive breakdown of the gas expenses incurred during the lifecycle of the `RealEstateRegistry` smart contract, transitioning from initial deployment to active market operations.

Function / Operation	**EVM Transaction Type**	**Gas Consumed**	**Cost Characterization**
`Contract Deployment`	Contract Creation	$\sim$ 2,845,312	High (One-time)
`registerProperty`	State Write (Minting NFT)	$\sim$ 343,514	Mod-High (Admin)
`listForSale`	State Modification	$\sim$ 64,598	Low (Market Setup)
`purchaseProperty`	Value Transfer & Swap	$\sim$ 132,540	Moderate (Atomic)
`updateLegalDocument`	Storage Update (Hashing)	$\sim$ 46,694	Low (Sync)
`getPropertyDetails`	Call / View Function	**0**	Free (Local RPC)

Table 3.2: Empirical gas consumption metrics for core smart contract operations.

3.6.2 System Architecture and Cost Analysis

The empirical data gathered from the Ganache simulation yields critical architectural insights:

1. **On-Chain Storage**
   The relatively low gas consumption of operations like `registerProperty` (343,514 gas) and `updateLegalDocument` (46,694 gas) directly validates the efficiency of our

smart contract design. Instead of attempting to store raw, heavy digital files (such as actual property images or complete PDF legal documents) directly inside the Ethereum Virtual Machine (EVM) state—which would cause prohibitive, exponential gas penalties,the contract is optimized to commit only short text fields and absolute cryptographic fingerprints (`docHash` and `imageHash`). This keeps execution costs within practical and economically viable boundaries.

2. **The Security-Cost Trade-off**
   Executing complex operations, such as the `purchaseProperty` function, consumes additional computational overhead. This includes deploying the `nonReentrant` modifier from OpenZeppelin's `ReentrancyGuard` and handling the dynamic balance adjustments required for an atomic swap. While state modifications and external ether transfers incur non-trivial gas costs, this expenditure is strictly necessary to guarantee compile-time and runtime immunity against reentrancy vectors, establishing absolute asset security over mere cost minimisation.

3. **Zero-Cost Read Operations**
   Functions designated with the `view` modifier, specifically `getPropertyDetails` and `getTotalProperties`, yield a gas cost of absolute zero. Because these operations only read the blockchain state via local RPC calls without causing a state mutation, citizens and administrative agents can query property data seamlessly and indefinitely without incurring any financial burden.

3.7 Future Work

Building upon the current implementation of the RESYS platform, future enhancements will focus on achieving true decentralisation and increasing system scalability. Key planned developments include:

- **Decentralised Storage Integration (IPFS):** To eliminate the potential single point of failure represented by the current Flask backend, the physical storage of property deeds and related documents will be migrated to the InterPlanetary File System (IPFS). By storing document hashes on the blockchain while keeping the actual files on IPFS, the system will achieve immutable, content-addressed, and fully decentralised data management.

- **Layer 2 Scalability Solutions:** Transitioning from the current local development environment to Ethereum Layer 2 solutions (e.g., Arbitrum or Polygon) to significantly reduce transaction costs and increase throughput for real-time property trading.

- **Formal Security Audits:** Subjecting the smart contracts to formal verification processes to mathematically prove the absence of logical vulnerabilities, ensuring long-term integrity before deployment on the Mainnet.

3.8 Conclusion

In conclusion, this chapter has successfully connected the theoretical foundations of blockchain technology with the practical realisation of the proposed real estate platform (RESYS). The application of a strict UML modelling approach yielded a clear structural and behavioural blueprint for the hybrid architecture. This meticulous design phase allowed for the seamless integration of an off-chain Flask backend that handles automated document processing and cryptographic hashing, with on-chain Solidity smart contracts that ensure immutable property tokenisation and secure transaction execution. Moreover, the in-depth demonstration of the Graphical User Interfaces (GUIs) showed that complex Web3 workflows (e.g., cryptographic authentication through MetaMask, administrative vetting, atomic peer-to-peer trading) can be effectively abstracted into a seamless and intuitive user experience. Ultimately, the successful implementation of this hybrid system demonstrates the technical feasibility of decentralised architectures in removing traditional intermediaries and creating a transparent, tamper-proof and highly efficient ecosystem for real estate management.

Conclusion

The real estate sector suffers from deep-rooted structural problems, namely lack of transparency, administrative complexity, and document forgery risks...., issues that make blockchain an architecturally justified choice rather than a mere technological trend.

RESYS was developed as a practical embodiment of this approach, adopting a hybrid architecture that combines off-chain administrative processing with immutable on-chain registration. The system demonstrated its ability to cover the full property ownership lifecycle: from registration and administrative verification, through the cryptographic minting of deeds as NFTs, to decentralized marketplace trading and automated ownership transfer.

However, several key limitations must be acknowledged. The system was tested exclusively on a local environment without deployment on a public network, and the legal standing of its generated deeds under Algerian legislation remains unresolved.

It is worth noting that this project was developed under the assumption that Algeria has recognized the necessity of embracing digital currencies and is actively moving towards their adoption, a prerequisite without which the practical deployment of any blockchain-based property system remains institutionally constrained, regardless of its technical maturity.

Accordingly, this work does not present a production-ready system, but rather a validated proof-of-concept prototype whose architectural decisions are defensible and whose development trajectory is clearly defined: integrating IPFS for decentralized storage, migrating to Layer-2 networks to reduce transaction costs, conducting a formal security audit of the smart contracts, and verifying legal compliance with the national legislative framework.

References

[1] S. Khare, A. Ashraf, M. M. Yousuf, and M. Rashid, “Blockchain: Structure, uses, and applications in IoT,” in *Blockchain Security in Cloud Computing*, Springer International Publishing, 2022, pp. 131–144. DOI: 10.1007/978-3-030-70501-5_6

[2] People's Democratic Republic of Algeria, *Algerian civil code, article 683*, Order No. 75-58 of 26 September 1975, Official Journal of the Republic of Algeria, Special Issue, 1975.

[3] People's Democratic Republic of Algeria, *Law no. 90-25 of 18 november 1990 on land orientation*, Official Journal of the People's Democratic Republic of Algeria, No. 49, 1990.

[4] New Western. “Real estate market,” Accessed: Jun. 3, 2025. [Online]. Available: https://www.newwestern.com/blog/real-estate-market/

[5] World Bank, “Land governance assessment framework,” World Bank, 2020.

[6] OECD, “Real estate market trends and economic impact,” OECD Publishing, 2022.

[7] UN-Habitat, “Land administration systems,” United Nations Human Settlements Programme, 2021.

[8] ARC Advisory Group. “What is digitization, digitalization, and digital transformation?” Accessed: Jun. 3, 2025. [Online]. Available: https://www.arcweb.com/blog/what-digitization-digitalization-digital-transformation

[9] SAGE Journals. “Digital platforms,” Accessed: Jun. 3, 2025. [Online]. Available: https://journals.sagepub.com/

[10] S. Nakamoto. “Bitcoin: A peer-to-peer electronic cash system,” Accessed: Apr. 30, 2026. [Online]. Available: https://bitcoin.org

[11] B. Saritha, “Blockchain technology: Features, characteristics with a focus on its types,” *Int. J. Engineering Technology and Management Sciences*, vol. 8, no. 2, pp. 100–107, 2024. DOI: 10.46647/ijetms.2024.v08i02.012

[12] MoonPay. “What are blockchain nodes?” Accessed: Apr. 30, 2026. [Online]. Available: https://www.moonpay.com/learn/blockchain/what-are-blockchain-nodes

[13] N. Moosavi, H. Taherdoost, N. Mohamed, M. Madanchian, Y. Farahani, and I. U. Khan, “Blockchain technology, structure, and applications: A survey,” *Procedia Computer Science*, vol. 237, pp. 645–658, 2024.

[14] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[15] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *Journal of Cryptology*, vol. 3, no. 2, pp. 99–111, 1991.

[16] V. Buterin, "A next-generation smart contract and decentralized application platform," Ethereum White Paper, 2014. Accessed: Apr. 30, 2026. [Online]. Available: `https://ethereum.org/whitepaper`

[17] Financial Action Task Force (FATF), "Reports and guidelines on financial systems and blockchain," FATF, 2021.

[18] B. Makala and A. Anand, "Blockchain and land administration," World Bank, 2019.

[19] Microsoft Corporation, *Visual studio code: Code editing. redefined*, `https://code.visualstudio.com`, Accessed: May 2026, 2024.

[20] Pallets Projects, *Flask: A lightweight WSGI web application framework*, `https://flask.palletsprojects.com`, Accessed: May 2026, 2024.

[21] Ethereum Foundation, *Web3.py: A python library for interacting with ethereum*, `https://web3py.readthedocs.io`, Accessed: May 2026, 2024.

[22] Ethereum Foundation, *Solidity: A statically-typed programming language for ethereum smart contracts*, `https://docs.soliditylang.org`, Accessed: May 2026, 2024.

[23] OpenZeppelin, *Openzeppelin contracts: A library of secure and reusable smart contracts*, `https://docs.openzeppelin.com/contracts`, Accessed: May 2026, 2024.

[24] Truffle Suite, *Ganache: A personal blockchain for ethereum development*, `https://trufflesuite.com/ganache`, Accessed: May 2026, 2023.

[25] MetaMask / Consensys, *Metamask: A crypto wallet and gateway to blockchain apps*, `https://metamask.io`, Accessed: May 2026, 2024.

[26] Richard Moore, *Ethers.js: A complete and compact library for interacting with the ethereum blockchain*, `https://docs.ethers.org`, Accessed: May 2026, 2024.