

Decentralized IP Protection System

PHASE II REPORT

Submitted by

I Mercia Jeno 22011102026

Jai Shree R S 22011102028

Nishasi S 22011102064

in partial fulfilment for the award of the degree of

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
(INTERNET OF THINGS)**



**DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING
SCHOOL OF ENGINEERING
SHIV NADAR UNIVERSITY CHENNAI**

APRIL 2026

SHIV NADAR UNIVERSITY CHENNAI

BONAFIDE CERTIFICATE

Certified that this report titled “**Decentralized IP Protection System**” is the bonafide work of **I Mercia Jen**o (Reg. No: **22011102026**), **Jai Shree R S** (Reg. No: **22011102028**), **Nis-hani S** (Reg. No: **22011102064**) who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T. Nagarajan

Professor & Head

Department of Computer Science &

Engineering

School of Engineering

Shiv Nadar University Chennai,

Kalavakkam -603110.

SIGNATURE

Dr. Rourab Paul

Assistant Professor

Department of Computer Science &

Engineering

School of Engineering

Shiv Nadar University Chennai,

Kalavakkam -603110.

ABSTRACT

The protection of Intellectual Property (IP) plays a critical role in creating a system that aims at promoting creativity and innovation but the conventional centralized IP management systems may be associated with time-consuming approval procedures, high administrative expenses, minimal international availability and the possibility of data manipulation or loss of records. These problems create challenges to the creators in demonstrating actual ownership particularly in disputes. On this basis, to overcome these challenges, this project proposes a decentralized IP registration and verification system that works based on blockchain technology in order to guarantee trust, transparency, and security. The given system places a cryptographic hash of the digital property of the user on the Ethereum blockchain, offering both immutable and tamper-proof proof of ownership with time-stamped information, but skews the proper content away. The registration and verification procedures carried out without intermediaries are automated in smart contracts written in Solidity and executed in the Sepolia Testnet. There is a React-based web application where users communicate with the platform, with the assistance of a secure authentication and signing of transactions using MetaMask and the ability to communicate with blockchain through Ethers.js. By lever The decentralized and distributed character of aging blockchain, the system does not produce the points of failure, minimises operational expenses, accelerates the verification, enhances universal accessibility. On the whole, this project shows that blockchain can update the system of managing IP rights and make it more efficient, transparent, and resistant to fraud, which will guarantee creators maintain. complete freedom and appreciation of their effort.

Keywords — Ethereum, MetaMask, Intellectual Property, Solidity, Blockchain security, Sepolia Testnet, Decentralized Application (DApp).

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have provided invaluable guidance, support, and encouragement throughout the course of this project.

First and foremost, We are deeply grateful to **Dr. T. Nagarajan**, Professor and Head, Department of Computer Science and Engineering, School of Engineering, Shiv Nadar University Chennai, for his steadfast guidance and unwavering encouragement. His insights into the field of artificial intelligence and cognitive science have been instrumental in shaping the direction of this research.

We would like to extend our heartfelt thanks to our supervisor, **Dr. Rourab Paul**, Assistant Professor, Department of Computer Science and Engineering, School of Engineering, Shiv Nadar University Chennai, for his mentorship and commitment to the success of this project. His expertise in the domain of Blockchain has been invaluable, providing us with the clarity and focus needed to tackle complex challenges and refine the methodology of this work.

We also acknowledge with gratitude the support of our family, whose understanding and encouragement have been crucial to the implementation of this project. Their unwavering belief in us has been a constant source of motivation and strength.

I Mercia Jenio
(22011102026)

Jai Shree R S
(22011102028)

Nishani S
(22011102064)

TABLE OF CONTENTS

CHAPTER	PAGE NO.
ABSTRACT	i
LIST OF TABLES	v
LIST OF FIGURES	vi
LIST OF SYMBOLS, ABBREVIATIONS	vi
1 INTRODUCTION	1
1.1 BACKGROUND	1
1.1.1 Concept	1
1.1.2 Motivation	2
2 LITERATURE SURVEY	5
2.1 BLOCKCHAIN-BASED INTELLECTUAL PROPERTY MANAGEMENT	5
2.2 SUMMARY OF RESEARCH GAPS	6
3 METHODOLOGY	10
3.1 ASSET REGISTRATION AND ORIGINALITY VERIFICATION	11
3.1.1 Step-by-Step workflow	11
3.2 LICENSING, PRICING, AND REVENUE DISTRIBUTION	12
3.2.1 Step-by-Step Licensing Workflow	12
3.2.2 Dynamic Step-Based Pricing Model	13
3.2.3 Co-ownership of the License	14
3.2.4 Step-by-Step Revenue Distribution Workflow	16
4 EXPERIMENTS	6

4.1	SYSTEM IMPLEMENTATION RESULTS	6
4.2	PERFORMANCE ANALYSIS	6
4.3	BLOCKCHAIN COMPARISON	7
4.4	SECURITY ANALYSIS	7
4.5	DISCUSSION	8
5	CONCLUSION	8
5.1	KEY CONTRIBUTIONS	8
5.2	LIMITATIONS AND FUTURE WORK	8
	REFERENCES	11
A	Appendix	12
A.1	IP registry	12
A.2	Licensing	13

LIST OF TABLES

1.1	Comparison of our project with existing blockchain-based IP management systems	4
2.1	Limitations in Existing Literature	5
4.1	Smart Contract Gas Consumption Analysis	6
4.2	Comparison of Transaction Cost and Latency Across Blockchain Networks	7
4.3	Comparison of Network Fees Across Blockchain Platforms	7

LIST OF FIGURES

1.1	High Level system architecture	3
3.1	IP registration flow	10
3.2	License purchase flow	12
3.3	Co-ownership flow	15
A.1	User Interface for Asset Registration via MetaMask	12
A.2	Version-based Licensing	13

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

1.1.1 Concept

Intellectual property (IP) is the digital property, like patents, wills, documents, etc. During the digital age, there has been a growing amount of creative and technical work, but the traditional ways to record, administer, license, and enforce these rights have not kept up with the amount of work. The conventional IP management is based on centralised registries, legal intermediaries, as well as manual licensing agreements, which are slow, costly, opaque, and are subject to dispute.

This problem can be solved by blockchain technology. Being a decentralized, immutable, and public register, blockchain allows writing records of ownership and rules of licensing straight into autonomous applications known as smart contracts. These are contracts which automatically impose agreed-upon terms without involving a trusted third party and generate a permanent, tamper-resistant audit trail . The on-chain storage scalability issue of blockchain is also addressed by integrating with decentralised storage systems like the InterPlanetary File System (IPFS) that store the file content off-chain but ties a cryptographic reference to the blockchain. Alternatively tools like Filecoin can be used for storing but here IPFS was an convenient option.

Our project, Decentralised IP Protection System, is implemented on the Ethereum blockchain. It integrates the representation of ownership with non-fungible tokens (NFTs) based on ERC-721, content storage in Pinata, plagiarism detection through perceptual hashing(pHash), and a new version-bound licensing model that is completely implemented through smart contract logic. The system also proposes a simple step-based dynamic pricing that is based on de-

mand and fractional co-ownership, whereby with the help of a per-property ERC-20 share token, creators are permitted to receive revenue and allocate it to co-owners without a central authority.

1.1.2 Motivation

Our project is driven by a list of unresolved and documented weaknesses in current IP management architecture. To begin with, it is already challenging to demonstrate originality of a digital item in a traditional system, where records are susceptible to changes and loss . Second, issuing and enforcing licenses by themselves often necessitate legal practitioners and lengthy negotiations, and thus could be unavailable and prohibitively expensive to small businesses and individuals, plus these legal practitioners cannot be fully trusted. Third, the existing platforms disclose royalty distribution in an opaque and delayed way, and intermediaries absorb a large portion of the revenue before it reaches the original creator.

Other studies on blockchain-based IP management, such as Alqarni[10], have shown the potential of NFT ownership and fractional co-ownership on the Ethereum blockchain as previously demonstrated by other studies such as Fractional Digital Asset Ownership (FDAO) framework, 2020,[3]. Nonetheless, an analysis of 15 similar papers demonstrated that no previous system considers versioning of licenses as a first-class on-chain primitive. Current systems define assets as fixed objects, whenever an author adds to their work the old licenses are implied to be transferable to the new work, which leads to confusion over which version a buyer has purchased . This is a major gap that our project, Decentralised IP Protection System, has implemented it.

Also, all existing systems use either static pricing or dynamic pricing which actually depends on some sort external oracle-based price,like Chainlink etc, which brings about economic inefficiency or external reliance, which inturn might not be trustworthy . Our project is a program which uses a simple version demand step-based pricing, inspired by the bonding curve pricing model . The combination of all these elements version-locked

licensing, demand-based pricing on-chain, and fractional ownership with automatic revenue distribution are the essence of this work.

Figure 1.1 illustrates the overall high-level architecture

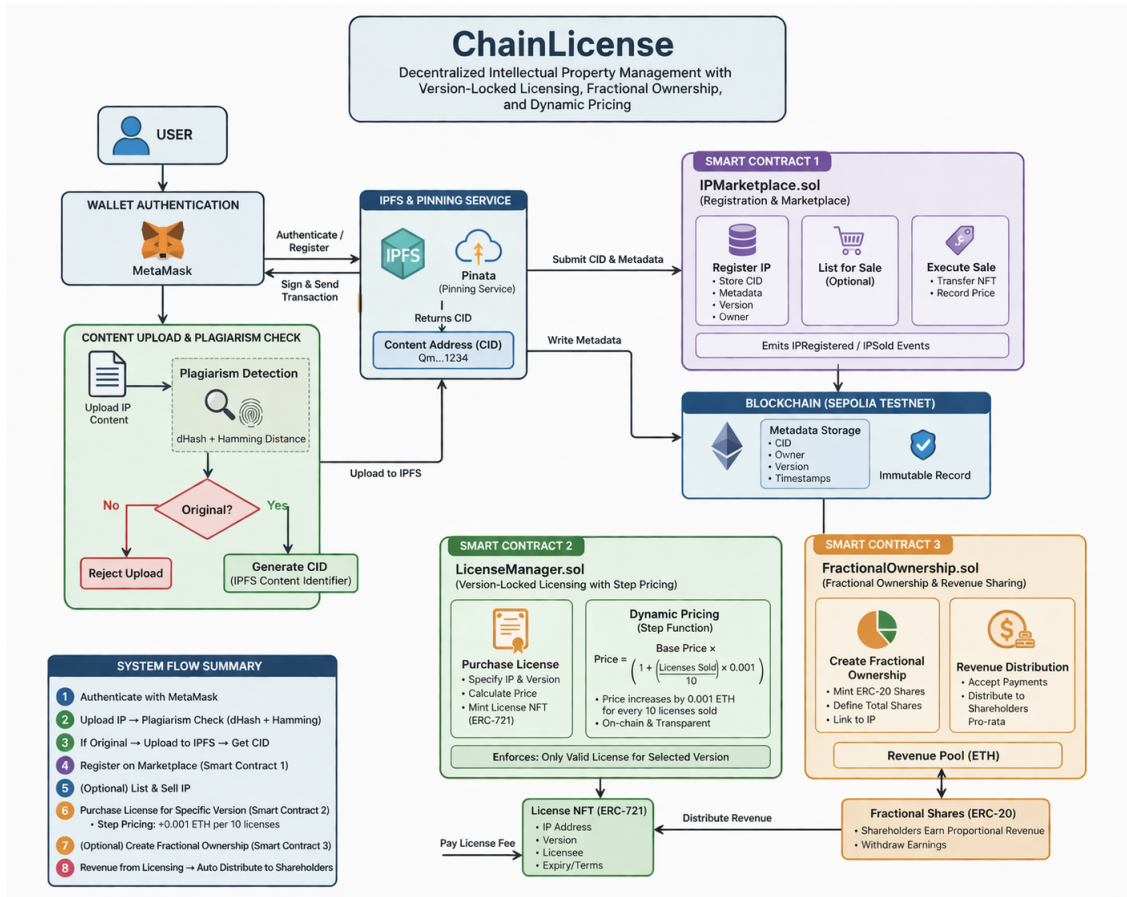


Figure 1.1: High Level system architecture

Table 1.1: Comparison of our project with existing blockchain-based IP management systems

Feature	Alqarni (2024)	FDAO (2025)	Ours
NFT-based ownership	Yes	Yes	Yes
IPFS content storage	No	No	Yes
Version-locked licensing	No	No	Yes
Demand-based pricing	No	No	Yes
Fractional co-ownership	No	Yes	Yes
Revenue distribution	No	No	Yes
Plagiarism detection	No	No	Yes

Table 1.1 presents a feature-level comparison between our project and the two most closely related prior systems identified in the literature review. Alqarni (2024) proposes a blockchain-based IP rights management ecosystem using NFTs and smart contracts, but does not address versioning, dynamic pricing, or fractional ownership. The FDAO framework (2025) introduces fractional NFT-based co-ownership for digital assets including software code, and is the closest prior work in terms of architecture — also implemented using Remix IDE and OpenZeppelin. However, FDAO does not implement version-locked licensing, demand-responsive pricing, or revenue distribution to shareholders. Our project advances by integrating all of these features into a single trustworthy and unified platform, with the version-locked licensing mechanism representing an original contribution not found in any of the fifteen papers surveyed so far.

CHAPTER 2: LITERATURE SURVEY

A comprehensive review of existing literatures was conducted to analyse the limitations in blockchain-based intellectual property (IP) management, decentralized storage, smart contract licensing, and NFT-based ownership. The motive was to find out the current gaps in our idea that was already implemented before and to see how we can proceed in our way.

Table 2.1: Limitations in Existing Literature

Reference	Identified Limitation
[1]	Does not integrate decentralized storage, limiting scalability
[2]	Lacks direct ownership transfer mechanisms
[3]	Relies on centralized storage, reducing decentralization
[4]	Does not incorporate decentralized file storage such as IPFS
[5]	Limited access control and enforcement mechanisms
[6]	Focuses only on storage without ownership or licensing
[7]	Lacks automated originality verification
[8]	Does not provide ownership or licensing functionality
[9]	Supports storage but lacks ownership integration
[10]	No marketplace or real-time interaction support

2.1 BLOCKCHAIN-BASED INTELLECTUAL PROPERTY MANAGEMENT

Blockchain technology has been widely explored for intellectual property protection due to its immutable property. Bhadauria et al. [1] proposed a blockchain-based copyright protection mechanism using digital watermarking; however, the system does not support decentralized storage, thus limiting scalability and also is expensive as the IP is stored directly on-chain . Yuan et al. [2] introduced a privacy-preserving authentication model but did not

address ownership transfer between users.

Chandratre and Pathak [3] presented a timestamp-based ownership verification approach, which might not be the effective way to prevent duplicate records . Lee [5] discussed blockchain-based copyright protection frameworks, but nothing mentioned on ownership of the records.

2.2 SUMMARY OF RESEARCH GAPS

The literature highlights several key limitations in existing systems:

- Lack of integration between ownership, storage, and licensing mechanisms
- Absence of version-specific licensing models
- Limited support for automated revenue distribution
- Dependence on centralised storage
- No efforts on reducing the gas fees

These gaps are the real motive for us to enhance the existing ones and develop an effective solution to the problem.

CHAPTER 3: METHODOLOGY

The methodology is proposed to be able to offer a decentralised, secure and scalable system of digital intellectual property management. The blockchain technology is used in system architecture, and it is also combined with decentralised storage and mechanisms of client-side document verification to provide transparency, immutability, and trustless functionality. The implementation is done in a modular system that includes user interaction, data validation, decentralised storage and smart contract execution.

The interaction takes place between the user and the application (dApp), a web-based, decentralised program, created with React and Vite. The authentication of the wallet and signing of transactions is done via the use of MetaMask, which provides safe communication between the user and Ethereum blockchain. Distribution of revenue (among others), registering assets, licensing, and all crucial functions are implemented by using smart contracts that are compiled in Solidity, and run on the Sepolia test-network.

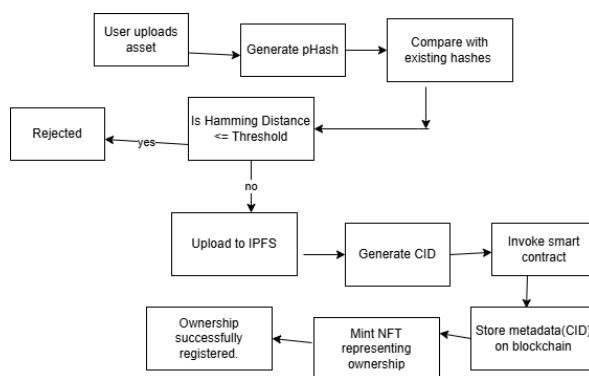


Figure 3.1: IP registration flow

3.1 ASSET REGISTRATION AND ORIGINALITY VERIFICATION

The initial step of the methodology aims at the registration of assets and originality tests. When a user registers a digital file, the system checks it by client-side plagiarism detection with the help of perceptual hashing (dHash). The image uploaded is translated into a fixed sized grayscale image and a 64-bit hash is created involving differences in pixel intensities. This hash is further compared with already registered hash by Hamming distance in order to identify similarity. Based on our experiments, we choose 10 to be the threshold . If the hamming distance is below 10, then the documents are similar .

When the similarity rate falls under a threshold, the upload is declined to avoid a duplicated content. Otherwise, the file is deemed to be original and uploaded in the InterPlanetary File System (IPFS) through the Pinata API. A smart contract that is stored on the blockchain using the Content Identifier (CID) representing the file results in a completed file. A token is issued as ERC-721 token which identifies that the ownership of the asset has been created, and it has immutable, verifiable evidence of ownership.

3.1.1 Step-by-Step workflow

The workflow of asset registration is described as follows:

1. User uploads digital asset through dApp.
2. System generates perceptual hash for the uploaded asset.
3. Hamming distance comparison is performed with existing hashes stored already.
4. If similarity threshold is below 10, upload is rejected
5. If it is valid, asset is uploaded to IPFS and CID is generated for that asset.
6. Smart contract is invoked to register asset metadata

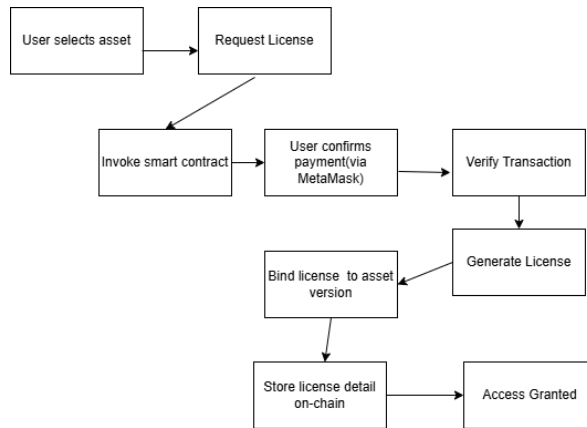


Figure 3.2: License purchase flow

7. ERC-721 NFT is minted to represent ownership of that asset.

3.2 LICENSING, PRICING, AND REVENUE DISTRIBUTION

The second step of the procedure is the licensing, pricing, and the revenue distribution. Upon a user requesting to access a digital asset, the system calls the smart contract function to buy a license. Pricing is achieved by utilizing a dynamic step-based pricing model that is on-chain, meaning that the price of a license will be raised by a fixed number of steps depending on the quantity of licenses that have been bought. Here we have used 10 as the step value for simplicity and is not ideal in real world.

Every license that is issued is fixed to a particular version of the asset, which is stored as the smart contract. The license validation function supports the authorized access to the system by admitting only the access to the version that corresponds to the version of the purchased license. This provides strong version control and guarding against the abuse of licenses in updated version of content.

3.2.1 Step-by-Step Licensing Workflow

The process of license purchase and validation is described as follows:

1. User selects a digital asset in the marketplace
2. User requests a license of that particular version of the asset
3. Smart contract is invoked here
4. User confirms transaction via MetaMask
5. System verifies transaction success
6. License is generated and given to the buyer
7. License verification check is shown in the UI

3.2.2 Dynamic Step-Based Pricing Model

Our system places a deterministic, step based dynamic pricing system on the issuance of licenses. The price of the license increases after every k (eg:10) license sold.

Let:

- P_0 = base price of the license (eg:- 0.01 ETH)
- n = total number of licenses sold
- k = step size (10 licenses)
- ΔP = price increment (eg:- 0.001 ETH)

The license price $P(n)$ is defined as:

$$P(n) = P_0 + \left\lfloor \frac{n}{k} \right\rfloor \times \Delta P$$

This implies that the price increases by a fixed increment after every 10 licenses sold.

Example:

- For $n = 0$ to 9 : $P(n) = 0.01$ ETH
- For $n = 10$ to 19 : $P(n) = 0.011$ ETH
- For $n = 20$ to 29 : $P(n) = 0.012$ ETH

Design Justification: The step pricing is selected as opposed to continuous pricing functions to reduce computing complexity and gas expenses. Being relatively straightforward due to all data being integer divisions and the fixed increments, it can be successfully performed in the smart contract without any need to access external data feeds or do any operations with floating points.

Limitations: The model is not based on the current market dynamics and user behavior and improvements can be made in future which involve adaptive or curve based pricing mechanisms. The given model can be viewed as a simplified form of a discrete bonding curve scaled down to execute a smart contract at low costs.

3.2.3 Co-ownership of the License

The money obtained through purchasing licenses is stored in a revenue pool which is controlled by smart contracts. According to the amount of ERC-20 token, shareholders will be able to withdraw corresponding share of revenue. This is a complete automated and transparent method of distribution with no need to go through intermediaries. Moreover, the shareholders receive special treatment privileges (access to free licenses) as an additional incentive to be part of the ecosystem.

The entire methodology shows that an integrated approach of using blockchain, decentralised storage and cryptographic validation is developed and used to build a powerful intellectual property management system. Its design is safe, transparent and scalable and it overcomes major limitations and also solves some of the real problems faced by us.

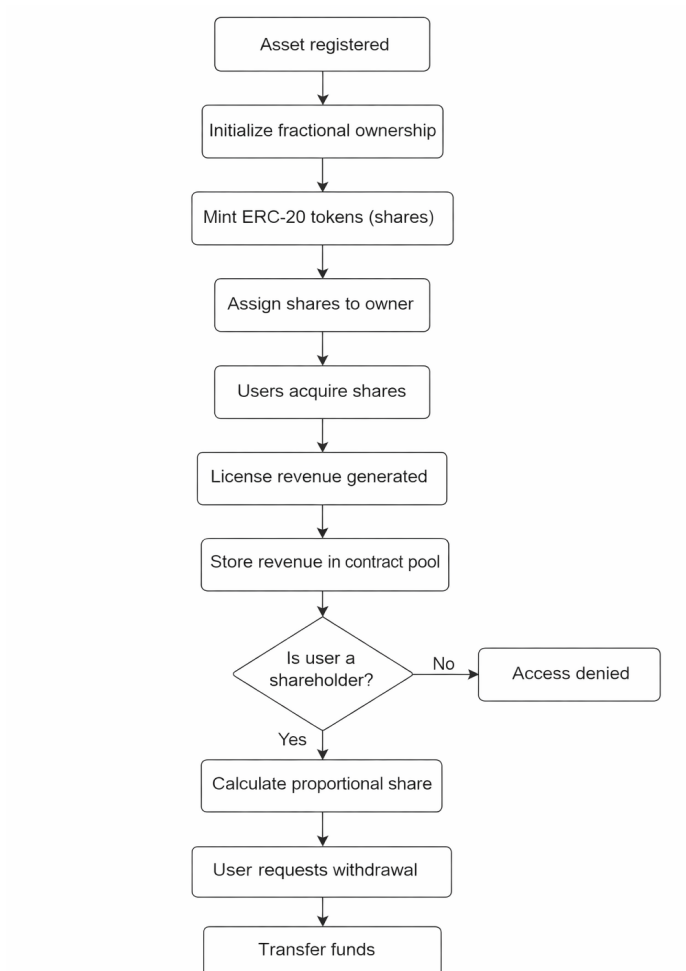


Figure 3.3: Co-ownership flow

3.2.4 Step-by-Step Revenue Distribution Workflow

The revenue distribution process is described as follows:

1. Asset is registered on our platform
2. Fractional ownership is initialized for the registered asset
3. ERC-20 tokens representing shares are minted(eg:- 1000 shares)
4. Shares are assigned to the owner
5. Co-owners acquire shares of the asset
6. Revenue is generated from the license purchases
7. Revenue is stored in the smart contract pool
8. System checks if the user is a shareholder
9. Proportional share is calculated
10. Co-owner requests withdrawal
11. Funds are transferred to the co-owners from the revenue pool

CHAPTER 4: EXPERIMENTS

This chapter demonstrates the experiments done using our system . The system was tested on the Ethereum Sepolia test network . The assessment is aimed at testing the performance of smart contracts, estimating gas usage, transaction latency, and comparing performance on various EVM-based blockchain systems, like Polygon Amoy, BNB chain, Monad testnet.

4.1 SYSTEM IMPLEMENTATION RESULTS

Our system is a decentralised application, combining a React-based frontend, a wallet application (MetaMask) and Solidity smart contracts along with an IPFS-based decentralised storage solution. More focus was put on licensing, revenue distribution, plagiarism check.

4.2 PERFORMANCE ANALYSIS

The system was tested in terms of its gas usage and the speed at which transactions are carried out on the Ethereum Sepolia network.

Table 4.1: Smart Contract Gas Consumption Analysis

Operation	Gas Consumption (Approx)
Asset Registration (ERC-721 Minting)	120,000 – 180,000 gas units
License Purchase	70,000 – 100,000 gas units
Revenue Withdrawal	40,000 – 60,000 gas units
CID Storage (IPFS Hash)	Constant gas cost (fixed-length storage)

The Sepolia network took a range of 8 to 12 seconds in confirming the transactions based on the network congestion. The findings indicate that, though the system is operationally efficient, latency and transaction cost are also critical measures of the scalability of a system.

4.3 BLOCKCHAIN COMPARISON

The performance of ChainLicense on different EVM-compatible blockchain networks, such as Ethereum Sepolia, Polygon Mumbai, and Monad Testnet, were compared. The purpose was to find out which network performed better based on our use case and also what are the challenges faced.

Table 4.2: Comparison of Transaction Cost and Latency Across Blockchain Networks

Operation	Ethereum (Sepolia)	Polygon (Mumbai)	Monad Testnet
Upload / Registration	~0.0019 ETH, ~8 sec	~0.0004–0.0006 MATIC, ~2–4 sec	~0.00002–0.00006 MON, ~1-3 sec
Licensing	~0.0002 ETH, ~10 sec	~0.00005–0.0001 MATIC, ~3–5 sec	~0.000005–0.00002 MON, ~0.5–1 sec

Table 4.3: Comparison of Network Fees Across Blockchain Platforms

Blockchain	Fee Characteristics	Observation
Ethereum (Sepolia)	High and dynamic gas fees based on network demand	Reliable and secure but comparatively expensive
Polygon (Mumbai)	Low transaction fees due to Layer-2 scaling	Suitable for scalable and cost-efficient applications
Monad Testnet	Extremely near-zero fees with high throughput design	Great performance but still under development

The values of Ethereum are determined by deploying experimental use in Sepolia test network. The Polygon and Monad values are computed by using the same amount of gas and the parameters of the network that are known as gas price and block time. This trade-off is demonstrated by this comparison between the options of decentralisation, cost, and performance.

4.4 SECURITY ANALYSIS

Security is essential because of financial and ownership implications that there is with blockchain-based intellectual property management. Our system ,though not fully secure, has made an effort to improve on security. Below are the practices we followed.

1. Reentrancy Protection: Smart contract functions that require the transfer of funds, e.g. the revenue withdrawal, follow the pattern of the checks-effects-interactions. External calls are calculated after state updates to avoid reentrancy attacks so that malicious contracts are not able to withdraw money repeatedly.

2. Access Control Enforcement: The ownership validation is used to control such critical operations as asset registration, licensing, and withdrawal of revenues. Sensitive functions can only be performed by the rightful

3. Input Validation and State Consistency: Any input of the user such as identification of the assets and the parameters of licensing are verified and then executed. This eliminates any non-contractual transactions and on-chain consistency.

4. Resistance to external sources for dynamic pricing: Although blockchain operations are transparent in nature, the price discovery mechanism is deterministic and not dependent on external or time-dependent information, which minimizes the effects of front-running involvement on the price manipulation process.

5. Overflow and Arithmetic Safety: The smart contracts are coded in Solidity versions that provide inbuilt overflow checks, which eliminate arithmetic errors that would result in wrong balances and unplanned behavior.

In general, the system adheres to best practices of smart contract security, offering a reasonable degree of security when it comes to a prototype implementation, but also pointing out the points where this can be improved.

4.5 DISCUSSION

The outcomes of the experiment confirm the efficiency of our system. The system is effective in implementing version-locked licensing, in providing safe ownership monitoring and in making revenue distribution without intermediaries decentralised.

The relative evaluation shows that Ethereum is more secure and reliable but has greater latency and transaction fees. Conversely, Polygon is more scalable and has fewer charges whereas Monad has much lower latency and almost zero transaction charges.

Our experiments directs us to the choice of the blockchain network and also the keep in mind the trade-offs for this particular use case.

CHAPTER 5: CONCLUSION

The proposed project is a decentralised framework to track digital intellectual property with blockchain technology. The system handles weaknesses associated with the traditional IP systems such as a lack of transparency, slower in process etc. The proposed solution would provide secure, transparent, and provable ownership of digital items through integration of Ethereum smart contracts, storage based on IPFS, and cryptographic validation.

A primary contribution of this work is the introduction of version-locked licensing, where each license is bound to a specific version of the asset and enforced directly on-chain. The system also adds fractional ownership through the use of ERC-20 tokens allowing decentralised co-ownership and automated revenue sharing. A fully on-chain dynamic step-based pricing scheme is added to the system to provide further improvement to the system.

5.1 KEY CONTRIBUTIONS

The proposed system integrates multiple components into a unified architecture, addressing several gaps identified in existing literature. Decentralised IP Protection System, by comparison to earlier methods, will incorporate ownership, licensing, pricing, and the distribution of revenue on one decentralised platform.

5.2 LIMITATIONS AND FUTURE WORK

There are a number of limitations which directs us to the future scope.

1. Scalability and Transaction Cost: The system is already running on the Ethereum Sepolia test network, where the cost of transactions and latency may be substantial when the network is heavily loaded. Though it is appropriate to test and validate, to use it extensively,

it would need to migrate to Layer-2 solutions or blockchain networks with high performance in order to enhance the throughput and become cost-effective.

2. Simplified Dynamic Pricing Model: The applied pricing scheme is the step-based one, according to which the price of licenses rises after a specified quantity of purchases. Although it is efficient and is entirely based on-chain, this model fails to acknowledge some of the more complex market dynamics like demand elasticity or user behaviour and can be improved with a more adaptive pricing approach.

3. Limitations of Plagiarism Detection: The existing system employs perceptual hashing (dHash) and Hamming distance of similarity detection which is useful in simplified image comparison. Nevertheless, such a method could fail this task and not identify advanced changes or other forms of data like text, audio or code. More sophisticated algorithms, such as similarity detector based on machine learning, or multi-modal, can be considered.

4. Lack of Cross-Chain Interoperability: The designed system is EVM-compatible network and is not prepared to complement cross-chain asset transfer or licensing. Future research has an opportunity to consider interoperability standards to facilitate a smooth cooperation between different blockchain systems.

5.Security: Our project has few bottlenecks. One of them is using pHash. This algorithm is implemented in client side and hence attacks can happen here. Also this algorithm only works for images and not any other type of media. Advanced algorithms like ZKProofs will be used .

Further development will be done to overcome these limitations by adding scalable blockchain solutions, complex pricing data, better plagiarism detection algorithms to strengthen the system and increase its use to real-world applications.

REFERENCES

- [1] S. Bhadauria, P. Kumar and T. Mohanty, "Intellectual Property Protection using Blockchain and Digital Watermarking," *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Hyderabad, India, 2021, pp. 1–6, doi: 10.1109/ANTS52808.2021.9936909.
- [2] S. Yuan, W. Yang, X. Tian and W. Tang, "A Blockchain-Based Privacy Preserving Intellectual Property Authentication Method," *Symmetry*, vol. 16, no. 5, p. 622, 2024, doi: 10.3390/sym16050622.
- [3] A. Chandratre and A. Pathak, "Blockchain Based Intellectual Property Management," 2019. [Online]. Available: <https://ssrn.com/abstract=3800734>
- [4] F. Miao, "An Intellectual Property Protection Platform Based on Blockchain Technology," in *Proc. ACM Conference*, 2024, pp. 67–74, doi: 10.1145/3689236.3689267.
- [5] K. K. Lee, "Design of a Blockchain-Based Framework for Digital Copyright Protection and Tamper Prevention," 2024. [Online]. Available: <https://ssrn.com/abstract=5580658>
- [6] B. P. Chintal, "Secure Decentralized Storage System Using Blockchain and IPFS," 2025. [Online]. Available: <https://ssrn.com/abstract=5142864>
- [7] D. Putranti and U. Putri, "Enforcement of Copyright Law on Non-Fungible Token (NFT) Through Smart Contracts," *Kosmik Hukum*, vol. 24, no. 1, pp. 40–50, 2024.
- [8] Y. Zhu, C. Lv, Z. Zeng, J. Wang and B. Pei, "Blockchain-based Decentralized Storage Scheme," *Journal of Physics: Conference Series*, vol. 1237, 2019, doi: 10.1088/1742-6596/1237/4/042008.
- [9] J. Benet, "IPFS - Content Addressed, Versioned, Peer-to-Peer File System," *arXiv*

preprint arXiv:1407.3561, 2014.

- [10] A. Alqarni, "A Blockchain-Based Solution for Transparent Intellectual Property Rights Management," *Kybernetes*, 2024, doi: 10.1108/K-04-2024-1074.

APPENDIX A: Appendix

The following appendix content is a supplementary material to support the implementation of our system like some more technical details, screenshots, and outputs of the system. These resources depict the actual implementation and use of the decentralised app on the Sepolia network of Ethereum.

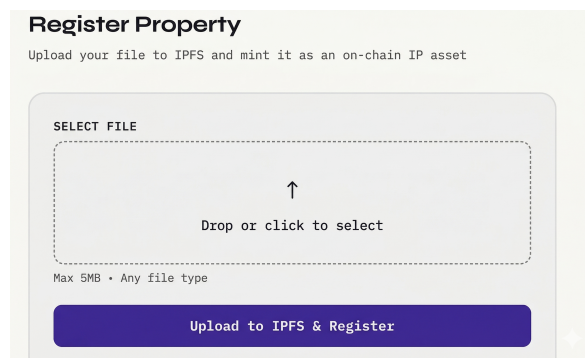


Figure A.1: User Interface for Asset Registration via MetaMask

A.1 IP registry

The figure above is the graphical representation of the front end of our system, as seen on which the user can upload digital assets, and interact with the blockchain using the MetaMask. When a file is chosen, plagiarism is first detected on the client-side but by doing a hashing before sending it to IPFS.

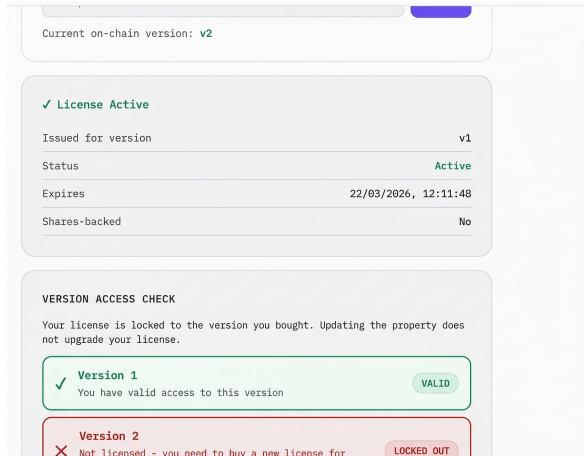


Figure A.2: Version-based Licensing

A.2 Licensing

The figure above is the version based licensing part where we show the users which version of license they have bought and it keeps updating as the seller updates his document.

CONFERENCE PROOF