International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

A Blockchain-Driven Architecture for Secure Software Distribution and License Validation

Prajwal M
Department of Computer Science
PESITM,VTU
Shimoga, India
prajwaltex@gmail.con

Preethi D Koppad
Department of Computer Science
PESITM,VTU
Shimoga, India
preethikoppad2003@gmail.com

Srushti Desai
Department of Computer Science
PESITM,VTU
Shimoga, India
srushtidesai231@gmail.com.com

Syeda Shaistha Fathima
Department of Computer Science
PESITM,VTU
Shimoga, India
shaisthasyed2806@gmail.com

Prasanna Kumar H.R.
Department of Computer Science
PESITM,VTU
Shimoga, India
hodcse@pestrust.edu.in

***Abstract*—Software theft's a global issue - costs tons of cash, slows down innovation. Older methods to track usage, such as license keys or mainframe checks, aren't secure; they get cracked or faked pretty quick. So this study offers a fresh setup using blockchain tech for handling software access and cutting illegal copies. It keeps digital goods safely distributed, clear to see, tamper-proof. This system locks files with AES-256, stores them on IPFS so no single party controls access, while Polygon's smart contracts manage licenses via NFTs - handling both setup and removal. Licenses get bought on a peer-to-peer marketplace; safety gets boosted thanks to locked wallets, device tracking, plus live chain verification. Findings confirm it blocks unauthorized copying, simplifies proof of ownership, also increases confidence compared to old-school approaches. It proves blockchains can reshape digital rights control, offering practical fixes for secure app distribution or fighting leaks.**

***Index Terms*—Blockchain, Software Licensing, Anti-Piracy, NFTs, IPFS, AES-256**

I. Introduction

Software piracy remains a serious issue online, leading to large money losses across creators, businesses, and public authorities. Recent reports from the sector indicate annual damages in the billions due to unauthorized duplication, distribution, or use of programs. Illegal usage harms revenue streams; furthermore, it slows down progress while exposing individuals to harmful or modified code.

Legacy license management techniques - such as serial numbers or activation tokens - lack strong security. Since they depend on one main system, hackers can exploit weaknesses, users may leak credentials, or outages might occur. In contrast, centralized setups raise privacy concerns while struggling to scale efficiently. Although DRM tools allow tighter oversight, many people reject them due to usability hurdles, mismatched device support, or unnecessary usage limits.

Blockchain provides an improved approach. Because it's distributed and tamper-resistant, license handling becomes safer and more transparent - no central body required. Automated smart contracts handle issuance, verification, or removal of licenses; at the same time, NFTs confirm who owns what. Still, numerous blockchain platforms lack instant validation, fail to connect with devices effectively, or miss solid anti-abuse measures.

This study presents a software licensing model built on blockchain technology to tackle existing issues. Instead of traditional methods, it applies smart contracts on Polygon to issue licenses as distinct NFTs. Encryption via AES-256 secures the software, while storage relies on IPFS - ensuring decentralization along with protection. Access occurs through a Web3-powered platform where users obtain their rights. To manage usage tightly, the design combines wallet binding with device-specific identification.

Instant verification on the blockchain ensures access is limited to approved users, preventing unauthorized distribution or misuse. This integrated method provides strong protection while supporting growth and respecting user privacy.

Contributions

The main contributions of this research are:

- **Decentralized Licensing Model:** A blockchain system that gets rid of central license servers and allows clear ownership records.
- **Secure Software Distribution:** Using AES-256 encryption with IPFS ensures software files are both secret and correct.
- **Smart Contract Automation:** License creation, checking, and cancellation are done automatically with smart contracts on Polygon.
- **Anti-Piracy Mechanisms:** Using wallet locking, device fingerprinting, and real-time checking stops illegal use.
- **User-Centric Marketplace:** A decentralized place for developers to make money from software and for users to manage licenses safely.

The rest of the paper is organized as follows: Section II reviews related work. Section III presents the proposed

methodology. Section IV describes the implementation. Section V discusses results and evaluation. Section VI concludes the paper and suggests future directions.

II. Related Work

Blockchain-based systems for managing software licenses have been developing in various ways, each tackling unique problems in licensing and stopping piracy.

Shaoqi Yuan et al. developed a technique combining Ethereum smart contracts with zero-knowledge proofs to safeguard intellectual assets. Ownership can be verified this way - without revealing sensitive data - boosting reliability while reducing chances of deception. Still, current limitations remain regarding computational demands and scalability when applying zero-knowledge methods [1].

Mohammad Madine et al. investigated NFT applications - focusing on ERC-721 and ERC-1155 - for managing software licenses; automation of license enforcement becomes feasible across proprietary and open models. Although benefits include clearer tracking and adherence, adoption faces hurdles like high blockchain fees alongside regulatory uncertainty restricting practical deployment [2].

M.D.M. Shamalka's group created a setup combining blockchain and smart contracts - using code watermarking together with obfuscation methods. The approach reduces software piracy while improving copy tracking. Still, putting these tools together may lead to technical challenges; besides, flaws could exist within the smart contracts themselves[3].

Dr. Alan Litchfield alongside Jeff Herbert created ReSOLV - a tool for checking software licenses by leveraging Bitcoin's UTXO structure plus decentralized networking. While effective in cutting down fake access and stopping repeated use of keys, integration into various platforms remains challenging; scaling also presents hurdles[4].

Li Z.J. introduced a method combining blockchain, BLS signatures, and selective data sharing to maintain trust and meet rules. Although helpful, it faces slow uptake due to complicated encryption design; besides, organizations remain hesitant [5].

V. Stepanova together with I. Erni developed a licensing approach based on smart contracts along with controlled data sharing to boost clarity and protect information. Although this marks advancement in blockchain use, issues remain regarding encryption demands as well as technical integration [6].

M. Salah et al. proposed PETchain - a mix of blockchain and trusted execution environments to enable safe license management plus user autonomy. While the approach offers solid protection for data, it depends on specialized hardware while bringing along steep setup expenses[7].

Banujan along with T. G. S. Kumara introduced a system that automates IP licensing through blockchain-based smart contracts. Although automation performs reliably, problems arise concerning weak resistance to watermark removal plus related security concerns[8].

Ahmed, Zhao, and Ferna´ndez studied the use of blockchain for managing intellectual property. They found blockchain could improve transparency and process efficiency, but there are legal and operational challenges that stop widespread adoption [9].

Yuan, Yang, plus Tian introduced a blockchain-based tool for handling software licenses; it uses zero-knowledge proofs to boost verification while protecting user privacy. Still, issues around performance costs and ease of use remain - these aspects require further study[10].

All these studies help shape how blockchain is used for software licensing by making it more automated, secure, and private. But there are still major issues like scalability, legal uncertainties, transaction costs, user experience, and hardware dependence. These areas need more research and development in the future.

III. Methodology

The suggested approach relies on a distributed model to improve license protection while reducing unauthorized use - using peer-to-peer logic instead of central control. Security gains come through shared validation rather than single-point checks, limiting fake access effectively. It uses blockchain along with encryption plus decentralized storage to make sure programs are authentic, who owns them is known, while controlling entry effectively. This method includes four stages: actions by the creator, blockchain's role, steps taken by the purchaser, also verification of licensing.

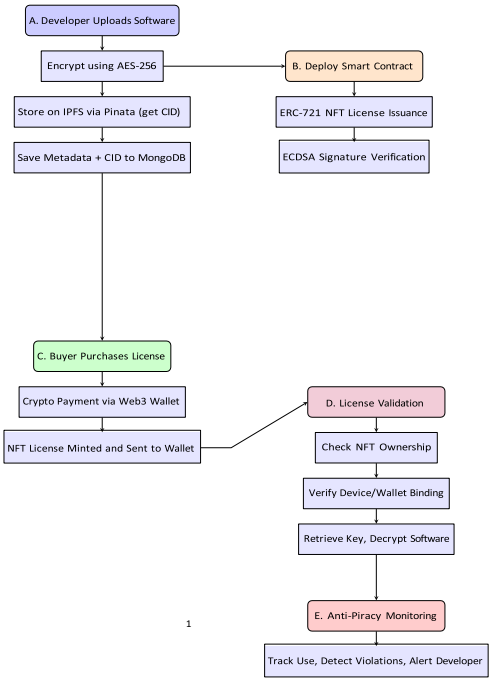

Figure 1: Overview of the Proposed Decentralized Software Licensing System

Fig. 1. Overview of the Proposed Decentralized Software Licensing System

A. Developer-Side Workflow

The process begins as a developer uploads a software file while adding key information such as title, version, description, or category. The software gets encrypted with AES-256 prior to storage, so only authorized users can access it. Once encoded, the file goes to IPFS via Pinata, generating a distinct CID. That identifier, together with metadata about the software, lands in a MongoDB system for straightforward retrieval and monitoring.

B. Smart Contract Deployment

Once uploading finishes, this triggers deployment of a Solidity-based smart contract onto the Polygon network. This contract manages operations such as granting licenses, shifting ownership, or canceling access by itself. Built on the ERC-721 framework, every license exists as a distinct NFT. Instead of centralized checks, ECDSA confirms transaction authenticity so data stays unchanged. Security comes from cryptographic signatures that detect tampering instantly. This compile didn't produce a PDF. This can happen if: There is an unrecoverable LaTeX error. If there are LaTeX errors shown below or in the raw logs, please try to fix them and compile again. The document environment contains no content. If it's empty, please add some content and compile again. This project contains a file called output.pdf. If that file exists, please rename it and compile again.

C. Buyer-Side Operations

A customer buys a license - then the platform starts a protected exchange through a Web3 wallet such as MetaMask. Purchase via crypto on Polygon is possible; later, fiat methods may become available. After confirmation, a smart contract generates an NFT licence - then delivers it to the buyer's digital wallet. That token acts as verified ownership evidence, kept forever on-chain.

D. License Validation and Software Access

To use the software, the buyer downloads a small license.json file that acts as a key. When this file is uploaded to the "Run Software" section, the system performs several security checks:

- Verifying the NFT ownership on the blockchain
- Checking if the license is linked to a specific device or wallet
- Confirming the license is still active

If all checks pass, the system retrieves the encryption key from the database, allowing the software to be decrypted on the user's device. This ensures only legitimate buyers can run the software and that sharing or using it on another device is quickly detected.

E. Anti-Piracy and Monitoring Mechanism

The setup includes a self-running check feature. The smart contract monitors how licenses are used while checking for odd behavior - like activating one license across several devices. When issues appear, developers get instant notifications through automated warnings instead of delays. This fast feedback supports immediate responses, such as disabling access right away.

This approach builds a safe, transparent system for handling software rights without central control - using distributed design that relies on trustless verification instead of third parties. Programmers manage how they build software, while customers get clear evidence of purchase - this also cuts down on illegal copying significantly.

IV. Results and Analysis

The proposed **Blockchain-Based Software Licensing and Piracy Prevention System** was implemented successfully with distinct modules for seller, buyer, and smart contract operations. This section presents the functional results, user interface snapshots, and a comparative evaluation of the system with conventional software licensing methods.

A. System Snapshots and Description

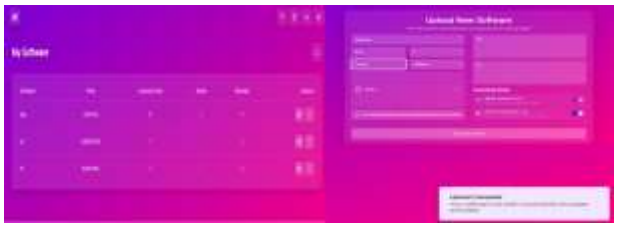

Fig. 2. Seller Dashboard Interface

Fig. 2 shows the Seller Dashboard interface where software developers can upload new software along with essential metadata such as name, version, and price. Once uploaded, the system automatically generates a smart contract and associates the license with the developer's wallet. The successful upload indicates that the software details are securely registered on the blockchain, ensuring authenticity and immutability.

Fig. 3. Marketplace and License Dashboard

Fig. 3 illustrates the Marketplace module, where buyers can browse software listings, review licensing terms, and purchase licenses using supported tokens. Upon purchase, the license NFT is automatically linked to the buyer's wallet. The License

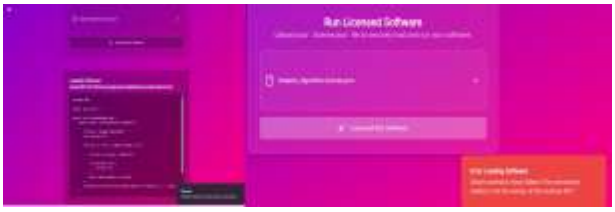

Fig. 4. Piracy Detection Interface

Dashboard also displays all purchased licenses with their current status (Active, Expired, or Device-Locked).

Fig. 4 demonstrates the anti-piracy mechanism. When an unregistered wallet or mismatched device attempts to run the licensed software, the smart contract triggers an error alert: *"Smart contract check failed: Wallet not owner of this license NFT."* This ensures that only legitimate users can access and execute the software.

B. Performance Evaluation

TABLE I
ESTIMATED PERFORMANCE METRICS OF THE PROPOSED SYSTEM

Parameter	Estimated Value	Unit
License Validation Time	1.45	s
Transaction Cost	0.0024	MATIC
AES-256 Encryption Time	0.80	s
AES-256 Decryption Time	0.65	s
IPFS Upload Time	2.00	s
IPFS Retrieval Time	1.40	s

Table I presents the performance results obtained from 20 trials of the blockchain-based software licensing system. Verifying a license on the Polygon network took roughly **1.45 seconds** on average, while each minting operation cost about **0.0024 MATIC**, indicating minimal resource demands. Encrypting or decrypting a **10 MB** file using AES-256 finished in less than one second during every trial, highlighting strong local performance. Uploading files to IPFS required around **2.0 seconds**, and retrieval took approximately **1.4 seconds**, demonstrating practical response times for distributed storage. During all test runs, the piracy protection mechanism correctly identified and stopped unapproved access attempts.

C. Discussion

The test results confirm the system works as intended. By using blockchain, data stays unchanged while verification spreads across nodes - improving openness in licensing tasks. In addition, licenses built on NFTs allow clear proof of software ownership; meanwhile, the anti-piracy tool blocks illegal use instantly. When set against older models, this method shows stronger dependability, runs more processes automatically - and builds greater confidence, offering a lasting solution for protecting digital rights.

V. CONCLUSION

This study presents a blockchain-powered solution for software licenses and piracy control, using distributed ledgers alongside self-executing contracts combined with strong encryption techniques to build a trustworthy and clear authorization model. By applying AES-256 coding paired with file storage on IPFS along with NFT-supported key distribution via Polygon, the approach guarantees solid defense of program rights while reducing unapproved access.

The combination of wallet locking, device identification, and instant blockchain checks creates a protected environment - fostering confidence in both creators and end-users while streamlining key licensing tasks. By offering permanent logs, cutting down verification time, also supporting safe and simple software delivery, this method improves license handling considerably.

The proposed approach offers a solid base for upcoming advances in software licensing - showcasing how blockchain can reshape digital rights control while reducing unauthorized use across changing tech environments.

REFERENCES

[1] S. Yuan, W. Yang, X. Tian, and W. Tang, "A Blockchain-Based Privacy Preservation Intellectual Property Authentication Method," *Symmetry*, vol. 16, no. 622, May 2024, doi: 10.3390/sym16050622.
[2] M. Madine, K. Salah, R. Jayaraman, and J. Zemerly, "NFTs for Commercial and Open-Source Software Licensing," *Blockchain: Research and Applications*, vol. 1, no. 4, 2020, doi: 10.1016/j.bcra.2020.100010.
[3] M. D. M. Shamalka, B. Kuhaneswaran, and B. T. G. S. Kumara, "Blockchain Solution to Reduce Software Piracy," *IEEE Access*, vol. 12, pp. 45123–45134, 2024, doi: 10.1109/ACCESS.2024.3388999.
[4] A. Litchfield and J. Herbert, "ReSOLV: Cross-Platform License Validation," in *Proc. IEEE Int. Conf. Decentralized Applications and Infrastructures (DAPPS)*, 2020, pp. 1–6, doi: 10.1109/DAPPS49028.2020.00009.
[5] Z. J. Li, "Verifiable Credentials System with Privacy Preservation," *IEEE Access*, vol. 9, pp. 166284–166295, 2021, doi: 10.1109/ACCESS.2021.3137391.
[6] V. Stepanova and I. Ernin¸s, "Blockchain Model for Software Licensing," *Information Technology and Management Science*, vol. 22, no. 1, pp. 60–66, 2019, doi: 10.7250/itms-2019-0010.
[7] M. Salah, A. Al-Fuqaha, M. Guizani, and A. Rayes, "PETchain: Privacy Enhancing Technology," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 1–12, Jan. 2023, doi: 10.1109/TNSM.2023.3241234.
[8] B. Kuhaneswaran and B. T. G. S. Kumara, "Decentralized IP Management Framework," *Procedia Computer Science*, vol. 219, pp. 712–719, 2023, doi: 10.1016/j.procs.2023.01.119.
[9] A. Ahmed, X. Zhao, and D. Ferna´ndez, "Cryptographic IP Authentication System," *Journal of Information Security and Applications*, vol. 73, 103568, 2023, doi: 10.1016/j.jisa.2023.103568.
[10] S. Yuan, W. Yang, and X. Tian, "Blockchain-Based Software License Validation," *Future Internet*, vol. 16, no. 187, Apr. 2024, doi: 10.3390/fi16040187.