Registration as MTech Student for

Master’s Degree

Student Details

Name: Ms. Pawar Ankita Sharad

Address: Karajgav, Lohara, Dist.: Osmanabad-413604

Mobile No.: 9665191315

Email ID: aspawar009@gmail.com

Category: Open

Subject: Department of Computer Science and Engineering

Guide Details

Prof. Dr. Sanjay N. Talbar

Professor

Department of Electronics and Telecommunication Engineering,

Shri Guru Gobind Singhji (SGGS) Institute of Engineering & Technology

Vishnupuri, Nanded-431 606 (MS)

Email(s): sntalbar@sggs.ac.in; sntalbar@yahoo.com

Phone : 2462269209 Mobile: 9850978050 Fax: 2462229236

Synopsis for Registration as Research Scholar

for

Master’s Degree

on

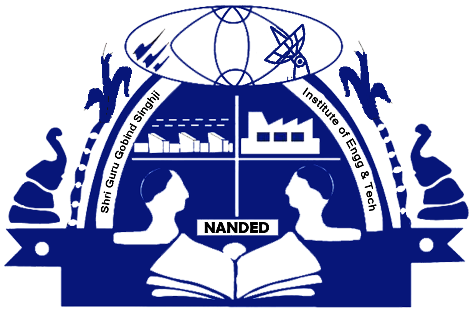
Blockchain-Based Secure Software Licensing Platform

By

Ms. Pawar Ankita Sharad

Supervisor

Prof. Dr. Sanjay N. Talbar



Department of Computer Science and Engineering

Shri Guru Gobind Singhji Institute of Engineering and Technology,

Nanded-431606. (M.S.) India.

**Abstract of proposed work plan/ problem:**

The Blockchain-Based Secure Software Licensing Platform addresses inefficiencies and vulnerabilities in traditional software licensing systems, such as piracy, unauthorized use, and administrative complexity. By leveraging blockchain technology, specifically smart contracts, the project aims to create a transparent, decentralized, and secure licensing system. The platform will automate license issuance, management, transfer, and revocation while ensuring that all transactions are immutable and auditable. Smart contracts embedded with license terms will eliminate the need for intermediaries, reducing the risk of fraud and enhancing user trust in the licensing process.

1. **Introduction:**

The digital landscape is evolving rapidly, with software increasingly being delivered via decentralized systems. However, traditional software licensing models are prone to several challenges, including piracy, fraudulent license creation, and inefficient license management. Software piracy leads to significant revenue loss for developers and undermines intellectual property rights. Centralized licensing solutions, while somewhat effective, often introduce overhead costs, lack transparency, and are susceptible to tampering or security breaches.

Blockchain technology, with its decentralized, immutable ledger and smart contracts, offers a revolutionary way to solve these issues. Blockchain-based platforms can ensure that once software licenses are issued, they cannot be altered, tampered with, or misused. Furthermore, smart contracts enable the automatic enforcement of licensing agreements, eliminating the need for intermediaries and ensuring compliance with licensing terms.

This project proposes the development of a Blockchain-Based Secure Software Licensing Platform that automates the creation, distribution, management, and verification of software licenses using blockchain and smart contracts. The platform will offer transparency, security, and efficiency, addressing the major shortcomings of traditional licensing systems.

1. **Review of Literature and Development in the Subject (Previous Work Done in the Relevant Area):**

Several studies and projects have explored the potential of blockchain technology in digital rights management (DRM) and software licensing:

* Traditional Software Licensing Models: Previous studies have highlighted how centralized software licensing systems are prone to piracy, unauthorized use, and complex management, leading to revenue losses for developers and inefficiencies for end-users. Solutions such as product keys and DRM systems have been implemented but often fail to prevent license fraud comprehensively.
* Blockchain and Smart Contracts: The introduction of Ethereum in 2015 brought smart contracts into focus, enabling decentralized applications (dApps) to automate and enforce digital agreements. Smart contracts have been used to manage digital assets, which has inspired their application in digital rights management, including software licensing.
* Case Studies: Several blockchain projects have explored licensing for creative content and intellectual property:
  + Po.et: A blockchain-based digital rights management platform for creative content, focused on timestamping and verifying ownership.
  + Ujo Music: A blockchain-based licensing system for music that allowed artists to issue licenses directly via smart contracts.
  + Microsoft Blockchain Licensing System: Microsoft explored blockchain-based licensing for its cloud software to improve transparency and reduce fraudulent software usage.
  + NFTs in Licensing: The rise of non-fungible tokens (NFTs) has demonstrated the use of blockchain for representing unique digital assets, which has implications for software license issuance and management.
  + Blockchain-Based License Validation for IoT Devices: Recent studies have explored how blockchain can manage software licenses for IoT devices, showcasing the applicability of the technology in decentralized environments.
* Smart Contracts for Licensing: Research has emphasized the benefits of smart contracts for automating digital agreements, ensuring that once a license is purchased, the terms of use, renewal, or revocation are automatically enforced without manual intervention.

These studies underscore blockchain's potential in reshaping software licensing, offering immutability, decentralized control, and transparent auditability.

1. **Objectives of Research / Proposed Hypothesis:**

The primary objective of the Blockchain-Based Secure Software Licensing Platform is to design and develop a decentralized, secure, and transparent platform for managing software licenses using blockchain technology. Key objectives include:

1. Automated License Issuance: Develop a smart contract-based system that allows software developers to create and issue licenses automatically upon purchase.
2. Decentralized License Management: Enable users and developers to manage software licenses (e.g., transfers, revocations, renewals) without reliance on a centralized authority.
3. Secure License Storage: Leverage blockchain’s immutable ledger to store licensing data securely, preventing unauthorized access or modification.
4. Real-Time License Verification: Provide APIs for software applications to verify the validity of a license in real time, ensuring compliance with licensing terms before software usage.
5. Transparent Auditing: Ensure that all licensing transactions are recorded immutably, enabling transparent auditing and compliance checks while maintaining user privacy.
6. Enhance Trust: Build a system that fosters trust between developers and users by ensuring that all license agreements are transparent, immutable, and enforceable.

The hypothesis is that blockchain technology, when applied to software licensing, can significantly reduce piracy, streamline license management, and provide a secure and transparent method for enforcing licensing terms.

1. **Technologies & Tools:**
2. Blockchain Platform:

Ethereum: Ethereum is a decentralized blockchain platform with support for smart contracts. The Ethereum Virtual Machine (EVM) allows smart contracts to be written in Solidity and deployed on the network.

Hyperledger Fabric: For enterprise solutions, Hyperledger Fabric can provide a permissioned blockchain where companies can manage software licenses more privately.

1. Smart Contracts:

Solidity: Use Solidity to write the smart contracts for creating, issuing, revoking, and managing licenses on Ethereum.

Chain code : For Hyperledger Fabric, smart contracts are written in Go or JavaScript.

1. License Storage:

IPFS (Interplanetary File System): For large files or metadata associated with licenses, IPFS can be used to store the data off-chain, while the hash of the data is stored on-chain to maintain integrity.

Blockchain Ledger: Use the blockchain itself to store key licensing information and smart contract states (license validity, purchase details, etc.).

1. Web Front-End:

React.js or Vue.js: Build a user-friendly web interface for developers and customers to interact with the platform (purchase licenses, manage licenses, etc.).

MetaMask or Web3.js: Integrate with blockchain wallets like MetaMask for Ethereum or use Web3.js for interacting with the blockchain from the browser.

1. Back-End Services:

Node.js or Express.js: The back-end service that interacts with the blockchain, handles the API for license verification, and manages off-chain processes like user registration.

IPFS API: For storing large licensing files or metadata off-chain.

1. Database (Optional):

MongoDB: While blockchain provides decentralized storage, off-chain metadata (e.g., user profiles, software details) can be stored in a NoSQL database like MongoDB for fast access.

1. APIs for License Verification:

GraphQL or REST API: Provide a robust API that third-party software can call to check the validity of licenses.

The API interacts with the blockchain to verify license status and return results to the requesting software.

1. Encryption & Security:

AES or RSA: Encryption standards for securing communication and transactions between users, developers, and the platform.

JWT (JSON Web Tokens): For secure API communication and user authentication.

1. Cloud Infrastructure:

AWS or Azure: Use cloud services to deploy the platform, ensuring high availability and scaling capabilities.

Kubernetes or Docker: Use containerization for microservice architecture, ensuring modularity and easier updates.

1. Payment Gateway:

Cryptocurrency Payments: Support for cryptocurrency payments (e.g., ETH, BTC) for purchasing licenses.

Fiat Payment Gateways: Integrate with traditional payment gateways like Stripe or PayPal for users who prefer fiat currencies.

1. **Methodology:**

The methodology for the development of the Blockchain-Based Secure Software Licensing Platform can be broken down into several phases:

Phase 1: Research and Feasibility Study

* Conduct a detailed analysis of existing blockchain platforms (e.g., Ethereum, Hyperledger Fabric) to determine the most suitable blockchain for the project.
* Study relevant licensing models and smart contract designs to identify key challenges and solutions for license management on the blockchain.

Phase 2: Platform Design

* Blockchain Selection: Choose a suitable blockchain platform (Ethereum for public solutions or Hyperledger Fabric for enterprise/private solutions) based on the findings from the research phase.
* Smart Contract Development: Design and implement smart contracts using Solidity (for Ethereum) or Chaincode (for Hyperledger Fabric) for key functionalities:
  + License creation and issuance
  + License transfer, renewal, and revocation
  + Automated enforcement of licensing terms
* License Storage: Design a decentralized storage mechanism using IPFS (InterPlanetary File System) for large files or metadata, ensuring that the blockchain only stores hashes for verification.

Phase 3: Web and API Development

* Front-End Development: Build a web-based dashboard using React.js or Vue.js that allows developers to create, issue, and manage licenses, and users to purchase, transfer, and verify licenses.
* Back-End Development: Use Node.js or Express.js to handle interactions with the blockchain and provide the API for license verification.
* License Verification API: Develop a REST or GraphQL API that software developers can integrate into their products to check license validity in real-time.

Phase 4: Security and Encryption

* Implement security features such as AES/RSA encryption for data communication, secure key management, and multi-factor authentication for developers and users.

Phase 5: Testing and Deployment

* Perform extensive testing of the smart contracts on testnet environments (e.g., Ethereum’s Ropsten or Goerli testnets).
* Deploy the platform on the chosen blockchain network and conduct scalability tests.
* Implement cloud services (e.g., AWS or Azure) for hosting off-chain processes, ensuring high availability and scalability.

Phase 6: Integration of Payment Systems

* Integrate cryptocurrency payment systems (e.g., ETH, BTC) and fiat payment gateways (e.g., PayPal, Stripe) for purchasing licenses.

5. References:

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf.
2. Buterin, V. (2013). Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform. Retrieved from https://ethereum.org/en/whitepaper/.
3. Chen, X., Zhang, W., & Liang, Y. (2020). Smart Contracts for Digital Licensing: A Blockchain Approach. In *Journal of Computer Science*.
4. Lu, Q., & Xu, X. (2019). Blockchain for Digital Rights Management: Research Directions and Future Trends. In *Journal of Internet Services and Applications*.
5. Microsoft (2018). Blockchain for Software License Management. Retrieved from <https://www.microsoft.com/en-us/research/project/blockchain-based-digital-licensing/>.
6. Po.et. (2017). Blockchain-Based Digital Rights Management for Creative Works. Retrieved from <https://po.et/>.
7. Ujo Music. (2016). Decentralized Music Licensing Platform. Retrieved from <https://ujomusic.com/>.
8. O’Leary, P., & Smith, J. (2021). Blockchain-Based License Validation for IoT Devices. In *IEEE Transactions on Blockchain*.
9. IPFS (2022). The InterPlanetary File System: Decentralized Storage for the Future. Retrieved from <https://ipfs.io/>.

Ms. Pawar Ankita Sharad Dr. Sanjay. N. Talbar

First Year Student Supervisor

Mtech, S.G.G.S.I.E. & T,

SVERI Nanded.(MS)