iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61466

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Decentralized Chatting Application Using Blockchain Technology

Jim Mathew Philip [1], Rajeswari R[2]
[1]*Associate Professor,* [2]*PG Student, Department of Computer Science and Engineering, Sri Ramakrishna Institute of Technology,*

Abstract: ***The emergence of blockchain technology has spurred innovation in various domains, including decentralized communication systems. In this context, a decentralized chat application leveraging blockchain, Remix Ethereum IDE, Ganache blockchain, and MetaMask offers a novel approach to secure and privacy-centric messaging. The application harnesses blockchain's immutable ledger to store chat messages, ensuring data integrity and censorship resistance. Remix Ethereum IDE provides developers with a robust environment for crafting smart contracts, the backbone of the chat application's functionality. Developers can design and deploy smart contracts defining chatroom logic, message encryption, and user authentication. Ganache blockchain serves as a local testing environment, enabling developers to validate smart contracts and simulate blockchain interactions in a controlled setting. This facilitates rapid iteration and debugging during the development phase. MetaMask acts as the gateway for users to access the decentralized chat application. Through MetaMask, users authenticate securely using their Ethereum wallets, ensuring pseudonymous identity verification and enabling seamless interaction with the Ethereum blockchain. Together, these technologies form the foundation of a decentralized chat application that prioritizes privacy, security, and user control. By leveraging blockchain and associated tools, the application offers a compelling alternative to centralized messaging platforms, empowering users with sovereignty over their data and communication.***
Keywords: Blockchain, Remix Ethereum IDE, Ganache blockchain, and Metamask

I. INTRODUCTION

A blockchain is “a distributed database that maintains a continuously growing list of ordered records, called blocks.” These blocks “are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. A blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the consensus of the network. A secure and open method of managing data and recording transactions is offered by blockchain technology, which functions as a decentralized, impenetrable digital ledger.

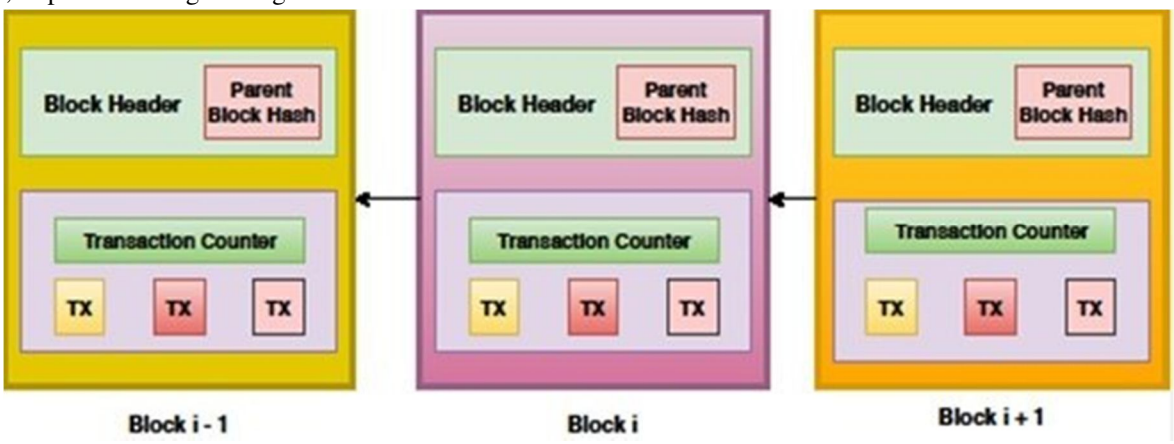


Fig. 1 Architecture of Blockchain Technology

A chain of blocks, each holding a list of transactions, is the fundamental building block of a blockchain. To guarantee the integrity of the data, these blocks are cryptographically connected. Blocks are extremely resistant to fraud and tampering because once they are added to the chain, they become immutable, meaning that changing one block will change all subsequent blocks.

Blockchain is the fundamental technology that underpins the Decentralized Chat Application (DCA), providing private and secure communication. User messages are kept in blocks on a peer-to-peer network and encrypted using cutting-edge cryptographic algorithms. Every communication joins the blockchain, adding security, time stamped integrity, and immutability. For DCA users, privacy and security are increased because of the decentralized structure of blockchain, which guarantees that no single entity controls the data.

The four main components of any blockchain ecosystem are as follows:

1) Node Application:

Each Internet-connected computer needs to install and run a computer application specific to the ecosystem they wish to participate in. Using the case of Bitcoin as an example ecosystem, each computer must be running the Bitcoin wallet application.

2) Shared Ledger

This is a logical component. The distributed ledger is a data structure managed inside the node application. Once you have the node application running, you can view the respective ledger (or blockchain) contents for that ecosystem.

3) Consensus Algorithm

This, too, is a logical component of the ecosystem. The consensus algorithm is implemented as part of the node application, providing the _rules of the game 'for how the ecosystem will arrive at a single view of the ledger.

4) Virtual Machine

The virtual machine is the final logical component implemented as part of the node application that every participant in the ecosystem runs.

The function of blockchain in a decentralized chat app:

Blockchain technology can play a significant role in decentralized chat applications by providing the underlying infrastructure for secure, private, and censorship-resistant communication. Here's how:

a) *Decentralization:* Blockchain allows for the creation of decentralized chat platforms where messages are not stored on a central server but distributed across a network of nodes. This decentralization makes it difficult for any single entity to control or censor communication, ensuring freedom of speech and expression.

b) *Security and Privacy:* By leveraging cryptographic techniques, blockchain-based chat applications can encrypt messages end-to-end, ensuring that only the intended recipients can decrypt and read them. Additionally, blockchain provides a tamper-resistant and immutable record of messages, enhancing security and transparency.

c) *User Control:* In traditional chat applications, user data is often owned and controlled by the platform provider. With blockchain-based chat applications, users have greater control over their data. They can choose how their messages are stored, who has access to them, and whether they want to monetize their data.

d) *Identity Verification:* Blockchain technology can enable secure and decentralized identity verification, allowing users to verify the identity of other participants without relying on a central authority. This feature enhances trust and reduces the risk of impersonation or fraud in chat applications.

e) *Micropayments and Incentives:* Some blockchain-based chat platforms utilize cryptocurrency tokens to incentivize user participation and contribution to the network. Users may earn tokens for activities such as participating in discussions, moderating content, or providing valuable insights. These micropayments can create economic incentives for users to engage with the platform and contribute positively to the community.

f) *Interoperability:* Blockchain technology enables interoperability between different chat platforms and networks, allowing users to communicate seamlessly across various decentralized applications (dApps). This interoperability breaks down silos and fosters a more interconnected and inclusive communication ecosystem.

g) *Immutable Record:* Every message sent on a blockchain-based chat application is recorded on the blockchain, creating an immutable and time stamped history of communication. This feature can be useful for auditing purposes, dispute resolution, or simply for preserving the integrity of conversations over time.

II. BACKGROUND AND RELATED WORK

Over the last ten years, the internet has been dominated by popular messaging apps like WeChat and WhatsApp. These platforms are based on a centralized server structure that stores all user data, including chat history and identities. But there are drawbacks to this centralized strategy as well:

Centralized chat applications often collect and store user data, raising concerns about privacy and data security. Users may be uncomfortable with the platform provider having access to their personal conversations, leading to potential privacy breaches or data misuse. Centralized chat applications rely on a central server or infrastructure, making them vulnerable to single points of failure. If the server experiences downtime or a cyberattack, it can disrupt communication for all users on the platform.

Centralized platforms have the authority to censor content or restrict access to certain users or groups. This centralized control can lead to censorship of dissenting opinions or politically sensitive discussions, limiting freedom of expression. Centralized chat applications are susceptible to security risks such as hacking, phishing, and malware attacks. A breach of the central server can expose sensitive user data, including messages, contact lists, and login credentials. Users of centralized chat applications often relinquish ownership of their data to the platform provider, who may monetize it for targeted advertising or other purposes without user consent. This lack of control over personal data raises ethical concerns and diminishes user autonomy.

As centralized chat applications grow in popularity, they may encounter scalability issues related to server capacity and bandwidth limitations. High volumes of messages or concurrent users can strain the infrastructure, leading to performance degradation or service outages. Users of centralized chat applications are typically locked into the ecosystem and reliant on the platform provider for continued service. Switching to alternative platforms can be challenging due to compatibility issues and the need to migrate data and contacts. Centralized chat applications must comply with regulatory requirements related to data protection, content moderation, and user privacy. Meeting these compliance standards can be resource-intensive and may vary across different jurisdictions, posing legal and operational challenges for the platform provider.

III. IMPLEMENTATION

Our decentralized chat application is built using Next.js, a powerful React framework that makes building server-rendered react applications easier, and the Ethereum blockchain network.

A. *Remix IDE:*

Remix IDE is a powerful tool for developing, debugging, and deploying smart contracts on the Ethereum blockchain. However, there's always room for improvement and customization based on user needs. Here are some ideas for remixing Remix IDE:

1) *Customizable Interface:* Allow users to customize the layout, color scheme, and placement of different panels and tabs according to their preferences. This can enhance user experience and productivity.
2) *Integration with External Tools:* Integrate Remix IDE with popular external tools and services such as Git for version control, Docker for containerization, and continuous integration (CI) pipelines for automated testing and deployment.
3) *Improved Collaboration Features:* Add real-time collaboration features such as shared editing, commenting, and code review capabilities to facilitate teamwork among developers working on the same project.
4) *Enhanced Debugging Tools:* Introduce advanced debugging tools like breakpoints, watchlists, and step-by-step execution for easier identification and resolution of smart contract bugs and errors.
5) *Smart Contract Templates:* Provide a library of smart contract templates for common use cases such as token creation, decentralized finance (DeFi) protocols, non-fungible tokens (NFTs), and more. This can accelerate development by allowing developers to start with pre-built templates and customize as needed.
6) *Integration with Oracles and External Data Sources:* Enable seamless integration with oracles and external data sources to fetch real-world data into smart contracts, expanding the capabilities of decentralized applications (DApps).
7) *Enhanced Security Analysis:* Incorporate static analysis tools to identify potential security vulnerabilities in smart contracts, such as reentrancy bugs, integer overflow/underflow, and other common pitfalls.
8) *Multi-Chain Support:* Extend support beyond Ethereum to other blockchain platforms such as Binance Smart Chain, Polkadot, and Solana, allowing developers to build and deploy cross-chain applications from within Remix IDE.
9) *Improved Documentation and Tutorials:* Provide comprehensive documentation, tutorials, and examples within the IDE to assist developers in learning smart contract development and best practices effectively.

10) *Community Plugins and Extensions:* Allow developers to create and share plugins and extensions to extend the functionality of Remix IDE, fostering a vibrant ecosystem of tools and utilities tailored to specific use cases and preferences.

.

B. Ganache blockchain:

Ganache is a popular personal blockchain for Ethereum development, often used for testing, debugging, and deploying smart contracts locally. Here are some common operations you can perform with Ganache:

Ganache simulates an Ethereum blockchain locally on your machine. It allows you to interact with the blockchain without connecting to the main Ethereum network. This is useful for development and testing purposes, as it provides a controlled environment. Ganache provides a quick and easy setup process. You can download and install it as a desktop application or run it as a command-line tool. Once installed, you can start a local blockchain instance with just a few clicks or commands. Ganache allows you to configure various parameters of the local blockchain node, such as the number of accounts, gas limit, block time, network ID, and more. This flexibility enables you to tailor the environment to your specific needs.

Ganache generates a set of Ethereum accounts with pre-funded Ether (fake ETH) for testing purposes. You can view the details of these accounts, including their addresses, private keys, and balances. Additionally, you can import external accounts or generate new ones as needed. You can send transactions to interact with smart contracts deployed on the local blockchain. Ganache provides instant mining, so transactions are confirmed immediately, speeding up the development and testing process. Ganache allows you to deploy smart contracts onto the local blockchain. You can compile your Solidity contracts using tools like Remix IDE or Truffle, then deploy them to Ganache using web3.js or Truffle migrations.

Ganache includes built-in debugging tools that help you diagnose and troubleshoot issues in your smart contracts. You can inspect transaction logs, review contract state changes, and track gas consumption to identify potential bugs or optimizations. While Ganache is primarily used for local development, it can also be integrated with external tools and services. For example, you can configure Metamask to connect to Ganache, allowing you to interact with your local blockchain using a browser extension. Ganache supports blockchain forking, allowing you to create a copy of the main Ethereum network or another network at a specific block height. This feature is useful for testing contract interactions under different network conditions or for simulating specific scenarios. Ganache integrates seamlessly with popular Ethereum testing frameworks like Truffle and Hardhat. You can write automated tests for your smart contracts and run them against your local blockchain instance to ensure their correctness and reliability.

C. MetaMask

MetaMask is a widely used browser extension and mobile app that serves as a gateway for interacting with the Ethereum blockchain ecosystem. Here are some key functionalities and operations you can perform with MetaMask:

1) *Wallet Management:* MetaMask allows users to create and manage Ethereum wallets directly from their web browser or mobile device. Each wallet is associated with a unique address and a corresponding set of public and private keys.
2) *Transaction Management:* Users can send and receive Ether (ETH) and ERC-20 tokens using MetaMask. They can initiate transactions by specifying the recipient's address, the amount to send, and optional parameters such as gas price and gas limit.
3) *Token Management:* MetaMask provides a built-in token management interface, allowing users to view, send, and receive ERC-20 and ERC-721 tokens. Users can also add custom tokens by providing the token contract address.
4) *DApp Browser:* MetaMask includes a built-in DApp browser that enables users to interact with decentralized applications (DApps) directly from their browser or mobile device. Users can access various Ethereum-based applications, including decentralized exchanges (DEXs), blockchain games, decentralized finance (DeFi) platforms, and more.
5) *Signature and Authentication:* MetaMask allows users to sign messages and authenticate themselves on Ethereum-based websites and applications. This functionality is commonly used for user authentication, identity verification, and access control.
6) *Network Switching:* MetaMask supports multiple Ethereum networks, including the Ethereum mainnet, testnets (Ropsten, Rinkeby, Kovan), and custom networks. Users can switch between different networks to interact with different environments and test their applications.
7) *Security Features:* MetaMask prioritizes security and provides features such as password protection, seed phrase backup, and hardware wallet integration (e.g., Ledger and Trezor). Users are encouraged to store their private keys securely and enable additional security measures to protect their funds.

8) *Customization Options:* MetaMask offers various customization options, allowing users to adjust settings such as gas fees, language preferences, and currency display. Users can also customize the appearance of the MetaMask extension through themes and color schemes.

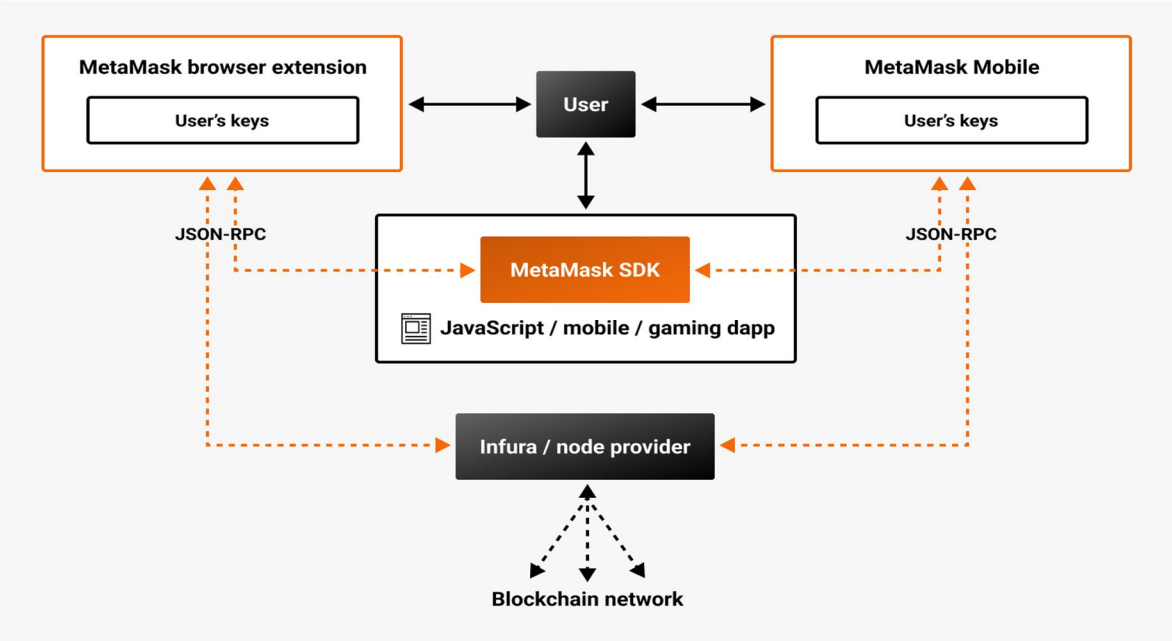


Fig. 2 MetaMask interact with the blockchain ecosystem

D. Etherscan

Etherscan is one of the most popular and widely used block explorers for the Ethereum blockchain. It provides users with a range of tools and features to explore, analyze, and interact with the Ethereum blockchain. Some key features of Etherscan include:

1) *Blockchain Explorer:* Etherscan allows users to search and explore the Ethereum blockchain by entering transaction hashes, addresses, or block numbers. Users can view detailed information about individual transactions, blocks, and smart contracts.
2) Transaction Tracking: Users can track the status of Ethereum transactions in real-time using Etherscan. This includes monitoring pending transactions, verifying transaction confirmations, and checking transaction details such as gas fees and timestamps.
3) *Address Monitoring:* Etherscan enables users to monitor Ethereum addresses and view their transaction history, token balances, and other relevant information. This feature is useful for tracking wallet activity and managing Ethereum assets.
4) *Smart Contract Analysis:* Etherscan provides tools for analyzing and interacting with smart contracts deployed on the Ethereum blockchain. Users can view contract source code, bytecode, and ABI (Application Binary Interface), as well as interact with contract functions and execute transactions.
5) *Token Information:* Etherscan offers comprehensive token information, including token balances, transaction history, and token contract details. Users can search for specific tokens and view their market performance, supply statistics, and token holders.
6) *Network Statistics:* Etherscan provides real-time network statistics for the Ethereum blockchain, including block propagation times, transaction throughput, and gas usage. Users can monitor network health and performance metrics to gauge overall blockchain activity.
7) *API Access:* Etherscan offers APIs (Application Programming Interfaces) for developers to access Ethereum blockchain data programmatically. Developers can integrate Etherscan APIs into their applications to retrieve blockchain data, verify transactions, and perform other tasks.

IV. RESULT AND DISUSSION

A. User Open App

The decentralized chat app initializes its user interface and connects to the underlying blockchain network or decentralized protocol it relies on. This may involve loading the chat interface, fetching user preferences, and establishing connections to the decentralized network.

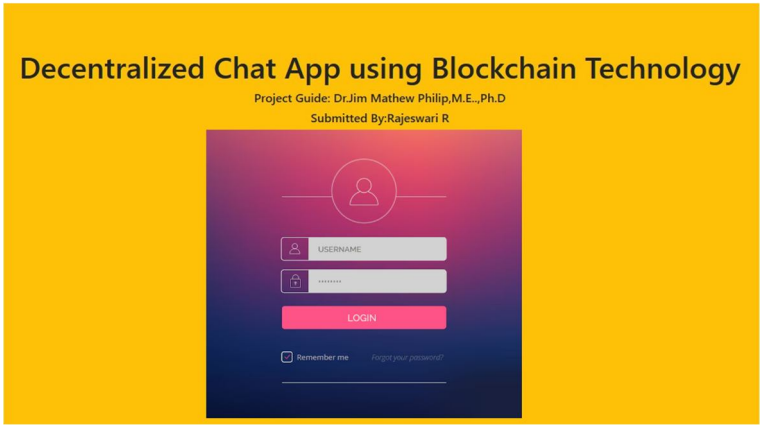


Fig. 3 DCA Login Page

B. Scan QR to Login To Our App

The decentralized chat app generates a unique QR code that encodes the necessary authentication information. This QR code may contain encrypted data, session tokens, or other cryptographic keys required for user authentication and session establishment. When the user opens the app, they are presented with the option to log in or authenticate

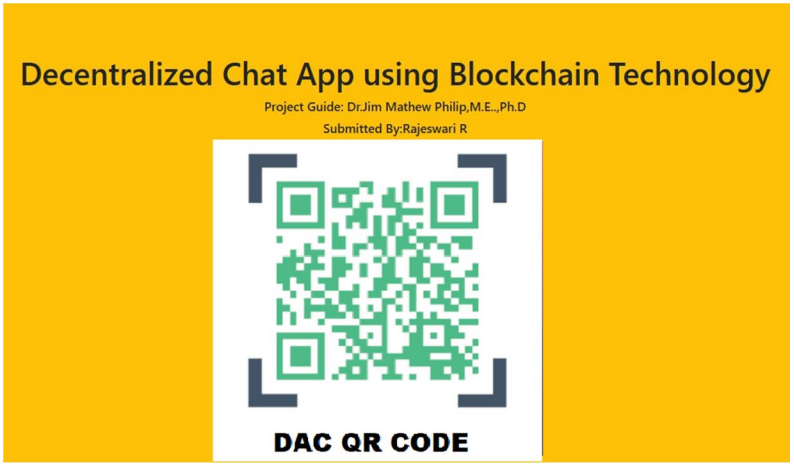


Fig. 4 DCA QR Code Page

C. Sender start to chat with this application

The sender composes their message within the chat interface. This could be text, emojis, multimedia content, or other supported message formats. Once the message is composed, the sender sends it through the application's interface. This action may involve clicking a send button or pressing enter on their keyboard.

Fig. 4 DCA Sender Page

D. Receiver receive the chat

Once authenticated, the receiver navigates to the chat interface within the application where they can view their messages. This could be a list of chatrooms, direct message threads, or another section dedicated to incoming messages. The receiver views the new message in the chat interface. The message is displayed along with any relevant metadata, such as the sender's name or avatar, the timestamp of the message, and any attachments or multimedia content included in the message.

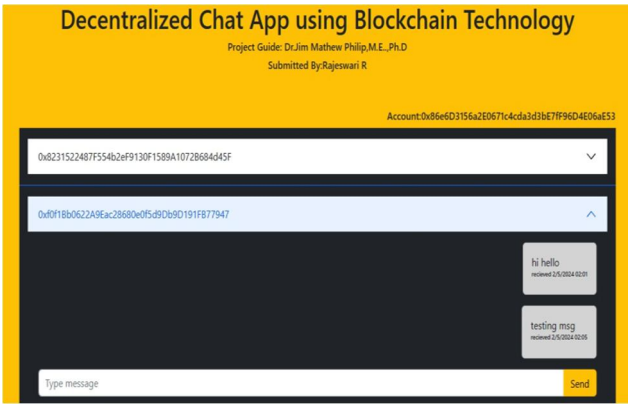


Fig. 4 DCA Receiver Page

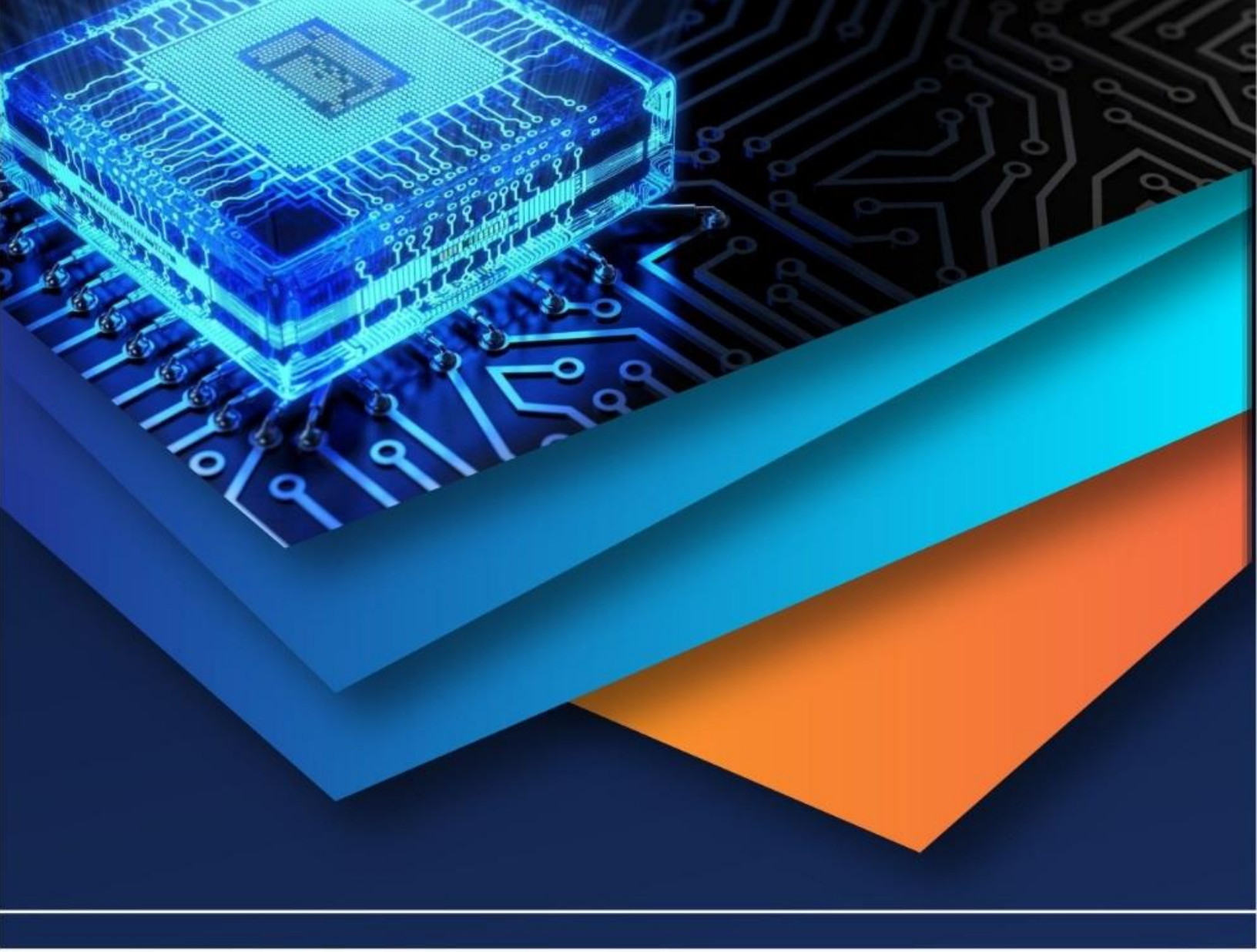
V. CONCLUSIONS

Decentralized chat applications represent a paradigm shift in how we communicate online, offering a range of benefits and opportunities. By leveraging blockchain technology and decentralized architecture, these platforms prioritize user privacy, security, and freedom of expression while mitigating the risks associated with centralized alternatives. Decentralized chat applications address concerns such as censorship, data privacy, and single points of failure by distributing control and responsibility across a network of nodes. This decentralization fosters a more resilient, transparent, and inclusive communication ecosystem where users have greater autonomy over their data and interactions.

However, decentralized chat applications also face challenges such as scalability, usability, and regulatory compliance. Achieving mainstream adoption and overcoming these obstacles will require ongoing innovation, collaboration, and community engagement from developers, researchers, and users alike. Overall, decentralized chat applications have the potential to revolutionize online communication by empowering individuals, enhancing privacy and security, and promoting decentralized governance models. As technology continues to evolve, decentralized chat applications will play an increasingly important role in shaping the future of digital communication.

REFERENCES

[1] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang, ―An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends‖, 2017 IEEE 6th International Congress on Big Data.
[2] S. Nakamoto, ―Bitcoin: A peer-to-peer electronic cash system,‖ 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[3] W. Peters, E. Panayi, and A. Chapelle, ―Trends in cryptocurrencies and blockchain technologies: A monetary theory and regulation perspective,‖ 2015. [Online]. Available: http://dx.doi.org/10.2139/ssrn. 2646618
[4] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, ―Hawk: The blockchain model of cryptography and privacypreserving smart contracts,‖ in Proceedings of IEEE Symposium on security and Privacy (SP), San Jose, CA, USA, 2016, pp. 839–858.
[5] Y. Zhang and J. Wen, ―An iot electric business model based on the protocol of bitcoin,‖ in Proceedings of 18th International Conference on Intelligence in Next Generation Networks (ICIN), Paris, France, 2015, pp. 184–191.
[6] Vaibhav Shakya, PVGN Pavan Kumar, Lakshay Tewari and Pronika. "Blockchain based Cryptocurrency Scope in India." IEEE Xplore Part Number: CFP21K74-ART; ISBN: 978-0-7381-1327-2. (ICICCS 2021)
[7] Ridhanshi Bhatia, Praveen Kumar, Shilpi Bansal and Seema Rawat. "BLOCKCHAIN –THE TECHNOLOGY OF CRYPTO CURRENCIES." In ICACCE-2018.
[8] Ujan Mukhopadhyay, Anthony Skjellum, Oluwakemi Hambolu, Jon Oakley, Lu Yu and Richard Brooks. "A Brief Survey of Cryptocurrency Systems." white paper 2016.
[9] Wenzheng Li and Mingsheng He. "Comparative Analysis of Bitcoin, Ethereum, and Libra" 978-1-7281-6579-0/20/$31.00©2020 IEEE.
[10] Dr. R. Raju, M. SaiVignesh and K. Infant Arun Prasad. "A Study of Current Cryptocurrency Systems" In 2018 International Conference On Computation Of Power, Energy, Information And Communication (ICCPEIC). 978-1-5386-2447-0/18/$31.00 ©2018 IEEE.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY