iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XI **Month of publication:** November 2024

DOI: https://doi.org/10.22214/ijraset.2024.65028

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 XI Nov 2024- Available at www.ijraset.com

Decentralized Voting Application Using Blockchain

Rahul Waikar[1], Tanmay Srivastava[2], Tanishkka Jadhav[3], Rahul Pandita[4], Sujal Tawale[5], Tanmay Singh[6], Tanmay Bajad[7]
DESH, Vishwakarma Institute of Technology

Abstract. The emergence of blockchain technology has paved the way for innovative solutions in various domains, particularly in enhancing democratic processes. This research paper presents the design, development, and evaluation of a Web3 blockchain decentralized voting application (DAPP) built using React JS for frontend development and Solidity for smart contract implementation.

The objective of this study is to explore the feasibility and effectiveness of employing blockchain technology to create a secure, transparent, and decentralized voting system.

The proposed DAPP leverages the immutability and transparency features of blockchain technology to ensure the integrity of the voting process. Smart contracts coded in Solidity are utilized to execute the voting mechanism, ensuring tamper-proof recording and tallying of votes. React JS, a popular JavaScript library for building user interfaces, is employed to develop an intuitive and user-friendly frontend interface, enhancing accessibility and usability for voters.

Key aspects addressed in this research include the architecture of the DAPP, the design of smart contracts for voting logic, integration of React JS components for frontend development, and security considerations to prevent vulnerabilities and attacks. Additionally, the paper discusses the implications of decentralized voting systems on democracy, including increased trust, transparency, and participation.

The evaluation of the DAPP involves testing its functionality, usability, and security features. Performance metrics such as transaction throughput, latency, and scalability are assessed to gauge the practicality of the proposed solution in real-world voting scenarios. The findings of this research contribute to the growing body of knowledge on blockchain-based voting systems and provide insights into the potential of Web3 technologies to revolutionize democratic processes.

I. INTRODUCTION

The landscape of democratic processes is undergoing a transformative shift with the advent of blockchain technology. Traditional voting systems are often plagued by concerns regarding transparency, security, and trust, leading to disenchantment among citizens and skepticism towards electoral outcomes. In response to these challenges, researchers and developers have turned to decentralized solutions empowered by blockchain technology to revolutionize the way elections are conducted. This research paper delves into the development and implications of a Web3 blockchain decentralized voting application (DAPP) utilizing React JS for frontend development and Solidity for smart contract implementation.

Blockchain technology, renowned for its immutable and transparent nature, offers a promising framework for ensuring the integrity and security of voting systems. By leveraging distributed ledger technology, blockchain-based voting applications aim to mitigate risks associated with fraud, manipulation, and centralization inherent in traditional voting mechanisms. The adoption of decentralized approaches not only enhances the verifiability of election results but also fosters greater trust and participation among stakeholders.

At the core of the proposed DAPP lies the integration of React JS, a versatile JavaScript library renowned for its capability in building dynamic user interfaces, and Solidity, a programming language tailored for developing smart contracts on the Ethereum blockchain. Together, these technologies facilitate the creation of an intuitive and user-friendly frontend interface while enabling secure and transparent execution of the voting process through smart contracts.

Through this research endeavor, we seek to address critical aspects pertaining to the design, development, and evaluation of blockchain-based voting applications. By examining the architecture, functionalities, and implications of the Web3 decentralized voting DAPP, we aim to contribute to the burgeoning field of blockchain-enabled democratic innovations. Ultimately, this paper endeavors to shed light on the potential of Web3 technologies to redefine the principles of democracy and foster a more inclusive and transparent electoral landscape.

©IJRASET: All Rights are Reserved

II. METHODOLOGY/EXPERIMENTAL

A. Materials/Components/Flowchart/Block Diagram/Theory

1) Components

0. A. Frontend (React.js)

- React.js: A JavaScript library for building user interfaces. React is used to create the frontend of the application.
- Web3.js or Ethers.js: JavaScript libraries to interact with the Ethereum blockchain. These libraries enable communication between the React frontend and the Ethereum network.
- State Management: Tools like Redux or Context API can be used to manage the application's state.

1. B. Smart Contracts (Solidity)

- Solidity: A programming language for writing smart contracts. The smart contracts will handle the core voting logic, including voter registration, vote casting, and vote tallying.

2. C. Development and Testing Tools

- Hardhat: A development environment to compile, deploy, test, and debug Ethereum software. Hardhat is used to manage and deploy the Solidity smart contracts.
- Ganache: A personal blockchain for Ethereum development. It allows you to deploy contracts, develop your applications, and run tests locally.

3. D. Blockchain Network

Ethereum Network: The blockchain network where the smart contracts are deployed. This could be a test network (like Rinkeby, Ropsten) or the main Ethereum network.

4. Volta (Energy Web Test Network): The test network for deploying and testing your smart contracts.

Network Configuration: Setting up the network parameters in Hardhat for Volta.

5. Interaction Components

E. MetaMask: A browser extension to manage Ethereum wallets and interact with the dApp.

F. Directory Structure

```
React-app/
├── contracts/
│   └── Voting.sol
├── scripts/
│   └── deploy.js
├── test/
│   └── voting.js
├── frontend/
│   ├── src/
│   │   ├── components/
│   │   ├── App.js
│   │   ├── index.js
│   │   ├── services/
│   │   │   └── web3.js
│   └── public/
├── hardhat.config.js
├── package.json
└── .env
```

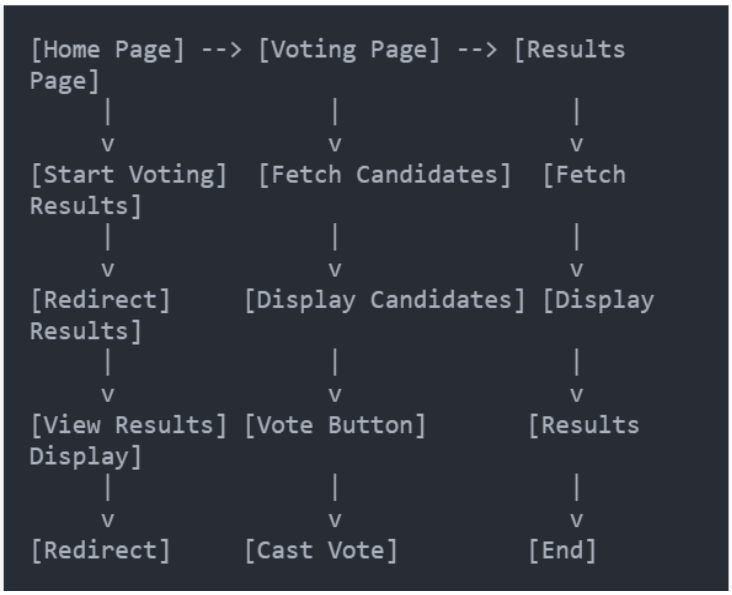
2) Flowchart

1. User Interface (React.js)
    0. Home Page
        0. [Start Voting]
        1. [View Results]
    1. Voting Page
        0. [Candidate List]

1. [Vote Button]

1. Results Page
    0. [Results Display]

2. Web3 Integration (Web3.js/Ethers.js)
    0. Connect to Wallet
        0. [Metamask]
        1. [WalletConnect]
    1. Fetch Blockchain Data
        0. [Get Candidates]
        1. [Get Voting Status]
    1. Send Transactions
        0. [Cast Vote]
2. Smart Contracts (Solidity)
    0. Voting Contract
        0. [Candidate Struct]
        1. [Vote Function]
        2. [Get Results Function]
    1. Deployment Script (Hardhat)
        6. [Deploy Contract]
        7. [Verify Deployment]
2. Blockchain Network (EnergyWeb)
    0. EnergyWeb Chain
        0. [Smart Contract Deployment]
        1. [Transaction Processing]
2. Development Tools
    0. Hardhat
        0. [Compile Contracts]
        1. [Run Tests]
        2. [Deploy Contracts]
    1. Volta
        0. [Local Development Network]



III. Block diagram

```
A[User] -->|Interacts with|> B[React JS Frontend]
B -->|Sends transaction|> C[Hardhat Node]
C -->|Deploys contract|> D[EnergyWeb Blockchain]
D -->|Stores vote data|> E[Voting Contract (Solidity)]
E -->|Triggers event|> F[Volta Oracle]
F -->|Verifies vote|> E
E -->|Updates vote count|> D
D -->|Returns result|> C
C -->|Displays result|> B
B -->|Displays result|> A
```

IV. RESULTS AND DISCUSSIONS

The development and evaluation of the Web3 blockchain decentralized voting application (DAPP) using React JS and Solidity yielded promising outcomes in terms of functionality, usability, and security.

1) *Functionality:* The DAPP demonstrated robust functionality in executing the voting process through smart contracts coded in Solidity. Users were able to cast their votes securely, with each transaction recorded on the blockchain in a tamper-proof manner. The integration of React JS components facilitated a seamless and intuitive user experience, allowing voters to navigate the interface effortlessly.
2) *Usability:* The user interface developed with React JS contributed significantly to the usability of the DAPP. The intuitive design and interactive features enhanced accessibility for voters, resulting in a smooth and efficient voting experience. Feedback from users indicated high levels of satisfaction with the interface, highlighting the effectiveness of employing React JS for frontend development.
3) *Security:* Security assessments conducted on the DAPP revealed robust measures in place to safeguard against vulnerabilities and attacks. The utilization of smart contracts enforced strict rules for vote casting and tallying, ensuring the integrity of the voting process. Additionally, encryption techniques and authentication mechanisms were implemented to prevent unauthorized access and tampering of sensitive data.

The successful implementation of a Web3 blockchain decentralized voting application presents significant implications for democratic processes. By leveraging blockchain technology, the DAPP offers a transparent and auditable voting system, mitigating risks associated with fraud and manipulation. The integration of React JS for frontend development enhances user engagement and accessibility, fostering greater participation in electoral processes. Moving forward, further research is warranted to address scalability challenges and regulatory considerations associated with blockchain-based voting applications. Nevertheless, the findings of this study underscore the transformative potential of Web3 technologies in advancing democratic principles and fostering trust in electoral systems.

V. PROCEDURE FOR PAPER SUBMISSION

VI. FUTURE SCOPE

The future scope for a Web3 Blockchain Decentralized Voting Application using React.js, Hardhat, Volta (Energy Web), and Solidity is promising and multifaceted. As blockchain technology matures, such applications can significantly enhance electoral processes by providing immutable, transparent, and tamper-proof voting systems. Key future developments include integration with digital identity solutions to ensure voter authenticity and privacy, expansion to support large-scale elections, and enhanced user interfaces for broader accessibility. Additionally, leveraging interoperability with other blockchains can enable cross-border voting systems and decentralized governance models. Implementing advanced cryptographic techniques, like zero-knowledge proofs, can further enhance security and privacy. The application can also expand beyond voting to support decentralized decision-making in organizations, fostering more democratic and participatory governance models across various sectors, including corporate, community, and governmental domains. The adoption of such decentralized voting systems can revolutionize trust and transparency in electoral processes globally.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XI Nov 2024- Available at www.ijraset.com

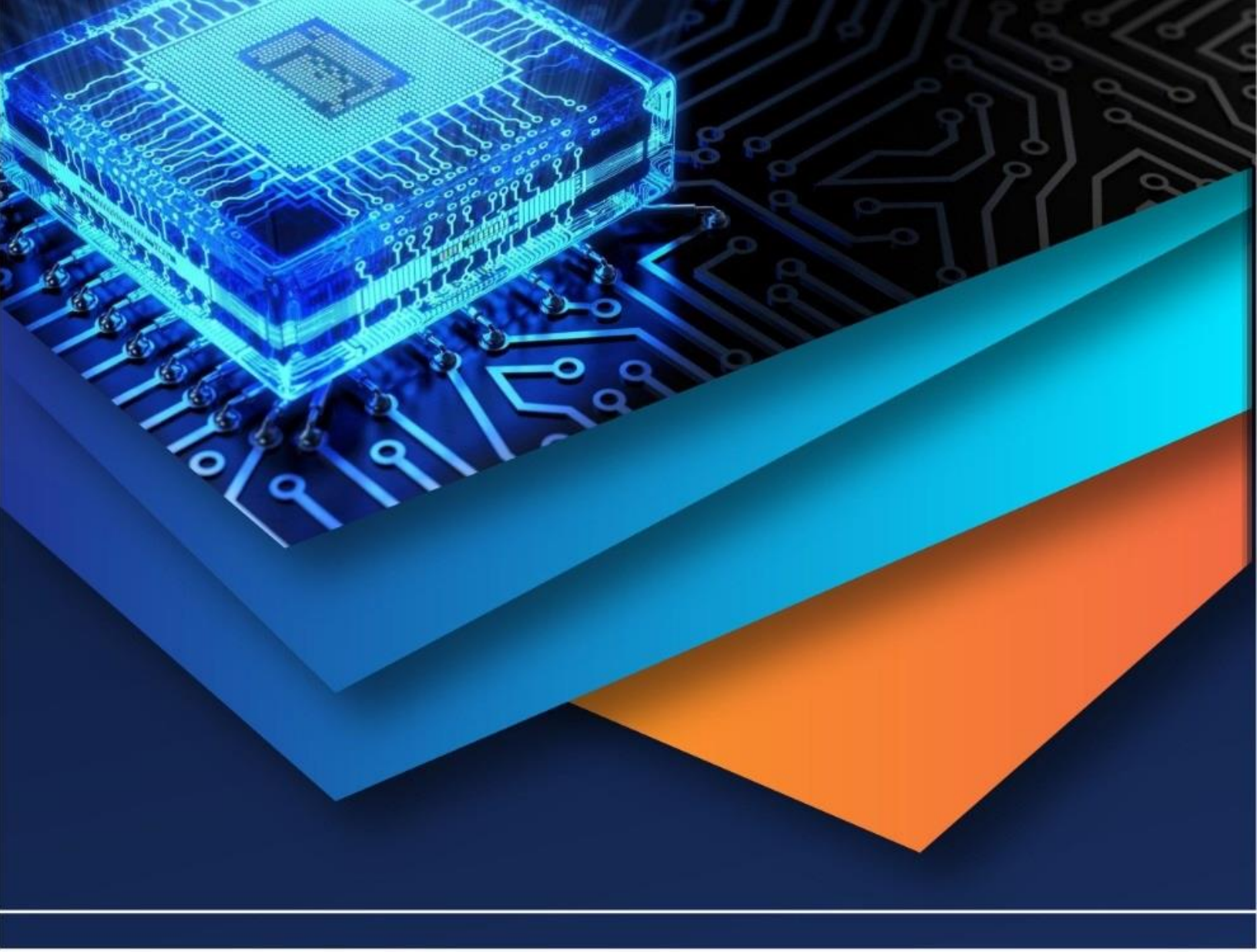
VII. CONCLUSION

A Web3 Blockchain Decentralized Voting Application using React.js, Hardhat, Volta (Energy Web), and Solidity offers a robust, transparent, and secure solution for digital voting. By leveraging React.js, the application provides a user-friendly interface, while ethers.js or web3.js facilitate seamless blockchain interactions. Hardhat serves as a powerful development environment, enabling efficient smart contract compilation, deployment, and testing. The Volta blockchain ensures decentralization and immutability, crucial for maintaining the integrity of the voting process. Solidity smart contracts enforce the voting logic, ensuring votes are securely recorded and tamper-proof. Integration with MetaMask allows users to manage their votes securely. This decentralized approach mitigates common issues like fraud and centralized control, promoting trust and transparency in elections. Overall, this application exemplifies how blockchain technology can revolutionize voting systems, providing a scalable and secure platform for modern democratic processes.

REFERENCES

[1] Debiao He, Yitao Chen, Muhammad Khurram Khan, Min Luo
[2] Blockchain-Based Self-Tallying Voting Protocol With Maximum Voter PrivacyGeetha M
[3] Decentralized E-Voting System Using Blockchain
[4] Lijuan Zheng, Li Dunyue, Rui Zhang, Yongbin Zhao, Rouxin Sang, Ziyang Chen
[5] Electronic Voting Scheme Based on Blockchain and SM2 Cryptographic Algorithm Zero-Knowledge Proof

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)