iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45717

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 10 Issue VII July 2022- Available at www.ijraset.com

Etrog: Ethereum Based Social Media

Sumukh R [1], Uvais Mon V V N [2], Vignesh V [3], Zabiulla Sheriff [4], Yashpal Gupta S [5]
[1, 2, 3, 4, 5] *Department of Information Science and Engineering, Vidyavardhaka College of Engineering, Mysuru, India*

Abstract: Today's OSNs are controlled by major companies such as Facebook, Instagram and Twitter who have complete control over user data. These companies store the user data in a centralized server which.This has several issues. For example, if their servers go down there no backup copies might be available and the chances of getting hacked are more in a centralized architecture. These companies can also censor the contents and post targeted ads in the user feeds. To overcome the drawbacks of the centralized OSNs, in this paper we are implementing a decentralized social media with the help of Ethereum smart contracts. Here, the data is not stored in a centralized server and the major disadvantages of centralized servers can be overcome by this approach.

Keywords: Blockchain, Smart contracts, Ethereum, Online Social Network.

I. INTRODUCTION

In this modern era everything is driven by data. When we take social media almost every application is using a centralized data storage system and most of these companies are using our personal data as a way to make more income by selling these data to other firms. Those firms are using this data for their purposes such as advertising etc. Therefore it is proved once we upload any data to these applications it's not private anymore. And also due to centralized storage we can lose data easily if the storage crashes. And the current systems are using centralized servers due to which we may face server down problems if the server crashes.

Etrog:

To Overcome all these issues we are proposing a new system which is based upon a decentralized network. Etrog is a Social media web application which uses an ethereum network to store data and it's built on Django framework. To connect applications to blockchain users have to create the ethereum wallet first after that they need to set up the environment variables by adding the ethereum wallet credentials. Thereby the application will be connected to the users block which will be residing in the ethereum blockchain network. SInce we are using blockchain to store the contents the data cannot be tampered, hence data will be secured.And we are storing all the images in firebase, and here user can create their own buckets and set the permission on their own and can give those credentials in our application to store the images. Since we have deployed the smart contract on ethereum there is no need to deploy our application on some centralized servers, we can run the application on the local server itself. So we don't need to worry about the server down issue.Since we are storing data inside the blockchain network the data cannot be deleted, So we don't need to worry about data crashes.

II. LITERATURE SURVEY

In [1] Chao Li,et al have presented an empirical analysis on steemit an incentivized blockchain based social media where there is no centralized storage of information and users are given incentives for the contributions they do.The analysis on steemit features i.e decentralized management and reward system showed that decentralization is very low in steemit than the ideal levels. This shows that DPoS consensus protocol may not be an ideal approach for establishing a decentralized social media and it also showed that 16% of crypto transfers to curators that are suspected to be bots. This paper provides an insight on the current cryptocurrency based social media including the effectiveness of design and working of consensus protocol and the incentivized system in social media

In [2] Le Jiang,et al implemented a blockchain based framework for OSNs which is combined with smart contracts where the blockchain is used as a trusted server instead of centralized server. The BCOSN provides security,efficiency and privacy they have also used encryption of to provide data security.The author also lists the limitations of this implementation such as implementing this can be very expensive as user data are stored in blockchain they may not comply with the existing laws that govern the social media,the efficiency of the blockchain is also major issues while deploying these kind of implementation projects

In [3] Anwitaman Datta,et al proposed an OSNs that includes P2P infrastructures with encryption and direct interaction between users.The prototype that the authors are proposed is a two tier system with direct communication among the peers and a separate lookup service.The authors have developed a protocols for direct communication and lookup service among peers.The protocols proposed span the range of typical centralized OSNs such as user login,upto date feed,posting content .

After implementing the protocol the next step is to integrate this solution for security,encryption, privacy control and also access control.The authors also proposes that a complete analysis of security and robustness of the PeerSon needs to be done.

In [4] Hrishikesh Bawane,et al mentioned the disadvantages of centralized OSNs such as data security,censorship of contents,data availability,to overcome these disadvantages the authors have proposed an ethereum based social media with rewards to users based on their contribution in the social media. The smart contracts are written in solidity. This language does not support complex data structures such as multidimensional arrays. The implementation is done only as a web application and works are in progress to develop a mobile application without compromising the functionality of the OSNs.

In [5] Keyur Paralkar, et al Photogroup is a decentralized image sharing social media platform that uses the Ethereum platform to store data.To initialize the Ethereum clients the system uses Ganache. The system is built using the truffle framework.This system enables the user to see, share, like and comment on the images shared by users.Here each user has to be a part of the blockchain and the new blocks are created for each user when he/she creates the account in this system. Each user block will contain the data of that block and the data of the previous block. The system uses the IPFS technology to move the data from the local storage to the blockchain using the ethereum smart contracts.

In [6] Shovon Paul,et al ,It is a system that is based upon blockchain that enables the users to manage, trace and helps the users to claim the ownership of the data they share. It mainly contains four key components i.e., blockchain, Hash table, Turing and a local personal certificate authority. Since it is a blockchain based social network the data stored in this system will be tamper proof and permanent. This system represents the state transition as a reduction of token value which represents the number of transactions that can be performed by that asset.

Here the data can be shared between the user and the users who are present in the user's circle respectively.The personal certificate authority encrypts the data with the public key of the user's circle and this encrypted data is then stored in the hash table.

The table contains three columns. ie,1) the hash of content that is encrypted with them, 2) the hash of the decrypted content that is re-encrypted using their user circles public key so that it can be reshared among other users, 3) the content that they encrypted.

In this social network the users share the hash id of the content they encrypted with every person in their user circle.The data transactions are stored in the blockchain with the identity of the user, the hash identity of the content to file the trails and the token value which is set by the owner of the content which will be reduced after each share. If the value of this token is zero then the content with that value can't be shared anymore.

In [7] Quanqing Xu, et al have taken to social media platforms for experiment. Here they have created a social media application which is based on ethereum. The small data will be stored in the ethereum using smart contracts. Since the solidity was not supporting the return of the complex data type such as struct at that time they have built the interplanetary file system(IPFS) to store larger data. The application contains the frontend webpage from where the users will interact with the system.

In [8] Koushik Bhargav Muthe, Decentranet is a decentralized internet which uses the blockchain network. The author has created a proof of concept of this system. Due to the current architecture of the internet, scalability becomes an issue for production. So the author proposes to create an incentives based system where the proxies get the incentives for their participation in the network. During that time they have used the Ethereum 1.0.The author also described that the Ethereum 1.0 is based upon the proof of work. This system is not reliable for complete decentralization.The author thinks the upcoming Ethereum 2.0 which is based upon proof of stake can help in decentralizing the complete system.

In [9] Mehrnoosh Mirtaheri et al A novel computational approach is presented for identifying and characterizing cryptocurrency pump and dump operations that are carried out on social media. Given the financial data and twitter data pertaining to a particular coin, this method detetcts, with sufficient accuracy, whether there is an infolding attack on that coin on telegram and whether the resulting pump operation will succeed in terms of meeting the anticipated price targets. Activities of users involved in pump operations are also analyzed and observe the prevalence of twitter bots in cryptocurrency related tweets in close proximity to the attack. In future work, the plan is to augment our data sets with other sources to help with the tasks of prediction . While the analysis relied on suspend accounts for bot activity , it looks interesting to develop a bit detection tailored to the crypto domain . As a practical outcome of the work presented here, it is envisioned that building a cryptocurrency monitoring system will detect impending pump attacks in real time and warn vulnerable users.

In [10] Shahar Somin,et al it is demonstrated for the first time that the ERC20 tokens transactional data displays several properties known to be in association with networks that are composed of human interactions especially referring to social media. This occurs despite the fact that the protocol of blockchain enables creation of unlimited tokens , causing diverse sub domains to reside together over the same protocol and regardless of an unlimited amount of wallets,resulting in different identities controlled by a single individual.

Specifically, we have modeled the transactions as a network that consists of wallets connected through transactions and found that the degree distribution of nodes in the network presents a power law pattern. In addition, we have shown that the token's popularity among buyers and sellers also follows a power law model. These initial results show that despite its vast diversity, ERC20 data represents social behavior, leading to further exploring of other aspects of network theory that can emerge from this. Such fields include short path lengths and clustering coefficient analysis, centrality measures, connected components behavior and community structure study. It has already been demonstrated some of these phenomena using the ERC20 data as well, however they were not included in this work due to space considerations.

III. IMPLEMENTATION DETAILS

A. Introduction To Technologies Used

1) *Blockchain:* In recent years, blockchain has become a popular technology. It joined the market as a crypto currency infrastructure. Blockchain may now be used to store and distribute data over the blockchain network thanks to smart contracts. The data that is written to the blockchain is unchangeable. Any modifications will need the creation of new blocks, although the prior data logs will remain accessible. This prevents data on the blockchain from being tampered with. Because blockchain is based on a distributed network, it eliminates the possibility of a single point of failure. Every sector is currently attempting to convert its business processes to blockchain. The growing number of users of crypto currency is a sign of blockchain technology's progress.

2) *Ethereum:* Ethereum is a blockchain technology that is open source. It is compatible with smart contracts. Ethereum transactions are both durable and immutable. Ether is the Ethereum network's native coin. Every transaction on Ethereum will be subject to a minor gas cost. The charge is also determined by the amount of data that must be written to the contract. As a consensus method, Ethereum employs Proof of Work. Proof of Work necessitates a node solving a difficult mathematical challenge. Finding the puzzle's solution should be difficult, but confirming the answer that has been provided should be simple. The challenge should neither be too difficult to solve, nor should it be too simple to allow a DOS attack to succeed.
Proof of Work has the following mathematical model.

An irreversible function f is defined for the entire network. For any transaction t, y is defined. A node is expected to find x such that,

$$f(x) = y, \quad (1)$$

Where,

$x \rightarrow$ *Nonce*

$y \rightarrow$ *Required hash prefix.*

Now, consider 3 nodes that provide the answers as x1, x2 and x3 respectively. Any participating node can check which answer is correct by applying the function on the answers as follows.

$$f(x1) = y1,$$
$$f(x2) = y2,$$
$$f(x3) = y3.$$

Then they compare y1, y2 and y3 with y. The corresponding value of x for which f(x) matches y is selected as the answer and the node who generated x is rewarded. The approval of the solution has to be done by 51% of the participating nodes. Finding the value of f(x) is a rather effortless process. So evaluation of the provided answer is easy.

3) *Ethereum Virtual Machine:* EVM is a blockchain-based software platform. EVM aids in the deployment of smart contract bytecode. It's a full Turing machine. An EVM implementation will run the smart contract on every complete node of the Ethereum network. EVM resembles a state machine. It will start in one state, accept input, and then switch to another.

$$F(S, T) \rightarrow S` \quad (2)$$

Where,

S is an old state,

T is a valid transaction,

F is an Ethereum state transition function, and

S` is the new state

4) *Smart Contracts:* Smart contracts are a groundbreaking technology that allows programmers to deploy programmes on the Ethereum Virtual Machine's blockchain for execution (EVM). In blockchain 2.0, smart contracts were introduced. Distributed apps arose as a result of this. Smart contracts will be used to read and write data to the blockchain, as well as ensure data privacy and security.

5) *Solidity:* Solidity is the programming language used to write smart contracts for Ethereum. The smart contract for the system logic will be written in Solidity. A solidity program will consist of compiler version declaration followed by imports and contract definitions. A contract will consist of state variables, modifiers and functions. State variables track the current state of the contract. The function helps the contract to modify the state variable, access the state variable data or perform other transactions. State variables are changed by transactions. Once the smart contract is written, it is compiled using solc compiler. The bytecode generated is deployed on the EVM. All interaction of the system with the blockchain will be through this smart contract.

6) *Django:* Django is an open source full stack web development framework for the python programming language. A django project consists of a project folder and different application folders. The project folder consists of settings file and root url file along with other project related metadata files. All the project level settings are set inside the settings file. The root url file defines the urls of the entire project. The urls can also be redirected to any url file inside any other project folders.
The main component of an application folder are views and models files. The views files define the backend logic that should be executed for every single url that exists for the application. The view file then renders a web page passing in all the dynamic data called contexts. The templates to render the required web page are stored in a folder named templates. The models file stores the database schema used in the project in the form of object relational mapping.

B. Methodology

The application consists of mainly 2 modules - the smart contract and the web app client. Entire logic for data storage, data access and user authentication is defined in the smart contract. Web app client consists of all the logic to fetch appropriate data and render it in an easily comprehensible user interface. This approach ensures that even if the users change any source code of the web app client, it only changes how the data is displayed but it cannot cause any foul play in authentication or authorization as these logics are implemented at smart contract level and users will not have access to modify a deployed contract. The media files uploaded by the users are stored at a Firebase storage bucket and are indexed at the smart contract. Users can set file access policies as they want on their firebase storage. Since all user media files are stored on their own Firebase storage, they own their entire data. No other information on the Firebase storage that helps to track back to the user is stored. The flow of the activity in the application is described below.

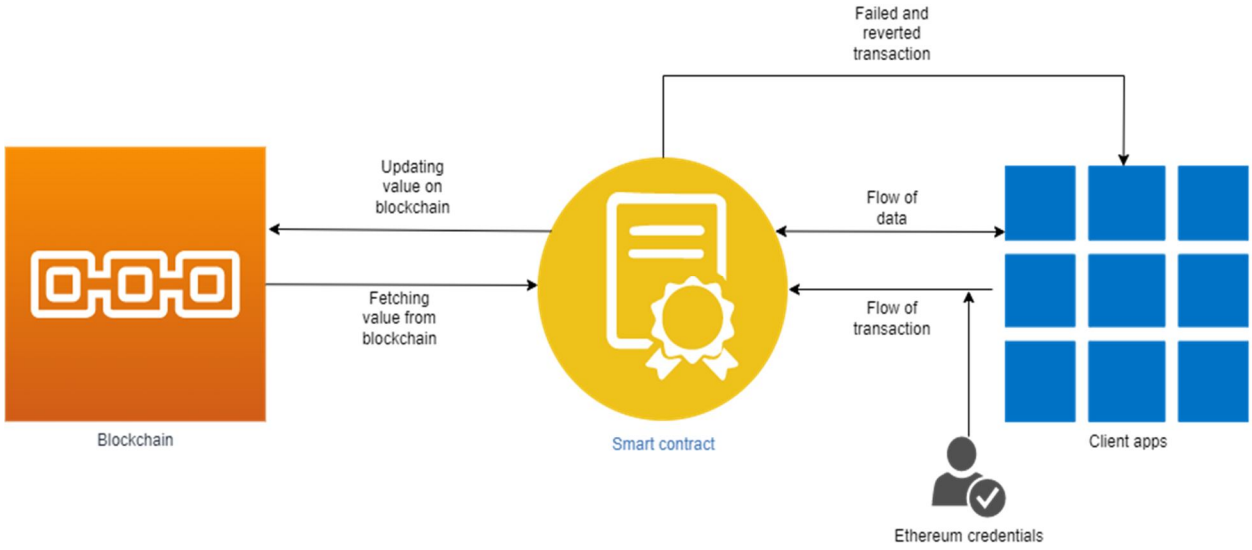


Fig 1: Etrog - System architecture

Users have to download the client program, install necessary dependencies and execute the client program locally. The .env file should be populated with user credentials that includes Ethereum account address, private key and firebase credentials. Once the client is running, new users have to first register themselves. User authentication is performed by Ethereum account address and private key and no exclusive password is used to register to the application. After registering, they can start posting, send follow requests to friends, approve follow requests sent by others, view and comment on posts of accounts they are following, and modify or delete any data they generated.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

C. Smart Contract Module

The smart contract handles all the authentication and authorization logics. The structures defined in smart contract are: profile_details, post_details, comment_details. Here profile_details stores all user data such as username, first name, last name, followers list, following list and all other personal details. Post_details stored all the data related to a particular post. This structure object also contains the URLs of the media files associated with a particular post. It also indexes all the comment_details objects of all the comments of a particular post. The comment_details structure stores all the user comments of a particular post.

Necessary mapping objects to map to a list of the above structure objects are created to easily index any post or comments. It helps to fetch a list of all the posts posted by a particular account or fetch all the comments of a particular post.

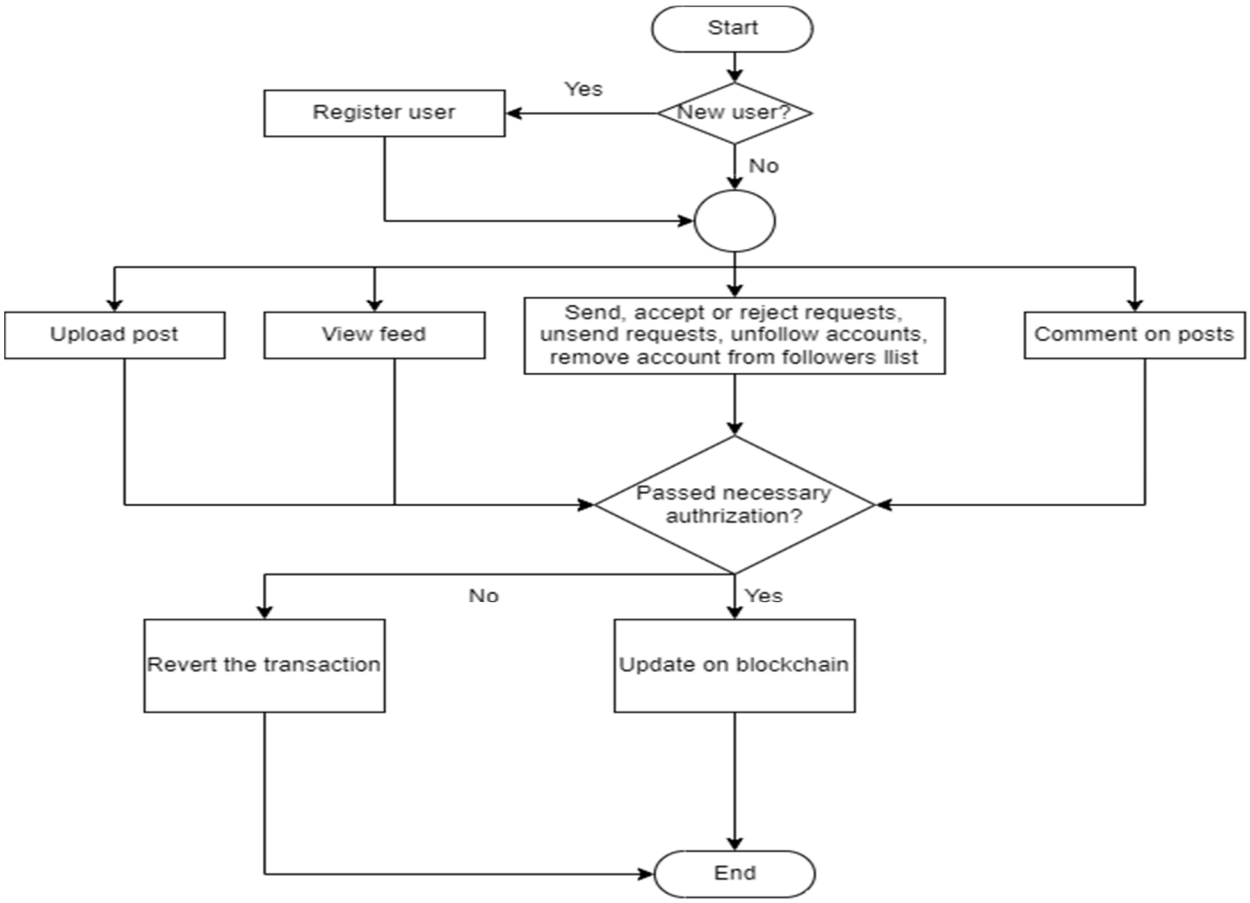


Fig 2: Etrog - Flow of different processes

Functions to perform various activities such as user registration, uploading posts and comments, fetching various user data, and sending, approving or rejecting, and canceling user requests are implemented. User authorization is done using the msg.sender value, which returns the Ethereum account address of the transaction initiator. Each activity that modifies any data on the smart contract is considered as a transaction. The client app signs a transaction using their Ethereum account private key. This helps to verify the identity of the sender and accordingly check if they have necessary rights to access the data.

The contract is developed and deployed on Ethereum's main network. The contract address is stored as a constant on the client app program. Contract once deployed will not be updated or redeployed. So the contract address does not change after the development of the app.

Web app client

Web app client is a django app. The app consists of a contract helper file. This file consists of all the interfaces required to communicate with the deployed smart contract for data exchange and execution of smart contract functions. The .env file stores all the user credentials and the constants.py file stores all the global constants related to the app. The global constants consists of the blockchain network URL and the smart contract address.

The views.py file consists of all the user side logic. It responds to the user interaction through web pages on browsers, calls necessary smart contract helper functions from the contract helper file, fetches data from the smart contract and renders it on the web page in a user friendly manner. The functions inside views.py consist of logic that makes up each web page of the web app. The templates folder consists of django template code (similar to HTML program) that decides how the data passed from the views.py functions should be rendered.

The entire client app is developed in a modular fashion. If the user does not want to store their media files on firebase storage, they can tweak the file_handler.py file and define their own storage bucket. If the user does not want to become a part of the global network but would rather like to have a private smart contract for their small circle of friends, then they can deploy the smart contract on any network they want and change the network URL and smart contract address in the constants.py file.

This provides users total control over their data. They can decide where to store their data and who can access their data.

IV. CONCLUSION

The developed application is a decentralized social media app. Users get complete control of their data. Users can customize the location where their data will be stored. They can deploy the smart contract at a different address and have many instances of the same app. Users get the authority to decide where their media files will be stored and they can decide the access policies for those media files. No data related to the ownership of the media files are stored on the storage bucket. Therefore, no data mining can be performed at the storage bucket by the bucket host. Users can also set up their own home servers to serve as file buckets if they do not want to rely on any 3rd party buckets.

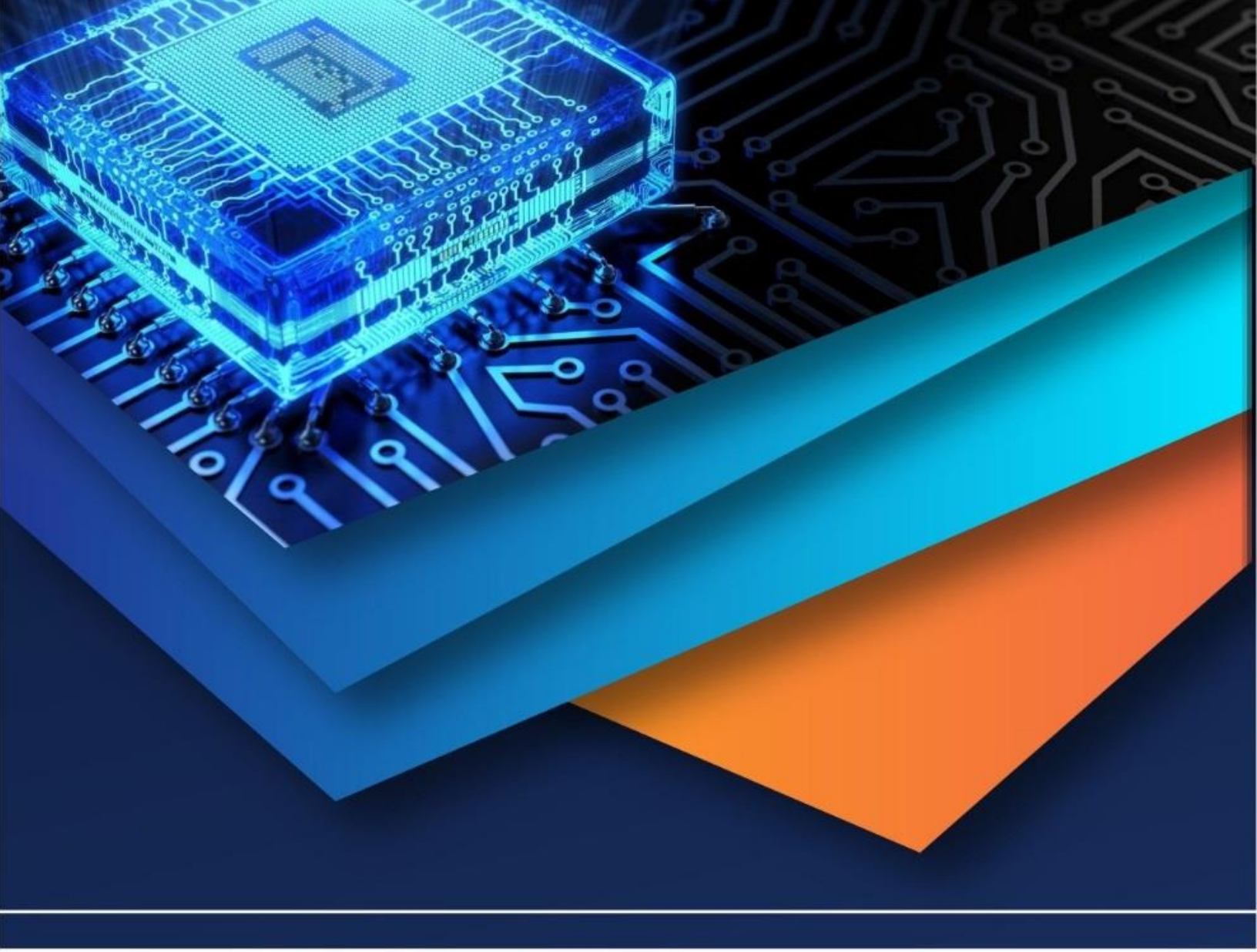
V. FUTURE WORK

The contract is deployed on Ethereum Testnet for testing purposes. Deploying the app on Ethereum Mainnet is costly. Users will soon find it expensive to perform actions on the app. To mitigate this, a new blockchain, exclusively used for the proposed app should be used. All the users of the application should be allowed to participate as a full node or as a miner of the blockchain. This provides an optimized and cost effective blockchain network exclusively developed for social media use cases.

Security of the data stored on the blockchain could further be increased by using an encryption system. There could be a central system that just performs the encryption and decryption of the transactions. The central system will store user credentials for the encryption and decryption purpose. The central system however will not store any other data that could be used to mine user information. This however poses a single point of failure. Users can also run their own instance of encryptor to mitigate single point of failure. However there will be a tradeoff between single point of failure and availability of users online. To mitigate a single point of failure, users will have to be online to provide the distributed service.

REFERENCES

[1] Chao Li, Balaji Palanisamy, “Incentivized Blockchain-based Social Media Platforms: A Case Study of Steemit”, WebSci ’19, June 30–July 3, 2019, Boston, MA, USA
[2] Le Jiang, Xinglin Zhang, “BCOSN: A Blockchain-Based Decentralized Online Social Network” 2019, IEEE transactions on computational social systems.
[3] Sonja Buchegger, Doris Schioberg, Le-Hung Vu, Anwitaman Datta, “PeerSoN: P2P Social Networking — Early Experiences and Insights” SNS March, 2009: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems.
[4] Hrishikesh Bawane,Tanuja Shinde,Abhishek Kadam,Yash Budukh,Prof. Pooja Mundhe, “Ethegram - An Ethereum and IPFS-based DecentralizedSocial Network System”, 2020, IRJET.
[5] Keyur Paralkar, Shiwani Yadav, Shikha Kumari, Apurva Kulkarni, S.P. Pingat, “Photogroup : Decentralized Web Application Using Ethereum Blockchain”, 2018 IRJET.
[6] Antorweep Chakravorty, Chunming Rong, “Ushare: user controlled social media based on Blockchain”, 2017, Conference : ACM IMCOM.
[7] Quanqing Xu, Zhiwen Song, Rick Siow Mong Goh, Yongjun Li, “Building an Ethereum and IPFS-based Decentralized Social Network System”, 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS).
[8] Koushik Bhargav Muthe,Thiru Srinivas Teja Vemuru, Khushboo Sharma, Nilofer Sultana Mohammad, “DECENTRANET - AN ETHEREUM, PROXY RE-ENCRYPTION AND IPFS BASED DECENTRALIZED INTERNET”, 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)
[9] Mehrnoosh Mirtaheri , Sami Abu-El-Haija, Fred Morstatter , Greg Ver Steeg, and Aram Galstyan, “Identifying and Analyzing Cryptocurrency Manipulations in Social Media”, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 8, NO. 3, JUNE 2021.
[10] Shahar Somin, Goren Gordon and Yaniv Altshuler,“Network Analysis of ERC20 Tokens Trading on Ethereum Blockchain”, 2018 Springer.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY