Ethegram - An Ethereum and IPFS-based Decentralized Social Network System

Hrishikesh Bawane[1], Tanuja Shinde[2], Abhishek Kadam[3], Yash Budukh[4], Prof. Pooja Mundhe[5]

[1-4]B.E. Student, Information Technology, MIT College of Engineering, Pune, Maharashtra, India
[5]Assistant Professor, MIT College of Engineering, Pune, Maharashtra, India

---***---

Abstract - *In today's world, most of the services are controlled by a centralized authority, causing concerns regarding user privacy, bandwidth usage and security of user data. As the blockchain technology has evolved, it has caused many applications to be distributed and decentralized without the loss of security. Ethereum is an open-source, blockchain-based, decentralized software platform. It enables Smart Contracts and Decentralized Applications (DApps) to be built and run without any downtime or interference from a third party. The InterPlanetary File System (IPFS) is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. Several DApps have been created with the use of these technologies and one such application is to use this design for a decentralized Social Media system that is resistant to censorship and single point of failure. In this paper, we propose Ethegram, a social network decentralized application created using these technologies and a simple rewards system. In addition, this paper examines how blockchain and distributed storage can solve the problems prevalent in traditional social network systems.*

***Key Words*: Decentralized Application, Social Network, Blockchain, Ethereum, InterPlanetary File System, Smart Contract**

1. INTRODUCTION

Blockchain, the technology that runs bitcoin, has been the top trending buzzword in technology for many years now. Blockchain is a distributed ledger that can be described as a ledger of transactions or contracts maintained in decentralized form at many different locations, eliminating the need for a central authority to keep a check against manipulation. All the information on it is securely and accurately stored using cryptographic hash functions. The use of social network services like Facebook, Twitter, Instagram, etc. has increased exponentially over the last decade. These platforms have a large number of users across the world. Among many other incidents, recently, Facebook was again caught on the back foot over its data privacy practices, disclosed by the New York Times which brought the security and privacy issues of social media networks to a higher level [1]. Thus, many researches are running through to overcome the challenges faced by existing social media platforms.

Due to the traditional client-server model, social media clients often suffer from service unavailability during server downtime or when server faces issues of single point of failure or DDoS attacks [2]. For example, Bitcoin blockchain was used to provide decentralized content trust for docker images to resist potential threats in docker content trust [3]. It is a well-known fact that governments of some countries have blocked social media services based on the IPs used by the social media servers. It is quite obvious that blockchain technology provides a way to address issues existing in the industry. Blockchain technology enables decentralization and security for the social media services, which makes it available during server downtime or even against censorship from governments or Internet Service Providers (ISPs).

Blockchain-based social networks are the future. They are the next generation of networks which will help an individual in many ways. Users will get control over their personal data. Users will get better compensation for their contribution. Users will be able to moderate the content according to them. Finally, users will get faster and cheaper payments from all over the world.

In this paper, we propose Ethegram, a social media platform developed by using the state-of-the-art o blockchain technology and introduce a social networking decentralized application. The application is based on Ethereum blockchain platform [4] for data records and IPFS [5] for distributed data storage service. The application would reward users for quality content they contribute on the platform. The Smart Contract of this decentralized application is deployed on the Ethereum Virtual Machine (EVM) and the frontend webpage UI of the application would interact with it using web3.js collection of libraries. The purpose of developing this application is to emphasize the significance of blockchain technology in improving traditional social media architectures.

The rest of the paper is structured as follows: Section 2 provides an overview of the technologies and concepts used in the development of application. Section 3 describes every aspect of the proposed solution in depth while section 4 presents the related work on similar approaches using blockchains and decentralized architectures for social networking and we conclude with section 5.

2. BACKGROUND

In this section we will discuss the technologies and concepts that are used to build Ethegram. This section includes brief introduction to Blockchain, Ethereum Virtual

Machine, IPFS (InterPlanetary file system), and Smart Contracts.

2.1 Blockchain

Blockchain is a continuously growing chain of blocks, each of which contains a cryptographic hash of the previous block, a time-stamp, and its conveyed data. Due to the existence of the cryptographic hash, the data stored in a blockchain are inherently resistant to modification: if one block of data is modified, all blocks afterward should be regenerated with new hash values. This feature of immutability is fundamental to blockchain applications. Maintenance of peer-to-peer (P2P) ledgers for cryptocurrencies has become the first application of blockchain. Thousands of cryptographic tokens, or coins, were delivered to the public market, after the huge leap in market cap of Bitcoin. Ethereum is a blockchain-based decentralized platform featuring smart contract functionality [6]. It supports stateful contracts in which values can persist on the blockchain to be used in multiple invocations.

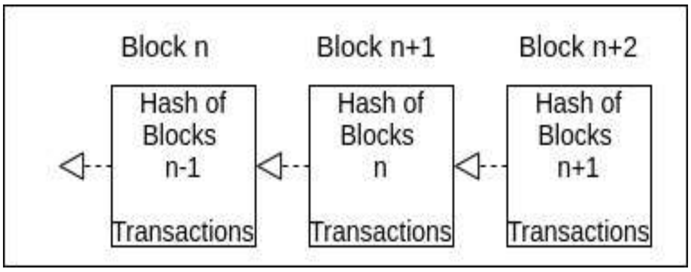


Fig - 1: Basic blockchain structure

2.2 Ethereum Virtual Machine

The Ethereum Virtual Machine (EVM) is a powerful, sandboxed virtual stack embedded within each full Ethereum node, responsible for executing contract bytecode. Contracts are typically written in higher level languages, like Solidity, then compiled to EVM bytecode. This means that the machine code is completely isolated from the network, file system or any processes of the host computer. Every node in the Ethereum network runs an EVM instance which allows them to agree on executing the same instructions. The EVM is Turing complete, which refers to a system capable of performing any logical step of a computational function. JavaScript, the programming language which powers the worldwide web, widely uses Turing completeness.

2.3 Smart Contract

A smart contract is a piece of program which runs on the blockchain and its execution is examined and enforced by the relevant consensus protocol. Smart contract is a special protocol designed to contribute, verify or implement the negotiation or performance of the contract [7]. It is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. These transactions are traceable and irreversible. Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism. A smart contract is invoked in current cryptocurrencies by sending transactions to the contract address deployed on the blockchain. This will cause a state change that will be reflected in the change of account balance or the state value of contract.

2.4 InterPlanetary File System (IPFS)

The InterPlanetary File System (IPFS) is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices. IPFS allows users to not only receive but host content, in a similar manner to BitTorrent. As opposed to a centrally located server, IPFS is built around a decentralized system of user-operators who hold a portion of the overall data, creating a resilient system of file storage and sharing. Any user in the network can serve a file by its content address, and other peers in the network can find and request that content from any node who has it using a distributed hash table (DHT).

3. PROPOSED SOLUTION

In this section, we will discuss all the architectural and design components of our proposed solution.

3.1 System Overview

Blockchain 2.0 technology (i.e., smart contract) has enabled a distributed P2P network between non-trusting peers without a third-party intermediary. We aim to apply the knowledge of Ethereum blockchain and smart contract for finding solutions to real world issues. This task is to develop a decentralized social network application based on Ethereum blockchain along with the use of InterPlanetary File System as the distributed storage service and hence bring merits to the current social media network architecture. The task also includes providing a straightforward, content-based rewards system for the users in order to compensate them for the quality content they contribute to the platform.

3.2 Architecture Design

The Smart Contract of this application will manage the entire transactions of the DApp (See Appendix). The user will have to register once initially with his/her wallet/account address and this user address will be stored in a mapping in the smart contract. Each account on the application will be bound to a single account address only, in order to avoid creation of multiple accounts from same address. This smart contract will be deployed on Ethereum Virtual Machine and the frontend UI will interact with users. Hence, the entire system forms a three-level architecture where the frontend UI is responsible for user interaction while the web3 and IPFS libraries are responsible for API calls from frontend client to the backend blockchain and the IPFS storage respectively. Figure 2 shows a complete view of the system structure. At the top is the frontend UI, which is in charge of receiving user input for registration data, posts data and messages and passing them to the web3 library and IPFS library. The web3 library then interacts with underlying Ethereum blockchain system for function calls, contract deployment and fund transfers. IPFS library is invoked when the user creates a post with an image or video, or when the user sends a message in the chat system of the application. The image, video or chat messages are stored on the IPFS data storage and a IPFS hash is returned which is stored on blockchain as record data.

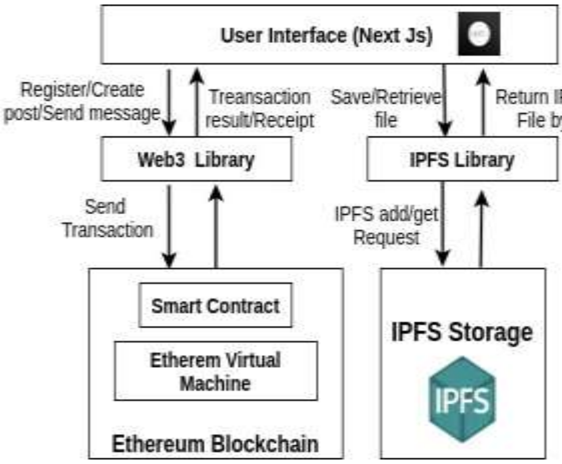


Fig - 2: System Architecture

3.3 Application Design

The entire system comprises of three main components, namely the backend Ethereum blockchain, IPFS storage and frontend web UI. When the client starts, the user can register into the application only if he/she is not already registered. This is handled by the users mapping and frontend. If the account address of user already exists in the users mapping, the frontend registration is disabled, else the frontend sends a transaction along with 0.002 Ether (the cryptocurrency of Ethereum blockchain) to the blockchain and a new user account is registered with the application. This amount goes into the contract balance and it enables the contract to reward the users based on the likes (or upvotes) their content gets. It also discourages malicious actors to get registered on the application.

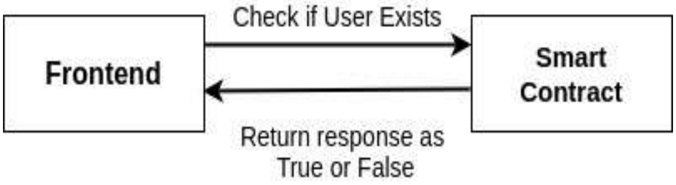


Fig - 3: Process for registering

While creating a new post (see Figure 4), if an image or video is uploaded by the user along with the post, the frontend client loads the image or video into a byte array and sends the byte data to IPFS storage endpoint through IPFS library. When the data storage is successful, it returns the corresponding IPFS hash in return. This hash is stored on the blockchain and is later used to retrieve the image or video when required. All the post data along with creator's address, image/video IPFS hash and current timestamp is sent as a transaction to the blockchain along with an amount of 0.001 Ether to the blockchain and the post is created. Other users can now like (upvote) or comment on the post.

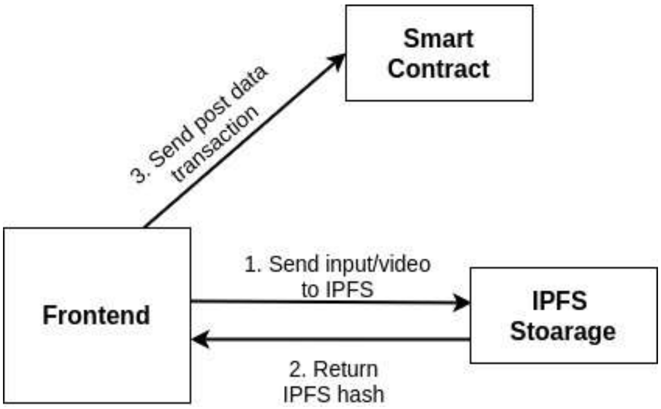


Fig - 4: Process for creating post

As Solidity does not support passing of complex data structures like array of strings, the messages of chat system of the application are stored as a single string with delimiters to indicate separate messages. When the user wants to send a message in the chat system (see Figure 5), it first retrieves the previous messages stored as byte data on the IPFS storage using the IPFS hash of chat messages stored on the blockchain. The frontend client converts the byte data into string, appends the current message and again sends it to IPFS storage endpoint in the form of byte array. The IPFS hash returned is then replaced and stored on the blockchain.

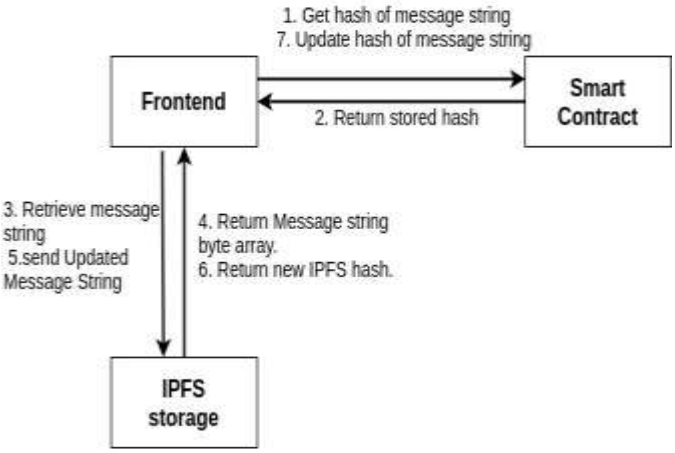

1. Get hash of message string
7. Update hash of message string
Frontend
Smart Contract
2. Return stored hash
3. Retrieve message string
5.send Updated Message String
4. Return Message string byte array.
6. Return new IPFS hash.
IPFS storage

Fig - 5: Process for sending message

3.4 Rewards System

When the user creates a post, other users of the application can view and upvote the post if they like the content. Each upvote generates a single token in the application account of the creator. This token count can be seen in the UI profile section of the user. These tokens can be redeemed into Ethers from the platform itself. The funds are directly transferred to the account address from the contract balance. Each token when redeemed gives 0.0001 Ether. Thus, the cost of creating a new post is recovered when the user receives 10 upvotes on his post. More upvotes would ensure that the one-time registration cost is also recovered. This makes the rewards system purely content dependent. Content creators are rewarded only based on how much their content is liked by other users of the application.

3.5 Smart Contract Design

The Smart Contract of this DApp consists of two main components: one is the *User* structure array and the other is the *Post* structure array. The *User* structure array stores data of all registered users of the application. The data comprises of components like user's name, account address, count of the posts he/she has created, total tokens and redeemable tokens. The name and account address are recorded when the user registers initially on the application, while the count of posts is incremented with every new post created by the user. The redeemable tokens show the number of tokens that can be redeemed into Ether. Once redeemed, the corresponding number is decremented from the redeemable tokens. Total tokens and redeemable tokens are incremented every time the creator receives an upvote on any of his/her posts.

Another component is the *Post* structure array which stores entire data of posts that are uploaded by the users on the platform. It includes the content created by the user consisting of text, images or videos in the form of IPFS hash, creator's address, timestamp, comments string and mapping which stores the users' addresses who have liked (upvoted) the post. When a user upvotes a certain post, the user's address is stored in this mapping so as to ensure that he/she would not like the same post again.

4. RELATED WORK

Several previous and current works have been working on developing decentralized applications with the use of Ethereum and IPFS as distributed data store. Here we discuss these applications and also the work that we referred to for insights and development of our application.

4.1 *Akasha* Project

Akasha is built on top of Ethereum and aims to solve the problem of impermanence of information online [8]. The cornerstones of the *Akasha* stack are Ethereum and IPFS, augmented by Electron, React with Redux, and Node.js. The *Akasha* Project also aims to build a knowledge architecture for social human advocacy in the context of social networks, freedom of expression, creative perpetuity and privacy for a better Internet in service of humanity.

4.2 *Synereo*

Synereo provides a decentralized and distributed social network designed for an attention economy. It offers a platform more as a social market place.

4.3 *Ushare*

Ushare provided a conceptual solution for creating a user centric social network that would enable users to control, trace and securely share content [9]. It also supported offsite encryption of data and mechanisms to share them through the blockchain.

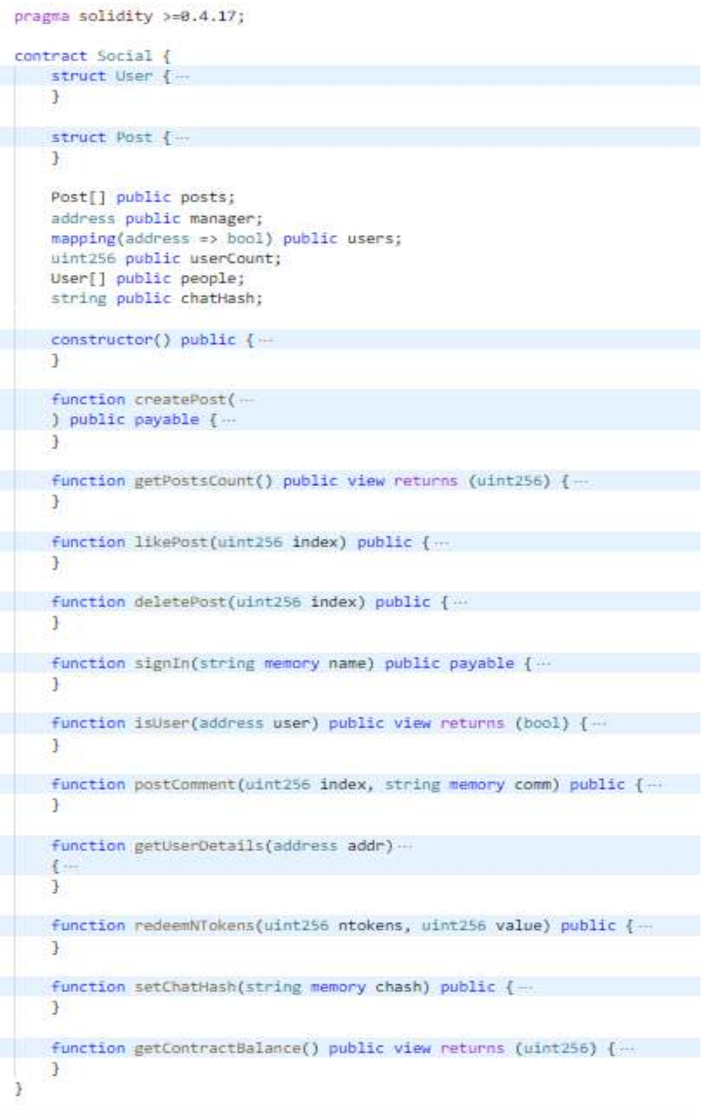
These solutions are still in their early phase of inception and are not available for evaluation. Even if the underlying technologies may coincide with Ethegram, the features provided or problems solved by them differ. Unlike these solutions, Ethegram provides a robust and profitable rewards system which incentivizes the time spent by the users on the platform. Ethegram also provides a chat feature over the secured architecture for users to interact with other users on the platform.

5. CONCLUSIONS

Misuse and monetization of user data by the social media companies, censorship issues, data security and data availability issues due to centralized industry are major concerns of people using social networks worldwide. In this paper, we presented a possible solution to these problems with the development of Ethegram, a social network as a decentralized web application. In addition, our proposed platform Ethegram, would also reward users based on the quality content they contribute to the platform. All the mentioned functionalities were successfully developed. The most important components of our underlying technologies due to which a purely decentralized application was developed are Ethereum blockchain and IPFS distributed storage system.

Future work needs to be done in order to develop the same as a mobile application without compromising on any of the aspects of web application. The smart contract of the application can also be improved since Solidity, the language of Ethereum Smart Contracts, currently does not support passing of complex data structures like multi-dimensional arrays.

6. APPENDIX



```
pragma solidity >=0.4.17;

contract Social {
    struct User {···
    }

    struct Post {···
    }

    Post[] public posts;
    address public manager;
    mapping(address => bool) public users;
    uint256 public userCount;
    User[] public people;
    string public chatHash;

    constructor() public {···
    }

    function createPost(···
    ) public payable {···
    }

    function getPostsCount() public view returns (uint256) {···
    }

    function likePost(uint256 index) public {···
    }

    function deletePost(uint256 index) public {···
    }

    function signIn(string memory name) public payable {···
    }

    function isUser(address user) public view returns (bool) {···
    }

    function postComment(uint256 index, string memory comm) public {···
    }

    function getUserDetails(address addr)···
    {···
    }

    function redeemNTokens(uint256 ntokens, uint256 value) public {···
    }

    function setChatHash(string memory chash) public {···
    }

    function getContractBalance() public view returns (uint256) {···
    }
}
```

Fig - 6: Smart Contract with list of functions

7. REFERENCES

[1] (2018) Facebook Cambridge Analytica data scandal. [Online]. Available: https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge Analytica data scandal.

[2] 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).

[3] Q. Xu, C. Jin, M. F. B. M. Rasid, B. Veeravalli, and K. M. M. Aung, "Blockchain-based decentralized content trust for docker images", Multimedia Tools and Applications, pp. 1–26, 2017.

[4] (2018) Ethereum project. [Online]. Available: https://www.ethereum.org/.

[5] (2018) Interplanetary file system. [Online]. Available: https://ipfs.io/.

[6] Q. Xu, K. M. M. Aung, Y. Zhu, and K. L. Yong, "A blockchain-based storage system for data analytics in the internet of things," in New Advances in the Internet of Things. Springer, 2018, pp. 119–138.

[7] A. Tar. (July 26 2018) Smart contracts, explained. [Online]. Available: https://cointelegraph.com/explained/smart-contracts-explained.

[8] (2018) Akasha project. [Online]. Available: https://akasha.world/.

[9] Antorweep Chakravorty and Chunming Rong, "Ushare: user controlled social media based on blockchain", Conference Paper – January 2017.