Distributed and Serverless Peer-to-Peer Chat application on IPFS

Saba Khanum
Department of Information Technology
MSIT, Janakpuri, New Delhi, India

Gunjan Beniwal
Department of Computer Science and Engineering
MSIT, Janakpuri, New Delhi, India

Abstract- Existing chat application can sense chat. Blockchain is a fully-decentralized communication platform. It allows you to message and video chat directly with anyone in the world, without any of the conversation information being stored. The mobile device / user is verified over the blockchain which means the identity of the user remains anonymous unless users choose to share. Once connected, the user is able to speak privately with no intermediaries. Users can even earn, send and receive digital assets as one chat. But the main disadvantage of this application is that all of the conversation data is ephemeral, meaning it isn't stored. This is the limit that our chat application would be overcoming. This paper presents a peer-to-peer distributed blockchain database messaging presents the development of an instant messaging application based on the concept of peer-to-peer using Inter-planetary File System (IPFS) and Blockchain.

Keywords – Decentralized, Blockchain, IPFS, Identity protection, chat application.

I. Introduction

Blockchain is a newly emerging and disruptive technology that can be an important aspect in providing a solution to authenticity of digital materials. Blockchain is the technology of the cryptocurrency but now is seen as a distributed ledger that can be accessed globally by anyone to verify stored data and content, with high integrity and above all traceability. All of this is done in a decentralized manner and without intermediaries. Later, Ethereum smart contracts provided the ability to upload and execute code. The smart contract code resides on a blockchain as multiple functions with unique addresses that can be called by any user of the blockchain.

Blockchain, however, is a medium for data storage, especially for large data and digital content. For efficient storage of large data and content, we propose using an IPFS file system for storing information. IPFS stands for Inter-Planetary File System which is a distributed, decentralized file system and a platform to store data such as photos and other multimedia, files with high integrity. Fundamentally, IPFS is a peer-to-peer, open source, content addressable globally distributed file system that can be used for storing and sharing large volumes of files with high throughput and making sure data is secure very effectively. Our proposed solution makes use of blockchain, smart contracts and IPFS, whereby the digital contents are stored on the IPFS and the IPFS hashes are stored into the blockchain smart contracts to provide traceability and authenticity which in turn are stored in our website. The website keeps record of our hashes and provides the backup system using the wallet. Specifically, the hash generated on storing the documents to IPFS, can be stored in the smart contracts effectively. If there is any change in the content of the digital document, the hash changes, to show that the original content was modified and altered. We propose a combined IPFS-blockchain-based solution to solve the security issue of content posted freely. We show how this problem can be solved for pictures, but our solution can be extended and adopted for, to other multimedia content[1].

Our solution has the ability to trace and track the content. The different published versions id traced back. We propose an IPFS-blockchain-based solution and framework for providing security to the content. Our solution provides a decentralized storage system and governance with different versions of the original content being stored, tracked, and traced with high integrity.

The rest of the paper is organized as follows. Literature Review is discussed in section II. Method and implementation screen shorts are presented in section III. Concluding remarks are given in section IV.

II.LITERATURE REVIEW

A. Background –

According to data from Statista, the total number of smartphone users across the globe will reach the 2.5 billion mark by the end of 2019. Smartphones along with the advent of various chat applications have made it incredibly easy for people to communicate, however, it is only in the last few years that people realized the cost of communicating with such ease. The price users pay for seamless communication through various instant messaging apps in today's centralized world. Two chatting application in use and using blockchain are obsidian and status network [2][3].

Obsidian (ODN) is a Proof-of-Stake coin which is going to offer master nodes and service nodes for messaging. Obsidian Master nodes are going to require 10,000 ODN ($4500 at the current market price) and will get fee transfers from Obsidian Secure Messenger (OSM.) The team, led by developer Raides J. Rodríguez, has already delivered a prototype of the OSM application which is in alpha testing. OSM has a robust roadmap for adding support for master nodes which will help secure the network and as the reward would get 10% ODN annually. The minimum amount at stake is going to be 10000 ODN providing investors a considerable incentive to hold their coins are driving prices up in the medium to longer term.

The Status Network is the mobile ethereum operating System is the king when it comes to blockchain based chat apps. The most prominent goal that Status has is to give users more control over their personal information. Status offers encrypted messaging, a built-in ether wallet and a browser for Decentralized Applications running on the Ethereum blockchain. Status is the one of the most prominent company in blockchain based messaging platforms which is at over $780 million in market capitalization compared to Obsidian's $10 million as of writing this article. Obsidian offers more privacy-related features compared to Statuses like end-to-end encryption, anonymous user registration and economic incentives like the ability to stake coins. Since Obsidian has a fraction of the market capitalization of Status, it has more room to grow as it is mostly a community-driven project. By comparison, Status is already working similar to a corporation and has a strict roadmap in place for further development [4].

In this section, we provide a brief background on the existing approaches found in the literature related to authenticity and originality of books in the publishing industry using blockchain technology. Ericsson [5] proposes a blockchain-based system for tracking the origin of digital assets. It is by converting the digital content and books into a binary file and to store the hash on blockchain. This hash is stored generally with an identifier for the owner. The paper focuses on the idea that ownership can be verified by checking the integrity of digital assets at any point of time. This is achieved through the verification by a centralized unit for security of digital documents such as Security Operations Center (SOC) to estimate the legitimacy of digital assets. But the system itself deviates from the decentralized concept as it operates in the presence of a centralized security unit and has a precarious state for breach of integrity. Moreover the file storage is done on a centralized server which can be a single point of failure, corruption, hacking or compromise.

Usually in authentication procedure username and password is asked and further, message exchange is only allowed after peers have authenticated to each other. Unlike Skype, when a user initializes the application, one is prompted for Gaetani et al [6] to propose a verification ID which can be used for blockchain-based authenticity of digital assets. This ID is inherently a digital block on the blockchain that can be used for verification of the e-document.

The author in [7] proposes a blockchain-based model for publishing online books and for providing integrity of the digital document. The author achieves authorship by storing the book/file hash and the owner's name in pairs. The author argues that by storing the hash of the file and the block timestamp as pairs, integrity of the document/file can be proved. If the content of the file was modified, then its hash will change, and the smart contract won't be able to access the file, therefore proving that the file content was modified. Sun et al. [8] describes a framework for evaluating the trust issues when storing online documents in decentralized networks. In this paper, authors present a framework to quantitatively measure trust, model trust propagation, and defend trust evaluation systems against malicious attacks. This system was used to secure ad hoc routing and support to unmask malicious nodes in a decentralized environment but is not yet implemented as a real-world application.

The authors in [9] propose a blockchain-based personal data management system to ensure that document owners have complete authority over their asset. This model features a blockchain-based automated online document access control system thereby eliminating trust in a third party. Blockchain and off–blockchain storage is combined to onstruct a management platform which precedes to trust based computing. But the work does not describe the easibility of storing larger files. Morgan [10] presents the idea of using blockchain technology to prove the existence of a document using the timestamping concept. The author discusses a method where the document is presented to a site which in turn converts the document into a cryptographic hash. The hash generated represents the content of the document. If the original document is presented, the same hash will be generated, notifying that the document is authentic. However, if there is any modification of content, the newly generated hash will not match with the previous hash. The legitimacy of the document can be verified, but this system is not focused about the authority of the owner on his/her document.

Acronis Notary system described in [11] is a blockchain-based notary service which aims at providing a solution for timestamping digital documents. As blockchain is a very expensive storage medium for storing large documents, the proposed approach is to send file hashes to the Notary service. This service calculates hash value, based on the received file hashes and saves the new hash obtained, on the Ethereum network. A a verification certificate is provided specifying the technical details of the document. Both the chatting server are not using Inter Planetary File System (IPFS)

III.PROPOSED WORK

The whole idea of this proposed application is to avoid a centralized system (registration, login and buddy list) as that found in Skype. Using Skype, during registration, the user profile will be stored in a centralized database and one can use the credential to login at anytime and anywhere as preferred. This certainly provides a certain level of robustness although the question arises as to how secure the centralized database can be used to prevent attacks. Recently, a study on decentralized systems was proposed but only for the purposes of improving the buddy list. The idea was about developing a robust index system using a distributed hash table for decentralized chat applications. An indexing system is responsible for storing the IP address and port of all users once they join the chat. Users initialize their own buddy list by contacting the centralized indexing system once they logged in. When a user A wants to communicate to user B, B will act as a server and authenticate client A. As authentication is one-way, this opens up an opportunity for attackers to masquerade as user B. To cater some of these problems, in our proposed application, we come out with the following principle ideas Pure P2P architecture with no centralized server, peers' profiles is managed locally by user registrations are done among themselves and therefore, multiple registrations are needed for communicating with different users. User login is done on each peer basis and it follows a two-way authentication protocol. Message is encrypted prior to exchanging between two or more peers. Another important feature of any chat application would be buddy listing. In this application, during the initialization process, the system reads through the buddy list from the local hash table and automatically determines their (device) availability by contacting them based on the IP address and port number of their devices. However, it could also be the case that someone else is on the device, for that

A. Inter Planetary File System

IPFS began as an effort by Juan Benet to build a system that is very fast at moving around versioned data. Versioning gives you the ability to track how different versions of software change over time. IPFS is considered as the distributed, permanent web. IPFS is a distributed file system that seeks to connect all computing devices and machines with different systems with the same system of files. In some ways, this is similar to the original aims of the web, but IPFS is actually more similar to a single bit torrent swarm. IPFS have the potential to be a new major subsystem of the internet. If built right, it could be used with net or replace HTTP. At its core, IPFS is a versioned file system that can take files and manage. It is also stored by them somewhere and then tracks versions over time. IPFS also accounts for how those files move across the network which makes it distributed file system. IPFS has rules as to how data and content move around on the network [12]. This file system layer offers very interesting properties such as:

1. Websites that are completely distributed and have no origin server

2. Websites that can run entirely on client side browsers or do not have any servers to talk to.

Content Addressing

Instead of referring to objects (pics, articles, videos) by which server they are stored on, IPFS refers to everything with something unique- by the hash on the file. The idea is that if in your browser you want to access a particular page then IPFS will ask the entire network to find the corresponding hash to it. Then a node on IPFS that does can return the file allowing you to access it. IPFS uses content addressing at the HTTP layer. This is the practice of saying instead of creating an identifier that addresses things by location; we're going to address it by some representation of the content itself. This means that the content is going to determine the address. The mechanism is to take a file, hash it cryptographically so you end up with a very small and secure representation of the file which ensures that someone cannot just come up with another file that has the same hash and use that as the address. The address of a file in IPFS usually starts with a hash that identifies some root object and then a path walking down. Instead of a server, you are talking to a specific object and then you are looking at a path within that object.

IPFS Nodes: IPFS is decentralized. Without a typical server providing web pages for every client that arrives at the website's domain, a different infrastructure must be imagined. Every machine running IPFS would be a node as part of a swarm. Consider the way torrents currently work. You choose a file to download, and when you use a torrent application to do so, you're essentially sending out a request to all of the computers attached to the same torrent network as you, and if any of them have the file you're requesting, and are able to upload at the moment, they begin sending pieces of it to your computer.

Hashing and IPNS: Every file that exists on IPFS would have a unique hash to represent it, and any minute change would result in a new hash being generated. These hashes are how content can be viewed. A client queries the system for a hash, and any node that has that content available can serve it to peers. The "swarm" provides a torrent-like experience, wherein peers are capable of serving each other content. This system will allow content to be served quickly and accurately to clients, regardless of their proximity to the original host of the content. Additionally, because hashes are employed, both ends of the exchange can be checked for correct content, as a single bit out of place would result in a different hash. The Inter-Planetary Naming System (IPNS) can be used to assign a name to mutable (changeable) content, so that your node publishes a piece of content, has a name attached to it, and then is able to republish changes with the same name. This, of course, could result in loss of available content, so IPNS entities, according to the developers, may someday function more like a Git commit log, allowing a client to iterate back through versions of the published content.

Advantages of Decentralization: Reliability and Persistence and Secured against DDoS-style Attacks

The content being served on the IPFS network is going to be around, essentially, forever, if people want it to be. There's not any single weak link, server, or failing point. With larger files, there may be a benefit to having multiple peers as options for your IPFS to choose from to acquire the file. But the real benefit comes from having those multiple options to start with. If one node hosting it goes down, there will be others.

Simply by its nature, distributed peer to peer content cannot be affected by "Direct Denial of Service" style attacks. These attacks are primarily concerned with bombarding host servers to bring down websites or services. However, if the same content is being served to you from multiple peers, an effective DDoS attack would have to find and target all of them.

With the caching system in place with IPFS, it's entirely possible that quite a lot of your regularly viewed content would be available offline by default. Any dynamic content might not be up to date, of course, but previously viewed static content resources could be at your fingertips whether you were in range of your Wi-Fi or not.

With IPFS as a major player, things would definitely change. Although IPNS nodes can be mapped to HTTP addresses currently, they would not necessarily need to be forever. Web browsers might change, or be removed entirely. Browsers, or other clients, might be the only necessary software. Remember that IPFS is peer to peer, so your IPFS installation is simply .The diagrams below depicts the working of the project.

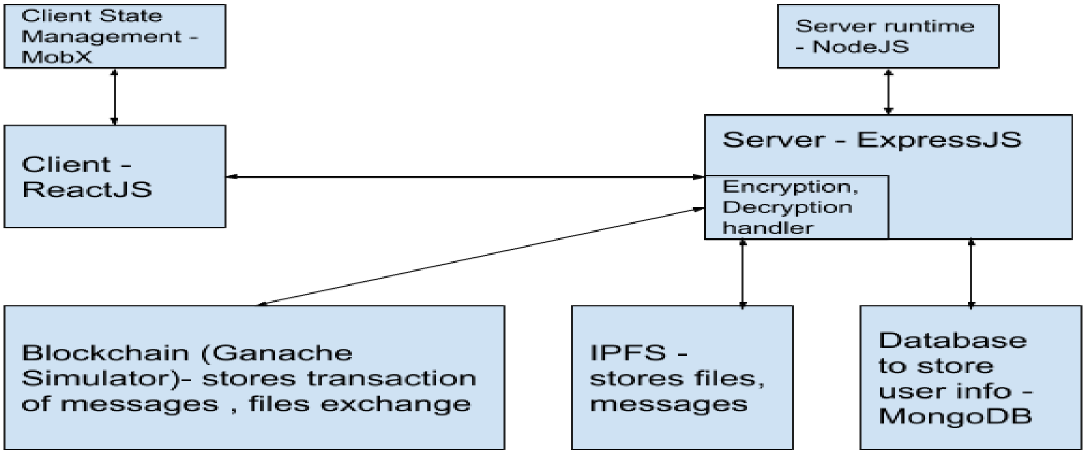
Client State Management - MobX
Server runtime - NodeJS
Client - ReactJS
Server - ExpressJS
Encryption, Decryption handler
Blockchain (Ganache Simulator)- stores transaction of messages , files exchange
IPFS - stores files, messages
Database to store user info - MongoDB

Figure 1: System flow chart with technologies

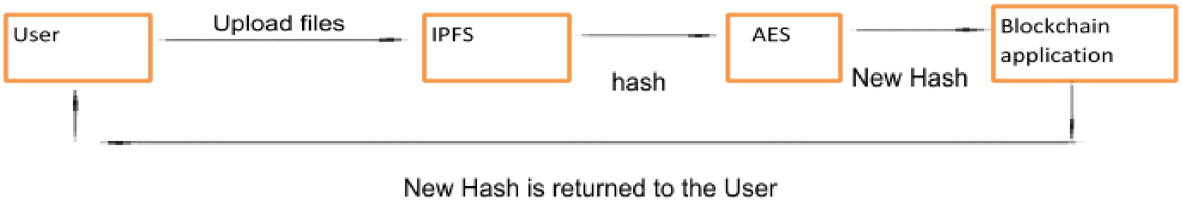

Figure 2: Flowchart of user registry

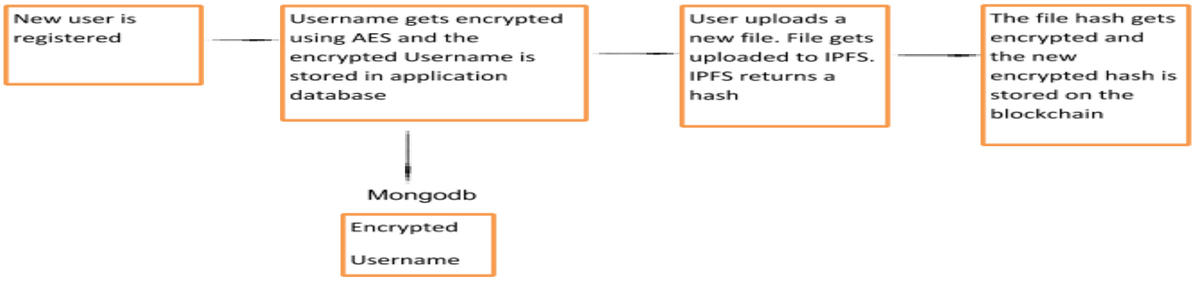
New user is registered
Username gets encrypted using AES and the encrypted Username is stored in application database
User uploads a new file. File gets uploaded to IPFS. IPFS returns a hash
The file hash gets encrypted and the new encrypted hash is stored on the blockchain
Mongodb
Encrypted Username

Figure 3: Flowchart of file uploading

In our Chat Application, a user can either create a new Channel or Join an existing channel to communicate with our users. The “handleJoinChannel” is used to join an existing Channel. The user can choose to join any public channel also. When a user sends a request to join a public channel, the asynchronous function “handleJoinChannel” is called and the Channel name is passed to this function. The channel is fetched from the Blockchain using the just_chat.join (channel Name)” method of the component. The channel then loads for the User, all events linked to the Channel are initiated and all the previous conversations are fetched and displayed. When the file is sent to the server, the file is uploaded to the IPFS, which returns a unique hash. This unique Hash is passed to the Advanced Encryption Standard Algorithm and a new Hash is generated. This is for the Security of the uploaded file in the Chat Application. The “handleSendFileMessage” is used to bind the uploaded file to the channel. The Channel is fetched using “just_chat.channels” method and then the “sendFile.bind” method binds the file to the channel. After the file is linked with the Channel, the Callback runs which sends the **response to the frontend which initiates the events linked to the Channel, fetches and updates the Channel data and the uploaded file is displayed to all the users on that public Channel.**

IV.CONCLUSION

Centralized chatting application goes through the server of the company owning the messaging app. This problem is due to the presence of a single point of failure. Blockchain in instant messaging help to overcome the major concerns of data security and privacy of users. The use of asymmetric end-to-end encryption and P2P connections eliminates the possibility of intercepting user correspondence. Confidentiality - The absence of servers that store user correspondence excludes the disclosure of this information at the request of government agencies or hacker attacks. Absence of Single Point of Failure - Absence of a server to exchange messages, as well as make audio and video calls, means that there is no central link and thus, no malfunction of a server affecting the service.

REFERENCES

[1] Atzori, Marcella, Blockchain-Based Architectures for the Internet of Things: A Survey (2016). Available at SSRN: https://ssrn.com/abstract=2846810

[2] Cai Y, Zhu D (2016) Fraud Detections for Online Businesses: A Perspective from Blockchain Technology. Financial Innovation

[3] Crosby MA, Pattanayak P, Verma S, Kalyanaraman V (2016) BlockChain Technology: Beyond Bitcoin. Applied Innovation, No. 2, pp. 6–10.

[4] Guo Y, Liang C (2016) Blockchain Application and Outlook in the Banking Industry[J]. Financial Innovation.

[5] Iivari J (2010) Twelve theses on design science research in information systems, in Design Research in Information Systems, Theory and Practice by Hevner, Alan, and Chatterjee, Samir, Springer, US, pp. 43–62.

[6] Kakavand, Hossein and Kost De Sevres, Nicolette (2016) The Blockchain Revolution: An Analysis of Regulation and Technology Related to Distributed Ledger Technologies, Luther Systems. Available at SSRN: https://ssrn.com/abstract=2849251.

[7] Kim, Henry M. and Laskowski, Marek (2016) Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance. CoRRabs/1610.02922.

[8] Lim IK, Kim YH, Lee JG, Lee JP, Nam-Gung H, Lee JK. (2014) The Analysis and Countermeasures on Security Breach of Bitcoin. In International Conference on Computational Science and Its Applications. Springer International Publishing, pp. 720–732.

[9] Paul G, Sarkar P, Mukherjee S (2014) Towards a More Democratic Mining in Bitcoins. In: Prakash A, Shyamasundar R, editors. Information Systems Security. vol. 8880 of Lecture Notes in Computer Science. Springer International Publishing, Switzerland, pp. 185–200.

[10] Peters GW, Panayi E (2015) Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money. Available at SSRN: https://ssrn.com/abstract=2692487

[11] Shi N (2016) A New Proof-Of-Work Mechanism for Bitcoin[J]. Financial Innovation.\

[12] Sun J, Yan J, Zhang K (2016) Blockchain-based Sharing Services: What Blockchain Technology Can Contribute to Smart Cities[J]. Financial Innovation. Underwood S (2016) Blockchain beyond bitcoin. Commun. ACM 59, 11 (October 2016), 15–17 i