GGANBUVERSE: A DECENTRALIZED DIGITAL DISTRIBUTION CATALOGUE FOR MULTI-PLATFORM APPLICATION HOSTING

Telegram: GGANBUversedev

Dedicated to the green-eyed I love

INTRODUCTION

Mobile and Desktop applications are the backbone of our society whether it is web apps, native or hybrid app; they affect every aspect of modern life. Almost every implementations of native, hybrid and web applications are based on the client-server architecture where the source codes and data of applications are stored on centralized servers, requests send to these web servers as APIs or otherwise to retrieve this data. Although useful, this architecture comes with certain drawbacks which are:

- Data Monopoly: Data can be accessed, altered, or removed by any organizations controlling the system
- Censorship: Applications hosted on centralized platforms can be shut down or certain users denied access to these applications.
- Availability: Data is accessed by location-based addressing of the web servers. If the centralized systems stop working (due to overload, denial-of service, distributed denial-of-service, or system errors), users cannot access their data at that time.

In order to solve these issues GGANBUverse proposes a decentralized solution for web/native application hosting using IPFS and Ethereum blockchain technology. The advantages of the IPFS platform are decentralized storage and addressable content. While the Ethereum blockchain provides anonymity, transparency, decentralization and auditability. GGanbuverse uses smart contracts to provide service registration features and; manage the IPFS network and clients. IPFS does not provide any mechanism to protect data privacy, it is therefore important to deploy encryption algorithms into native and webapps. The combination of IPFS, Ethereum blockchain, and encryption could be a feasible solution for the decentralized mobile and web application hosting.

Brief explanation of terminology and technologies used-

1. INTERPLANETARY FILE SYSTEM (IPFS)

   IPFS is a peer-to-peer distributed file system proposed by Juan Benet. Nodes on IPFS communicate directly with each other over a peer-to-peer network and own a key pair (public

and private key). The public key is used to generate NodeID and the private key is used to sign in an IPNS service. When two nodes initialize a connection, they exchange their public keys and check to ensure that their NodeIDs match the public keys being exchanged otherwise the connection is terminated. There are three types of nodes namely: client node, retrieval miner node, and storage miner node.

- Client node: This node only uses the IPFS network for storing and retrieving files. However, it does not provide storage space for the IPFS network.
- Retrieval miner node: This node provides its storage space to store files for other nodes on the IPFS network. Nevertheless, files stored on this node are saved temporarily and removed by the garbage collection feature of IPFS.
- Storage miner node: This node offers storage space for the IPFS network and enables the pinning service to retain files for a long time.

Each miner node on the IPFS network owns a Distributed Hash table (DHT) that stores the location of files and the node's connectivity information. IPFS employs a DHT to support discovery of contents and peers on its network. IPFS uses Kademlia as its DHT.

Data storage: Files are stored inside IPFS objects. Each object can store up to 256 kilobytes of data. An IPFS object is a data structure that includes two fields: The data field is used to store binary data, and the links field containing an array of links that connect to all the other pieces of the file. A link structure is composed of three parts:

- Name: The name or alias of the link.
- Hash: The cryptographic hash of the linked object.
- Size: The total size of the linked object.

Pinning service: In order to keep useful files available, storage miner nodes use the IPFS pinning service to pin important objects. Hence, when the garbage collection is triggered on miner nodes, any of the pinned objects are automatically exempt from deletion.

InterPlanetary Name System (IPNS): Objects on the IPFS network are identified by the cryptographic hash of their contents. Miner nodes locate objects through their corresponding hash values. When updating the content of an object to form a new version a new hash value is generated. Thus, IPNS allows nodes to use their NodeId to publish mutable objects and use their private key to sign these published objects. This ensures that other nodes only use the same link to access these mutable objects. However, an IPNS link is very hard to remember so this link is mapped to a domain name. For instance, /ipns/ <your_peer_id> is mapped to example.com. In order to associate an arbitrary text with a domain name, the DNS provides a type of resource record called the TXT record.

IPFS is a peer-to-peer data storage facility, which could be used effectively to store blockchain networks as well as decentralized applications, the IPFS platform plays an important role for the decentralized web/mobile hosting.

2. BLOCKCHAIN

A blockchain is a decentralized system of nodes that communicate directly with each other through a peer-to-peer network. Some nodes own a ledger consisting of a sequence of

blocks that they have reached consensus. Blocks are linked together via the hash of the previous block. This hash value also uses to verify the integrity of the content of the previous block. The first block of the chain is called the genesis block. Each block is comprised of a block header containing management information of the block as well as the chain and a block body containing set of transactions. Blockchain technology uses consensus algorithms to ensure the synchronization of data on the ledger between miner nodes.

Ethereum is a distributed blockchain network that provides a decentralized turing-complete platform called Ethereum Virtual Machine (EVM). The programs on EVM are called smart contracts. Basically, a smart contract is an agreement of rules between participants which is written in bytecode and executed automatically when specific conditions are met. Smart contracts reside on top of the blockchain network and each implementation of functions in smart contracts is recorded on the ledger of the blockchain network. Solidity is the most popular programming language for writing smart contracts.

3. ERC20

ERC20 is a set of standards that allow developers to create their own tokens built on the Ethereum network. It is the most used standard on the Ethereum network.

4. ENCRYPTION AND DECRYPTION

Encryption is the process of transforming plaintext data into ciphertext protecting data confidentiality. The reverse of the data encryption is called data decryption. Depending on the number of keys used, encryption algorithms are classified into two main categories: asymmetric and symmetric encryption algorithms. In symmetric encryption algorithms, the same key is used for both encryption and decryption, some symmetric algorithms such as DES, TRIPLE DES, AES, RC4, etc. In asymmetric encryption algorithms, encryption and decryption are performed by two different keys. In which, a public key is published and a private key is kept secret. The key pair are related mathematically, one used for encryption and the other used for decryption. Some asymmetric algorithms are RSA, DSA, Diffie-Hellman, etc. Concerning web security, encryption algorithms are often applied for protecting data and securing communication channel.

Currently, IPFS does not provide any solution for privacy-preserving data storage. Therefore, nodes on the IPFS network can view data through hash values they have. The encryption algorithm of choice for protecting website/apps' data is from unauthorized entities is the Advance Encryption Standard (AES) which is a symmetric key algorithm that supports block length of 128 bit and can have a key size of 128, 192, and 256 bits.

5. STORAGE MINER NODES

storage miner nodes are IPFS nodes that provide storage space and enable the pinning service to pin all important data of websites (such as source code web application, sensitive data, etc.). Each storage miner node can host many websites by creating many key pairs, then it publishes each website to each key by executing commands as follows:

```
#ipfs key gen --type=rsa --size=2048 keyname
#ipfs name publish --key=keyname ${hash}
```

6. RETRIEVAL MINER NODES

Retrieval miner nodes have to enable the garbage collection function to remove the old data in the system to preserve free space for new data. Some retrieval miner nodes are used as public gateways that allow users to connect to websites and webapps. Normally, the gateway nodes have high performance and bandwidth. Also, almost all of the gateway nodes and storage miner nodes must be online constantly to remain available for the network

7. PEERS

These are users of GGanbuverse who own miner storage nodes on IPFS and have pledged to rent their storage for clients (independent developers) to store files in exchange for $GGANBU.

8. Admin

The admin is a human or program used to manage the GGANBUverse decentralized app hosting service and responsible for auditing new peers.

DECENTRALIZED HOSTING ARCHITECTURE

A. PROTOCOL

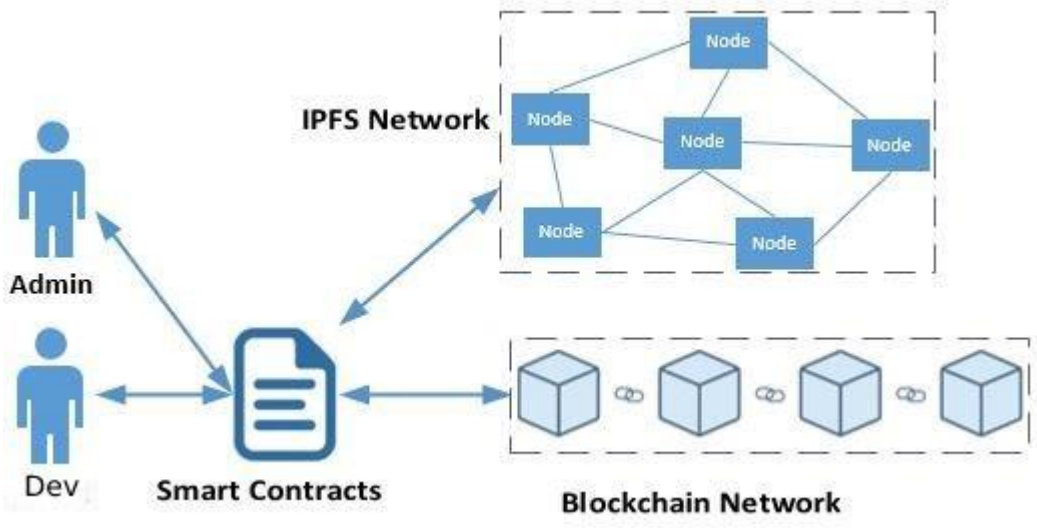

Fig. 1. GGANBUverse platform for decentralized web/mobile app hosting

Two main workflows are proposed for the GGANBUverse protocol. These are the IPFS Peer registration miner storage nodes and the web hosting service registration.

The workflow of registering Peer miner storage nodes through GGANBUverse for the IPFS network is composed of the following steps:

1. Send registration request: The peer sends a request to the admin for registering their device. The peer describes device information such as type of devices, IP address, total storage space, bandwidth.
2. Verify registration: The admin verifies the information related to the declaration and adds this information to a smart contract or GGANBUverse platform database (Still to be decided during implementation

3. Add the device to the IPFS network: The admin allows the registered device to connect the IPFS network and starts sending $GGANBU rewards every time storage is used for web or app hosting

To access the system, users first sign up on metamask and login with the registered credentials. Successful login takes users to the home screen for selecting the file to upload. The workflow of registering the web/mobile app hosting is composed of the following steps:

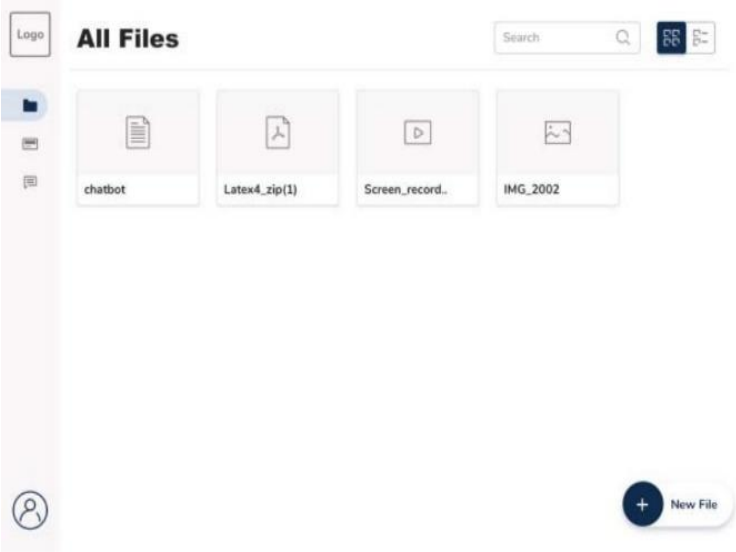

Fig. 2. GUI to upload files

1. Uploading of application files: User uploads the source codes of the website/app using the file picker. System checks for storage availability based on file size. The selected file is encrypted using AES 256 bit algorithm when sufficient storage is available. The system will then compute the total cost of hosting. Once the cost is the calculated system will check if the developer's $GGANBU balance wallet balance is more than the calculated cost. If the developer has sufficient balance then he/she is prompt to pay to store the files.
2. Encryption of file: The uploaded file is encrypted using AES 256 bit algorithm. The encryption key is generated using the developer's wallet address and randomly generated salt value. This encryption key along with an IV is used to encrypt the source code. This maintains the confidentiality of the application.
3. Uploading the webapp to the IPFS network: The system selects an arbitrary number of nodes on the IPFS network based on the computational resource requirements of the application and developer balance. It will then host this application on the selected node. The accounts of the Peer node are automatically rewarded with a percentage of hosting fees. For better performance the platform blacklists peers every time they are unavailable for data retrieval placing their storage offers as last options.
4. Return IPNS value: The selected nodes return the PeerID to the developer. The platform then stores this information on the Ethereum network through a smart contract and the client uses the PeerID for configuring the DNS.

5. Configure DNS: The developer buys a domain name from GGANBUverse with $GGANBU, and configures on the dashboard with the information as follows:
   <domain> I IN A <Ip_Address>
   //IP_Address is the IP address of the IPFS Gateway
   _dnslink <domain> IN TXT dnslink=/ipns/<PeerID>

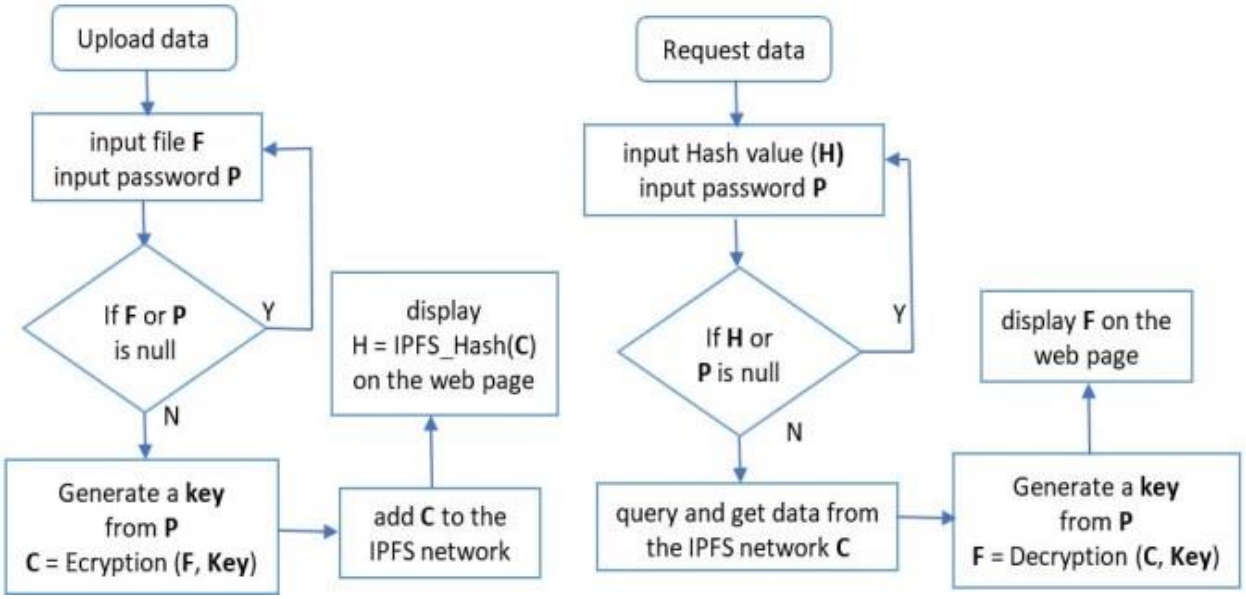

Fig. 3. The flowchart of steps for uploading data (a) and getting data (b) on the IPFS network.

CONCLUSIONS

This is a proposal for the GGANBUverse decentralized platform for web/mobile app hosting. The proposed system takes the advantages of the blockchain technology, IPFS, and encryption. That allows independent developers to host and have autonomy of their mobile and web applications without using any central system from service providers like play store or app store. Also, a combination model between IPFS , blockchain, and the protocol for managing decentralized web/app hosting service in the IPFS network has been proposed. This proposal also serves as a blueprint for a future decentralized cloud hosting layer built on the GGANBUverse platform enabling users to monetize their data.