STORAGE AND ORGANIZATION OF GEOSPATIAL DATA ON IPFS

AYUSH TAH 12000119129

SHIVANSHU SINGH 12000120131

PROBLEM STATEMENT

Ever-growing geospatial data needs secure and scalable storage for archiving data. Blockchain-based technologies can offer strong byzantine fault tolerance and distributed data storage and access. However, currently, there are no mature options for storing and organising large geospatial datasets on a distributed blockchain. Our aim is to create a software toolset for storage browsing and permission-based access to geospatial data using blockchain.

INTRODUCTION

Recently Blockchain or rather IPFS which is based or derives many principles from blockchain has been realized as a possible solution to the problems we face with geospatial data because of the tamper-proof, traceable, trust-free, transparent, and decentralized characteristics it possesses. IPFS achieves these de-centralized and security characteristics by integrating consensus mechanisms, asymmetric cryptographic algorithms, and distributed data storage to name a few. Hence, IPFS presents itself as a major player in finding a solution for the problem we are trying to tackle.

WHY STORING AND ORGANIZING LARGE GEOSPATIAL DATASET IS IMPORTANT?

- It is estimated that 100 TB of weather-related data is generated daily. This alone presents considerable storage and access problems for most organizations.
- Geospatial data typically involves large sets of spatial data gathered from many diverse sources in varying formats and can include variety of information.
- Since information from geospatial data are very useful, we need to figure out a way of how to store and organize large datasets, so that it can be easily accessed.

WHAT IS IPFS?

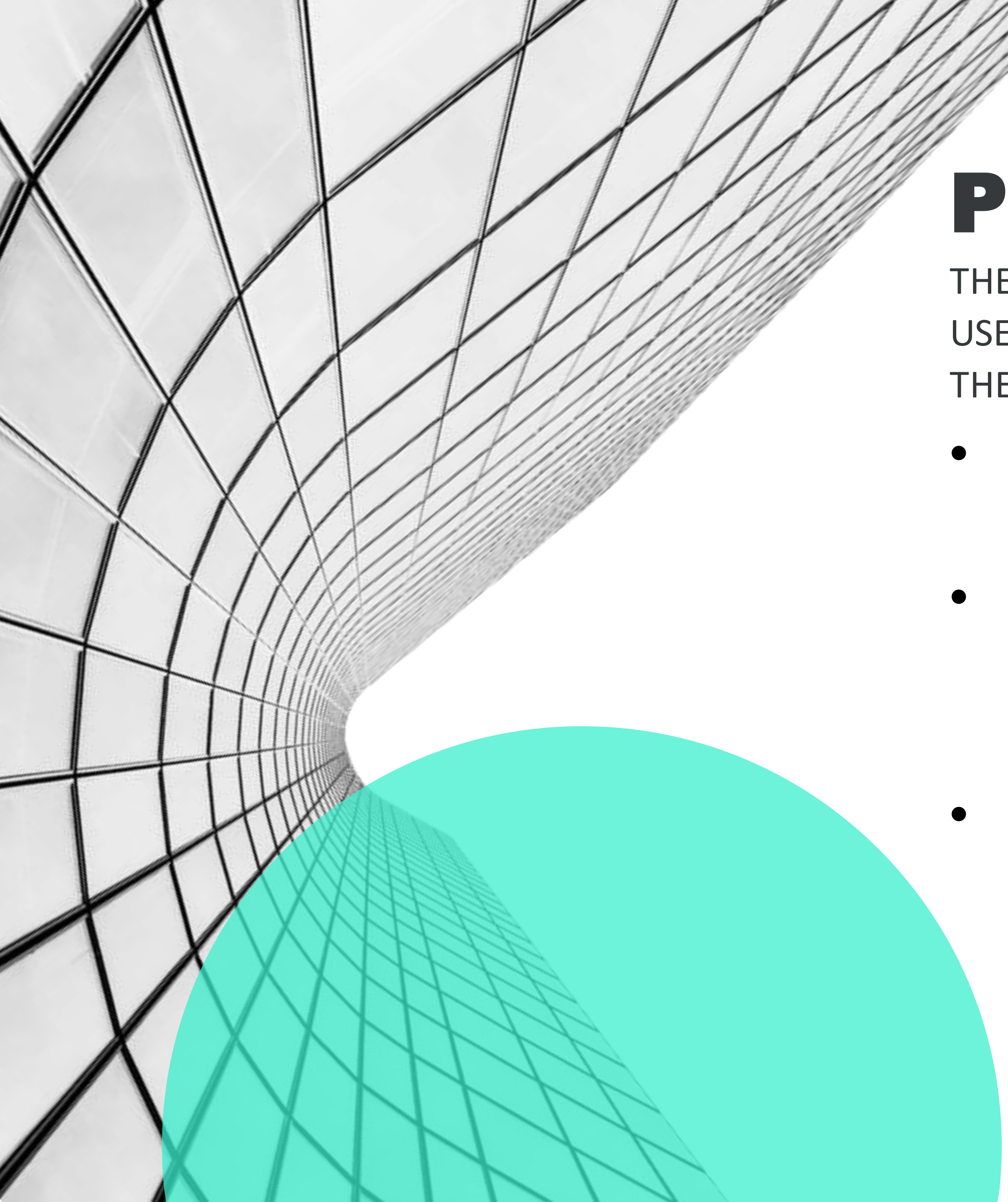
InterPlanetary File System or IPFS for short, is a protocol, hypermedia and file sharing peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting IPFS hosts. It borrows many features from blockchain such as SHA256 encryption and ledger system that makes a prime solution to the problem.

Literature Survey, Current work and Gaps

There are several studies that have been conducted in this domain, though most are generally geared towards the validation of data rather than the simple task of uploading and retention of data in IPFS.

•One study by J. R. Cedeno Jimenez, A. Folini, and M. A. Brovelli, pushed the idea of using IPFS to validate user-contributed geospatial data in the IPFS network. They use the Velas blockchain infrastructure and implement a combination of a Discrete Global Grid System(DGCS) with smart contracts. Two kinds of smart contracts got produced: cell and registry smart contracts. Their work and effort confirms the potential of creating decentralized apps for validating geographical data as a citizen science solution.

•Another review paper by Jesus Rodrigo Cedeno Jimenez, Pengxiang Zhao, Ali Mansourian, and Maria Antonia Brovelli focuses more on the crowdsourcing of geospatial data and two main platforms that were developed for it, namely FOAM and D-GIS. But they found out the platforms were just conceptualized not realized/ deployed.



PRIMARY GOAL

THE OBJECTIVE OF THE PROJECT IS TO BUILD A WEB APP THAT USES IPFS TO STORE DATA LOCALLY OR ON THE IPFS NETWORK. THE FEATURES WE ARE STRIVING TOWARDS ARE:

- TO BUILD A BETTER UNDERSTANDING OF FILE STORAGE, ESPECIALLY GEOSPATIAL DATA STORAGE ON A BLOCKCHAIN.
- TO BUILD A WEB APP TO DEMONSTRATE OUR PROJECT WITH A ROBUST AND EFFICIENTLY CODED BACKEND AND SIMPLE LOOKING AND EASY TO USE GUI ON THE FRONTEND.
- TO DEVELOP A MORE SECURE FILE STORAGE SYSTEM THAN THE EXISTING TRADITIONAL CENTRALISED DATA STORAGE.

TIMELINE/PLAN

MARCH

The front-end should be a lot more robust and the back-end will be modified to work with multiple files as well as world-wide sharing instead of a local network.

APRIL

By the end of April the User Authentication/Login functions should be somewhat complete and the necessary visual changes would be complete.

MAY

After the deployment of the previous step, we would work on polishing the app even more and ironing out the wrinkles.

WORK DONE TILL NOW

Currently, we are building the web app/web page for the project. We finished coding the uploading of files to our IPFS network. The GUI is currently under development. We are using JavaScript or more precisely Node.JS for the back end and HTML, CSS and JavaScript for the front end. We're working on a lot more features that we want to add, so we work even harder towards adding new features and mechanisms later down the road.

At present the backend is operational but right now only accepts one file at a time which we are working to be able to do as many as the user wants. There’s also a basic front-end with renaming, uploading and downloading features but this will also be totally reworked to be more robust and fluent.

The app can be made on any operating system, but we used ubuntu because of its reliability and general ease of installation of the various packages that we are going to use.
We are using WSL2 (Windows Subsystem for Linux) which is available for systems running Windows 10 ver. 1903 or above, because with its recent milestones, it is very reliable and easy to work with and as we will do most of our work in CLI we don’t need to do a standalone installation of ubuntu.

Our backbone of the project runs on IPFS (InterPlanetary File System). IPFS is a protocol, hypermedia and file-sharing peer-to-peer network for storing and sharing data in a distributed file system. It uses content-based addressing to uniquely identify each file in a global namespace connecting IPFS hosts. IPFS can replace the location-based hypermedia server protocols HTTP and HTTPS to distribute the World Wide Web.

HOW IT WORKS

Every file added to IPFS is given A unique address derived from A hash of the file's content. This address is called A content identifier (CID) and it combines the hash of the file and A unique identifier for the hash algorithm used into A single string. IPFS currently uses SHA-256 by default, which produces A 256 bit (32 byte) output. SHA-256 is also used by bitcoin.

Once a file is uploaded to a node, the file is broken to pieces. Based on the contents of each chunk of the file, each of them is assigned A CID, and the file is converted into A Merkle DAG (directed acyclic graph).Each node of the Merkle DAG consist of chunk of the file.

HOW IT WORKS

INITIALISING THE IPFS DAEMON

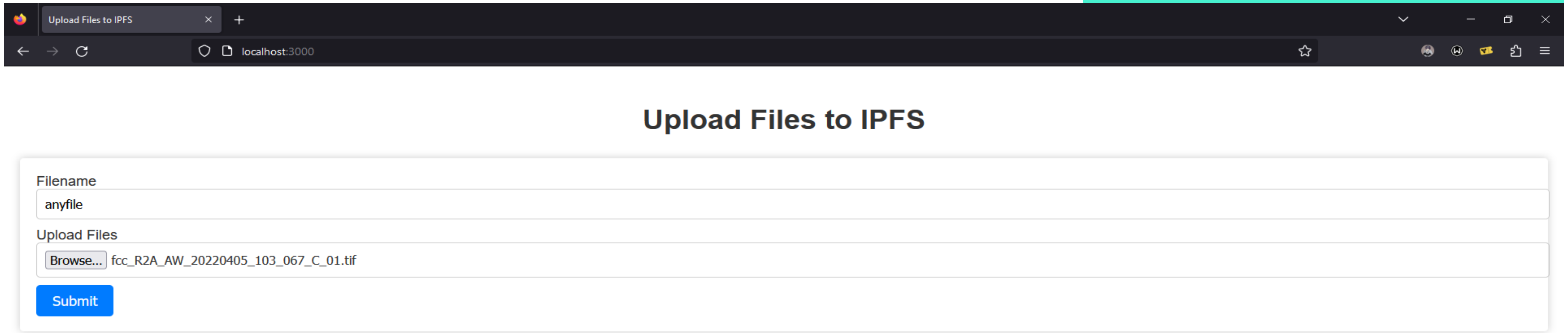
```
shiverbolt@Eureka:~$ ipfs daemon
Initializing daemon...
Kubo version: 0.18.1
Repo version: 13
System version: amd64/linux
Golang version: go1.19.1
```

INITIALISING THE SERVER

```
shiverbolt@Eureka:~/fyp$ node bcend.js
Server is listening on port 3000
```

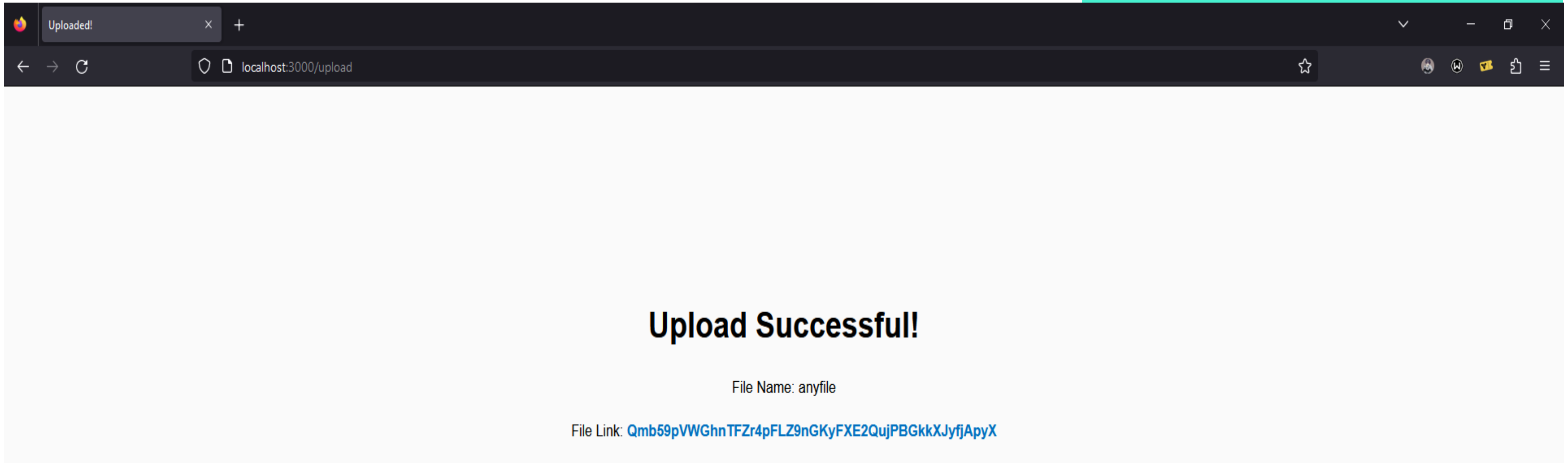
HOW IT WORKS

ACCESSING THE WEB APP ON LOCALHOST:



HOW IT WORKS

WHEN THE FILE IS SUCCESSFULLY UPLOADED:



FILE CAN BE IMMEDIATELY DOWNLOADED BY CLICKING THE LINK

HOW IT WORKS

WE CAN ALSO VIEW THE FRAGMENTS OF THE FILE USING:

```
shiverbolt@Eureka:~$ ipfs object get Qmb59pVWGhnTFZr4pFLZ9nGKyFXE2QujPBGkkXJyfjApyX | jq
{
  "Links": [
    {
      "Name": "",
      "Hash": "Qmb8YTiRTv9h6vTaf6Zr1NsqSzm12wvZa6w84q1fe5FngH",
      "Size": 262158
    },
    {
      "Name": "",
      "Hash": "QmNQrMAGE3SJJ3oSLNiTTndsaKFU6AytMR4uJiqQ4jQ3yH",
      "Size": 262158
    },
    {
      "Name": "",
      "Hash": "QmRRiTsWfoRAKfngmbpgU6NppjvmCEj7pej3VVxLAHRJQo",
      "Size": 262158
    },
    {
      "Name": "",
      "Hash": "QmXnbMduXThTSPpGqCRfk9mdcf14DvAPQCAcRJPiRsWxxW",
      "Size": 262158
    },
    {
      "Name": "",
      "Hash": "QmPXmbkAPQ8rkhHtG4PTUww2zX7dJhq2gCsuaqmeWzRhfo",
      "Size": 262158
    },
    {
```

SUMMARY

All in all I would like to think the tough part is over as the backend is working and we only have to work on the front-end a lot more but in reality the work is nowhere near over. The User authentication part will present more difficulties as well ease of use features that we are trying to implement but we will get it done.