

CS4037D CLOUD COMPUTING

Project Design



Submitted by:

Abin K Paul - B170217CS

Robin Ainikkal - B170169CS

Rahul Ram KK - B170776CS

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

Content

1. Problem Statement	3
2. Solution Design	3
3. Framework and Other Tools used	4
3.1 NodeJS	4
3.2 Truffle	5
3.3 Solidity	5
3.4 MetaMask	6
3.5 InterPlanetary File System (IPFS)	7
3.6 Web3	7
4. Methodology and Working	7
4.1 Flow Chart	8
4.2 Front end	9
4.3 Back End	9
4.4 Deployment of Portal	9
Conclusion	10

IPFS storage Based Government Documents

Storing Portal Using Blockchain

1. Problem Statement

A lot of documents are piled up in shelves of government offices which may be lost or damaged, so an alternative method is to store them digitally. This project is to create a medium where we can store those documents in a safe and secure environment.

2. Solution Design

Our motive is to store the documents in a safe and secure place digitally. We designed decentralized applications or dApps which allow the government staff to store the documents using a blockchain account for the transaction. Blockchain, a decentralized and distributed public ledger, can provide immutability, transparency, security, and auditability in storing the documents.

So the government staff can access the portal where they can upload a document by mentioning the document type and the document owner which then is stored into the Ethereum blockchain and the document itself is stored in IPFS which is commonly used a distributed file system to store

data. The hash of the document stored in IPFS is stored using the smart contract. So once uploaded the document is totally safe from tampering.

3. Framework and Other Tools used

3.1 NodeJS



NodeJS is an open-source, cross-platform, JavaScript runtime environment (Framework) that executes JavaScript code outside a web browser.

Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

3.2 Truffle

Truffle Suite is a development environment based on Ethereum Blockchain, used to develop dApps (Distributed Applications). Truffle is a one-stop solution for building dApps: Compiling Contracts, Deploying Contracts, Injecting it into a web app, Creating front-end for dApps and Testing.



3.3 Solidity

Solidity is a statically-typed programming language designed for developing smart contracts that run on the EVM. Solidity is compiled to bytecode that is executable on the EVM. With Solidity, developers are able to write applications that implement self-enforcing business logic embodied in smart contracts, leaving a non-repudiable and authoritative record of transactions.



3.4 MetaMask

MetaMask is a browser extension that lets you run dApps without being part of the Ethereum network as an Ethereum Node. Instead, it lets you connect to another Ethereum Node called INFURA and run smart contracts on that Node. MetaMask manages your Ethereum wallet, which contains your Ethers, and allows you to send and receive Ethers through a dApp of interest.



3.5 InterPlanetary File System (IPFS)



IPFS stands for Interplanetary File System. At its core it is a versioned file system which can store files and track versions over time, very much like Git. It also defines how files move across a network, making it a distributed

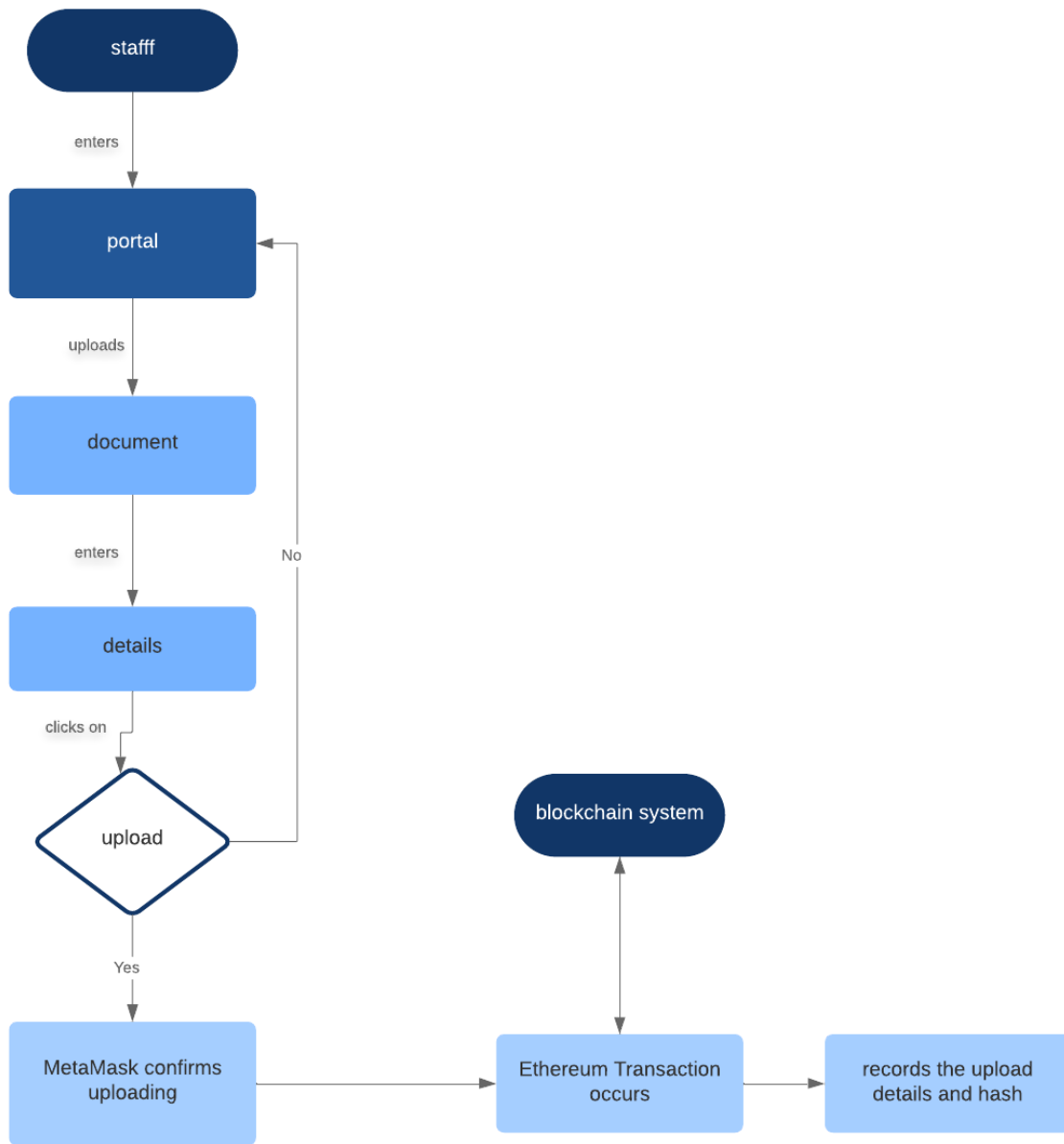
file system, much like BitTorrent. In combining these two properties, IPFS enables a new permanent web and augments the way we use existing internet protocols like HTTP.

3.6 Web3

The web3 JavaScript library interacts with the Ethereum blockchain. It can retrieve user accounts, send transactions, interact with smart contracts, and more. An analogy is that instead of using a jQuery to read and write data from a web server, you can use Web3.js to read and write to The Ethereum Blockchain.

4. Methodology and Working

4.1 Flow Chart



4.2 Front end

The staff can upload the document in the upload section of the portal and can provide the details of the documents. The staff can view all the documents uploaded and the details of the documents already uploaded. In the documents details section there would be IPFS hash info of the document and the transaction details.

The front end was done on React, Redux and Bootstrap 4

4.3 Back End

Once the staff clicks on the upload button the hash of the uploaded document and the details of the document is stored in blockchain. Web3 provides an interface to interact with the deployed smart contract.

Metamask provides a wallet to manage transactions directly from the browser. After the transaction is done the data cannot be altered by any individual.

When the staff tries to get details of uploaded documents the hash of the document and the transaction details of the document is fetched making use of the corresponding function in the smart contract.

4.4 Deployment of Portal

In chrome

- Connect MetaMask to one of the Ganache Ethereum accounts
- Import a new account and use the account seed phrase provided by Ganache from the terminal where u run ganache-cli.

In the terminal:

Install Truffle Framework and Ganache CLI globally.

- npm install -g truffle

- npm install -g ganache-cli

Run the development blockchain.

- ganache-cli

Open another terminal

- cd (file location of the portal)
- npm install

Compile and migrate the smart contracts.

- truffle compile
- truffle migrate

Type the command

- npm run start

Conclusion

In today's world everything is becoming digital as it is more reliable and efficient so why not store documents digitally? It will save lots of time and space and also it would be easy to maintain. So when we store a document, we want it to be totally safe and secure and the document should be tamper-proof.

Blockchain happens to be the safest place to store data as it cannot be altered once stored. So what we needed to do is somehow store the document in a blockchain, but you cannot store the document directly because of the storage constraint of blockchain. So what we had to do is store the documents in IPFS storage system which is a great for storing and sharing data in distributed file system and we store the hash of the file in blockchain. So basically we made a portal to store document in a blockchain in which there will be no tampering of data at the back end side and also highly reliable.