Devault (graduation project)

A Blockchain-based end-to-end encrypted cloud storage

Participants:
Abd El-Twab M. Fakhry
Hossam A. Eissa

Supervisor:
Dr. Abdurrahman Nasr

Al-Azhar University
Faculty of Engineering
Computers & Systems Engineering Department

July 23, 2022

Table of contents

Introduction

What is Web 1.0?

Basically, this first version of the web was designed to help people better find information. This web version dealt was dedicated to users searching for data. This web version is sometimes called *"the read-only Web"* because it lacks the necessary forms, visuals, controls, and interactivity we enjoy on today's Internet.

- No user-to-server communication.
- Static websites.
- Hyper-linking and bookmarking pages.
- Read-only Web.

Active 1989–2005

What is Web 2.0?

If Web 1.0 was made up of a small number of people generating content for a larger audience, then Web 2.0 is many people creating even more content for a growing audience. Web 1.0 focused on reading; Web 2.0 focused on participating and contributing.

- Functions such as online documents, video streaming, etc.
- Cloud computing operations
- Centralized data.
- Read and Write Web.

Active 1999-2012

What is Web 3.0?

Web 3.0, which is also referred to as Web3, is built on a foundation consisting of the core ideas of decentralization, openness, and more excellent user utility. Web 1.0 is the "read-only Web," Web 2.0 is the "participative social Web," and Web 3.0 is the "read, write, execute Web."

- The Internet of Things (IoT).
- Semantic searches.
- Decentralized processes.
- Read, Write, and Control Web.

Active 2006-ongoing

What is wrong with the current approach to store data?

- ✗ `Censorship`
  As the internet currently works on a centralized model, it is susceptible to censorship. However, this is an issue that can be easily mitigated with decentralization.
- ✗ `Data Hack`
  It's not recommended to store your sensitive data on a centralized server that is financially profitable to get hacked.
- ✗ `Data Loss`
  Of course, you can always stick with local storage, But once they are lost, stolen, or most likely encrypted by ransomware, you cannot make a recovery.

What is wrong with the current approach to store data?

- ✗ `Censorship`
  As the internet currently works on a centralized model, it is susceptible to censorship. However, this is an issue that can be easily mitigated with decentralization..
- ✗ `Data Hack`
  It's not recommended to store your sensitive data on a centralized server that is financially profitable to get hacked.
- ✗ `Data Loss`
  Of course, you can always stick with local storage, But once they are lost, stolen, or most likely encrypted by ransomware, you cannot make a recovery.

What is wrong with the current approach to store data?

- ❌ `Censorship`
  As the internet currently works on a centralized model, it is susceptible to censorship. However, this is an issue that can be easily mitigated with decentralization..
- ❌ `Data Hack`
  It's not recommended to store your sensitive data on a centralized server that is financially profitable to get hacked.
- ❌ `Data Loss`
  Of course, you can always stick with local storage, But once they are lost, stolen, or most likely encrypted by ransomware, you cannot make a recovery.

What is wrong with the current approach to store data?

- ✖ `Censorship`
  As the internet currently works on a centralized model, it is susceptible to censorship. However, this is an issue that can be easily mitigated with decentralization..
- ✖ `Data Hack`
  It's not recommended to store your sensitive data on a centralized server that is financially profitable to get hacked.
- ✖ `Data Loss`
  Of course, you can always stick with local storage, But once they are lost, stolen, or most likely encrypted by ransomware, you cannot make a recovery.

Proposed Solution

The **solution** we propose for such a problem is to use:

- A distributed database system that will store data in a peer-to-peer network where is no central authority with the right to modify or censor clients' data.
- Encryption, so that everything should be encrypted before being uploaded.
- Diffusion, so that each object is shredded into small chunks. And object chunks are stored on different Nodes around the globe.
- A Blockchain and smart contract for identity without a central authority. Verification of data that cannot be faked or changed. Combine this with encryption, data ownership, and replication, and that's what true decentralization means for applications.

Proposed Solution

The **solution** we propose for such a problem is to use:

- ✅ A distributed database system that will store data in a peer-to-peer network where is no central authority with the right to modify or censor clients' data.
- ✅ Encryption, so that everything should be encrypted before being uploaded.
- ✅ Diffusion, so that each object is shredded into small chunks. And object chunks are stored on different Nodes around the globe.
- ✅ A Blockchain and smart contract for identity without a central authority. Verification of data that cannot be faked or changed. Combine this with encryption, data ownership, and replication, and that's what true decentralization means for applications.

Proposed Solution

The **solution** we propose for such a problem is to use:

- ✅ A distributed database system that will store data in a peer-to-peer network where is no central authority with the right to modify or censor clients' data.
- ✅ Encryption, so that everything should be encrypted before being uploaded.
- ✅ Diffusion, so that each object is shredded into small chunks. And object chunks are stored on different Nodes around the globe.
- ✅ A Blockchain and smart contract for identity without a central authority. Verification of data that cannot be faked or changed. Combine this with encryption, data ownership, and replication, and that's what true decentralization means for applications.

Proposed Solution

The **solution** we propose for such a problem is to use:

- ✅ A distributed database system that will store data in a peer-to-peer network where is no central authority with the right to modify or censor clients' data.
- ✅ Encryption, so that everything should be encrypted before being uploaded.
- ✅ Diffusion, so that each object is shredded into small chunks. And object chunks are stored on different Nodes around the globe.
- ✅ A Blockchain and smart contract for identity without a central authority. Verification of data that cannot be faked or changed. Combine this with encryption, data ownership, and replication, and that's what true decentralization means for applications.

Proposed Solution

The **solution** we propose for such a problem is to use:

- ✅ A distributed database system that will store data in a peer-to-peer network where is no central authority with the right to modify or censor clients' data.
- ✅ Encryption, so that everything should be encrypted before being uploaded.
- ✅ Diffusion, so that each object is shredded into small chunks. And object chunks are stored on different Nodes around the globe.
- ✅ A Blockchain and smart contract for identity without a central authority. Verification of data that cannot be faked or changed. Combine this with encryption, data ownership, and replication, and that's what true decentralization means for applications.

Concepts and terminology

What is a blockchain?

A **blockchain** is a public database that is updated and shared across many computers in a network.

Block refers to data and state being stored in consecutive groups known as **blocks**. Think of it as a Git commit.

Chain refers to the fact that each block cryptographically references its parent. In other words, blocks get chained together. The data in a block cannot change without changing all subsequent blocks, which would require the consensus of the entire network. Think of it as a Git history.

What is ethereum?

Ethereum is a platform powered by blockchain technology that is best known for its native cryptocurrency, called ether, or **ETH**, or simply ethereum. The distributed nature of blockchain technology is what makes the **Ethereum** platform secure.

The Ethereum platform supports ether in addition to a network of **decentralized apps**, otherwise known as **dApps**. Smart contracts, which originated on the Ethereum platform, are a central component of how the platform operates. Many applications use smart contracts in conjunction with blockchain technology.

As a cryptocurrency, Ethereum is second in market value only to Bitcoin as of January 2022.

What is ethereum?

Ethereum is a platform powered by blockchain technology that is best known for its native cryptocurrency, called ether, or **ETH**, or simply ethereum. The distributed nature of blockchain technology is what makes the **Ethereum** platform secure.

The Ethereum platform supports ether in addition to a network of **decentralized apps**, otherwise known as **dApps**. Smart contracts, which originated on the Ethereum platform, are a central component of how the platform operates. Many applications use smart contracts in conjunction with blockchain technology.

As a cryptocurrency, Ethereum is second in market value only to Bitcoin as of January 2022.

What is ethereum?

Ethereum is a platform powered by blockchain technology that is best known for its native cryptocurrency, called ether, or **ETH**, or simply ethereum. The distributed nature of blockchain technology is what makes the **Ethereum** platform secure.

The Ethereum platform supports ether in addition to a network of **decentralized apps**, otherwise known as **dApps**. Smart contracts, which originated on the Ethereum platform, are a central component of how the platform operates. Many applications use smart contracts in conjunction with blockchain technology.

As a cryptocurrency, Ethereum is second in market value only to Bitcoin as of January 2022.

ETH, EVM, and Smart contract

- `ETH?`
  The native cryptocurrency of Ethereum. Users pay ether to other users to have their code execution requests fulfilled.
- `EVM?`
  The Ethereum Virtual Machine is the global virtual computer. Any participant can request the execution of arbitrary code on the EVM; code execution changes the state of the EVM.
- `Smart contracts?`
  A smart contract is code that lives on the Ethereum blockchain. Once smart contracts are deployed on the network you can't change them. Dapps can be decentralized because they are controlled by the logic written into the contract, not an individual or company. This also means you need to design your contracts very carefully.

ETH, EVM, and Smart contract

- **ETH?**
  The native cryptocurrency of Ethereum. Users pay ether to other users to have their code execution requests fulfilled.
- **EVM?**
  The Ethereum Virtual Machine is the global virtual computer. Any participant can request the execution of arbitrary code on the EVM; code execution changes the state of the EVM.
- **Smart contracts?**
  A smart contract is code that lives on the Ethereum blockchain. Once smart contracts are deployed on the network you can't change them. Dapps can be decentralized because they are controlled by the logic written into the contract, not an individual or company. This also means you need to design your contracts very carefully.

ETH, EVM, and Smart contract

- `ETH?`
  The native cryptocurrency of Ethereum. Users pay ether to other users to have their code execution requests fulfilled.
- `EVM?`
  The Ethereum Virtual Machine is the global virtual computer. Any participant can request the execution of arbitrary code on the EVM; code execution changes the state of the EVM.
- `Smart contracts?`
  A smart contract is code that lives on the Ethereum blockchain. Once smart contracts are deployed on the network you can't change them. Dapps can be decentralized because they are controlled by the logic written into the contract, not an individual or company. This also means you need to design your contracts very carefully.

What is dapps?

Decentralized applications, or dApps, are software programs that have their backend code running on a distributed computer network. This is in sharp contrast to standard apps which typically run on centralized servers.
A dapp can have frontend code and user interfaces written in any language (just like an app) to make calls to its backend. Furthermore, its frontend can get hosted on decentralized storage such as IPFS.

What is the IPFS?

`IPFS`, The Interplanetary File System is a distributed system for storing and accessing files, applications, and websites. It is a worldwide peer-to-peer file-sharing system created by Protocol Labs.

A dApp is entirely open source. By way of its open-source nature, changes to the protocol must be decided via consensus of its network users.

Wallets, accounts, and addresses

It's worth understanding the differences between some key terms.

- An Ethereum account is an entity that can send transactions and has a balance.
- An Ethereum account has an Ethereum address, like an inbox has an email address. You can use this to send funds to an account.
- A wallet is a product that lets you manage your Ethereum account. It allows you to view your account balance, send transactions, and more.

  Most wallet products will let you generate an Ethereum account. So you don't need one before you download a wallet.

Web3 vs Web2

Web2 refers to the version of the internet most of us know today. An internet dominated by companies that provide services in exchange for your personal data. Web3, in the context of Ethereum, refers to decentralized apps that run on the blockchain. These are apps that allow anyone to participate without monetising their personal data.

Table 1: Practical comparisons

Web2	Web3
Twitter can censor any account or tweet	Web3 tweets would be uncensorable because control is decentralized
Payment service may decide to not allow payments for certain types of work	Web3 payment apps require no personal data and can't prevent payments
Servers for gig-economy apps could go down and affect worker income	Web3 servers can't go down – they use Ethereum, a decentralized network of 1000s of computers as their backend

Methodology

File processing

Our dApp will take a file as input from a user and upload it to the IPFS by invoking an Ethereum contract. The hash of the file will be stored on Ethereum.

This is the process we'll go through:

1. Take file as an input
2. Convert file to buffer
3. Read key used for encryption/decryption
4. Encrypt file using AES-256-cbc block cipher.
5. Split file into small chunks
6. Upload encrypted chunks to IPFS
7. Store hash of file returned by IPFS
8. Get user's Metamask Ethereum address
9. User confirms transaction to Ethereum via Metamask
10. IPFS hash is written on Ethereum

Development Methodology

Software Development Approach

We have chosen the Scrum methodology. It's a popular way to implement agile, and it allows the team to deliver software regularly

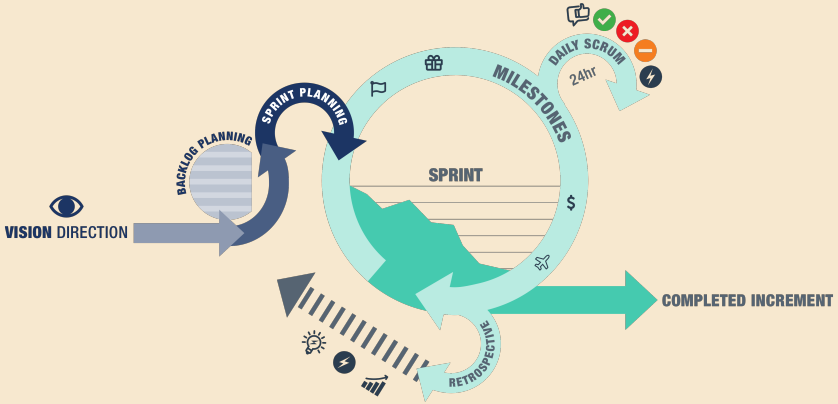

Figure 1: Scrum Methodology

Tools and Technologies

- Node.js
- Solidity (Smart contract)
- Github Actions (CI/CD)
- Next.js (React.js framework)
- Hardhat (Solidity framework)
- Docker (Deployment)
- Ethers (Library)
- Infura (IPFS gateway)

Diagrams

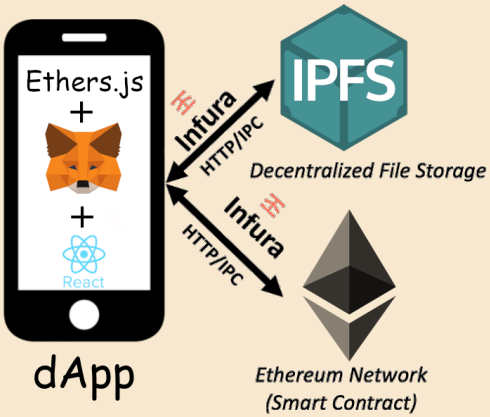

Figure 2: dApp Architecture

Project Diagram

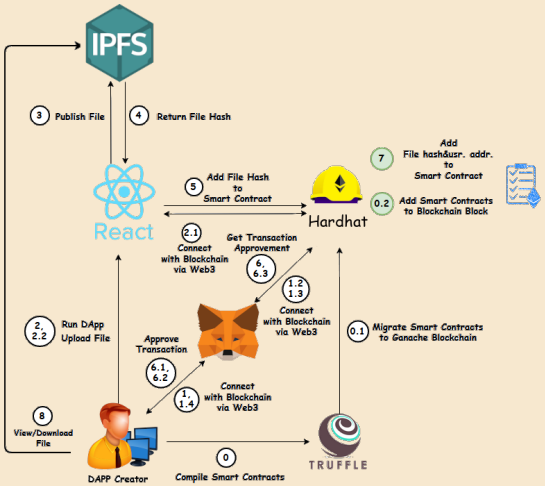

Figure 3: Project Diagram

Thanks!

References i