Functional Specification

Student 1: Aaron Cleary (19495324)

Student 2: Joao Pereira (19354106)

Project Title: Fileflo

Date Completed: 14/11/22

Table of Contents

1. Introduction

1.1 Overview

Fileflo is a secure, decentralised file-sharing application that allows users to securely upload files on the blockchain through our web application,
which can then be shared with and downloaded by other Fileflo users.

Fileflo offers several advantages over typical file-sharing applications. Since Fileflo is decentralised, there is no central entity that is in control of user data, allowing Fileflo to offer a true guarantee of privacy. Furthermore, Fileflo uses asymmetric encryption [1] to protect user files by ensuring that only certain users can access files. Files uploaded through our application are stored in a distributed file system, meaning there is no single point of failure and that files are retrieved in the most efficient way possible. Fileflo's incorporation of blockchain technology means that files are immutable and cannot be tampered with.

The Fileflo architecture consists of three main components; a decentralised application (dApp), IPFS (InterPlanetary File System) [1] file storage and a serverless AWS backend.

Users will interact with our web application, which is hosted on AWS, in order to upload their own encrypted files or download files from other users. The files are encrypted before being published on IPFS, which is a peer-to-peer hypermedia protocol and distributed file system. Every file uploaded to IPFS is associated with a unique hash. This unique hash is then stored in a smart contract on a private Ethereum blockchain.

1.2 Business Context

Fileflo provides many advanced and modern features that can be utilised by several fields and institutions around the world. Any company, institution or individual who seeks decentralisation and encryption with regard to storing and transferring sensitive data can use Fileflo to solve their issues.

The potential market Fileflo could obtain is enormous. All industries, especially medical and research institutions, tackle sensitive data such as medical and research records every day and with the use of Fileflo, we will be able to ensure that their documentation/data cannot be tampered with and that it is inaccessible to any unsafe third parties. By using Fileflo, many organisations can eliminate the risk of having their documentation/files stolen, tampered with or accessed by unauthorised personnel.

1.3 Glossary

- **Blockchain** - A shared, immutable digital database that consists of a growing list of records, called 'blocks', which are duplicated and distributed across the entire network of computer systems on the blockchain.
- **Decentralised** - The transfer of control and decision-making from a centralised entity (individual or organisation) to a distributed network.
- **Ethereum** - A decentralised global software platform powered by blockchain technology.
- **Smart contract** - A self-executing computer program that digitally facilitates the terms of an agreement.
- **Solidity** - An object-oriented, high-level programming language that is used to implement smart contracts.
- **Truffle** - An Ethereum-based framework used to compile smart contracts and deploy them on our private Ethereum blockchain.
- **Web3.js** - An Ethereum JavaScript API that is needed to interact with the Ethereum network.
- **Asymmetric encryption** - The encryption and decryption of data using two separate yet mathematically connected cryptographic keys.
- **Public key** - A key used to encrypt data.
- **Private key** - A key used to decrypt data.
- **Seed phrase** - A cluster of randomly generated words, which is used as a form of user authentication.

- **dApp** - Stands for ‘decentralised app’, which is an application that is powered by blockchain technology and is not controlled by a central entity.
- **IPFS** - Stands for ‘InterPlanetary File System’, which is a distributed system for storing and accessing files.
- **Node** - An IPFS node is a program that runs on a computer and can exchange data with other IPFS nodes.
- **Hash** - A unique 256-bit string generated by IPFS, which refers to a specific file.
- **AWS** - Stands for ‘Amazon Web Services’, which is a cloud platform that offers a variety of services, several of which we will be using for our project backend.
- **JIRA** - A software application designed for issue tracking and project management.

2. General Description

2.1 Product / System Functions

Fileflo is a web application with a simple functionality architecture consisting of a user-friendly interface to ensure all users are capable of using and understanding the purpose of the application. Upon opening Fileflo for the first time, users will have the ability to create an account using information such as a Username, Email, Company, Password and a regulated Seed Phrase. Once logged in, Fileflo will generate a profile with the mentioned parameters apart from the Password and Seed Phrase. A public and private key will also be created.

A navbar comprising three pages; upload, download and profile. This navbar will be displayed at the top of the website. On the Upload page, users are able to upload files where the encryption, IPFS and blockchain process takes place to ensure the file, once uploaded, is in safe hands. Once a file is uploaded, users can see information in regard to that upload such as the history, size and hash. The download page gives users the chance to download other users' files/documents as long as they have access to their personal private key. Finally, the profile page simply displays the user's information along with an option to view their private key. The private key is only accessible by inputting the user’s unique seed phrase, which was initially generated when the user first registered for Fileflo. Finally, users are capable of logging out and logging back in without the worry of their information or files being lost.

2.2 User Characteristics and Objectives

Our primary target audience will be organisations, institutions and individuals who seek the ability to store files and documents in a decentralised and secure environment. Such organisations include medical clinics, universities or research institutions. These organisations have a need to transfer confidential patient data for example, so Fileflo would be of tremendous utility to

them. With this in mind, we will design the Fileflo application to be intuitive to users of all digital literacy levels, which will enable our application to be used by the widest variety of organisations as possible. Ultimately however, any individual or organisation can take advantage of Fileflo for their own benefit to ensure the security of their files.

With regards to the digital literacy of potential users, any organisation or individual will be able to use Fileflo as we aim to provide an easy-to-use and user-friendly environment so that anyone feels at ease when using Fileflo. The process and architecture behind Fileflo itself is rather complex, and therefore converting that complexity to simplicity so that anyone can use the application is our biggest goal. With this in mind, not only will research clinics or medical centres benefit, but also your day-to-day user who wants the extended security that FIleflo offers for their personal files. All of our users will be able to enjoy the advantages Fileflo provides without being daunted by any preconceived notions of complexity regarding a blockchain dApp.

2.3 Operational Scenarios

2.3.1 User Registers for Fileflo

Upon visiting the Fileflo website, the user can choose to either sign in or register for an account. In order to register, the user will be required to enter their name, email address and company. After completing email verification, the user can then choose a username and password. This user information will be stored in AWS Amplify and Cognito and will never be used for any other purpose other than user authentication.

2.3.2 User Uploads a File

When a user wishes to either store a file or send a file to another Fileflo user, they will navigate to the upload page, which is one of the three main pages in our application. The user can then upload a file by selecting it through the file explorer or dragging it into the window. If the user wants to store the file, they will select the storage option and the file will automatically be encrypted with their public key before being uploaded to the blockchain. On the other hand, if the user wishes to send a file to another Fileflo user then they will select the share option and encrypt the file with the public key of the user they wish to send it to. The file will then be uploaded to IPFS and its unique hash will be stored on the blockchain.

2.3.3 User Searches for Uploaded File

Once a user has uploaded a file, they will have the ability to search through all of their uploaded files. The user can scroll through a list of their uploaded files on the upload page. There will also be a search bar, where users can search by filename, file type or file hash. The metadata for each file will be retrieved from the blockchain before being rendered under the search input.

2.3.4 User Downloads a File

When a user wants to download a file, they will open the download page, which is another of the three main pages in the Fileflo application. There will be a download tab where the user can input the unique IPFS hash of the file they wish to download, before then entering their private key to decrypt the file. After doing so, a call will be initiated to IPFS, which then allows the file to be downloaded onto the user's local machine.

2.3.5 User Saves a Contact

If a user wishes to send a file to another user, they will be required to encrypt the file with the other user's public key. This process could become tedious, which is why users can save the public keys of other users under a 'contact'. After inputting an unrecognised public key on the upload page, Fileflo will ask the user if they wish to save this public key under a name. This will then allow the user to have the option to automatically add the public key of certain users in the future.

2.3.6 User Views Their Profile

When a user wishes to view or edit their profile, they will navigate to the profile page, which is the last of the three main pages in the Fileflo application. The profile page will display the user's name, email address, username and contacts. The user can edit any of these details by intuitively selecting the edit option. There will also be a password-protected tab which contains the user's public key and private key, which are immutable.

2.4 Constraints

Below is a list of constraints that may possess a sense of risk or damage to completing and releasing Fileflo however with each point we have highlighted how these issues are resolved.

2.4.1 Data Protection

One of the biggest challenges with any application that deals with users and sensitive data/information are the risks of breaching Data Protection. Any individual who uses Fileflo as intended risks information such as private keys, file information as well as transfer details between parties being shared. To overcome this constraint, Fileflo will

not be storing any of the mentioned data. File information and the transfer of files will all be stored within IPFS, blockchain and from the user's local storage.

2.4.2 Scalability

Fileflo's release will be kept minimal in terms of user performance. This regards the number of users we will be able to have at once and respectively how many files can be uploaded/downloaded at once within the application. This is an issue that arises from Fileflo being hosted on a cost-less server however to ensure this does not become a problem, we are only going to test Fileflo with a small number of users as well as limited uploads/downloads being processed at once. In any case, if Fileflo was to proceed on a larger scale and community outreach, the possibility to upgrade the servers and maintenance would not come as a concern.

Another scalability constraint is the private Ethereum blockchain's limitation. A private blockchain can only possess a few nodes whereas a public [5] blockchain provides access to a lot more nodes. If Fileflo was to notice a huge upsurge in usage this would come as a constraint as we only allow access to a few nodes however if it was the case of an increase in popularity, we will develop our own public blockchain with an enhancement on possible nodes to be distributed between the users.

2.4.3 Speed

Considering that two of the biggest processes Fileflo undergoes are uploading and downloading files, we want to guarantee our user's that both of these processes are as speedy as they can be. We plan to minimise this constraint by producing time-efficient uploading and downloading programs/algorithms with the use of the correct programming language to further reduce the time both processes take.

2.4.4 Time

All of the aforementioned constraints lead to another constraint, which is time. To ensure the composure and efficiency of Fileflo's launch, we have taken crucial steps to minimise this risk as much as possible. We have planned a timetable as well as dedicated JIRA stories in which significant portions of the project will be completed throughout each story to ensure we are sticking within deadlines. The use of stories will ensure efficiency in producing tasks within a certain timeframe so that as the weeks come closer, Fileflo also comes closer to being finished.

3. Functional Requirements

3.1 User Registration

Description	Allows the user to register for the Fileflo application. The user will be required to provide their name and email address in

	order to register.
Criticality	Highly Critical - The user must be able to register to use the application.
Technical Issues	N/A
Dependencies With Other Requirements	N/A

3.2 User Login

Description	Allows the user to log in to the Fileflo application using the username and password they chose during registration.
Criticality	Highly Critical - The user must be able to log in to use the application.
Technical Issues	N/A
Dependencies With Other Requirements	User Registration

3.3 User File Upload

Description	Enables the user to upload a file to the blockchain through Fileflo. The user must first select a file to upload, before then encrypting it with a public key.
Criticality	Highly Critical - File storage/sharing requires the user to have the ability to upload files.
Technical Issues	There will likely be speed issues if the user uploads a massive file. Furthermore, any files not accessed after 6 months risk being deleted by IPFS.
Dependencies With Other Requirements	User Registration, User Login

3.4 User File Download

Description	Allows the user to download a file that has been shared with them by another user. The user must input the hash of this file before decrypting it using their private key.
Criticality	Highly Critical - File sharing requires the ability to download the files that have been shared with the user.
Technical Issues	There will likely be speed issues if the user downloads a massive file.
Dependencies With Other Requirements	User Registration, User Login, User File Upload

3.5 User File Search

Description	Enables the user to search through either the list of files they have uploaded or downloaded. The user should be able to search using the file hash, filename or file type.
Criticality	Important - Searching through uploaded/downloaded files is a feature that will provide real utility to users.
Technical Issues	N/A
Dependencies With Other Requirements	User Registration, User Login, User File Upload, User File Download

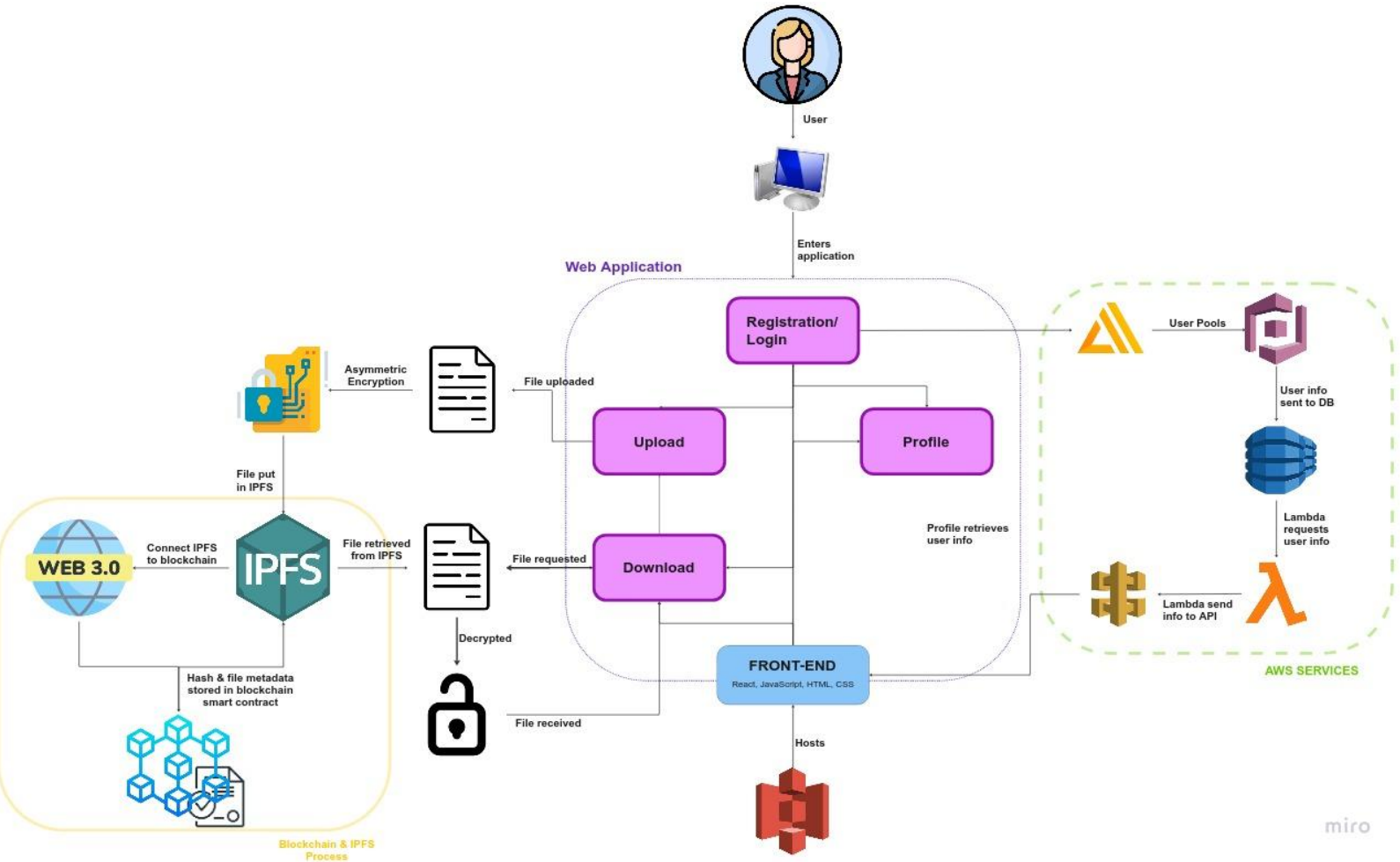
3.6 User Profile

Description	A profile that contains the user's information, including their name, email address, username, public key, private key and contacts. The user should be able to edit this information, save for their public key and private key.
Criticality	Important - The user should be able to view and edit the information in their profile.
Technical Issues	The user may forget the seed phrase that was generated when they first registered for Fileflo. The seed phrase is needed to view their private key.
Dependencies With Other Requirements	User Registration, User Login

3.7 User Contacts

Description	Enables the user to save the public key given to them by another user under a contact name. After adding a contact, the user can autofill the public key of this contact when they want to send a file to them.
Criticality	Moderate - This is a quality-of-life feature that will make it easier for users to share files with others.
Technical Issues	N/A
Dependencies With Other Requirements	User Registration, User Login

4. System Architecture



The above system architecture diagram illustrates the relationships between each component of Fileflo. Upon opening the Fileflo web application, the user must first either register or log in, which is a process that is handled by AWS services. Firstly, Fileflo uses the AWS Amplify framework to manage the backend of our application and interact with other AWS services [4]. We use AWS Cognito user pools for user authentication [3], while user login credentials are stored in a DynamoDB database. API Gateway is used to invoke different Lambda functions based on the various API calls made by the application. Finally, the app is deployed on Amazon S3 cloud storage.

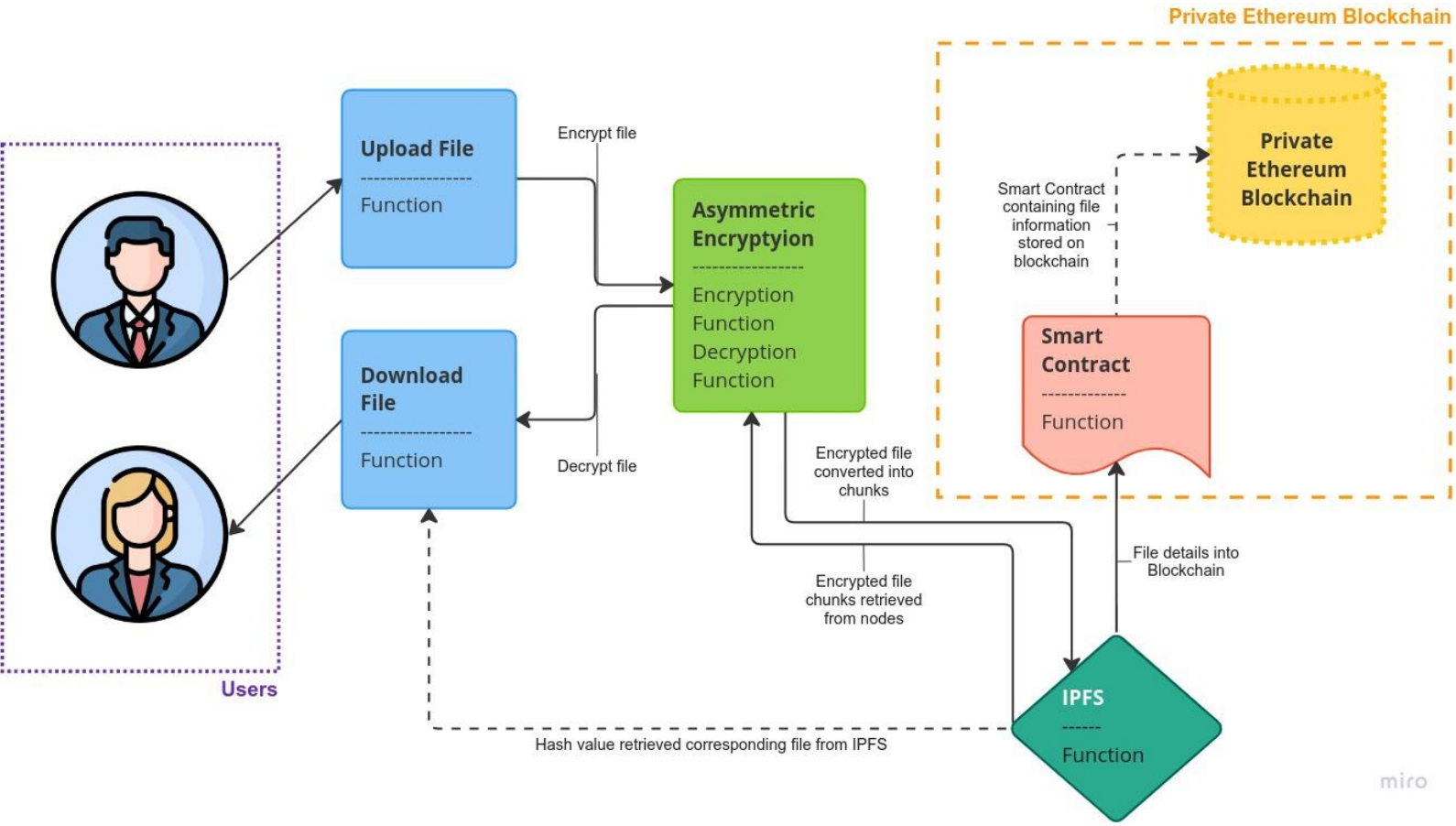
After logging in to the Fileflo application, there will be three main pages for users to navigate to; the upload page, download page and profile page. When a user wants to upload a file, their motive will be to either share the file with another user or to store the file for their own personal use. If the user wishes to share the file, they will encrypt the file with the public key of the user they wish to send it to. Otherwise, if they wish to store the file then the user will encrypt it with their own public key. All files that are uploaded through Fileflo will be sent to IPFS, which generates a unique hash for the file before splitting it into chunks and distributing them across a

network. This IPFS hash is stored in an Ethereum smart contract along with other file metadata, before then being deployed to our private Ethereum blockchain. These smart contracts will be created using the Solidity [2] programming language, the Truffle Ethereum framework and the JavaScript API Web3.js.

When a user wishes to download a file from another user, they will input the IPFS hash of their desired file. Fileflo will then initiate a call to both IPFS and the Ethereum blockchain, which returns the corresponding smart contract containing the file metadata. The file itself will then be retrieved from IPFS, which can then be viewed or downloaded by the user, provided they have the private key needed to access the file.

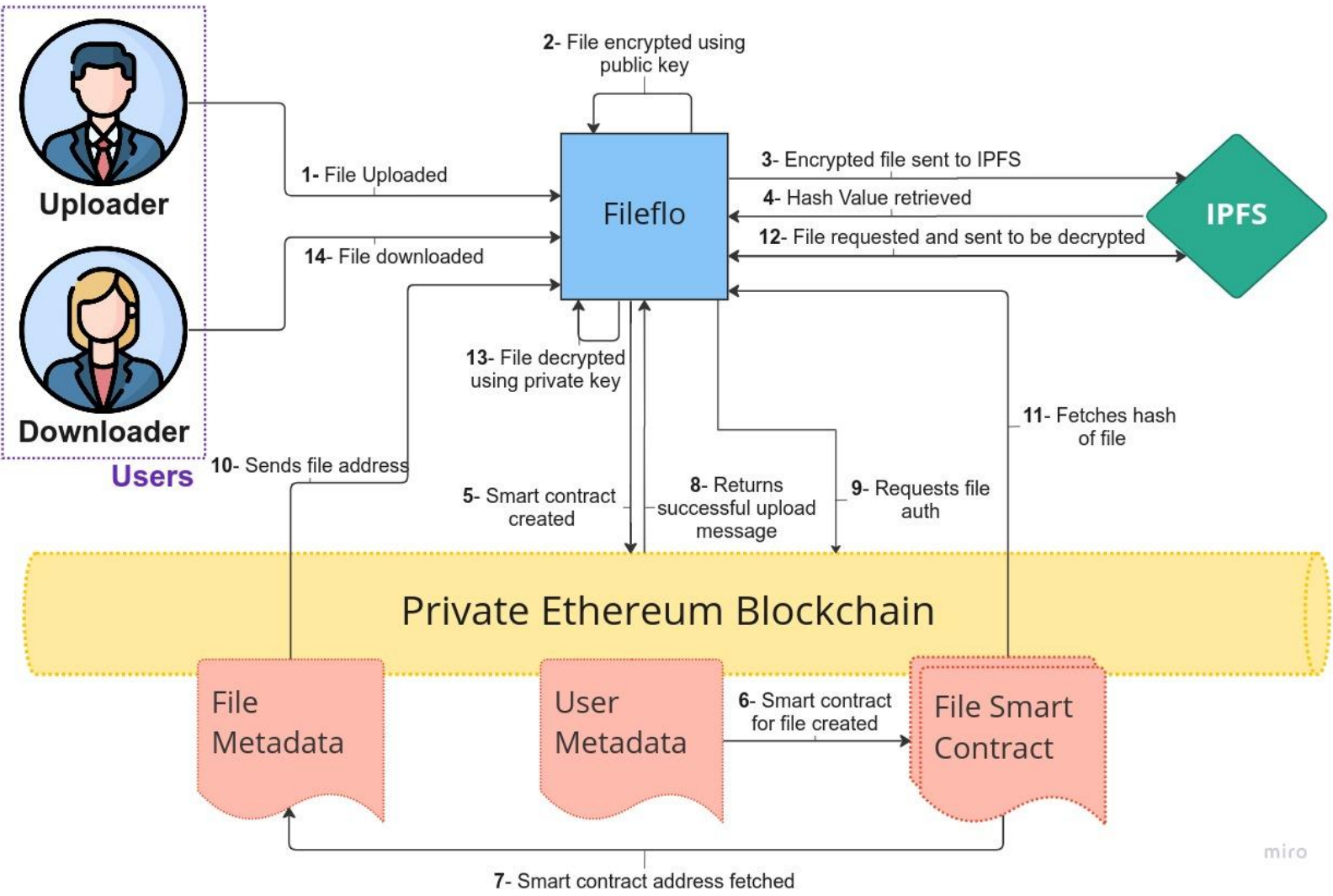
5. High-Level Design

5.1 Fileflo Data Flow Diagram



Displayed above is Fileflo's Data Flow diagram which represents how users can accomplish the process of uploading and downloading files. A file which has been uploaded will pass through an encryption method for it to then be converted into chunks into IPFS. IPFS then shares the file details with the Private Block Ethereum system. This system consists of the Blockchain alongside a smart contract. The smart contract is responsible for storing relevant and sensitive data such as file and user info. This entire blockchain system ensures the immutability of the entire file transfer system. On the other hand, IPFS then provides a hash value that can be accessed by permitted users wishing to download a file.

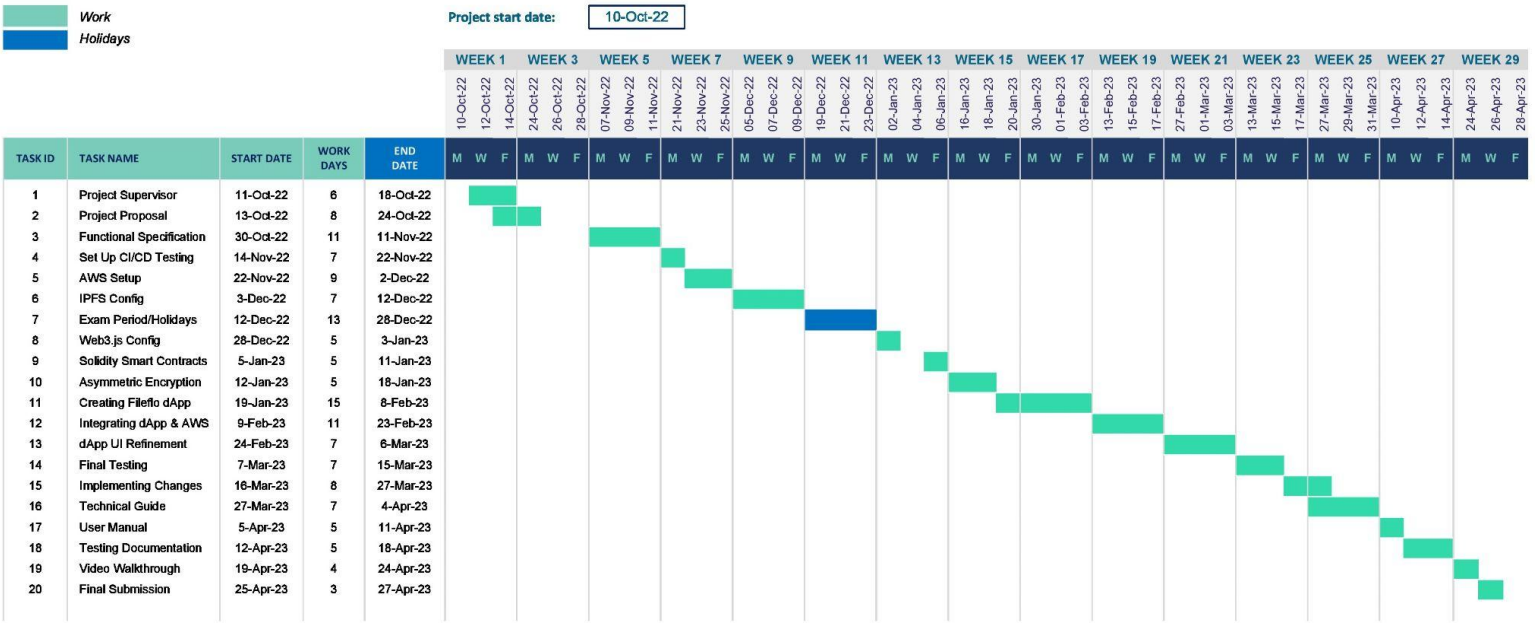
5.2 Fileflo System Interaction Diagram for Sharing Files



The interaction diagram for sharing files demonstrates the entire travel or steps a file makes before reaching the downloading user. The system contains two users, one who uploads a file and the other who downloads the file, as well as major components like the Private Ethereum Blockchain, IPFS as well as the Fileflo application itself. Files pass through Fileflo where the encryption takes place. Files then are transferred to IPFS where a hash value is retrieved. Furthermore, the Blockchain architecture gathers the file from IPFS and a smart contract is generated as well as extending components such as User Metadata and File Metadata to categorically organise the relevant information. Finally, the blockchain returns a successful message once all the previous steps are completed and it sends a file address to Fileflo which is thereby accessible by users with authorization.

6. Preliminary Schedule

6.1 Gantt Chart



TASK ID	TASK NAME	START DATE	WORK DAYS	END DATE
1	Project Supervisor	11-Oct-22	6	18-Oct-22
2	Project Proposal	13-Oct-22	8	24-Oct-22
3	Functional Specification	30-Oct-22	11	11-Nov-22
4	Set Up CI/CD Testing	14-Nov-22	7	22-Nov-22
5	AWS Setup	22-Nov-22	9	2-Dec-22
6	IPFS Config	3-Dec-22	7	12-Dec-22
7	Exam Period/Holidays	12-Dec-22	13	28-Dec-22
8	Web3.js Config	28-Dec-22	5	3-Jan-23
9	Solidity Smart Contracts	5-Jan-23	5	11-Jan-23
10	Asymmetric Encryption	12-Jan-23	5	18-Jan-23
11	Creating Filefio dApp	19-Jan-23	15	8-Feb-23
12	Integrating dApp & AWS	9-Feb-23	11	23-Feb-23
13	dApp UI Refinement	24-Feb-23	7	6-Mar-23
14	Final Testing	7-Mar-23	7	15-Mar-23
15	Implementing Changes	16-Mar-23	8	27-Mar-23
16	Technical Guide	27-Mar-23	7	4-Apr-23
17	User Manual	5-Apr-23	5	11-Apr-23
18	Testing Documentation	12-Apr-23	5	18-Apr-23
19	Video Walkthrough	19-Apr-23	4	24-Apr-23
20	Final Submission	25-Apr-23	3	27-Apr-23

7. Appendices

[1] IPFS
https://mycoralhealth.medium.com/learn-to-securely-share-files-on-the-blockchain-with-ipfs-219ee47df54c

[2] Solidity Web3
https://www.dappuniversity.com/articles/how-to-build-a-blockchain-app

[3] AWS Authentication
https://blog.logrocket.com/authentication-react-aws-amplify-cognito/

[4] AWS Architecture
https://labs.sogeti.com/aws-architecture-design-startup-use-case/

[5] Private vs Private Blockchain
https://www.investopedia.com/news/public-private-permissioned-blockchains-compared/#:~:text=In%20a%20public%20blockchain%2C%20anyone,delete%20entries%20on%20the%20blockchain.

[6] Asymmetric encryption
https://www.okta.com/identity-101/asymmetric-encryption/