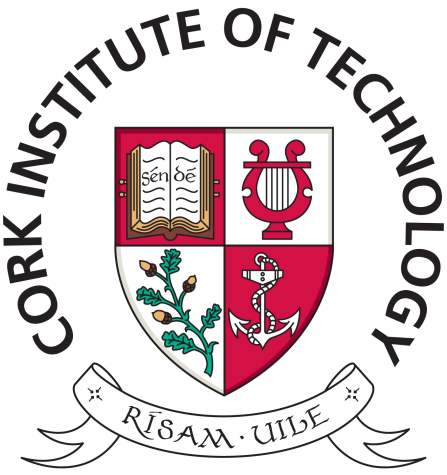


# Blockchain: automated insurance risk analysis and data validation

by

Jack Hickey

This thesis has been submitted in partial fulfillment for the degree of Bachelor of Science in Software Development

in the
Faculty of Engineering and Science
Department of Computer Science

May 2019

# Declaration of Authorship

I, Jack Hickey , declare that this thesis titled, 'User Data as a Commodity: Decentralising User Data Privacy' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an undergraduate degree at Cork Institute of Technology.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Cork Institute of Technology or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

CORK INSTITUTE OF TECHNOLOGY

# *Abstract*

Faculty of Engineering and Science
Department of Computer Science

Bachelor of Science

by Jack Hickey

Blockchain has made headlines for its use as a cryptocurrency, most notably with Bitcoin; however, academics and industry alike are researching its other potentials in diverse sectors.

This project proposes the use of smart contracts and blockchains immutability of records to perform risk analysis of potential policyholders in a more efficient, cost-effective, and secure manner than conventional insurance models.

Blockchain-based risk analysis can automate these tasks as well as use the immutability of the blockchain to get more valuable data. The proposed solution uses Hyperledger Fabric, a permissioned blockchain network. The application allows policyholders to retain ownership of their data and claims. As the policyholder changes insurers, policyholders carry their data with them to new insurers, allowing for the smart risk analysis to make more accurate assumptions. As time proceeds, insurers are thus enabled to make smarter offers based on this assessment. Hyperledger Fabric allows policyholders to retain privacy with regards to their data, and for insurers to offer contracts using the blockchain network that are only visible between the policyholder and the insurer.

The project ultimately determines that smart contracts can indeed automate and increase the efficiency of how risk analysis is performed for the insurance industry. The adoption of blockchain technology within the insurance industry will revolutionise the way we do insurance.

# *Acknowledgements*

**Oonagh O'Brien** for being the project advisor during the research phase and providing continuous insight and support throughout.

**Ruairí O'Reilly** who acted as supervisor during the implementation phase where he assisted with narrowing the focus of the project and reinventing the objectives.

**Stephanie Hua** for proofreading the thesis and providing insightful feedback on the writing style.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **IPFS** | **I**nter **P**lanetary **F**ile **S**ystem |
| **CAP** | **C**onsistency **A**vailability **P**artitioning |
| **DApp** | **D**ecentralised **A**pplication |
| **P2P** | **P**eer-To-**P**eer |
| **MVP** | **M**inimum **V**iable **P**roduct |
| **TDD** | **T**est **D**riven **D**evelopment |
| **ECC** | **E**lliptic **C**urve **C**ryptography |
| **POW** | **P**roof **O**f **W**ork |
| **PI** | **P**ersonal **I**nformation |
| **MFS** | **M**utable **F**ile **S**ystem |
| **REST** | **R**epresentational **S**tate **T**ransfer |
| **API** | **A**pplication **P**rogramming **I**nterface |
| **CRUD** | **C**reate **R**ead **U**pdate **D**elete |

# Chapter 1

# Introduction

## 1.1 Motivation

The motivation behind this project is to make use of the latest technology for web applications for the next generation of the Internet and create a user oriented web application. Social media has changed the way we socialise and allowed us to connect with people in ways that were unimaginable before. In the third quarter of 2018 Facebook had 2.27 billion monthly users[1]. Social media sites have become a major part of our daily lives and have achieved this in a relatively short space of time. Social media allows anyone to receive updates on everything around the world and keep connecting with friends and family no matter where they are.

However the internet in its current form is centralised with a few key players generating and controlling the majority of data online. These have become known as the internet giants with Amazon, Alphabet Inc., Facebook and Alibaba at the forefront[2]. In fact for the first time ever the amount of people the U.S. with a social media platform dropped in 2018[3]. The reasons for this may be due to concern for general mental health to feeling that it invades their privacy too much.

The amount of data gathered on users and the value is greater than ever, and this data does allow users to access many services for free however at the cost of user privacy and little control over online identities. This data is all stored centrally leaving it more exposed to hackers and mishandling which we saw earlier last year in the Facebook-Cambridge Analytica case[4].

Blockchain technology has no central authority and has shown to be extremely resilient to hacking and misuse. Many companies are already looking at its wide variety of uses and one of these is create the 3rd evolution of the internet. By using blockchain in a

blogging application that gathers user data it is possible to return ownership of users online identities and the content they create back to the rightful owners who are the users. Blockchain tokens can also be used to reward users for their site contributions and data sharing, generating an online economy and valuing the user.

## 1.2 Executive Summary

The internet is centralised and relies heavily on the gathering and monetisation of user data for providing web applications free to users. The amount and types of data social media applications gather on their users allow them to build up very accurate profiles for targeted advertising and analytics purposes which funds these companies and allows free access for users. This can result in better and more personalised websites but often leaves the user with little control over their personal data.

This project has two core objectives: Returning ownership of content and data to the natural owners and designing a reward system for users contributions to the site. By valuing the users contributions to the site it can result in wealth distribution and a better online experience for all.

Combining traditional social media with blockchain technology we can resolve some of the issues that the author feels are present today which are a singular controlling authority over data and a lack of recognition for content creators contributions.

## 1.3 Contribution

The target of this project is to address the misuse of data and the growing privacy concerns that users have with the internet. By returning control over data and content to the users there will be stronger sense of trust between web applications and users.

The objectives are to return control of user data to the rightful owners and remunerate them for their data that is provided. It is also important to recognise the content creation by users as this attracts people to the site and to remunerate these users also for creating content. The ultimate goal though is to ensure that users are in full control over their data and content by removing a central authority that stores and has access to the data.

## 1.4 Structure of This Document

This thesis is broken down into the following 10 chapters.

### 1: Introduction

This chapter introduces the problem being addressed and builds up a simple view over what this project will be and the issue being addressed.

### 2: Background

This chapter looks in detail at what the project will entail. It breaks down and describes how the various technologies being employed will play a role in the final project and discusses why this is being used.

It uses current state of the art research as a comparison to help build an idea of what tools and technologies are best for developing this product as well as to form a discussion as to why this project is necessary.

### 3: User Data as a Commodity! Decentralising the Online Profile

The problem being addressed is broken down further and the requirements for developing the final product are discussed.

### 4: Implementation Approach

This looks at how the final product will be achieved and details the overall architecture that will be used. It looks at use cases, development methodologies, risk analysis and a detailed breakdown of what must be implemented for each use case.

### 5: Evaluation

The evaluation chapter reviews the research phase and looks towards the implementation. It takes a critical view of the proposed project to see if it is still feasible. It ultimately concludes it was not and proposes a new project.

### 6: Related Work

This section takes into account the review conducted into the previous chapter and looks at current state of the art relating to the newly proposed project.

### 7: Proposed Solution

The proposed solution is discussed in detail here outlining use cases, architecture, goals and the technologies that will be employed.

### 8: Implementation

This chapter reflects on the implementation of the project and reviews issue encountered throughout the development.

### 9: Testing and Evaluation

The testing metrics that were used to evaluate if the project was successful and functional are discussed in this chapter.

### 10: Discussion and Conclusions

This is an overall review of the project. It looks at what conclusions can be drawn for this technological contribution and looks at future developments given the knowledge obtained.

# Chapter 2

# Background

## 2.1 Thematic Area within Computer Science

This section looks at breaking down the project into its core areas to get a better understanding of it from a high level overview. This will allow us to take a step back and review the technological requirements for the project by examining current state of the art research into related fields. This will also open a discussion as to why a project like this is necessary and will define the objectives more clearly.

The objective of this project is to develop a blogging application with the core features of web3 instilled in it's implementation. While there is still no concrete definition of what a web3 application is there has been common reoccurring features in these applications that allow themselves to be distinguished from web2 applications and these features are decentralisation, a lack of a central authority, and incentivised. While these features do appear repeatedly in many definitions of what web3 will be many also feel that anything that contributes to a new more user oriented Internet could be a web3 application.

This will be accomplished in this project by making the blog fully decentralised, giving users an area in which they have full control and ownership of their data and content, remunerating users who choose to share this data for analytics and advertising purposes, and by allowing users to support other blogs using tokens. Making this system as relatable to what users know while introducing them to a new form of blogging application and way to interact with the Internet is one goal that will be used as a measure of success for this project. Some web3 applications currently require users to pay a fee known as gas on Ethereum for writing to the blockchain network and this may be a one of the reasons why web3 applications are still inaccessible to most people as they may not wish to pay for the use of a online service while there is other options that are free.

By remunerating users for their online data this cost can be diminished as the payment provided could be partially used to pay this fee.

With all this in mind it is safe to say that the concrete area this project falls under is a web application. The blogging application is the front-end that user will interact with in a similar manner to a traditional site, where they will also be able to monitor their data. The backend of this application will opt to use blockchain and a form of decentralised storage system as opposed to a web2 applications that would use a a centralised and sometimes distributed server system to provide the application. The implementation of this project will help open accessibility to blockchain technology to a wider audience than previously as little knowledge of the system will be required to access it.The benefits of opening up this technology to a wider audience would be that users could become more conscious of their online interactions and have a better understand of the true value of their data.

## 2.2 Project Scope

Here will break down the projects key areas within computer science, giving a high level overview of the technologies and how they are related and will be used within the scope of this project. We look at how these technologies work and the role they will ultimately play in the implementation.

### Web 3.0 Application

The term web3 was first coined in 2006 by John Markoff of the New York Times[5] where he depicted a Internet with a layer of meaning on top what already existed making it more than just a catalogue of data but also a guide for users. This has since evolved beyond that to our current features of web3 which see it take a nostalgic turn towards the original vision of the Internet with the addition of it being more human and privacy oriented. What is it meant by this is that web3 would be a P2P network similar to what the Internet was originally[6].

Although the term was first coined back in 2006, it is only now that it is becoming a reality as the tools and technologies become available. This is in a large part thanks to Bitcoin which provided the means to decentralisation and managed to solve the problem of double spending that had not been possible up till this point[7].

Decentralisation is at the core of what web3 is now which means that data is not stored in any centralised location. This increases security but also privacy as no central authority

controls all the data. In essence each user owns their own data. Blockchain adds accountability to this by being fully transparent allowing anyone to look up who owns what[7].

The main player in web3 applications is Ethereum. Ethereum is a blockchain designed to host decentralised applications or DApps as they are more commonly known[8]. DApp is just another term to describe an application that runs on blockchain however they do not necessarily incorporate the values previously outlined of web3.

The design of these applications on the client side is the same as traditional web applications that we have become so familiar with and are implemented using languages such as JavaScript, Bootstrap, CSS, and so on however the backend is where it differs as there is no server side per se. In replacement of your traditional server you have a decentralised system typically, and as per the case of this application, implemented using the Ethereum blockchain.

It must also be noted though that Ethereum is by no means a storage solution but can be used to store small amounts of data, and the smaller the better as this means less cost to the user as to write data to a blockchain costs a fee. The fee is used to pay miners, a term used for a person(s) that validates a transaction is real. This means that in conjunction with the blockchain there must also be another tool to provide storage, ideally a decentralised storage tool.

### Backend Functionality

The backend of this application is what makes it differ from traditional websites. This section will put into context what role the backend of this system will have in the application. As mentioned previously this application will use both blockchain and decentralised storage to build this system.

The decentralised storage system will be used for the storage of user data, blogs, and the hosting of the client side code itself. This system will have to be secure if it is to store user data and so a form of cryptography will also have to be applied to ensure the security of this. There will be a requirement for references to the location to find this data also so that it can be retrieved easily by the user and those with authorised access. These references could be in the form of a unique data hash or IP address.

The blockchain will store these references to the data and blogs so that they can be accessed. Blockchain is a tool that can be publicly queried which will make it very useful for accessing users blogs and proving content ownership however it will also necessitate the security of user data even further. The blockchain is what will allow users to connect

on the site and glue all the functionality together. The other unique feature that will be used of blockchain is cryptocurrency to pay users for their data and allow them to exchange tokens to support other bloggers.

## Blockchain

Blockchain plays a core role in this project and is essentially the backbone. Blockchain has in recent years seen much hype in the public eye for the fortunes made and lost through Bitcoin, however there have been many alt coins developed which are not just a way for someone to invest in for profit. In fact this is not what Bitcoin was created for either but it solved the issue of reversibility of payments [7].

While Bitcoin is not the blockchain tool that will be used in this project as it serves a different purpose it is important to note it's contribution to the area of blockchain and decentralisation as it has paved the way for solving many other issues using it's methodology. Some feel that Bitcoin itself will not last as a payments system but will however be remembered for it's contribution to the field.

In essence blockchain allows data to be distributed and the transaction transparent but not copied as it is publicly known who owns what in a blockchain network. It is a complete record of transactions that do not have to be strictly financial but can be transactions of anything. Figure 2.1 provides a high level overview of how blockchain functions.

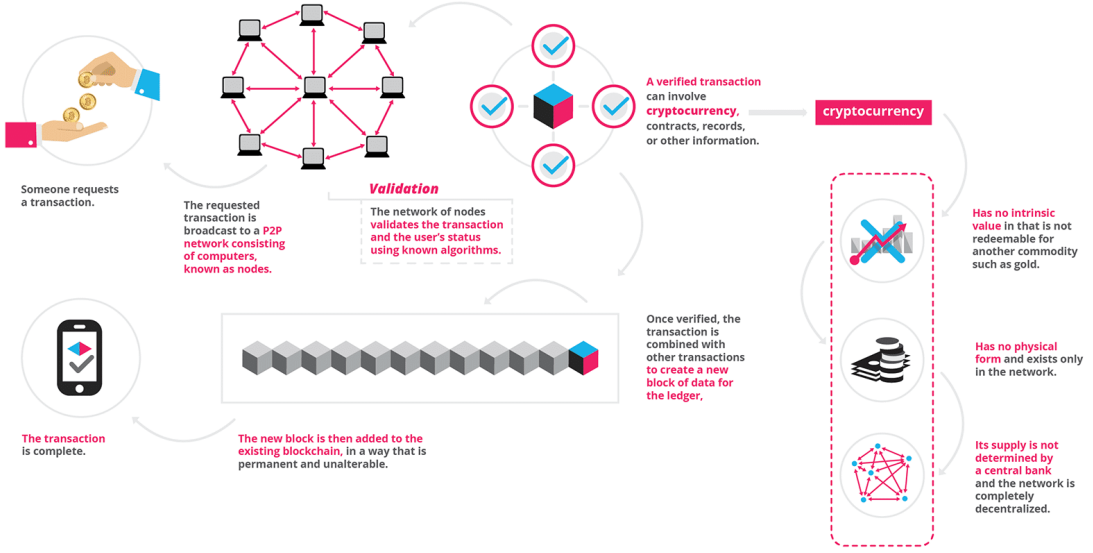


FIGURE 2.1: High Level Overview of Blockchain Technology[9]

The technology since it's debut has been used in a wide variety of applications ranging from voting systems to decentralised autonomous organisations. The use this project is

most focused on though is it's impact on the way we use the Internet. Ethereum is at the forefront of blockchains being used for the decentralised web with some other smaller competitors such as EOS and Tron being the most notable.

Ethereum is designed for the development of decentralised applications using it's development language known as solidity[8]. Solidity is used for the development of something called smart contracts which are simply just a collection of functions used for interacting with the blockchain.

Ethereum uses a system known as proof of work to validate all transactions that occur on the network, which is also the system used by Bitcoin. In simple terms this means when a transaction occurs it is placed inside a block and then miners verify the transactions within each block are legitimate by solving a complex mathematical puzzle known as proof-of-work. The first miner to solve each problem announces this to the entire network and is given a cryptocurrency reward. This block is then added to the chain.

This is quite an intense operation and slow with roughly the ability to handle around 5 transactions a second [10] meaning the system will have issues with scaling. However Ethereum has announced a new protocol known as Casper which uses a system known as proof of stake.

Unlike proof of work where the miners are rewarded with a cryptocurrency for solving mathematical problems, the creator of a new block is chosen in a deterministic way, depending on the wealth, and the age associated with this wealth which is known as stake. Instead of being rewarded they take a transaction fee. This system is currently used by EOS and allows for a lot more transactions to occur every second and it scales significantly better.

This is important to know for the development of this application as the number of transactions that will occur on the chain will determine the overall performance of the web application and so must be considered in how it's designed and which blockchain technology is chosen. Users have become accustomed to the speed of our current websites, and so if the technology chosen to develop the blog application on proves slower than traditional sites it will be unattractive to users.

### Decentralised Storage

Decentralised storage is the similar to blockchain and also has no central authority that controls all the data. This method of storing data will allow for a higher level of privacy and potentially is more secure as it does not give attackers a single point of failure.

Some of the most recognisable decentralised storage systems at current are Storj, Swarm, and IPFS. Storj is built on the Ethereum network and offers a decentralised cloud storage system. They have launched a test network that developers can contribute to however it is still in it's infancy.

Swarm is another protocol that is part of Ethereum's vision for the decentralised web. This product is still very much in development at current however there is strong progress being made towards it's completion.

Out of these IPFS is probably the most advanced and seeming as it is much further along in development at the time of writing this document it would make sense that this is used for the decentralised storage. It has aims to replace HTTP as the next protocol of the web, instead of location based addressing it actually addresses the content that user is looking for by creating a unique hash for it[11]. This would be ideal for this project as the hash could be stored on the blockchain for the location of blogs and user data. The disadvantage of this system though are that files cannot be updated easily once posted to the system and they also cannot be removed because of the immutability of the system. IPFS have however created a tool known as the "Mutable File System" that makes the management of editing and updating files much simpler and uses a technique that is comparable to git versioning. While this is far from ideal for this project, the other solutions available are far less ideal and outside of developing a decentralised storage solution there is not much choice.

The reason you cannot remove data is because it is not currently possible to remove a file from another node once it has been uploaded. This means that data lives on forever in this system. Many argue that has benefits as the system cannot be censored by anyone and so allows for stronger freedom of information however it can't be dismissed that the system has potential privacy issues.

### User Ownership

The overall objective of this application is to design a user oriented application that gives back to the users. This is achieved by giving users full ownership of their data, rewarding users for their contributions, and addressing censorship issues.

This application will allow free expression of users opinions and ideas which we have seen become an issue in places like Turkey[12] and China[13] where governments control what users have access to. This will be possible because of IPFS and the immutability of blockchain.

The decentralisation of data ensures that data is not only secure from hackers but also from singular authorities leaving the user in full possession and control over their data. Blockchain introduces accountability and keeps a record of all transactions that occur on the application. Users are rewarded using cryptocurrency for their data contributions and they have the option to support content creators using cryptocurrency also which allows these users to decide the value of a blog.

Connecting users using this new form of technology allows for an overall more user oriented experience and removes the interference of a central authority in the application. This means that what users see and want to see is not manipulated by an algorithm based on their assumed interests.

## 2.3 A Review of Using Blockchain to Protect Data

It is important to identify work related to this area as a means for comparison. This will help with identifying what has worked successfully and what flaws there are in similar preexisting systems. Reviewing these will give a better understanding of the technologies that could be employed here and how to use them.

- The top International Conferences and Journals.
  - Journal: IPFS - Content Addressed, Versioned, P2P File System; by Juan Benet;
  - Journal: Decentralizing Privacy: Using Blockchain to Protect Personal Data; by Guy Zyskind, Oz Nathan, and Alex 'Sandy' Pentland; published in 2015 IEEE Security and Privacy Workshops
  - Journal: Open Peer-to-Peer Systems over Blockchain and IPFS: an Agent Oriented Framework; by Antonio Tenorio-Forns, Samer Hassan, and Juan Pavn; published in Proceedings of the 1st Workshop on Cryptocur is a tokenized Profile specification standard that enables the creation and use of decentralized pseudo-identitierencies and Blockchains for Distributed Systems
  - Journal: The interplay between decentralization and privacy: The case of blockchain technologies; by Primavera De Filippi; published in Journal of Peer Production, Issue n.7: Alternative Internets
  - Conference: Web3 Summit: Berlin; A conference for developers focused on P2P protocols, Platform neutral computation language, Data distribution protocols, Blockchains, Transient data/messaging, Encrypted storage, Protocol-extensible developer APIs.

- The top companies/organisations
  - VETRI: VETRI is a blockchain-enabled personal data management platform with the aim to empower individuals to take part in the rapidly growing data economy.
  - Steemit: Steemit is a social media platform where everyone gets paid for creating and curating content.
  - Mastodon: Mastodon is a decentralised, open source social network that has similarities to Twitter
  - EID: EID is a tokenized Profile specification standard that enables the creation and use of decentralised pseudo-identities
  - SocialX: SocialX is a community driven social media platform built on blockchain that rewards content creators.
- The top wiki/forums/blogs/Youtube channels
  - YouTube: Dapp University; This YouTube channel provides tutorials on building decentralised applications with Ethereum.
  - Blog: Hackernoon; This is a blogging site that posts a lot of great information and tutorials around building decentralised applications on top of Ethereum.
  - Blog: Medium; This is another great site for blogs relating to all areas, and has a lot of great blogs relating to web development, blockchain, and IPFS which provide a great learning resource as well as detailed analysis.
  - Forum: Stack Overflow; A large online community of developers with lots of information pertaining to issues experienced and there resolution.

## 2.4 Current State of the Art

This section examines the issues this project is looking to resolve and compares them to the latest developments and research into the area. Web3 and blockchain applications are still in their infancy however there has been a strong community working towards making these applications more common place and accessible to users. This examination will uncover how best to implement a solution to this issue and open discussion regarding how best to build and design these applications. First lets examine why we even need web3.

### Web3

The Internet in its current form relies very heavily on the client-server architecture for the sharing and storage of information, which means that data is stored largely in central locations that are controlled by a single entity. This type of architecture has provided speed and for companies to easily access data from a singular location however this also exposes a single point of failure in these systems making data breaches a common occurrence. According to one report[14] there was over 4.5 billion data records compromised in the first half of 2018 which is up 133% compared to the first half of 2017, with social media accounting for over 56% of these breaches. This high level of breaches shows that there are flaws in our current model for the Internet and pushes the necessity to start looking at alternative more secure models.

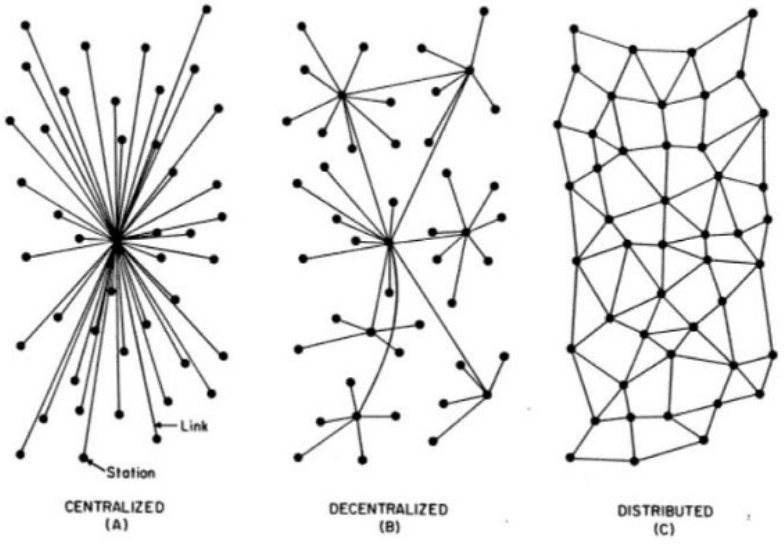


FIGURE 2.2: Centralised Vs. Decentralised Vs. Distributed Architecture[15]

Figure 2.2 shows a comparison between the different types of network architectures and decentralised is the one this project is most interested in. Most large companies like Google and Facebook use a distributed architecture which has some similarities but ultimately it is closer to a centralised architecture. The data in these companies is distributed globally among a few massive server farms that ultimately are centralised and so still have the single point of failure issue.

The other issue is that these companies own massive amounts of data with the users having little control or view over what data is gathered. This allows these companies to build up massive and very accurate profiles on their users and even non-users for targeted advertising and data analytic purposes. This data possess' a significant value

to it and has allowed these companies to become very profitable as a result. The value of this data is not truly known to the users of these sites.

This is where web3 has the ability to make a difference. Web3 is viewed as a Internet where data is not owned but instead shared and is the next evolution in the Internet[16]. It combines a number of features to make the Internet more user oriented and personal. For this project that is a vital point as user data and it's ownership is a concern being addressed. Users should have full control over their data and choose what they wish to share with companies so this project will look at a design of instead of users subscribing to a service and the company gathering data, the company will be able to subscribe to the users data in exchange for remuneration. By doing this we allow for distribution of wealth and creating an application that values the contributions of the users. This opens up to where blockchain comes into this project.

### Blockchain

In recent years there has been a growing interest in the blockchain market and this has also been coupled with a growing lack of trust in current social media platforms and the security of user data. According to a report done by LinkedIn there was 33x growth in the area of Blockchain related jobs during 2018[17] with solidity being one the main sought after skills in this area which is the programming language used to create smart contracts on Ethereum. This shows that there is a growing number of decentralised applications under development.

Blockchain has played a major role in the advancement of web3 as it is a fully decentralised system with accountability and proof of ownership. The most notable blockchain cryptocurrency is Bitcoin which was the first of this system to resolve the issue of the reversible payment without the need for a trusted third party[7]. However blockchains are not limited to being a ledger of financial transactions but can be used for anything. The blockchain technologies that this project focuses on is those that were developed with decentralised applications in mind. The benefit of DApps over web2 applications is they are immutable, corruption and tampering is extremely difficult, and overall secure.

Decentralised applications have existed as long as the Internet has been around in a simple format. The earliest of these were basic HTML sites that were stored locally on PCs and accessed by external parties using P2P protocols. The Ethereum network is most well established of these blockchain applications designed for developing DApps.

Ethereum implements blockchain in a general manner so that other developers can focus on developing applications and not have to worry about the complexities of developing their decentralised blockchain network[8].

Ethereum and blockchains in general are not without their issues and concerns most notably related to privacy concerns with sharing this information over blockchain because of it's transparency anyone can look up a transaction on the network, however this does not mean they can look at what data was transferred[18]. It does mean though that all communications routed through the network require the metadata related to every communication be made available to the whole network and so it must be considered that if blockchain is not implemented properly privacy may become more vulnerable than on centralised networks[18]. This enforces the need for strong encryption and assurance that no personal user data is directly added to the chain and abstraction is used.

Another concern that is not directly related to this project but should be noted is the environmental impact mining blockchain has as it consumes a massive amount of power. One report found the power consumption used for mining Bitcoin alone is comparable to Ireland's electricity consumption[19] and this grows with time as it becomes more complex to perform mining on Bitcoin. Ethereum currently uses the same technique for validating transactions on it's network however it is moving towards a less intensive method that will reduce the power consumption and environmental impact of Ethereum significantly. This protocol has been named Casper and uses the proof of stake method for validating transactions.[20].

Casper will use proof of stake instead of the power intensive model used now known as proof of work. It is important to recognise the differences in these protocols as Ethereum will be changing to proof of stake but still currently uses proof of work. The main differences of this have been highlighted in table 2.1.

| Proof of Work | Proof of Stake |
|---|---|
| All miners attempt to solve the complex calculation to create the new block | The creator of the new block is determined based on the wealth and age, also known as stake |
| The first miner to solve the issue is rewarded | The miner takes a transaction fee |
| Can be extremely expensive and slow with roughly 5 transactions per second on Ethereum | Much less expensive and much quicker with roughly 1,000,000 transactions a second eventually for Ethereum |

TABLE 2.1: Proof of Work Vs. Proof of Stake

A report[21] that proposed a solution with similarities to what this project hopes to achieve was a personal-data management system to ensure that users own and control their own data using blockchain as an access control that could perform instructions, such as storing, querying and sharing data. It uses a distributed hash table for storing this data and ensuring it's security. As blockchain is public, how the data is stored on it is of utmost importance.

Another reason, outside of security, for not storing data directly on Ethereum is because of Gas[8]. Gas is the name for a unit that measures how much work an action or set of actions takes to perform and this results in a price that must be paid. The larger the amount of data being stored the higher the gas and so it is essential to try and keep it to a minimum, hence why a hash table or some other form of reference is preferable.

$$Cost = GasLimit \times GasPrice \tag{2.1}$$

Equation 2.1 shows how the cost of gas is calculated. Gas limit represents the maximum amount of gas allowed in a block and this determines how many transactions are allowed in this block. Gas price then represents the cost per unit of gas. An analogy to make this easier is think of gas limit as litres of petrol and gas cost as the price per litre of petrol. 21000 is the standard gas limit for transactions.

This is also something that may be off-putting to new users as we have become familiar with websites that allow us to store unlimited amounts of personal information and posts for free on applications such as Blogger, Facebook and Twitter. However if the user is provided with tokens that not only they could exchange for fiat currency but also use to pay gas prices for creating new content it could encourage them to become comfortable with the platform.

This raises question though of where do we store user data and blogs if not on the blockchain itself?

### Storage

One of the major issues that centralised applications has resolved is the availability of data and ensuring that the data being made available to users is the most up to data. CAP theorem [22] states that a networked system can only have two out of three from the following: consistency, availability, and partitioning. In a decentralised application we have already opted for partitioning as the data is distributed among the various nodes we must try and find a good balance between consistency and availability of data.

One paper addresses this issue by combining blockchain and IPFS to build a decentralised application[23]. It recognised that existing P2P models needed revision in order to cover the full spectrum of potential systems that can now be implemented and the challenges that a fully distributed system will encounter. These challenges mostly related to data discovery and when looked at in terms of the CAP theorem, as has previously been mentioned, partitioning is a requirement and so finding the best balance between consistency and availability is what remains for the developer to do. The solution the paper[23] presented that best fulfilled these requirements as well as data trust was combining IPFS and blockchain.

As mentioned previously IPFS is the most well established of the decentralised storage platforms currently available at the time of writing this paper. IPFS[11] is a P2P distributed file system that seeks to connect all computing devices with the same system of files and replace HTTP. Instead of using location based addressing it uses content addressing and is similar to a BitTorrent swarm objects within one Git repository. This systems is resistant to censorship and can also potentially be accessed without an ISP. This system also has no single point of failure unlike the current centralised servers.

The IPFS network of data forms a Merkle DAG structure upon which versioned file systems, blockchains, and even a permanent web can be built. DAG stands for Directed Acyclic Graph, and Merkle signifies that this is a cryptographically authenticated data structure that uses cryptographic hashes to address content. Figure 2.3 gives a high level outline of the differences between centralised and IPFS networks.

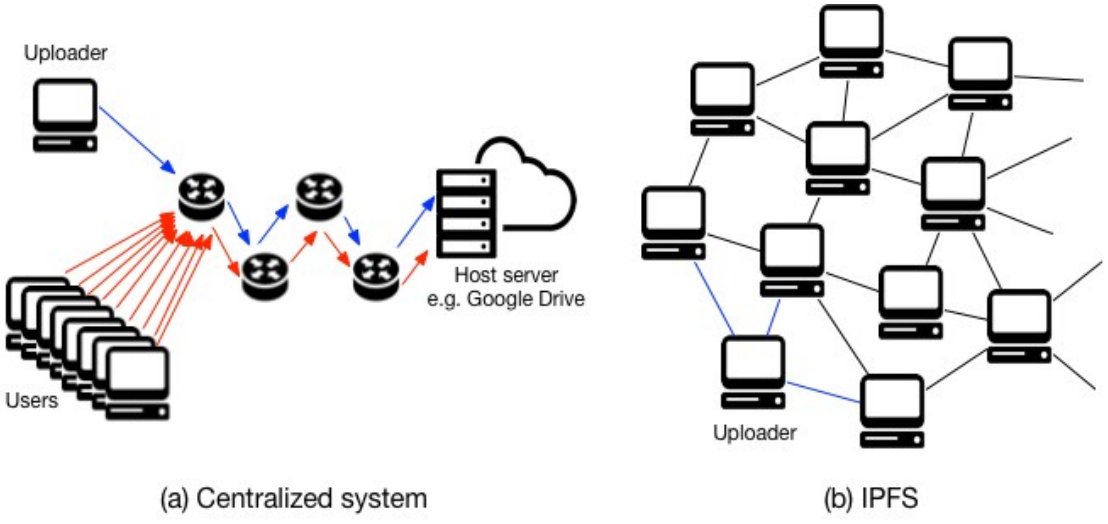


FIGURE 2.3: Centralised Vs. IPFS storage[24]

When using IPFS, instead of requesting the data from a server, you ask the entire network for the content through its hash, and the nodes with that data will respond meaning this is potentially quicker than HTTP which gets the data from a single point.

So you are looking for a certain file instead of a server. The blogs are an obvious choice to store on a system like this and the content hash then made publicly available on the chain for users to access these blogs. If this system is to also be used for storing user data though it must incorporate encryption of the data on IPFS.

### Remuneration

This brings forward the question of how does one remunerate a user for their data through tokens? It has been noted that users are more concerned with the type of data being gathered as opposed to the quantity of data[25]. In terms of data remuneration for this project this is interesting as it means that a potential method of valuing data is by type and not by quantity gathered. This would have similarity's to a subscription service, whereby instead of the user subscribing to the company, the company subscribes to access of the users data. This poses an issue though as some users may provide a lot more data than others with some having little to no activity at all. The other alternative that will be considered for this project is to remunerate users based on the amount of data they provide and it's type. The issue that has been recognised with this method is that users may start creating fake sites, or bots, in order to be remunerated for more data. This form of rewarding users for the amount of data they provide could also encourage social media addiction[26] as users may begin to feel rewarding for more online activity. Social media addiction has been recognised as an official condition in the UK and so this should be considered when designing such an application.

Using a subscription service would require that data is broken down into types and valued accordingly. If we look at the Facebook Data Policy[27] and analyse the data types that they gathered we could break it down into the following categories:

- Personal: This would include things like name, location, age, political beliefs, etc.
- Interests: This includes things such as metadata from posts liked, shared, and interacted with.
- Social: This would include your social interactions, connections and so on.

While this is a very high-level depiction of what Facebook gathers it can give a good idea of the types of information that this site could collect on users and how to separate it. As this project is not heavily focused on the gathering of data itself it won't be a major part however it will be implemented on a small scale to provide a proof of concept.

If we look at the data categories above it obvious that personal is by far the most valuable of these, with interest's and social having a similar value. By breaking the data up into

these categories it will give the user more control over the data they share and allow us to more accurately remunerate them for this data.

How we value this data would really depend on the market and so as popularity in the site would grow so would the potential price of data for users. One company found that 50% of the average users mobile data is for ads and trackers, costing as much as $23 a month[28].

This is not to say that advertising is bad on social media, it plays a vital role in supplying the service for free to the users. This project agrees that advertising and user data has provided a very good economic model for profit but feels that a percentage of these profits should be distributed to the users for their data contributions.

Another point to note is that the user will also be remunerated for original content creation. Content creation is important as this is what will attract users to the site and keep them interacting with it so it is important to reward users for this. This will be achieved through an initial token for creating the content and then users have the option to support this content also.

Two notable social media sites also built on blockchain that have implemented token support for content creation are SocialX[28] and STEEMIT[29]. STEEMIT uses a cryptocurrency upvote system where an automatic amount is sent to the content creator. SocialX on the other hand has features closer to what this project implements. SocialX has a like button which does not support the content creator with cryptocurrency and has a separate superlike feature for this instead.

This project will incorporate both a like feature and a support feature into the system, however it will differ from SocialX in that users are rewarded for their data as well content creators being rewarded for creating new content.

# Chapter 3

# User Data as a Commodity! Decentralising the Online Profile

This section breaks down the problem that this project is addressing and discusses why it is felt this must be addressed. Here the objectives of the project are clearly defined as well as the functional and non-functional requirements of the system.

## 3.1 Problem Definition

Blogs have been around since the start of the Internet and have played a major role in the sharing of information online and allowing people to connect with each other, however our current systems are not without their flaws. These flaws that this system is looking to resolve are data and content ownership and remuneration for the personal and intellectual property of users.

Users should control and own their data however that is not the case in the majority of current social media systems which often provide little insight into the data gathered. This has somewhat improved with GDPR[30] introducing laws that require sites to provide users with their data only on request.

These systems require users to fully trust the third party to safely handle their data and not misuse this data. This data is often gathered with little knowledge on the users end of what data is being gathered and how it is being used. They also are often not fully aware of who has access to this data. This data has a huge amount of value to it and is what allows these companies to make massive profits through advertising and sharing data with third parties.

These users should be rewarded for their data contributions to the site and the companies distribute the wealth more evenly among its users. This creates a stronger trust between the user and the organisation as the user is in full control of their data.

Content creation is another area that users need to take back ownership over as social media sites are nothing without the contributions of these users. Content creators provide the content that attracts traffic to the site and generates revenue for the organisations. Content creators receive a very small percentage with Google and YouTube taking 73%[28] of all ad dollars. It is time we as users start valuing these contributions more by letting the community decide how much these creators should receive, allowing for the community to be in full control of what they feel is worth their support.

Social media sites have been places for people to express themselves freely however this freedom of expression has come at a price with Turkey banning Facebook and WhatsApp[31], North Korea banning most social media, and China attempting to control what people view online. This is not what the Internet was intended for, it was meant to be a free sharing of ideas and information. Blockchain has the ability to give this back to users as no government or organisation has the ability to censor and control what is put into blockchain. This is a tremendous step forward for freedom of information online.

Another issue the author sees with our current social media centralised models that the networks are built around. This provides a single point of failure in the system and are constantly under attack from hackers. By distributing this data over a decentralised network it makes it extremely difficult for attackers to gain any significant amount of data as they lose the single point of failure. This further strengthens the argument to move away from centralised data storage to a more secure decentralised network.

## 3.2 Objectives

- Have fully functional decentralised web application that works in parallel with blockchain and IPFS.
- Store data from the users interactions with the site on a distributed network and provide a web-based control panel in which they can view and manage the sharing of data.
- Remunerate users for the sharing of this data in the form of a token.
- Reward content creators for new content creation.
- Allow users to access other users blogs using the IPFS content addressing protocol.

- Users will be able to support content creators through the sharing of tokens.

## 3.3 Functional Requirements

- Be able to gather metadata based on the users interactions and store this securely in the IPFS.
- The web app should allow users to view and choose what data wish to share.
- The application should automatically remunerate users in the form of tokens for their data once a month. This comparable to a subscription service whereby the administrators are subscribing to user data.
- The application should prompt the user to support a content creator with tokens when supporting their blog post.
- The application should allow the users to create and post new blogs which are stored using IPFS.
- The application should have a homepage that will show a feed of blogs from users followed in order of time posted.
- The application should allow users to share another users blog to their page and retain the link to the original content creator.
- The application should automatically detect and connect to the users wallet.
- On start-up it should be able to retrieve the users blogs by querying the chain for the IPFS hash and then by querying the IPFS network for the blog.
- The application should encrypt all users data securely.
- The application should stop gathering data on users when they are opt out and remove all links to this from the chain.
- The administrator application should only be able to view data that they are permitted to.
- If the user chooses to stop sharing data they will be paid the tokens they are owed up to that point of the month immediately as opposed to the end of the month.
- If MetaMask is not installed the user should be given instructions on how to set this up.

## 3.4 Non-Functional Requirements

- The web application should be simple enough that relatively new users of decentralised applications are comfortable with it.
- It should automatically be able to connect to the chain if the user has MetaMask setup and also be able to connect to IPFS without the user having any technical knowledge of the areas.
- It should be clear what data is being gathered on the user.
- Connecting to the chain and IPFS should take under 30 Seconds.
- The application must accommodate all screen sizes.
- All user data should be encrypted.
- All IPFS hashes for user data should be encrypted when on the chain.

# Chapter 4

# Implementation Approach

## 4.1 Architecture

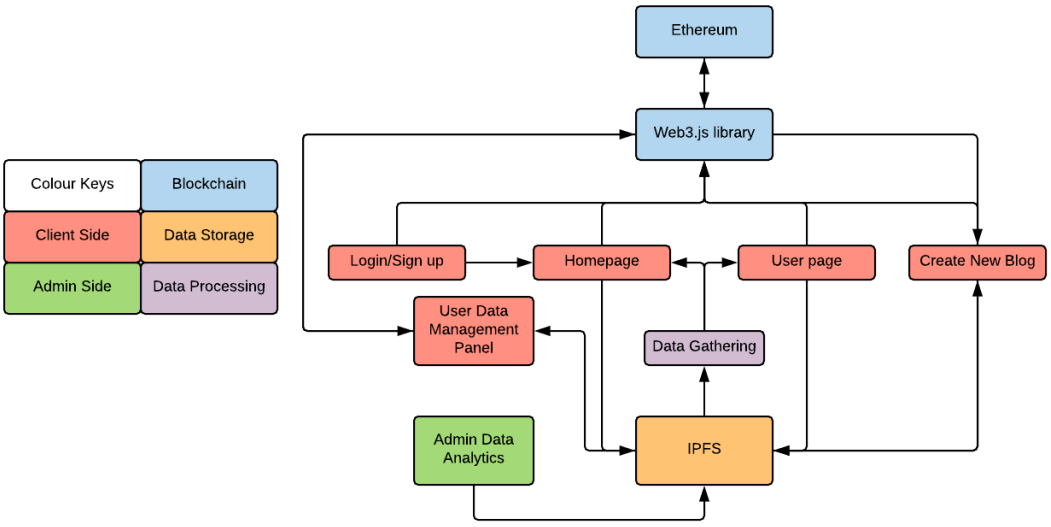


FIGURE 4.1: System Overview

For this we will look at the planned architecture for the implementation of this project and discuss in detail the technologies that will be used in order to achieve this. We will look at the major programming languages, frameworks, and tools that will be employed throughout the development of the project and that will build it's core by breaking the project down into it's three main section which are a client-side web application, back-end blockchain application, and distributed storage solution. All of these combine to create a cohesive web-application.

It is important to note that the diagrams in this section will not necessarily reflect the final product as things will change as the project develops. Firstly before delving into

the details of each section lets list out all the main technologies and tools that will be used during the development of this project.

- Solidity: is the programming language used for developing applications that interact with the Ethereum blockchain network.
- JavaScript: is a high level interpreted language that is often used in web-development.
- AngularJs: is a fully client side JavaScript library for front-end web-development.
- Web3.js: this is the JavaScript library that allows our web application to communicate with the blockchain backend.
- Bootstrap: is a front-end framework for developing web-applications.
- SCSS: this is the latest version of CSS and is a super-set of CSS3's syntax.
- Node.js & NPM: this is a JavaScript runtime environment that will be used during the development of this project.
- IPFS: is a p2p network protocol that will be used for the storing and sharing of user data and blogs.
- MetaMask: is a tool that allows us to connect the Chrome browser to web3 applications.
- Truffle Suite: is a tool-suite for developing and testing DApps.
- Ganache: is a local virtual blockchain testing environment.
- OpenPGP.js: is a JavaScript implementation of the OpenPGP protocol encryption standard.

### Blockchain

Blockchain is what this project revolves around as the goal is to create a truly decentralised blogging application that rewards users for their data and content contributions to the site and so without it this would not be possible. The chosen blockchain technology is Ethereum as this was designed for the development of DApps and provides a Turing complete contract oriented programming language known as solidity[8].

The smart contracts that are developed for this will act similar to a traditional backend to any site such as providing functionality and access to data. The big difference here to a traditional backend is that all the code and transactions that take place are transparent.

A transaction that takes place is not strictly financial but can be of anything on the network. For this reason if we are performing transactions of secure data it should always be encrypted. This part particularly applies to when the transaction of the hash for personal data, it should be encrypted using the public key of the recipient which in this case would be the administrator account.

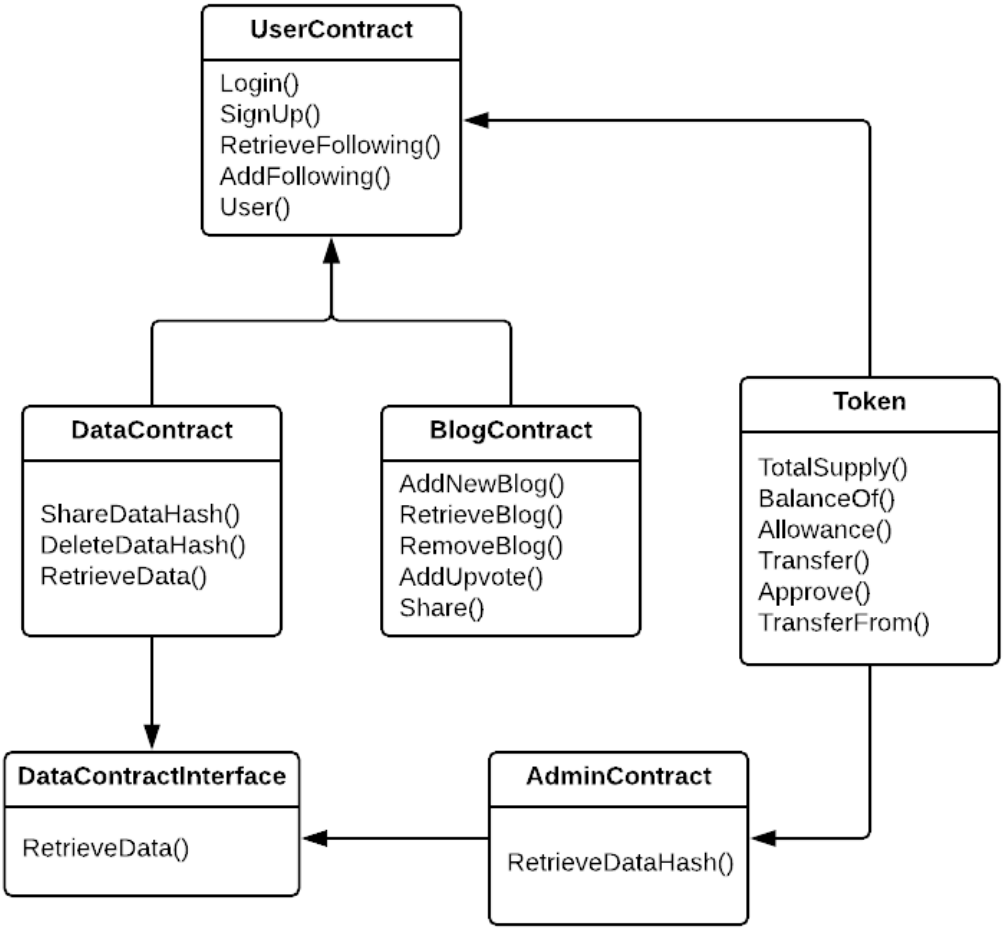


FIGURE 4.2: Blockchain Contracts Overview

Figure 4.2 shows a high level overview of the requirements for the contracts on the blockchain. As you can see there are number of contracts that have been divided based on functionality. The user contract is important as the will store things such as the blockchain account number and other user details that will be used by almost all other contracts. The reason for this is anytime the user wants to perform a function on the chain which involves a new transaction their account number is required and to get their balance and check if they have enough tokens to pay gas for the transaction.

The token contract has been modelled on the ERC20 Token Standard[32] which is a standard that describes the functions and events that an Ethereum token contract has

to implement. As this project does not aim to do anything fancy with the token side of the project and to ensure it can work with the main wallets, in particular MetaMask these are the functions that we will stick to and not expand upon.

As has been previously mentioned Ethereum is by no means a storage solution as each item uploaded costs gas and so it would be very expensive to use this for the storage of blogs and data. The functions in BlogContract simply store and retrieve content hashes to the blogs and small information that is not too costly for the user to perform, this is similar in the DataContract. The thing to remember when interacting with Ethereum is that it doesn't cost you anything to retrieve data however to add data, or a new block, it costs gas.

### Distributed Data Storage

As this project aims to be a fully distributed and decentralised application it is important that the data is also stored in this manner. This gives the application a higher level of security and makes it harder for attackers to gain data from the system as there is no single point of failure. The technology that will be used to implement this distributed storage is IPFS which is a protocol designed for the storage and retrieval of data on a decentralised distributed system.

The two types of information that will be stored on IPFS are the blogs and the user data. When storing the blogs they must be processed into a new text file that can then be pushed to the IPFS network. This will require that there is a standard format for displaying each blog into which text can be inserted and stored. Once the blog has been pushed to IPFS the returned hash for addressing this content must be then stored on the chain in order to be able to retrieve these blogs. A high-level overview of steps involved is outlined in figure 4.3.


FIGURE 4.3: Blog Storage Process

The process for adding user data will be similar however will require significantly higher security and encryption in order to ensure that user data is securely stored on the network

and can only be accessed by those with permission. On top of the steps outlined in figure 4.3 there will be additional encryption steps and the format of the data will be a JSON file and not a text file. It should also be noted that not all data is stored in the same location but is divided based on its type such as personal data, likes, and so on. This will allow the user a higher level of control over the data they share as they do not have to share all the files. Once the file has be converted to JSON it will be encrypted using the public key of the user and another copy of the file is encrypted using the public key of the administrator account. The IPFS hash of the encrypted data is then pushed to the blockchain for the administrators to access it, provided that the user has chosen to share their data. This will also be divided into the various subsections of data gathered. The user will also have an option to end all data gathering which will mean no data will be pushed to IPFS and saved.

As IPFS is an immutable tool it is difficult to edit and manage files on the system however they have provided a mutable file system tool built into IPFS that allows developers to manage files just like traditional file systems and this is the tool that will be used for managing user data.

The process of receiving the tokens for both of these is different also. When a user creates a new blog they will be automatically rewarded with a token for providing new content to the site. When the user shares data they will receive the tokens for the length of time that they have shared their data on a monthly basis, similar to paying for a subscription service. If they choose to end the sharing of their data they will receive their tokens at this point and will be provided with the amount of tokens for the number of days they had chosen to share it, since the last payment.

### Client-Side

The client side interface of this web application that will be extremely familiar to what users are use to in traditional web applications. There will however be some changes that will be needed in order for users to be able to connect to the Ethereum network. This project will be designed alongside the MetaMask Google Chrome extension which allows people to access web3 sites through their chrome browser without needing a specific browser for web3 sites. The MetaMask extension allows for the easy connection to the Ethereum network. Users will require the MetaMask extension in order to be able to access the site or another web3 compatible browser.

The site will be designed using traditional client-side languages which include JavaScript, HTML5, and Bootstrap to create a dynamic and easy to use application. The important new libraries that will be incorporated into this are web3.js and the IPFS HTTP Client

library. Web3.js is a collection of libraries that allow users to interact with Ethereum using HTTP or IPC. The IPFS HTTP Client library allows for the client side implementation of IPFS in JavaScript.

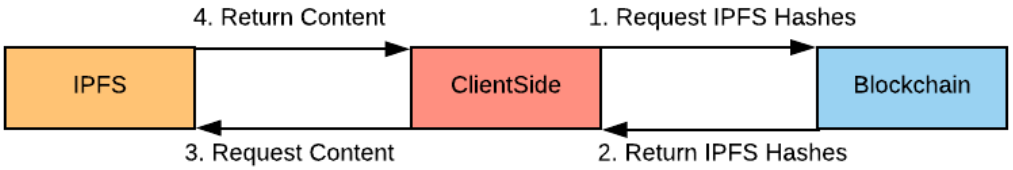


FIGURE 4.4: Retrieve Content Process

The process outlined in figure 4.4 is the same process that will be used for populating the user page with their blogs and the homepage with blogs that the user follows. The user will select a page and that will send a request to the blockchain network for the IPFS hashes that are relevant to that particular page. The blockchain network will then return these IPFS hashes and then a request is sent out to the IPFS network for this content that is returned and loaded to the page.

There also needs to be a page in which the user can manage their data stored on IPFS which will be retrieved in a similar manner to figure 4.4 however with the addition of decrypting this data. The user will be able to view their data in a clean and easy to read format that they can interact with. They will be able to read and delete data from here and also be able to control what, if any data, they share with the administrators. This is all possible by using IPFS mutable file system tool.

### Security

Security for this application is important because of the transparency of blockchain and it is an application that wants to return data control to the users and not expose it. As DApps are very much client-side application and there is no traditional server-side to this application it means that all security will have to be performed on the client side.

OpenPGP is the most widely used email encryption standard [33] and they also provide a JavaScript implementation that would be ideal for this project. It has many forms of encryption including ECC which is consider to be very secure. The library also allows for password encryption.

The users data will be encrypted using public key cryptography, ECC to be specific, and the public key stored in the OpenPGP cloud that is provided. The private key will

then be encrypted using a strong password provided by the user and then stored in the blockchain network that the user can access and decrypt so that they can access their data.

To share this data with the administrators another copy of the data must be made and encrypted using the public key of the administrator account. The IPFS hash of the location of this data is then shared to the chain giving access to the administrator to decrypt this data using their private key.

### Use Case Description

Figure 4.5 shows a user use case diagram highlighting the main features of the system that the user has access to. Blue represents features that the user has direct access to and can interact with and the red use cases represent back-end implementations that are a result of these features. The view wallet function will be as a result of the MetaMask Chrome extension.

Figure 4.6 Outlines a high level overview of the relationship between administrator and users. The main function of administrators is to perform data processing and as a result of this promote posts and advertisements directed at the user.

Table 4.1 to 4.9 is an outline of the use-cases for this project and outlines the basic steps for accessing each of these.

| Use Case Name | Create a new blog |
|---|---|
| Actor | User |
| Description | A user wants to create a new blog |
| Flow of Events | • User selects option to create new blog<br>• User enters the blogs data<br>• User selects post and then must wait for confirmation that this has successfully posted to IPFS.<br>• Once confirmation is received the user must accept the storage of the IPFS hash to blockchain. |

TABLE 4.1: Create New Blog Use Case

| Use Case Name | View Blogs |
|---|---|
| Actor | User |
| Description | A user wants to view the blogs of the users they follow |
| Flow of Events | • User selects homepage button<br>• Waits for the data to load and display the titles of blogs in order of creation on the homepage<br>• They click on the title of the blog they wish to view and are brought to that blogs page to view it. |

Table 4.2: View Blog Use Case

| Use Case Name | Support another blog |
|---|---|
| Actor | User |
| Description | A user wants to support a blog they like with tokens |
| Flow of Events | • User selects blog they wish to support.<br>• The user then clicks on the support button and a pop-up appears asking for how much tokens they wish to give.<br>• The user enters an amount and they select support.<br>• User approves the transaction. |

Table 4.3: Support a Blog Use Case

| Use Case Name | Share a post |
|---|---|
| Actor | User |
| Description | A user wants to share another users blog to their page |
| Flow of Events | • User selects blog they wish to share.<br>• The user then clicks on the share button.<br>• User approves the transaction to write the IPFS hash to the blockchain for their account. |

Table 4.4: Share a Blog Use Case

| Use Case Name | View Gathered Data |
|---|---|
| Actor | User |
| Description | A user wants to view data gathered on them |
| Flow of Events | • User selects Data Sharing option.<br>• All gathered data on the user will be displayed here. |

TABLE 4.5: View Gathered Data Use Case

| Use Case Name | Turn on/off Data Sharing |
|---|---|
| Actor | User |
| Description | A user wants to share data in exchange for tokens. |
| Flow of Events | • User selects Data Sharing option.<br>• The user will be able select the types of data they wish to share and turn these either on or off and also view how much each category of data is worth.<br>• User selects the types of data they wish to share and then click on the share option.<br>• The user confirms this transaction of the IPFS hash to the blockchain. |

TABLE 4.6: Share Data Use Case

| Use Case Name | Follow a Blog |
|---|---|
| Actor | User |
| Description | A user wants to follow the blog posts of another user |
| Flow of Events | • User selects blog of user they wish to follow.<br>• The user will be able select the follow button.<br>• The user confirms this transaction of the users ID to the blockchain. |

TABLE 4.7: Following a Blog Use Case

| Use Case Name | Advertise to the User |
|---|---|
| Actor | Admin |
| Description | Admin posts adverts that the user can see. |
| Flow of Events | • Admin selects advert to be posted.<br>• Confirms the storing of the IPFS hash of this advert in the blockchain.<br>• Advert is then pulled down to relevant users and displayed at various points on the homepage. |

TABLE 4.8: Advertise to User Use Case

| Use Case Name | View and Query User Data |
|---|---|
| Actor | Admin |
| Description | Admin wants to view and query user data that they have access to. |
| Flow of Events | • Admin opens their control panel.<br>• They enter a query in the search area.<br>• The results of relevant data is displayed on their screen. |

TABLE 4.9: View User Data Use Case

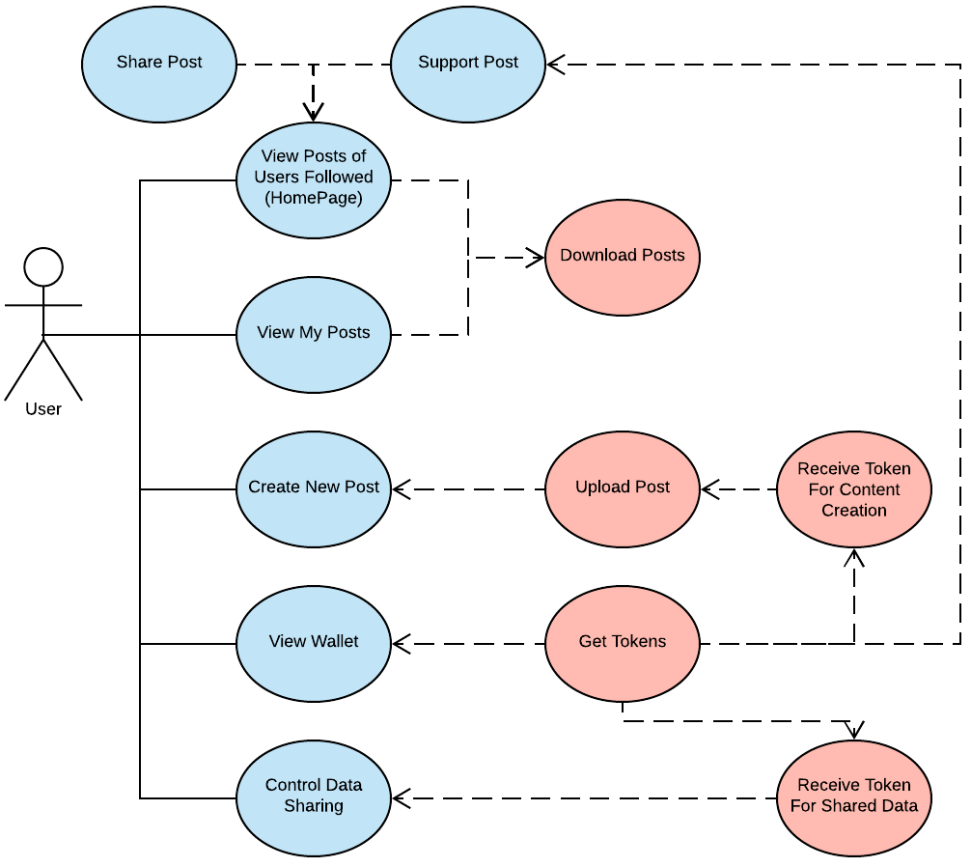


FIGURE 4.5: User Use Case Diagram

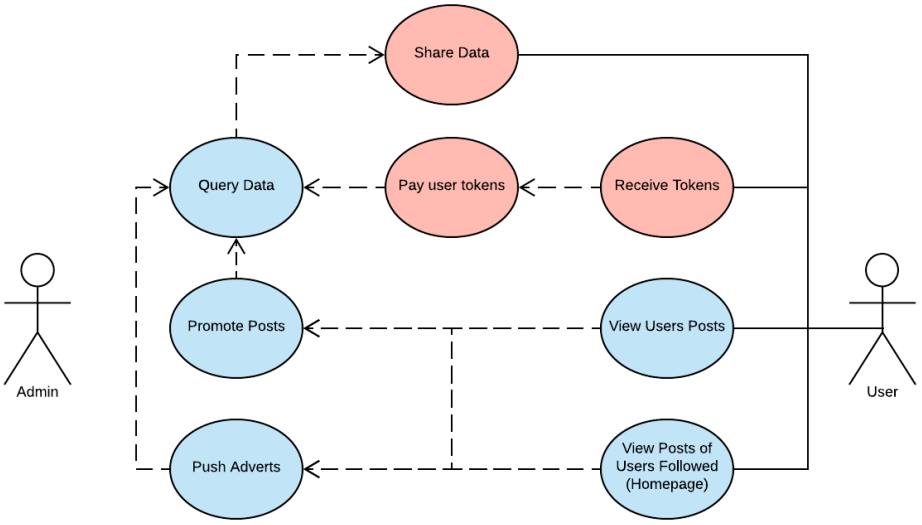


FIGURE 4.6: Admin User Relationship Use Case Diagram

## 4.2 Risk Assessment

This sections outlines potential risks that could affect the outcome of this project and have been classified using the table in 4.15. Anticipating these risks allows preparation to mitigate these and as a result be better equipped to handled them. Tables 4.10 through 4.14 outline potential risk that may arise during this project and the resolutions for them.

As well as preparing for certain risks by outlining them here some have already been mitigated through testing and implementing small parts of the project as it was researched to ensure that the technologies chosen would work and to identify any potential flaws in them early on.

| | |
|---|---|
| Risk | Inability to reach the sprint goal |
| Category | Rare Major |
| Details | This will occur usually due to an underestimation of the time required to complete a certain task or if an unforeseen issue has occurred that prevents the task being completed. |
| Solution | If this is caused due to an underestimation of time then the time required should be reevaluated and and brought into the next sprint for completion. If this is due to an unforeseen issue with the task it will require further analysis to find a feasible solution to this issue, however this issue cannot be allowed to hold back the progress of other tasks that it does not involve. |

Table 4.10: Sprint Goal Analysis

| | |
|---|---|
| Risk | Deprecation of features in use |
| Category | Rare Critical |
| Details | Some features in use may become deprecated during an update to the tool or library being used. IPFS and Solidity are particular vulnerable here as they are quite new features still under heavy development and often roll out new features in replacement of old ones. |
| Solution | While this may cause a setback time-wise they would usually release details on what features have been deprecated and their replacements. This would require analysing code and implementing these new features in replacement of the older ones. |

Table 4.11: Deprecation of Feature Analysis

| | |
|---|---|
| Risk | IPFS and Ethereum network speeds |
| Category | Remote Minor |
| Details | As these are both decentralised and distributed networks they may at times experience slower responses than what users have come to expect from centralised systems. |
| Solution | This is something predominantly out of our control however an evaluation of the data sizes being sent over these networks could be done to see if any improvements can be made here. |

Table 4.12: Network Speed Analysis

| | |
|---|---|
| Risk | Ethereum Gas Cost |
| Category | Occasional Minor |
| Details | Everytime a user performs a transaction on Ethereum that requires a new block they pay gas for this. Depending on the size of the block and the data the price can be quite high |
| Solution | Reevaluate the size of the data being written to the Ethereum network and look for potential solutions to write this data elsewhere. |

Table 4.13: Ethereum Gas Cost Analysis

| | |
|---|---|
| Risk | Underestimation of Feature Difficulty |
| Category | Rare Fatal |
| Details | If a features difficulty has been underestimated this could be potentially fatal for this particular feature as it may show that there is not enough time to implement it or to gain the skills necessary for its implementation. |
| Solution | This will require an evaluation of the feature and if no solution can be found it may have to be removed from the project. |

Table 4.14: Feature Difficulty Analysis

| **Frequency/ Consequence** | **1-Rare** | **2-Remote** | **3-Occasional** | **4-Probable** | **5-Frequent** |
|---|---|---|---|---|---|
| **4-Fatal** | Risk 5 | | | | |
| **3-Critical** | Risk 2 | | | | |
| **2-Major** | Risk 1 | | | | |
| **1-Minor** | | Risk 3 | Risk 4 | | |

TABLE 4.15: Risk Matrix

## 4.3 Methodology

In order to tackle the various components of this project it was necessary to break it down into it's various subsections. As the main features of this project are still in their infancy it was of extreme importance dedicate time to researching these elements in detail and get a good understanding of how to execute these. The potential main difficulty with features being relatively new is a lack of documentation however due to the hype surrounding these technologies there proved to be a strong development and support community. The following outline methodologies that are used for researching this project:

- Blogs are source of detailed information and tutorials on how to use and implement IPFS and write smart contracts in solidity. Even though both these areas are relatively new they have attracted a large amount of hype and so it proved easy to find developer blogs on these topics.
- Reaching out to people in industry can be incorporated as a form of guidance and often is useful for answering any queries that could not be found online. Industry leaders can be an excellent source for discussing ideas and how best to implement them, and even if the project approach was something worth while.
- YouTube also contains some helpful videos on implementing DApp projects, often produced by those working in the industry or by the creators of the product.
- Google Scholar is a search engine designed to retrieve scholarly literature that can be used for research and citations on the topic.
- Reading over the white papers and technical specifications of the tools gave insight into the capabilities available within these systems.

The research phase will prove invaluable for learning any new technologies that have not been covered during the course such as IPFS and Solidity. These topics are still

relatively new however they have attracted a lot of attention from developers and so there is quite detailed blogs on how to use these. The main sites of both also provide excellent tutorials and detailed documentation of using them.

JavaScript is a skill that is taught as part of the Software Development course and so applying the skills gained here to understanding these libraries and how best to implement them will make the learning curve lesser. In order to create a dynamic and overall appealing site there will be other web skills incorporated too such as Bootstrap and SCSS.

As this will be a self managed project it is important to have a strong but flexible approach towards the development of the project. A variation of the agile process that revolves entirely around one individual will be used in combination with test driven development to ensure that the best code is been written.

A tool that will be used to manage this is Trello. Trello is a web based project management tool. It allows for the easy tracking and management of features and the creation of boards for this. This tool is ideal for the agile approach to software development.

The agile process being followed will be similar to that of scrum except as this will be a self-managed project the roles typically assigned to various individuals will be all undertaken by the developer. The development of the project will be done in 2 week sprints with goal of each sprint being to produce a minimum viable product (MVP).

The features being implemented must will be decided on before development begins and added to a product backlog. The beginning of each sprint will involve selecting a feature from the product backlog to implement over the course of that sprint. If an issue is encountered within that sprint and it is realised that there is not enough time within the sprint to complete a particular feature than this must be addressed immediately. This will be addressed by having daily reviews of the progress being made. It will also be important to have a larger review at the end of each sprint and this should provide a good idea of how complicated the next feature will be to implement.

As the project develops further there may be a realisation that a certain feature might not work and will require a review and possibly another implementation. If another implementation is required then it is important that this is reviewed and a report written on why it was necessary, what the new implementation requires and the research that brought the conclusion that this is the best way to continue with implementation.

## 4.4 Implementation Plan Schedule

This section breaks down the use cases that were outlined earlier into the features that will be implemented during each use case. This will be used to build the product backlog and each use case will be given an estimated delivery time so that we can estimate what will be involved in each sprint.

### Create New Blog

This includes an implementation of majority of the tools that will be used throughout the project and so is a good starting point to get estimates of difficulty for each tool.

- Create front-end page for writing a blog.
- Implement IPFS storage.
- Implement writing to the blockchain network and storage of IPFS hash.
- Get user input and transform this into a blog page.
- Store blog on IPFS and the hash on the blockchain.

### View Blogs

This page must be able to retrieve and display blogs that a user has either created, shared, or followed.

- Create the front-end page for displaying blog titles.
- Retrieve blog details and IPFS hash from the blockchain.
- Create redirect to blog on IPFS when selected.

### Support a Blog

This is the feature that allows users to support a blog with tokens or simple just like the blog post.

- Implement pop-up box for asking if the user wants to support a blog with tokens.
- Get blog creators details from the metadata.

- Add ability to perform token exchange with blog creator.
- Store support count with the blog.
- Create web applications token that will be used on the site.

### Share a Blog

This must allow the user to share another users blog to their page however it must retain the original content creators details.

- Save IPFS hash to the chain under the users ID.
- Visually distinguish this from the users blogs on their page.

### Follow a Blog

Add another users details to the list of users that are being followed so that their blogs are viewed when the user accesses the homepage.

- Add the users details who is being followed to a list that is linked to the current user logged in.
- Retrieve details of the users being followed from the chain.
- Search the chain for blogs added by the users being followed and display these in chronological order.
- Add ability to remove a user from the list of followed users.

### Turn On/Off Data Gathering

Here the user must be able to turn on/off data gathering but also this data must be secured using encryption and the transfer of it's location must also be implemented securely.

- Implement background data gathering features.
- Add ability to store this data in JSON using IPFS MFS.
- Add ability to turn data gathering on/off.

- Setup and save the users public key and private key.
- Encrypt and decrypt the data using public key encryption.
- Store IPFS hash in table accessible the administrators.
- Display data in a readable format for the user.
- Allow user to delete data using IPFS MFS.

### View User Data

This allows administrators to view user data that they have permission to access.

- Get IPFS hash from the blockchain network.
- Decrypt this using the administrators private key.
- Display data in a meaningful way for the administrators.
- Allow administrators to query this data.

### Advertise to the User

This is really only to show a proof of concept as no real advertising will be done.

- Push advert to IPFS and store the hash in the blockchain network.
- Retrieve advert IPFS hashes and display these at set points throughout the users homepage.

The following is an outline of the sprints for the implementation of this project. Each sprint will meet a MVP and will incorporate the full two weeks. This plan is likely to change as the agile process means that it is hard to predict what each sprint will look like however this is just a general guide.

As you can see below there is 6 sprints outlined. For the implementation of "Turn on/off data gathering" will take 2 sprints as there is so much to implement within it. However this is acceptable as a it is so feature rich that a minimum viable product will be produced at the end of each sprint.

### Sprint 1: 28th January - 11th February

- Create new blog.

**Sprint 2: 11th February - 25th February**

- View blogs.
- Support a blog

**Sprint 3: 25th February - 11th March**

- Follow a blog.
- Share a blog.

**Sprint 4/5: 11th March - 8th April**

- Turn on/off data gathering.

**Sprint 6: 8th April - 22nd April**

- View user data.
- Advertise to the user.

## 4.5 Evaluation

As a form of agile methodology will be used for managing this projects development, progress can be easily measured through the success of each sprint. However even on a daily basis the progress being made on each feature can be evaluated leading to early detection of issues and how to resolve them.

The agile methodology will be enforced by using the project management tool Trello. This allows for tracking of events and making notes on issues so that no feature is forgotten about. At the beginning of each sprint, which will be 2 weeks long, features will be taken from Trello and pulled into implementation. These features together have to represent a MVP that can be presented at the end of each sprint, building up to the final project. By aiming for a MVP we can demonstrate more easily the progress of the project and have better successes.

Another measure of the success of this project will be through test driven development (TDD). TDD is a software development practice that involves testing your code consistently as you write it. This allows for further proof that the code functions as should

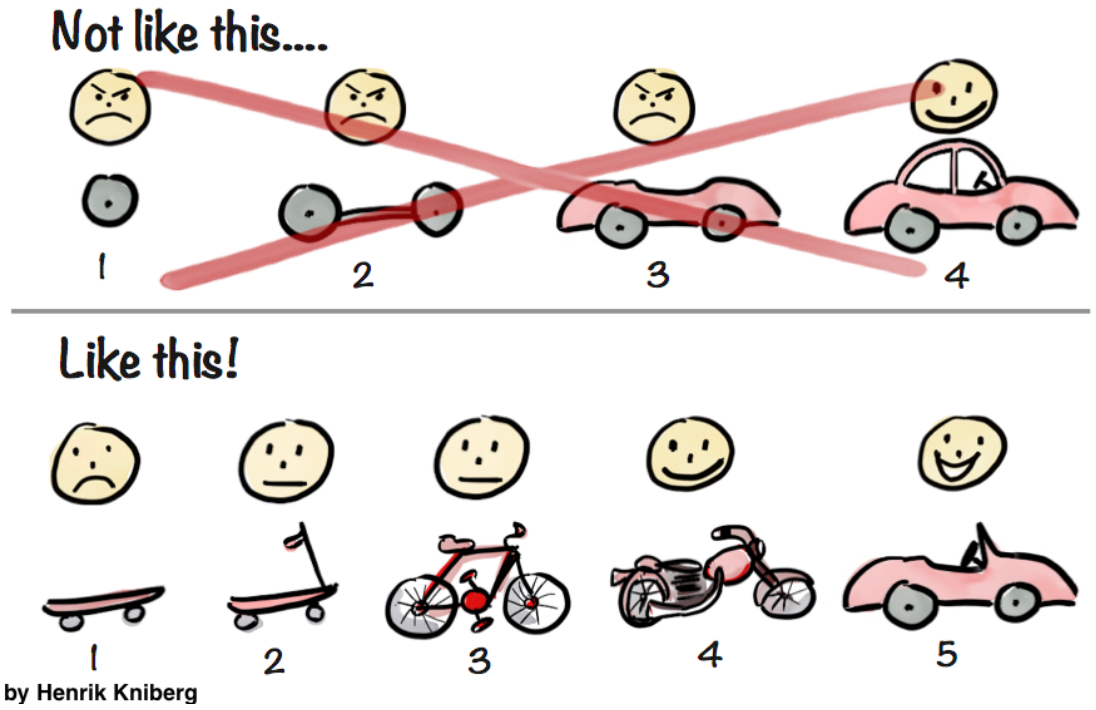


FIGURE 4.7: Minimum Viable Product Development[34]

but also can handle issues and this methodology has shown to generate better code. The tests generated themselves can be used for an evaluation of the success of a sprint.

## 4.6 Prototype

This section includes wireframes that show a basic outline for how the site will look without containing much detail in terms of design but with more focus on the layout. The overall general aesthetic that will be used is a dark minimalist look as this not only makes the site easy to use but is also appealing to the eye as it does not overload the user with information.

There have also been prototypes implemented to test certain tools. This was done to test the features of IPFS and Ethereum and understand if the technologies would fulfil the needs of what was being developed. The prototype allows users to add a file to the IPFS network and store the IPFS hash in the Ethereum blockchain network. This can then be retrieved and viewed on a homepage. This proves the basic concept is a viable application.

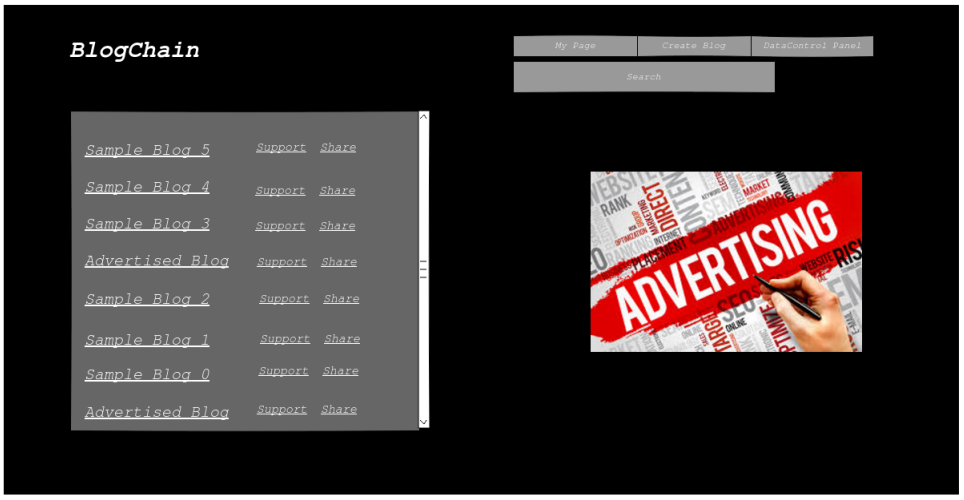


FIGURE 4.8: Homepage Wireframe

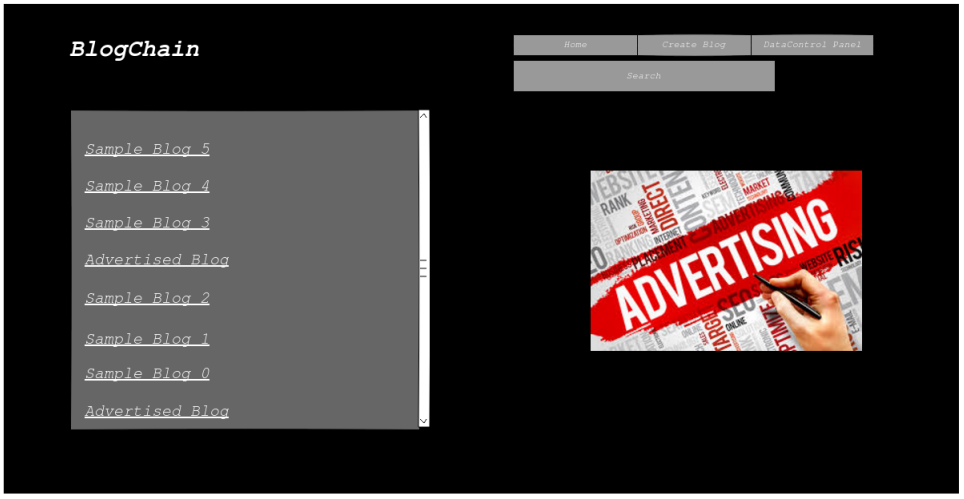


FIGURE 4.9: User Page Wireframe

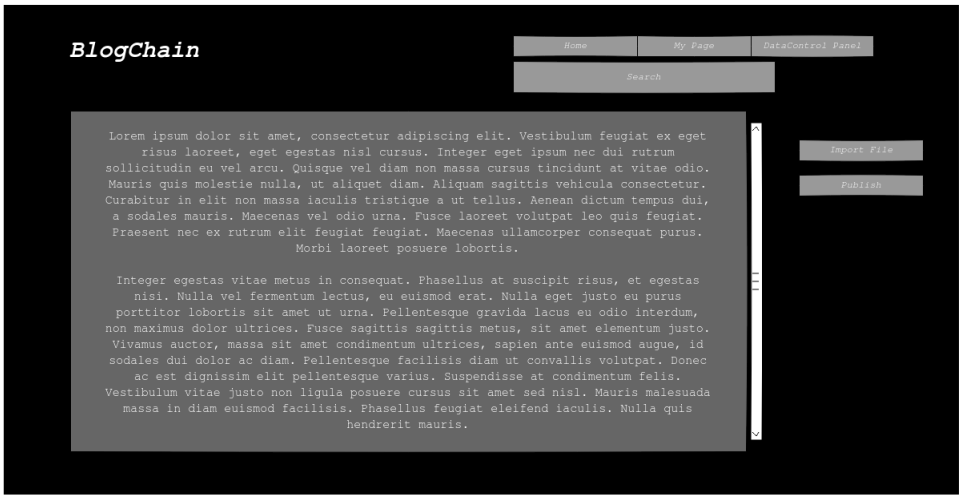


FIGURE 4.10: New Blog Wireframe

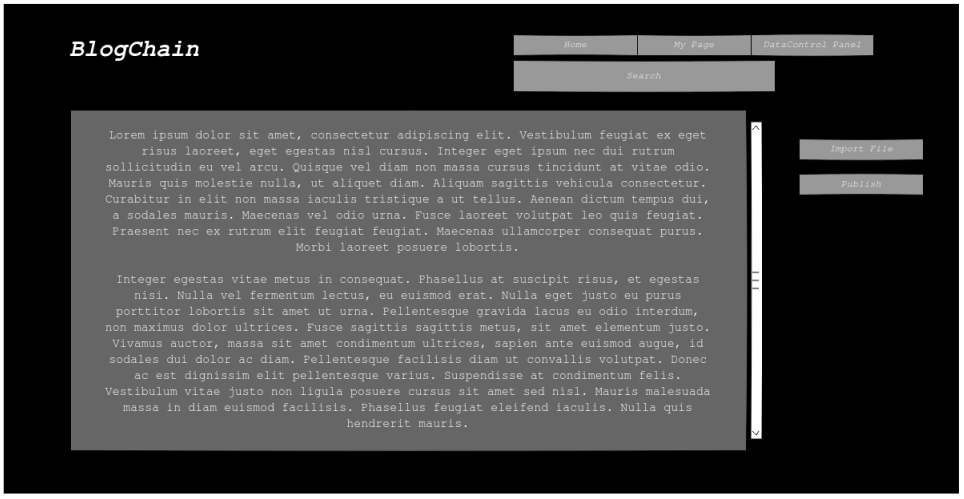


FIGURE 4.11: Data Control Panel Wireframe

# Chapter 5

# Evaluation

## 5.1 Discussion

Blockchain was a main driving force of the research phase, presenting a new perspective on solving issues, which at times, was challenging, due to its different approach to problem-solving, in comparison to more conventional technologies.

The two main features of blockchain that define it as potentially revolutionary technology are its immutability and smart contracts. There was a basic understanding of blockchain upon beginning the research phase; this knowledge grew significantly as the research progressed. As a result, it is now the opinion of the author that the proposed project did not recognise the full advantage of the potentials of blockchain.

The proposed solution of using a blog only leveraged blockchain to demonstrate that a blog post was not altered by a third party and to propose the use of tokens to support the content creators. To some extent, it showed that user data was not modified, but the design had flaws. The project did not contribute to the advancement of the understanding of the technology.

At the commencement of the implementation, the discovery that IPFS MFS was not as mutable and easy to implement as the documentation had implied was cause for concern. The MFS had been tested prior and showed to work however the feature appears to have been removed with little notice from IPFS. The documentation also outlined the ability for a system similar to git versioning for files; however, this feature did not seem to be present in the actual product making IPFS unusable for its intentions.

## 5.2 Conclusion

Given this reflection, the project will focus in a new direction, one that will contribute to blockchain technology and display its capabilities in a meaningful way. Research was conducted into industries that could benefit from the advent of blockchain technology.

Looking at what areas academics and industries are seeing as potential use cases for blockchain has inspired ideas that have lead to a project that makes use of the features of blockchain while also contributing to its advancement.

## 5.3 Project Proposal

A mature industry with little growth that could realise the benefits of blockchain technology is insurance. Insurance companies handle massive amounts of data, which continues to grow. Finding what data is useful can be difficult using conventional methods.

Blockchain could revitalise this industry bringing efficiency, security and cost reducing solutions. The benefits would be nondiscriminatory, profiting both insurers and policyholders. Policyholders could have more personalised insurance and competitive prices, whereas insurers could benefit from validated data and the automation of slow and laborious processes, such as claims, into more efficient methods.

This project will look at the use of smart contracts and verified data to produce a risk analysis system that is more cost-effective, efficient and secure.

All data relating to a policyholder, such as claims, will remain in possession of the policyholder even as they change insurers. This data will be used to perform risk analysis of the policyholder when they are looking to buy an insurance policy and give them a score that insurers can use to make informed insurance offers.

The system will provide those looking for insurance to 'auction' their needs to the pool of insurers. Insurers would then propose a cost and contract, based on the risk analysis score and insurance requirements. The advantages of this application allow insurers to make more knowledgeable and risk evasive decisions, while also providing the policyholders with more competitive choice and personalised insurance offers.

Beyond risk analysis, this data will be used to allow insurers to query a larger sample size of data outside of their policyholders. Doing this will allow the insurers to gain a stronger insight into the market and trends. This would be an exceptionally beneficial tool for a smaller business who may not have sufficient data themselves to form proper conclusions.

The key to making this element successful will be a permissioned blockchain that does not expose the personal information of the policyholders in other companies but does allow querying. This mass sharing of data amongst companies without the risk of exposing the policyholders would be extremely beneficial to understanding and predicting trends within the market as well as giving companies a larger sample size of data and reducing the margin of error for statistical analysis.

## 5.4 Contribution

The contribution to the insurance industry will automate the task of risk analysis performed by an actuary, increasing efficiency. It also will reduce fault in the industry by using blockchain as a method for verifying data as valuable and trustworthy, removing the potential for human error.

This system is not just limited to the insurance industry, as the technological advantages of data sharing using permissions is applicable to numerous industries. The contribution extends into the field of blockchain technology as it hopefully furthers its appropriation within industry and also data analytics.

# Chapter 6

# Related Work

## 6.1 Thematic Area within Computer Science

This project is a blockchain-based, insurance network, with a front-end web application that interacts with the blockchain network. The core functionality of the application is to perform smart risk assessments of assets for insurance companies and to allow these companies to offer contracts accordingly.

This high-level description of the core topic allows for it to be evaluated and categorised within the field of computer science. The most obvious of these areas are web applications, as the core interaction with the project will be through a web application.

Blockchain in itself can be categorised as a relatively new area within computer science. However as a blockchain network is a distributed computing network, it could be apt to consider it a new field within networking as well.

## 6.2 Current State of the Art

This section looks at published research into similar areas. The current state of the art provides a means for comparing the project and its feasibility based on what is already published. Academic research will also reveal if better methodologies and tools can be taken advantage of, helping to define the details of the smart risk analysis application.

An IEEE published paper[35] discusses how blockchain has the propensity for innovation and the ability to revolutionise many industries, rather than just being used as a cryptocurrency, particularly Bitcoin. This project, like many others, aims to expound the

countless ways blockchain applications are not limited to cryptocurrency. Blockchain offers a secure way to exchange any good, service, or transaction.

As determined in the earlier stages of this project, blockchain is not suitable for all applications and probably will not replace ones like Airbnb or Uber.

The issue of speed is being addressed by stepping away from the original POW, as can be seen with Ethereum[20], however, this "makes cryptocurrency platforms step away from their original purpose and enter the domain of database-replication protocols"[36]. These systems do solve the issue of scalability by increasing the number of transactions per second dramatically. Though the applications of these systems are still limited in use, due to security concerns with their visibility to the public and the trust they give to pseudonymity. This visibility is also how blockchain achieves trustless consensus and though while not suitable for all applications, blockchain certainly has the ability to bring advance improvements to specific sectors.

To understand the uses of blockchain, first, it is necessary to enumerate its distinguishabilities. Blockchain networks, in the simplest terms, allow for information to be distributed, but not copied. For example, if a user has ten bitcoins, no other user can copy those coins or obtain these except through lawful transaction on the bitcoin network. The records stored on the chain are immutable and no single entity controls these records.

A prominent feature of blockchain is smart contracts, which are "the rules that participants have collectively agreed upon to govern the evolution of facts in the distributed ledger"[37]. In layman's terms, this is code that the network must abide by for all transactions and interactions with the network.

Many have explored blockchains potentials in advancing the finance[38], IOT[39] and medical[40] fields with promising applications. These applications take advantage of its consensus, immutability, management of distributed systems, and so on. As finance and medical fields generally handle large amounts of sensitive data, the applications can increase the validity and relevancy of data however security is a concern. This issue of the privacy and security of data is being tackled by private blockchains.

Public blockchains use transparency for their trust and this system allows anyone on to query all transactions on the network. There are very apparent security concerns surrounding such an application if these transaction were to contain sensitive information. Private blockchains can add further security, increase transaction speeds and bring privacy. Private blockchains "aim to disrupt applications which have so far been implemented on top of database system"[41]. These applications could be in areas such as finance, medical and insurance where privacy is a requirement.

The insurance industry, as a sector that handles large quantities of data that require verification, is starting to explore the potentials of blockchain[42][43]. Blockchain can potentially reduce fraudulent claims, increase claims processing speed, and perform more accurate risk analysis faster.

Smart contracts can be used to automate and bring transparency to processes. Risk analyses, performed by actuaries, are used to calculate the financial consequences of an investment. Because risk analyses are so subjective, as they are the result of data analysis, they are prone to fault, as the data provided may be inaccurate. This manual task is quite laborious and time-intensive as well. If Blockchain networks were used to validate and store the data, as well as perform risk analysis using the data, the accuracy would increase, while the resources needed would significantly reduce.

Another use case is the application of smart contracts for handling claims, for example, travel insurance. If a flight is cancelled, the smart contract automatically triggers the full payment of the claim[44]. The smart contract reduces the overhead required for the customer in the claim and the claims department in verifying the claim. It can also help reduce fraudulent claims, as the integration of systems would lead to more reliable and verifiable information. The savings of such a system could be passed onto the policyholders[44]

IBM was involved in the creation of an insurance-based application using Hyperledger. OpenIDL is the first blockchain platform aimed at streamlining regulatory reporting. It enables the efficient and permission-based collection of statistical data on behalf of insurance carriers, regulators and other participating contributors.

Hyperledger Fabric[45] is a permissioned blockchain network, unlike its counterparts bitcoin[7] and Ethereum[8]. A network built on Hyperledger validates authorisation in the protocol. Hyperledger also provides privacy of transactions, even when it is still in the process of validating them. It does this by ensuring only the parties, who are allowed access, have access to the data, while still including the validators.

In comparison to Ethereum, Hyperledger "Fabric intends to provide a modular and extendable architecture that can be employed in various industries, from banking and healthcare over to supply chains. Ethereum also presents itself as utterly independent of any specific field of application. However, in contrast to [Hyperledger] Fabric, it is not modularity that stands out but the provision of a generic platform for all kinds of transactions and applications"[46].

Hyperledger Fabric does not require the entire network to know the details of a transaction - only the participants of a transaction are required to participate in its consensus, whereas Ethereum involves the entire network for its consensus[47][48]. By involving

only the required participants in a transaction, Hyperledger offers enhanced privacy and also increases the speed of the transactions. The consensus in Hyperledger Fabric involves "the entire transaction flow, from proposal and endorsement, to ordering, validation and commitment"[48]. Hyperledeger Fabric offers more fine-grained control over the consensus in the creation of blockchain networks, ultimately improving performance and privacy of the network, especially compared to Ethereum.

Another benefit of Hyperledger is that it does not have a "systemic dependency on a native cryptocurrency"[49]. Without this dependency, developers have more freedom to design applications. Moreover, Hyperledger eradicates the deterrent requirement to pay for transactions on the network: on Ethereum, users of the system must pay in Ether or the applications' provided cryptocurrency for all write transactions to the blockchain network. Though Hyperledger does not have a dependency on cryptocurrency, it does have the ability to support one.

Finally, the latency of transactions is not a concern, with fabric boasting more than 3500 transactions a second[49], whereas we currently see about five transactions a second with Ethereum[10]. The disparity in transactions per rate is stark. In terms of scalability, five transactions a second is extremely slow, thus Ethereum is only suited for specific applications. However, the rate of transactions per second should improve with their proposed Casper protocol, which uses a different method, known as proof of stake, to validate transactions. For now, Hyperledger is undoubtedly the more scalable solution available.

The privacy and control that Hyperledger can offer over Ethereum make it more attractive for an insurance-based application, where the privacy of customers and their data are a concern. Privacy is a considerable concern for the proposed application, where all data and claims are accessible on the chain, but limited to those with permission. As a permission-based blockchain, Hyperledger is the clear choice, especially as it is already being tested in the insurance industry with OpenIDL.

Blockchain is still a very new area, with its potential yet to be fully realised. The insurance industry has been stagnant for some time and could reap the benefits of a blockchain system, creating a more efficient and cost-effective operation. Blockchain smart contracts could be used to provide immutable and more personalised contracts for customers looking for insurance, providing them at a much faster pace. Efficiency could be improved, being particularly attractive to customers seeking micro-contracts, such as short-term holiday insurance or weekend car rentals. Blockchain can handle these use cases without the overhead of human paperwork which is also prone to error.

The question, "is blockchain mature enough for insurance applications?", has been asked in many academic papers[43][50]. The technology is still new and, indeed, some advances need are still to come, but blockchain cannot mature unless it is tried, tested, and iterated, as is true for any technology if it is to grow. This is an opportunity for the insurance industry to grow. Many innovative, fintech start-ups companies initially posed a threat to traditional financial services however, these companies now represent the future of the financial sector.

## 6.3 Research Question

This thesis is exploring how blockchain can be used in alternatives ways to cryptocurrency and can the currently available technologies meet the demands of these systems. The system described is focused on risk analysis within the insurance industry, but numerous applications could appropriate the core concepts.

Risk analysis is prone to fault due to human error and a potential lack of valuable data. The blockchains ability to validate data will be used here to provide data which can be trusted for the automation of risk analysis. The risk analysis itself will be automated by utilising smart contracts that check this data to provide a trusted analysis of policyholders. The challenge here is can data be stored within a permissioned blockchain and shared securely?

This is where the core concept can crossover to numerous applications. Sharing data without exposing all of the sensitive information is an application that could be beneficial to a plenitude of industries. In this application, all the insurers within the network access their policyholders' data on the same network with access control lists dictating what they can access. Designing the access control lists in such a way that only specific data that would not be personally identifiable is shared with specified insurers would allow these companies to query significantly larger sample sizes. This would be particularly beneficial to newer companies who may not have a large enough sample size of their own to obtain accurate market predictions.

Insurance is ideal for challenging this idea as it is heavily dependent on large amounts of data to make informed decisions. This could actively change market research. Addressed previously in section 6.2

# Chapter 7

# Proposed Solution

## 7.1 System Overview

The system will consist of two participants: policyholders and insurers. Policyholders are an individual who is looking for insurance or has insurance. The insurers then are the companies who insure these policyholders.

Policyholders will retain ownership over all data relating to them and will carry this profile with them between insurers. The aggregation of data over time will lead to more accurate evaluations of the policyholder when performing the risk assessment.

The risk assessment will be implemented using a smart contract which is automatically triggered when an asset is seeking insurance. The smart contract will access the data of the policyholder to perform an analysis, and the result represented through a score. This score, visible to insurers, will be used to make a more risk-averse insurance offer to the potential policyholder.

Insurers can form agreements for data exchanges to increase the sample sizes they have for performing data analytics. Data exchanges over the blockchain network could be particularly useful to industries who may want to expand their operations into another sector but may not have the required data to do so. Let us take for example 3 independent insurers referred to as insurer1, insurer2 and insurer3. Insurer1 is a significant health insurer. Insurer2 is mostly car insurance but also does a small amount of health insurance. Insurer3 is another much smaller health insurer who is new to the market. Insurer3 and Insurer2 establish an agreement to share their health insurance data, increasing the amount of data they can access. This agreement increases the statistical

accuracy in terms of percentage points for insurer2 and insurer3 when it comes to forming conclusions about the health insurance market, giving them a stronger chance to compete successfully with health insurer1.

### Functional Requirements

The following is the outline of the requirements for each participant of the system. These requirements will be used as a means for comparison to the final product and its success. As a policyholder I want to:

- Add new assets to get insured
- Open an asset to receiving new insurance offers
- Accept and decline insurance offers
- Remove an asset
- Make claims for an asset
- View data associated with an asset
- Carry this data with the asset to new insurers

As an insurer I want to:

- View insured assets and private individuals
- Add data relating to an asset
- Approve/deny claims
- View claims
- View assets open to being insured
- Make offers of insurance to potential policyholders
- Perform queries on my own policyholders/ assets
- Perform queries on all assets/ policyholders within the system

### Non-Functional Requirements

The non-functional requirements of this system are as follows:

- Risk analysis will automatically perform when a new asset has been uploaded
- Risk analysis will automatically perform when an asset becomes available for insurance offers
- Insurance offers will only be visible between the insurer and the policyholder
- Claims will only be visible between the insurer and the policyholder
- When an insurance offer is accepted all other offers for that asset should be automatically rejected
- A write to the network must persist the data before another write can occur for that data
- A transaction will take under 1 second to complete
- Data should only be directly accessible and viewable by those with permission

Risk analysis will be a smart contract that looks at the requirements of insurance for an asset and the information of the individual and produce a score for the asset based on this assessment.

The value of the system to the policyholder is they can receive many insurance offers and accept the best one for their needs creating a more competitive market. As well as this they get to carry the information relating to claims and their assets with them as they transfer between insurers. This system would reward people with a good history.

The insurer has access to a verified history of claims and data for the policyholders they are ensuring. They can also rely on the risk analysis smart contract to automate and produce an accurate risk analysis of an individual based on their history and the assets, allowing the insurer to make smarter "investments". By using this system, it removes the need for insurers to gather data on their policyholders. All insurers also have a high-level overview of the other insurers giving them the ability to query outside of their own market.

The goals of the system can be broken down into the following:

- Private individuals carry with them their insurance history
- Risk analysis is impartial and provides quick results using smart contracts

- Insurers can offer personalised contracts to individuals based on their insurance needs that are implemented using smart contracts
- Data can be shared amongst insurers without exposing personal information

The smart contracts are a distinguishing feature of this project that set it apart from traditional insurance methods which can be laborious.

The smart risk assessment will examine the policyholder's data such as frequency of past claims to give the policyholder a risk score. In future implementations and research, this could be expanded to include machine learning.

The risk score will allow the insurer to make smart offers based on the risk assessment and requirements of the customer.

All of this makes for a more efficient system which requires less human interaction and will also reduce errors caused by people as the smart contracts are following a set of rules which cannot be changed and using data that is verified.

How policyholders carry their claims and data, is another feature that evolves this system from traditional insurance companies. Giving the user ownership of this data takes away the need for insurers to gather this data and validate it as the data has already been validated on the blockchain network.

The system will be composed of RESTful APIs which will allow policyholders and insurance companies alike to interact with the blockchain network. For this implementation, the potentials on both the policyholder and insurance end will be demonstrated.

## 7.2 Use Cases

This section outlines the use cases for this project which will be used as a standard to measure the success of the implementation of the project. Table 7.1 to 7.7 outlines use cases specific to each policyholder. The tables 7.8 to 7.11 are all insurance company related use cases that will be implemented for this project. The use case with the widest scope is 7.10 which is regarding the queries an insurer can perform on their policyholders' dataset, as there are several queries for this project that will require implementation and testing.

The three main use cases of this system are as follows:

- The automation of risk analysis within the insurance industry

- The control of data access using access control lists
- It can be an evolving risk analysis; Investors can see the kind of risks that insurers are taking and invest accordingly.

| Use Case Name | Add a asset |
|---|---|
| Actor | Policyholder |
| Description | A policyholder wants to add an asset to be insured |
| Flow of Events | • Policyholder selects option to add an asset<br>• Policyholder enters the information about the asset<br>• Policyholder submits the asset<br>• The asset is sent for risk analysis<br>• The asset then becomes open for bidding |

TABLE 7.1: Add a asset

| Use Case Name | Make a claim |
|---|---|
| Actor | Policyholder |
| Description | A policyholder wants to make a claim |
| Flow of Events | • Policyholder selects option to make a new claim<br>• Policyholder enters the claims data<br>• Policyholder submits claim |

TABLE 7.2: Make a claim Use Case

| Use Case Name | View claim status |
|---|---|
| Actor | Policyholder |
| Description | A policyholder wants to view the status of a claim they made |
| Flow of Events | • Policyholder selects the claims<br>• Policyholder navigates to the claim they are searching for<br>• Policyholder selects the claim |

TABLE 7.3: View claim status Use Case

| Use Case Name | Open to insurance offers |
|---|---|
| Actor | Policyholder |
| Description | A policyholder wants to open their profile to insurance offers |
| Flow of Events | • Policyholder changes status to open to offers<br>• The profile is analysed by the risk analysis smart contract<br>• The policyholder risk score is published to the network |

TABLE 7.4: Open to insurance offers Use Case

| Use Case Name | Open to insurance offers |
|---|---|
| Actor | Policyholder |
| Description | A policyholder wants to open their asset to insurance offers |
| Flow of Events | • Policyholder changes status to open to offers<br>• The profile is analysed by the risk analysis smart contract<br>• The policyholder's risk score is published to the network |

TABLE 7.5: Open to insurance offers Use Case

| Use Case Name | Accept insurance offers |
|---|---|
| Actor | Policyholder |
| Description | A policyholder wants to accept insurance offers |
| Flow of Events | • Policyholder can select the best insurance offer for them<br>• The alternative is the policyholder can set up a smart contract to select an offer that best suits their needs<br>• The Policyholder's status is set to closed to offers<br>• The insurance company is given permission to access the policyholder's data |

TABLE 7.6: Accept insurance offers Use Case

| Use Case Name | View data |
|---|---|
| Actor | Policyholder |
| Description | A policyholder wants to be able to view their data |
| Flow of Events | • Policyholder can select from a number of queries regarding their data such as view all<br>• The data is displayed in a human readable format |

Table 7.7: Accept insurance offers Use Case

| Use Case Name | Approve/deny claim |
|---|---|
| Actor | Insurer |
| Description | A insurer wants to be able to approve/deny a claim made by a policyholder |
| Flow of Events | • Insurer selects a claim<br>• Alternatively a smart contact could be implemented to approve/deny the claim<br>• Insurer changes the claims status |

Table 7.8: Approve/deny claim Use Case

| Use Case Name | Make insurance offers |
|---|---|
| Actor | Insurer |
| Description | A insurer wants to be able to make insurance offers to potential policyholder |
| Flow of Events | • Insurer can set up smart contract to make offers based on risk assessment<br>• Offer is made to policyholder based on their risk and insurance needs |

Table 7.9: Make insurance offers Use Case

| Use Case Name | Query policyholder data |
|---|---|
| Actor | Insurer |
| Description | A insurer wants to be able to perform queries on all policyholders' data |
| Flow of Events | • Insurer selects query and enters details for the query to perform<br>• Query details are returned in a human readable format |

Table 7.10: Query customer data Use Case

| Use Case Name | Write policyholder data |
|---|---|
| Actor | Insurer |
| Description | A insurer wants to be able to write data to policyholders profiles |
| Flow of Events | • Insurer selects policyholder for data to be added to<br>• Insurer enters the details of the data which is then added to the policyholders profile |

Table 7.11: Write customer data Use Case

| Use Case Name | Perform risk assessment |
|---|---|
| Actor | Risk Analysis |
| Description | Risk assessment must be performed on an newly open asset |
| Flow of Events | • Smart contract analyses the policyholders profile and provides a risk assessment score |

Table 7.12: Perform risk assessment Use Case

## 7.3 Architecture

Here we look at the proposed architecture for this project and discuss further in detail how it will be implemented. It looks first at a higher level overview of the whole system and then gets into the details of how communications and transaction in the system will take place.

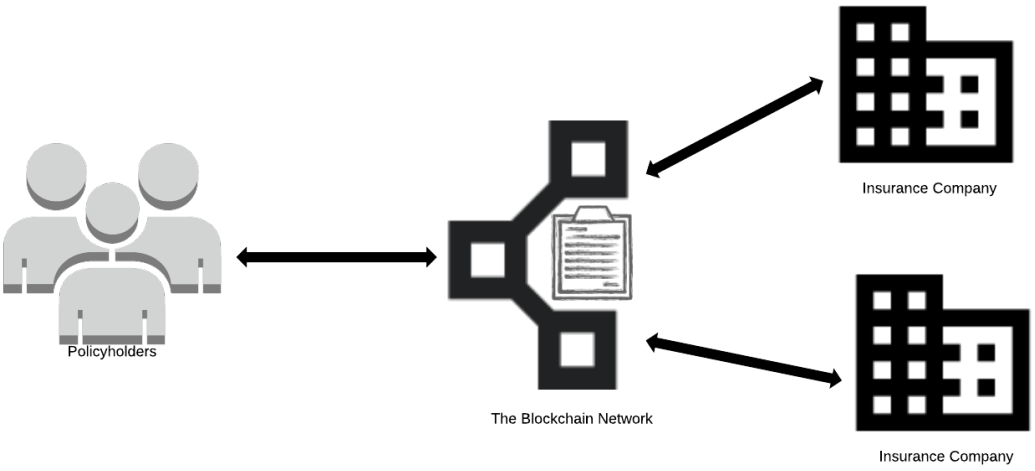


FIGURE 7.1: High Level Overview of the System

As can be seen in figure 7.1 the 2 participants of the system are policyholders and insurers. A RESTful API exposes the blockchain network, and the participants can interact with the network through a web application. All data is stored securely on the blockchain network where the access control lists determine who has access to what.

### 7.3.1 Policyholder

This section looks at the architecture of the policyholder. Figure 7.2 outlines the interactions the policyholder has with the system. A high-level description of the background tasks also involved in each of these that are triggered by these events is also outlined. These actions all involve either reading or writing to the blockchain network and are performed using a REST API that exposes the network through a web application.

All of these actions are verified on the blockchain using smart contracts. The policyholders and insurers may share access to some REST APIs, but for the most part, there will be no crossover. There will also be access control rules in place which uphold the permissions of each user.

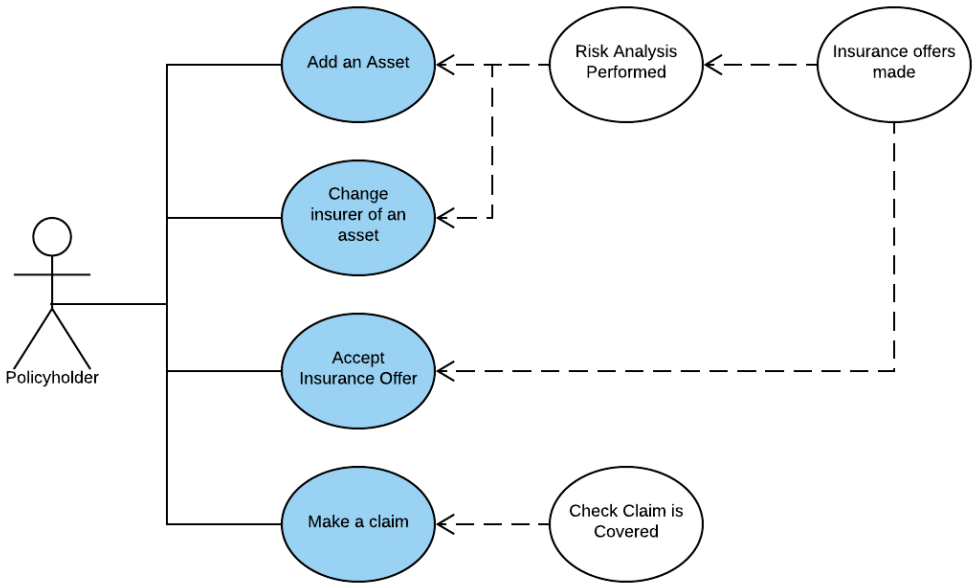


FIGURE 7.2: Policy Holder System Interaction

The following outlines the requirements of the Restful API for the private individual system:

| Query Name | CreateNewAsset |
|---|---|
| Method | Post |
| Description | This post request for creating a new asset to be insured |

TABLE 7.13: Create a New Asset Request

| Query Name | PerformRiskAnalysis |
|---|---|
| Method | Post |
| Description | This performs risk analysis of an asset and gives it a score |

TABLE 7.14: Perform Risk Analysis Request

| Query Name | CreateClaim |
|---|---|
| Method | Post |
| Description | This allows the policyholder to make a claim for an asset they have insured |

TABLE 7.15: Make a Claim Request

| Query Name | AcceptInsuranceOffer |
|---|---|
| Method | Post |
| Description | This allows the policyholder to accept an insurance offer |

Table 7.16: Accept Insurance Offer Request

| Query Name | selectOpenOffersToIndividual |
|---|---|
| Method | Get |
| Description | This returns all open insurance offers to a specified policyholder |

Table 7.17: Select Open Insurance Offers

| Query Name | selectAssetsByIndividual |
|---|---|
| Method | Get |
| Description | This returns all assets belonging to a policyholder |

Table 7.18: Select All Assets

| Query Name | selectClaimsByIndividual |
|---|---|
| Method | Get |
| Description | This returns all claims belonging to a policyholder |

Table 7.19: Select All Claims

### 7.3.2 Insurer

The insurer side of the application much like the policyholder is a web application that exposes RESTful APIs to communicate with the blockchain network. A high-level overview of this interaction is outlined in figure 7.3. The smart contracts have to been designed in such a way that they allow input that can vary from insurer to insurer while still retaining the integrity of the system. There can be no queries that are only available to one insurer, but instead, all insurers will have access to the same RESTful API. However, the data they can access will be dictated by the ACL (access control list) to ensure they are only performing CRUD actions on which they are permitted.

The following outlined the insurer application requirements for the REST API:

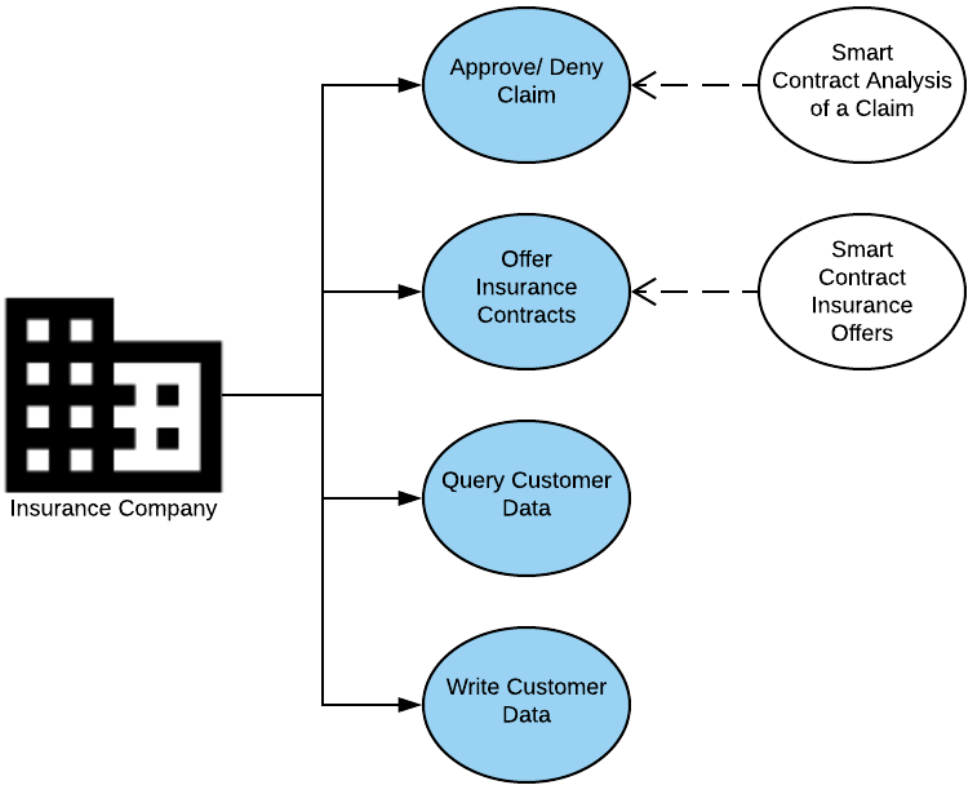


FIGURE 7.3: Insurer System Interaction

| Query Name | claimsDecision |
|---|---|
| Method | Post |
| Description | This decides the writes the results (approved or denied) of claim. |

TABLE 7.20: Approve/Deny a Claim

| Query Name | makeInsuranceOffer |
|---|---|
| Method | Post |
| Description | This writes a new insurance contract and offers this to a policyholder |

TABLE 7.21: Make Insurance Offer

The insurance side will want to be able to perform a wide variety of queries on the data of their policyholders and those who are not. This will require developing a flexible smart contract that can interpret the input of the insurer and return the results accordingly. There must be an agreement between all insurers to share the same type of data.

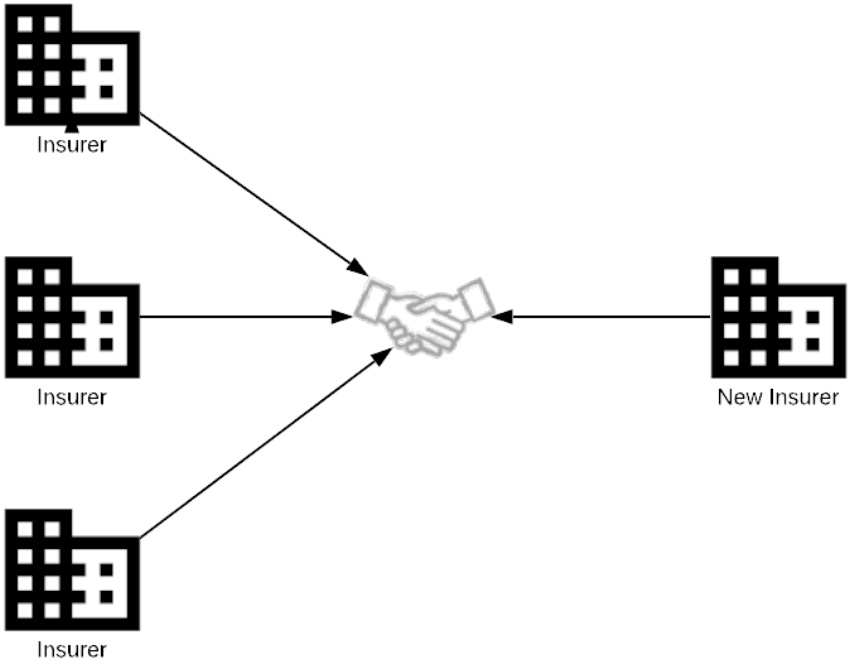


FIGURE 7.4: Data Exchange Agreement

Insurers may also want to form arrangements with certain parties on the chain whereby the exchange more than the agreed upon amount amongst all parties. This could involve smart contracts to initialise this arrangement ensures that the terms of data exchange are met.

## 7.4 Technologies Employed

**Hyperledger Fabric** - This is a blockchain framework and is one of the Hyperledger projects hosted by the Linux Foundation[51]. It was chosen as it offers a private blockchain with excellent controls over permissions for access to all aspects of the blockchain network. It provides a flexible framework that can be used to develop directly to the requirements of the application.

**Hyperledger Composer** - Hyperledger Composer is a set of collaborative tools for the the development of blockchain networks[52]. It makes the development of applications for Hyperledger Fabric simplier by leveraging tools such as npm and node.

**JavaScript** - Hyperledger Fabric smart contracts are written in JavaScript. As well JavaScript is a powerful language with many frameworks for web development, choosing

this as the main language for web development as well as smart contract development will add consitency to the development languages used.

**React** - React is a JavaScript user interface library developed by Facebook[53]. The library allows for the development of powerful frontend applications with simplicity.

**Node.js** - Node.js is a JavaScript runtime environment that executes code outside of the browser[54]. The Hyperledger Composer JavaScript SDK is a set of Node.js APIs that enables developers to create applications to manage and interact with deployed blockchain networks[55].

**Representational State Transfer (REST)** - Hyperledger Fabric easily integrates with REST when developed using Hyperledger Composer which turns the queries into a RESTful API.

## 7.5 Methodology

Many new technologies are appearing in this application that have not been used previously by the author and so will require research and upskilling before implementation can occur. The application can be broken down into two subsections, the blockchain network and the web applications.

The blockchain network is hugely different from the previously proposed project in chapter 3 as it is a permission private blockchain network. The network will be built using the Hyperledger Fabric framework. Unlike Ethereum network it is a private blockchain network that allows permissioning and access control. Hyperledger Composer is a tool that will be utilised to implement the Hyperledger Fabric blockchain network.

The model file is the underlying code which defines the domain model for the network. All users, assets and transaction attributes are defined within this file.

The permissions file implements the rules that all users of the network must abide by and will be utilised to uphold who has access to what data and transactions.

The queries file, written in JavaScript outlines all the queries to the blockchain network. The permissions file limits access to these queries and the data returned by them. The queries language resembles SQL and so will not require much training to become familiar with it.

The smart contract is the logic for the network. This file is also written in JavaScript, and it implements the logic for each transaction and interaction with the network.

The web applications will access the blockchain network over the RESTful API. Both of these applications will be written using ReactJS which is a library for building front-end web applications and will be similar to each purely for ease of development of the demo.

The new features that require upskilling are ReactJS, Hyperledger Fabric and RESTful APIs. While this may appear to be a significant proportion of the project ReactJS is written in a variation of JavaScript which is JSX. Having already worked with JavaScript previously should assist with understanding how to use the ReactJS library. There are many great resources for tutorials on using the tool on the official site of ReactJS as well as blogs and YouTube.

As regards RESTful APIs concern with understanding how this technology works is not of too much importance as much of its implementation is taken care of by the Hyperledger Composer tool. This allows for a stronger focus to be placed on developing the blockchain network.

Hyperledger Fabric has many tutorials and detailed documentation. Hyperledger Composer eases the amount of work that must be performed for developing a network using Hyperledger Fabric. This is quite a detailed and robust framework, and so it does require extensive upskilling to become comfortable with using it. By completing the tutorials, it will give a firm base knowledge of the tool, and that will suffice for the development of this project.

As previously outlined in chapter 4 the development approach that will be taken is agile. The reason being is that it allows for flexibility in development and each implementation cycle will produce a minimum viable product that can be demonstrated to show progress. The project supervisor will partake in weekly meetings to discuss the project, difficulties and review overall progress while guiding how best to approach problems. This will have similarities to a Scrum methodology whereby the supervisor is the Scrum Master and Product Owner.

## 7.6 Implementation Plan Schedule

As the project has been redefined the time allotted to implementation has been reduced greatly and so what time remains after the research phase must be used wisely as well as implementing what is feasibly possible within this time.

### Blockchain Network

The blockchain network will be the most time-consuming element of the project. It will consist of the following:

- Define users, transactions and assets of the network.
- Add queries that can be accessed by the smart contract and over the RESTful API.
- Smart contract functionality for adding a new asset to the blockchain network.
- Smart contract functionality for performing risk analysis on an asset.
- Smart contract functionality for making an insurance offer to an individual.
- Smart contract functionality for accepting or rejecting an insurance offer.
- Smart contract for creating a new claim for an asset.
- Smart contract for accepting or rejecting a claim for an asset.
- Create the login functionality for each user type.
- Add the access control rules to the permissions file for each type of user.

### Policyholder Web App

This section will involve developing the front-end application to display information from and allow interactions with the blockchain network for private individuals.

- Connect with the blockchain network over the RESTful API.
- Retrieve information relating to a specific individual.
- Add a new asset to the blockchain network.
- Display and accept or deny insurance offers.
- Create a new claim.
- Add the login feature for the policyholder.

### Insurance Company Web App

The insurance company web app will have similarities to the policyholder web app and will be built in conjunction with it to be able to continuously test the interoperability between the two and deliver on the minimum viable product each time.

- Connect with the blockchain network over the RESTful API.
- Retrieve information relating to assets open to insurance.
- Display information relating to assets that are insured by the insurance company.
- Make insurance offers to open assets.
- View and perform actions on claims.
- Add login feature for the insurance company.
- Add dynamic querying contract
- Add access control lists for data
- Enable the ability to share more data with certain insurers

This plan has been informed by implementing sections of the project while research occurred and some parts have already neared completion. The development of the web applications will take place on a reasonably simultaneous basis as they both rely on each other significantly for receiving information and performing actions.

## 7.7 Evaluation

While the agile approach inherently is a measure of the progress a project is making due to the minimum viable product delivery for each sprint it does not guarantee that the project is functioning as should.

To ensure the functionality is being achieved a means for testing the performance of the project is required. This should demonstrate that each element is working as should and handles errors appropriately. As the project is relying on RESTful APIs for exposing the blockchain network, it would be appropriate to use tests of these as a good measure of success for the project.

Postman is an API development environment that can be used for testing RESTful APIs. It allows for quick and responsive testing of the network that will show if functions are performing as they should be similar to how unit testing works.

# Chapter 8

# Implementation

This section documents the development process and looks at difficulties encountered and changes that have been made to the project.

## 8.1 Introduction

This project undertook a large reevaluation at the beginning of the implementation phase resulting in delays of beginning the implementation. This reevaluation was necessary due to issues encountered early on in the development of the decentralised blogging application.

As a result of this change the period of allocated time that would have predominantly been used to implement the project originally outlined now involved the conduction of the research phase again and then an implementation however this research phase could carry over a lot of what had previously been learned in terms of blockchain technology. These changes are discussed in chapters 5 through 7.

The major changes to the project are that it is now an insurance application built on blockchain technology using smart contracts to provide more personalised insurance contracts to customers. This resulted in the following changes to technologies and applications being used:

- The blockchain network will be created using a private blockchain network on Hyperledger fabric, and so Ethereum will no longer be used.
- Two web application will be implemented using React.

- The Hyperledger network will be exposed using a RESTful API that the web applications connect with to communicate over the network.
- There is no need for a decentralised storage system, and so IPFS will be removed entirely from the project.

## 8.2 Difficulties Encountered

The first difficulty encountered was the mutuable file system and the git like versioning of files being available on the IPFS API. There appeared to be little to no information regarding why these features were unavailable however a footnote on a GitHub merge request suggested that documentation should be updated to reflect that the git like versioning was unavailable. As for the mutable file system, some stack overflow answers suggested that it was due to ongoing updates within the system as to why it was currently unavailable.

These issues put the project into immediate jeopardy, and so required a substantial rethinking of the project and if it was still feasible to complete. The result was it would not be suitable to continue on this path, and so the project was shifted in another direction which is discussed in chapters 6 through 7. The result is a major delay at the beginning of the implementation phase however a much more valuable project is produced.

Once this was overcome and the project redefined, the implementation phase continued with the newly outlined project. This project encountered several difficulties; some were the fault of the author's lack of knowledge and others were outside forces the following lists these issues, the difficulty to resolve them and the solution if any was found.

### Installing Hyperledger

The first difficulty encountered with the newly defined project was setting up Hyperledger on a local machine. When attempting to install certain parts did so without issue; however, the main Hyperledger environment itself kept running into an error. The issue and its solution were unclear from the error thrown. If this were not solved, it would have meant further research into another technology and an even greater setback on the implementation of the project.

| | |
|---|---|
| Resolution Difficulty | Easy |
| Solution | It required removing all instances of the successfully installed applications and doing a clean reinstall. |
| Affect | The only affect of this was a mild setback starting implementation |

### Running Hyperledger

When first running Hyperledger there was an issue when a new command line tab was opened to perform another operation. This issue was a result of Hyperledger only working with Node version 8.9 which was set when starting up the command line window before undertaking any operations however even though Node version 8.9 had been set in the first tab, it did not carry over to the new tab. Again the error thrown here did not indicate an error with Node version. The issue was discovered through testing.

| | |
|---|---|
| Resolution Difficulty | Easy |
| Solution | It required ensuring that the correct version of Node was in use whenever a new command line tab was opened. |
| Affect | The only affect of this was a mild setback regards time |

### Integration of RESTful API and ReactJS

This issue arose out of a lack of familiarity with the two technologies. Both of these technologies were being learnt as the project was being implemented. It was unclear at first how to integrate the web frontend with the blockchain backend. This represented a risk to the design of the project if this integration could not be understood and implemented.

| Resolution Difficulty | Moderate |
|---|---|
| Solution | Various tutorials were undertaken to understand both REST and React separately first and then as knowledge was built on the technologies individually, tutorials were undertaken on the integration of the two technologies |
| Affect | This resulted in a few days where there was a lack of implementation however it did not dictate any changes to the design of the project |

### Persistence of Data

When accepting an insurance offer, there was an issue with persisting this to the blockchain network before running the transaction to update the other insurance offers as rejected. The result varied from all insurance offers being rejected, only the accepted offer would update, and no offers would update.

| Resolution Difficulty | Easy |
|---|---|
| Solution | A sleep function was defined that took a 3-second break between running both functions allowing time for the blockchain network to update |
| Affect | This was just a minor inconvenience that would have been better resolved had the knowledge of the technologies being used stronger |

### Linux System Failure

Ubuntu was the operating system being used for project development. This machine, for unclear reasons, was becoming noticeable slower until the system failed. Debugging was performed to try and identify the root cause of the slowdown and ultimately failure of the system however it was never identified. This could have been significantly worse had the project not been regularly pushed to Github.

| | |
|---|---|
| Resolution Difficulty | Moderate |
| Solution | A clean reinstall of the entire OS and software required for the development. The project was mostly backed up on Github and so very little was lost |
| Affect | Between debugging and setting up the development environment it took some valuable time away from the project. This set back came at the inconvenient time of the final week of project implementation |

### Login Feature

It was hoped that a login feature would be added to project however time was limited and so it was not possible to implement this successfully.

| | |
|---|---|
| Resolution Difficulty | Easy |
| Solution | It was decided to remove this feature from the project and add it to future development. The login details are hardcoded for demonstration purposes |
| Affect | This resulted in removing the feature from the project and adding it to future development |

### Queries

Running queries to get specified information such as a particular users ID was not possible, you could only access the entire object. For example searching for the IDs of users between 25-30 was not possible but you could return the whole object. It was successfully tested in a controlled test environment outside of the web application.

| | |
|---|---|
| Resolution Difficulty | Easy |
| Solution | This meant writing smart contract logic that would remove all the unnecessary information from such queries |
| Affect | This resulted in more smart contract code being written |

## 8.3 Actual Solution Approach

As is known this project has changed dramatically from that described in the first four chapters. The real solution and the issue being addressed are no longer the same as that first described, and these changes have been outlined in chapters 5 to 7. However, even the solution outlined in chapter 5 onward which better reflects the actual project has been changed somewhat.

The solution was dictated by time, and so some features have not made it into the final project. As outlined above one of these is the login feature. While this is a necessary feature on a final product the time was not there to implement and test it successfully. This, unfortunately, means that the product has hardcoded values for its demonstration.

Another feature mentioned in the non-functional requirements and several times throughout the document is the use of the access control list. The access control list was going to dictate who had access to what data and how data was queried, unfortunately, the time was not there to implement this fully. On the web front-end, users are only displayed information that they are meant to access however if they were to use a REST call in a way they were not meant to they would gain access to data they are not meant to. This means that instead of the access controls list the REST calls are specified to query based on the user. However, this feature was implemented successfully outside of the web application in an environment accessible to administrators for each insurer only. Data was successfully shared and secured amongst the administrators of each respective insurer. It would required more time to be successfully integrated into the web application.

The most unfortunate feature that would have been very exciting to implement was the ability for dynamic querying of data. This feature required significantly more time than was allotted to the project. This feature would have ultimately lead to insurers defining their own queries.

The actual project is still presentable and the features above that have not been added is not due to an inability with the technology and is purely because of time constraints. All of these features will be carried into future development of the project.

# Chapter 9

# Testing and Evaluation

## 9.1 Metrics

The system will be evaluated using Postman which is an application that can be used to test RESTful APIs. The tests are looking to examine the following of the system

- Does the REST operation perform as should, i.e. correct response code, correct data returned and so on?
- Do the smart contracts respond appropriately to the conditions met and is there any unhandled error paths.
- How does the REST operation handle unexpected data or events occurring, i.e. Error handling?

As well as doing software tests there is a need for comparison of the actual project to the finished project. The functional and non-functional requirements of the system that were outlined in 7 will be used as a means for comparison. This will identify if the vision of the project reflects the implementation. It will also allow for contemplation on why certain features may not have turned out as expected.

## 9.2 System Testing

Postman tests CRUD operations and the results to expect from them such as return code and the data if relevant.

For the POST requests, the responses that were tested are 200, 422 and 500. If the project had completed to a login stage, the response code 401 would have also been tested which indicates unauthorised access.

- 200: The request was received and understood. There are no errors.
- 422: The server was unable to process the request. This may be due to semantically erroneous instructions.
- 500: The server cannot process the request for unknown reasons. This may be thrown if the asset already exists.

The responses were also tested for the data returned if it was a GET request. This ensured that the GET was returning the correct data in the correct format for the request sent and that the response code was also 200. On top of testing if the right asset was returned it was also tested that the correct number of assets were returned. For GET requests the response code of 404 was tested for when searching for an asset that does not exist in the system.

Testing the smart contracts was the same as the POST request testing as all the functions in the smart contracts are POST requests. In addition to these though the error messages were also tested to ensure that the smart contract was exiting the code at the appropriate point if the right conditions were not being met. This involved more rigorous sets up than the standard POST and GET requests as the smart contracts were performing a lot more. This had similarities to unit testing whereby each possible route in the function was tested.

# Chapter 10

# Discussion and Conclusions

This thesis explores two concepts; the use of smart contracts to automate risk analysis and how permissioned blockchains can be applied to share data securely. This work shows that both of these are possible uses of blockchain technology. Smart contracts can be used to automate and add security to many of the processes by removing a centralised authority and adding validation to all the transactions that take place on the system.

The technology also demonstrated that blockchain could be used beyond risk analysis to a certain extent. Demonstrated in the technological contribution is how claims can be automated and insurance contracts delegated to the blockchain network.

The mutual exchange of data securely and without exposing PI is a concept that could be applied to numerous applications and could revolutionise how market research is performed. When multiple companies of the same industry share the network, they could easily expand their market research. Using the permissions, it merely allows the companies to control the level of data exposed. This could be taken further to a point where access control lists could be defined to allow certain parties access to more data than others in exchange for something.

## 10.1 Project Review

The project was ambitious, and while not all features were successfully implemented it does still present an technological contribution that demonstrates the abilities blockchain can have in the area of risk analysis. From the problems that did arise none of them drastically changed the design of the system. Working with relatively new and in some

cases undocumented technologies was challenging, but it has given a strong appreciation and understanding of blockchain and its potential use cases.

With this knowledge now and having reviewed the project there are elements to it that would be changed. The inclusion of machine learning and combined with the smart risk analysis would be the a focus if undertaking this project again. The claims and insurance contracts elements. The reason being is that there is a strong potential for the combinations of these two technologies to have a significant impact. It would have also focused the project on risk analysis alone.

This project also touched on the potentials of private blockchains to share structured data securely amongst its participants. Sharing data on the blockchain in this manner has the potential to be revolutionary with many uses. Using blockchains to share data securely could be explored extensively in an individual project. This could see blockchains change how machine learning and data analytics is performed. It could even incorporate the blockchain network to perform machine learning or data analytics in a secure way that neither party can view the data but get access to the results allowing the sharing of data for analytic purposes without exposing this data.

This skills gained throughout the research and development are invaluable. Web design is a minor part of the software development course and frameworks is something that is never addressed. The project uses ReactJS which is a common front-end framework. Familiarity with node and how it works are another resulting skill. Ultimately a strong understanding of both public and private blockchains has been achieved.

The most apparent skill and by far the most complex one to learn was Hyperledger Fabric. Hyperledger Fabric is an immensely broad framework and very powerful. The innovation of this framework certainly means there is a lot yet to come from it and in time it may become even more powerful, but it is certainly at a point where industries can start adopting it and using it for their needs now. This skill should be valuable going into the future as blockchains popularity grows and Hyperledger Fabric will likely be one of the leading frameworks for private blockchain networks.

To a lesser extent, a familiarity with Ethereum from the initially planned project. Ethereum too is one of the most popular public blockchain networks, and a basic understanding of how to write contracts using solidity and expose the network through web applications using the Web3.js library was gained.

## 10.2 Conclusion

Blockchain will change the way many industries work. If the insurance industry is to adapt and keep up with the change that technology is bringing it needs to adapt blockchain. Blockchain can improve the insurance industry in many ways. It can bring efficiency and security to way claims are handled. It can provide more personalised contracts to policyholders. It can reduce insurance fraud. The adoption the technology is what will ensure traditional insurance companies remain relevant.

The technology though can change many industries and will bring about new types of applications. One thing, however, is that blockchain will not entirely replace the current model of the internet but will work alongside Web 2.0. While yes blockchain can revolutionise many industries, there are also many industries that will not benefit from its adoption.

## 10.3 Future Work

- Completion of the implementation: Complete the project so that all of the requirements laid out in 7.1 are met.
- In future work, the investigation of combining the risk analysis smart contract with machine learning techniques would be interesting to explore. The machine learning algorithm would be able to benefit for the large datasets of the blockchain network, in particular, using the shared industry data system, allowing for it to make accurate predictions about the market.
- Explore the concept of using a permissioned blockchain for sharing data amongst to its full extent and implement an application that solely displays this purpose.

This project has many directions in which it could be taken purely because of what is being explored could be applied to so many applications. It would be exciting to see the implementation of sharing data using a permissioned blockchain network and also for a new type of insurance industry to emerge that is based on blockchain.

# Bibliography


[1] Statista, "Number of monthly active facebook users worldwide as of 3rd quarter 2018." [Online]. Available: https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/

[2] Investopedia, "The worlds top 10 internet companies," 2018. [Online]. Available: https://www.investopedia.com/articles/personal-finance/030415/worlds-top-10-internet-companies.asp

[3] Statista, "Percentage of u.s. population with a social media profile from 2008 to 2018." [Online]. Available: https://www.statista.com/statistics/273476/percentage-of-us-population-with-a-social-network-profile/

[4] C. Cadwalladr and E. Graham-Harrison, "Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach," 2018. [Online]. Available: https://www.https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election.org/

[5] J. Markoff, "Entrepreneurs see a web guided by common sense," 2006. [Online]. Available: https://www.nytimes.com/2006/11/12/business/12web.html

[6] J. P. Barlow, "A declaration of the independence of cyberspace," 1996. [Online]. Available: https://www.eff.org/cyberspace-independence

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[8] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," 2014. [Online]. Available: http://gavwood.com/paper.pdf

[9] Blockgeeks, "What is blockchain technology?" [Online]. Available: https://blockgeeks.com/wp-content/uploads/2016/09/infographics0517-01-1.png

[10] J. Young, "Vitalik buterin: Ethereum will eventually achieve 1 million transactions per second," 2018. [Online]. Available: https://www.ccn.com/vitalik-buterin-ethereum-will-eventually-achieve-1-million-transactions-per-second/

[11] J. Benet, "Ipfs - content addressed, versioned, p2p file system," 2014. [Online]. Available: https://github.com/ipfs/reading-list

[12] BBC, "Turkish authorities block wikipedia without giving reason," 2017. [Online]. Available: https://www.bbc.com/news/world-europe-39754909

[13] ——, "China condemns decision by google to lift censorship," 2010. [Online]. Available: http://news.bbc.co.uk/2/hi/asia-pacific/8582233.stm

[14] Gemalto, "Gemalto data breaches report," 2018. [Online]. Available: https://www.gemalto.com/press/pages/data-breaches-compromised-4-5-billion-records-in-first-half-of-2018.aspx

[15] "A diagram of different network structures." [Online]. Available: https://cdn-images-1.medium.com/max/800/1*NtR5k9fDDr66cLAclhWhuw.jpeg

[16] G. Khiste and Y. Surwade, "Publication productivity of web 3.0 by using science direct during 2008-2017," *International Journal for Science and Advance Research In Technology*, vol. 4, pp. 1632–1635, 03 2018. [Online]. Available: https://www.researchgate.net/publication/324079477_Publication_Productivity_of_Web_30_by_Using_Science_Direct_During_2008-2017

[17] Linkedin, "Linkedin 2018 emerging jobs report," 2018. [Online]. Available: https://economicgraph.linkedin.com/research/linkedin-2018-emerging-jobs-report

[18] P. D. Filippi, "The interplay between decentralization and privacy: The case of blockchain technologies," *Journal of Peer Production, Issue n.7: Alternative Internets*, 2016. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2852689

[19] K. J. ODwyer and D. Malone, "Bitcoin mining and its energy footprint," *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies*, pp. 280–285, 2014. [Online]. Available: http://karlodwyer.com/publications/pdf/bitcoin_KJOD_2014.pdf

[20] V. Griffith and V. Buterin, "Casper the friendly finality gadget," 2017. [Online]. Available: https://arxiv.org/pdf/1710.09437.pdf

[21] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," *2015 IEEE Security and Privacy Workshops*, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7163223

[22] J. Han, H. E, G. Le, and J. Du, "Survey on nosql database," *2011 6th International Conference on Pervasive Computing and Applications*, 2011. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6106531

[23] J. Pavn, S. Hassan, and A. Tenorio-Forns, "Open peer-to-peer systems over blockchain and ipfs: an agent oriented framework," *CryBlock'18 Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pp. 19–24, 2018. [Online]. Available: https://dl.acm.org/citation.cfm?id=3211937

[24] "Comparing the movement of data in ipfs to centralized client-server models." [Online]. Available: https://cdn-images-1.medium.com/max/880/0*1rGbLPMxd_6CwFCJ

[25] J. P. Carrascal, C. Riederer, V. Erramilli, M. Cherubini, and R. de Oliveira, "Your browsing behavior for a big mac: economics of personal information online," pp. 189–200, 2013. [Online]. Available: https://dl.acm.org/citation.cfm?id=2488388.2488406

[26] D. J. Kuss and M. D. Griffiths, "Online social networking and addiction: A review of the psychological literature," *Int J Environ Res Public Health*, 2011. [Online]. Available: https://www.mdpi.com/1660-4601/8/9/3528

[27] Facebook, "Facebook data policy." [Online]. Available: https://www.facebook.com/full_data_use_policy

[28] SocialX, "Socialx whitepaper." [Online]. Available: https://socialx.network/wp-content/uploads/2018/12/Whitepaper-SocialX-v1.2.pdf

[29] Steemit, "Steem: An incentivized, blockchain-based, public content platform." 2018. [Online]. Available: https://steem.com/steem-whitepaper.pdf

[30] D. Hunter, "In-depth guide to gdpr," 2018. [Online]. Available: https://gdpr.report/news/2018/08/20/whitepaper-in-depth-guide-to-gdpr/

[31] M. Bulman, "Facebook, twitter and whatsapp blocked in turkey after arrest of opposition leaders," 2016. [Online]. Available: https://www.independent.co.uk/news/world/asia/facebook-twitter-whatsapp-turkey-erdogan-blocked-opposition-leaders-arrested-a7396831.html

[32] F. Vogelsteller and V. Buterin, "Erc-20 token standard," 2015. [Online]. Available: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md

[33] OpenPGP, "Openpgp." [Online]. Available: https://www.openpgp.org/

[34] "Minimum viable product." [Online]. Available: https://cdn-images-1.medium.com/max/1200/0*--J7k11WKnkcsLTU.png

[35] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *2017 IEEE Technology Engineering Management Conference (TEMSCON)*, June 2017, pp. 137–141.

[36] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *Open Problems in Network Security*, J. Camenisch and D. Kesdoğan, Eds. Cham: Springer International Publishing, 2016, pp. 112–125.

[37] P. Treleaven, R. Gendal Brown, and D. Yang, "Blockchain technology in finance," *Computer*, vol. 50, no. 9, pp. 14–17, 2017.

[38] A. Tapscott and D. Tapscott, "How blockchain is changing finance," *Harvard Business Review*, vol. 1, no. 9, 2017.

[39] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[40] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, Aug 2016, pp. 25–30.

[41] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 1085–1100.

[42] M. Raikwar, S. Mazumdar, S. Ruj, S. Sen Gupta, A. Chattopadhyay, and K. Lam, "A blockchain framework for insurance processes," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Feb 2018, pp. 1–4.

[43] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?" *Future Internet*, vol. 10, no. 2, p. 20, 2018.

[44] M. Henk and R. Bell, "Blockchain: An insurance focus," *Milliman, Nov*, vol. 3, 2016.

[45] G. Cachin, "Architecture of the hyperledger blockchain fabric," 2016. [Online]. Available: https://pdfs.semanticscholar.org/f852/c5f3fe649f8a17ded391df0796677a59927f.pdf

[46] M. Valenta and P. Sandner, "Comparison of ethereum, hyperledger fabric and corda," *[ebook] Frankfurt School, Blockchain Center*, 2017.

[47] "Ethereum consensus description," http://www.ethdocs.org/en/latest/mining.html#introduction, accessed: 2019-04-02.

[48] "Hyperledger Fabric consensus description," https://hyperledger-fabric.readthedocs.io/en/latest/fabric_model.html#consensus, accessed: 2019-04-02.

[49] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2018, p. 30.

[50] F. Lamberti, V. Gatteschi, C. Demartini, C. Pranteda, and V. Santamaria, "Blockchain or not blockchain, that is the question of the insurance and other sectors," *IT Professional*, 2017.

[51] "Hyperledger Fabric description," https://www.hyperledger.org/projects/fabric, accessed: 2019-05-02.

[52] "Hyperledger Composer description," https://www.hyperledger.org/projects/composer, accessed: 2019-05-02.

[53] A. Fedosejev, *React. js Essentials*. Packt Publishing Ltd, 2015.

[54] "Node.js runtime environment," https://nodejs.org/en/about/, accessed: 2019-05-02.

[55] "Hyperledger Fabric solution architecture," https://hyperledger.github.io/composer/v0.19/introduction/solution-architecture, accessed: 2019-05-02.