Trusted Participatory Journalism by Blockchain

Final Report
Tanmay Nautiyal (3035452941)
Sthavir Murthy (3035453464)
Praneet Kumar Pandey (3035437276)

Supervisor: Dr. John Yuen
Date of Submission: 18th April, 2021


Abstract

The ease of access of cameras and communication devices has led to the rise of the new trend of citizen journalism, for the first time giving the average citizen a direct chance at contributing to news and media. Although it is a great step in the empowerment of the common man and the democratisation of information, the trend also poses the risk of enabling the fast spread of something far more volatile: misinformation. Altered images and videos, be it a screenshot of a fake tweet or a clip taken out of context, can destroy someone’s reputation or even shake the stock market. Thus, in order to make the web a more suitable medium for news, this report proposes a solution using the concept of immutable photos and metadata using blockchain to tackle the problem. The goal of the project is to create a new platform that allows user’s the same freedom of freely contributing to the news cycles while verifying that this information really was collected at the time and place it is said to have been collected. The solution makes use of Ethereum blockchain, allowing the users to upload their photos on a decentralised platform while allowing the media access when needed. The project aims to even set a new industry standard, encouraging news sources to only use media whose authenticity has been verified by a digital seal which, if broken by attempted tampering, makes the data invalid due to the decentralised nature of the solution. So far, the project is running ahead of schedule, with the individual components like the camera app, blockchain and hashing system working as expected. Further research will primarily be focused on integrating these individual components and make them work in-sync, although time will also be devoted to studying the solution’s scalability and how it can be improved.
Acknowledgements

“If I have seen further it is by standing on the shoulders of Giants” – Isaac Newton

This project would not have been possible without the invaluable guidance of Dr. John Yuen, our supervisor and Dr. K.P. Chan, our second examiner. Gratitude also goes out to the Department of Computer Science, University of Hong Kong, for providing this opportunity and the required resources. Thanks are also due to my teammates, Praneet Pandey and Sthavir Murthy, for taking on this herculean task with the vision of making the world a better place to live in.

Finally, I would like to thank Ms Mable Choi, for helping us succinctly capture the progress of our work in a manner that it may be useful to others.
# Table of Contents

1. **Introduction** .................................................................................................................. 8  
   1.1 Background .................................................................................................................... 8  
   1.2 Objectives ..................................................................................................................... 10  
   1.3 Scope ............................................................................................................................ 11  
   1.4 Outline of the report ..................................................................................................... 11  

2. **Literature Review** .......................................................................................................... 12  
   2.1 Introduction ................................................................................................................... 12  
   2.2 Technology Review ....................................................................................................... 12  
      2.2.1 Blockchain .............................................................................................................. 12  
      2.2.2 Metadata ............................................................................................................... 13  
      2.2.3 Hashing .................................................................................................................. 14  
   2.3 Related Works .............................................................................................................. 16  
   2.4 Summary ....................................................................................................................... 17  

3. **Methodology** .................................................................................................................. 18  
   3.1 Introduction ................................................................................................................... 18  
   3.2 Ethereum ...................................................................................................................... 18  
   3.3 Django Server, Solidity and Database .......................................................................... 20  
   3.4 Mobile dApp ................................................................................................................. 22  
   3.5 Summary ....................................................................................................................... 24  

4. **Project Development** ................................................................................................... 25  
   4.1 Introduction ................................................................................................................... 25  

# LIST OF TABLES AND FIGURES

**Figure 1.1:** Digitally altered viral photograph of Prince Williams [6] .............................................. 10

**Figure 2.1:** Visual representation of a blockchain network [10] .......................................................... 13

**Figure 2.2:** Hashing of Name ................................................................................................................. 14

**Figure 2.3** Kodakcoin .......................................................................................................................... 16

**Figure 3.1:** Ethereum recommended development stack [19] .............................................................. 18

**Figure 3.2:** Ethereum internal diagram ............................................................................................... 19

**Figure 3.3:** Modified System Architecture ......................................................................................... 20

**Figure 3.4:** Visual Representation of digital verification stamp on article [25] ................................. 21

**Figure 3.5:** Example of a certificate of authenticity ............................................................................. 22

**Figure 3.6:** Ganache CLI Simulation ................................................................................................. 23

**Figure 3.7:** Wireframe Prototypes of dApp ......................................................................................... 23

**Table 4.1:** Project Plan ......................................................................................................................... 26

**Figure 4.2** Veritas App Home Page and Modal View ........................................................................... 26

**Figure 4.3** Add Photo to Blockchain Page ........................................................................................... 28

**Figure 4.4** Signal Receivers ................................................................................................................ 29

**Figure 4.5** Image Comparison Task .................................................................................................. 31

**Figure 4.6** Transactions Syncing on the Network ................................................................................. 32

**Figure 4.7** Django and IPFS ............................................................................................................... 32

**Figure 4.8** Image Model in the backend .............................................................................................. 33

**Figure 4.9** Infura and Celery, used in the backend ................................................................................. 34

**Figure 4.10** ngrok in use during dev .................................................................................................... 35

**Figure 4.11** API Endpoints .................................................................................................................. 36

**Figure 4.12** Image Search .................................................................................................................... 37

**Figure 4.13** Scrumboard snapshot from week 4 ................................................................................... 37

**Figure 4.14** Performance comparison of our blockchain ................................................................. 39
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>dApp</td>
<td>Decentralized Application</td>
</tr>
<tr>
<td>DLT</td>
<td>Distributed Ledger Technology</td>
</tr>
<tr>
<td>Exif</td>
<td>Exchangeable Image File Format</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>UI/UX</td>
<td>User Interface/User Experience</td>
</tr>
<tr>
<td>TPS</td>
<td>Transactions Per Second</td>
</tr>
</tbody>
</table>
1. Introduction

The rise of citizen journalism due to smartphones and the internet has resulted in a disproportionate increase in the spread of fake news by making use of altered photographs, or even unaltered photographs in the wrong context, to push agendas and spread propaganda. [1] Anyone can edit a photo without having an advanced skillset or tools, allowing them to create content which can destroy the image of individuals and even cause turbulence in the stock market. Hence, a decentralised photo database which also serves as a verification tool would serve as a potential solution to deal with this issue.

1.1 Background

The demand for newspapers has fallen to less than half of the sixty million which were printed daily in the US at the turn of the century[2], pushing the journalism industry to adapt to the new medium of the world wide web in order to deliver its services. Yet, unlike the process of ordering a cab or getting groceries delivered to your home with a tap of a button, news and media have an additional obstacle to deal with: they now exist on a platform which can spread unchecked misinformation as fast as fact checked information, if not faster. This makes their role even more crucial, not just because their entire brand value is dependent on the integrity and accuracy of the information they provide, but because they have the additional duty of keeping the government and society accountable for their actions.

To put things in perspective, more than a third of Americans use the internet as their primary source of news [3] and although television still remains the more popular means of consumption, it has become more and more common place for clips of television news itself to be shared on social media for indirect consumption. The commonality that cuts across all mediums is the fact that the
frequency of usage of media provided by the public is on the rise, as can be seen in the coverage of recent movements like Black Lives Matter and the protests in Hong Kong [4]. Although most of these cases are examples of the positive impact of citizen journalism, there are cases as well where various images and videos have been used to spread propaganda, be it to push a political agenda or spread anti-Islamic sentiment.

An example of the same is when during the early stages of the coronavirus pandemic in India, a video of a Muslim fruit seller started circulating online with the claim that he was ‘spitting on his fruits to spread corona’. The video was later picked up by the news channel ‘TV9 Bharatvarsh’, which gave the fruit seller the title of ‘corona criminal’ and led to the video being watched by millions of viewers, only for it to be later found out that the video was actually recorded two months before the coronavirus had even been identified [5]. The fruit seller suffered from extreme harassment and it was later also discovered that he was simply sampling the fresh product he had picked up that day, to ensure quality.

Another example is this viral photograph of Prince Williams showing ‘the middle finger’ to a crowd at his first public appearance after the birth of his third child, which led to an outrage by the British public and the world, only for it to later turn out the photo was slightly digitally altered and that he was actually saying that he has ‘three time the worry now’ to a journalist. [6] Such incidents can destroy careers and tarnish the reputations of innocent people, and some media channels may even take advantage of such ‘outrageous’ incidents to fuel viewership.
Thus, an assessment of the situation leads to the conclusion that there is a need for a platform which can make a photograph and its meta-data (i.e. where it was taken, who took it etc) immutable and transparent, essentially making the information it provides verifiable and reliable. This is where blockchain presents itself as a potential solution to the problem, but also raising the need to clearly define our objectives so that the approach can be chosen accordingly.

### 1.2 Objectives

The core objectives of this project are to research and ensure the feasibility of such a platform, choose the most optimum development approach and then build a working model of the system itself. The system developed should incorporate blockchain technology into photographs and their metadata, to ensure immutability, while also offering users the ability to directly and easily upload, verify and use the contents of the database. The platform should also be able to store the actual photographs itself, but do so without affecting the performance and agility of the blockchain application. These objectives allow for a myriad of ways in which the system can actually be built, raising the need to narrow down on the particular setup which will be used in this project.
1.3 Scope

The aim of the project is to develop a platform with three working parts: a web platform which serves as a database of the actual photos, the blockchain system which stores all the image related information and a mobile camera app which can allow direct upload of photos to the blockchain without giving anyone a chance to edit the photo or metadata before it is uploaded to the system. The database will also offer the option of allowing users to use the photos, while allowing them to hyperlink the photo to a “digital certificate of authenticity” which can be viewed by any other viewers wishing to verify the authenticity of the media.

1.4 Outline of the report

The report begins by introducing the concept of citizen journalism and elaborates on the problems faced by this rising phenomenon. It then introduces a potential solution to those problems, one which uses blockchain and hashing to make the metadata of photos immutable, and identifies the objectives and scope of this project. A comprehensive literary review is carried out of the technologies shortlisted and existing applications of those technologies, followed by a thorough review of the development stack. Finally the report presents the progress made so far on the project along with extensive documentation of the difficulties faced and future plans.
2. Literature Review

2.1 Introduction

As this project revolves around combining different existing technologies in a new way while simultaneously developing the bridging technology requires to make these combinations possible, a literature review was carried out which focused on two aspects: a deep dive into the technology itself and the existing applications of the same in the field of photography.

2.2 Technology Review

This project makes use of three main concepts: blockchain, metadata and hashing to develop the platform. A brief overview of the related findings during the research phase are given below.

2.2.1 Blockchain

Blockchain, also known as Distributed Ledger Technology (DLT), essentially stores data blocks in such a manner that the hash of one block is the key to access the data of the next block in the chain.\[7\] This means that if the data in any block is changed, its hash function changes, making all subsequent blocks in the chain inaccessible. In simpler terms, it is like the contents of one box are the key to open another box: change the contents and you lose access to the next box (which is the key to the box after that and so on). It is almost like a digital seal, which breaks if the data is tampered with. This is the “Ledger” part of the technology, as a blockchain is simply a record of data points, more commonly known as transactions. \[8\]

The one workaround which is still possible if one has time, is that they edit the data in a certain transaction and then use the new hash generated to ‘repackage’ every subsequent block to modify the whole blockchain in order to restore its
integrity. This is where the ‘Distributed’ part comes in, the key to the success of blockchain is to give multiple owners on the platform a ‘copy’ of the blockchain records, allowing only new records to be added at the end of the blockchain and have all the copies updated, but also warn the system if there are any inconsistencies. [9]

![Figure 2.1: Visual representation of a blockchain network](image)

In our project, the blockchain would reliably and transparently store metadata about media uploaded to our platform. By its nature, this data would be permanent and immutable, thus establishing veracity and a point of origin for all media on the platform.

### 2.2.2 Metadata

Exchangeable Image File Format (Exif) is the metadata attached to a photo, which includes information like the shutter speed, ISO, date created and location of the image among other things. [11] The information is created automatically when a photo is taken and can easily be edited. In fact, this data is removed automatically by social media platforms when a photograph is uploaded in order to protect your privacy.[12] The flexibility of the format is what also makes it unsuitable for reliable journalism[13], and this is where blockchain comes in.

Storing the same meta data using blockchain technology ensures that this data, once captured when the photo is taken and uploaded to the platform, cannot be manipulated. This allows users to verify the claims a photograph makes. For
example, if the location mentioned in the metadata is different from where the photograph claims to have been taken, it will prevent citizens from misusing photographs in the wrong context.

2.2.3 Hashing

Hashing is a mathematical concept which involves condensing a large piece of data into a minimal unique representation. It is almost like ‘tokenising’ a piece of data and serves great advantages, especially in encryption and information comparison, as it allows large pieces of data to be compared simply by comparing their hash values. [14] A hash value is generated by passing the data through a hash function: a unique mathematical function which exhibits the following properties:

1) One way function: a hash function needs to be chosen in a manner that even though it is extremely easy to compute the output with a given input, it should be extremely difficult to figure out the input given an output. This ensures that the original data is protected, making the hash a simple unique identifier.[15]

2) Low probability of collisions: hash functions map infinite data points onto a finite range of ‘identifiers’, which allows for the possibility of two
different data points to actually have the same hash value, which can cause misidentification of two different pieces of data as the same. If chosen carefully, hash functions have an extremely low chance of collision, as can be seen in Figure 2.2.

Thus, hash values are to data what a Student ID is to a university student. No two students have the same ID, allowing the SID to be used to uniquely identify and verify a student really is who he says he is, without the system even needing to know his name. As images are also a form of data, hashing can be done on their bit values to generate unique identifiers as well.

Hash functions in encryption actually have a third property as well: that for two similar inputs, the output should be extremely different, making a pattern difficult to spot and keeping the encryption airtight. It is at this point that normal hashing standards defeat our purpose, as this platform requires an algorithm which can not only identify identical images irrespective of their resolution and format (i.e. generate the same hash function for images which are visually similar) but also be able to identify images which are similar yet different, to allow for the identification of tampered images (i.e. generate similar yet different hashes for similar images).

Hashing also offers another advantage which is invaluable on a blockchain: saving space.[16] As blockchains contain millions of transactions in them and need to have multiple copies stored across the network, a large blockchain not only becomes difficult to maintain but also reduces the resiliency of a system. The Bitcoin blockchain is already more than 300GB of data, making it difficult for any and every device to act like a node on its system. If images were stored directly on blockchains, they would grow to be exabytes of data (a single transaction is merely a few kilobytes of data while an HD photograph can be up to 8MB in size), making practical implementation impossible. Hashing allows us
to store the actual photograph on a separate database, and simply store the hash value of images in the blockchain.

Once the concepts were understood, it was vital to study existing applications of these technologies, to figure out what worked and what did not in order to choose the correct technology stack, system design and features.

2.3 Related Works

Although Blockchain has been applied to almost every field in the last decade, including medicine and finance, applications in the field of photography have been limited due to lack of a profitable business model being developed around it.

The most ambitious venture was by Kodak, which recently attempted to help photographers enforce the ownership of their content by launching KodakCoin, a blockchain based cryptocurrency which works in sync with their digital rights management platform to allow professionals to better manage their image licensing. The idea, although progressive, never ended up launching and was even declared the “Worst Idea” at CES 2018 by BBC, not because of its goal but because of its execution.[17] The biggest flaw in the system was that it violated the core goal of blockchain technology, decentralisation, by giving Kodak the control over the platform.

A similar but more successful effort was made by Binded, which used blockchain to authenticate photos by inscribing their date of creation and ownership into a
blockchain. Again, the focus was not on ensuring the authenticity of a photograph, but instead on enforcing ownership. The company was later acquired and the technology simple serves as a backend for Pixsy, a start-up focused on fighting image theft. Their technology is also proprietary and thus, cannot be made available for public use. [18]

These past works serve not just to prove the feasibility of such a platform, but also show the potential blockchain based photos have in terms of restoring our trust in visual media. What makes our project’s solution different is its goal and execution, the goal being to create a platform focused not on professionals but the average citizen, and the difference in execution being the focus on creating a minimum open industry standard to be used by news and media sources when taking advantage of citizen journalism rather than copyright ownership.

2.4 Summary

A thorough analysis of the technologies enabled the team to foresee potential obstacles like the limitations of standard hashing, allowing the various components of the technology stack to be handpicked to suit the needs of the platform. More importantly, the past works review warned us of the need to ensure the ownership of the platform is separate from that of the developer, to avoid the mistakes Kodak made. Thus, the next step was to finalise the system design and technology stack.
3. Methodology

3.1 Introduction

This chapter elaborates on the different layers of the project and the frameworks finalised for each layer. The project makes use of the structure recommended by Ethereum in Figure 3.1 with a few differences: the middle layer also incorporates a server and database to store and manage the actual photos, along with adding custom APIs between all the layers to make them work in sync.

![Ethereum recommended development stack](image)

**Figure 3.1:** Ethereum recommended development stack [19]

3.2 Ethereum

As there are multiple ways in which blockchain has been applied, each with its own advantages and disadvantages, choosing the right network was critical to begin development. There were three main choices for the blockchain framework: Ethereum, Bitcoin, Hyperledger or building a custom blockchain from scratch. Ethereum was shortlisted because features like state database and smart contracts are built-in and are extremely easy to implement in comparison to Bitcoin or custom blockchains, which do not have such libraries. Even
Hyperledger, despite its similar specs, was discarded as it is better suited for private networks. [20]

As can be seen in Figure 3.2, Ethereum Blockchain consists of three layers made of four tries, namely the storage, transaction, state and receipts trie. Each trie serves a particular purpose, with the transaction trie storing the actual data itself while the storage trie contains the smart contracts. The storage trie data is encrypted using the associated account’s key to keep it secure. Storage Tries combines together to form the State Trie, as it represents the overall state of the blockchain. The State Trie is also the place which contains information like wallet balance, which can be manipulated with every transaction.

![Figure 3.2: Ethereum internal diagram](image)

The roots of the blockchain store the hash values of each trie, ensuring that the data cannot be tampered. This structure allows for different parts of the data to be stored separately, improving the system design and creation of blocks for the blockchain itself, making Ethereum the ideal choice.

Ethereum also offers the option for smart contracts, which are automatically executed when certain conditions are met. These are extremely useful for automating the process of issuing digital certificates, as the system can verify on its own whether an image is real or not. This is where the backend comes into the picture.
3.3 Django Server, Solidity and Database

The backend server will be used for running the verification system (generating digital certificates) and for managing the database of images. Django, a Python framework, meets the needs of the project and comes pre-packaged with libraries which make functions like session management and user authentication easy to implement. Additionally, Python and Django both are designed to allow for rapid prototyping, which will speed up the development process.[21] The server and database also solve the storage management problem raised when a photo was stored on the blockchain itself, allowing it to be stored separately and letting the hash value and hyperlink serve as low storage alternatives. The system design recommended by Ethereum was thus modified to incorporate the server, as is shown in the Figure 3.3.

![Diagram](image)

**Figure 3.3: Modified System Architecture**

The hashing system will also be incorporated into the backend, and will make use of the Python ImageHash 4.1 Library to vastly cut down development time and increase system efficiency. A deep dive was done into the various options supported: average hashing, perception hashing, difference hashing and wavelet hashing [22]; finally, average hashing was selected as the optimum choice as it generates the same hash value for a photo irrespective of resolution and format and also generates similar hash values for similar images, unlike encryption hashing.[23]
Solidity, a language made specially for creating smart contracts by Ethereum [24], will also be used in the backend for approving the creation of digital certificates of photographs, and make them accessible anywhere on the web using a RESTful API.

An example of the workflow is given below.

![Visual Representation of digital verification stamp on article](25)

1. A journalist can feature an image stored on the platform using a unique link generated for each photograph in their article. The photograph will have a unique border (tentatively green in colour) along with the logo of the platform to signify to a user easily that the image is real, as can be seen in Figure 3.4.
2. If a user wishes to further ensure the image is real, they can click on the image to be redirected to a ‘Digital Certificate of Authenticity’ issued by our platform, like the one in Figure 3.5, which makes visible all the metadata of the image. This Digital Certificate was automatically created by the Solidity system when the image met all the criteria required to be certified authentic.

![Certificate of Authenticity]

**Figure 3.5:** Example of a certificate of authenticity

3.4 Mobile dApp

The main User Interface (UI) of the platform needs to be easily accessible from any device that can connect to the internet, irrespective of the operating system or form factor. At the same time, it also needs to be able to access a decentralised database, a problem which can be overcome by using the Ethereum Decentralised Application (dApp) using React native. [26] React-Native offers the flexibility of being platform independent, and thus allows for the development of a single application that can be deployed on Android and iOS. Ganache, a blockchain network simulator and development environment, can easily be integrated with dApps, allowing for rigorous testing before deploying the application and rolling out onto the cloud infrastructure, as can be seen in Figure 3.6.
The workflow of the application will be as follows:

1. The user registers himself onto the system through the dApp.
2. The user uses the camera function in the dApp to capture photographs.
3. When a photograph is captured, a transaction is registered against the user on the system containing all the information of the image. This is added to the blockchain.
4. The image is uploaded onto the database and a link is generated for it to be used in any article.

**Figure 3.6: Ganache CLI Simulation**

**Figure 3.7: Wireframe Prototypes of dApp**
After the tech stack was decided, sample mock-ups, as shown in Figure 3.7, were created to ensure the UI/UX of the app is easy to understand by the average citizen, as they are the target market for the product being developed.

### 3.5 Summary

The methodology comprehensively covers the technology stack used, while also covering the reasoning behind choosing each framework, which were chosen after a thorough analysis of the pros and cons of all the alternatives. The finalisation of the system design and technology stack enabled the team to learn the frameworks needed, distribute the work and begin development.
4. Project Development

4.1 Introduction

A development timeline was created with an estimated completion duration of one year, with initial focus on learning the unfamiliar frameworks, incorporating inputs on the plan from the project supervisor and rapid prototyping.

4.2 Tentative Schedule

In line with the Agile Methodology, Sprints with average duration of two weeks will be carried out with reviews at the end. Table 4.1 highlights the major targets and deadlines.

| Sep 2020 – Oct 2020 | Stage 1 | • Conduct Literary Review  
|                     |         | • Finalise Blockchain design  
|                     |         | • Decide Database schemas  
|                     |         | • Finalise Tech Stack  
|                     |         | • Reach consensus on user functionality  
|                     |         | • Set up internal standards and metrics  
| Nov 2020 – Dec 2021 | Stage 2 | Parallel initial development of  
|                     |         | • The blockchain  
|                     |         | • Backend server  
|                     |         | • Frontend Mobile dApp  
|                     |         | Component Testing  
| Jan 2021 – Feb 2021 | Stage 3 | • Develop Minimum Viable Product  
|                     |         | • Integration Testing  
| Mar 2021 – Apr 2021 | Stage 4 | • Further Integration testing  
|                     |         | • Scalability measurement and performance improvement  
|                     |         | • Adding additional functionality  


### 4.3 Mobile App Progress

The comprehensive mobile app acts as a one stop shop for all of the citizen journalist’s needs. Built with react native, it supports both Android and iOS.

![Figure 4.2 Veritas App Home Page and Modal View](image)

After logging in, the first page is a gallery of the images already on the blockchain, so that users can view the citizen journalism contributions of other
users. These are fetched from the IPFS server using an API call and are all verified images.

The page has multiple other features. If a user wants to know more about a particular image, they can tap on it to load a modal view of the metadata and all other details of the image, along with the ability to view a larger version of the same. This feature also allows users to access the “similar images” tab, which makes another API call to the backend and fetches similar images to the one currently being viewed. The current threshold is set at 50% similarity, the technicalities of its functionalities are explained in the next section.

Furthermore, the app has a direct upload page, which can be accessed by pressing the “Add” button on the Home Page. This page allows the user to access the camera, click a photo, give it a name, attach an article to it, include tags and then upload it to the blockchain. Thus, this page makes use of multiple internal libraries for camera access, managing permissions and making axios based POST API calls to add the image to the blockchain. Once the image is added, it gives a confirmation of the same, along with the IPFS address of the image, as can be seen below. It should be noted that the article attached to the image is also separately put on the blockchain, allowing the system to technically verify images and text.

Thus, the app is extremely minimalist and intuitive, to allow people unfamiliar with blockchain to use it like a normal camera app. Multiple animations and transitions are included to improve the user experience, along with a handpicked colour scheme used a unique shade of blue.
Choosing how and where to add image comparisons within the application was a difficult task, primarily because it is a process which can take an exponential amount. There were various factors to take into consideration and various design issues which arose from the implementation of such a system. Eventually, using basic Django signals and the celery package, the implementation was performed such that every time a user uploads a picture and after it gets stored on to the IPFS
and server, it is compared to every other picture on the server to get a uniqueness value.

Django signals are unique ways to communicate between models and functions. For example, in this application, the “post_save” signal was used as way to signal the completion of the creation of a model. The code is displayed within Figure 4.7:

```python
from django.dispatch import receiver
from django.db.models.signals import post_save
from .tasks import compare_images
from .models import Image, Similar

@receiver(post_save, sender=Image)
def comparison_handler(sender, instance, **kwargs):
    print(instance._id)
    compare_images.delay(instance.imgipfsHash)

@receiver(post_save, sender=Similar)
def similar_save(sender, instance, **kwargs):
    print("SIMILAR MODEL SAVED")
```

*Figure 4.4 Signal Receivers*

Once this signal is read, the IPFS hash on the image is then passed into a celery task. Celery is library designed for performing asynchronous functions in Django. The Django workflow only allows for synchronous operation, i.e. each request is only performed one after the other. This would be an issue if we were to upload two images at the same time. Django would compare them to all the other pictures on the server but not each other. Using the `@shared_task` decorator from the celery library, we can make sure that this does not happen. The celery worker communicates with the Django server using an active Redis server, a message broker which transfers requests from Django to the Celery worker running on another terminal.
The workflow on the task method is provided in Figure 4.8. Understanding that image comparison is potentially a very processor heavy and time consuming tasks, we have used python multiprocessing to split the processes into a number of threads (4 by default), which ensures that the comparison process is performed many times faster than usual. The image comparison process itself was chosen amongst the best and most efficient methods available to the team. The following methods were tried:

1. Bit-wise pixel by pixel comparison using Pillow: This method is the most brute force of all that were attempted. It basically entails converting images to byte arrays using the Pillow library and then going through them element by element and measuring how many were shared between the two images. Understandably, this took the most time to calculate and the accuracy also suffered as a result. Ways to make it faster resulted in reducing image resolutions by a non-marginal amount, decreasing quality even further.

2. Scikit-image and OpenCV: These libraries make use of machine learning models and convoluted neural networks in order to calculate the Structural Similarity Index Metric (SSIM) between two images. Using this method resulted in the best accuracy out of all the methods attempted. However, it was also the slowest, taking upwards of ~50 seconds per comparison, which even with multithreading and testing on a database of ~40 images would take upwards of 20 minutes of processing. For this reason, this method was also rejected.

3. PixelMatch-py: This library uses average colour comparison and average gradient measuring to calculate the difference between two images and is marketed as the fastest image comparison library available. Upon application to our system, it resulted in faster processing speeds than any other method that was attempted and even though the accuracy may suffer as a result of the averaging tendencies of this method, it would be a reasonable trade-off in favour of faster processing speeds.
Once the image comparison is done, all the similar images and their percentages are stored in individual similarity models which can then be queried using the appropriate endpoint (see table 4.6). If no similarities are returned the image is marked as verified and will show up on the journalist client website.

Figure 4.5 Image Comparison Task

This same library is also used from the frontend app through an API call, to fetch similar images to the one being viewed by the user. As the size of the blockchain increases with more images being uploaded, the system becomes more and more efficient in finding similar images as well, making the solution benefit from scaling.

4.5 Blockchain and IPFS

Using Ethereum development environment and Solidity, a modified blockchain was setup which interpreted each photograph’s metadata as a unique transaction, bundling them up into blocks and adding them to the blockchain. Ganache
enabled the testing of this blockchain in a simulated environment, and successful syncing was achieved as can be seen in the figure below.

Each user is initially given the same amount of tokens at sign up, which are used as gas to upload images onto the blockchain. IPFS is closely integrated with the system, with the workflow being that the image is first uploaded to the IPFS system to reduce memory load on the blockchain. The generated IPFS hash, along with all the metadata, is then uploaded to the blockchain.

![Figure 4.6 Transactions Syncing on the Network](image)

### 4.6 Backend Server and Database

The backend server was setup in Django, a python based web development library. Views were created for all the different functionalities, including fetching a single image on the blockchain using IPFS hash, fetching all the images on the blockchain, searching images by tags, measuring image similarities etc.

![Figure 4.7 Django and IPFS](image)
Endpoints were created for all of these features as well, to allow them to be accessed from the website and the app where needed. A mongoDB server was also set up, to store all the other information related to the operations of the system itself.

As the bridge between all the moving parts of the application, the server performs a variety of functions, of which the implementations differ by a significant amount.

Django as a factor of its own design, works based off models (Figure 4.5) and every time any data is received, the framework stores an instance of a models based on its field within its local or remote storage. Our preferred methods of a database server was using MongoDB, which is a noSQL database system, using a system of documents to store elements in a freeform fashion. This immediately presented a design issue for the application – whether we must store an object model on IPFS or Mongo.

![Figure 4.8 Image Model in the backend](image.png)
Since Django stores models in its own formats and uses its own drivers to store information on the server, it was an implementational difficulty to translate that using a custom driver and store all information on the database. Secondly, not only did we have the Image model, but also a user models for authentication and a similarity model to store how similar two images are to each other. It would not make any sense to store those on the distributed server because of two reasons mainly:

1. Django queries to the server are faster than retrieving data from the IPFS system because of how the systems are each construed.
2. User information is confidential, storing it on public servers would be a drastic breach in privacy, even encrypted. As the application owners, we have a duty to protect user information and based on our readings, Django’s SHA-256 security on its users and their passwords is incredibly more efficient to both execute and implement than a custom encryption strategy based on a public server.

Infura’s development suite was used for the deployment of the blockchain and IPFS related APIs as it provides the scalability needed for the same. The backend also has the authentication system built in, which is used for login and sign up.

![Infura and Celery, used in the backend](image)

To make image comparison efficient, Celery, an open source asynchronous task queue system, was used. Celery allows scheduling and use of distributed resources in real time, ensuring that the image comparison system, which takes
time due to the large scale of images on the system, does not slow down system performance.

Various other technologies were incorporated to improve system performance and speed up development. For example, ngrok was used to keep the site online and accessible from external endpoints, increasing functionality and simplifying the development work.

### 4.7 RESTful APIs

Since both the news agency image retrieval website and the react native app uses an API to connect to the server and glean meaningful information from the various aspects of data that we use, it was crucial to build a robust and accurate REST API for the system to use. Thankfully, Django has a library provided for this very purpose. Django-rest-framework is the most popular among the community to develop REST APIs and is quite easy to implement in its structure. Through the Django MCV (Model, controller, view) architecture of web design, different pieces of the API are suspended in different parts of the application. For example, in an API call to retrieve a particular image using its IPFS hash, the flow of information is as follows:

1. The user makes an API call using HTTP GET with the IPFS hash of the image in the URL of the request.
2. The URL is parsed through Django’s URL parser within urls.py and passed to the corresponding view in views.py.
3. The view then queries the server using the models described afore, and retrieves the relevant data.
4. A HTTP 200 response is returned along with the data object.

A description of the various endpoints made available is demonstrated in Table below:

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /images/</td>
<td>Retrieves all images on server</td>
</tr>
<tr>
<td>GET /images/&lt;ipfsHash&gt;</td>
<td>Retrieves specific image using its IPFS hash</td>
</tr>
<tr>
<td>POST /images/</td>
<td>Uploads an image, its label, tags and description</td>
</tr>
<tr>
<td>GET /images/similarity/&lt;ipfsHash&gt;</td>
<td>Retrieve all images similar to this one on the server</td>
</tr>
<tr>
<td>POST /auth/register</td>
<td>Creates new user on database</td>
</tr>
<tr>
<td>GET /auth/logout</td>
<td>Logs current user out</td>
</tr>
<tr>
<td>POST /auth/login</td>
<td>Logs current user in</td>
</tr>
<tr>
<td>POST /search/</td>
<td>Returns images matching tags posted in request</td>
</tr>
</tbody>
</table>

*Figure 4.11 API Endpoints*

### 4.8 Image Search Website

A website was developed for journalists to search through the image base easily and incorporate them into their articles. The website is also minimalist and intuitive, allowing the journalists to search for images related to term, which are matched with the tags of the image to return results.

If a journalist wishes to use an image, they can simply use the share feature to directly embed the image into their articles, where they are represented with a green border and also are hyperlinked to the original IPFS image. Thus, any viewer of the article can verify the authenticity at two levels if they chose to do
so. Like the app, the website was designed with sound web design principles in mind and also uses Material-UI.

![Image Search](image1.png)

**Figure 4.12 Image Search**

### 4.9 Agile Methodology

Scrum and agile development methodology were used throughout the development of this project. Sprints normally lasted two weeks, with code reviews and performance reviews at the end of each sprint. The Atlassian suite was used during the development, including the Jira scrum board for task management and Confluence for creating a wiki of all project data.

![Scrumboard snapshot](image2.png)

**Figure 4.13 Scrumboard snapshot from week 4**
4.10 Authentication

It was necessary to create a user authentication system to be in line with the project’s stated goal of recording and maintaining authorship. Django is well-equipped for this task and has a built-in standard user model that can be used. It was then a simple case of connecting the image model to the user model, and creating serializers for the user model to use in API communication between the client and server.

Views for handling logout and registration were written in an ‘authentication’ app in Django, and login was handled by default.

The resulting authentication worked on DRF’s token system, where authenticated users were appointed a token confirming their user status. Then permissions were altered so that only authenticated users could upload images to the platform.

On the client side, forms were written in React to collect and send the appropriate data to the backend to enable authentication.

4.11 Performance Metrics

The camera app and backend followed standard protocol, and not much could be further improved or measured in terms of performance. On the other hand, setting up an efficient blockchain required getting familiar with how each factor affects its performance, be it the block size, transactions per second or time needed to reach consensus.

Initially, attempts were made to store the photographs directly on the blockchain, but that reduced the number of transactions stored in each block due to the following inverse relationship:

\[
\frac{\text{Block Size}}{\sum_{i=1}^{n} \text{Transaction}_i} \cdot \text{(Total Transactions)}
\]

Storing photographs increased the size of each block to more than 2 MB (even after compression), making the blockchain unscaleable. This is because the
performance of a blockchain depends on the number of transactions it can execute per second, which greatly reduces because of the large number of blocks needed to store the transactions.

$$\text{Transactions per sec} = \frac{\text{( # of Transactions per Block)}}{\text{( # of Blocks) (Block Generation Time + Block Relay Time + Block Consensus Time)}}$$

Thus, the novel solution of storing only the hash of an image on the blockchain was introduced to reduce the average size of a transaction, allowing each block to store a larger number of transactions and thus, increasing the max throughput.

Further optimisations were done to bring down block generation time (the time required for the hash function to generate a block, depends on the complexity of the function), block relay time (time required for a newly generated block to be added to each node storing the blockchain) and block consensus time (time required for all nodes to agree on the present state of the chain). The throughput was then calculated and compared with other blockchain protocol, as can be seen in Figure

**Figure 4.14 Performance comparison of our blockchain**

As hashing the image consumes time, the custom blockchain was able to reach quite close to Ethereum’s default max of 15 TPS and far exceeded Bitcoin’s
throughput of 4.6 TPS although, it should be noted that Bitcoin transactions tend to be slower due to higher block consensus time due to added security.

### 4.12 Progress Website

A website has also been created for the project so that followers can have an idea of what the project entails, current progress and development, as well as the timeline for project progress. The link for the website is at: [https://wp.cs.hku.hk/fyp20049/trusted-participatory-journalism-by-blockchain/](https://wp.cs.hku.hk/fyp20049/trusted-participatory-journalism-by-blockchain/)

### 4.13 Limitations and Difficulties

With the unique remote work situation in place due to the coronavirus, a major hurdle in making this platform a reality has been efficient communication. One of the team members is in a different time zone altogether, and even though biweekly meetings have been organised regularly, obstacles have become a little harder to navigate.

One of the biggest initial obstacles was the inability of the blockchain to hold the actual photos due to size constraints. This was also one of the reasons this project had not been attempted before, but when the literary review revealed the technique of custom hashing, it not only allowed us to use a unique token for each image on the blockchain itself, but also made it possible to quantitatively measure the difference between two similar photos.

Another obstacle was the lack of the basic network of nodes required to test the simulations for the blockchain, which was solved by using Ganache, as mentioned previously. This helped solve the paradox that the platform cannot be developed if the device network isn’t in place and the device network cannot exist unless the platform is in place. Ganache allows us to simulate all possible cases that the system can face, thus improving resiliency and testing.
The lack of experience in working in Solidity was also a barrier initially, but time was dedicated to online tutorials and courses to ensure a sound understanding of the fundamentals and application of the core concepts.

A big hurdle was the question of image comparison and how it must be done. Initially we planned to compare image hashes, but that would prove too inaccurate and inefficient and hence many other decisions were made with regards to the issue, as have been highly before. Upon reading and experimentation, however, we were able to get a robust system working.

Despite all the obstacles, the was completed on time and met its deadline. The platform goes beyond its original vision of being a proof of concept and has created a beta level product which can be released to the public already.

4.14 Future Plans

Citizen journalism goes beyond photographs, and the future plans include implementing a similar system for videos and audio recordings. As these media formats are also essentially streams of data and also have metadata, it is feasible to use blockchain on them as well although hashing is relatively difficult, which is why these were not incorporated into the current plans.

Another important goal in the long run which can revolutionise citizen journalism is the incorporation of artificial intelligence to allow the system to verify even photos not taken by our app and also automatically verify authenticity of images already featured on news websites by comparing them to the photos on the database. This will vastly increase the flexibility of the system, as most media is recorded through either native camera apps or social media apps.

As was said at the start of the report, the final goal of this project is to create not just a proof of concept but, an open standard that can be used by all media channels, giving news its rightly deserved place on the internet while
empowering citizens by giving them a chance to contribute to the knowledge capital of the human civilisation.

4.14 Personal Contribution

Sthavir Murthy primarily handled the authentication system and infura related development, while Praneet Pandey was responsible for the backend and the blockchain. My contribution to the project was primarily in the development of the react native application and the website, as these were the primary sources of input and output for the system. I also assisted in the development of the endpoints in the backend, as these were the directly called from the frontend.

Along with this, I also played a primary role in conducting literary review, implementing agile methodology, managing the sprints and coordinating with the supervisors. More importantly, I assisted in finalising the system architecture and choosing the development roadmap. Finally, I was also responsible for deploying the initial rudimentary blockchain and IPFS system and integrating them, although this code was rewritten with superior performance and features later by the development team. Bug fixing was also part of my role.

4.15 Summary

The chapter encapsulates the difference in theory and the real world by capturing the difficulties faced when implementing the project plan, while also capturing how those obstacles were overcome to make significant progress on the individual components of the system. This concludes all the work that has been carried out so far on the project.
5. Conclusion

The report proposes and discusses a viable potential solution to combat fake news by verifying image authenticity using blockchain. The solution has three major components: the blockchain, the backend server and the frontend dApp along with APIs which allow each component to work with the other while also allowing the platform to work directly on other websites as well. The decentralisation, immutability and transparency of the blockchain ensures that the data is trustable, while also allowing a user complete control over the photos they have taken. Smart contracts automatically verify new images and the dApp offers a straightforward end-to-end integrated solution which can be used by the average citizen who has no knowledge of blockchain or the other technologies used.

The entire process of literature review, system design, choosing the technology stack and progress is well documented, with rationale provided for each and every decision along with an analysis of the pros and cons of the alternatives. Continuous feedback is being incorporated from the supervisor and journalists have also been reached out to for their inputs. The development is still in progress, and is expected to complete in time, with a minimal viable product available in January.

In conclusion, Blockchain as a relatively new technology has myriads of applications, and the project’s research suggests it will create immense positive impact in the field of citizen journalism.
References


