COMP4801_COMP4802

BeatChain: Blockchain Based Music Licensing

Final Project Report

Supervised by

Dr Allen Au

Mahima Anand (3035550987)

Bevan Varghese (3035552777)

Siddhant Bagri (3035551785)

Submitted April 18, 2022

Abstract for progress report of Final Year Project entitled

“BeatChain: Blockchain Based Music Licensing”

Submitted by ANAND Mahima

The rise of streaming services has enabled the large scale dissemination of music over the internet in an efficient manner that has cut industry costs and significantly raised revenue. However, while the accessibility of digital music has increased its consumption, new problems surrounding the protection of copyrights and distribution of artist incomes have surfaced. Tracking and assessing instances of copyright infringement becomes difficult when the relevant information is stored asynchronously across different databases. Moreover, a sizeable portion of the income is collected by intermediaries leaving the artist with little. Upon conducting background research for similar projects in this domain, gaps were found in the technical designs of these applications which hinder them from effectively addressing both of the preceding issues concurrently.

To tackle these issues, our team has developed BeatChain – a solution that leverages blockchain technology and smart contracts in order to create a direct channel between the artist and listener and synchronise copyright data storage over a decentralised network. The deliverable has been implemented as a web streaming platform characterised by three core features: (i) a decentralised file storage system, (ii) a decentralised rights ledger, and (iii) integrated cryptocurrency wallets. The system enables the artist to upload their music and independently manage copyright licensing on a public ledger through smart contracts. Payment is automatically and instantly made to the artist and when a song is streamed. Without the intermediary, the artist collects a greater share of the royalties. Additionally, infringement is dealt with by encrypting the audio file stored on the blockchain leading to a reduced amount of lost royalties.

The project has completed all fundamental aspects of development and is available is a fully functional web-application which allows the user to link their cryptocurrency wallets through MetaMask, stream music, upload their own music to the Polygon blockchain through Solidity-powered smart contracts, encrypts their uploaded audio files to protect against infringement and provides additional features such a track filtering and playlist creation.

Acknowledgements

As I step into my final year of undergraduate study, I am grateful to The University of Hong Kong for the role it has played in my academic growth by equipping me with the necessary skills and confidence to undertake the Final Year Project.

I would like to begin by expressing my profound gratitude towards our team's supervisor, Dr Allen Au. Starting from the ideation stage of the project, Dr Au has been very supportive. He is always happy to help and has consistently taken out the time answer our queries on a near weekly basis. The depth of knowledge he possesses on our project topic is immense and I really value the guidance provided by him through his meaningful feedback. I have learned a lot from him over the last few months and am deeply inspired by his sincerity and vision. Next, I would like to thank Dr John Yuen for his valuable feedback and pointers which greatly helped shape this project.

I would also like to thank my teammates Bevan and Siddhant for their positive attitudes and strong work ethic which has been monumental in fostering a robust and encouraging team dynamic. As a result of this, we have always been able to approach our work in an enjoyable and organised manner.

Finally I would like to thank my CAES instructor, Dr Locky Law. I greatly admire his enthusiastic approach to teaching that has provided me with the adequate tools to write this report in a structured and articulate manner.

Table of Contents

Table of Figures

List of Tables

List of Abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
B2C	Business-to-Consumer
CBC	Cipher Block Chaining
COALA IP	Coalition Of Automated Legal Applications — Intellectual Property
DRM	Digital Rights Management
EVM	Ethereum Virtual Machine
GUI	Graphical User Interface
HKD	Hong Kong Dollar
HTTP	Hypertext Transfer Protocol
IPFS	InterPlanetary File Storage
ML	Machine Learning
PPS	Pay Per Stream
RPC	Remote Procedure Call
UI	User Interface

1. Introduction

1.1 Background

Technological advancements have greatly influenced the production and dissemination of digital media. This is particularly evident in the case of digital music whose large scale propagation has been made possible with the rise of streaming services. In 2020, a report by The World Bank observed that over US$350 billion was collected in intellectual property charges with music royalties taking up a majority of the share [1]. The rising trend of streaming services frontrunning the revenue earned by the music industry is depicted in Figure 1 below [2]. Streaming services have attained this success due to their growing convenience and accessibility among smart devices.

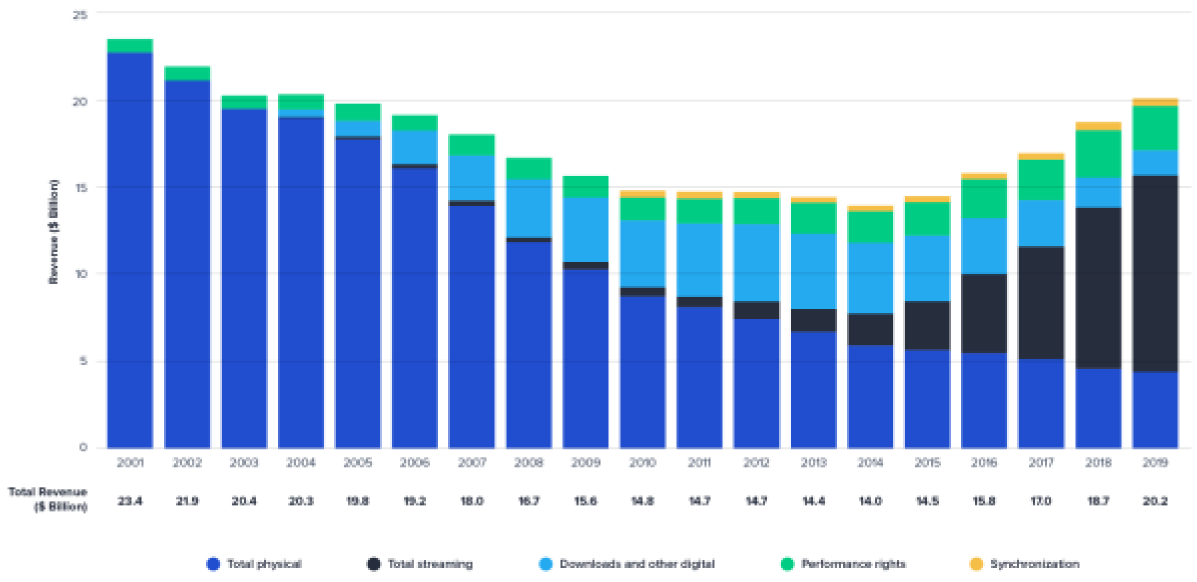


Figure 1. Music industry recording revenues from various sources from 2001 to 2019 [2]

These services are connected to the artist though middlemen involved in the publishing and recording process. The publishers work to promote, distribute and monetize the artist's musical compositions while record labels are involved in the recording process of the final product. Moreover, there exist numerous streaming platforms today such as iTunes, Spotify, Google Play, and YouTube music each possessing their own mechanisms and models for storing audio files and generating revenue. This kind of structuring has led to two key problems surrounding music streaming – preventing copyright infringement and fairly distributing royalties to the original artist. This will be elaborated upon in the next section.

1.2 Problem Statement

Music copyright equips the artist with the exclusive right publish, distribute and perform their work [3]. However, the expansion due to online streaming has brought forth problems regarding copyright license management and the transparent calculation of royalty payments leaving the artist vulnerable

The global revenues earned from music streaming have more than quadrupled since 2015 [4]. Within the music industry, streaming services now generate over 62% of the total global revenue [4]. However, a majority of this revenue is pocketed in stages by various intermediary parties involved in the licensing and dissemination of music such as publishers, and record labels. This leaves only a small portion of income for the original performers and composers. Moreover, intense competition in the industry deprives smaller artists of the time and liberty to grow their music careers since they rely on other jobs as their primary source of income. Figure 2 below reveals how major dependence on non-music income is universal across all age groups of smaller artists[5].

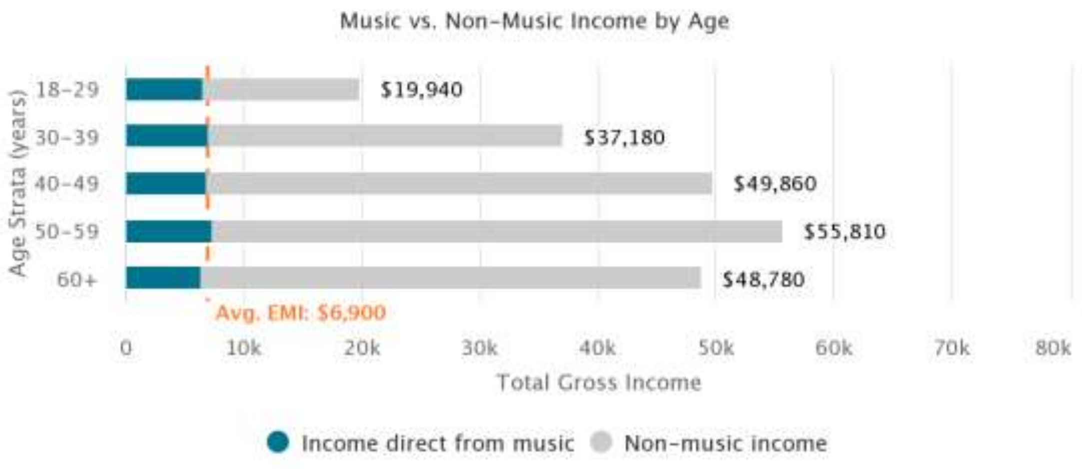


Figure 2. A small musician's aggregated music vs non-music income across various age groups [5].

Another obstacle arises in transparently calculating the artist's income due to different subscription models among streaming services such as Google Music, iTunes, YouTube etc. For instance, iTunes users are only allowed to listen to music upon purchase of a single or an album while Spotify charges monthly subscription to stream all the music available on their catalogue. Further, it is difficult to assess and track instances of copyright infringement via illegal propagation since the databases maintaining copyright owner information differ between countries as well as companies [6].

1.3 Project Objective, Scope, and Contribution

On the basis of the discussion that has taken place thus far, our Final Year Project has sought to develop a blockchain driven solution to the issues surrounding music licensing. Taking the shape of a web-streaming platform, the proposed solution directly links the artist to the consumer thereby eliminating the middleman concerned with distribution. Moreover, with a consistent decentralised database storing copyright owner information, royalties can be correctly and fairly calculated.

The platform offers 3 key features:

1. <u>Decentralised file storage</u> enabling musicians to upload and store their work on the network and permitting its accessibility through a streaming fee.
2. <u>Decentralised rights ledger</u> which maintains and validates the copyright license of the uploader in manner that a song can only be streamed upon a confirmed transaction.
3. <u>Cryptocurrency wallet integration</u> which allows users registered on the platform to link their wallets so as to enable direct payment to an artist every time a song is streamed.

Such a system seeks to eliminate distribution costs associated with intermediary parties allowing the artist to secure a greater percentage of the revenue in a timely manner. Smaller artists for whom the current distribution methods are unaffordable would be particularly benefited. Further, concerns over royalties lost as a consequence of copyright infringement are adequately addressed given the synchronised database and trackable smart contracts.

The suitability of blockchain in developing such a platform lies in its three key characteristics namely:

1. <u>Decentralisation:</u> Blockchain is a distributed network of peers with no centralised governing authority monitoring the network.
2. <u>Immutability:</u> Data stored on the blockchain cannot be tampered with.
3. <u>Transparency:</u> All members on the blockchain network can view the information maintained on it.

These technologies and their characteristics will be discussed in more detail in section 2.

1.4 Report Structure

This report comprises of 8 sections. Section 2 provides a background on the key technologies that will be used to build the application namely blockchain, smart contracts, and layer 2 scaling solutions. This section also reviews and evaluates some of the existing projects employing blockchain to tackle the same problem. Section 3 delves into the methodology of our project application by explaining the scope of the deliverable and explaining the various components of its design architecture. Section 4 highlights the current status of the project in terms of the features that have been developed along with their detailed implementations. The challenges we faced during the development of the application and some of the current limitations are discussed in section 5. An assessment of our progress is done in section 6 along with the next steps to be undertaken in the future development process. Section 7 analyses and describes the business model we wish to pursue in a way that benefits both the artist and the platform. Finally, section 8 provides a conclusion to the paper.

2. Literature Review

This section delves into the key technologies and their suitable characteristics that form the backbone of our project, namely – blockchain, smart contracts, and the applicability of layer 2 scaling solutions. Following this, it reviews some of the existing projects navigating solutions to the aforementioned problem. Finally, it describes how our platform leverages these technologies to develop a suitable solution.

2.1 Technical Background

2.1.1 Blockchain

A blockchain can be defined as a digital public ledger for recording transactions [7]. It essentially serves as a database which stores information as a growing chain of ‘blocks’. The suitability of a blockchain arises from its properties of being decentralised, transparent and immutable.

Through its decentralised network architecture, each member has access to an identical copy of the data stored on the blockchain ledger which updates in real time. New data is verified and added via a consensus algorithm and any compromise to a member’s ledger would be rejected by majority members on the network [7]. This ensures data transparency and keeps the blockchain reliable. Moreover, without regulation of a centralised authority, peer to peer transactions can take place instantly eliminating the need to pay an intermediary fee [7].

Once created, it is impossible to alter the chain rendering it permanent and immutable. This is achieved by cryptographic hashing. Each new block comprises of – (i) the recorded data, (ii) the hash value of the previous block linking it to the rest of the chain, and (ii) a new hash value generated from the contents of the current block and the previous block’s hash which allows for linkage to the next block. Thus, any modification to a block would also alter its hash value and conflict with the existing block and would subsequently require every other block to be modified.

2.1.2 Smart Contracts

Smart contracts are programs stored on the blockchain that execute when predefined conditions, or “trigger-events”, are met [8]. Hence, the execution of an agreement or another event can be automated without the effort or time of any intermediaries. Much like regular contracts, they define rules in the form of code which are automatically enforceable by specified function calls. Smart contracts are most commonly used for executing virtual currency transactions [8]. They possess the benefit of being trackable since any interaction with a smart contract is irreversible.

2.1.3 Layer 2 vs Layer 1 Networks

The increasing popularity of cryptocurrencies in everyday life has necessitated the creation of blockchain layers for improved transaction rates, network security and recordkeeping. When the demand on the network is high, it gets clogged leading pending transactions to pool up thereby taking more time to process and execute them. In order to handle this, miners on the network begin to prioritise transactions with higher gas prices. Consequently, the minimum cost of executing a transaction increases to the point where the gas fees skyrockets unreasonably.

This is where layering alleviates the issue. Where layer 1 is the native underlying blockchain, layer 2 is a third-party integration that operates on this layer so as to improve its efficiency [9]. The two layer structure allows for an increased number of nodes which results in a higher transaction throughput [9]. The layer 2 blockchain offloads a portion of layer 1's transactional burden onto another system architecture which manages the processing load. The processed load is then reported to Layer 1 for finalisation. This reduces congestion and makes the blockchain network more scalable. The layer 2 solutions utilise smart contracts to make these transactions [9].

2.2 Existing Projects

With a keen interested generated in solving problems surrounding music licensing and royalty collection through blockchain, the following start-ups and research projects have emerged.

2.2.1 Ujo

Built on an Ethereum blockchain platform, Ujo serves to clarify confusion surrounding music copyright ownership and automate payments from listener to artist [10]. COALA IP specification is used to enable easy license access and IPFS is the decentralised storage platform of choice. Nonetheless, Ujo suffers from insufficient copyright protection mechanisms post track purchase which can result in potential illegal distribution of music.

2.2.2 Singular DTV

SingularDTV also utilises the Ethereum network for distributing content by virtue of smart contracts to manage intellectual property rights and democratise the process of collecting royalties [11]. The platform allows artists to create their own tokens as a form of creating incentives for their fans to stream or purchase their work. These tokens increase in value alongside the popularity of the artist which ultimately benefits the fans when they support new talent. As in the case of Ujo, SingularDTV does not prioritise the protection of copyright with regards to illegal distribution. Additionally, the economics of token management complexifies the utility of this platform to creators and artists who may not have an immediate grasp on these concepts.

2.2.3 Vezt

Vezt makes use of an alternate business model creating a marketplace of sorts for music rights [12]. Fans fund artists by directly buying into a partial ownerships of the artist's royalty rights. Songs take the form of mini-corporations with investment in them resembling the environment of purchasing shares in a stock market. This creates incentive among the fans and also enables artists to secure sizeable funding for their music. However, Vezt comes with a steep learning curve where it is impossible for artists to successfully operate on such a platform without knowledge of financial markets. Moreover, it also leads for loss of autonomy for the artist with regards to decision making since key decisions would now require voting

2.3 Inferences Drawn and Project Motivation

The qualities of the aforementioned technologies show great promise in building a decentralized music sharing platform over a peer to peer network. The implementation of blockchain ensures a timely, transparent, and traceable payment system. Moreover, the decentralised network layout of the blockchain would permits for the development of a uniform, universal database through the homogenous update and synchronization of data across every node thereby eliminating room for discrepancies. Payments can be made via cryptocurrencies and enforced using smart contracts which automatically distribute the revenue among owners in pre-decided proportions as per the copyright agreement.

Moreover, upon examining other similar projects and initiatives that currently exist, it is evident that two key aspects need to be taken care of namely illegal distribution and developing a platform that levels the playing field for all artists provides ease of use for them to succeed on it. Our project has taken all these aspects into account to build a web-streaming platform which will be discussed in great detail in the following section.

3. Methodology

This section provides a description of the project deliverable and its primary features. It then goes on to provide an overview of the system's underlying design architecture and its various components. It also explains and elaborates the reasons behind certain engineering choices.

4.1 The Deliverable

As suggested in the previous section, BeatChain, a decentralised music streaming platform will be developed over the course of this project. The platform will be implemented as a blockchain-based web application that manages audio file storage, copyright licensing, and royalty collection via its three core features:

1. **A decentralised database** system to store and host the music files uploaded by the artists which are served to listeners upon request after a valid transaction has taken place. This is implemented using IPFS and Infura and the access of these files is restricted to those who pay for streams through an encryption mechanism. This prevents illegal distribution upon purchase.
2. **A decentralised rights ledger** to validate artist copyrights and manage their licensing. This is implemented via smart contracts which manage and validate the copyrights and licenses of the artist. They also manage the availability of the media on the platform where the track is provided to the listener upon a confirmed transaction of the stream. The contract then redirects the payment to the wallet address of the artist listed in the contract
3. **Cryptocurrency wallets** are integrated into the application using MetaMask. This allows for quick, efficient and automatic transfers from the listener to the artist.

Figure 3 provides a visual overview of the operations of these three feature components on the platform.

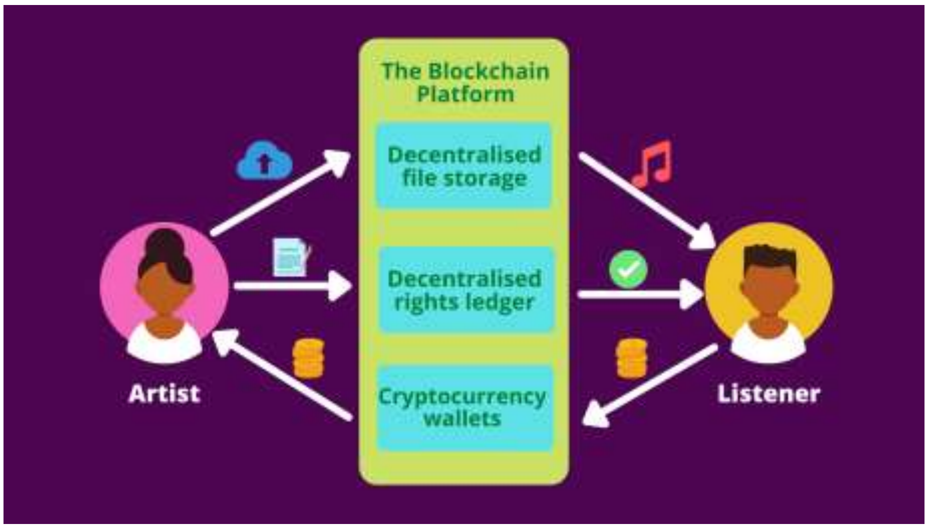


Figure 3. The three core components of BeatChain with their functions

4.2 Design Architecture

This section describes the design architecture of our application with regards to its constituting components and the interactions between them. A diagram of the system architecture is illustrated in Figure 4

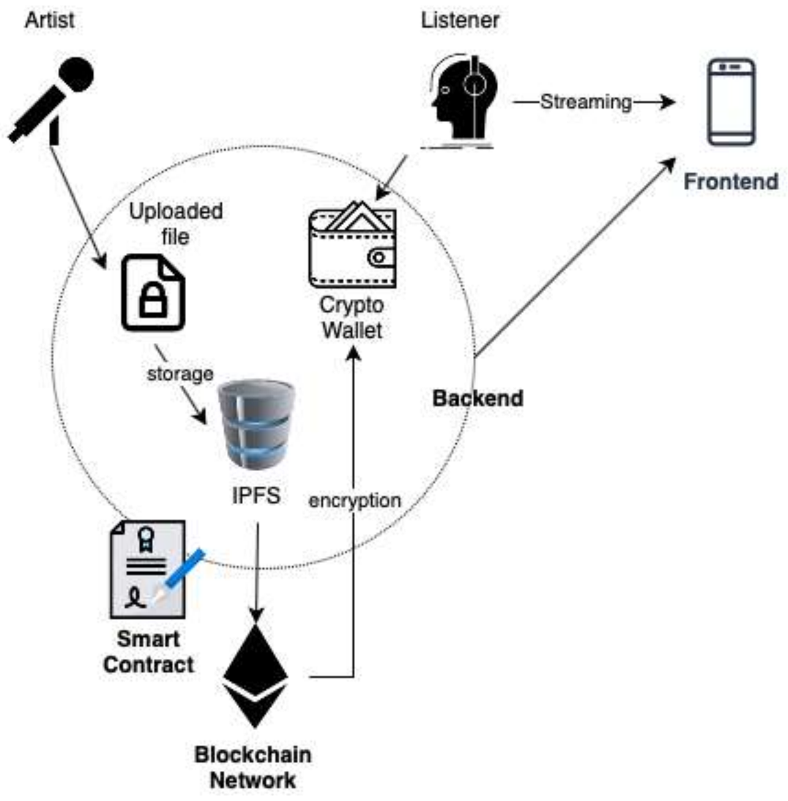


Figure 4. System architecture diagram for BeatChain

4.2.1 The Underlying Blockchain Platform

Presently, Ethereum and Hyperledger Fabric are two of the most sought after blockchain development platforms with regards to their developments tools, performance, and consensus mechanisms[13]. Upon careful analysis of their respective features, benefits, and drawbacks, Polygon, a layer two implementation of Ethereum has been chosen to be the most suitable platform for developing this project.

Ethereum is an extensively tested and well-documented open-source platform. As a result, it effectively assists the development process and provides the essential functionalities of any desirable blockchain platform with regards to initialising, validating, and executing transactions. Moreover, Ethereum's ability to support both private and public platforms makes it more suited to conducting Business-to-Consumer (B2C) transactions. Since the general public is the audience targeted by our project, Ethereum is favoured over Hyperledger Fabric which only permits predefined users to access the blockchain. Polygon convenient right now to test.

The central utility of Ethereum lies in its capability of supporting smart contracts. This allows for programmability within the blockchain where the contracts can execute code segments upon encountering specified trigger events. A more detailed implementation pertaining to the project is provided in the next subsection. Ethereum also offers the benefit of Ether (ETH) – its native cryptocurrency for making transactions whose popularity ranks second to Bitcoin as of August 2021 [14].

However, the Ethereum network by itself has some limitations as a blockchain development platform in terms of low throughput, risk of clogging and a non-customizable technology stack. It is also more expensive to store data directly onto Ethereum from a development perspective. To deal with this, Polygon along with Matic - its associated cryptocurrency is used as the underlying blockchain network. Polygon is a framework which serves the purpose of developing and connecting to networks that are Ethereum compatible [15]. It also provides the benefits of higher security alongside lower gas fees for processing transactions and faster speeds for the same [15]. It has adaptor modules and a protocol to facilitate the exchange of messages with Ethereum [15].

4.2.2 Smart Contracts

Smart contracts form the most crucial element of the project's functionality. These contracts are written in the Solidity programming language since it is compatible with the Ethereum Virtual Machine (EVM) paving the way for seamless integration [16]. Solidity also allows for complex data types and member variables. The provision of the Application Binary Interface (ABI) is useful in keeping track of type mismatches with regards to the information stored in the contract.

When a music file is uploaded by the artist, the "upload" event in the contract is triggered and the blockchain creates a smart contract containing the relevant ownership information. Some gas fees is deducted and paid to the contract upon user confirmation. This information stored comprises:

- The song title
- Artist name

- Cost per stream specified by the artist
- The song hash
- The album art hash
- The cryptocurrency wallet address of the author

This information remains consistent and is tamper-proof. Once the song has been uploaded and stored in the contract, none of these variables of the song structure can be changed.

Another aspect of the contract pertains to payments, when a user clicks on a song on the platform to stream it, a “stream” event is triggered and the value of the cost per stream entered by the artist and stored in the contract is deducted from the listener’s wallet and paid to the artist’s wallet listed in the contract.

In order to develop, prototype and test smart contracts, the blockchain pipeline has been implemented using Truffle which is a development environment for EVM based blockchains. This helps facilitate the implementation of smart contracts. MetaMask was used for testing these contracts using the Polygon Testnet due to the ease of availability of test Matic from the polygon faucet.

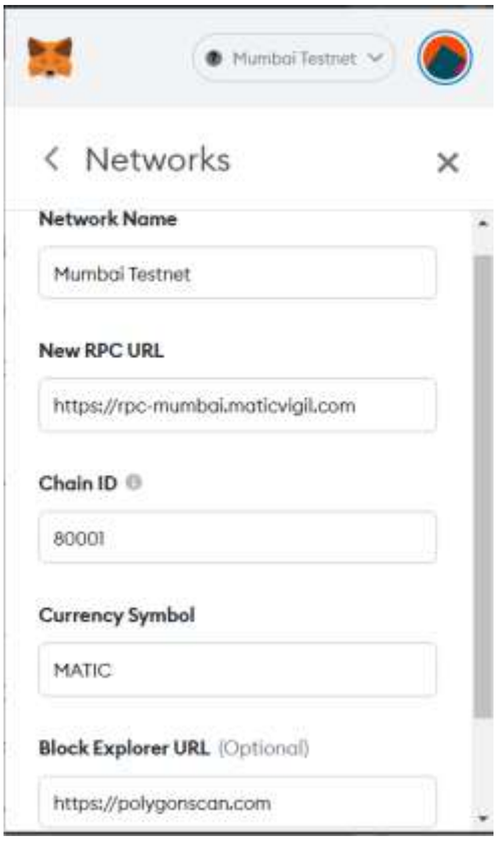
4.2.3 MetaMask

MetaMask was chosen as the cryptocurrency wallet to link to the platform since it provides for direct and secure interaction with the Ethereum blockchain underlying our Polygon layer [17]. Since MetaMask is available as both a browser extension and a mobile app, users can easily access their wallet and interact with the music available on our platform.

MetaMask also allows for users to own their data by generating keys and passwords on the user’s device giving them sole access to their accounts.

To interact with our Polygon blockchain using MetaMask, we added a new Remote Procedure Call (RPC) endpoint by specifying the network name, RPC URL, and Chain ID along with the currency symbol and block explorer URL.

Figure 5 depicts this setup process.



Network Name
Mumbai Testnet
New RPC URL
https://rpc-mumbai.maticvigil.com
Chain ID
80001
Currency Symbol
MATIC
Block Explorer URL (Optional)
https://polygonscan.com

Figure 5 Setting up an RPC endpoint to link MetaMask to Polygon

4.2.4 The Backend

The application's backend application programming interface (API) provides important functionality with regards to encrypting the the audio files uploaded to the platform via IPFS. It also provides routes for storing the album art uploaded to IPFS as well as the remaining song details from the frontend of the application to our database for which we are using MongoDB. We chose MongoDB for its documented-oriented data model which provides flexibility adding and changing database fields. This is beneficial is long term and evolution of our application if changes need to be made to the song structure for instance.

To design the API, Express was used as the web framework on top of Node.js - a JavaScript runtime environment for server-side execution of JavaScript. Express provides libraries to address numerous aspects of the web development process and simplify it. Following the use of JavaScript to build the server-side, Web3.js was employed to connect the Node.js server to the Ethereum chain. Web3.js is a collection of JavaScript libraries that facilitate interaction with a local or remote Ethereum node using HTTP requests [18].

4.2.5 The Frontend

The client-side of the project presents the users with a Graphical User Interface (GUI) for interacting with the application. This comprises of prompting the user to link their MetaMask wallet to the application on loading and access it henceforth. The users can then interact with the platform in the following ways:

- View the tracks that have been uploaded to the chain
- Search for songs by title or artist name
- Sort and filter tracks by release, artist name and title name (alphabetically)
- Stream tracks
- Upload their own music
- Create playlists to store selected tracks

Figure 6 gives and overview of the application’s frontend interface.

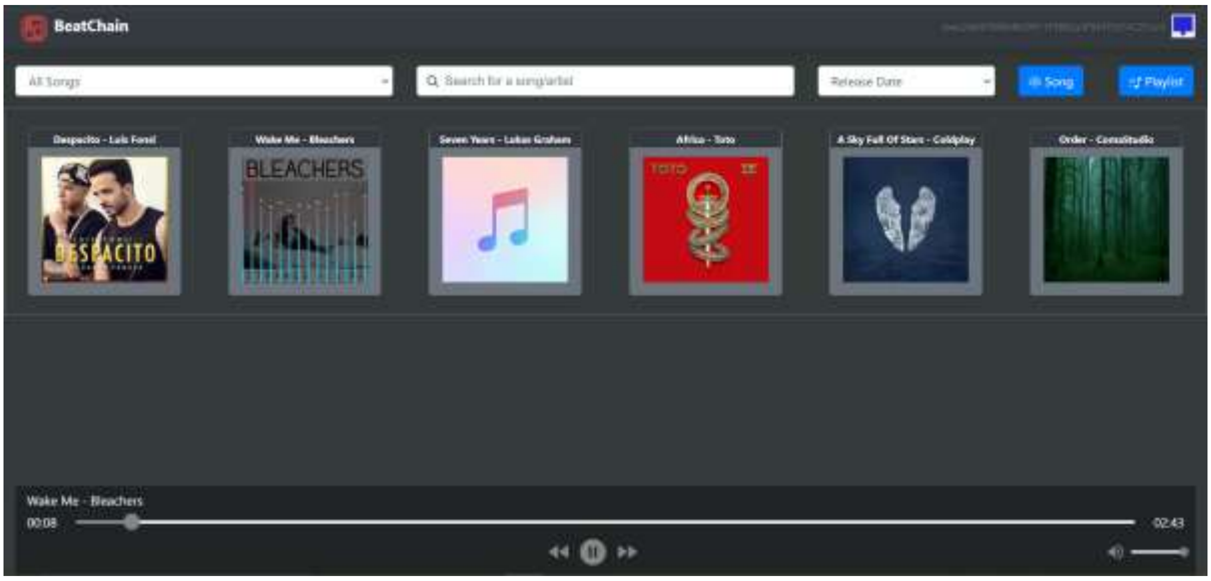


Figure 6 Frontend interface overview for BeatChain

While functionalities of searching, sorting and creating playlists interact with the backend API of the application, other functions such as uploading tracks interact with the blockchain using the API. Streaming tracks interacts with the chain.

ReactJS was used to develop the frontend of the application since it is the industry standard framework to develop responsive applications through flexible and reusable UI components. It also provides for easy connectivity to our NodeJS backend.

4.2.6 Peer to Peer Data Storage

From a development perspective, storing large amounts of data in the form of files on and Ethereum blockchain would grow considerably expensive. To manage this, we linked our chain to Inter Planetary File System (IPFS) which is a distributed peer to peer storage system [19]. It serves the purpose of storing and accessing files via their cryptographic hashes stored on nodes on the blockchain.

However, if the IPFS node is down for some reason, the audio file may become temporarily unavailable. Infura was used to solve this problem. Infura is a Web3 infrastructure that makes access to the blockchain faster by connecting instantly to the Ethereum and IPFS networks [20]. Infura pins uploaded IPFS files and keep them available to the network in a reliable manner. It will also accommodate scaling as the number of files grows.

4.2.7 Note on Semi-decentralised Architecture

From the design architecture that has been described thus far, it is evident that although decentralisation is a fundamental property of blockchain, our application is organised by a semi-decentralised network architecture. This is due to the fact that blockchain based systems can take a significant amount of time in processing transactions [21]. This includes the time taken to create a new block, process the gas payment and verify the transaction. From a user-experience point of view, this is already a hindrance. Apart from streaming and uploading, if searching, sorting and encryption were also to be performed on chain, it would be quite computationally expensive and unfeasible.

Combining the blockchain solution with a backend API in a manner that extracts the best utility from both proves to be a useful solution [22]. For this reason, we decided to perform encryption on the backend, and the searching and sorting of tracks on the frontend with support of the backend to enhance the user experience by saving time on computation.

4. Current Status of Project

4.1 Features Implemented and Functionality

This section explains the existing features of the application and their implementation details.

4.1.1 User Login

On launching the BeatChain website, a connection is created between the chain and the user's browser via Web3.js. Then, the user is prompted to log into their MetaMask account for accessing their wallet. Upon a successful login, the website loads and the wallet address of the user is displayed on the top right navigation panel indicating that the platform is now accessible to them. Figure 7 shows the MetaMask login pop-up and Figure 8 shows the case of successful authentication with the wallet address being displayed on the user's browser.

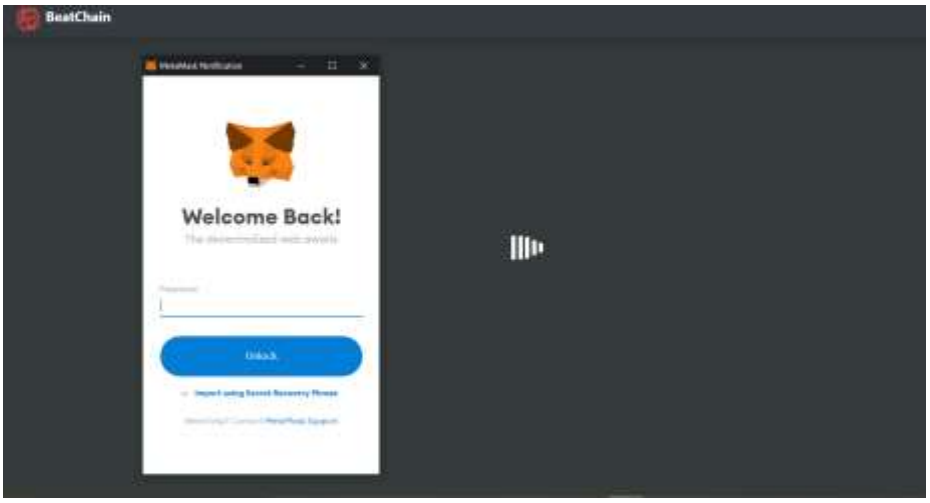


Figure 7. MetaMask login pop-up for user wallet authentication

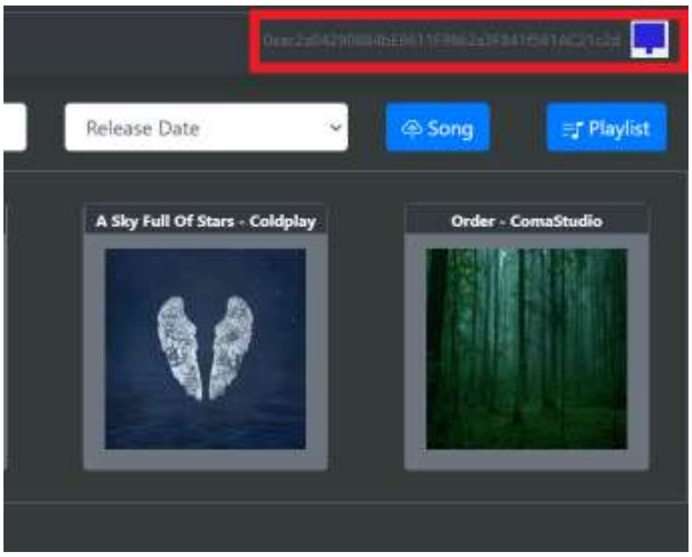


Figure 8. Wallet address of a successfully authenticated user

4.1.2 Song Upload

Every user connected to the BeatChain platform has the option to listen as well as upload their own music. If a user wishes to upload a song onto the platform, they click on the upload button and modal form opens prompting the user to upload a song. Using this form, the user uploads the audio file, the album artwork and the name of the song, artist and the amount they wish to earn per stream in MATIC. While entering the amount, the platform indicates the equivalent amount in HKD to save the user the hassle of doing manual calculations and conversions. These real time conversions are done by fetching data from CoinGecko's API. Figures 9 and 10 show the upload button and the upload form respectively.

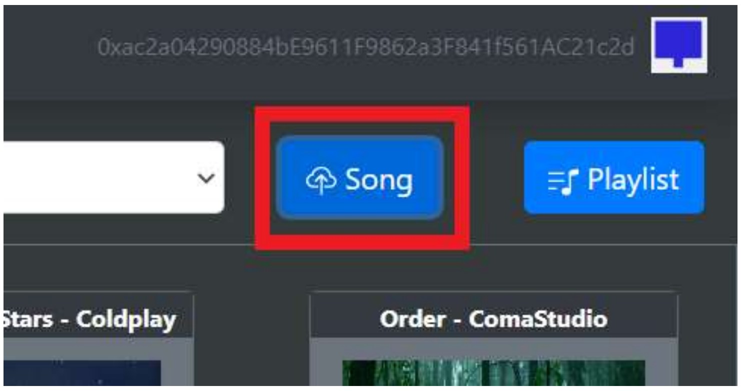
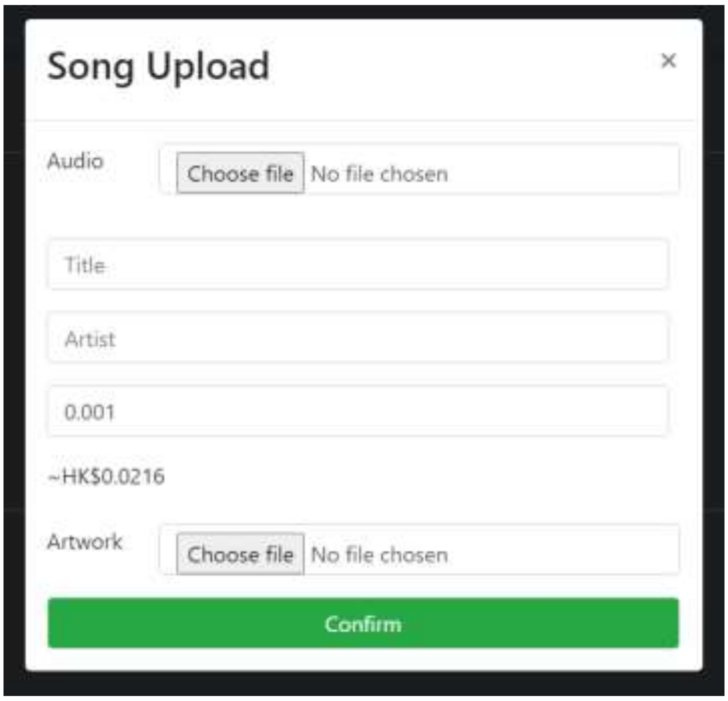


Figure 9. The button through which a musician can upload a track


Song Upload

Audio Choose file No file chosen

Title

Artist

0.001

~HK$0.0216

Artwork Choose file No file chosen

Confirm

Figure 10. The modal form upon pressing the upload button

Upon pressing confirm, a conversion of the audio file is performed from a base64 string to a Buffer object which is sent to the backend. Encryption of the file is performed in the backend following which this encrypted file is uploaded to the IPFS node. IPFS sends back the hash value of this uploaded file to the backend which it then saves to the MongoDB database along with the file's encryption key and the other song details submitted by the user.

Once all the relevant information has been saved to the database, the backend sends this IPFS hash value to the frontend of the application. The frontend then submits this hash and the other song details to the smart contract which saves them. Finally, the song appears on the frontend. Figure 11 shows the new song being appended to the head of the track catalogue.

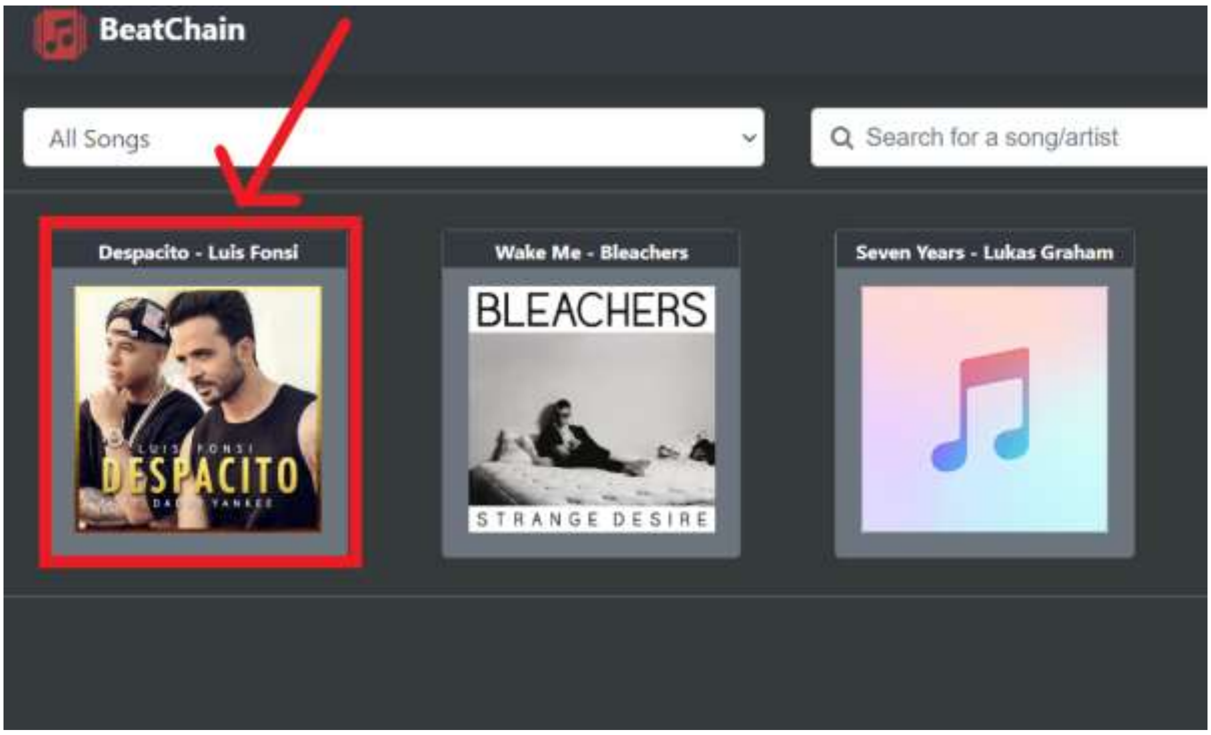


Figure 11. Upon successful upload, new song appended to head of catalogue.

4.1.3 Audio Encryption

Encryption is a key aspect of protecting the rights of the artist against copyright infringement. Originally anyone with access to the IPFS link of the song upon streaming it could have shared it with others. To manage this, before uploading to the IPFS node, BeatChain encrypts the audio file by converting it to a Buffer object through use of a unique key and the 'aes-256-cbc' encryption algorithm.

The Advanced Encryption Standard (AES) – 256 algorithm was chosen since it is currently the strongest encryption standard with a key length of 256 bits, it is able to support the largest bit size making it virtually impossible to break with regards to current computing power [23]. AES achieves this by repeatedly performing the same encryption operation multiple times in order to generate an encrypted message. AES-256-CBC refers to AES using a 256 bit key Cipher Block Chaining mode. CBC makes use of an initialisation vector which is a randomly generated number. Upon breaking the input message into blocks, these blocks are XORed with the vector and each encryption output becomes the next operation's input giving it the name Cipher Block Chaining.

This operation runs 14 times for a 256 bit key to generate the final encrypted message. Similarly, decryption runs 14 times using the same key and vector. AES makes use of symmetric key encryption where one secret key and vector work to cipher and decipher data. For this reason, it is imperative to securely store the key and vector. Figure 12 illustrates how symmetric key encryption works [23].

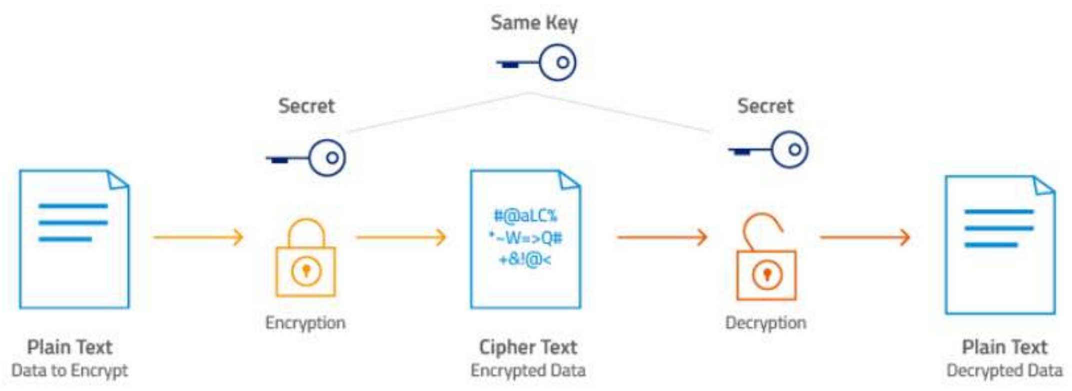
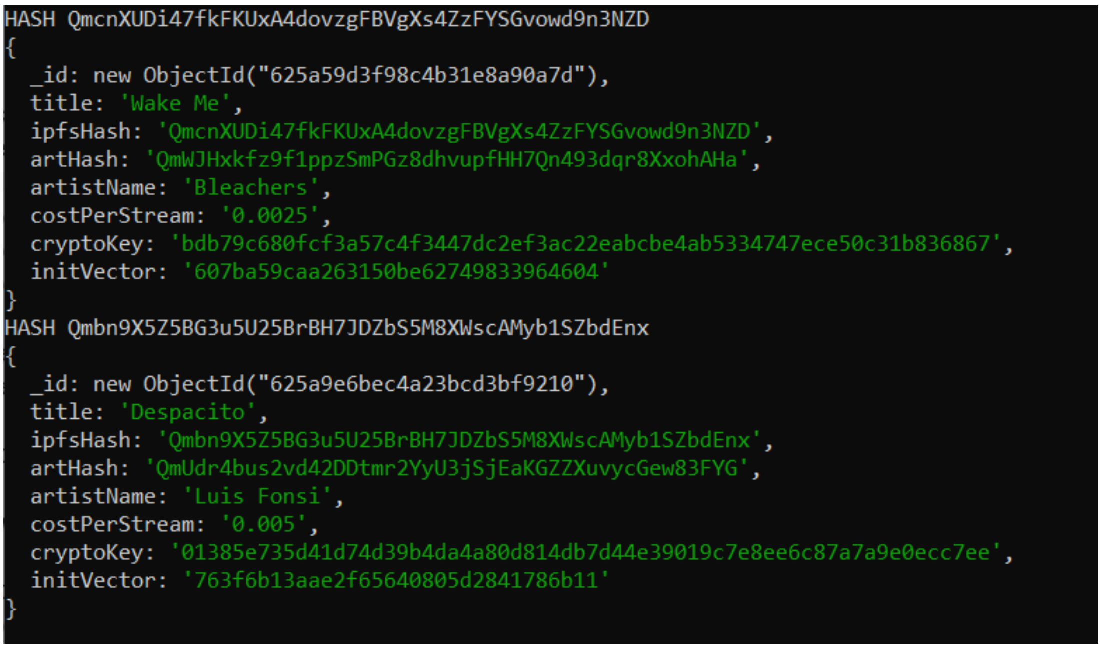


Figure 12. Symmetric key encryption mechanism [23]

In the case of BeatChain, an encryption key is generated for every audio file that is uploaded to the platform and saved to the database after encryption. This key is later used to decrypt the file before sending it the frontend when the user streams a song. Figure 13 shows how encryption data is stored in the database with each song having its own key and initialisation vector (initVector).



```
HASH QmcnXUDi47fkFKUxA4dovzgFBVgXs4ZzFYSGvowd9n3NZD
{
  _id: new ObjectId("625a59d3f98c4b31e8a90a7d"),
  title: 'Wake Me',
  ipfsHash: 'QmcnXUDi47fkFKUxA4dovzgFBVgXs4ZzFYSGvowd9n3NZD',
  artHash: 'QmWJHxkfz9f1ppzSmPGz8dhvupfHH7Qn493dqr8XxohAHa',
  artistName: 'Bleachers',
  costPerStream: '0.0025',
  cryptoKey: 'bdb79c680fcf3a57c4f3447dc2ef3ac22eabcbe4ab5334747ece50c31b836867',
  initVector: '607ba59caa263150be62749833964604'
}
HASH Qmbn9X5Z5BG3u5U25BrBH7JDZbS5M8XWscAMyb1SZbdEnx
{
  _id: new ObjectId("625a9e6bec4a23bcd3bf9210"),
  title: 'Despacito',
  ipfsHash: 'Qmbn9X5Z5BG3u5U25BrBH7JDZbS5M8XWscAMyb1SZbdEnx',
  artHash: 'QmUdr4bus2vd42DDtmr2YyU3jSjEaKGZZXuvycGew83FYG',
  artistName: 'Luis Fonsi',
  costPerStream: '0.005',
  cryptoKey: '01385e735d41d74d39b4da4a80d814db7d44e39019c7e8ee6c87a7a9e0ecc7ee',
  initVector: '763f6b13aae2f65640805d2841786b11'
}
```

Figure 13. Structure of encryption data stored in MongoDB database

4.1.4 Stream Music

After browsing the catalogue, when the user has decided upon a song they would like to stream, they are able to see how much a stream costs by hovering over it as presented in Figure 14.

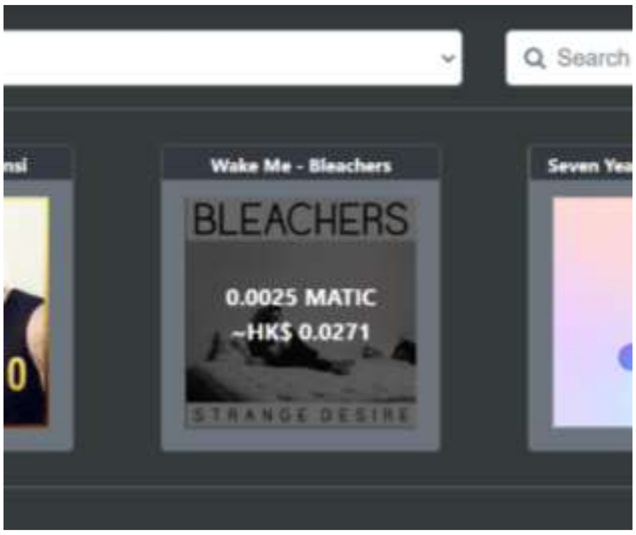
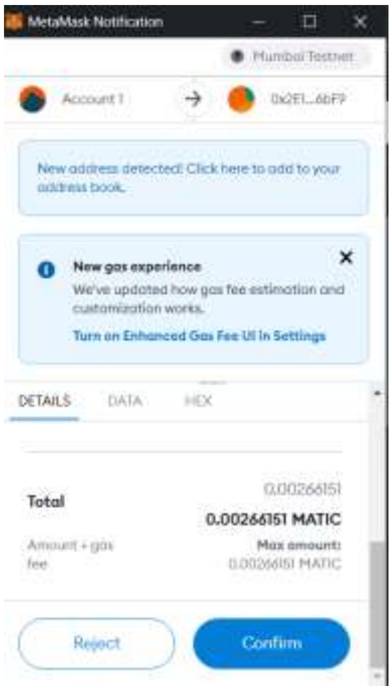


Figure 14. User can view stream cost before clicking.

On choosing to proceed, the song ID and its IPFS hash is retrieved from the contract and MetaMask notifies the user of the total amount that will be deducted from their wallets factoring in the gas fees shown in Figure 15.


MetaMask Notification

Mumbai Testnet

Account 1 → 0x2E1...6bF9

New address detected! Click here to add to your address book.

New gas experience

We've updated how gas fee estimation and customization works.

Turn on Enhanced Gas Fee UI in Settings

DETAILS DATA HEX

Total	0.00266151 0.00266151 MATIC
Amount + gas fee	Max amount: 0.00266151 MATIC

Reject Confirm

Figure 15. MetaMask pop-up indicating the cost stream + gas fees

After confirming to move forward with the transaction, the amount of the cost per stream is sent to the contract. When the transaction has completed successfully, the song ID and its IPFS hash are sent to the backend which makes use of this information to request the file from IPFS and to locate the song's encryption key. After decrypting the file using the key, it is returned to the frontend as a base64 string which is then converted to a Blob (binary large) object. A unique URL is generated for this Blob which is sent to the audio player as the source of the song. Along with the listing on the player, a visualizer is used to indicate the current song playing (see Figure 16).

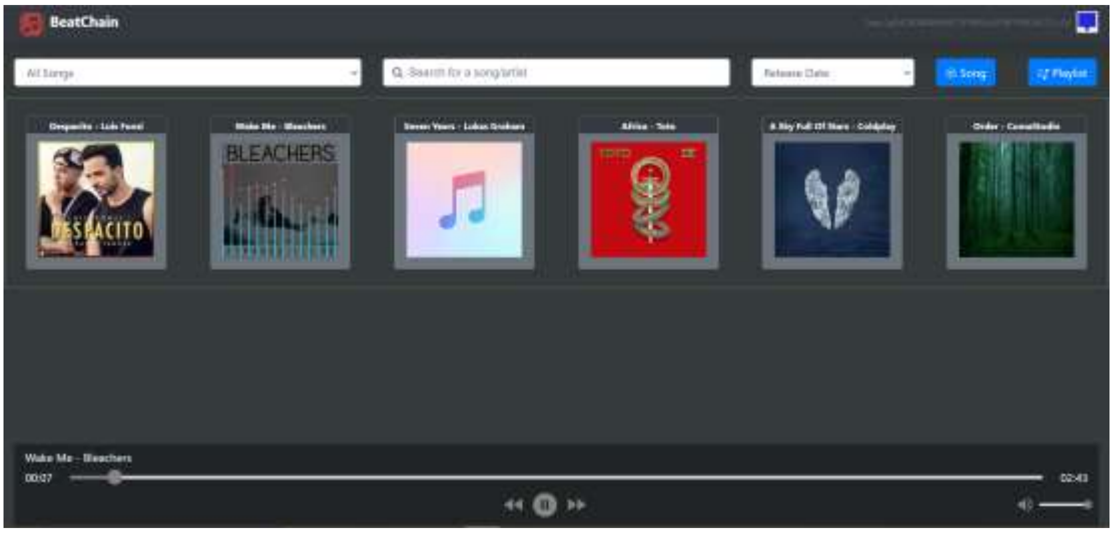


Figure 16. Song successfully plays after transaction has been processed.

4.1.5 Searching and Sorting

BeatChain also provides users with the functionality to search for tracks and sort them by release data, song title and artist name (Figure 17). These functions are performed on the frontend where we have saved the array of songs currently uploaded to the chain as a React state variable. When the user searches for a song, the elements of the array are filtered so as to match the user’s search query. Similarly, when sorting, songs are rendered in the output array according to the filtered applied based on the criteria.

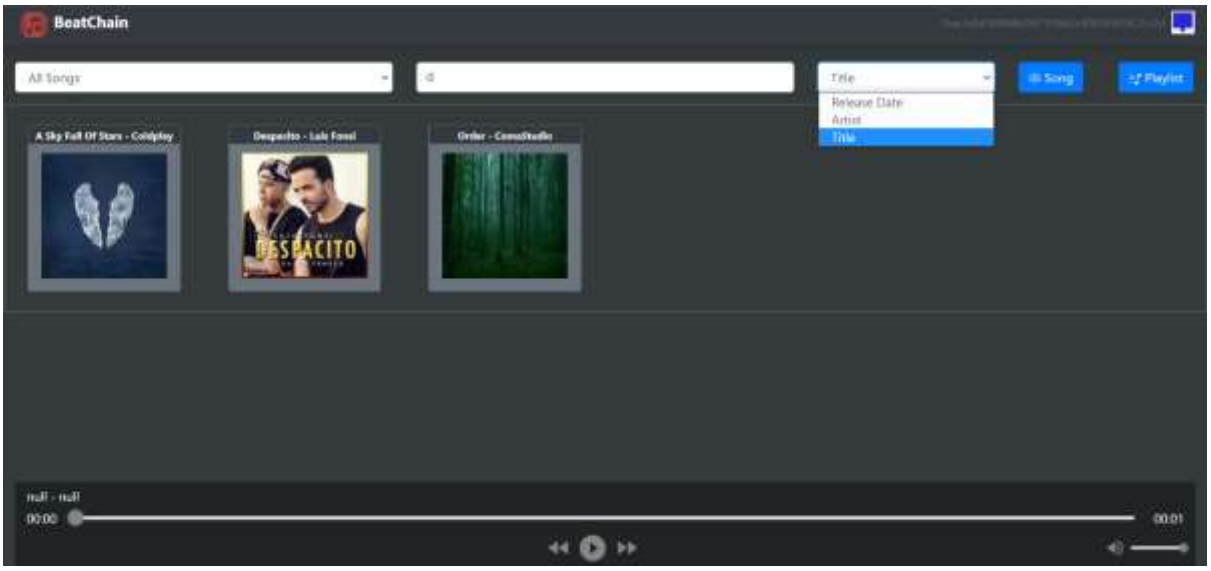


Figure 17. BeatChain's searching and sorting functionality in action

4.1.6 Playlist Creation

Finally, BeatChain gives users the option to create their own custom public playlists. When the user clicks on the create playlist option, a modal form opens prompting the user to enter the name of their playlist (see figure 18 and 19).

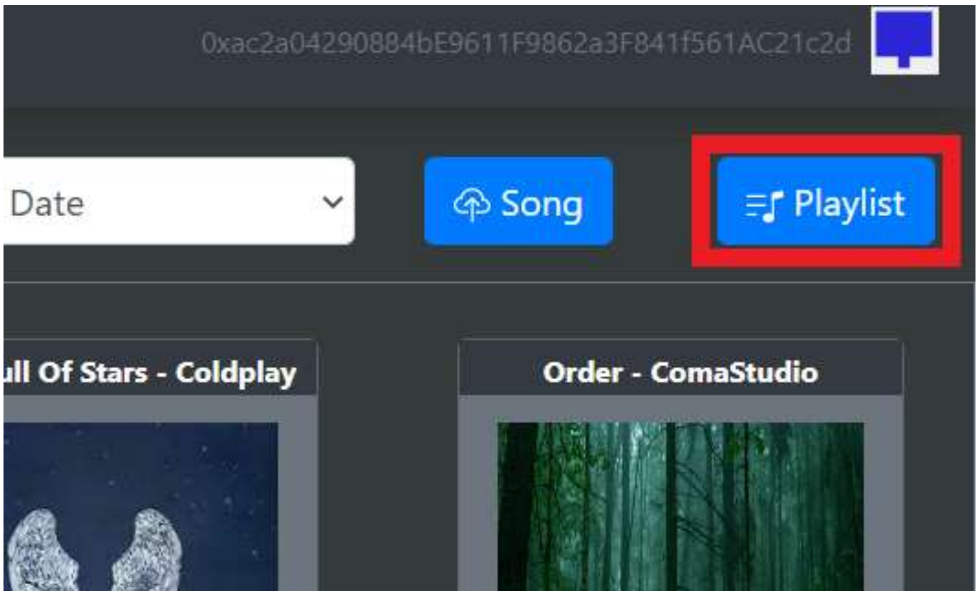


Figure 18. The button to create a new playlist

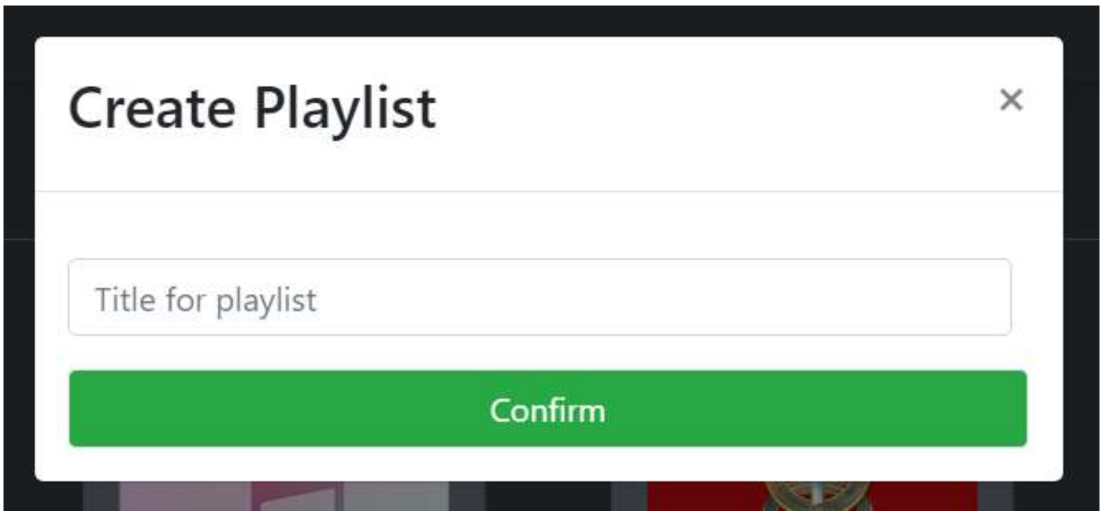


Figure 19. The modal form which appears on creating a new playlist.

The user can then locate their playlist using the dropdown menu on the top right (Figure 20) and select the songs they wish to add as shown in Figure 21. The user can also browse playlists that have been created by other users.

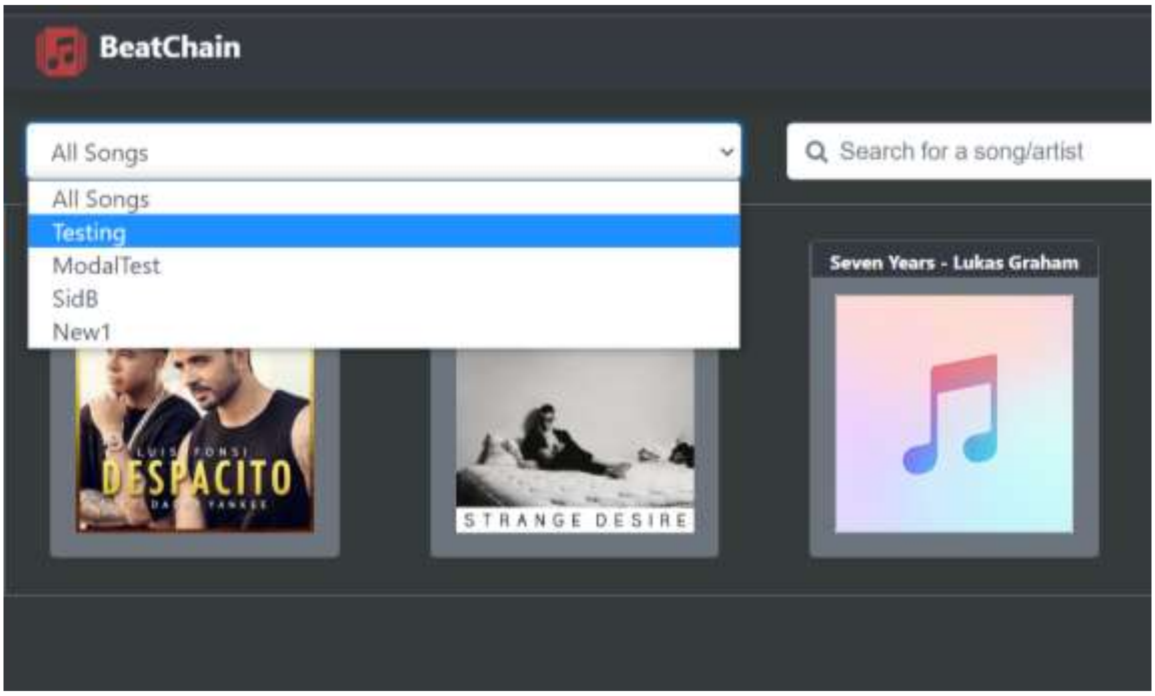
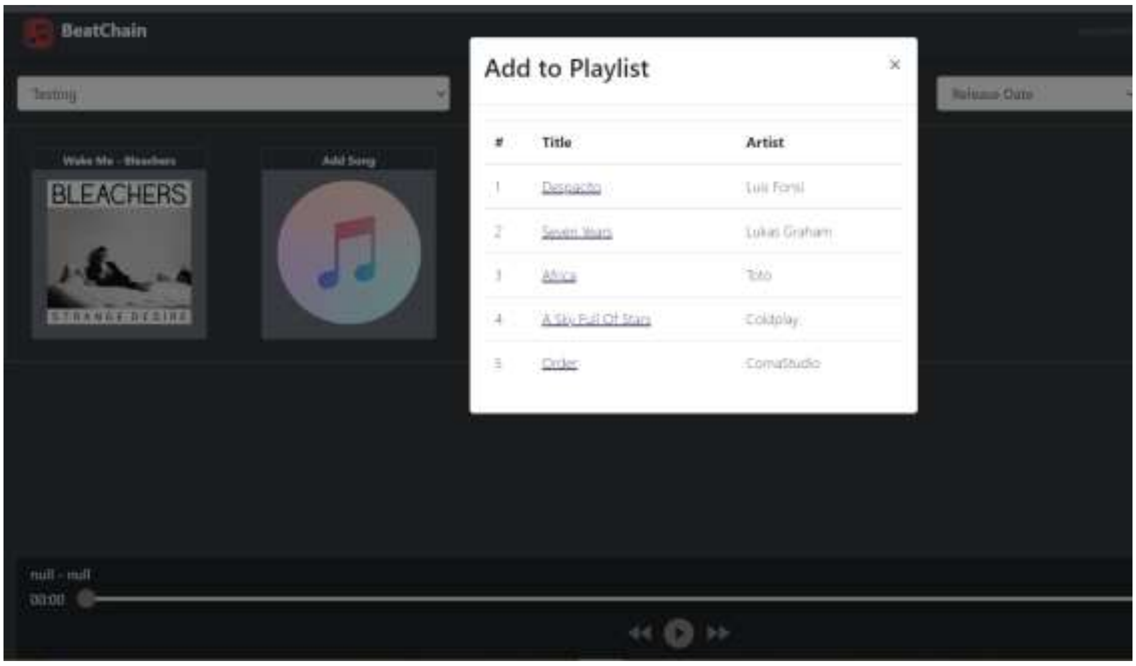


Figure 20. The dropdown that lets users browse all playlists created on the platform.

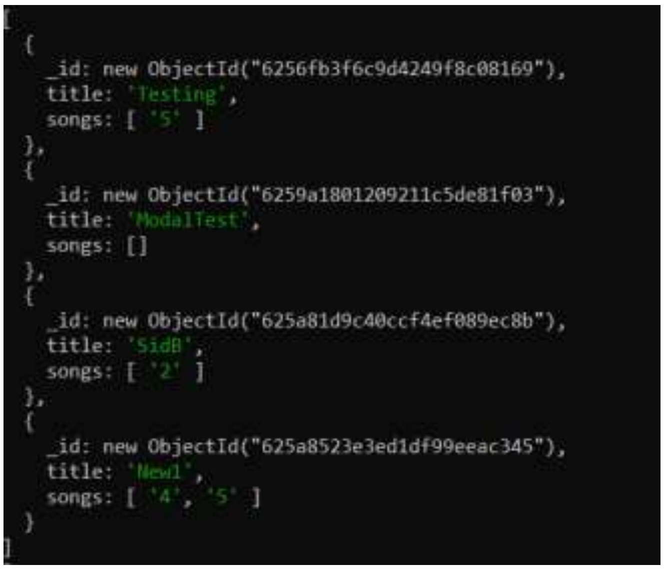


Add to Playlist

#	Title	Artist
1	Despacito	Luis Fonsi
2	Seven Years	Lukas Graham
3	Africa	Toto
4	A Sky Full Of Stars	Coldplay
5	Order	ComaStudio

Figure 21. The modal which appears when user chooses to add songs to a playlist.

This functionality has been implemented in the following manner: When the user clicks on the “Playlist” button and enters the title, a POST request is sent to the backend which creates a playlist object comprising of the title and an empty array of song IDs. This is saved to the database and a response is sent to the frontend which consists of the playlist ID and the title of the playlist. Every time the web platform loads, a GET request is sent to the backend to fetch all the playlists from the database. The database sends a list of the all playlists created on BeatChain to the backend (see Figure 22) which in turn is sent to the frontend and displayed in the dropdown menu of Figure 20.



```
[
  {
    _id: new ObjectId("6256fb3f6c9d4249f8c08169"),
    title: 'Testing',
    songs: [ '5' ]
  },
  {
    _id: new ObjectId("6259a1801209211c5de81f03"),
    title: 'ModalTest',
    songs: []
  },
  {
    _id: new ObjectId("625a81d9c40ccf4ef089ec8b"),
    title: 'SidB',
    songs: [ '2' ]
  },
  {
    _id: new ObjectId("625a8523e3ed1df99eeac345"),
    title: 'New1',
    songs: [ '4', '5' ]
  }
]
```

Figure 22. Storage schema of playlist data in the database.

When the user selects a playlist, filters out the list of songs from its properties and displays them on the platform. Should the user wish to add more songs to their playlist, a modal appears which presents the songs that are not currently in the playlist. On selecting a song, a POST request is sent to the backend with the playlist ID and song ID. The database locates the playlist and adds the specified song to it. The updated playlist is then sent to the backend which sends the same to the frontend.

5. Challenges and Limitations

This section outlines some of the previous challenges we encountered in the development of this project application. It then goes on to discuss the current limitations faced by BeatChain in the aspects of the fallibility of encryption, need for lower gas fees and faster transaction processing times.

5.1 Solutions to previous challenges

i) While Ethereum was initially chosen to form the underlying blockchain network of the application, it was noted that storing large amounts of data on Ethereum is considerably expensive from a development perspective and synchronizing numerous large files across the network could waste its resources. As a result we switched to pursue development using Polygon, a layer 2 scaling solution upon Ethereum covered in more detail in sections 2.1.3 and 4.2.1.

ii) Another challenge pertained to maintaining access control in a decentralised setting where we wanted to restrict users from accessing audio files unless they have been paid for since the file stored on IPFS could easily be shared with others once accessed. There were two approaches considered: The first one was concerned with encrypting the IPFS hash in the backend with a secret key which would be returned on streaming the song. The second approach was concerned with directly encrypting the uploaded audio file itself. We decided that proceeding with the second approach would be more secure since the user would only be able to access the encrypted audio file even if they were able to obtain the IPFS hash. Implementation details of this approach are given in section 4.1.3.

iii) The third important challenge surrounded provided UI functionalities of searching and sorting among tracks on the platform. Since the song data is stored on IPFS and in the smart contracts, performing searching and sorting through the chain would have proven to be highly computationally expensive and compromise the user experience. Accordingly, we chose to move forward with a semi-decentralised approach while partially relies on a backend API (covered in section 4.2.7) and perform these functions on the frontend with support of the backend (as described in section 4.1.5).

5.2 The shortcomings of encryption

Although BeatChain follows in the footsteps of other big streaming services such as Netflix, Hulu, HBO etc in performing digital rights management (DRM) and protecting their content against copyright infringement through encryption, this approach is not entirely effective.

Currently, BeatChain content is still vulnerable to being stolen in the following manner: once the streaming transaction is complete, the backend sends the decrypted audio file to the frontend where a temporary URL is created to source the audio player. At this point, it is possible to retrieve this URL and disseminate it.

However, content on the other major platforms is also still susceptible to piracy despite using sophisticated DRM technologies such as Widevine. Since data is required to be stored on the user's machine in order to be streamed, this loophole is exploited by specialised tools which automate the process of downloading the encrypted media stream and work towards reverse engineering the encryption [24]. Netflix loses around $200 million every month due to piracy [24]. While a solution to this is yet to be found, emphasis is currently being laid on developing attractive business models which dissuade consumers from turning to pirated sites [24].

6. Schedule & Milestones

This section assesses the progress made by our group over the course of the project and takes into consideration the next steps involved in the future of developing this application.

6.1 Progress Evaluation

Figure 23 and Table 1 outline the development schedule followed for the project. We have completed all three phases of our deliverables that we originally set out to complete.

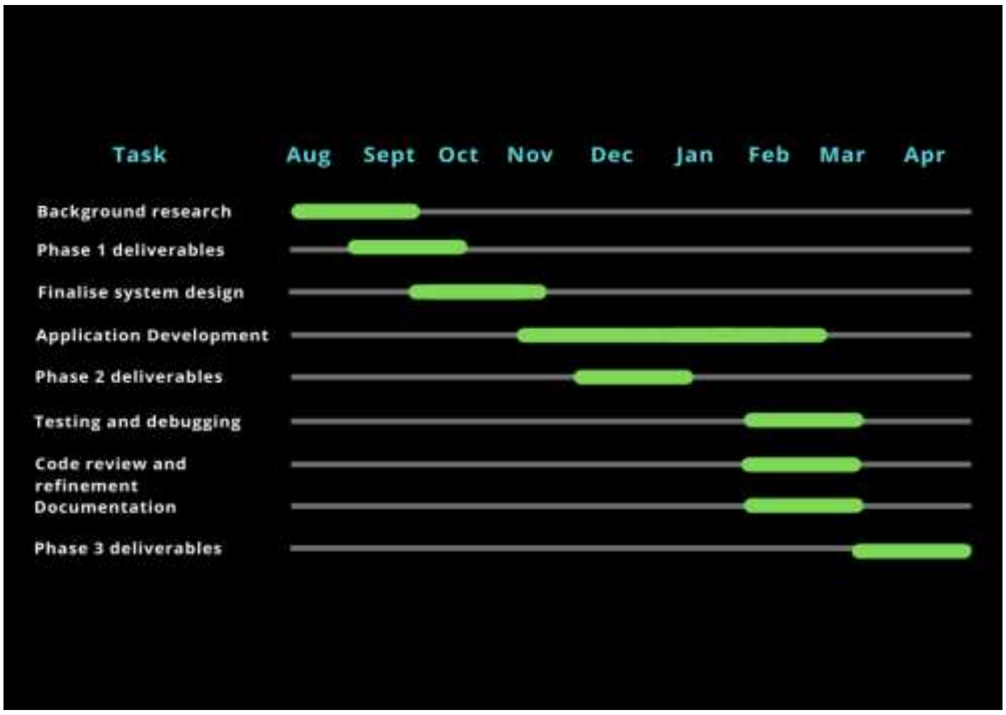


Figure 23. Gantt chart of the development schedule followed in project development

.During phase 1, we worked on developing our project proposal and focused on developing our domain knowledge of blockchain and decentralised applications (DApp). We also worked on designing the system architecture of the application and began working developing the chain and establishing interaction between the chain and the frontend.

During phase 2, the functionality of uploading songs to the chain was developed by writing smart contracts and linking the application to MetaMask. The functionality of streaming a song on payment was also built.

During phase 3, the backend of the application was created and linked to a database as well as the frontend of the application. We also worked on an encryption mechanism for audio files uploaded to the platform and modified the structure of songs stored in the smart contract to include more information fields. The payment function was modified in the smart contracts so as to execute payments based on the amount specified by the musician. Searching and sorting functionalities were added as well as the feature to create playlists. Lastly, we worked on enhancing the UI to make it more responsive and built a custom audio player along with a music visualiser to make the interface more appealing to the user. We also added a real time Matic to HKD converter to make it convenient for both the musicians and listeners to be aware of their charges and expenditure. Lastly, the application has been deployed to this link.

Time Period	Tasks Planned/Deliverables	Status
August 2021 - September 2021	Project topic deliberation and background research	**Complete**
October 2021 - November 2021	• Phase 1 Deliverable: ◦ Project proposal report ◦ Project webpage • Enhancing our domain knowledge and development capabilities regarding blockchain and smart contract development over the Ethereum Platform • Finalising the system design of our application • Commencing application development	**Complete**

December 2021 - January 2022	• Development continues
	• Phase 2 Deliverable:
	  ◦ Interim project report
	  ◦ First project presentation
		Complete

February 2022 - March 2022	• Completion of application development • Testing and debugging • Code review, documentation and refinement	**Complete**

April 2022	• Phase 3 Deliverable:
	  ◦ Final Project Presentation
	  ◦ Final Report
		Complete

Table 1. The development schedule for the project

6.2 Next Steps

This section discusses the next steps in the development of the application.

6.2.1 Including more authors for a song

Currently, the application enables artists to upload their work to the platform on the assumption that they are the sole owners of their work. However, music production is a complex and evolving process and it is important to factor in the dues owed to producers, instrumentalists, additional lyricists and featured artists who are also entitled to portions of the copyright.

Taking this into account, the logical next step would be to modify the smart contracts and frontend upload modal by adding more fields that factor in the cases where more than one party is involved in the recording and production process. The uploader can specify the names and wallet addresses of the other involved parties as well as the percentage of the total revenue per stream owed to each of them. These additional details would be saved to the contract and each time a song is streamed, everyone would automatically receive their revenue in a pre-decided manner.

6.2.2 Rating system and comments section

Building a rating system and comments section within the BeatChain application would help foster a sense of community where users come together to support up and coming artists and help with user retention on the platform.

The rating system could be implemented by maintaining an additional database for song ratings comprising of the song ID and rating, and number of ratings fields. Whenever a listener rates a song, a POST request is sent to the backend consisting of the song ID and new rating. The backend uses the song ID to fetch the song from the database, increments the number of ratings by 1 and recalculates the average ratings based on this new value. This is saved to the database and also sent to the frontend to be updated. The comments section would be implemented in a similar manner.

6.2.3 Recommendation Systems

As the amount of music uploaded to the platform grows, it may get increasingly difficult for listeners to navigate and find songs they might enjoy. As a result, it would be valuable to build a recommendation system using machine learning models which would suggest to users the music they might enjoy. For instance, this could be done by tracking the genres listened to by a user and the frequency of streams so as to recommend music from similar genres. The application could also recommend songs and artists based on what else listeners of a particular artist listen to. In this way, each user could have their personalised home page of recommendations when the application is loaded.

6.2.4 Public and Private Playlists

Currently, all playlists on the BeatChain platform are public. However, some users may not wish to share their playlists. Based on this, users can be provided with the option of creating either public or private playlists. This can be implemented by storing a user ID to every playlist in the central database. If a user chooses to make their playlist private, the backend would fetch the playlist data from the database and send it to the frontend where it would be rendered only to the user whose ID is specified in the database. On the other hand, a public playlist could have 'ALL' specified in the user ID field and accordingly be rendered to every user's application. Another approach could involve associating an uploader address a Boolean value with the newly created playlist with the backend/DB interaction being implemented in a similar manner.

6.2.5 Mobile and desktop application

BeatChain is currently available as a web application. We aim to expand it to a desktop and mobile application as well. The desktop application would provide better data security and a more optimized performance while with a mobile application, listeners can stream music on the go.

7. Business Model

This section explains the business model we propose for our application. It begins by weighing the pros and cons of pay per stream vs subscription models and sheds light on our decision to move forward with a pay per stream model. Following this, a market comparison is done with current streaming giants: Spotify and Apple Music. Finally, a cost analysis is performed between BeatChain, Spotify and Apple Music from a consumer perspective.

7.1 Pay Per Stream vs Subscription Model

A pay per stream (PPS) model would involve the listener paying individually for each song they choose to stream while a subscription model would give listeners access to the entire streaming catalogue for a monthly fee. Table 2 compares the benefits and challenges of both models.

	Pay Per Stream	**Subscription**
Benefits	• It is flexible and affordable. It relieves the consumer off the commitment and pressure to use the service. Consumer can stream at their discretion • The revenue is distributed to only those artists whose music has been streamed on the platform • Advertisements are not required to sustain the financial health of the platform. Moreover, consumers often see advertisements as an inconvenience to the user experience	• Once consumers have paid the subscription fee, incentive has been created for them to use the platform. This makes it easier to manage the fickle-mindedness of the consumer by inserted of an element of stickiness. • With a subscription model being more predictable, the business can add data-driven value to the product. • Re-subscription rates are likely to increase.
Challenges	• Customer retention becomes challenging when consumers are not entirely convinced that this model is more economically viable for them. There is also less incentive to use the product when subscription fee hasn't been paid in advance.	• The upfront cost is generally higher and the model demands a more long term commitment from its users by making annual plans cheaper than monthly plans. • Risk of unsubscription increases when subscribers

	• Predicting revenue streams becomes difficult since they are contingent to irregular monthly streams.	feel guilty of overspending when not fully utilising the platform. • Royalties are distributed unfairly to the artists. A paid subscriber's money is redirected towards artists with greater streaming share even if the subscriber does not listen to those particular artists.

Table 2. Benefits and challenges of the pay per stream model vs the subscription model.

Thus, it can be inferred that the subscription model benefits the platform and its business to a much greater degree than it benefits the artists. While it is a good and viable way to earn revenue, very little is done to support the livelihood of the artists releasing their work on the platform.

The pay per stream model provides benefits for both the listener and the artists. It also captures the essence of a decentralised blockchain based application in a more apt manner. It does away with the need for a centralised party to control the platform and peers on the blockchain network can directly interact and support each other's work. Taking all these considerations into account, a pay per stream model would be more fitting for BeatChain where our platform takes a portion of the stream cost roughly in the range of 15% - 25%. The remaining amount is directed to the artist.

7.2 Market Comparison of Streaming Giants

Having justified the viability of a pay per stream model, we conducted a market analysis of Apple Music and Spotify – the two streaming giants in the music industry. Utilising the revenue and usage statistics of each platform for the year 2022, we reviewed how much an artist earns off each platform.

7.2.1 Apple Music

- On average, the platform pays US $0.0076 (HK $0.06) per stream. This varies based on the subscription charged in different countries.
- Out of this amount, the publishers take a ~50% cut and record labels take approximately 10% - 35% based on the deal signed between all parties.

- This leaves the artist with roughly 15% - 40% of the amount paid per stream. This amounts to roughly HK$ 0.009 – HK$ 0.024 per stream. Moreover, in the cases the artist does receive 40% of the revenue, they are provided with inadequate services on behalf of the record label.

7.2.2 Spotify

- Spotify does not release direct data on the royalties distributed through their platform. Moreover, with Spotify there is also the case of larger and more popular artists getting paid regardless of whether their music has been streamed by a paid subscriber.
- The amount paid per stream approximately between the range of US $0.0026 – 0.0049. On average this equates to $0.00375 (HK$0.029)
- Again, the publishers take a ~50% cut and record labels take approximately 10% - 35% based on the deal signed between all parties.
- In a similar manner, this leaves the artist with roughly 15% - 40% of the amount paid per stream. This amounts to roughly HK$ 0.00435 – HK$ 0.0116 per stream.

BeatChain on the other hand would give the artists around 75-85% of the revenue from the stream after taking a 15-25% cut specified in the previous section. This with BeatChain the revenue earned by the artist can be up to four times higher and playing field for all artists is levelled.

7.3 Cost Comparison Analysis

In this section, we compare the monthly cost incurred by a listener among Spotify, Apple Music and BeatChain.

- The regular plans on both Spotify and Apple Music cost HK$ 58 per month.
- Although BeatChain currently gives artists to the freedom to determine what they wish to charge for their music, considering that an average stream amounts to a revenue of HK$0.02 (or 0.00183225 MATIC):
    - The artist would receive ~HK$0.016 on a 20% commission taken by BeatChain
    - At this cost, the user would pay HK$20 per month for around 1000 streams
    - For the HK$58 subscription fee charged by Spotify and Apple music, the user could get up to 58/0.02 = 2900 streams.
    - With the average length of a song being 4 mins, the user could get an average monthly listening time of 2900 * 4 = 11600 mins.

- This compares favourably with Spotify and Apple Music which run on a subscription model and have average monthly usages of 3540 mins and 3960 mins respectively.
- Thus for the same monthly price paid by the user for the other two streaming giants, the average BeatChain stream cost could be as high as HK$0.058. Out of this, the artist would earn an average of approximately HK$ 0.0464 which is nearly four times higher than Apple Music which pays an average of HK$ 0.0165 and even higher than Spotify which pays HK $0.007975 on average.

Table 3 provides a summary of the comparisons of the revenues and commissions in HK$ for every 1000 streams on the streaming platforms that have been discussed above.

	Spotify	**Apple Music**	**BeatChain**
Average pay	29	60	20
Publisher Commission	14.5	30	0
Average Record Label Commission	6.525	13.5	0
Total Average Commission	21.025	43.5	4
Average Artist income	7.975	16.5	16

Table 3. Revenues and commissions in HK$ per 1000 streams on different streaming platforms

8. Conclusion

In the digital age, streaming services have proven to be a double edged sword. On one hand, artists are provided with platform to expand their listener base but on the other, they lose a majority of their revenue to fees levied by various intermediaries. Furthermore, although many countries have laws protecting copyrights in place, navigating the digital landscape of infringement can be murky making it challenging to recover royalties.

In this paper, we have reviewed the applicability of blockchain and smart contracts in protecting these rights of the artist and have consequently developed BeatChain, a web streaming platform with the capabilities of decentralised file storage, a rights ledger and crypto wallets. This forms a direct bridge between the artist and listener and facilitates automatic, transparent, and rapid royalty collection through smart contracts while also significantly reducing the risk of copyright infringement through encrypted audio files. All the fundamental features of the application have been completed comprising of user login, wallet linkage through MetaMask, track uploading, streaming, audio encryption along with additional features such as searching, sorting, and filtering tracks as well as playlist creation.

Over the course of development, a few challenges and limitations were identified in terms of high data storage costs and gas fees on Ethereum, implementing encryption and searching and sorting functionalities in a decentralised setup. To combat this, we switched to Polygon, a layer 2 scaling solution over Ethereum and moved towards a semi-decentralised architecture which would provide the integrity of blockchain and while maintaining performance efficiency by virtue of the backend. Nonetheless, creating a robust DRM encryption mechanism that is unexploitable still remains a challenge for the streaming industry at large.

Moreover, we were also able to devise a pay per stream business model which suits the requirements of the platform as well as the interests of all parties involved. While all development over the course of the past year has been on track leading to the completion of the deliverable, next steps and future improvements include modelling the smart contract to simulate cases of a track having more than one copyright owner, adding a ratings system and comments section, using machine learning to provide music recommendations and extending the web application to a mobile and desktop application.

8. References

[1] “Charges for the use of intellectual property, receipts (BOP, current US$),” *The World Bank DataBank*. [Online]. Available: https://data.worldbank.org/indicator/BX.GSR.ROYL.CD?end=2020&most_recent_value_desc=true&start=1980&view=chart.

[2] J. Stone, “The state of the music industry in 2020,” Toptal Finance Blog, October 06, 2020. [Online]. Available: https://www.toptal.com/finance/market-research-analysts/state-of-music-industry.

[3] J. Dimont, “Issue 2 Article 5 2-2018 Royalty Inequity: Why Music Streaming Services Should Switch to a Per-Subscriber Model, 69 Hastings L,” *Hastings Law Journal*, vol. 69, 2018, [Online]. Available: https://repository.uchastings.edu/cgi/viewcontent.cgi?article=3808&context=hastings_law_journal.

[4] “Global music streaming revenue 2019,” *Statista*. https://www.statista.com/statistics/587216/music-streaming-revenue/.

[5] “Money from music survey data portal,” *Future of Music Coalition's Artists Revenue Streams Project*. [Online]. Available: http://arsdata.futureofmusic.org/dashboard/show?utf8=%E2%9C%93&role_composer=true&role_recording=true&role_salaried=true&role_performer=true&role_session=true&role_teacher=true&mgenre=ALL&ft=false&trained=false&careerexp=ALL&gender_male=true&gender_female=true&gender_transgender=true&gender_unanswered=true&emigroup=1.

[6] Masataka Goto and Takuichi Nishimura. Rwc music database: Music genre database and musical instrument sound database. In ISMIR, pages 229–230, 2003.

[7] [M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, “Blockchain,” *Bus. inf. syst. eng.*, vol. 59, no. 3, pp. 183–187, 2017.

[8] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, “Blockchain-enabled smart contracts: Architecture, applications, and future trends,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 11, pp. 2266–2277, 2019

[9] Bybit Learn, “Blockchain layer 1 vs. Layer 2: Things you must know,” *Bybit Learn*, 08-Oct-2021. [Online]. Available: https://learn.bybit.com/blockchain/blockchain-layer-1-vs-layer-2/.

[10] S. de la Rouviere, “Introducing ujo portal: Making musicians more money.,” *Medium*, 13-Dec-2018. [Online]. Available: https://blog.ujomusic.com/introducing-ujo-portal-making-musicians-more-money-9224d808a57a.

[11] SingularDTV, “Introducing the Breaker Royalty Management Platform,” *Medium*, 24-Jun-2021. [Online]. Available: https://singulardtv.medium.com/introducing-the-breaker-royalty-management-platform-60a819b80c1e

[12] “Music fans share ownership with artists in their favorite songs,” Vezt. [Online]. Available: https://vezt.co/

[13] H. Anwar. “Hyperledger vs Ethereum,” 101 Blockchains, April 04 2019. [Online]. Available: https://101blockchains.com/hyperledger-vs-ethereum-2/

[14] J. Hyatt. “Decoding crypto: The 10 most popular cryptocurrencies,” Nasdaq, August 06 2021. [Online]. Available: https://www.nasdaq.com/articles/decoding-crypto%3A-the-10-most-popular-cryptocurrencies-2021-08-05

[15] “Ethereum's internet of blockchains,” *Polygon*, 15-Sep-2021. [Online]. Available: https://polygon.technology/

[16] “Solidity - Solidity 0.8.9 documentation,” Solidity, September 29 2021. [Online]. Available: https://docs.soliditylang.org/en/v0.8.9/

[17] “The crypto wallet & gateway to web3 blockchain apps,” *MetaMask*. [Online]. Available: https://metamask.io/.

[18] [15] “Blockchain music rights in (about) 3 minutes”, YouTube, Feb 16 2018. [Video file]. Available: https://www.youtube.com/watch?v=1LUKgbWihGU&ab_channel=BruceBalensiefer

[19] “IPFS powers the distributed web,” *IPFS Powers the Distributed Web*. [Online]. Available: https://ipfs.io/

[20] “Ethereum API: Ipfs API & gateway: ETH Nodes as a service,” *Infura*. [Online]. Available: https://infura.io/

[21] K. L., “The blockchain scalability problem & the race for visa-like transaction speed,” *Medium*, 23-Jul-2019. [Online]. Available: https://towardsdatascience.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44. [Accessed: 18-Apr-2022].

[22] “Semi decentralised applications (Semi-DApp),” *Blockchain Patterns*, 01-Sep-2021. [Online]. Available: https://research.csiro.au/blockchainpatterns/general-patterns/deployment-patterns/semidapp/. [Accessed: 18-Apr-2022].

[23] “Secure your data with AES-256 encryption,” *How does AES-256 encryption work to protect your data*. [Online]. Available: https://www.atpinc.com/blog/what-is-aes-256-encryption. [Accessed: 18-Apr-2022].

[24] A. Tiwari, “This is how your favorite netflix movies and shows are pirated,” *Fossbytes*, 12-Jun-2021. [Online]. Available: https://fossbytes.com/how-pirate-netflix-amazon-prime-movies-shows-piracy/. [Accessed: 18-Apr-2022].