

# **Introduction to Blockchain and CryptoCurrency**



## **Project Progress Report**

Adwit Singh Kochar (2018276)

Anuneet Anand (2018022)

Pankil Kalra (2018061)

# About

Music has always been an essential part of our lives, and it is a way for many artists to express themselves. The music industry has grown exponentially in the past few years with the success of digital music streaming services. People can now access music from various parts of the world using their computers and smartphones.

However, the major music streaming platforms are centralized, and as a result, the artists lack control over their content and receive only a small fraction of the revenue generated by these platforms. The current system requires artists to depend on intermediaries like studio companies, record labels, and streaming service companies for managing and marketing their work. A technology that can improve transparency and provide more control to the artists over the money they make can revolutionize the industry. The direct interaction between consumers and artists would also eliminate the need for intermediaries from the revenue stream.

We propose **blockstudio** - a blockchain-based decentralized music streaming platform to connect music enthusiasts directly to independent music artists. The artists can use this platform to share their music with greater freedom while ensuring ownership and avoiding duplication of their music. People can listen to their favourite songs and support the artists by making micropayments through our custom crypto tokens. The platform can be hosted as a DApp on the Ethereum blockchain.

# Week - 1

We studied various technologies that are used in making DApps from various online resources like blogs, tutorials and videos. Since most of these technologies were new to us, we spent a good amount of time getting familiar with them. We also read about some concepts related to blockchain and cryptocurrency in general.

## Frontend

1. **React:** It is an open-source Javascript library used for frontend development. We use it because it is easy, fast and has a lot of support in the form of tutorials and forums. Many DApps we explored use React to develop their frontend and thus we follow the same.

## Backend

1. **Web3.js:** It is an extensive collection of libraries that will help our web app to communicate with an Ethereum blockchain using JSON based Remote Procedure Calls.
2. **IPFS (InterPlanetary File System):** It is a distributed storage system that utilizes Peer-to-Peer (P2P) networking. Unlike traditional “location-based” storage systems, IPFS is “content-based” as it maintains unique global hashes of all the files. This gives it many advantages like independence from a single controlling entity and preventing duplicate files, thus being perfect for implementing our DApp.

## **Blockchain**

1. **DApp:** A decentralized application uses a decentralized system to perform different tasks. DApps help the user interact with the smart contract programs stored on a blockchain.
2. **Solidity:** It is the high-level language used to write smart contracts that can be stored on the blockchain. This will define the logic behind our blockchain, dictating the behaviour of all the accounts.
3. **Ganache:** It is used to create a personal local Ethereum blockchain, whose dummy accounts can be accessed by a cryptocurrency wallet and used in a DApp. This helps us run our DApp like it is connected to the actual Ethereum network without long transaction times.
4. **MetaMask:** It is a crypto-wallet that can be accessed through a browser extension. We can use this to interact with our local blockchain created using Ganache.
5. **Truffle:** It is the development environment that will be used to develop smart contracts and DApps. It can be used for the compilation, testing as well as deployment of smart contracts.

## **Week - 2**

We explored projects on GitHub which have a similar use case as ours and obtained insights into the features which we should implement and also technologies that we can use. Short descriptions of some of them:

### **Resound**

- A project by Julien Tregoeat from the USA.
- It is a decentralized music marketplace for independent artists and labels.
- It uses Ethereum Virtual Machine (EVM) for transaction logic and as a database for primitive data types.
- IPFS (InterPlanetary File System) is used as a location for blob storage.
- Limitations: No check for piracy, Weak user authentication, Conversion errors due to direct use of ETH for transactions.

### **musicDAPP**

- A project by Raj Chakravarthy from India.
- It is a decentralized concert ticketing system.
- It has different user roles like Organisers who can create and deploy concerts using smart contracts and Users who can buy tickets to the deployed concerts.
- Images of the concert are stored on IPFS.
- Blockchain properties help to ensure that each user can buy only one ticket to each concert and no user can use some other user's ticket.
- The developer has tried to achieve minimum usage of Truffle to prevent code breaks.
- The developer has achieved variable-based routing using Next.js

### **jelly-beats**

- Music player built on blockchain.
- It is now archived and developers have moved on to [Hound.fm](https://hound.fm).
- Does not have a lot of documentation.
- It is a desktop app built using Node.js and React with Electron on top.
- The most interesting part was that it used LBRY (LBRY is an open-source protocol providing distribution, discovery, and purchase of digital content via a decentralized network).
- The app currently does not work when installed through code (hound.fm works though). It's most probably due to the current version of LBRY not being supported.
- Code is comparatively more complex than other projects that have been explored.

# Week - 3

We brainstormed and listed potential features of the DApp which we aim to implement.

## **1. Two types of users:**

### ○ **Artists**

- Can post songs and track activity on their songs.
- Can assign fees for listening to songs after the free-plays have been exhausted.
- Can also choose to keep the song free for all.

### ○ **Audience**

- Can browse artists and songs.
- Can listen to songs upto a free limit.
- Can get access to unlimited plays of a song by paying the fees as defined by the artist.

**2. P2P Storage System:** All data will be stored on a distributed database like IPFS.

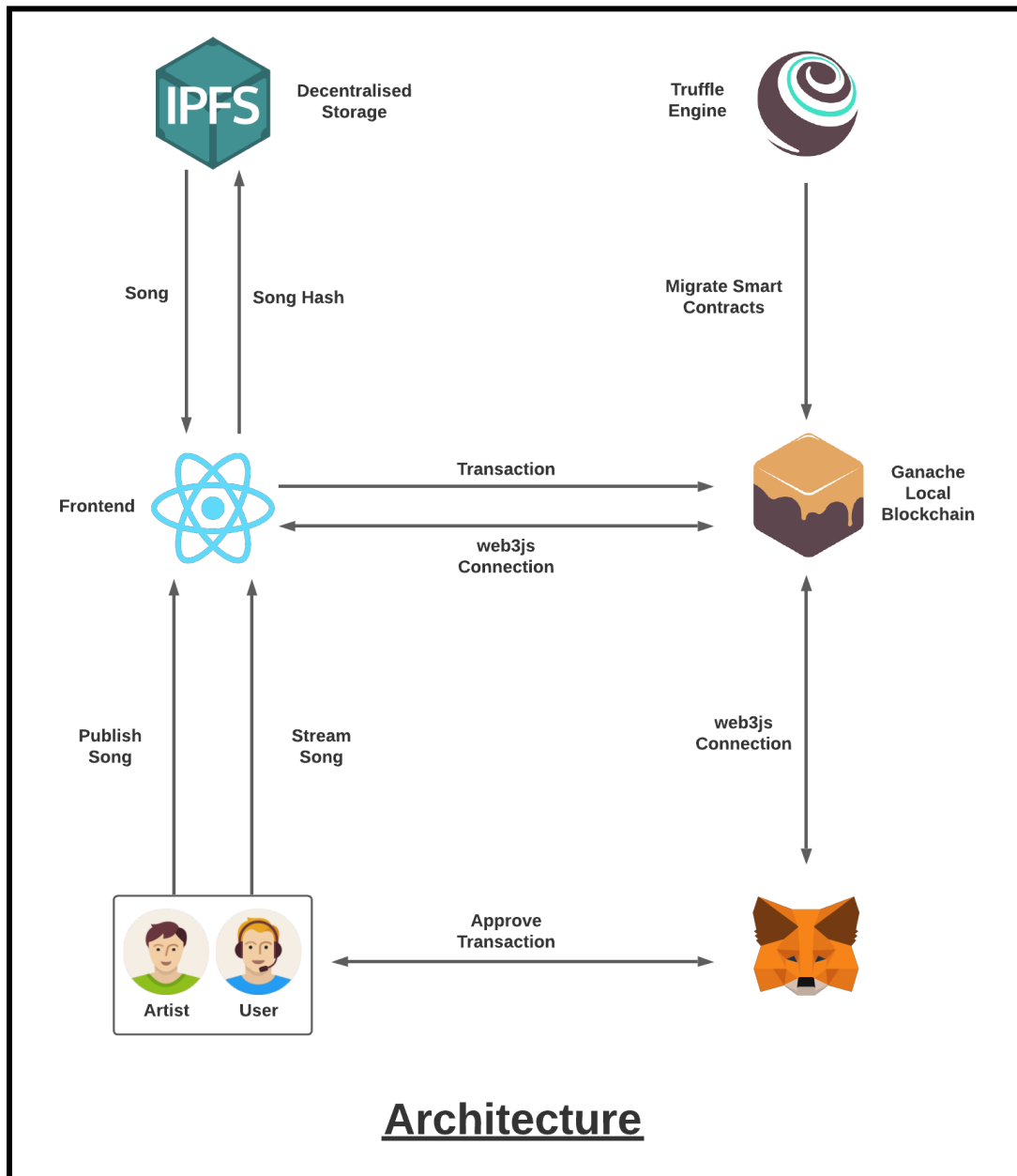
**3. Cryptocurrency:** An in-app crypto coin and wallet for in-app purchases will help prevent the transaction of ETH in decimal points. The wallet can be refilled using ETH funds.

**4. CrowdFunding:** Support favourite artists by funding their albums and tracks.

**5. Survey System:** Give anonymous reviews for songs, artists and albums and vote for the best songs.

## Week - 4

We designed a tentative architecture of our DApp and set up the code repository with the starter code. The repository can be found [here](#).





# Work Distribution

- **Week 1:** All of us explored the basics of the technologies mentioned.
- **Week 2:** Anuneet explored the resound project, Adwit explored the musicDAPP project and Pankil explored the Jelly-Beats project.
- **Week 3:** All of us brainstormed together to come up with potential features. Adwit proposed we should use a custom token for our system to prevent the transaction of Eth in decimal values. Anuneet proposed we can add a crowdfunding feature where users can support their artists. Pankil proposed that we should also display some stats about songs and give users the ability to rate songs.
- **Week 4:** All of us collectively worked on the architecture diagram and setting up the repository.

## References

- <https://reactjs.org>
- <https://web3js.readthedocs.io/en/v1.5.2/>
- <https://ipfs.io>
- <https://docs.soliditylang.org/en/v0.8.9/>
- <https://www.trufflesuite.com>
- <https://metamask.io>
- <https://ethereum.org/en/dapps/>
- <https://www.youtube.com/watch?v=nUOuqk0jLdw>