BLOCKCHAIN - AY 2023/2024

# Project Report

Emanuele Civini [247198]     Federico Nardelli [248334]     Jacopo Manenti [247279]

# Contents

# 1  Mission

Our mission is to empower small record companies by providing a decentralized platform that enables them to quickly monetize their emerging artists' music rights by selling them to fans and investors. We aim to create a fair and transparent marketplace leveraging blockchain technology to ensure record companies gain liquidity while offering fans and investors unique and valuable NFTs (album rights) linked to their favorite artist.

# 2  Market Overview

## 2.1  The role of Record Labels

Record labels play a crucial role in the music industry. They are responsible for discovering, signing, and promoting artists, as well as managing the production, distribution, and marketing of their music. In exchange for these services, record labels take a percentage of the revenue generated by the artists they represent.

Record labels also provide artists with access to resources and connections that can help propel their careers to new heights. Additionally, they often assist with negotiating contracts and licensing deals, ensuring that artists are fairly compensated for their work.

The music industry has undergone significant changes over the years. In the past, record labels relied heavily on physical album sales, but with the advent of digital technology, the landscape has shifted. Online platforms and streaming services have become the primary means of consuming music, presenting both challenges and opportunities for record labels.

## 2.2  Revenues Channels

With the rise of these channels, record labels now have new avenues to generate revenue. Below is a list of potential revenue channels (not exhaustive):

- **Album Sales and Digital Downloads**: Physical album sales have declined but still contribute to revenue, along with digital downloads.

- **Streaming Services and Royalties**: Streaming services like Spotify, Apple Music, and Tidal dominate music consumption, with record labels earning revenue through royalties. However, per-stream payouts are much lower than album sales, creating challenges in generating substantial income.

- **Merchandising and Licensing**: Record labels maximize revenue through merchandising and licensing deals, selling artist merchandise and licensing music for commercials, movies, and TV shows. These strategies However, to effectively implement this strategy, the artist needs to already be popular, making these operations more profitable.

## 2.3   The struggle

### 2.3.1   For labels

[1]

Despite the many available solutions, today's record labels are grappling with significant financial challenges, particularly the struggle to generate stable revenue. They face rising production and promotion costs while contending with shrinking margins from music sales.

This pressure is further exacerbated by the dominance of a few industry giants, making it difficult for smaller labels to penetrate the market and retain key artists. These major players often lure away talent with their superior resources. While there are occasional stories of artists being unexpectedly "discovered" by a small label and skyrocketing to fame, such cases are rare exceptions.

### 2.3.2   For artists

For an artist today, connecting with a major record company is often crucial for reaching a broad audience. However, the contractual terms offered by these big labels are frequently unfavorable, placing artists in difficult situations. In contrast, smaller labels typically offer better and fairer conditions, although they lack the extensive reach and resources of the major players. This dynamic creates an unfair barrier to entry for genuine artists, as the monopolistic practices of the larger labels undermine a free and fair market, making the already challenging path to success even more difficult.

## 2.4   The opportunity

In a landscape dominated by a few major labels, where small record companies struggle to compete and artists often face unfavorable conditions, we have identified a niche opportunity to introduce a new form of financing.

The concept is straightforward: a small record company can sell a portion of its royalty package linked to an album (for example, 10% of Album X by Artist Y), which includes an assigned percentage of future earnings from an emerging artist they choose to invest in. This approach allows the label to convert a non-liquid asset (emerging artist's royalties) into immediate financial resources, which can be reinvested into other projects or marketing efforts. On the other side, fans and investors interested in speculative opportunities can purchase these royalties, giving them the potential to benefit from the artist's future success. This creates a win-win situation, providing financial flexibility for the label and a unique investment opportunity for supporters and speculators.

To provide this service, we developed an NFT marketplace where album royalty rights are tokenized. By tokenizing just the royalty rights, the platform is able to streamline and manage the distribution of royalty streams accurately without employing a third party that could take a significant percentage of this revenue stream. The token holders receive regular payments in proportion to their share of the royalty rights.

Our platform is designed to serve several key audiences:

- Small record companies seeking to monetize their music rights.

- Fans and collectors looking for unique and potentially valuable music NFTs.

- Investors aiming to speculate on the future success of artists.

### 2.4.1 Royalty digression

[2] Traditionally, the royalty rates negotiated in artist contracts directly impact how much revenue record labels make. These rates can vary depending on the artist's popularity, the label's bargaining power, and market conditions. Furthermore, royalty rates can be structured in various ways, such as a percentage of gross revenue, net revenue, or a combination of both. Record labels aim to strike a balance between providing artists with fair compensation and ensuring their own profitability.

This explanation outlines how the music industry relies on royalties as a primary form of payment for musicians. Here's a breakdown of the key concepts:

1. **Royalties**: These are payments made to the owner of an asset (like an album) for the right to use/own that asset. In the music industry, royalties are generated from the licensing of copyrighted songs/albums and recordings. Royalties are paid out before other financial obligations, such as stockholder dividends, and are typically distributed at regular intervals (e.g., monthly or quarterly).

2. **Types of Royalties**: There are different kinds of royalties depending on how the music is used:

   - **Reproduction Royalties**: Earned from sales or streaming of the music.
   - **Performance Royalties**: Earned whenever the music is played publicly (e.g., on the radio, in bars, or during live performances).
   - **Synchronization (Synch) Royalties**: Earned when music is licensed for use in TV shows, films, commercials, or video games. This is typically a one-time payment.

3. **Stakeholders in Royalties**: Several parties may have a claim to royalties:

   - **Sound Recording**: Bands typically sign contracts with labels that own the Sound Recording Copyright and distribute royalties to band members, producers, and session musicians.
   - **Composition**: Songwriters often sign deals with publishers who manage the copyright and collect royalties. These royalties are usually split 50/50 between the songwriter and the publisher, with multiple songwriters potentially involved, each receiving a share.

Typically, both songwrites and artists assign their rights to a third party for management, instead of attempting to track a song's use and seek payment independently. Song copyrights are typically assigned to a record label.

# 3 Technology

Our platform is built on the Polygon network to ensure low transaction fees and fast processing times. Key technologies include:

- **Blockchain and Smart Contracts:** We employ three distinct smart contracts:

  - **Factory Contract:** Oversees the deployment of NFTs that represent record companies and manages the associated addresses (e.g., record company managers) that are authorized to establish and operate these entities.
  - **NFT Contract:** Automates the minting process for each artist and album, as well as the management of the generated royalties that will be distributed to the rightful holders. The latter one happens periodically off-chain.

- **Marketplace Contract:** Facilitates the ownership management of NFTs, including buying, selling, and bidding transactions on the platform.

- **Web3 Integration:** MetaMask is integrated to enable seamless user interaction with the blockchain, ensuring secure and straightforward access to the platform's features.

- **Frameworks:** Our development stack includes:

  - **Foundry:** A comprehensive smart contract development toolchain that handles dependencies, compiles projects, runs tests, deploys contracts, and allows command-line and Solidity script-based interaction with the blockchain.

  - **Next.js, TypeScript, Shadcn:** Utilized for building the user interface, ensuring a responsive, scalable, and modern frontend experience.

  - **Ethers.js:** A JavaScript library that facilitates interaction with the Ethereum blockchain and its ecosystem, making it easier to build and deploy decentralized applications.

- **Pinata Integration:** We use Pinata for storing and managing files on the InterPlanetary File System (IPFS), a decentralized file storage protocol. NFT metadata and images are stored on Pinata, with the resulting CID (Content Identifier) retrieved and stored on the blockchain using the following URL format: `"https://gateway.pinata.cloud/ipfs/${cid}"`. IPFS ensures that files are distributed across a decentralized network of nodes, providing robust security and ensuring that data stored in the network cannot be altered, even in the event of a hacking attempt. This guarantees the immutability of NFT data. [3]
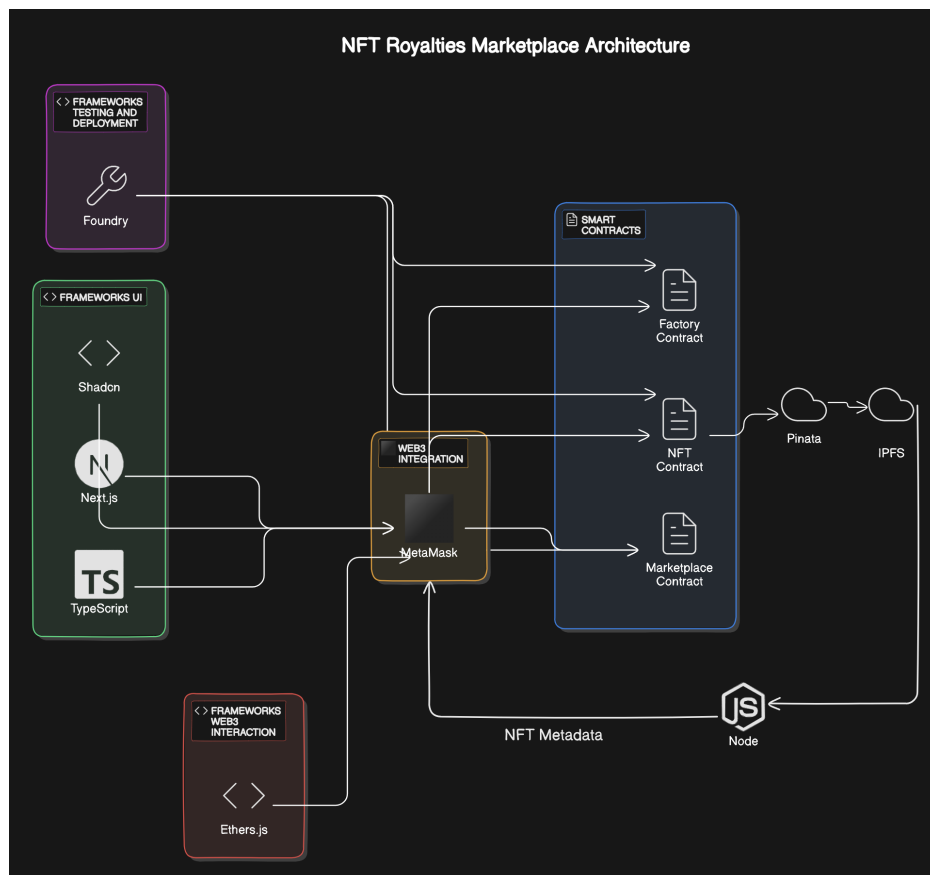


Figure 3.1: Diagram showing an overview of the applictio

# 4  Why, When, How

**Why:**  Small record companies often struggle to gain liquidity and fair compensation. Our platform provides a solution by offering a decentralized marketplace that rewards both record companies and fans/investors. The latter will be able to access to royalty streams that were previously available only to industry insiders, private equity, or institutional funds.

**When:**  The project will be rolled out in phases, starting with the beta launch of the platform to onboard initial users and record companies. Full launch is planned after thorough testing and feedback incorporation.

**How:**

- Record companies will pay a small fee to publish their music rights on the platform (i.e., to be whitelisted).

- Upon publishing, the record company will provide their wallet address. The admin will whitelist this wallet to grant access to the necessary functionalities.

- After an album is minted, the specified number of music rights will be tokenized (e.g., 1000 tokens).

- When an NFT is sold, 5% of the sale price will be paid as royalties to the record company, and 1.5% will go to the platform to cover operational costs.

- NFT holders will benefit from potential appreciation in value if the artist gains popularity. Additionally, NFTs may offer other privileges such as access to exclusive content, presales, discounted concert tickets, and special merchandise.

# 5  Operation

The platform operates as follows:

- A small record company submits a request to join the platform and is whitelisted by the admin through a dedicated section of the web interface accessible only by the admin ("whitelist" section).

- Upon approval, the record company's wallet is registered in the system, unlocking a section of the web interface where new artists can be added and their albums minted ("mint" section).

- The platform supports login via MetaMask, utilizing the Polygon testnet network Amoy (Ethereum Layer 2) for transactions.

- If the wallet belongs to a registered record company, the relevant functionalities for managing collections and minting albums become accessible.

- If the wallet belongs to a regular user, they can browse various collections, apply filters, view album details, and purchase or bid on albums as desired.

- Both record companies and regular users can buy or auction owned or created albums.

# 6  Value Proposition Canvas

By using the value proposition canvas, we go on to analyze the value proposition, the customer (small record) and the client profile (Retail investor).
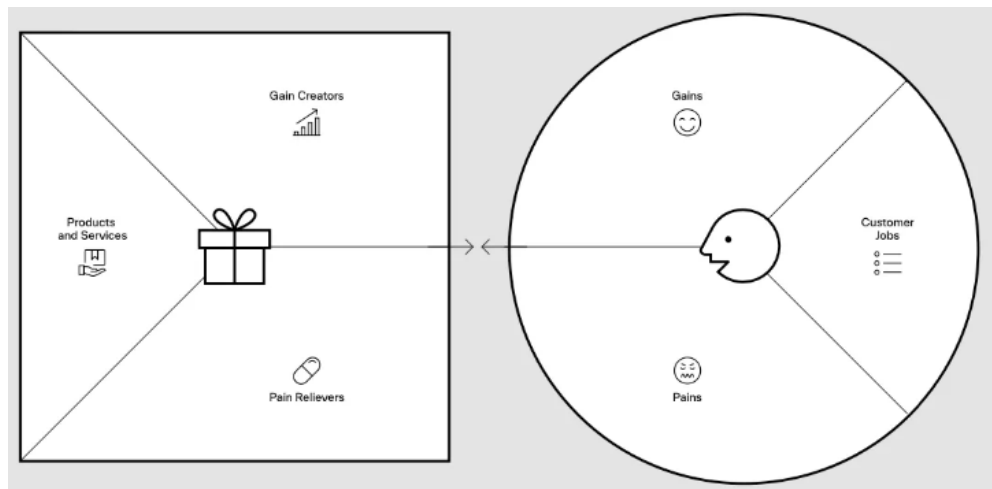


Figure 6.1: Value proposition canvas

## 6.1  Customer Profile

### 6.1.1  Customer Jobs

1. **Small records:**

   Small record labels are heavily involved in artist discovery and development, where they actively scout and sign emerging talent. Once an artist is signed, the label provides resources and guidance to help develop their music and brand. They also manage music production and distribution, overseeing the recording process and ensuring the music is available on both physical and digital platforms.

   In marketing and promotion, label owners create campaigns to increase the artist's visibility, utilizing social media, media appearances, and live performances. They handle the monetization of music rights by managing royalties, exploring licensing deals, and creating merchandise opportunities. Additionally, they are responsible for financial management, ensuring budgets are adhered to and payments are made on time.

2. **Retail investor:**

   Retail investors, on the other hand, focus on buying stock in record companies or publishers by researching and investing in publicly traded music companies. They may also engage in investing in royalty funds, purchasing shares in funds that manage music rights and distribute royalties.

### 6.1.2 Gains

1. **Small records:**

   For small record labels, one of the primary gains is revenue generation through music sales. By effectively producing and distributing music, labels can reach a broad audience and generate income from both physical and digital sales. Additionally, brand visibility and artist promotion are crucial gains. Strategic marketing efforts not only boost the visibility of their artists but also lead to higher sales, increased streaming numbers, and stronger fan engagement.

2. **Retail investor:**

   Retail investors, on the other hand, benefit from earning passive income from royalties. By investing in royalty funds or directly purchasing music rights, investors can receive a steady stream of income through regular royalty payments. Another potential gain for investors is the possibility of high returns. Although there is risk involved, investing in the right artist or music catalog can lead to substantial profits, particularly if the artist becomes popular or the value of the music rights appreciates over time.

### 6.1.3 Pains

1. **Small record:**

   Small record labels often face significant challenges, starting with limited financial resources. These constraints make it difficult to invest in the talent development, marketing, and production needed to compete with larger, more established labels. Additionally, high operational costs associated with producing, promoting, and distributing music can strain the financial stability of small labels, making it hard to sustain operations over the long term.

   Another major pain point is challenges in artist retention. Without the financial muscle to offer resource to push the artist's popularity, small labels often struggle to keep successful artists, who may be tempted by the better deals offered by larger labels. Furthermore, limited market reach is a persistent issue, as small labels typically lack access to extensive distribution networks and marketing channels, which limits their ability to reach wider audiences and expand their brand.

2. **Retail investor:**

   Limited direct investment options can be frustrating for those who want a more personal connection to the music they support, as buying stock in record companies or publishers only offers indirect exposure to the success of specific artists. Moreover, access restrictions for royalty funds often pose a barrier, as many of these funds are private and only available to accredited investors, excluding many retail investors who don't meet the necessary criteria. Finally, high costs and barriers to entry can be prohibitive for retail

   investors interested in purchasing music rights directly. The price of acquiring rights to songs, albums, or entire catalogs—especially high-profile ones—can be exorbitant, making it difficult for most retail investors to participate in this potentially lucrative market.

## 6.2 Value Proposition

### 6.2.1 Gain Creators

1. **Small record:**

   For small record labels, the value proposition includes several key benefits that address their specific needs. Liquidity is a major advantage, allowing labels to swiftly monetize

their investments in emerging artists by selling or auctioning music rights. This provides immediate cash flow, which is crucial for sustaining operations and funding new projects. Additionally, alternative financing options are made available, offering quick access to funds that help stabilize revenue streams and enable reinvestment into other ventures.

Another significant benefit is increased autonomy through decentralization. By eliminating the need for intermediaries, small record labels can retain more control over their operations and ensure they receive fair compensation for their artists' work. This autonomy is complemented by enhanced market reach, which allows record companies to access a global audience. This expanded reach increases the visibility of their artists and opens up new revenue opportunities on a larger scale.

2. **Retail investor:**

For retail investors, the value proposition includes access to a broad range of diverse investment opportunities. Investors can choose from different genres, artists, and types of rights, allowing them to tailor their portfolios according to their preferences and risk tolerance. Transparency and trust are also key components, with blockchain technology ensuring that all transactions are transparent, which helps build confidence among investors. Portfolio diversification is another important gain for retail investors. By

investing in the music industry, they can diversify their portfolios with assets that are uncorrelated with traditional markets, potentially reducing their overall investment risk. Additionally, these investments are structured to provide passive income generation through regular royalty payments, allowing investors to earn steady income from their holdings and enhance their overall financial returns.

### 6.2.2 Pain Relievers

1. **Small record:**

For small record labels, the value proposition offers several solutions to common challenges. Risk mitigation is achieved by diversifying revenue streams through the sale or licensing of music rights, reducing financial risk and providing more stable income. This is complemented by access to alternative and cheaper financing, where selling or auctioning music rights helps alleviate financial pressure and reduces reliance on traditional loans or credit lines.

Simplified rights management through decentralized platforms and blockchain technology lowers transaction fees, streamlines the management and distribution of music rights and royalties, and reduces administrative burdens. This approach eliminates intermediaries, allowing labels to retain more revenue. Additionally, enhanced artist retention is possible by offering flexible monetization strategies and fair compensation, helping to keep artists from leaving for larger labels.

2. **Retail investor:**

For retail investors, lower barriers to entry are provided by enabling fractional ownership of music rights or smaller-scale investments, making it easier to participate in the music industry without large capital requirements. This is supported by a simplified investment process through a user-friendly platform with clear information and streamlined procedures, making it easier for investors to navigate the complexities of music rights investment.

### 6.2.3 Products and Services

- **Tokenized Music Royalties:** Unique digital assets that represent fractional ownership of music rights, allowing both retail investors and record companies to easily buy, sell, and trade music royalties on the blockchain.

- **Record Company Profiles:** Customizable and interactive profiles that enable record companies to showcase their artists, music catalogs, and upcoming releases, while directly connecting with fans and potential investors.

- **Decentralized Marketplace Platform:** A user-friendly, decentralized platform built on the Polygon network, designed for the secure buying, selling, and trading of music NFTs and royalty rights, with low transaction fees and high transparency.

- **Automated Royalty Management:** Smart contract-powered automation of royalty calculation, distributions, buy and sell, ensuring accurate and timely payments to rights holders, without the need for intermediaries.

- **Investment Analytics and Insights:** Tools and resources that provide investors with detailed analytics, market trends, and insights into the potential performance of their music rights investments, helping them make informed decisions.

# 7 Our Solution

## 7.1 Project Environment Setup

To set up the project environment, you can clone the GitHub repository using the following link: `https://github.com/jacopomanenti01/Blockchain`. Alternatively, you can access it through the public link: `https://blockchain-one-bice.vercel.app/`.

Follow these steps to set up the environment:

1. Clone the repository:

   ```
   git clone https://github.com/jacopomanenti01/Blockchain
   ```

2. Navigate to the project directory:

   ```
   cd Blockchain
   ```

3. Follow the detailed instructions provided in the project README file.

   Once finished, the environment is set up.
   To run the server, use the following command:

```
npm run dev
```

It will start locally on: `http://localhost:3000`

## 7.2 Platform Features

After setting up and running the server, you can view the front-end landing page. Users can log in with MetaMask on the Polygon Amoy testnet. Depending on the role assigned to you, the navbar will display different functionalities. If you are a valid music label and your address has already been whitelisted by us as an admin, you can access the "Whitelist" section at the top of the page. Here, you can deploy NFTs for record label companies with the following details:

- **Record Name**: The name of the record label company.

- **Record Address**: The address associated to interact by adding artists and albums.

- **Record Treasury**: The address of the treasury wallet (likely a cold wallet) of the record label company.

- **Initial Fee**: The percentage of future minted royalties that will be allocated to the future owner.

Only one NFT can be deployed per record address to represent the record label company. The data entered in the form are directly used as parameters for the deployNFT function of the NFTfactory contract (see `./contracts/src/NFTfactory.sol`).

Once the address has been deployed as an NFT music label, if the user logs in with that address, they will see the "Mint" section in the navbar. This indicates that the NFT has been successfully deployed and they have granted access to minting artists and albums.

Before minting albums, it is mandatory to add the artist owned by the record label company. The artist has the following attributes:

- **Artist's Name**

- **Description**

- **Genre**

- **imageUrl**[1]

As done previously, the data are passed directly to the createSinger function in the NFT smart contract (see `./contracts/NFT.sol`).

Once the transaction is confirmed, a valid singer is created, and the album can be minted. To mint an album, the following parameters must be entered in the form:

- **Image**

- **Title**

- **Year**

- **Songs**

- **Singer Associated**

- **Royalties**[2]

The data is first formatted into a JSON file, which is then uploaded using the IPFS system with Pinata. If the transaction is successful, the data is saved, and the corresponding CID is retrieved; otherwise, the file is deleted, and the minting process is stopped.

Once the data has been correctly uploaded, the createAlbum function is called, and the following three parameters are passed:

- **shareCount**: Corresponds to the total amount of the album's royalties available on the website.

- **singerId**: Corresponds to the ID of the chosen singer. This number increments whenever a new singer is created, and the contract keeps track of it using the Singers mapping.

- **Metadata**: A link pointing to the location where the JSON data is stored. The link is constructed as follows: `https://gateway.pinata.cloud/ipfs/${cid}`. The CID parameter refers to the unique ID of the uploaded JSON file in Pinata, previously retrieved.

We chose to store metadata in the album instead of full information because it is more cost-effective in terms of gas fees. This approach makes the entire uploading process faster and more scalable, allowing for additional information to be stored in the future at the same cost.

Once an album is minted, the owner can choose to sell it or auction it through their user page in the "Owned" section. If the NFT has not yet been sold or listed for auction, the contract will request permission to list the token on the marketplace. After permission is granted, the

---

[1]unused for the moment

[2]This represents how many rights the record label wants to register on the website. It is not mandatory to sell all of them.

smart contract will execute the appropriate function to either sell or auction the album [3]. Once the transaction is confirmed, the album will be visible to everyone in the marketplace, accessible through the "Featured Section" of the website.

At this point, the album can be purchased or bid on by anyone, or the owner can cancel the listing if they wish. Token management and royalty distribution are handled automatically by the smart contract.

All transactions can be seen on the block explorer of the Amoy Polygon testnet: `https://amoy.polygonscan.com/`
.

## 7.3   Smart Contracts

Our platform utilizes smart contracts to manage the creation, sale, and royalties of music NFTs. Below is an overview of the three main smart contracts used in our system: NFTFactory, NFT, and Marketplace.

### 7.3.1   NFTFactory.sol

The primary purpose of NFTFactory contract is to allow an admin to create new NFT contracts, associate these contracts with specific record company admins, and facilitate querying of NFTs associated with a particular wallet.
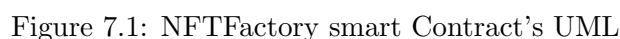


Figure 7.1: NFTFactory smart Contract's UML

**State variables:**

The contract includes several key state variables for managing NFT deployments. The **isFactoryDeployed** mapping tracks whether a specific NFT contract was deployed by this factory, using the NFT contract's address as the key and returning a boolean value. The **associatedNFT** mapping links an admin's address to their associated NFT contract, ensuring that each admin can only manage one NFT. The **deployedNFTs** mapping stores the addresses of all NFT contracts deployed by the factory, indexed by a unique numeric ID, making it easy to track and retrieve these contracts. Lastly, the **nextNFTId** counter increments with each new deployment, providing a unique ID to each NFT contract, which ensures distinct identification within the system.

**The constructor:**

grants the DEFAULT_ADMIN_ROLE to the contract deployer. This role is necessary to perform admin-level actions like deploying new NFTs.

**Functions:**

The contract includes several important functions for managing NFT deployments and associations.

The **deployNFT** function is responsible for deploying a new NFT contract for a record company. It takes the name of the NFT, the record company's admin address, the treasury address, and an initial fee as parameters. This function ensures that each admin can only have one associated NFT. Upon successful deployment, it updates the relevant mappings and emits a 'NewNFT' event to signal the creation of the new NFT.

The **setAssociatedNFT** function links an admin's address to a specific NFT contract. This function can only be called by an NFT contract that was deployed by the factory, ensuring that the association is legitimate. It checks that the admin doesn't already have an associated NFT unless the association is being reset to address(0), which allows for safe management of admin-to-NFT relationships.

The **batchGetNFTs** function retrieves a batch of NFTs owned by a specific wallet. It accepts the wallet address, a starting index, an ending index, and a limit on the number of NFTs to retrieve. The function returns arrays containing NFT addresses, token IDs, amounts owned, URIs, balances, and the actual number of NFTs found, enabling users to efficiently query and manage their NFTs.

## 7.3.2 NFT.sol

The NFT contract represents the NFTs managed by a record company, including functionality for creating singers and albums, and updating record company fees.
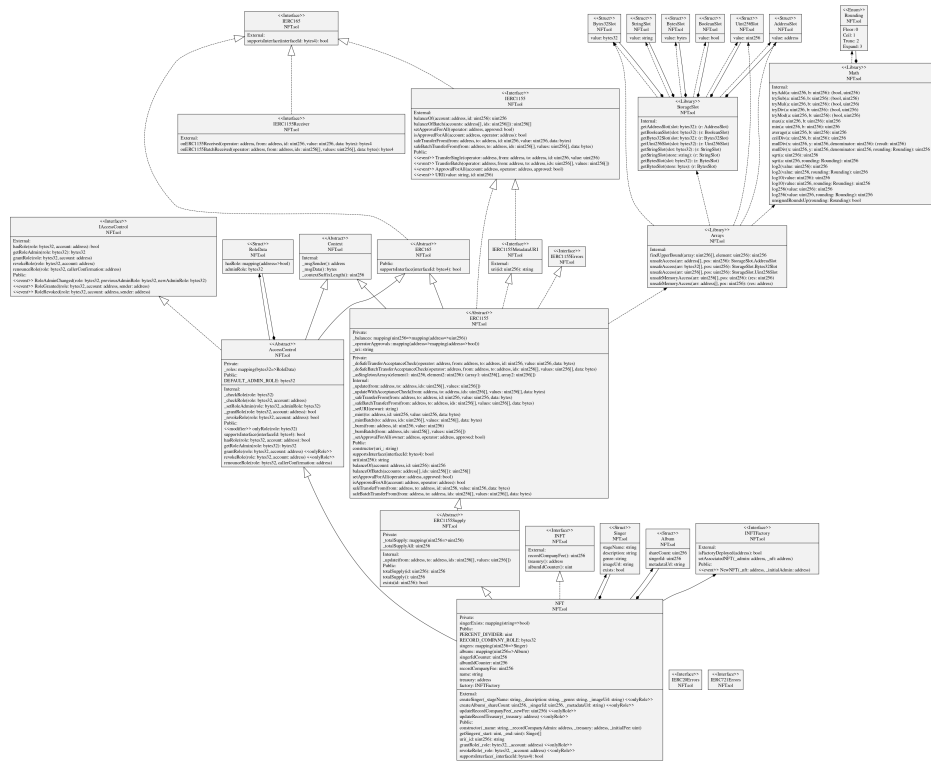
Figure 7.2: NFT smart contract's UML

### State variables:

The state variables in the contract are central to its functionality, defining both the roles and the data structures it manages. The **PERCENT_DIVIDER** is a constant used to represent percentage values with six decimal places of precision, ensuring accurate calculations for fees and shares. The contract also defines two key structures: **Singer** and **Album**. The 'Singer' struct stores details about an artist, including stage name, description, genre, image URL, and a flag indicating whether the singer exists. The 'Album' struct similarly stores data for albums, including the number of shares, the associated singer's ID, and a metadata URL.

To manage these entities, the contract uses mappings. The **singers** mapping links a unique singer ID to a 'Singer' struct, while the **albums** mapping links a unique album ID to an 'Album' struct. A private mapping, **singerExists**, prevents duplicate singers by tracking whether a singer with a specific stage name has already been created. The counters **singerIdCounter** and **albumIdCounter** are used to assign unique IDs to each new singer and album, respectively.

The contract also defines roles and permissions, with the **RECORD_COMPANY_ROLE** being crucial for tasks like creating singers and albums. The **name** variable holds the name of the NFT, typically representing the record label or company. Financial management is handled through the **treasury** address, where funds like fees or royalties are collected, and this address can be updated by the record company.

Finally, the **factory** variable references the 'INFTFactory' contract that deployed this NFT contract. This connection allows the NFT contract to interact with the factory for tasks such as role management, maintaining a close relationship between the NFT and its creator.

### The constructor:

Initializes the NFT contract with the name of the label, the record company admin's address, the treasury address, and an initial fee. The constructor grants the DEFAULT_ADMIN_ROLE to the deploying entity and assigns the RECORD_COMPANY_ROLE to the specified record company admin. The factory contract that deployed this NFT contract is recorded, and role management

is set up so that only the record company can manage its accounts.

**Functions:**

The contract includes several functions that manage the creation and handling of singers, albums, roles, and associated data.

The **createSinger** function allows the creation of a new singer profile by providing a stage name, description, genre, and image URL. This function is restricted to users with the 'RECORD_COMPANY_ROLE' and ensures that no duplicate singers are created by checking if the singer already exists. Upon successful creation, it increments the 'singerIdCounter'.

Similarly, the **createAlbum** function is responsible for minting a new album as an NFT. It requires the number of shares, the ID of the associated singer, and a metadata URL. Like the 'createSinger' function, it is accessible only to users with the 'RECORD_COMPANY_ROLE' and ensures that the specified singer exists before proceeding.

For retrieving information, the **getSingers** function returns an array of singers within a specified range of indices. It validates the range to ensure that '_end' is greater than or equal to '_start' and within the bounds of existing singers, providing a method to view singers created within the contract.

The **updateRecordCompanyFee** function allows the update of the fee charged by the record company. This function is restricted to users with the 'DEFAULT_ADMIN_ROLE' and ensures that the new fee does not exceed 100%, as enforced by the 'PERCENT_DIVIDER'.

To manage financial flows, the **updateRecordTreasury** function updates the address of the record company's treasury. This function is accessible only to users with the 'RECORD_COMPANY_ROLE', ensuring that only authorized users can make such changes.

The **uri** function returns the metadata URL associated with a specific token ID (representing an album). Before returning the URL, it checks that the token exists, ensuring that only valid tokens are queried.

Role management is handled by the **grantRole** and **revokeRole** functions. The 'grantRole' function assigns a specific role to an account, and if the role is the 'RECORD_COMPANY_ROLE', it also associates the account with the current NFT contract. This function is restricted to users with the admin role for the specified role. Conversely, the 'revokeRole' function revokes a specific role from an account, and if the role is the 'RECORD_COMPANY_ROLE', it also removes the association of the account from the current NFT contract, ensuring that role and access management remains secure and well-regulated.

### 7.3.3 Marketplace.sol

The Marketplace contract facilitates the buying and selling of NFTs, including auctions and fixed-price sales. It handles payments, collects fees, and transfers NFTs.
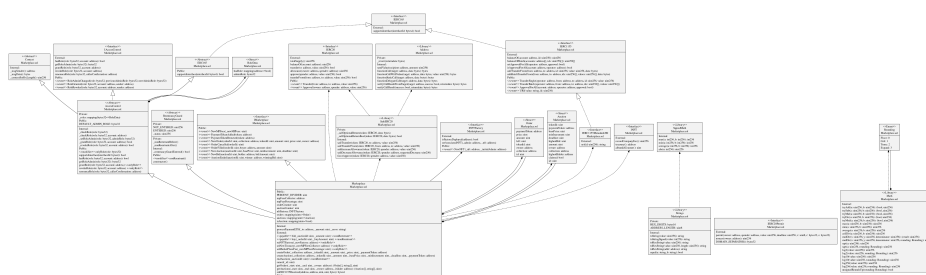


Figure 7.3: Marketplace smart contract's UML

**State variables:**

The Marketplace contract defines several state variables that are critical to its operation. The **PERCENT_DIVIDER** is a constant used to represent percentage values with six decimal places of precision, which is essential for accurate fee calculations across the platform. The **mpFeesCollector** is the address designated to collect marketplace fees, ensuring that a portion of each transaction is appropriately allocated. The **mpFeesPercentage** variable stores the percentage of each transaction that is taken as a marketplace fee, scaled by $10^6$ to allow for fine-tuned control over fee settings.

The contract uses **orderCounter** and **auctionCounter** as counters to keep track of the total number of orders and auctions created on the platform. Each order and auction is assigned a unique ID based on these counters, which increment with every new listing. The **nftFactory** variable holds a reference to the INFTFactory contract, which the marketplace uses to verify that NFTs listed for sale or auction were created by an approved factory.

Mappings play a crucial role in this contract. The **orders** mapping stores all the details of each sale order, identified by its unique ID, while the **auctions** mapping holds details of each auction. The **isAuction** mapping tracks whether a particular order ID corresponds to an auction, helping to differentiate between regular sale orders and auctions.

### The constructor:
The constructor of the Marketplace contract initializes several key settings. It takes in the address of the NFT factory, the address where marketplace fees should be collected, and the initial marketplace fee percentage. These inputs are crucial for setting up the core functionalities of the marketplace. Additionally, the constructor grants the deployer of the contract the DEFAULT_ADMIN_ROLE, allowing them to manage roles and other critical administrative tasks, ensuring the contract can be properly governed.

### Functions:
The Marketplace contract includes a variety of functions to manage NFT sales and auctions. The **setNFTFactory** function allows the marketplace to update the address of the NFT factory, ensuring that only NFTs from a valid and recognized factory can be listed. The **setNewTreasury** function enables the marketplace to change the address where marketplace fees are collected, maintaining flexibility in financial management.

To manage marketplace fees, the **setMarketPlaceFee** function allows administrators to update the fee percentage, with a built-in safeguard to prevent the fee from exceeding 100%. This function ensures that the marketplace can adjust its fee structure while staying within reasonable limits.

For creating listings, the **createOrder** function allows users to create a new sell order for an NFT, transferring the specified amount of tokens from the seller to the marketplace and recording the order details. The _paymentToken parameter in this function can be set to address(0) if the seller wishes to accept the native cryptocurrency (such as ETH) as payment, rather than an ERC20 token. The **createAuction** function works similarly but is tailored for auctions, allowing sellers to specify a base price, minimum bid increment, and deadline for the auction. In this function, the _paymentToken can also be set to address(0) to indicate that bids should be made in the native cryptocurrency.

The **bid** function enables users to place bids on active auctions, processing payments and ensuring that bids meet the auction's conditions. If the auction is using the native cryptocurrency as indicated by **paymentToken == address(0)**, the bid amount is sent directly with the transaction (msg.value). If a bid is successful, the previous highest bidder is refunded, and the new bid becomes the highest. The endAuction function concludes an auction once its deadline has passed, transferring the NFT to the highest bidder if the auction was successful, or returning it to the seller if it was not.

For direct purchases, the **buy** function allows users to buy tokens from a sell order, processing the payment, deducting fees, and transferring the purchased tokens to the buyer. If the payment method is in native cryptocurrency (paymentToken == address(0)), the buyer sends the payment with the transaction. If a seller wishes to cancel an active order, the **cancel** function allows them to do so, returning the tokens to their control.

To help users query marketplace listings, the **getOrders** function returns a list of orders within a specified range of indices, optionally filtered by owner. The _owner parameter can be set to address(0) to return all orders regardless of ownership. Similarly, the **getAuctions** function returns a list of auctions, with optional filters for the owner and highest bidder. The _owner and _bidder parameters can be set to address(0) to ignore these filters, providing flexibility in searching for active listings.

Finally, the **processPaymentETH** function is an internal utility that processes payments in ETH, ensuring successful transactions, while the **onERC1155Received** function enables the contract to receive ERC1155 tokens, making it compliant with the ERC1155 standard and allowing it to interact with NFTs seamlessly.
.

## 7.4   Compilation and Testing

To compile and build the ABIs of the contracts, navigate to the 'contracts' folder and execute:

```
cd contracts/
forge build
```

This will generate an 'out' directory containing the ABIs for the contracts.

Several tests are written to test the coverage of the smart contracts. To run the tests, navigate to the contracts directory and execute:

```
forge test
```

# 8   Roadmap

Our development roadmap outlines the key milestones and future implementations planned for our platform:

**Phase 1: Beta Launch**

- Develop and test the smart contract for NFT creation and royalty distribution.

- Launch a beta version of the platform to onboard initial record companies and users.

- Collect feedback to improve user experience and functionality.

**Phase 2: Full Launch**

- Implement additional features such as record company profiles and fan/investor interaction tools.

- Expand marketing efforts to attract more record companies and users.

- Establish partnerships with music blogs and independent radio stations.

**Phase 3: Future Implementations**

- Introduce new types of NFTs, such as limited edition releases and exclusive content.

- Develop a mobile application for easier access and engagement.

- Explore integration with other blockchain networks to enhance scalability and reach.

- Offer presale and discounted concert tickets to NFT holders.

- Provide special signed merchandise and other exclusive items for fans and investors.

# Bibliography

[1] labelgrid, "Challenges faced by modern record labels," *labelgrid*, 2023,
   https://labelgrid.com/blog/content-distribution/
   challenges-faced-by-modern-record-labels-navigating-the-digital-age/.

[2] royaltyexchange, "Music royalties 101 – intro to music royalties," *royaltyexchange*, 2020,
   https://www.royaltyexchange.com/blog/music-royalties-101-intro-to-royalties#sthash.
   WJgzrAxe.dpbs.

[3] B. Metaversal, "Where has your nft metadata gone?" *hackernoon*, 2022,
   https://hackernoon.com/where-has-your-nft-metadata-gone.