Resonance: ERC-1155 Tokens for Music Artists

Sailesh Bechar
Computer Engineering
University of Waterloo
Waterloo, Canada
sailesh.bechar@uwaterloo.ca

Abstract. Every day, up-and-coming music artists around the world struggle with the financial obstacle of turning their passion into a career. Faced with the decision of signing to a record label (along with most of the music's rights and royalties) or staying independent and building a passionate following through their own means, most feel there is no other option than to choose the former. However, as the technology of the internet continues to evolve, the latter path offers ever-increasing potential to new creators.

In this paper, we present Resonance, a Smart Contract that enables rising artists to leverage fungible and non-fungible tokens to engage with their fans on a level traditionally unattainable. Our approach combines Chainlink's Verifiable Random Function (VRF) with the ERC-1155 standard for Smart Contracts to prove the scarcity of each token minted. In addition, each non-fungible token is given a popularity score that favors small artists who are yet to be discovered by the mainstream. This score is calculated on-chain using a FIFO data structure and captures the rate at which new tokens are minted. This method is efficient in both time and space complexity, minimizing gas fees and as a result, overhead costs for the artists' fans.

Keywords: Smart Contract, NFT, ERC-1155, Popularity, Chainlink

CCS Concepts: Software design engineering, Secure online transactions

1. Introduction

The popularization of fungible tokens created with ERC-20 [1] and non-fungible tokens created with ERC-721 [2] has open the gates for creators and fans to gain previously untapped value. Artists with a larger following such as Kings of Leon [3], Nas [4] and The Weeknd [5] are releasing their latest music projects as NFTs, paving the way for artists with a smaller following to follow in their footsteps. These growing, up-and-coming music artists struggle to make a living in the face of financial obstacles posed by music labels and streaming services. The most popular streaming platform Spotify, pays \$0.00274 per stream [6]. From this, music labels will additionally take over 80\% of the royalties [7]. Using Chainlink's Decentralized Oracle Network along side the ERC-1155 token standard within Smart Contracts, this project will help music artists create meaningful relationships with their fans and allow them to move closer to financial independence.

Music artists are consistently met with financial obstacles when trying to grow and build their fan base. From the record labels that trap young artists with egregious loans, known as ``advances", on top of taking 80-90\% of all album sale and performance royalties [8]. To the streaming services that only give back pennies to the artist for a song with hundreds of plays [6]. What both of these obstacles have in common is the centralized big pocket acting as the middleman, muddying the relationship the artist has with their audience. The artists' priority becomes turning a profit as opposed to creating their best artwork. It becomes increasingly more difficult for the artist to achieve this when they are constantly distracted with the stress of, ``How I financially make my dreams a reality?" More often then not, they will sacrifice the quality of their work to conform to the inequitable formula laid out by record labels and streaming platforms, who bear the fiduciary responsibility to maximize shareholder value [9].

This paper proposes leveraging ERC-1155 tokens to enable small artists to gain financial independence, while building a strong relationship with their fan base. The ERC-1155 Multi-token standard is used for creating both fungible and non-fungible tokens in one Smart Contract.

The fungible aspect of this contract will be a currency for the artist, as every token is identical. There will be a fixed amount initially created, and then distributed to their fans, as seen fit by the artist. When an artist creates a piece of music, being a song or an album, they want to create as many identical licences to use the artwork as possible. This suitably translates to a fungible token. However, to create additional value the artist can add a unique element to their artwork for every token, making them non-fungible. By introducing scarcity through Chainlink's Verifiable Random Function (VRF), the music's artwork can be dynamically chosen with a given probability. Additionally, the artist's current popularity score can be computed on-chain and added to each token. This score is calculated using a FIFO data structure which captures the notion of `frequency of tokens minted` using the block number difference of each token from the token minted before it. Value is provided to both parties, by adding a dimension of collecting and trading to the artist's music. Furthermore, imprinting an on-chain, immutable score of the artist's popularity can serve as cryptographic proof for how long super-fans have been supporting their favorite artist and the role they may have played in their successes.

1.1 Contributions

1. Design and implement a smart contract to allow artists to create fungible tokens representing a currency and non-fungible tokens of their artwork with an assigned scarcity and popularity score
2. Implement a Python script for uploading the tokens' metadata to IPFS, compatible with OpenSea's marketplace
3. Experimentally evaluate the project using Brownie's testing suite which includes an integration test of the complete token generation workflow and a unit test of the correctness of popularity scores between artists

2. Background

Artists are currently using the ERC-721 standard to create non-fungible tokens (NFTs) within Smart Contracts, for the Ethereum Virtual Machine (EVM). Throughout 2020 and 2021, this method has found tremendous success for digital artists, allowing fans to own their artwork on the Blockchain and to also gain utility from such ownership. Prior to NFTs, the ERC-20 standard within Smart Contracts has been widely used to create fungible tokens. These tokens can provide a variety of utility such as functioning as a currency, asset, lottery ticket or in-game reputation points [1].

Blockchains can use decentralized oracles to interface with the outside world. Chainlink is a standard that enables Smart Contracts to access many external data sources through their Decentralized Oracle Network. Decentralized oracle networks (DONs) enable the creation of hybrid smart contracts, where on-chain code and off-chain infrastructure are combined to support decentralized applications (dApps) that react to real-world events and interact with traditional existing systems. This eliminates the vulnerability of a single-point of failure seen in a centralized oracle and maintains the security of the decentralized Blockchain. Chainlink's VRF ensures randomness is preserved when nodes are reaching consensus to push a new block, proving the scarcity of an asset [10].

3. Implementation

```
uint256 avg_delta = delta_sum / delta_queue.length();
return (15000000 / avg_delta) + (
        item_counter * 1000000 / avg_delta
    );
```

Figure 1. A Code Snippet of The Popularity Calculation.

Small, up-and-coming music artists are constantly struggling to make ends meet financially. Rather than taking a loan from a record label and signing their soul away, many artists are now going independent with the age of the internet. The adoption of fungible and non-fungible digital assets stored on the Blockchain is gaining popularity as artists are finding new ways to engage their fans. This project gives super-fans the opportunity to collect a unique and rare version of their favorite artist's work, that contains a receipt of the artist's popularity at the time of acquisition.

3.1 Design/ Solution

The solution involves integrating both the ERC-1155 Multi-Token standard and Chainlink Verifiable Random Function (VRF) into a single Solidity smart contract. When a user is ready to mint a non-fungible token (NFT), a transaction occurs to the VRF's Subscription Manager, to generate 2 random words. These random words are used to denote the scarcity of each NFT minted and to determine if a token of fungible currency will additionally be transferred to the minter. The next component of the minting process is generating a popularity score for the artist, at the time of NFT creation. Instead of gathering the popularity data of an artist from a centralized data source such as Spotify, this project aims to preserve the decentralized nature of the various Blockchain technologies and calculate a score representing their popularity on-chain, leveraging a FIFO data structure. To conserve gas, this FIFO data structure is implemented as a mapping datatype. Each element in the queue stores the difference in the block number of the most recently minted NFT from the block number of the second most recently minted NFT, up to a size of 50. The block number delta is computed by the difference of the current block number from the previous one in the queue.

The intuition behind the formula in Figure 2.1 is to capture the rate at which tokens are minted. This can be approximated by computing the average block number delta of the whole queue. It is implemented by storing a moving sum of the block number deltas in the queue, updating every time a new block number delta is enqueued. To capture this notion of token mint rate, the formula involves keeping the average block number delta of the whole queue, in the denominator. This quantity is essentially `1/time = frequency`, meaning the time between token mints is inversely proportional to the token's popularity score. As floating point precision is unsupported by the current latest version (v0.8.0) of the Solidity language, the token mint rate is multiplied by a factor of 15,000,000. There is an additional factor to the equation which includes the total number of tokens minted. This factor is included to differentiate small artists, from larger artists. It is this paper's assumption that the lower the popularity score, the more valuable the token will have, giving incentive to users to support artists who are less well known. This factor is then scaled by a factor of 1,000,000 to remove the need for floating point precision, and to have an appropriate proportion to the token mint rate.

4. Experiments

```
link_before = interface.LinkTokenInterface(
config["networks"][network.show_active()]["link_token"])
    .balanceOf(deployer, {"from": deployer})
tx_cancel = resonance.cancelSubscription(
        deployer, {"from":deployer}
    )
tx_cancel.wait(1)

link_after = interface.LinkTokenInterface(
config["networks"][network.show_active()]["link_token"])
    .balanceOf(deployer, {"from": deployer})

assert link_after > link_before
```

Figure 2. Integration Test Clean Up Assertions.

The Python framework Brownie was used to deploy the smart contract, execute all on-chain functions and test using Brownie's built-in integration of PyTest. The first function was an integration test for the full end-to-end creation of an NFT. The second function was testing the correct relative popularity scores of three different artists' NFTs, where each artist has their own distinct level of popularity. Instructions to run the experiments can be found at https://saileshbechar.github.io/Resonance/.

4.1 Integration Test

The preparation of the first experiment begins with deploying the smart contract. Görli was the chain chosen for the testing on the Ethereum Blockchain because it is the only testnet compatible with Chainlink's VRF. When initializing, a new subscription to Chainlink's VRF is created, returning an `id` for subsequent interaction. In addition, 1000 of fungible tokens representing the artist's currency is minted to the contract. Next, the contract is funded with 2 of Chainlink's ERC-20 token, LINK. The contract then transfers the LINK to the VRF Subscription Manager. This manager is responsible for all random word generation and controls permissions for can generate and consume random words.

The first experiment begins by generating 2 random words. The words are of type `uint256` and are stored on-chain. Then, the `mint_nf_artwork()` transaction is called in the smart contract. The NFT representing the artist's music is subsequently minted and given a scarcity based on the specified distribution. The scarcity is emitted as an event to compare in the next portion of the test. In addition, one token is transferred to the contract caller, with a probability of 50%. Finally, the subscription to the VRF Subscription Manager is cancelled, with the remaining LINK transferred to the contract deployer's address.

To test the accuracy of the integration test, the two generated random words are asserted against the same logic as executed in the contract. As the first random word determines the scarcity, the test also applies the same operations to verify an equivalent generated scarcity. The test uses a uniform distribution to distinguish between the three different possible values more easily. To test if the artist's fungible currency was transferred to the caller of the mint transaction, the second generated random word tested against the same logic as executed in the contract. Built into the ERC-1155 standard `balanceOf()` conveniently returns the balance of any `itemID` associated with any address. In addition, the clean up functionality is tested so that once the subscription is cancelled. The deployer's address should have more LINK than before the clean-up was initiated.

4.2 Popularity Score Test

```
Artist 1 Popularity: [0, 0, 2250000, 2375000, 2500000]
Artist 2 Popularity: [0, 0, 1500000, 1583333, 1666666]
Artist 3 Popularity: [0, 0, 900000, 950000, 1000000]

Artist 1 Popularity: [0, 0, 2250000, 1727272, 2000000]
Artist 2 Popularity: [0, 0, 1500000, 1583333, 1666666]
Artist 3 Popularity: [0, 0, 900000, 826086, 909090]
```

Figure 3. Example of Popularity Scores Between Artists.

The second experiment involves an identical set up process to the integration test, except repeated for three contracts, representing three artists. Each contract is first deployed and then cleaned up before deploying the next contract. When cleaning up, the unused LINK from the previous Subscription Manager is retrieved to use for the next contract's Subscription Manager. When an NFT is minted, its popularity score is emitted as an event. Each contract mints 5 NFTs and stores the popularity scores in a list in the test. Each subsequent contract has a longer duration between each token mint. The expected result is the longer the period is between each mint transaction, the lower that NFTs popularity score will be of the artist.

The popularity scores of each token are then compared across contracts. As the delay between mint transactions is constant for each of the artists, it is expected that each element `i` of `artist_1` should be greater than the popularity score of element `i` of `artist_2`, which should then be greater than the popularity score of element `i` of `artist_3`. In the case where there are less than 2 prior mint transactions, each of the artists will share the popularity score of 0. An example output of the popularity scores is given in Figure 3.3. We can see the descending order of the popularity as the time between mint transactions increases.

5. Limitations

This project currently does not wish for the token to become regulated by the SEC, thus the fungible currency should not pass the Howey test [12]. However, if the token were to be officially registered as a security, a fungible token acting as a security for the artist may potentially provide tremendous value.

6. Conclusions

Music artists often struggle financially to grow their careers when they are still building a loyal and adamant following. Instead of signing a one-sided deal with a record label, many artists are choosing to remain independent and leverage technologies such as smart contracts. This paper proposed using the ERC-1155 Multi-token standard for Smart Contracts, in addition to Chainlink's Verifiable Random Function to create added dimensions to their artwork. Albums can now have different levels of scarcity and be imprinted with the current popularity of the artist, at the time of creation. The decentralized nature of these technologies keep these qualities of the artwork free from the tampering of any centralized authority. The popularity score is calculated on-chain and uses efficient data structures to minimize gas fees.

Going forward, this project would be benefited to be deployed on a low-gas fee blockchain such as Polygon. When an artist is small, their fans do not want to spend hundreds of dollars to support them, especially when the bulk of that cost is sent to the network miners, executing the transactions on-chain.

References

[1] V. Buterin, F. Vogelsteller, Eip-20: Token standard, 2015. URL: https://eips.ethereum.org/EIPS/eip-20.

[2] D. Shirley, W. Entriken, Eip-721: Non-fungible token standard, 2018. URL: https://eips.ethereum.org/EIPS/eip-721.

[3] S. Hissong, Kings of leon will be the first band to release an album as an nft, 2021. URL: https://www.rollingstone.com/pro/news/kings-of-leon-when-you-see-yourself-album-nft-crypto-1135192/.

[4] A. Hennis, Earn lifetime royalties from grammy-award-winning artist nas, 2022. URL: https://marketrealist.com/p/how-to-buy-nas-nft/.

[5] R. Perper, The weeknd's limited-edition nft collection raises over $2 million usd, 2022. URL: https://hypebeast.com/2021/4/the-weeknd-nft-collection-song-artwork-2-million-nifty-gateway-auction.

[6] M. Kristiansen, How much does spotify pay artists per stream, 2021. URL: https://homestudioideas.com/how-much-does-spotify-pay-artists/.

[7] P. Schwartz, Ponterio; Levenson, 3 reasons why musical artists are getting underpaid, 2021. URL:https://www.splaw.us/blog/2021/11/3-reasons-why-musical-artists-are-getting-underpaid/.

[8] K. Cornell, The artist; record label relationship – a look at the standard "record deal", 2017. URL:https://www.tunecore.com/blog/2017/05/artist-record-label-relationship-look-standard-record-deal-part-1.html.

[9] D. Andersson, How record labels screw artists, 2020. URL: https://elevenbstudios.com/how-record-labels-screw-artists/.

[10] Chainlink, What is an oracle in blockchain? " explained: Chainlink, 2021. URL: https://chain.link/education/blockchain-oracles.

[11] N. Reiff, Decentralized autonomous organization (dao), 2022. URL: https://www.investopedia.com/tech/what-dao/.

[12] N. Reiff, Howey test, 2021. URL: https://www.investopedia.com/terms/h/howey-test.asp.