

MusicChain :

A Music Cover Directed Acyclic Graph Profit Sharing System based on Ethereum Smart Contract

作者：陳映樵，葉曜德，施貽仁

指導教授/實驗負責教授：林宗男教授

摘要：

我們實作了一個網站，讓音樂創作者可以上傳原創或是二創的音樂，並且在上傳的同時，會利用音樂辨識演算法，將新上傳的音樂和網站中的音樂做比對，而形成音樂之間 Cover 的關係。這個 Cover 的關係我們使用 Direct Acyclic Graph 紀錄在 Ethereum 上。此外，我們更進一步使用這樣的關係去做音樂版權分潤機制，並以智能合約的方式實作之。

一、動機

在音樂產業裡，版權、分潤、曝光度是非常重要的要素，也是支持整個產業的核心。而在 youtube 或是其他的影音平台常見到眾多音樂創作者去 cover 自己喜歡的音樂，並且不同人的 cover 各有自己的特色。有些人甚至會自己將原著做些微的改編，像是將抒情歌改成爵士曲風，或是用別的語言來翻唱一首歌。有些甚至是將一些不同的歌接在一起，來回穿插。這麼多有創意的作品卻沒有一個機制將他們串連，也沒有平台對於 cover 這件事情做保障，是非常可惜的。因此我們希望可以開發一個平台，將 cover 音樂的辨識技術和著作權保障這個要素結合。

二、相關背景回顧

在著作權中，音樂產業相關規定為最複雜的部分 [1]。因為著作權無需登記，在作品產生的當下即受到著作權法之保護，此外音樂的製作過程通常都涉及多人參與，因此相關群權利關係，相當複雜。再者，著作權法對於音樂的公

開播送，公開傳輸，公開演出，重製權等等都有不同的規範。加上，音樂相關創作具有諸多型態，像是詞曲創作，錄音製作等等，都使得整體規則相當複雜。對於音樂創作者或是一般聽眾是一項享受音樂過程中無意義的負擔。因此，隨這數位科技的進步，目前已有許多應用將區塊鏈技術的透明、安全、不可篡改、可追蹤、去中心化等特性應用於音樂版權管理甚至是購買分潤。比較知名的幾個平台像是「Ujo Music」[2]，給予創作者自行管理授權，和使用智能合約來收權利金。而 KKBOX 也推出「Soundscape 在田發行」音樂平台及「Muzeum 區塊鏈創意產業協定」[3]。前者方便創作者管理各大音樂串流平台上的收益，後者制定了平台商和音樂創作者之間的權利和分潤關係，並以智能合約實行。以國外來說，則有 MUSICOIN、MEDIACHAIN、UJO 等網站，他們都讓權利分配更透明，更直接（避免多方剝削音樂創作人權益）。即使音樂在區塊鏈上的應用可以說是雨後春筍，片地開花，然而卻沒有人將 cover 這項的思維整合在分潤和區塊鏈技術上。

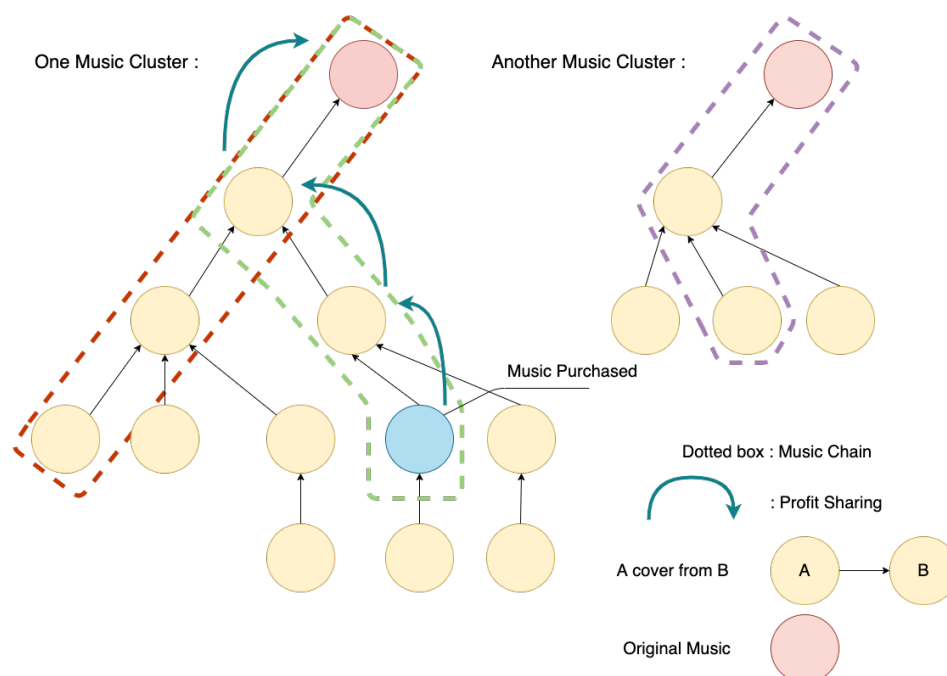
音樂相似度辨識主要有兩種技術：Audio Fingerprinting 和 Cover Song Identification。前者的主要用於辨識一模一樣的音樂，因為它對於到音調或是節拍的變化很敏感。而後者比較是翻唱辨識，主要的概念是去抽取音樂相關的特徵（像是和聲，調性等）[4]，而不是使用原始聲音訊號的特徵。有了這些特徵後，再進行距離的運算，而常見的距離計算方式有 Cross-correlation、Dynamic Time Warping、Similiarity Matrix 等。近年來，隨著計算資源的增加，也有許多方法是使用深度學習的方式去抽取特徵 [5,6]。

三、簡介

我們的網站主要是將 cover 音樂的辨識技術和著作權保障這個兩個要素結合。在 cover 音樂辨識的部分，我們利用辨識的結果，將音樂建立成 Direct Acyclic Graph (DAG) 並紀錄在區塊鏈上。而利用這樣的關係，我們提升 cover 音樂的可見度，同時也提供我們分潤的資訊。

在著作權的部分，我們使用 Ethereum 來實作。利用區塊鏈的不可竄改

性來保障創作者的著作權。同時利用 Ethereum 上的 Smart Contract [7]來進行音樂購買的交易。此外，我們提出了一個新的分潤方式，就是依照音樂 cover 的相似度，先後來進行分潤。以這樣的分潤方式，可以鼓勵音樂原創者去創作一首受歡迎的音樂，因為當有其他的音樂創作者來 cover 該首歌時，原創者也會因為 cover 音樂的收益而獲利，同時有增加曝光度。而對於二次創作的音樂者而言，其作品也可以再被別人拿去改編，因此同樣也受到鼓勵。而這些分潤機制都實作在 Smart Contract 裡，所以是公開，公平，透明且不能竄改[8]。

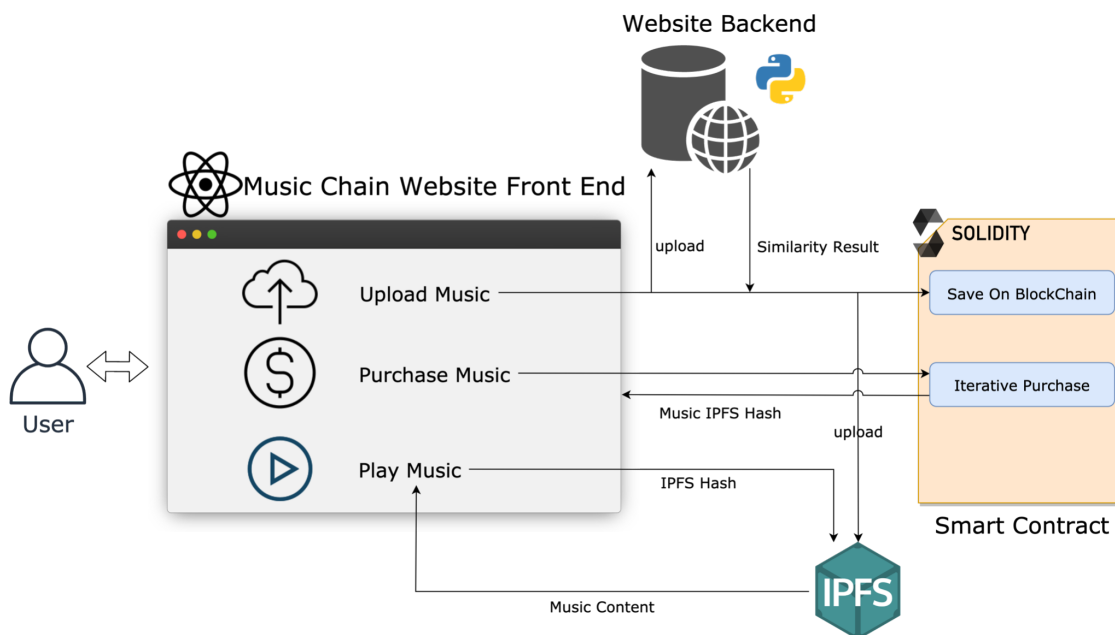


圖表 1 音樂 Cover 關係 DAG

圖表 1 中總共有兩首音樂在系統上被翻唱。而首音樂都可以向上 traverse 到 root node 形成一條 Music Chain，其中 edge 的建立都是根據音樂辨識的結果，和上傳的先後順序建立。分潤的部分會由所購買的音樂向上 traverse 做 profit sharing。

而在網頁設計的概念上，我們特別在音樂的詳細頁面加上音樂的 cover 關係鏈。使用者可以在聆聽一首歌的同時，知道其 cover 關係，增加整條 chain 上音樂的曝光度。

四、架構圖



圖表 2 網站架構圖

圖表 2 為網站架構。網頁主要三大功能：上傳音樂，購買音樂，播放音樂。上傳時，會將 mp3 檔存在 IPFS 上，並會根據後端傳來的辨識結果，寫相對應的資訊到區塊鏈上。購買時，前端會去呼叫 Smart Contract 去分潤，並將所購買歌曲的 IPFS Hash 回傳給前端。最後，播放音樂就是購買後，所得的 Hash 去和 IPFS 要檔案即可。而完整的程式碼也放在 Github¹上了。

五、使用技術

在我們的系統裡需要存取音樂檔案，因此我們使用 IPFS。IPFS 是一個分散式檔案系統，具有以下幾個優點

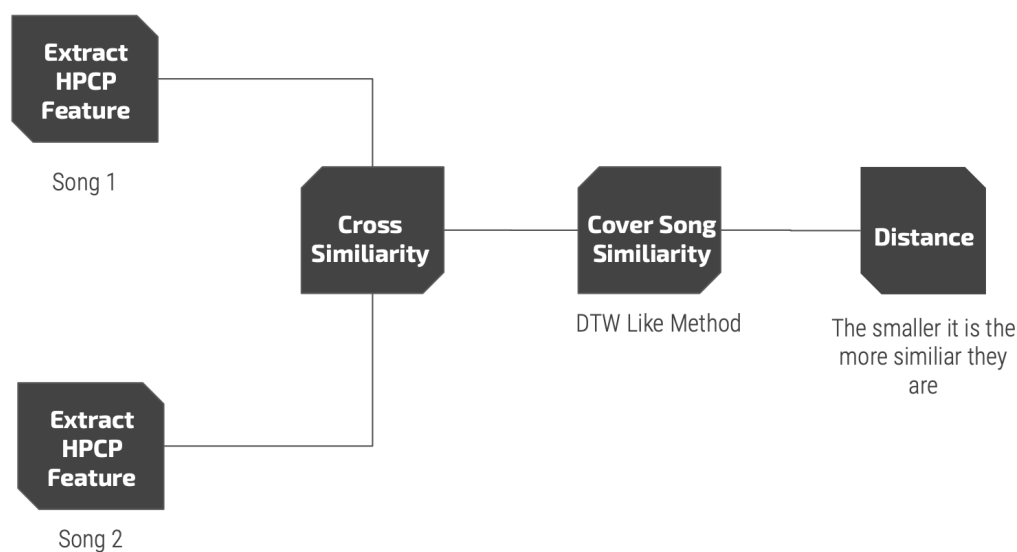
- 內容定址 (Content Addressing)
 - 唯一性
 - 無重複性
- 去中心化特性
 - 直接向 peer 節點要資料，增加下載速度
 - 公開透明

¹ <https://github.com/atosystem/nmlab-final>

- 能夠抵抗政府或組織監控

在我們的 project 中，我們用 IPFS 當作音樂的儲存系統，因為若要把音樂儲存在區塊鏈上的話，會花費太多的 gas，而且甚至沒辦法在一個 block 裡面就做完，因此也無法完成交易。

音樂辨識技術的部分，我們使用 Essentia [9] 套件。Essentia 是由 Pompeu Fabra University 的 Music Technology Group 所設計的音樂信號處理分析套件。裡面包含諸多常見功能，像是音樂分類，噪音移除，頻譜分析等等。在這個 project 裡，我們使用裡面的 Cover Song Detection [10]。根據文獻回顧，和我們的硬體計算能力考量，我們使用決定抽取聲音訊號的 (Harmonic Pitch Class Profiles) HPCP 特徵來協助辨識。如下圖。



圖表 3 Cover Song Identification 原理圖

原理如下：先將兩個原始音訊檔抽 HPCP feature，再計算他們的 cross similiarity matrix，並使用 Cover Song Similiarity 演算法跑在 cross similiarity matrix 上，最後得到一個分數，也就是距離。其中，Cover Song Similiarity 的計算方式是一種 Dynamic Time Warping 的演算法，常被用來比較兩段資料

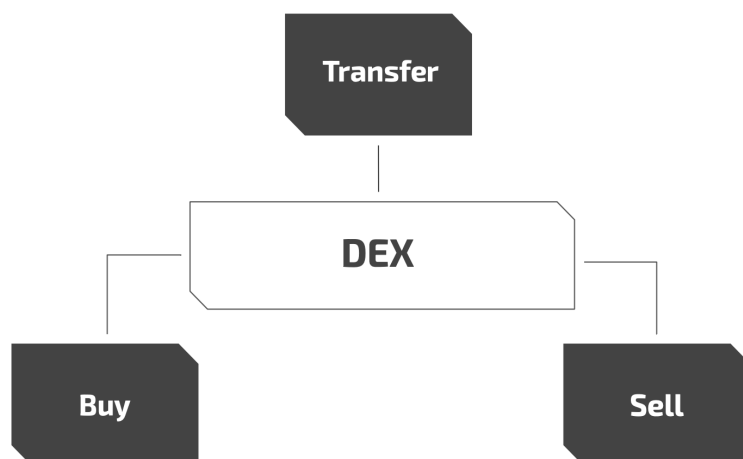
在允許多種 alignment 下的差距。在這個比較的部分，我們可以決定 HPCP 頻譜的解析度，和 sliding window 的大小及每次移動的步伐。這些 hyperparameter 都會影響辨識的準確性和辨識速度。在這兩者之間，我們透過幾次的實驗，將幾個大眾的音樂類型去跑結果，最後決定以 $\text{frame_window_size} = 9$ 、 $\text{frame_window_stride} = 3$ ，來設定。並且以距離 = 0.1 做為 cover 關係的 threshold（第 5 點會詳細說明）。

另外，我們網頁前端的部分是使用 React 框架。React 是 Facebook 團隊所開發的套件，它可以方便開發者去實作動態式更新的網頁。而網頁的後端辨識系統，我們使用 python 來處理，並且使用 EventSource [11] 的通訊協定和前端網頁進行進度更新回報。

而由於我們的 project 設定較為繁複，且架構複雜，因此我們使用 Docker 工具，將後端辨識系統和 IPFS 客戶端包在 Docker 容器內，方便在不同機器上快速部署。

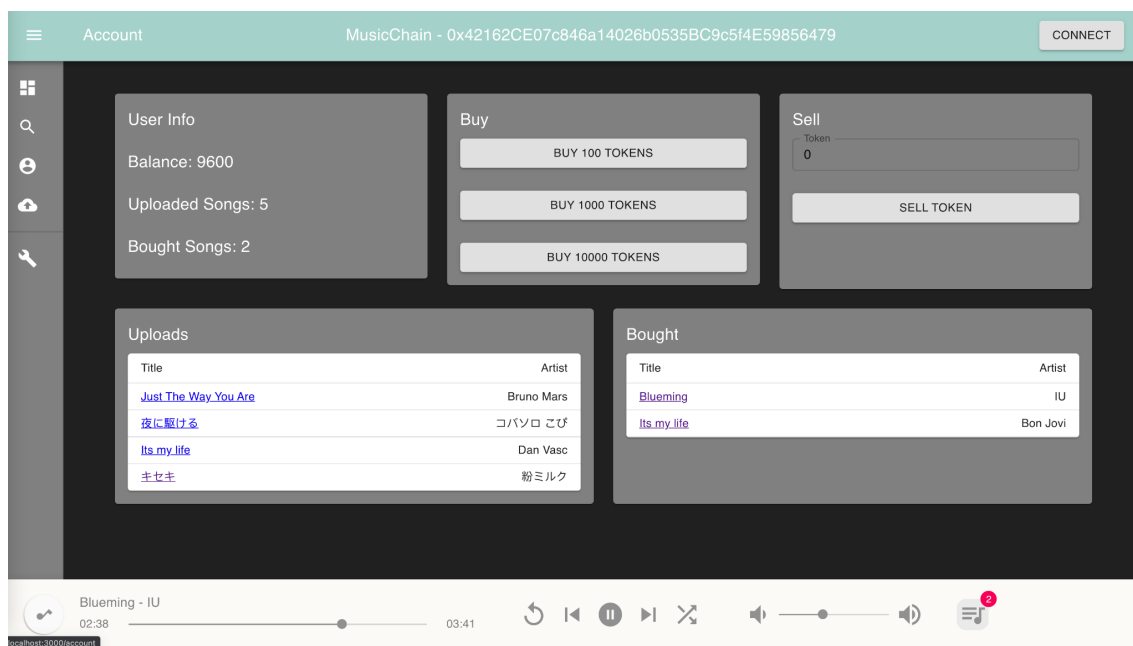
六、實作細節與設計

我們在我們的智能合約中實作了一個簡單的去中心化交易系統（圖表 4）。DEX 有三個功能：購買、賣出和轉移。我們在內部有一個 MusicToken 當作我們交易的標準貨幣。



圖表 4 DEX

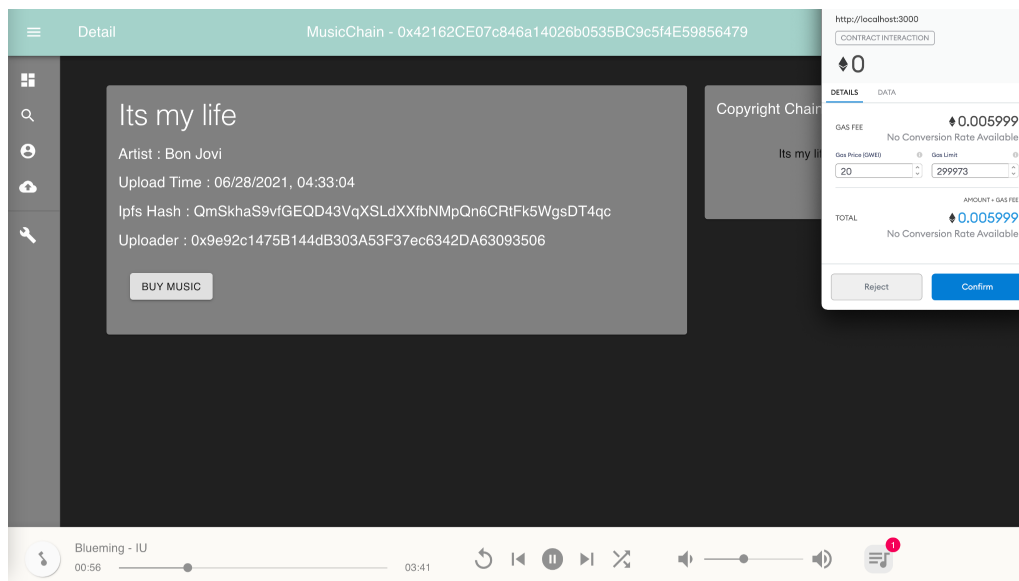
DEX 實現購買和賣出的功能為轉換 ETH 和 MusicToken。因此，使用者可以花費 ETH 來轉換成 MusicToken 來購買音樂（如圖表 5），而創作者也可以將收益從 MusicToken 轉換為 ETH。而轉移的功能讓購買音樂的流程可以直接發生在合約內部，避免額外交易以及價格波動。



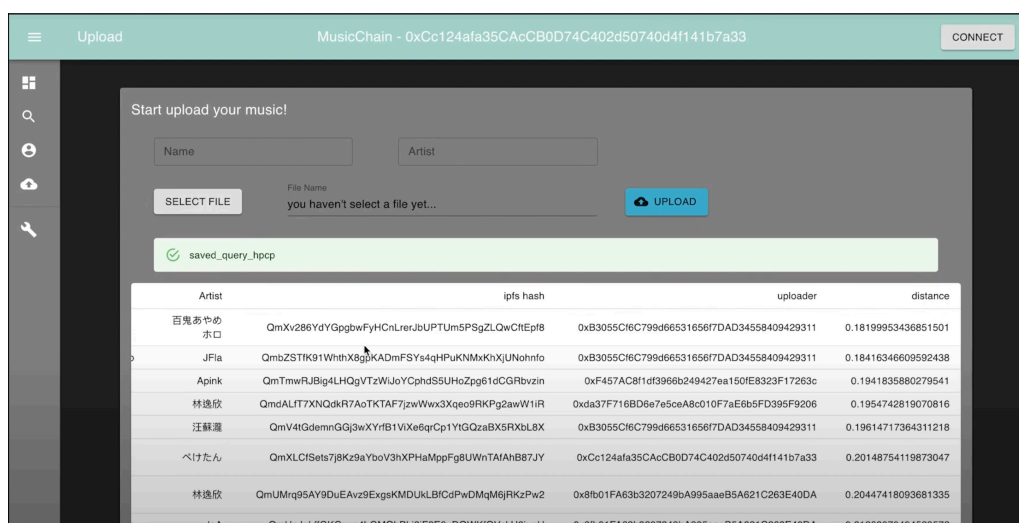
圖表 5 帳戶頁面（可轉換 ETH 和 MusicToken）

而至於購買音樂（圖表 6）的部分，每個音樂的定價為 100 個 MusicToken。我們有設計一個簡單的演算法去分配利潤：創作者和被翻唱的創作者會有共同的 1:1 利潤分配。舉例來說，假設我們現在有三首音樂，分別為 A、B 和 C。而翻唱的關係為 C cover B ($C \rightarrow B$)，B cover A ($B \rightarrow A$)，而完整的 cover chain 為 $C \rightarrow B \rightarrow A$ 。當使用者花費 100 tokens 去購買音樂時，C 和 B 會各拿到 50 tokens，但是 B 其實也有 cover A，因此我們要再做一次利益分配，也就是 B 和 A 會各拿到 25 tokens。最後總結：C 拿到 50 tokens，B 拿到 25 tokens，A 拿到 25 tokens。為了避免無限迴圈，我們設定最大上限為 7，也就是在 cover chain 的尾端會拿到 1 個 token，若是更長的話，我們則會跳出迴圈。從另外一點來講，因為 Ethereum 的實作關係，1 個

token 事實上也不可繼續分割。



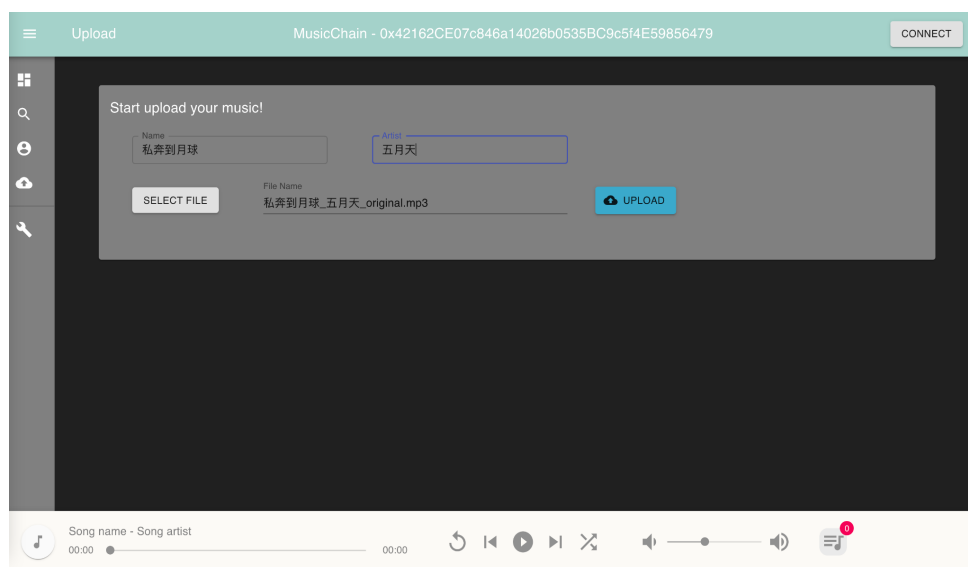
圖表 6 音樂詳細頁面（可發送交易來購買音樂）



圖表 7 音樂上傳頁面（圖中顯示音樂相似度）

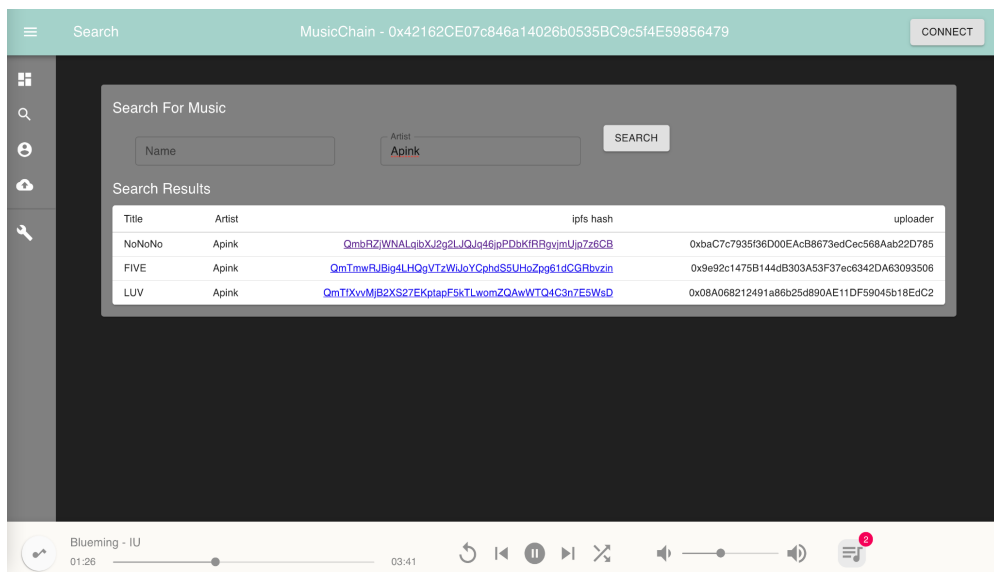
在上傳音樂的部分，我們會進行以下流程：首先，先將上傳的音樂丟到我們後端的音樂辨識系統，該系統會儲存所有以上傳過之音樂的 HPCP 特徵。每當有一首音樂上傳時，我們會拿它跟其他音樂的 HPCP 當作 input 跑 Cover Song Identification（圖表 7）。如果音檔是和現有音樂一模一樣，網站會禁止使用者上傳，因為這樣等於違反著作權法。此外，在這部分我們定義 0.1 為 threshold。如果一兩首歌之間的距離大於它，我們會將這兩段音檔視為不同歌曲。反之，則會視他們為 cover 的關係。如果所有的音樂最小的距離都大

於 0.1，我們則認定這首音樂是原創的，(在這個定義下就是一首歌第一個上傳者)。如果多一首音樂和上傳的音檔距離小於 0.1，那我們就用距離最小者當作新增音樂所 cover 的音檔。確定 cover 資訊後，我們就連同歌名，演奏者，上傳者等資訊紀錄在 Ethereum 上。若是從 DAG 來看，每個音檔就是一個 node。儲存 cover 關係就是建立一條有方向的 edge，把二次創作的音樂指向原著。



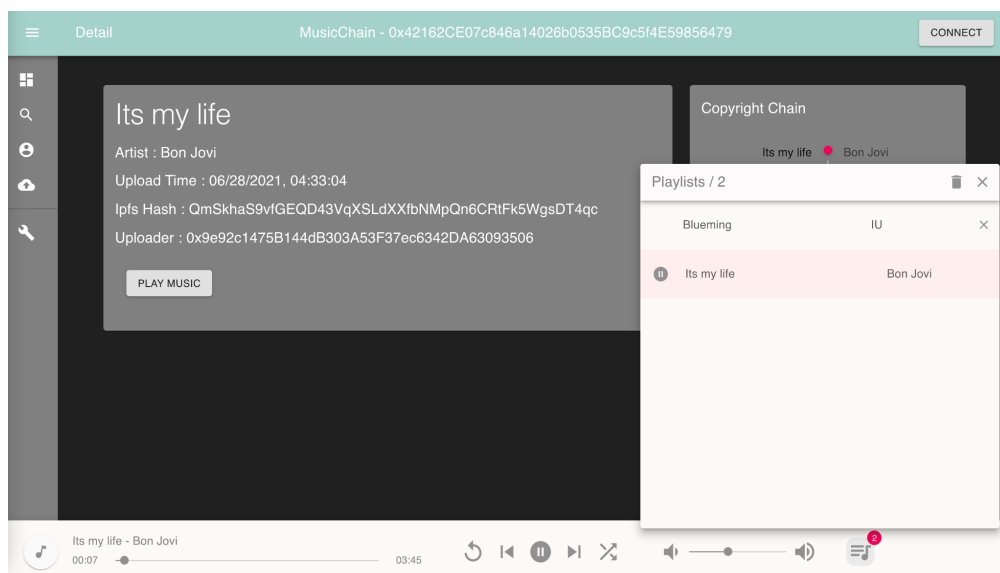
圖表 8 音樂搜尋頁面

至於音樂搜尋的部分（圖表 8,9），我們讓使用者可以用音樂的名稱或是歌手名稱來搜尋，我們會看一個音樂的名稱或是歌手名稱是否符合剛剛輸入的關鍵詞，接著我們回傳符合的音樂。我們目前是在前端實作模糊搜尋的功能，預計之後要把他移到智能合約上面。



圖表 9 音樂搜尋結果呈現

音樂播放（圖表 10）的部分，當使用者購買了音樂後，可以前往音樂的內容頁面點選播放音樂，便可以直接聆聽當前的音樂，該首音樂也會被加到當前的播放清單中。在我們設計的音樂播放器上支援了很多功能，使用者可以選擇切換音樂、調整音量、單曲或隨機播放等等。



圖表 10 網頁播放器及播放清單

七、結論與未來展望

這個 project 裡，我們成功的實作出以 cover 關係的 DAG 來作為分潤的交易機制，並將 Cover Song Identification、區塊鏈技術，IPFS 和網站做整合。對於 IPFS 的檔案儲存，雖然 IPFS 上面的檔案有可能隨著時間就被刪除，然而，

我們可以去利用 IPFS 內建的 pin 的機制讓檔案比較不容易被刪除。另外，我們也可以定時的去 refresh 我們區塊鏈上所儲存的音樂。具體來說，因為我們有把音樂的 IPFS Hash 存在區塊鏈上，我們可以架設一個系統在一定時間之後就去下載每個音樂，然後重新上傳。因為 IPFS 它的特性，因此不會有重複的檔案，這樣子便可以保證所有的音樂都能夠一直存在在區塊鏈上面。

在 Graph 的部分，以目前的情況，DAG 是不會存在一個 node 同時指到兩個以上的 node，因此比較像是一個樹的結構。而在更廣泛來說是有可能是有用樹無法滿足的情況（像是同時 Remix 兩首歌）。這部分尚待在音樂資訊檢索的領域上做更深入的研究，才能支持這樣的功能。

此外，目前的 edge 並沒有隨著 cover 關係遠近而有不同的 weight，可能可以利用 weight 的不同來動態調整，原創和二次創作之間的利益分配。另一方面，目前只有實作單向的結構，只有從翻唱指向被翻唱者，而沒有從另一個方向指回來。而實作從創作者指回來有很大的好處，因為我們就可以有一棵關於這個音樂整個翻唱的關係圖，而整個創作社群也可以獲得好處。比方說目前我們很難去避免一些惡意的使用者來直接上傳他的音樂而不是透過我們的後端，但是如果有實作剛剛講的關係圖的話，我們可以額外實作一個功能：只要我們在這個關係圖上無法找到，我們就不會回傳搜尋的結果，因此即便有惡意上傳者，也不會影響到原本的創作者。

再者，如果我們的 DAG 有夠多的資料，他其實也可是另一種音樂搜尋的方式。這部分可能可以套用一些 graph 的演算法像是 Breadth First Search, Depth First Search 等去做音樂關係的搜尋，這樣的方式可以讓音樂和音樂的關係更緊密。而如果這些音樂之間 cover 關係的資料到達一定的數量，這樣的資訊甚至可能可以用作深度學習之音樂生成的訓練資料。因為區塊鏈的特性，我們可以確保我們的資料是有一定的品質（上傳的資料不可更改，創作者會慎重上傳，且資料不會被竄改）。

在分潤機制的部分，我們目前只考慮了比較簡單的狀況，然而事實上可

能需要做出一些調整。比方說某首歌是因為某個翻唱者才紅起來的，因此翻唱者的音樂下載次數遠遠比被翻唱者多，雖然我們仍然需要注重著作權，但是卻可以做出一些細部的動態調整，而不是現在靜態的 1:1 方式。比例的詳細制定也會牽扯到經濟學的知識，這部分我們認為也是這項 project 未來值得努力的方向，也有很大的潛力和發展空間。

八、參考資料

- ¹ PETERWU,(2019), 音樂人、影像工作者不能不知！音樂著作權完整解析, Retrieved From <https://blog.addmusic.tw/2019/07/29/音樂人、影像工作者不能不知！音樂著作權完整解/>
- ² Ujo Music : <https://blog.ujomusic.com>
- ³ Museum Protocol : <https://www.muzeumprotocol.org/zh/>
- ⁴ Christopher J. Tralie. (2017). Early MFCC And HPCP Fusion for Robust Cover Song Identification.
- ⁵ Du, Xingjian & Yu, Zhesong & Zhu, Bilei & Xiaoou, Chen & Ma, Zejun. (2021). Bytecover: Cover Song Identification Via Multi-Loss Training. 551-555. 10.1109/ICASSP39728.2021.9414128.
- ⁶ Chang, Sungkyun & Lee, Juheon & Choe, Sang & Lee, Kyogu. (2017). Audio Cover Song Identification using Convolutional Neural Network.
- ⁷ Solidity Docs : <https://docs.soliditylang.org/en/v0.8.6/>
- ⁸ Angela Beklemysheva,(2021), *Making Effective Use of Smart Contracts* Retrieved From <https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/>
- ⁹ Essentia : <https://essentia.upf.edu/>
- ¹⁰ Joan Serra. Identification of Versions of the Same Musical Composition by Processing Audio Descriptions. PhD thesis, Universitat Pompeu Fabra, Spain, 2011.
- ¹¹ MDN Web Docs EventSource : <https://developer.mozilla.org/en-US/docs/Web/API/EventSource>