

Vanft: Version-enabled NFT Platform

Wenqian Yan (wy2249)

May 12, 2022

Abstract

Non-fungible tokens, or NFTs, are a fairly new milestone in the blockchain with unique tokens to represent digital assets. Its uniqueness makes it perfect for ownership and have become very popular for use in digital collectibles and other applications. While the common suggestion for NFTs is that they should be permanent and immutable after minting for the reason of copyright and rarity, there is a practical problems for owners of NFTs that they might want to edit or modify their works for various valid reasons. In this paper, we present the design and implementation of Vanft[1], a version-enabled NFT Platform with original NFT and its append-only version history, to provide an approach for maintaining ownership without sacrificing editability.

1 Introduction

As a way to represent unique items as an Ethereum-based asset, NFTs are currently taking the digital art and collectibles world by storm. By tokenizing these digital assets, NFTs provide strong ownership and copyright because it is impossible to copy and paste one of existing NFT to a new NFT. This digital uniqueness is so rare and valuable in the current internet market, where copies and fakes flourish, that NFT attracts many digital artists and collectors.

To depend on digital tokens to represent the uniqueness of works, NFT expects that owners should not change NFT tokens and keep them immutable after minting. However, it's very possible that owners want to change their minted NFTs if they realize later the NFT does not meet community standards or find a better way to interpret and express their works. Also, in the NFT video and gaming world where everything changed rapidly, the creator of the NFT may also need to update information of the NFTs to keep items as being desired in the game.

With the principle of keeping NFTs unique and the practical needs to modify them, the existing approach on the market is to change the metadata representing their properties.[4] There are two kinds of metadata of NFTs: mutable metadata and immutable metadata, and only mutable metadata could be edited or updated. Examples of mutable metadata includes description, owner, categories, and image of NFTs, and in most cases, immutable metadata includes the creator, token name and token ID. Vanft[1] employs a hybrid approach to maintain the immutability of NFTs and provide editability by creating metadata of new versions and appending with old versions.

2 Originality

We build Vanft[1] as a new NFT platform with versions enabled, inspired by git and append-only database in the area of computer science. Instead of overwriting the whole digital asset, the uniqueness and originality of NFT tokens will be kept with their initial minted versions, and any modifications their owners want to make is made by appending a metadata of the new version. That's to say, each modification will create a new version of metadata json file and append to the previous versions, instead of overwriting any initial values and losing the uniqueness of hash. The new version metadata will be stored in ipfs in infura and the url will be appended to versions in the NFT. The owners of each NFT can also view their version histories and check the specific version they prefer to work on. Even though Vanft cannot perfectly solve the problem of uniqueness and editability, it provides an experiment and approach for it.

3 Dapp Showcase

In this section, we will showcase the dapp Vanft[1]. It's mainly implemented in solidity (for smart contract) and Vue.js (for frontend pages). By hardhat, we deployed the smart contract to huygen_dev nodes on CCN (Computecoin Networks)[2] and currently supports Ale wallet. Users should install Ale wallet Chrome extension[5] before using Vanft.

3.1 Home Page

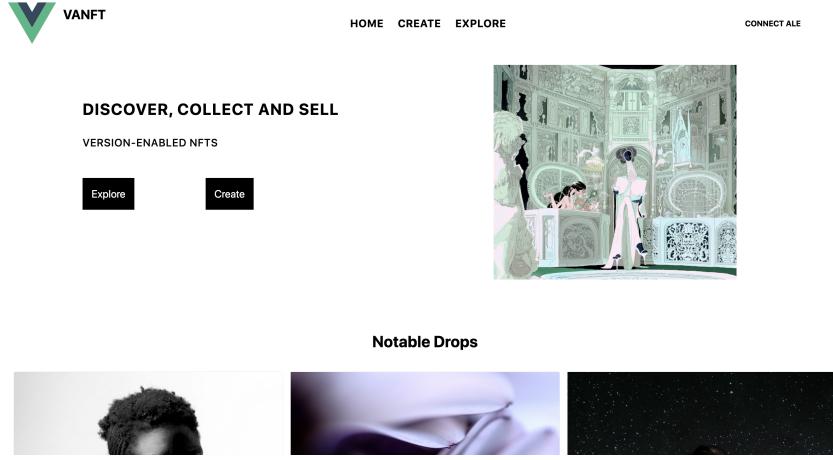


Figure 1: Home Page

The Home Page (Figure 1) exhibits the goal of Vanft and routes to two main pages on navbar, Create Page and Explore Page. Users are expected to install Ale Chrome Extension[5] in advance, and they should click on "Connect Ale" on the left-top to connect their Ale wallet to Vanft for any transactions to make in the future. An Ale window will be popped up after clicking, and users will be asked to enter their password and authorize. Then, the user's account address will replace "Connect Ale" to represent that their accounts are connected. After connecting, users can click on their account address to be redirected to User Collection Page showing the collections they bought and listed. In the bottom of Home Page, there are three Notable Drops for showcase and ascetics.

3.2 Explore Page

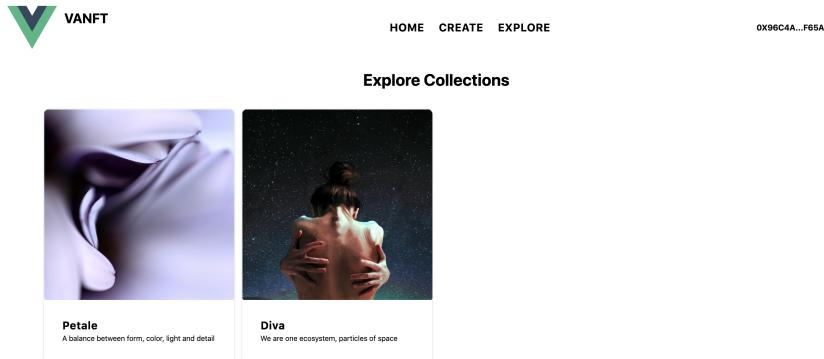


Figure 2: Explore Page

The Explore Page (Figure 2) shows all the collections minted by all the users on market for sale. Each collection is represented as a board with image associated, name and description. Users can click on each board to know about more details of the NFT item. If the NFT item is bought by the user, it will disappear from the Explore Page and only shows in User's Collection Page.

3.3 Create Page

Figure 3: Create Page

Users can create and mint their NFTs on the Create Page (Figure 3). They will upload their digital assets from local to Vanft, and set name, price and description for the NFT on this page. Under the hood, Vanft will store the digital assets to IPFS on infura.io, store metadata containing name, description and url (CID) of the digital assets as a json file to IPFS[3], and then set the url (CID) of this initial created metadata as the tokenUrl in smart contract that binds to the token. Even though the owner of NFTs can add new versions to the NFT, the tokenUrl is not changed and the copyright and uniqueness of NFTs on Vanft is always based on this initial asset.

3.4 User's Collection Page

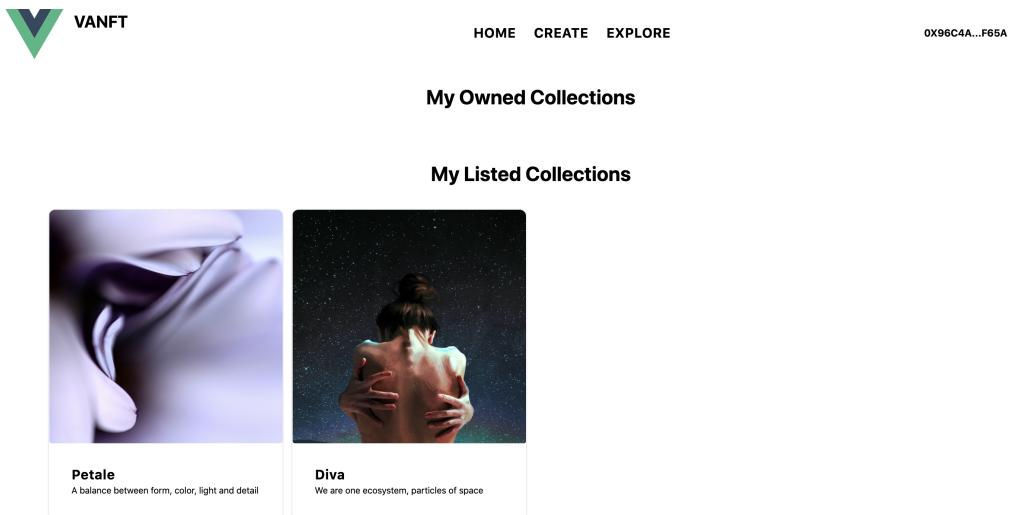


Figure 4: My Collection Page

The User Collection Page (Figure 4) shows the collections the user have bought and listed to the blockchain networks. Each collection will be layed out as a card with the image associated, name of NFT and its description. In the picture shown above, this user have two listed collections and he/she can click on them to view more details about these NFTs and execute transactions, e.g. buy, resell and create new version, depending on permission. Items in "My Owned Collections" will not appear on market (Explore Page) until the user chooses to resell it.

3.5 NFT Detail Page

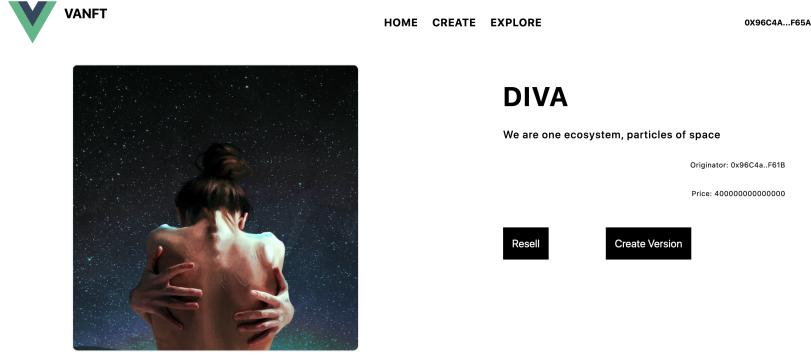


Figure 5: NFT Detail Page

The NFT Page (Figure 5) shows detail of this NFT assets including its name, description, originator, image and price. If the NFT is owned by this user, he/she can resell it or edit it by creating a new version on this page, as shown in the figure above. If the NFT is neither owned nor listed by this user, he/she can buy it (a "Buy" button will appear on webpage replacing "Resell" and "Create version"). For transactions like buy and resell, a popup alert will appear to inform user if the transaction succeeds or not. If it succeeds, the page will be redirected to User Collection Page for them to view and check collections.

3.6 NFT Edit Canvas Page

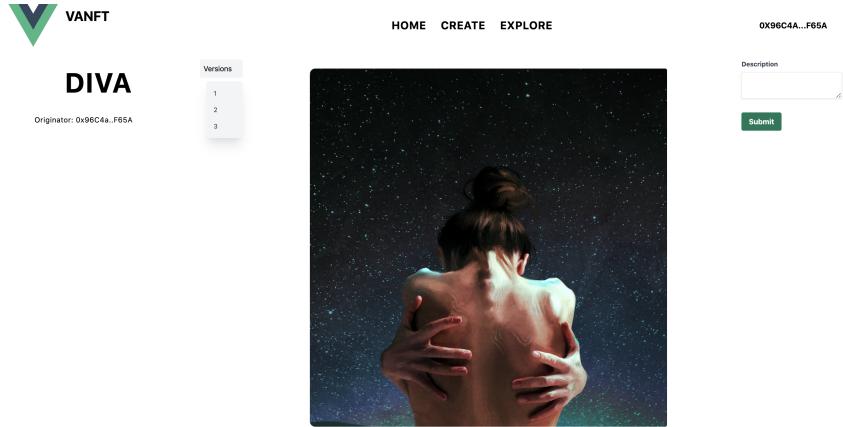


Figure 6: NFT Edit Page

Users will be redirected to Edit Canvas Page (Figure 6) if they click "Create Version" on NFT Detail Page. On this page, they can view different past versions of this NFTs by selecting the corresponding

version number in the drop-down button. In the middle of page, there is a canvas board that users can edit and draw (not fully supported yet). By clicking "submit", a new version will be created, and under the hood, a new metadata file for this version will be created and stored to IPFS and the url of the new metadata will be appended to old versions of the NFT.

4 Conclusion

In this paper, we present the design and implementation of Vanft, a version-enabled NFT Platform with original NFT and its append-only version history, to provide an approach for maintaining ownership without sacrificing editability. With Vanft, the immutability of NFTs is kept and meanwhile users can edit their minted NFTs by creating new versions of it. The modifications will be saved as metadata in IPFS and all the versions are kept by append-only, instead of overwriting the old data. Even though Vanft cannot perfectly solve the problem of uniqueness and editability, it provides an attempt and approach for it.

References

- [1] "wy2249/Vanft" GitHub, <https://github.com/wy2249/vanft>.
- [2] Computecoin. "About Computecoin (CCN)", <https://computeconetwork.gitbook.io/computecoin/computecoin/about-computecoin-ccn>.
- [3] <https://infura.io/>
- [4] <https://blog.assetmantle.one/2022/05/09/can-you-edit-an-nft-after-it-has-been-minted/>
- [5] <https://chrome.google.com/webstore/detail/ale/cflgahhmjlmnjbikhakapcfkpbcmllam>