# Artwork NFTs for Online Trading and Transaction Cancellation

Elsi Ahmadieh, Nour El Madhoun

# Artwork NFTs for Online Trading and Transaction Cancellation

Elsi Ahmadieh*, Nour El Madhoun†‡ §
* Lebanese University, Faculty of Technology, Department Communications and Computer Networks Engineering, Beirut, Lebanon
† LISITE Laboratory, ISEP, 10 Rue de Vanves, Issy-les-Moulineaux, 92130, France
‡ Sorbonne Université, CNRS, LIP6, 4 place Jussieu 75005 Paris, France
§ Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91190, Gif-sur-Yvette, France
E-mails: elsiahmadieh@gmail.com; nour.el-madhoun@isep.fr
nour.el_madhoun@sorbonne-universite.fr; nour.el-madhoun@universite-paris-saclay.fr

***Abstract*—Blockchain, a revolutionary technology that has attracted a great deal of interest in recent years, is transforming various industries and redefining the way we think about data, transactions and trust. It began as a secure environment for online commerce and has extended its scope to other areas such as healthcare systems, e-voting, supply chain, digital assets and telecommunications. Its key features are decentralization (no need for a Trusted Third Party (TTP)), immutability, transparency and the implementation of smart contracts. These are used to automate and execute agreements between blockchain users. They are also used to create unique digital assets (such as artworks, real estate, collectibles, etc.) by representing them as Non-Fungible Tokens (NFTs). In this paper, we present a smart contracts application that allows the creation of an NFT, that represents an artwork, with the possibility of trading this NFT on a marketplace, selling it and transferring the ownership from the seller to the buyer, as well as cancelling transactions and reimbursing buyers with paid amounts.**

***Index Terms*—Blockchain, NFT, Smart-Contract, Token, Ethereum, Solidity, Artwork, Trading, Buy, Sell, Cancel, Refund.**

## I. Introduction

Unlike traditional centralized systems, where a central authority controls and validates transactions, blockchain technology operates on a distributed network where participants collectively validate and maintain data integrity. This distributed nature eliminates the need for intermediaries and fosters the transparency, security and efficiency of various processes. Recently, and due to its wide use and implementation, blockchain technology has vastly developped and is now applied in finance, medical systems, e-voting, supply chain, telecommunications, etc. [1]. It uses a decentralized approach to record transactions, meaning that data are stored on multiple computers instead of a single central server. This makes the blockchain resistant to manipulation and data loss, as it is extremely difficult to modify data stored on many computers simultaneously. In addition, transactions stored on the blockchain are validated by a multitude of users, which increases the reliability and transparency of data [2].

Blockchain technology allows to implement smart contracts in its environment. These smart contracts are automated programs used to perform specific actions once predetermined conditions are met. They are also used to create unique digital assets (such as artwork, real estate, collectibles, etc.) by representing them as Non-Fungible Tokens (NFTs). These NFTs are stored in the blockchain and can be securely transferred between users without the intervention of a Trusted Third Party (TTP) [3].

Indeed, blockchain can also be used to generate and implement fungible tokens via smart contracts. Fungible tokens are digital assets that are interchangeable on a one-to-one basis. This means that each unit of a fungible token is identical to another unit of the same token and can be exchanged for it without distinction [4]. The first objective of this paper is to present an overview of smart contracts and the types of tokens (fungible and non-fungible). Our second objective is to improve the application of smart contracts that we developed in our previous paper [5], by taking the business logic a step further: implementing transaction cancellation and refund to the buyer. This paper is organized as follows: in section II, we present the principle of smart contracts and tokens (fungible tokens and NFTs). In section III, we present our improved smart contracts application. The final section concludes this paper.

## II. Overview of smart Contracts and Tokens

### A. Definition of smart Contracts

A smart contract is a computer program that automatically applies and executes the terms of an agreement or contract between participants. They allow to replace traditional contracts, that are usually written in legal language and often require a TTP to be executed. Instead, smart contracts define the terms of a contract in a computer language, making them automatically executable on the blockchain without needing to go through human intervention or a TTP [6] [7].

Smart contracts are immutable, meaning that once deployed on the blockchain, they cannot be modified. This guarantees contract transparency and security, as the terms defined in the code are immutable and can also be verified by all blockchain participants at any time. In addition, smart contracts are decentralized, meaning that they are executed on a distributed network of nodes rather than on a centralized server. This makes them less vulnerable to attacks and server failures, as

there is no single point of failure in the network. Indeed, smart contracts can be used in a variety of use cases, such as money transfers, decentralized elections, real estate management and virtual asset sales. They can also be used to implement fungible or non-fungible tokens (see next section II-B) [8].

### B. Tokens and Smart Contracts

Tokens are digital units of value that can be used to represent various assets in the blockchain, such as real estate, stocks, products, etc. Tokens can be traded or used to access products or services. There are two types of tokens [9] :

- Fungible tokens: these are digital assets that can be easily divided and traded in equivalent units, such as bitcoins or ethers. They are considered fungible because their units are interchangeable.
- Non-Fungible Tokens (NFTs): these are unique and indivisible. They cannot be divided or exchanged into equivalent units. NFTs are often used to represent objects, such as images, artworks, video, etc.

The value of tokens in blockchain lies in their ability to facilitate the management, tradability and security of assets using decentralized blockchain technology. Tokens can also provide an easier, faster and cheaper way to carry out financial transactions, while enabling greater transparency and accountability in transactions [10]. There are two types of smart contracts that are used to implement them [11]:

- Fungible tokens can be implemented using standard contracts: Ethereum Request for Comment 20 (ERC20). An example of using smart contracts to implement a fungible token could be the launch of a new cryptocurrency. The ERC20 smart contract could be deployed on the blockchain to automate the creation of new tokens and manage transactions between users. When a user wishes to purchase tokens, he can send funds to the smart contract address. The smart contract, using preprogrammed rules, can automatically create and allocate the new tokens to the user's address. Similarly, when a user wishes to exchange his tokens for other tokens or funds, the smart contract can automate this process by transferring tokens and funds between the user's different addresses without the intervention of a TTP.
- NFTs can be implemented using standard contracts: Ethereum Request for Comment 721 (ERC721). An example of using smart contracts to implement NFTs could be registering the ownership of a unique a unique artwork. The smart contract can be deployed on the blockchain to automate the creation of an NFT representing that artwork: when an artist wishes to register ownership of his artwork, a smart contract can be deployed on the blockchain to create the NFT associated to his artwork. When a buyer wishes to acquire this artwork, he can carry out a transaction to purchase the associated NFT. The smart contract can automate this process by transferring ownership of the NFT from the artist's address to that of the buyer.

## III. Smart Contracts Application for NFTs

NFTs play an essential role in the trading of digital assets and the transfer of ownership of these assets on the blockchain. We therefore rely on this characteristic of NFTs and the blockchain to trade an artwork visual on the blockchain (see sections I and II-B) as if we were doing the same operation in real life. As our previous paper [5] focused on this type of application and guaranteed: (a) that artists could show their artworks online, exchange them and earn money (ether) in return (from buyer to seller), (b) that ownership was transferred, while retaining the inventor's signature even when ownership changed, we rely on this application to implement a new workflow, allowing the inventor to cancel the transaction, and our smart contract to then refund the buyer the amount he paid. In this section, we present a summary of the previously undergone functions on our smart contracts in our previous application, and the new implemented logic developed with the solidity language on the Ethereum blockchain.

- Solidity: it is an object-oriented programming language for implementing smart contracts on various blockchain platforms, most used on Ethereum. Programs in Solidity run on Ethereum Virtual Machine [12].
- Remix-Ethereum IDE: it is one of the most commonly used development environments for developing Ethereum smart contracts. It can provide network-based solutions to quickly compile and deploy Solidity and Vyper code on local VMs or external Web3 providers (such as Metamask) [13].

```
contract NFTCreation is ERC721URIStorage {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIds;
    address contractAddress;

    constructor() ERC721("NFTCreation", "NFTCreation") {}

    function createToken(string memory tokenURI) public returns (uint) {
        _tokenIds.increment();
        uint256 newTokenId = _tokenIds.current();
        _safeMint(msg.sender, newTokenId);
        _setTokenURI(newTokenId, tokenURI);
        setApprovalForAll(contractAddress, true);
        return newTokenId;
    }
}
```

Fig. 1. Code in solidity of NFTCreation Contract [5]

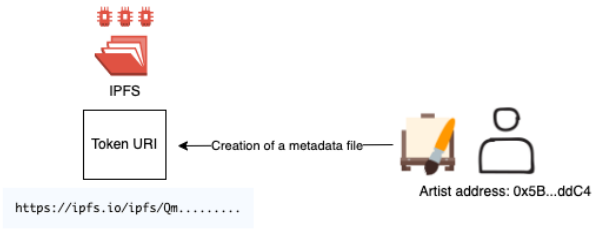


Fig. 2. Creation Token URI [5]

### A. Application Development

The first step in our previous contracts is to develop a smart contract for the creation of NFT (see Fig. 1), creating the token from the image URI resulting from the conversion of the visual of this artwork using a system called IPFS (InterPlanetary File System), as shown in Fig. 2. We then add this created token to

the marketplace by creating another contract that will manage all other operations: the NFTMarketplace contract (see Fig. 3). This contract will deal with the function addItemToMarket, which adds this token so that it can be traded on the marketplace [5].

```
contract NFTMarketplace is ReentrancyGuard {
  using Counters for Counters.Counter;
  Counters.Counter private _itemMarketIds;
  Counters.Counter private _itemSoldIds;
  address payable owner;
  constructor() {
   owner = payable(msg.sender);
  }
  mapping(uint256 => MarketItem) private idToMarketItem;
  struct MarketItem { ..... }
  mapping(uint256 => ItemSold) private idToItemSold;
  struct ItemSold { ......}
  function addTokenToMarket(address nftContract, address inventorOwner, address sellerOwner,
   uint256 tokenId,  uint lastItemMarketId, uint256 price)  public payable nonReentrant returns (uint256) {
       ....
   }
   function sellTokenAndChangeOwner(address nftContract, address buyer, uint256 tokenId,
   uint256 itemMarketId) public payable nonReentrant returns (uint256) {
       ....
   }
   function getMarketItemById(uint256 marketItemId) public view returns (MarketItem memory) {
       MarketItem memory item = idToMarketItem[marketItemId];
       return item;
   }

   function getItemSoldById(uint256 itemSoldId) public view returns (ItemSold memory) {
       ItemSold memory item = idToItemSold[itemSoldId];
       return item;
   }
}
```

Fig. 3. Code in solidity of NFTMarketplace Contract [5]

After adding the item to the marketplace, buyers are now able to purchase this item (token). The sellItemAndChangeOwner function (see Fig. 4) will therefore handle this operation, by transferring ownership from the seller to the buyer, and transferring ether or the initialized price (by the inventor) back to the inventor. We note that even if ownership is transferred, the inventor's signature is always tracked and will never change.

```
function sellTokenAndChangeOwner(address nftContract, address buyer, uint256 tokenId,
uint256 itemMarketId) public payable nonReentrant returns (uint256) {
      ERC721(nftContract).transferFrom(idToMarketItem[itemMarketId].sellerOwner, buyer, tokenId);
      _itemSoldIds.increment();
      uint256 newSoldItemId = _itemSoldIds.current();
      idToItemSold[newSoldItemId] = ItemSold(newSoldItemId, itemMarketId,
      payable(idToMarketItem[itemMarketId].inventorOwner), payable(buyer),
      nftContract, tokenId);
      return newSoldItemId;
}
```

Fig. 4. Function sellTokenAndChangeOwner [5]

If we want to move on to a more commercial logic, we absolutely need to consider the case of a transaction or trading being cancelled by the inventor. This also happens in real life. For example, the inventor shows the item he wants to sell, then someone tries to buy that item, and the inventor may later change his mind and try to cancel that transaction (keeping in mind that only the inventor has the right to carry out this cancellation) while reimbursing the buyers. To remedy this, we've added a function called cancelTransaction, as shown in Fig. 5 and Fig. 6. It takes tokenId as input. First, it checks whether or not this transaction has been previously cancelled. Then, it makes sure that only the inventor has the privilege to cancel this transaction and sets the canceledTransaction boolean to true to prevent the buyer from completing his transaction. The final step in our logic is to refund the buyer the amount he paid before the cancellation. By making "owner.callvalue: amount("")" return true, the ether is sent to the buyer's address (Fig. 7). Throughout this process, we present two scenarios of our application. The first scenario is when a user attempts to purchase the NFT after it has been cancelled by the inventor (in this case, our smart contract sends an error to prevent the purchase, see Fig. 8). The second is when the inventor attempts to cancel the trading after the buyer has completely purchased the item and paid the appropriate amount (here, the smart contract also sends an error to prevent the trading from being cancelled, see Fig. 9). Fig. 10 summarizes our process.

```
function cancelTransaction(uint256 tokenId) public payable {
    require(!canceledTransaction, "No active transaction to cancel.");

     nftToOwner[tokenId] = msg.sender;
     address owners = nftToOwner[tokenId];
    canceledTransaction = true;

    uint256 amount = pendingWithdrawals[owners];
    pendingWithdrawals[owners] = 0;
    (bool success, ) = owner.call{value: amount}("");
    require(success, "Failed to send ether to owner.");
}
```

Fig. 5. Cancel Transaction function (1)

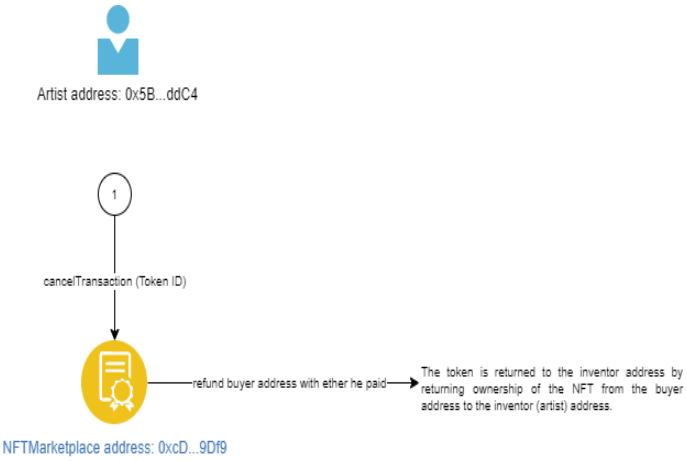


Fig. 6. Cancel Transaction function (2)

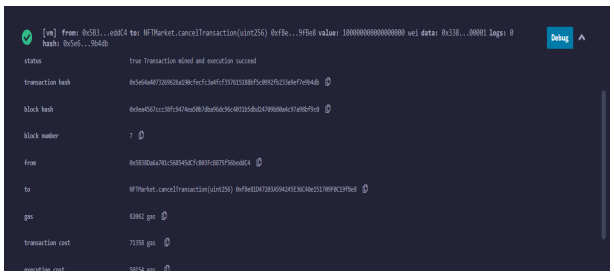

Fig. 7. Successful Cancellation of Trading

### B. Advantages of Trading with NFTs

In this section, we mention the main advantages of trading with NFTs [14] [15] [16]:


Fig. 8. Sell Item After Cancellation

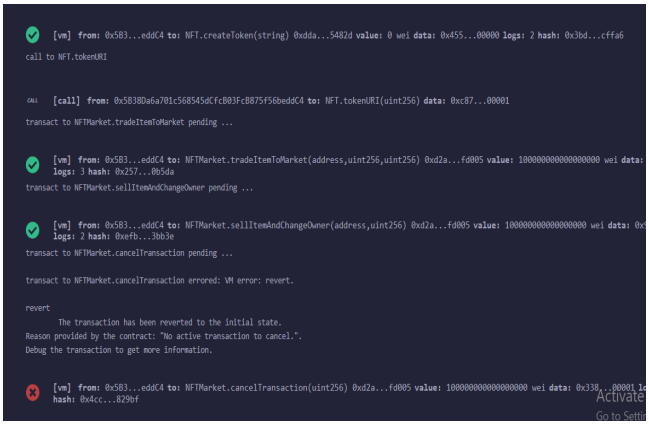


Fig. 9. Cancel Trading of Sold Item

- Unique tokens: NFTs are unique cryptographic tokens that are presented and stored on the blockchain. Unlike fungible tokens, that all have the same value and can be traded in the same way, NFTs cannot be duplicated. This enhances the security of transfers and trading (see section II-B).
- More reliable and private: one of the great advantages of tokenization of items and assets is that their trading becomes more reliable thanks to the blockchain technology, because no TTP is involved, and therefore no delay in transfer time, no TTP can know what is going on, and most importantly, there is less risk of fraud.
- Protection of rights and ownership: NFTs represent the identity of individuals, their property rights, etc. Thus, if a person owns an item that is presented to the public, he will have full access to track transactions and purchase requests for this item, putting his name and rights on that item, and no one will be able to update or change the information about this asset.
- Reduced transaction costs: before the NFT revolution, artists (or people) used to travel to art shows in different countries to sell (or buy) artworks. Today, trading with NFTs has given a great advantage, insofar as artists (or people), have the possibility to sell (or buy) artworks from where they are, without needing to travel and spend money and time to do their things.

### C. Drawbacks of Trading with NFTs

In this section, we mention the main disadvantages of trading with NFTs [14] [17] [18] [19]:

- Volatility and risk of loss: NFTs can be very volatile due to the speculative nature of the market. Prices can change quickly and often for no apparent reason. In fact, trading with NFTs carries a high risk of loss due to the volatility of the market and the scarcity of certain NFTs.
- Limited liquidity: the NFT market is still young and liquidity may be limited, which can make it difficult to sell an NFT quickly.
- Complexity: trading with NFTs can be complex and require a thorough understanding of the underlying technology, which may deter some investors.

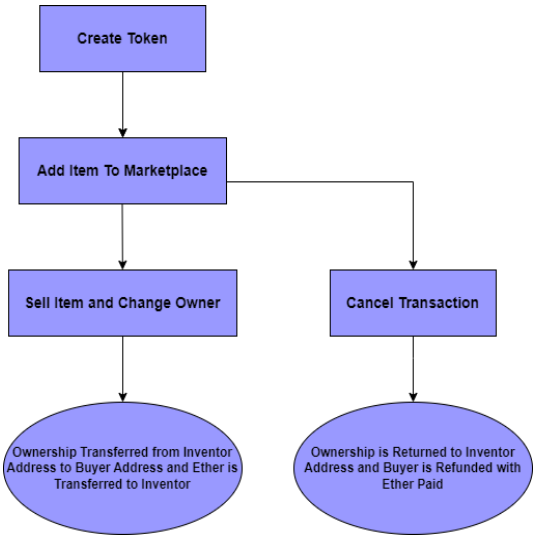


Fig. 10. Flowchart for our Application Summary

## IV. Conclusion

Blockchain technology is favored today because of its high level of security, high availability and low cost for transferring data and money, as well as for trading and owning objects. In this paper, we presented an overview of smart contracts and NFTs, and we improved our previous application of smart contracts (which included the creation of an NFT that represents an artwok and can then be sold on an NFT marketplace while transferring ownership from the seller to the buyer) by adding the possibility of cancellation of the transaction by the inventor and reimbursement to the buyer.

## References

[1] N. El Madhoun, J. Hatin, and E. Bertin, "A decision tree for building it applications," *Annals of Telecommunications, Springer*, vol. 76, no. 3, pp. 131–144, 2021.

[2] A. S. Rajasekaran, M. Azees, and F. Al-Turjman, "A comprehensive survey on blockchain technology," *Sustainable Energy Technologies and Assessments, Elsevier*, vol. 52, p. 102039, 2022.

[3] M. D. Murray, "Nfts rescue resale royalties? the wonderfully complicated ability of nft smart contracts to allow resale royalty rights," *The Wonderfully Complicated Ability of NFT Smart Contracts to Allow Resale Royalty Rights (July 15, 2022)*, 2022.

[4] R. Descio-Trineto, M. Pillon, G. Koslovski, and C. Miers, "Dcanon: Towards distributed certification authority (ca) with non-fungible token (nft)," *International Conference on Advanced Information Networking and Applications, Springer*, pp. 286–298, 2023.

[5] E. Ahmadieh and N. El Madhoun, "Nfts for online trading of artworks," *1st IEEE international workshop on cryptocurrency exchanges (CRYPTOEX 2023) Co-located with IEEE Int'l Conference on Blockchain and Cryptocurrency (ICBC 2023)*, 2023.

[6] M. Fries and B. P. Paal, "Smart contracts," *Mohr Siebeck*, 2019.

[7] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.

[8] M. Ndiaye and K. Konate, "Security strengths and weaknesses of blockchain smart contract system: A survey," *International Journal of Information and Communication Engineering*, vol. 16, no. 5, pp. 134–143, 2022.

[9] L. Ante, "The non-fungible token (nft) market and its relationship with bitcoin and ethereum," *FinTech*, vol. 1, no. 3, pp. 216–224, 2022.

[10] K. Cornelius, "Betraying blockchain: accountability, transparency and document standards for non-fungible tokens (nfts)," *Information*, vol. 12, no. 9, p. 358, 2021.

[11] J. Gilcrest and A. Carvalho, "Smart contracts: Legal considerations," *International Conference on Big Data (Big Data), IEEE*, pp. 3277–3281, 2018.

[12] C. Dannen, "Introducing ethereum and solidity," *Springer*, vol. 1, 2017.

[13] D. Pramulia and B. Anggorojati, "Implementation and evaluation of blockchain based e-voting system with ethereum and metamask," pp. 18–23, 2020.

[14] A. Mani, S. Verma, and S. Marwaha, "A comprehensive study of nfts," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 4, pp. 1656–1660, 2021.

[15] S. Wang, "Crypto art: Nft art trading and the art market," *Asian Journal of Social Science Studies*, vol. 7, no. 10, p. 14, 2022.

[16] D. P. A. D. Rafli, "Nft become a copyright solution," *Journal of Digital Law and Policy*, vol. 1, no. 2, pp. 43–52, 2022.

[17] E. Lidén, "Potential advantages and disadvantages of nft-applied digital art," 2022.

[18] A. Lavrova and O. Iakushkin, "Nft performance and security review," *Computational Science and Its Applications–ICCSA Workshops: Proceedings, Springer*, pp. 217–228, 2022.

[19] Y. Mhatre, D. Dixit, R. Salunkhe, and S. Sharma, "Challenges of implementing an nft marketplace," *Impact Factor Value*, vol. 7, p. 529, 2008.