# Non-Fungible Tokens (NFTs) with Intellectual Property Rights (IPR) proved by digital signature

Konstantinos I. Papageorgiou

Computer Science and Biomedical Informatics, University of Thessaly, Lamia 35100, Greece.

Correspondence: kopapageorgiou@uth.gr

## Abstract

As the technology advances, more services are embedded in the digital world. Citizens tend to use these digital services to exchange information or to share their digital creations. As the data are increasing rapidly on the internet, it is getting more complicated for the citizens to claim the ownership of their digital creations. Efforts have been made to implement measures to enforce copyright claims by governments, but they are more complicated to implement on the internet. In this work, a decentralized system that enables citizens to create digital evidence for their creations is presented. This system is based on the blockchain technology and allows the citizens to claim the ownership of their digital art, as well as, the verification of who owns a particular digital piece of art. Once the user signs their data and successfully get verified by a smart contract, data will be permantly stored in the blockchain and a token id will be generated and get associated with the account of this particular user. Any user in the blockchain network can search this particular token id and verify its owner.

## Introduction

The concept of blockchain technology has its roots in the fields of cryptography and digital currencies. In 1991, researchers Stuart Haber and Scott Stornetta published a paper titled "How to Time-Stamp a Digital Document," which introduced the idea of using a time-stamped ledger to securely store and validate data [1]. This ledger, which is the underlying structure of a blockchain, is both cryptographic and registry-based, and it is linked internally from one transaction block to the next. Several copies of the ledger are then distributed, and this process does not require a central administrator [2]. Another concept called "electronic cash" or "digital currency" was proposed by David Chaum in the 1980s, and it contributed to the development of blockchain technology. In 1997, Adam Back introduced the concept of "hashcash," which offered a solution for controlling spam emails. This led to the creation of "b-money" by Wei Dai, which was based on a peer-to-peer network. Overall, the development of blockchain technology was influenced by a range of concepts and advances in the fields of cryptography, digital currencies, and peer-to-peer networks. These ideas laid the foundation for the creation of decentralized and secure networks, which are the core features of blockchain technology [1].

In 2009, the cryptocurrency known as Bitcoin was introduced to the world. The concept of Bitcoin was first described in a 2008 paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System," which was authored by an individual or group using the pseudonym Satoshi Nakamoto. The paper proposed a system for direct online payment between two entities without the need for a third-party authority overseeing the transaction. The paper described the use of cryptography to create a digital currency that could not be copied and could not be spent more than once, thus solving the problem of "double spending". The paper also introduced the concept of a public ledger, where transactions involving the digital currency could be traced and verified to ensure that the currency had not been spent previously. A few months after the publication of the paper, an open-source program was released to implement the Bitcoin system, and the first Bitcoin network was launched on January 3, 2009, when Nakamoto created the first bitcoins. Although the identity of the inventor of Bitcoin remains unknown, the cryptocurrency has continued to grow and gain support from a large community of users. This community has helped address various issues with the code and has contributed to the development of the Bitcoin network [2].

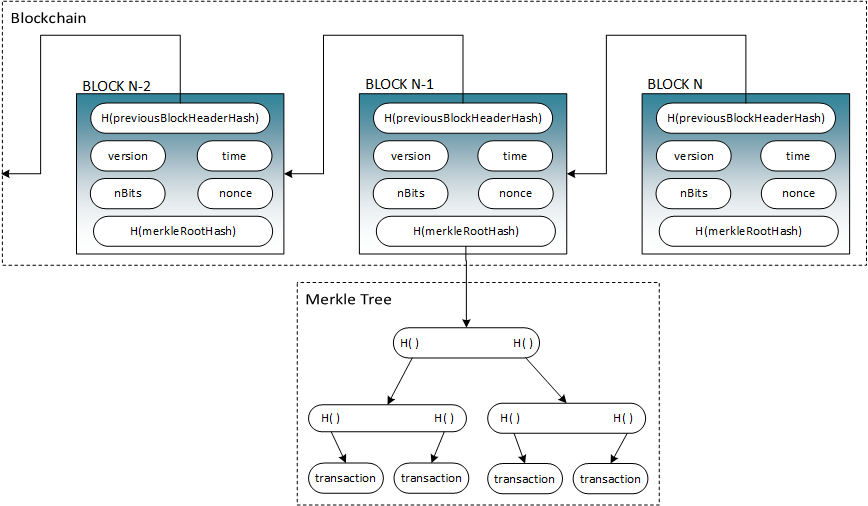


Figure 1: Structure of Bitcoin

Satoshi Nakamoto, the inventor of Bitcoin, took the idea of a distributed ledger from the research of Stuart Haber and Scott Stornetta and added a financial incentive to maintain linked copies of the ledger. Nakamoto's key contribution was the invention of "mining," which allowed people to earn bitcoins by solving mathematical puzzles associated with verifying transactions in a block. This reward for contributing to the development of the blockchain acts as an incentive, making it resistant to counterfeiting in a decentralized environment.

The addition of mining by Nakamoto ushered in a subsequent wave of cryptocurrencies whose progress moved very slowly for several years.

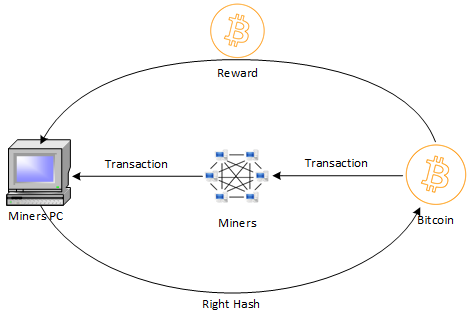


Figure 2: Bitcoin mining

After five years from the first block of the Bitcoin blockchain, a second generation of blockchain named Ethereum was introduced [3]. In 2014, Ethereum blockchain enabled developers to execute smart contracts. Smart contracts, are in essence, programs stored on a blockchain that executes when predetermined conditions are met. Once these conditions are satisfied, a transaction is executed. With the integration of smart contracts on blockchain, developers can create financial applications that use tokens such as cryptocurrencies. As the technology of the distributed ledger advances, more notions have been developed and came to the foreground. In the recent time, anyone holding digital or even physical assets, can convert them into a digital token using the blockchain technology. This cost effective concept simplifies the process of exchanging any value between entities [4].

According to the laws designed for the internet in the decade of 2000, concerning the intellectual property licensing, people who buy digital goods do not actually own them but instead they are simply users [5]. One of the concepts of Ethereum blockchain technology that has been developed, is the ERC-721 (Ethereum Request for Comments) standard for Ethereum-based tokens. The ERC-721 specifies the standard token interface with providing a model implementation for non-fungible tokens (NFT) [6]. NFT can represent the ownership of physical or non-physical assets in a decentralized way [7]. One of the first application of NFTs that made this standard viral, was digital art such as CryptoPunks [8].

In the current work, an application concept is proposed that enables users to create NFTs for their digital art. The digital art is stored in a decentralized way and also, users are able to verify the validity of the owners’ NFTs with the use of cryptographic techniques.

The work is structured as follows: Section II presents an analysis on the blockchain technology. A brief description on consensus mechanisms is included in Section III. In Section IV, the technology of Interplanetary File System (IPFS) is analyzed. Section V presents a description of the design of the proposed solution. In Section VI, the implementation of the solution, as well as, the complete protocol is analyzed. In Section VII, results and future work will be discussed.

## Blockchain technology

Monetary transactions between entities are typically managed by a third party. For example, when two people want to transfer money or make an electronic payment, they need a bank or credit card company to verify the transaction and carry it out. This is not only true in the financial sector, but also in other areas such as music or software distribution. In all these cases, there is a central third party authority that gathers information and uses it to control and manage the transaction [9].

Blockchain is a type of technology that has been developed to remove the need for intermediaries, or third party authorities, in transactions between two entities. This technology creates a decentralized environment in which the processing and confirmation of transactions can take place without the involvement of a central authority. The Blockchain is a distributed database that maintains a constantly growing list of data records, known as blocks, which contain information about transactions that have taken place within the network. This information is recorded in a public ledger, which is tamper-resistant and contains a record of every transaction that has occurred on the network. Because the ledger is distributed across all nodes in the network, it is impossible to alter or delete any transaction once it has been recorded. This makes the blockchain a highly secure and transparent way of conducting transactions. However, there are still some technical challenges and limitations that need to be overcome in order to fully utilize the potential of this technology.

In order to ensure the integrity and security of transactions on the Blockchain, it is crucial to protect and maintain the health of the nodes involved in the verification and execution of these transactions. This is necessary to prevent any potential attacks or attempts to tamper with the transactions on the Blockchain [10]. Additionally, the verification process for transactions on the Blockchain requires significant computing power in order to effectively verify the accuracy and legitimacy of the transactions.

Blockchain is a system that uses digital ledgers, or lists of transactions, that are distributed across a network of computers. These transactions are cryptographically signed, meaning that they are securely encrypted, and are grouped into blocks. Each block is linked to the previous block in the chain through a process called cryptography, which makes it difficult to alter or tamper with the data. Before a block is added to the blockchain, it must be validated and approved by a consensus of users within the network. As more blocks are added to the chain, it becomes increasingly difficult to change the data in older blocks. The ledger is replicated on multiple computers within the network, and any conflicts or discrepancies are automatically resolved using pre-determined rules. Overall, the use of blockchain technology ensures the integrity and security of the data within the digital ledger [11].

There are two general high-level categories for blockchain approaches: permissionless and permissioned. Permissionless blockchains allow anyone to access and participate in the network, while permissioned blockchains restrict access to only certain members of a consortium. Governments and businesses often prefer to use permissioned blockchains because they can control who can participate in the consensus process. Permissioned blockchains can be divided into syndicated and fully private blockchains. In syndicated blockchains, only trusted participants have authority, and the protocol can be modified if a majority of these participants agree. Blockchains can also be customized to fit different situations and needs. For example, in partially decentralized blockchains, the public can only query the data for a limited time. In fully private blockchains, only one or a group of participants can add data, while the public can read it [12].

Table 1: Permissionless and Permissioned Blockchains comparison

|  |  |  |
| --- | --- | --- |
| Properties | Permissionless Blockchain | Permissioned Blockchain |
| Speed | Low performance and high latency | High performance and low latency |
| Peers | High number of peers writing and reading | High number of peers reading, low number of peers writing |
| Consensus | Proof of Work (PoW) or Proof of Stake (PoS)[[1]](#footnote-1) | Practical Byzantine Fault Tolerance Protocols (PBFT), which tolerate malicious peers and trust the consensus of the majority |
| Central authority | No | Yes |
| Privacy | Achieved using cryptographic techniques, but with lower efficiency | Read permissions can be restricted by the central authority, writers and readers can reside on parallel blockchains which are interconnected |
| Verifiability | Observers can verify the status of the blockchain | |
| Availability | High, provided through peer-to-peer replication | |

## Consensus Mechanisms

Consensus mechanisms are algorithms used in blockchain technology to reach agreement on the state of the ledger. These mechanisms allow network participants to validate transactions and ensure the integrity and security of the network. There are several different types of consensus mechanisms, and each has its own unique features and benefits. The most common mechanisms include Proof-of-Work (PoW), Proof-of-Stake (PoS), and Byzantine Fault Tolerance (BFT).

### Proof-of-Work

Proof-of-Work (PoW) is a widely used consensus mechanism in blockchain technology that allows network participants to reach agreement on the state of the ledger. It is a key feature of decentralized systems and is used to secure and validate transactions on the network.

The PoW mechanism is based on the principle of "mining" new blocks of transactions, which involves solving complex mathematical puzzles. Each puzzle is unique and has only one correct solution, and it requires a significant amount of computational power to solve. When a miner successfully solves a puzzle, they can create a new block and add it to the blockchain. The process of mining new blocks is competitive, and miners compete to be the first to find the correct solution and add the new block to the chain. This competition helps to ensure that the blockchain remains secure and that transactions are validated in a fair and transparent manner. The PoW mechanism also has several key features that make it secure and efficient. For example, it is designed to be resistant to attacks and tampering, and it requires a high level of computational power to perform, which makes it difficult for a single entity to control the network. Overall, the PoW mechanism is an important part of the blockchain technology and plays a crucial role in ensuring the security and integrity of decentralized networks [13].

### Proof-of-Stake

Proof-of-Stake (PoS) is another consensus mechanism in blockchain technology. Unlike the Proof-of-Work (PoW) algorithm, which relies on mining new blocks of transactions, PoS uses a different approach to validate transactions and secure the network.

In a PoS system, network participants are required to "stake" a certain amount of their assets (usually the native cryptocurrency of the blockchain) as collateral. This collateral acts as a form of security and is used to incentivize participants to act honestly and validate transactions in a fair and transparent manner. When a new block of transactions is added to the blockchain, the network selects the participants who will validate the transactions in that block based on their staked assets. These participants, known as "validators," are chosen using a random selection process, and they are responsible for verifying the transactions in the new block and adding it to the chain. The PoS algorithm allows for a more decentralized approach to consensus, as the selection of validators is based on the amount of assets staked, rather than the computational power of the miner. It also reduces the energy consumption and costs associated with mining, as there is no need for complex cryptographical problems to be solved. The PoS algorithm is an important part of the blockchain technology and offers an alternative approach to achieving consensus and securing decentralized networks in a less energy consumption way by reducing the computational power [14].

### Delegated Proof-of-Stake

Delegated Proof-of-Stake (DPoS) is a variant of the Proof-of-Stake (PoS) consensus algorithm. It is a fast and efficient approach to achieve consensus on a decentralized network.

In a DPoS system, network participants can delegate their voting power to other participants, known as "witnesses" or "delegates," who are responsible for validating transactions and adding new blocks to the blockchain. These witnesses are selected based on their reputation and the amount of stake they have in the network. The features of DPoS algorithm gives the potentiality to be more efficient and secure. For example, it allows for a egalitarian approach to consensus, as the selection of witnesses is based on the amount of stake they have, rather than their computational power. It also reduces the energy consumption and costs associated with mining, as there is no need for complex mathematical puzzles to be solved. In addition, the DPoS algorithm allows for faster transaction processing and improved scalability, as the witnesses can process transactions in parallel and add multiple blocks to the chain simultaneously. This allows for higher transaction throughput and improved network performance [15].

### Practical Byzantine Fault Tolerance

The Practical Byzantine Fault Tolerance (PBFT) Protocol is a variant of the Proof-of-Stake (PoS) algorithm and is based on the concept of "quorum slicing."

In a PBFT system, the network is divided into multiple slices, and each slice is responsible for validating a portion of the transactions in a new block. This allows for faster transaction processing and improved security, as the network can process transactions in parallel and add multiple blocks to the chain simultaneously. The PBFT is designed to be resistant to attacks and tampering, and it requires a high level of computational power to perform, which makes it difficult for a single entity to control the network. In addition, the PBFT protocol allows for a more decentralized and egalitarian approach to consensus, as the selection of validators is based on the amount of stake they have, rather than their computational power. This helps to ensure that the network remains secure and that transactions are validated in a fair and transparent manner [16].

## Interplanetary File System

Interplanetary File System (IPFS) is a peer-to-peer file sharing system that aims to revolutionize the way information is distributed across the globe. It is a combination of distributed systems, communication protocols, and file system technologies. Communication protocols are the set of rules that govern electronic transactions between connected users, and IPFS uses a combination of client-server and HTTP web-based models to facilitate P2P file sharing. The system consists of several layers, each of which is responsible for a specific function. IPFS combines a range of novel and existing approaches to improve the efficiency and reliability of information distribution. Its main component is discussed below:

Distributed Hash Table (DHT) is a type of data storage system that uses a hash table to efficiently coordinate and distribute information over a network of computers. The main advantages of DHTs are their scalability, decentralization, and fault tolerance. Unlike traditional systems that require central coordination, DHTs can function reliably even after node failure. They are also able to accommodate a large number of nodes. DHTs have a wide range of potential applications, including electronic file security systems in sectors such as banking, legal, education, and healthcare. They can provide a secure and efficient way to store and access information in decentralized networks [17].

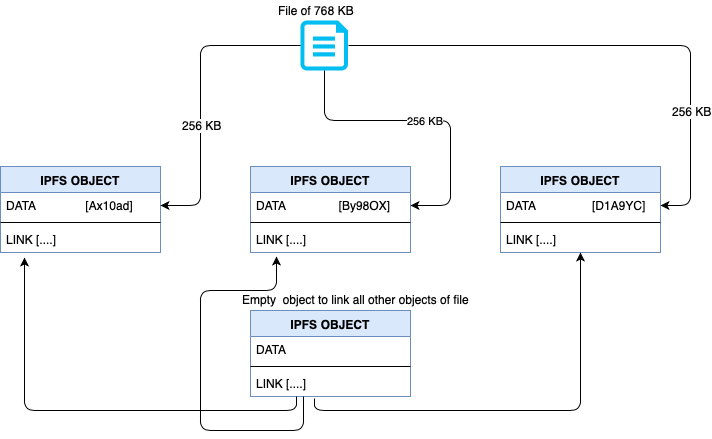


Figure 3: How IPFS works

## Application design

In this section a concise description of the solution and miscellaneous processes that take place are given.

As discussed in the previous sections, any digital or physical asset can be converted into tokens. In order to simulate these assets, the solution developed for a use case scenario where these assets are digital art creations. This digital art is randomly generate different parts of the face (such as hair, ears, nose, eyes, skin color etc.) and combine them in order to create a picture.

The solution is a blockchain-based application. The main concept of the application is based on the idea that people can establish their own Intellectual Property Rights (IPR) for their digital art. Once a user convert his digital art into a Non-Fungible Token, it cannot be refused or refuted by anyone. This is due to the fact that the process of creating this NFT is a robust process that is based on a pre-arranged rules contained in a smart contract, which is deployed to the Ethereum blockchain network.

To add a more decentralized approach, the digital art is stored in the IPFS node of the user. In this way, the digital art is not in a server where the data are accumulated and a potential attack could harm them. Anyone using the application, can request to see any NFT created in this smart contract by typing the token id of the NFT. If a user requests to see an NFT, the digital art, as also the metadata (creator, title of the digital art, description) of the digital art will be presented.

### Creating NFT

The main steps of creating a Non-Fungible Token are depicted in the Figure 4:

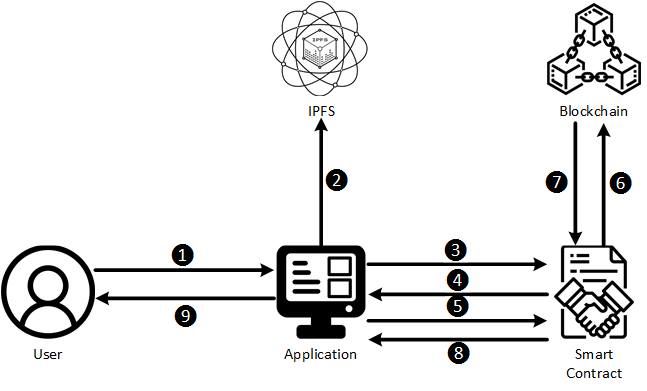


Figure 4: NFT creation

For NFT creation:

1. User generates a digital art and fills out the metadata information
2. Application uploads the generated digital art to the IPFS node and get the link pointing to the file on the IPFS node, in order to insert it in the metadata.
3. Application makes a request to the smart contract, to get the current token id.
4. When the token id is received from the smart contract, it is automatically inserted in the metadata. After the generation of the metadata, application signs them with the web3 account of the user.
5. Application send the metadata and the signature to the smart contract in order to be verified.
6. After the successful verification, the smart contract stores the metadata to the blockchain.
7. Smart contract gets the result of the metadata store.
8. If the transaction is successful, the smart contract sends the transaction receipt back to the application. If the transaction is unsuccessful, it sends an execution revert exception in order to be handled properly by the application.
9. Application shows the results to the user.

### Retrieving NFT

A user can also retrieve an NFT with the steps depicted in Figure 5:

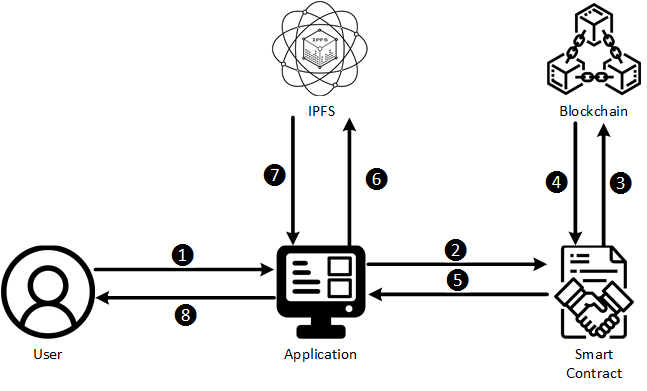


Figure 5: NFT Retrieval

For NFT retrieval:

1. User types the token id.
2. Application makes a request to the smart contract for this particular token id.
3. If the token id exists, it requests from the blockchain the metadata of this particular token id.
4. Smart contract receives the metadata from the blockchain.
5. When the smart contract receive the metadata, it sends them to the application. If the token id does not exist, it sends an execution revert exception in order to be handled properly by the application.
6. When the application receives the metadata, it extracts the link that refers to the digital art of the token id entered by the user, and makes a request to get the digital art from the IPFS.
7. The application receives the digital work from the IPFS.
8. Application presents the NFT of the given token id which consists of the digital art and the NFT’s metadata (creator, title of the digital art, description of the digital art).

## Implementation of the application

The proposed solution is based on four components, a blockchain network, a smart contract, the user’s application that is developed in python and the IPFS daemon.

* **Blockchain Network:** The current application uses an Ethereum blockchain test network for deploying the smart contract that creates NFTs. This is a private test network.
* **Smart Contract:** The smart contract of the application is the main component responsible for creating NFTs based on the ERC-721 standard. In order to generate NFT, it also verifies the signature of the data that are going to be converted into Non-Fungible Tokens. In this current case, the smart contract has been developed in Brownie suite and compiled with Solidity 0.8.17. It is consisted from methods that verifies web3 signatures, writes data to the blockchain as also reading data from it.
* **User’s application:** This application is used in order to interact with the smart contract and IPFS. It is developed in Python 3.10. The application calls the methods of the smart contract in order to generate or to receive NFTs.
* **IPFS daemon:** It is an executable file that sets up an IPFS node in order to communicate with the IPFS network and execute some actions such as storing files or receiving files from a node.

### Generating/Loading Web3 account

In order for the users to use the application, they must create or load their ethereum account. In Ethereum, each account has a 20 byte address, and state transitions are transfers of value and information between accounts. An Ethereum account is associated with four fields:

* The **nonce**, is a serial counter and is used to ensure that each transaction on the network can only be performed once.
* The **current ether balance** in the account.
* The **contract code**, if any.
* The **account storage**.

The ether is the main digital currency traded on the Ethereum blockchain. It is used to pay for transaction fees on the network. Ethereum has two types of accounts: externally owned accounts, which are controlled by private keys, and contract accounts, which are controlled by their contract code. Accounts owned by external owners have no code associated with them, and users can send transactions from these accounts by creating and signing a transaction. Contract accounts, on the other hand, are activated whenever they receive a message, allowing them to read and write data to their internal storage, send messages or create new contracts [3].

To create an Ethereum account, BIP-39 protocol is used. BIP39, or Bitcoin Improvement Proposal 39, is a standard for generating a set of words (also known as a mnemonic phrase) that can be used as a backup to recover a cryptocurrency wallet. These mnemonic phrases are typically used in combination with a seed password to create a binary seed that is used to generate the encryption keys for the wallet. Having a standardized method for generating mnemonic phrases and creating seeds makes it easier for users to manage their cryptocurrency wallets and ensures that different wallets are compatible with one another.

For this use case scenario, methods of creating or loading accounts from Brownie suite have been used.

### Creating NFT

In order for an NFT to be generated, the following protocol depicted in Figure 6, is been used.

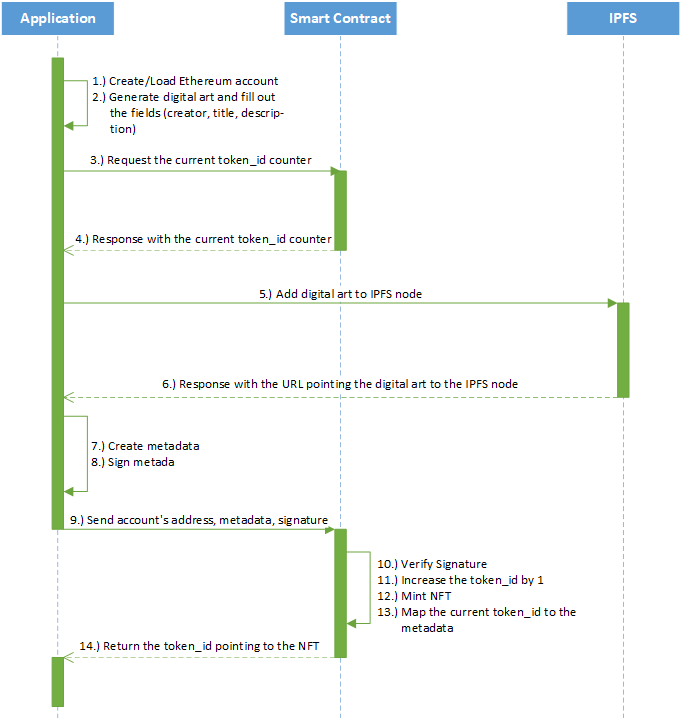


Figure 6: NFT creation protocol

User create or load their Ethereum account in order to sign messages (data) and execute transactions (step 1). Then, user can generate their piece of digital art by generating a random image. Also user is required to fill some specification about their digital art, such as creator’s name, the title and the description of the digital art (step 2). In step 3, the application automatically requests the current token\_id counter. When the application receives the token\_id (step 4), it automatically adds the digital art to the IPFS node (step 5) and receives the URL that points to the digital art on the IPFS node (step 6). The application collects the details of the digital art, the received URL of the IPFS node and the token\_id received from the smart contract **increased by one**, in order to create the metadata (step 7).

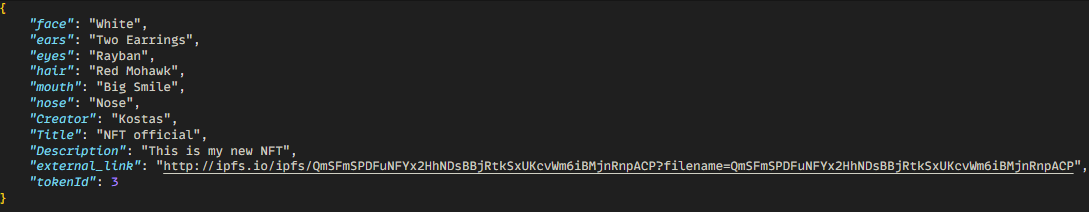


Figure 7: Metadata

When the metadata are generated, user must sign them (step 8).

In order for the signature to be created, the metadata are getting converted in string format. After, the metadata string is converted to bytes using a method called “encodePacked”. EncodePacked is using only the minimal required memory to encode data. The result of this method is hashed with the “keccak256” method. Keccak-256 is cryptographic function, part of Solidity (SHA-3 Family). This function computes the hash of an input to a fixed-length output, yielding a singular 32-byte hash from any number of inputs. This cryptographic hash function can only be used in one direction and cannot be reversed. The result of this method with an extra string: **"\x19Ethereum Signed Message:\n32"** are used as an input again in the encodePacked method and the output is hashed again with the “keccak256” method. The final hash result is signed with the private key of the user’s Ethereum account.

After the successful generation of the signature, application calls the “getNFT” method from the smart contract with the arguments (step 9):

* User’s Ethereum account address
* Signature
* Metadata in string format

In step 10, the signature verification is being performed in the smart contract. The metadata string is hashed with the same process mentioned above. The signature that passed as an argument is split in three values using assembly [18] in order to call the pre-built method “ecrecover”. The “ecrecover” takes the hashed metadata (created by the smart contract) and the three values generated by the signature and returns as an output an Ethereum address. If the returned address is equal to the address passed as an argument, then the signature is valid. The “token\_id” counter is inceased by one in order to mint the new NFT (step 11, 12). After calling the mint method, the smart contract maps the “token\_id” to the metadata string passed as an argument (step 13). Finally the smart contract returns the receipt of the transaction with the new “token\_id” value to the application (step 14).

### Retrieving NFT

In order for an NFT to be generated, the following protocol depicted in Figure 8, is been used.

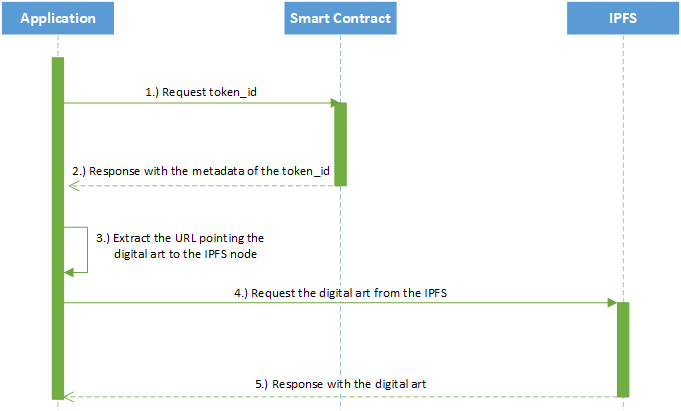


Figure 8: NFT retrieval protocol

Application requests the “token\_id” in which the user is interested to retrieve the NFT (step 1). The smart contract accesses the mapping of the “token\_ids” and pulls the metadata for this particular id in order to send them back to the application (step 2). In step 3, application automatically extracts the URL from the metadata and requests from the IPFS the digital art corresponding to this particular URL (step 4). When the application retrieves the digital art from IPFS (step 5), it represents the result (digital art, creator’s name, title and description of the digital art) to the user.

## Conclusion

The aim of this work was to create an application that converts digital assets, such as digital art into tokens, and proving the ownership of them with the use of the digital signature that utilizes cryptographic techniques. The proposed solution utilizes the blockchain technology and its tools to provide convenience on proving ownership in the context of Intellectual Property Rights (IPR).

The application is in an early stage but very flexible in order to integrate additional characteristics.

A possible addendum would be the support of converting more types of digital assets into NFT. Also, an interesting characteristic for digital artists would be the ability to create NFT for an entire collection of their digital art. Finally, the tokenization of the physical assets would be a characteristic that enables the users to prove their ownership of their physical assets and relieves them from the bureaucracy.

## References

[1] S. S. Sarmah, ‘Understanding Blockchain Technology’, p. 8.

[2] A. Whitaker, ‘Art and Blockchain: A Primer, History, and Taxonomy of Blockchain Use Cases in the Arts’, *Artivate*, vol. 8, no. 2, pp. 21–46, 2019, doi: 10.1353/artv.2019.0008.

[3] V. Buterin, ‘A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM’, p. 36, 2014.

[4] S. M. H. Bamakan, N. Nezhadsistani, O. Bodaghi, and Q. Qu, ‘A Decentralized Framework for Patents and Intellectual Property as NFT in Blockchain Networks’, In Review, preprint, Oct. 2021. doi: 10.21203/rs.3.rs-951089/v1.

[5] J. A. T. Fairfield, ‘Tokenized: The Law of Non-Fungible Tokens and Unique Digital Property’, *Indiana Law J.*, vol. 97, p. 1261, 2022.

[6] S. Casale-Brunet, P. Ribeca, P. Doyle, and M. Mattavelli, ‘Networks of Ethereum Non-Fungible Tokens: A graph-based analysis of the ERC-721 ecosystem’, in *2021 IEEE International Conference on Blockchain (Blockchain)*, Sep. 2021, pp. 188–195. doi: 10.1109/Blockchain53845.2021.00033.

[7] I. D. de Figueiredo Novo, ‘Property-based testing of ERC-721 Ethereum smart contracts’.

[8] E. Lee, ‘The Cryptic Case of the CryptoPunks Licenses: The Mystery Over the Licenses for CryptoPunks NFTs’, *Available SSRN*, 2021.

[9] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, ‘Where Is Current Research on Blockchain Technology?—A Systematic Review’, *PLOS ONE*, vol. 11, no. 10, p. e0163477, Oct. 2016, doi: 10.1371/journal.pone.0163477.

[10] M. Swan, *Blockchain: Blueprint for a New Economy*. O’Reilly Media, Inc., 2015.

[11] D. Yaga, P. Mell, N. Roby, and K. Scarfone, ‘Blockchain Technology Overview’, Oct. 2018. doi: 10.6028/NIST.IR.8202.

[12] J. Chen, Y. Ruan, L. Guo, and H. Lu, ‘BCVehis: A Blockchain-Based Service Prototype of Vehicle History Tracking for Used-Car Trades in China’, *IEEE Access*, vol. 8, pp. 214842–214851, 2020, doi: 10.1109/ACCESS.2020.3040229.

[13] A. Bracciali, J. Clark, F. Pintore, P. B. Rønne, and M. Sala, Eds., *Financial Cryptography and Data Security: FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers*, vol. 11599. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-43725-1.

[14] P. Gaži, A. Kiayias, and D. Zindros, ‘Proof-of-Stake Sidechains’, in *2019 IEEE Symposium on Security and Privacy (SP)*, Feb. 2019, pp. 139–156. doi: 10.1109/SP.2019.00040.

[15] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, ‘Delegated Proof of Stake With Downgrade: A Secure and Efficient Blockchain Consensus Algorithm With Downgrade Mechanism’, *IEEE Access*, vol. 7, pp. 118541–118555, 2019, doi: 10.1109/ACCESS.2019.2935149.

[16] X. Hao, L. Yu, L. Zhiqiang, L. Zhen, and G. Dawu, ‘Dynamic Practical Byzantine Fault Tolerance’, in *2018 IEEE Conference on Communications and Network Security (CNS)*, Feb. 2018, pp. 1–8. doi: 10.1109/CNS.2018.8433150.

[17] H. Sharma, M. K. Gupta, G. S. Tomar, and W. Lipo, Eds., *Communication and Intelligent Systems: Proceedings of ICCIS 2020*, vol. 204. Singapore: Springer Singapore, 2021. doi: 10.1007/978-981-16-1089-9.

[18] J. Katz and H. Shacham, Eds., *Advances in Cryptology – CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part II*, vol. 10402. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-63715-0.

1. Consensus mechanisms will be discussed in the next subchapters. [↑](#footnote-ref-1)