Introduction to NFTs: The Future of Digital Collectibles

Muddasar Ali, Sikha Bagui
Department of Computer Science
University of West Florida, Pensacola, FL 32514

Abstract—**This paper commences by introducing the essentials of blockchain technology and then goes into how Ethereum blockchain revolutionized blockchain. Smart contracts are presented in the context of showing how they play an important role in implementing rules regarding the Ethereum blockchain, allowing the user to regulate digital assets. The standards used in the Ethereum blockchain to build Non-Fungible Tokens (NFTs) are discussed. The paper concludes by presenting the benefits of NFTs as well as the use of Ethereum blockchain for future applications.**

***Keywords*—Blockchain technologies; smart contracts; cryptocurrencies; ethereum; non-fungible tokens (NFTs)**

I. Introduction: What is Blockchain Technology?

Blockchain technology, considered to be one of the most secure technologies to keep data safe, is constructed on blocks of data. Blockchain technology is fast, public, cheap, easy to use, transparent and programmable technology that allows for the transfer of information and/or financial assets instantaneously from one part of the world to another. It is a protocol that governs the rules and regulations for value exchange. Blockchain technology can contain any type of information and the link between each block and the next is called a chain because the blocks are connected in a way that cannot be altered [1]. Blockchain also enables users to safely transfer values globally. The safety comes from the fact that once the information is recorded in the blockchain, it becomes infeasible to alter it. This feature is called immutability [2]. This is especially important for storing bank records or transactions. The safety of this technology is based on the characteristics of the chains between the blocks. A new block is added to the chain by calculating the hash of the previous block and using that as part of the data for the next block [3].

To understand how the blockchain technology works, one needs to first understand what is stored in a single block. Each block contains three types of information, the data, its hash, and a pointer to the hash of the previous block. And each application of blockchain technology can have its own type of data that is completely different from all other applications. For example, bitcoin stores information about the transactions of bitcoins, such as the sender, the receiver, and the amount of bitcoin that is transferred. The hash of the block is a unique identifier for the block. Each block's hash will be different based on the data that is stored in it, so if the data gets changed the hash also changes. The addition of the previous block's hash is the main reason behind the success of blockchain technology and this is also why the alteration of data stored in the blockchain becomes infeasible. And this is what makes it reliable to store any kind of information that needs to be kept safe. If one wants to alter the blockchain by changing the information stored in a block, this will also lead to a change in the hash of the block, which will consequently not match with the hash that is in its next block, causing a chain reaction. So, any alteration in a single block will invalidate the whole blockchain [3].

Even though blockchain technology is not totally new, its implementations are still in their early stages, and it is for this reason that many people struggle to understand the real use of blockchain technology. For some time, its implementation seemed limited to cryptocurrencies but there is much more to where this technology can be applied and in today's world. We constantly see new applications based on blockchain technology and we will get more familiar with them as they become part of our daily lives.

Blockchain technology has the potential to upgrade any system that requires a third-party regulatory authority that serves to define the authenticity of any changes that will take place in the system. More specifically, the technology can replace the authority decentralizing the authenticity process to every member that is present in the organization. One of the most widely known applications of blockchain technology is proposed by the Bitcoin whitepaper published in 2008 [4]. This work introduced the concept of cryptocurrencies or electronic cash that would be transferred without going through a financial institution. A digital ledger was introduced that would be distributed to every node present in the blockchain to keep track of all the transactions that were taking place in an organization. The development of bitcoin has radically changed the concept of money and currency [4]. Bitcoin is the first-ever form of a digital asset or money that not only has no backing or intrinsic value but also no centralized issuer or controller. It is decentralized. While its introduction is proof of the potential of blockchain technology, the fact that it is used as a distributed tool has attracted even more attention and has inspired more applications of blockchain technology. Colored coins [5] also use blockchain digital assets to represent custom currencies and financial instruments; Smart Property represents the ownership of an underlying physical device, NameCoin [6]. NameCoin also represents non-fungible assets such as domain names. Complex applications rely on digital currency or assets as being directly controlled by pre-programmed code with functionalities, enabling the implementation of Smart Contracts [7]. Hence a turning point in blockchain technology took place with the introduction of Ethereum in 2014 [8]. This is where the nodes were not only keeping track of the transactions, but they could also program and execute software, hence taking the name of smart contracts. Being able to write specific instructions atop Ethereum's digital protocol makes it more resourceful for many more applications.

With this introduction to blockchain technologies, next this paper goes into how Ethereum blockchain revolutionized blockchain. Smart contracts are presented in the context of showing how they play an important role in implementing rules regarding the Ethereum blockchain, allowing the user to regulate digital assets. The standards used in the Ethereum blockchain to build Non-Fungible Tokens (NFTs) are discussed. The paper concludes by presenting the benefits of NFTs as well as the use of Ethereum blockchain for future applications.

II. Ethereum Blockchain

Transactions in bitcoin are based on who is moving money to whom and how much. But it is also important to know how much money is present in the network. For this, there must be a database storing how much money can be spent and how much money can be moved. There must also be a centralized environment with centralized authority that has access to this database where the amounts are registered. But what if we want a decentralized environment, which is the basic spirit behind blockchain technologies and more specifically, is the most important factor that made bitcoin blockchain famous [bitcoin-cash]. Nakamoto's solution [4] is the most basic and practical solution in this open permission-less context. To extend the concept of decentralization and make it accessible to any type of application, the Ethereum blockchain provides simple features. Ethereum, by its founder and inventor Vitalik Buterin [8], is described to be based on a concept of crypto-economics. Ethereum is a combination of major concepts of cryptography and economics. Concepts like hashing and digital signatures used in cryptographic algorithms, and the economic incentives that keep systems like bitcoins going, are used to create decentralized networks with memory [8]. This means that not only the network but also the database will be decentralized. This was considered a breakthrough in blockchain technology as it helped to break new ground in the areas of application of the technology [9]. This was a broader view of how blockchain could be applied, greatly improving the concept of peer-to-peer digital currency transactions. Many different applications were introduced. The first one was NameCoin [6] which was based on bitcoin technology and provided a peer-to-peer decentralized DNS service.

Using this new concept, many different applications can be decentralized. Many other digital assets can be created. But a problem arises when we want to apply this concept for each digital asset and create a different system for every type of service to be developed on the blockchain. This will end up in there being too many services running separately and keeping track of all of them would become tedious; hence the idea behind creating Ethereum. Ethereum is a single general purpose blockchain that can have different types of applications running on it [8]. This blockchain will not have specific rules for transactions for each type of application but will have general rules and access to a general-purpose programming language. Hence Ethereum gives the freedom to build and run applications on the blockchain. Though Ethereum offers this flexibility where a broad range of applications can be built, the main categories of applications still remain related to currencies, mainly due to the fact that the main aspects of a blockchain are decentralization and shared memory, making it suitable for cryptocurrencies.

An Ethereum blockchain state is composed of accounts, which actually are objects. These accounts have a 20-byte address. The blockchain state transitions when there are direct transfers of value and information between accounts or objects. An Ethereum account has four parts:

- The nonce. A nonce is a counter set to make sure each transaction can only be processed once.
- The current balance of the ether account.
- The contract code of the account.
- The storage of the account, which is empty by default.

Ether is used to pay, and is the key crypto-fuel or cryptocurrency of Ethereum. It is used to pay transactional or operational fees. In general, there are two account types: externally owned accounts and contract accounts. Externally owned accounts, which have no code, are controlled by private keys, and contract accounts play by the rules of their contract code. Messages (or instructions) can be sent from externally owned accounts by creating transactions and signing them. Contract accounts, however, are used to send messages or instructions and/or create contracts every time the contract account receives a message. Also, when messages or instructions are received in contract accounts, internal storage can be read from and written to.

III. Smart Contracts

The concept of Smart Contracts was introduced [7] in the 1990s. This is another category of applications that will suit blockchain technology. To understand the concept behind smart contracts we can make an analogy to a vending machine. A vending machine is a device implemented in hardware that implements conditions of an agreement. In this case, a simple condition for this agreement would be that a user puts an amount of money in the vending machine and a related item comes out. If the user does not insert adequate money, the item will not come out. The vending machine is encoded with a set of rules that also keeps the items secured. The security is proportional to the value of the items that the vending machine holds. The vending machine is a very simple example of a contract with a bearer. In this case, anybody with coins in the right currency can participate in an exchange with the vendor, the vending machine. And, the lockbox and other security mechanisms (including programmed security mechanisms) protect the stored coins and contents from attackers.

This concept is applied to digital assets, but with broader rules and more complicated types of contracts. Smart contracts provide stronger and more secure verification for access to the property. Another example is digital security systems for automobiles. The protocol defines the conditions to give authority of the property based on who is the rightful owner at a certain time. Considering cryptographic keys as means to access a car, we can determine conditions that will make the owner change, and therefore change the conditions in which the car can be used. A smart lien protocol can be created, described as follows: if the owner fails to make the scheduled payments, the smart contract invokes the lien protocol. This would give back authority of the car keys to the bank. A further step would probably remove the lien in the event the loan gets paid off [7]. A graphical representation of this example can be seen in Fig. 1.

To keep digital assets secure, a computer program that has direct control of them is put into service. This means that there is not a person that will control the asset but a computer that follows a digital contract that provides specific rules to handle it, and it cannot be controlled by any user. The decisions will be made directly by the computer program. Once again, the smart contract should include conditions that are strong enough to keep the asset secure, in proportion to the importance of the digital asset. This specific environment can be implemented on the Ethereum blockchain. Once an amount of ether is sent to a computer program as a digital asset, and this follows a smart contract, it can then handle the money as specified in the contract. So, if the contract specifies that a certain amount of money will go to an address, the money will move as per the conditions of the contract and no human interaction will take place. The concept of smart contracts can be extended to any other application which is based on a self-executing financial contract such as insurance. Here a set of rules would be specified and when the condition is met an asset would be unlocked for the user.

Smart contracts have been used in the area of crowdfunding where the contract defines certain conditions before letting the money go to a certain project that people are supporting. For example, if a developer requires a certain amount of money to kickstart his/her project, the money will be managed by a computer program that will follow the rules given by the smart contract. In this case, no user can intervene in the process to move any asset, and everything is managed by the program. This transparency gives all the members that are participating in crowdfunding more trust in the process.

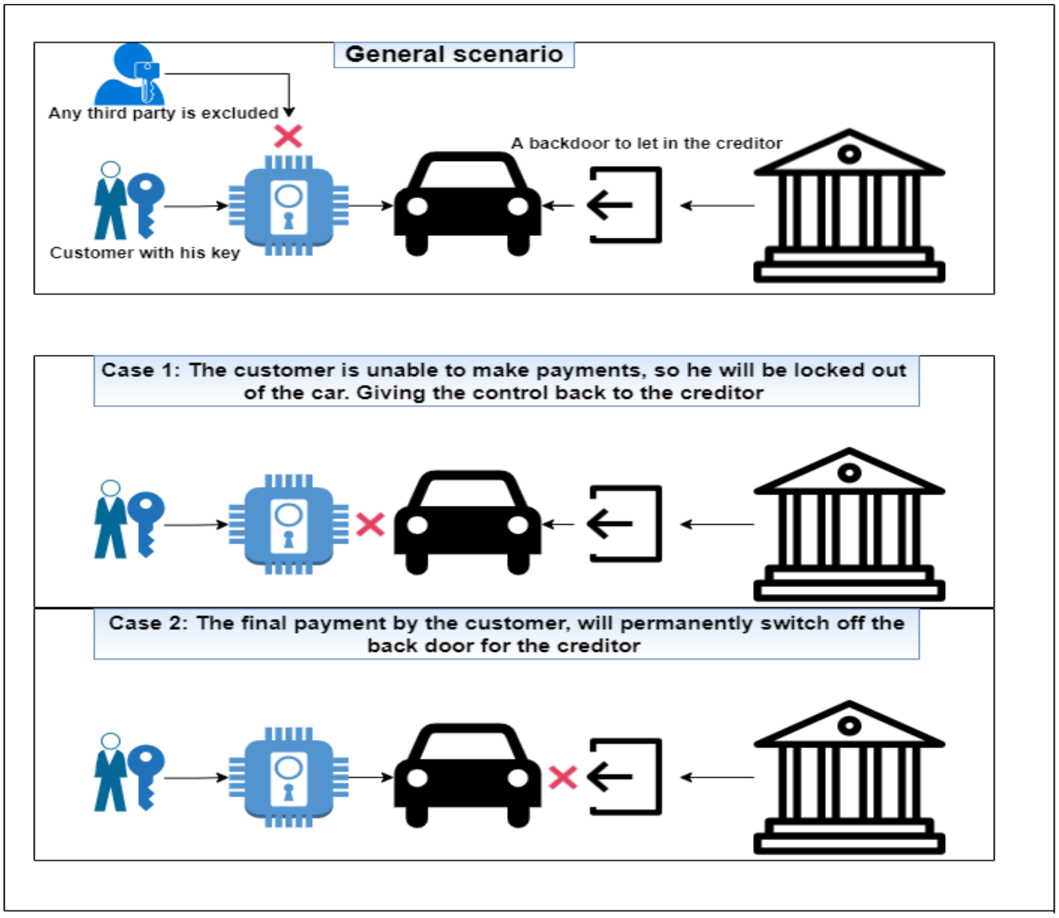
General scenario
Any third party is excluded
A backdoor to let in the creditor
Customer with his key
Case 1: The customer is unable to make payments, so he will be locked out of the car. Giving the control back to the creditor
Case 2: The final payment by the customer, will permanently switch off the back door for the creditor

Fig. 1. Hypothetical Digital Security System for Automobiles that Relies on a Smart Contract.

IV. Ethereum Standards

As presented above, smart contracts are basically pre-programmed or intelligent programs that get executed when certain conditions are met or certain events happen. Since the process is automated, they have no downtime. They also allow for the creation of decentralized applications [9].

The Ethereum Virtual Machine (EVM), which runs on an Ethereum network, simplifies the process of building blockchain applications. EVM allows developers of each application to build new blockchains from scratch. And, developers do not have to use any specific programming language. This allows Ethereum to technically build the first decentralized world computer on a public blockchain [9].

The native currency for Ethereum is Ether (ETH). The platform is mainly maintained through transactional or operational fees. One of the most useful advantages of Ethereum is its ability to create unique tokens, with various functionalities, and operate on the Ethereum blockchain. These tokens have multiple utility-like purposes.

Ethereum Request for Comments (ERC) are application level standards for Ethereum. ERCs include token standards, name registries, library/package formats, and other features. An ERC can be created by anyone, but it falls on the creator to clearly explain the standard. This will also help the creator gain support from within the community. There are common ERC standards for functions of each token type, allowing applications and smart contracts to interact with them in pre-determined or pre-planned ways. ERC-20, which allows for the simple creation, use, and exchange of Ethereum-based tokens, is the most common ERC standard used to date.

A. Explanation of Terms

Ethereum accounts, identified by 20-byte addresses, have externally owned user accounts and contract accounts. Users can create accounts, call or access accounts, or issue transactions (which are signed data packages) that transfer value to users of other accounts and contracts. All transactions are recorded on the Ethereum blockchain. A contract gets activated by a user transaction or by a message or instruction from another contract. Messages only exist in the execution environment of the Ethereum Virtual Machine (EVM), hence are not recorded on the Ethereum blockchain. This is because messages are consequences of an initial transaction. Their existence is limited to the execution environment of the Ethereum Virtual Machine (EVM) and this can be seen in the execution trace logs and potential state changes.

Abstract Binary Interface (ABI) [10]: Most Ethereum contracts follow the ABI standard. The ABI standard identifies functions by signatures. These signatures are made of the first four bytes of the Keccak-256 hash of the function name plus parameter types. The presence or absence of a function in a contract is checked by locating the corresponding 4-bytes hash in the deployed bytecode. This also allows the compliance of a contract with interface standards to be determined by its bytecode.

B. Functionalities Available for Token Contracts

There are presently several functionalities available for token contracts: (i) keeping track of token holdings; (ii) transferring ownership of token contracts. The transfer of ownership is shown in the logs. A safe transfer is where the information is known. Hence, after approval, tokens are taken or withdrawn from the known address, as opposed to being sent or transferred to an unknown address where they may be lost. So, in a safe transfer, both the sender and receiver would be known to the sender and receiver respectively; (iii) the creation of tokens and destruction of tokens. This is also called minting or burning of tokens; (iv) the distribution of tokens. This includes trading (e.g., via ICOs and airdrops) of tokens; (v) token contracts are also programmed to check for authentication and have various roles built into the system. Roles include roles that control the system, like pause or lock, and information provisions like view functions; and (vi) other functionalities including utilities.

C. Token Standards

There are several accepted token standards which are determined by the community that establishes standard interfaces for tokens. The programming language used is Solidity. This programming language is prevalent on Ethereum. The standards that have been accepted so far are:

- ERC-20 Token Standard [11]: This is the most widely used and most general token standard. This standard is programmed to provide very basic functionality like ability to transfer tokens. This standard also allows tokens to be approved and disapproved. This means that they can be spent by another on-chain third party. ERC-20 has six mandatory and three optional functionalities as well as two events to be implemented by a conforming API.

- ERC-721 Non-Fungible Token Standard [12]: In ERC-721, each token is distinct and non-fungible, hence this allows for the tracking of unique assets. Each asset has individual ownership and is atomically tracked. ERC-721 requires each token to have ten functions that are considered mandatory and three events.

- The ERC-777 Token Standard [13]: ERC-777 provides advanced functionalities for interaction with tokens while keeping compatibility with ERC-20. It’s advanced functions include operations to send tokens on behalf of other addresses and ways or hooks for sending and receiving. This allows token holders more control over their tokens. ERC-777 requires each token to have 13 functions that can be considered mandatory and five events.

- ERC-1155 Multi Token Standard [14]: ERC-1155 is considered a multi-token standard since any combination of fungible and non-fungible tokens can be managed in a single contract with this token standard. Multiple token types can also be transferred at once. ERC-1155 requires each token to have six functions that can be considered mandatory functions and four events.

D. Proposed Security Token Standards

In addition to the accepted standards discussed in the last section, others token standards are being discussed. ERC-1462 is more general, while ERC-1450 and ERC-1644 are project oriented.

- ERC-1462 [15] Base Security Token. Though ERC-1462 this is similar to ERC-20, it is an extension to ERC-20. ERC-1462 provides compliance with securities regulations and legal enforceability. Though it is mainly used for general use cases, added functionality and limitations as pertaining to particular projects or markets can be enforced effectively. ERC-1426 also includes KYC (Know Your Customer) and AML (Anti Money Laundering) regulations and the ability to lock tokens for an account and restrict them from transfer due to a legal dispute. An added benefit is that this standard also allows for the attachment of documents to tokens. ERC-1426 requires it's compliant tokens to implement four mandatory checking functions while still implementing ERC-20 standards functions. It also gives the opportunity of two optional documentation functions.

- ERC-1450 LDGR Token [16]. ERC-1450, which also extends ERC20, is a security token for issuing and trading SEC-compliant securities. The ERC-1450 standard allows for the recording of ownership and transfer of securities sold in compliance with the Securities Act Regulations CF, D and A. ERC-1450 has mandatory functions of its own, and it requires some optional parts of ERC-20 to be mandatory. It also requires certain modifier and constructor arguments to be set up and implemented.

- ERC-1644 Controller Token Operation Standard [ERC-1644]. Sometimes the issuer (or an entity delegated to by the issuer) or owner, may have to have the ability to force transfer tokens. ERC-1644 allows a token to clearly declare whether or not a controller can unilaterally transfer tokens between addresses. The ERC-1644 standard requires compliant tokens to have three mandatory functions and two events.

E. ERC-20 Standard

The ERC-20 standard was the first to lay down the concept of dealing with tokens and their implementations. A token can virtually represent anything in Ethereum, from lottery tickets to an ounce of gold or financial assets like a share in a company. This concept has been used to re-invent the concept of crowdfunding, introducing the Initial Coin Offerings (ICOs) [17] that are equivalent to Initial Public Offerings (IPOs) but for companies looking to raise money to create a new coin, app, or service. ICOs are tokens that are bought by people that agree to invest in a company, whereas IPO investors buy stocks of a company. Having a powerful feature that can represent this variety of valuable elements comes along with the necessity of having a robust standard that is also capable of handling all the assets in a proper way. The ERC-20 standard allows for the implementation of a standard API for tokens within smart contracts. This standard provides basic functionality to transfer tokens, as well as allows tokens to be approved so they can be spent by another on-chain third party. The implementation of this standard also provides a standard interface to allow any tokens on Ethereum to be re-used by other applications.

ERC-20 also introduces a standard for Fungible Tokens, which means that they have a property that makes each token exactly the same in type and value as another token [11]. The ERC-20 standard is one of the most important standards that led to the concept of NFTs.

If a Smart Contract implements the methods and events provided by the ERC-20 standard, it can be called an ERC20 Token Contract and once deployed it will be responsible to keep track of the created tokens on Ethereum.

F. ERC-721: The Standard for Non-fungible Tokens

The ERC-721 (Ethereum Request for Comments 721), proposed by [12], is a Non-Fungible Token Standard. ERC-721 has an API for tokens within Smart Contracts. This is a free open standard that allows for the building of non-Fungible or unique tokens on the Ethereum blockchain [12]. ERC-721 introduces an extension to the common interface for tokens by additional functions, which led to having unique tokens, thus non-fungible. These unique tokens took the name of NFTs [18]. The fundamental characteristic of NFTs is the uniqueness, which means they cannot be exchanged with like-for-like items, making it the most suitable way to identify something or someone that is not substitutable. This type of token can be used on platforms that offer collectibles, access keys, lottery tickets, numbered seats for concerts or sports matches, etc. Once one comes in possession of any of these unique items, he/she can use these NFTs to represent ownership. Uniqueness is guaranteed by NFTs because there is only one token that has some specific characteristic, and they are completely different from any other NFT that is present in the market. Furthermore, there is only one official owner at a time and the ownership is secured by the Ethereum blockchain, which guarantees that no one can modify the record of ownership or copy/paste a new NFT into existence.

Every NFT is essentially a decentralized application, and its properties can be summarized as follows [9].

- Verifiability: This means that the NFT's token metadata as well as ownership can be publicly verified.

- Transparent Execution: This means that all activities on NFTs including minting, selling, and purchasing are publicly available.

- Availability: This means that NFTs are always available to be sold and bought. The system for NFTs never goes down.

- Tamper-resistance: This means that trading records related to NFTs are persistently stored and cannot be changed or manipulated in any way after the transactions are confirmed.

- Usability: This means that each NFT has its most recent ownership information, and this information is clear and user friendly.

- Atomicity: The process of trading NFTs is atomic.
- Tradability: Every NFT can be traded and exchanged.

In other words, each token created or minted has a unique identifier that is not interchangeable with any other token and has only one owner which is easily verifiable. If the owner decides to sell an NFT, he/she can only use the Ethereum-based NFT market. In some cases, the original creator can also earn resale royalties. In some cases, an owner can also decide to hold the NFT forever without worrying about losing the asset because it is secured in the wallet on Ethereum. Every NFT has a creator that decides an element on which he/she wants to apply the standard to make it a digital asset. The creator can:

- Easily prove that he/she is the creator of that particular NFT.
- Determine the scarcity: The creator of an NFT can determine how many replications will exist. To make an asset unique, the creator may create an NFT where only one is minted as a special rare collectible.
- Earn royalties every time it's sold. Some creators can program royalties into it so each time the item is sold from one owner to another they earn a percentage as royalties.
- Sell it on any NFT market or peer-to-peer. The creator will not be locked to any platform and will not have the necessity of anyone to intermediate.

Non-fungible tokens are powerful due to the standards. The standards provide developers with the guarantee that the assets will behave in certain ways. Standards also describe how to interact with the basic functionality of the assets. Fungible tokens are regulated by the ERC-20 standard, whereas NFTs are regulated through an ERC-721 standard. ERC-721 was the first standard for representing non-fungible digital assets. ERC-721 is an inheritable Solidity smart contract standard, meaning that developers can easily create new ERC-721 compliant contracts. The standard relates the unique identifiers (each of which represents a single asset) to addresses. Basically, the owner is mapped to the unique identifier. It also provides a regulated way to transfer these assets using the *transferFrom* method.

In the beginning potential use of these unique tokens was unclear, mainly because they were totally different from the concept of fungible tokens and their applications. But they soon became the most suitable way for digital artists to make their art worth [18]. Digital artists create immense amounts of content, which require a lot of time and dedication. But often they are not compensated enough for their work. The problem with creating any type of digital content is that it can be easily copied and replicated without giving the creator any compensation or credit. The market of digital art and digital collectibles grew from fifty-two million to four hundred and ninety million dollars within the past year [19].

An artist who creates digital content and wants to sell his/her product online will not be able to fully secure his/her art, since anyone can make a copy of it. But, if the same digital work is combined with the Ethereum blockchain standard ERC-721 and converted into a NFT, this will make the art unique; in this case, nobody can have ownership of the digital product other than the person written on the NFT. In this situation, even though the same product can be replicated on the internet, no one else can have the ownership until the NFT is bought from the creator. This solution gave a huge boost to the market of digital art and digital collectibles, which had a growth of more than four hundred million dollars within the past year. Table I lists the ten most valuable NFTs ever sold [19].

TABLE I. THE TABLE SHOWS THE TOP 10 MOST VALUABLE NFTS EVER SOLD (RETRIEVED ON 08/04/2021) [19]

Name	**Artist**	**Description**	**Price sold $**
Metarift	Pak	Group of spherical objects moving	904,4013
Forever Rose	Kevin Abosch	Digital photograph of a rose	1 Million
CryptoPunk 4156	Algorithm-generated	Pixel art	1.5 Million
"Genesis" Estate	Axie Infinity virtual game	Estate bought within the Game	1.5 Million
CryptoPunk 6965	Algorithm-generated	Pixel art	1.6 Million
Twitter's first-ever tweet	Jack Dorsey	Twitter's first ever tweet	2.9 Million
Crossroads	Beeple	Digital video	6.6 Million
CryptoPunk 7804	Algorithm-generated	Pixel art	7.57 Million
CryptoPunk 3100	Algorithm-generated	Pixel art	7.58 Million
Everydays: the first 5000 days	Beeple	Piece made up of 5,000 images	69 Million

V. Conclusion

Blockchain technologies have been around for some time now and have been used for various types of applications in the last few years. One of the most recent applications, NFTs, are revolutionizing the market of digital collectibles and digital assets. Since NFTs are a complicated technology to understand, these applications are accessible only to a restricted set of people. This paper presented a balance between the technical and theoretical aspects that make up this technology, making it suitable for beginners who have an interest in learning the background and implementation of NFTs. This paper is also suitable for those who have a basic comprehension of blockchain technologies and want to extend their knowledge in the field.

References

[1] Richard Twesige. A simple explanation of Bitcoin and Block Chain technology. Jan. 2015. doi: 10.13140/2.1.1385.2486.

[2] Dejan Vujičić, Dijana Jagodic, and Siniša Ranđić. “Blockchain technology, bitcoin, and Ethereum: A brief overview”. In: (Mar. 2018), pp. 1–6. doi: 10.1109/INFOTEH.2018.8345547.

[3] B.Singhal, G. Dhameja, and Panda. How Blockchain Works. Beginning Blockchain. 2018, pp. 31–148. doi: 10.1007/978-1-4842-3444-0_2.

[4] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. In: Cryptography Mailing list at https://metzdowd.com (Mar. 2008).

[5] Assia Yoni et al. “Colored Coins whitepaper”. url: https://docs.google.com/document/d/1AnkP_cVZTCMLIzw4DvsW6M8Q2JC0lIzrTLuoWu2z1BE/edit#heading=h.wxrvzqj8997r.

[6] Satoshi Nakamoto. namecoin white paper. url: https://www.namecoin.org.

[7] Szabo Nick. “The Idea of Smart Contracts”. In: (). url: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html.

[8] Dejan Vujičić, Dijana Jagodic, and Siniša Ranđić. “Blockchain technology, bitcoin, and Ethereum: A brief overview”. In: (Mar. 2018), pp. 1–6. doi: 10.1109/INFOTEH.2018.8345547.

[9] Buterin Vitalik. “Ethereum white paper”. In: GitHub repository (2013).

[10] Contract ABI Specification. 2019. url: https://solidity.readthedocs.io/en/latest/abispec.html.

[11] Vogelsteller Fabian and Buterin Vitalik. “EIP-20: ERC-20 Token Standard”. In: (Nov. 2015). url: https://eips.ethereum.org/EIPS/eip-20.

[12] Entriken William et al. “EIP-721: ERC-721 NonFungible Token Standard”. In: (Jan. 2018). url: https://eips.ethereum.org/EIPS/eip-721.

[13] J. Dafflon, J. Baylina, and T. Shababi. “ERC-777 token standard”. In: (2015). url: https://eips.ethereum.org/EIPS/eip-777.ù.

[14] W. Radomski et al. “ERC-1155 multi token standard”. In: (2015). url: https://eips.ethereum.org/EIPS/eip-1155.

[15] M. Kupriianov and J. Svirsky. “Base security token standard draft”. In: (2019). url: https://eips.ethereum.org/EIPS/eip-1462.

[16] H. Marks J. Shiple and D. Zhang. “Ldgrtoken standard draft”. In: (2019). url: https://eips.ethereum.org/EIPS/eip-1450.

[17] Frankenfield Jake. ICOs explanation. 2020. url: https://www.investopedia.com/terms/i/ initial-coin-offering-ico.asp.

[18] Qin Wang et al. Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges. 2021. arXiv: 2105.07447[cs.CR].

[19] Luno Team. Top 10 most valuable NFTs ever sold. May 2021. url: https://www.luno.com/blog/en/post/top-10-most-valuable-nfts-ever-sold.