

Smart Contracts and NFTs: Non-Fungible Tokens as a Core Component of Blockchain to Be Used as Collectibles



Akash Arora, Kanisk, and Shailender Kumar

Abstract Non-fungible tokens are one of the most important future application domains for smart contracts. Ethereum is the pioneer of a blockchain-based decentralized computing platform that has ultimately standardized these types of tokens into a well-defined interface, now known as ERC-721. Blockchain-based cryptocurrencies have received extensive attention recently. Massive data has been stored on permissionless blockchains. This paper aims to analyze blockchain and cryptocurrencies' technical underpinnings, specifically non-fungible tokens or "crypto-collectibles," with the help of a blockchain-based image matching game. While outlining the theoretical implications and use cases of NFTs, this paper also gives a glimpse into their possible use in the domain of human user verification to prevent misuse of public data by automated scripts. This demonstrates the interaction of the ERC-721 token with the Ethereum-based decentralized application. Further, we aim to reach a definitive conclusion on the benefits and challenges of NFTs and thus reach a solution that would be beneficial to both researchers and practitioners.

Keywords Blockchain · Dapp · Decentralized · Ethereum · NFT · ERC-721 · Smart contract · Truffle

A. Arora (✉) · Kanisk · S. Kumar
Department of Computer Science Engineering, Delhi Technological University,
Shahbad Daulatpur, Bawana, New Delhi, India
e-mail: akasharora_2k17se13@dtu.ac.in

Kanisk
e-mail: Kanisk_2k17se13@dtu.ac.in

S. Kumar
e-mail: shailenderkumar@dce.ac.in

1 Introduction

The concept of blockchain was initially envisaged as a decentralized network that could store records of transactions that happened on it and thus act as a source of trust. “The data stored on the blockchain is immutable and updated by the peer-to-peer network” [1]. By design, entities called “blocks” constitute a blockchain. These blocks have the capability by design to store transactions that have been broadcasted. Only such transactions are deemed to be valid that have been broadcasted and hence been verified. To put it quite simply in layman terms, a blockchain is a database [1]. To grasp this concept clearly, we have to understand what a database is. A database points to collecting stored information either electronically or otherwise on either a computer system or a ledger. In databases, information or data is typically structured in table format to allow for easier searching and filtering specific details. While a blockchain ultimately aims to serve a similar purpose as a database, it offers many advantages over the latter. The primary difference arises in the way that the data is stored in a blockchain. While in a database, data is often stored in a tabular format, a blockchain stores data in small chunks or blocks (and hence the name blockchain) that hold sets of information that are all connected together.

The entities which constitute a blockchain and are known as “Blocks” which are designed to have storage capacities. When the storage capacity of one such block is completely full, it is then attached to the previously filled block. In this manner, a chain of data known as the “blockchain” is created. New information is written after creating that freshly added block is added into a newly created block that will also be added to the chain once it is completely full (Fig. 1).

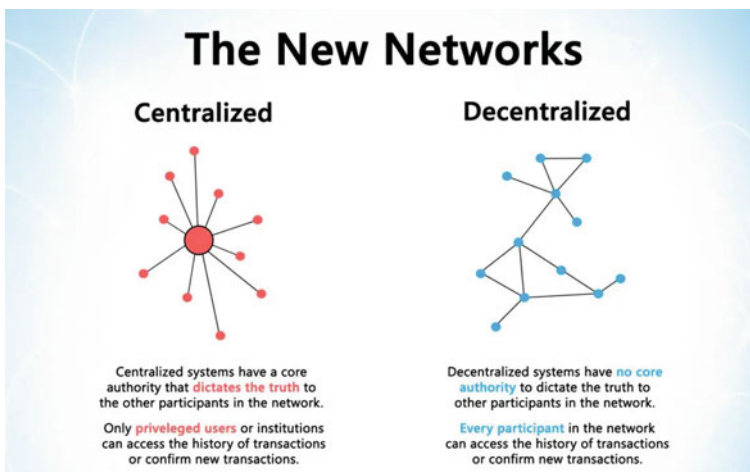


Fig. 1 Centralized versus decentralized system. Source <https://blockgeeks.com/blockchain-infographics/>

One of the central tenets of blockchain technology is “Decentralization” [2, 3]. The concept of decentralization can be best understood by contrasting it with a centralized network. The basic idea behind any centralized network is straightforward, “any centralized entity will anyways store all the data, and hence, an individual will always have to interact solely with this particular centralized entity to get whatever information or data that one requires” [2]. The traditional Client–Server network model is a perfect example of a centralized system. However, centralized systems have their drawbacks. The primary disadvantages include security vulnerabilities, lack of redundancy, and excessive dependency on the central node, which increases the chances of complete failure. These are the drawbacks that a decentralized system aims to address. By definition, in a decentralized system, “the information will not be stored by any one single entity. Everyone in the network owns the information” [2, 3].

Every node in a blockchain is designed to have a complete record of all the data that has been stored on the blockchain since its inception. The error in one node’s data can be corrected by referencing the correct data in thousands of other nodes. This way, the creators of blockchain have ensured by a design that a single malicious node cannot inject information on the entire chain. This also means that the transaction history present in each block that constitutes the Bitcoin blockchain cannot be altered or modified. Another advantage of this decentralized nature of blockchain is transparency. A user can independently verify any transaction transparently, either by a personal node or by using automated scripts that crawl through the blockchain. This would enable interested individuals to witness ongoing transactions. Moreover, every constituting node contains a private version of the blockchain that gets refreshed as new blocks are verified and included.

Non-fungible tokens are an integral part of this paper. Non-fungible tokens are dissimilar tokens that do not conform to the concept of fungibility. The idea of fungibility deals with the currency’s ability to maintain a standard value and uniform acceptance. Fungibility implies the immunity of the currency’s value from its precedents; this ensures that each piece of that currency is equal in value to every other piece [4]. This means that one ₹100 note in my pocket is equal in value, identical, and replaceable with any other ₹100 note in any other individual’s pocket. Hence, a ₹100 note is a fungible asset.

On the contrary, non-fungible tokens are blockchain assets that are designed not to be equal. Non-fungibility is the USP of such investments. This characteristic is of enormous importance when we are talking about rare and unique collectibles. The value of such items is derived from their non-fungibility (Fig. 2).

Another essential part of a blockchain network is smart contracts. Strictly speaking, a smart contract can be classified as a self-executing contract with the terms of the agreement between the buyer and seller being written from the get-go into the lines of code that make it up. The script and the contract contained within the smart contract are present across a distributed ledger and a decentralized network. The code controls the execution, and transactions are trackable and are hence irreversible.

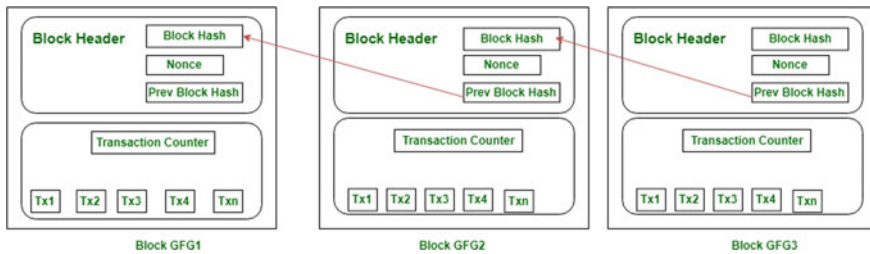


Fig. 2 Blockchain structure. Source <https://www.geeksforgeeks.org/proof-of-work-pow-consensus/>

“The biggest advantage of smart contracts is that they permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism” [5]. Smart contracts were originally proposed in the late twentieth century by Nick Szabo. Nick was a western computer scientist who had invented a virtual currency called “Bit Gold.” His original intention behind creating smart contracts was to scale up the capabilities of contemporary transaction methods, such as point of sale (POS) to the digital realm.

Having glossed through this paper’s fundamental building blocks, it would be prudent to revert to this paper’s intention. We intend to showcase the interaction of NFTs built on the ERC-721 standard with smart contracts on the Ethereum blockchain using an Ethereum Dapp. To achieve this, we will be mining NFTs using an image tile-matching game. In this game, we will be collecting tokens in the form of collectibles. Furthermore, this collection is driven by smart contracts, and hence, the interaction between smart contracts and NFTs is established on the Ethereum network.

2 Related Work

2.1 Blockchain

A blockchain can be referred to as a decentralized network that can keep transaction records and thus get the capability to act as a bank of reliability [6]. The electronic details cached on the blockchain are perceived to be unchangeable, and it is designed to be continually updated by the peer-to-peer network. These blocks have the capability by design to store transactions that have been broadcasted. Only such transactions are deemed to be valid that have been broadcasted and hence been verified.

Blockchain technology can be said to have evolved from the development of Bitcoin as a distributed, immutable ledger that is maintained and verified on a network of peers. Since then, several industries have sought to explore the

fundamental peer-to-peer technology and its myriad other applications, including creating extremely cost-effective and decentralized business network models or architecture.

Blockchain can also be a working specimen of a distributed computing system with high secure fault tolerance. Satoshi Nakamoto pioneered blockchain in 2008, who initially conceptualized it and then implemented it in the following year as a core component of the digital currency Bitcoin. Blockchain forms the public ledger for all transactions that are done using Bitcoin. Blockchain databases are designed to be managed without any human intervention, i.e., autonomously using a peer-to-peer network and a distributed timestamping server. Blockchain helped Bitcoin to attain the status of the first digital currency to solve the double-spending problem.

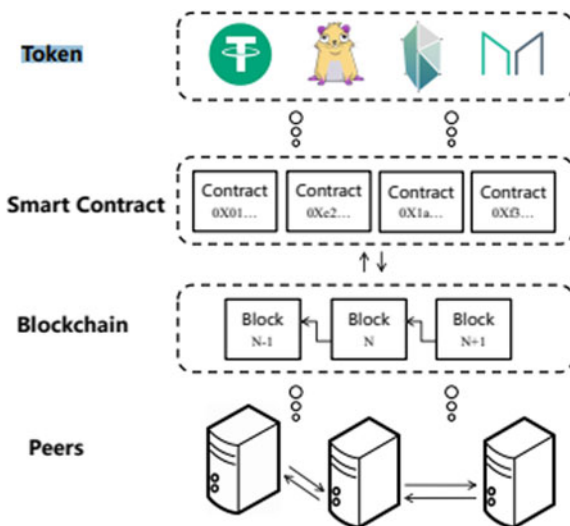
Blockchains, based on ownership and exclusivity, can be classified into private and public. The primary difference between the two is that anyone can join and contribute to a public blockchain. In contrast, one requires permission from a central authority to do so on a private blockchain. An advantage of public blockchains is that they are pseudo-anonymous by design as all transactions are public and hence hard to link them to identity.

2.2 *Ethereum*

Ethereum was initially conceptualized in 2013 by Vitalik Buterin, a 19-year-old Canadian developer. He sought to expand on the Nakamoto whitepaper and published a paper which was titled “A Next-Generation Smart Contract and Decentralized Application Platform.” Buterin worked on all the primary strengths of Bitcoin, which included an incentive scheme for miners, proof of work, and hashing, to name a few, and further capitalized it to create a new Blockchain known as Ethereum. A significant advantage possessed by Ethereum can be said to be the implementation of a turing-complete language. What that meant was that the Ethereum blockchain could handle complex code to exploit the immense computing power that was being harnessed (but ultimately wasted) by the incentive scheme, something which could not be done successfully or optimally by the Bitcoin blockchain due to inherent defects of designs. It can be said that while Bitcoin was envisaged to be an unhackable store of value with few capabilities other than this one objective. Ethereum was designed and intended from the start to expand and ultimately realize the full scope of capabilities that could potentially be offered by blockchain technology and hence, theoretically, create a decentralized computer that could operate on a global scale.

It would be worth noting that Ethereum is a blockchain designed to build on the abstractions introduced by Bitcoin. Hence, Ethereum does not have an overarching outfit to handle the transactions leading to all recorded events on a blockchain. A consensus algorithm called proof of work is used to secure this open and decentralized network. The general public has complete access to all events

Fig. 3 Overview of ethereum blockchain. *Source* Zheng et al. [7]



happening on the blockchain. Ethereum is supposed to tackle use cases related to Bitcoin, with error avalanche, underlining and dictating the creation of coins on the network and arriving at a network-wide consensus. A spinoff of this capability is that Ethereum has become one of the most popular platforms for developing new blockchain-based applications (Fig. 3).

2.3 Nodes

Nodes can be described as the participants in the blockchain. All the nodes are connected on a peer-to-peer network. Every node has complete access to the blockchain and can thus verify any incoming transactions. The operational requirement of a full node is a copy of the blockchain and ample storage space. The basic requirement for a user to become a miner is to run a full node. There is one other type of node, which is known as a lightweight node that stores only the hashes of the blocks. It occupies significantly less space as it receives additional information from full nodes.

2.4 Mining

Mining is the method through which new transactions are added to the blockchain. Hence, mining can be described as the process through which a block of transactions can be created and added to the Ethereum blockchain. Ethereum currently

uses a consensus mechanism known as proof of work (PoW) [8]. Mining is said to be the lifeblood of proof of work. Ethereum miners are generally computers running specific software which use time and high computation power to process transactions and produce blocks. Further, miners are greatly incentivized to compete with each other because finding a reward in the form of newly minted coins and transaction fees is guaranteed on finding the right solution. When compared to Bitcoin, Ethereum does not have a limit on the number of possible Ether that can be mined. Subsequent to the successful creation of the target block and its verification by the network, “miners start competing for the next block” [1].

2.5 Proof of Work

It is a type of consensus algorithm which is also used in Ethereum blockchain. Proof of work can be best described as the algorithm that has to be solved by miners so as to find an appropriate hash for the next block.

Hash rate denotes the rate of mining activity on the network. We can suitably adjust the difficulty of this algorithm if the hash rate increases. This algorithm is said to be innovative because not only does it allow unrestricted entry to anyone who is running a full node but it also accredits the network to frame a protocol that can be used to amend the blockchain. Furthermore, it has error avalanche handling capability, which means that it can positively handle setbacks and malicious users that disrupt the network (Fig. 4).

2.6 Addresses and Wallet

Asymmetric cryptography is a fundamental concept to blockchain. Any user can spawn a random digital signature in the form of a private key. This key can be used to spawn a public key. The user address can be spawned using the public key only. This is the address where the number of funds can be stored, and then, the user can use his private key to sign transactions from his address. The public key shall be used to check the origin of those transactions and get them verified by the network.

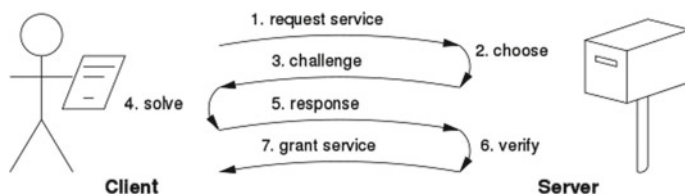


Fig. 4 PoW consensus. Source https://en.wikipedia.org/wiki/Proof_of_work

This means that access to funds would be lost if the private key is lost. Wallets help users to keep track of their financial assets and addresses. In the case of Ethereum, the wallet also can communicate internally with smart contracts and Dapp.

2.7 Transactions

“Messages that are signed by the address that has triggered them and can thus change the state of the network are known as transactions” [1, 9]. The fundamental concepts of the proprietary algorithm used are as follows:

Proof of authority: Unique signature of the owner of the private key.

Non-repudiation: The above is undeniable.

Unchangeability of transaction data: The transaction contents are unchangeable, and its integrity is not compromised.

The transaction structure on the Ethereum blockchain is the following.

Nonce: Numerical quantity of transactions that have been sent from an address.

Gas price: The financial cost to be levied on the initiator of the transaction for its execution.

Gas limit: The maximal gas permitted.

Recipient: The smart contract address or the payable address for the transaction.

Value: The financial weight in terms of Ether in the transaction.

Data: Quantity of binary input.

v , r , s : Worth of transaction signature.

2.8 Non-Fungible Tokens (NFTs)

Non-fungible tokens possess a specific digital signature that is contained within their smart contract. This identifying information is what makes each NFT different from another, and hence, an individual cannot swap them for another token. What this means is that they cannot be supplanted one for another, considering no two are undifferentiated. To put things in perspective, banknotes can be simply exchanged one for another if they possess the same value, and thus, it makes no difference to the holder. Non-fungible tokens are indivisible, in the same way as it is not possible to send someone part of a movie ticket. Part of a movie ticket is not redeemable and is not worth anything on its own. Crypto-kitties collectibles were the first famous non-fungible tokens. When referring to crypto-kitties collectibles, each cat is unique; i.e., if anyone sends someone a crypto-kitty and receives a crypto-kitty from someone else, the received kitty will be a unique and different crypto-kitty from the one you sent. Some noteworthy attributes are associated with non-fungible tokens. For instance, tokens are generally chained to a specific asset. Moreover, they can also prove the right of possession over items like vinyl's used in games and even the ownership of physical assets. Other tokens can be purported to possess

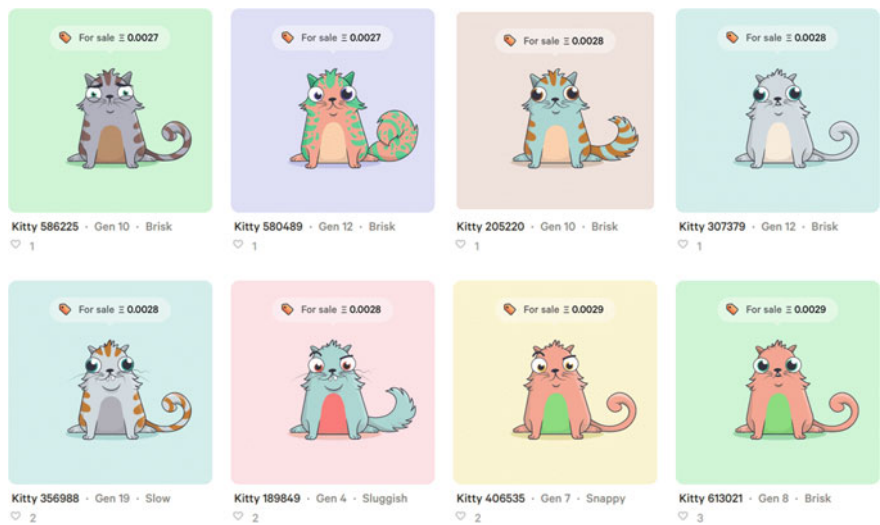


Fig. 5 Crypto-kitties collectibles. <https://www.investopedia.com/news/cryptokitties-are-still-things-heres-why>

the characteristic of fungibility in the same way as any fiat currency is said to be fungible. Fungible tokens are different from NFTs as they are identical, and thus, they have the same denomination when exchanged.

NFTs are as follows:

- (a) Rare (b) Unique (c) Indivisible (d) Transferable (Fig. 5).

2.9 Smart Contracts

For a layman, “smart contracts can be described as lines of code that are stored on a blockchain and are automatically executed when predetermined or predefined terms and conditions are met” [10]. Technically speaking, smart contracts were defined in 1994 by Nick Szabo as “a computerized transaction protocol which can be used to execute the terms of any contract.” As is evident from the description, business collaborations can demonstrate most of the smart contracts’ upsides. “Smart contracts can be used to enforce some agreement so that all participants can be certain of the outcome without an intermediary’s involvement in such business collaborations” [10]. Ethereum pioneered the use of smart contracts among blockchain, which led to the development of high-utility applications in various sectors, including the finance sector. “The code can only be deployed on the network after its high-level code is compiled to low-level code and is sent to a 0×0 address” [1]. Subsequently, “an address is allotted to it which can subsequently be used to utilize the contract’s functionalities by calling its functions. Although the high-level

Fig. 6 Representation of smart contracts. *Source* <https://www.edureka.co/blog/smart-contracts/>



language of choice for writing Ethereum code in solidity, there are various other high-level languages which can be used for writing Ethereum code” [1] (Fig. 6).

3 Literature Review

After assessing a substantial number of research papers on the subject matter at hand, we learned the following points. Our learnings can be categorized under the following sub-headings.

3.1 Decentralized Finance (DeFi)

The YoY growth of the market for NFTs is expected to be above 60% in this FY. Adopting an iterative research approach to the use-case about ownership of NFTs, a novel solution that proposes to increase market penetration of NFTs has been reached. This solution will provide a platform for developing increasingly complex financial instruments, enabling NFT users to leverage and lease their valuable assets linked/stored in tokens [1].

Some proposed use cases include:

1. Leasing tokens to generate income for passive users.
2. Lending/leveraging tokens generate quick liquidity, thereby eliminating the overhead costs associated with selling the asset.

3. Creating financial tools on the lines of futures will help to stabilize the market.
4. Creating a legal framework for the tokenization of real estate assets.

3.2 Blockchain Development Platforms

There are multiple platforms that users can utilize for the development and implementation of smart contracts based on the level of advancement attained by the user and the specific use cases that the user is targeting. Truffle suite is best suited for developers who are just starting and are nascent to this technology owing to the limited capacity of truffle suite to implement multi-layered smart contracts. Developers who are looking for a robust compiler and development environment would be better suited to utilizing remix. An ideal combination would be using truffle suite for testing while remix would be utilized for compiling the smart contracts, and the deployment would be handled by Mist and Geth [10].

3.3 Recent Advancements and Future Trends in Smart Contracts

Smart contracts are fundamental to the optimal exploitation of the immense potential offered by blockchain and associated technologies. Due to this, a large amount of research is going to be done on smart contracts in the academic field, and the resultant interest generated in the industry will be immense. The trump card of smart contracts is their immutable and irreversible characteristic. These characteristics facilitate the exchange of money and other assets in a way which avoids the involvement of a third party. Smart contracts face certain challenges which include reentrancy vulnerability, transaction ordering dependence, time-stamp dependence, untrustworthy data feed, and privacy issues. The future of society is a transition into a cyber physical social system (CPSS). Blockchain and smart contracts will play an instrumental role in aiding this transition [5].

3.4 Trends on Crypto-Collectibles

The meteoric rise of crypto-collectibles on the blockchain network can be best understood by taking a glance at crypto-kitties. Crypto-kitties can be said to be the pioneer among collectibles in terms of raising financial interest. An initial amount of \$12 million followed by another \$15 million was successfully raised by VC method. Thus, it can be argued that crypto-games, just like crypto-Kitties, are financially viable. These games would serve to amplify the movement toward blockchain-based gaming. Coming back to crypto-kitties, the game works on the

principle of ownership of tokens, which are very valuable due to a limited supply by algorithmic design, being proven by blockchain. This lends to value addition of collectibles. Moreover, cryptocurrencies also reflect a kind of “digital materiality” due to this design [9].

Hence, it can be concluded that the major factors which would determine the worth of crypto-game tokens can be refined to three most influential ones. The first factor can be said to be the limited technical infrastructure which impedes the capabilities of the network. Limited technical infrastructure can refer to the lack of scalability of the blockchain itself or the steep learning curve involved in learning how to operate crypto-wallet. The second limiting factor is the unequal playing field for blockchain users. By an unequal playing field, we are referring to the gas fees required to be paid by the users to cryptocurrency miners for the purpose of infrastructure maintenance which gives an unfair advantage to users with greater financial resources. Finally, the third limiting factor is the unclear or unvalidated legal ownership of tokens due to the anonymity accorded to the users by design on a blockchain. Any attempt to validate or prove legal ownership of tokens can only be done after the user de-anonymizes himself which would then defeat the purpose of cryptocurrencies based on a blockchain network.

3.5 Blockchain Digital Art

One of the major use cases of blockchain technology lies in the creative industry. The creative industry relies heavily on the revenue generated by the trio of rights management, licensing, and data management of digitized/digitizable products. Blockchain technology could be leveraged to manage the aforementioned in a manner which is automated and decentralized. Block technology will increase the power in the hands of the original content creators by offering a shared data layer on the application level rather than on the data level which is offered by existing Internet-based solutions. A business model which is based on token issuance and management by the creator can be built on the framework provided by blockchain technology [6].

4 Research Background and Methodology

4.1 Decentralized Applications (DApps)

Dapps were created to operate as applications compatible with smart contracts enabled, which had the capacity to work on distributed ledgers. Users have the capability to interact with a wide range of Dapps on Ethereum.

4.2 *How Do You Make a Token?*

The definition of tokens is contained within smart contracts. A token smart contract, once deployed, keeps track of the number of tokens owned by any address. Addresses also have the capability to transfer tokens attributed to them to other addresses. The smart contract keeps track of the number of tokens owned by each individual and not the underlying blockchain itself in contrast to ETH [11]. Hence, we would need to query the smart contract in order to find out how many tokens an address has. This is a distinction that everyone must recognize. When one is querying how many ETH an address has, one is querying the blockchain. When user query the amount of ETH (Ethereum currency) linked to an address, blockchain is queried. In contrast, when user query the amount (number) of tokens linked to an address, smart contract is queried. Hence, to know how many ETH and tokens a wallet has, an individual needs to know all the addresses of where the token smart contracts are deployed on the blockchain [1].

4.3 *ERC-721*

ERC-721 has been established as the basest standard that a smart contract will have to adhere to so that it is allowed to own and trade unique tokens. ERC-721 has not decreed a caliber for token metadata and neither does it restrict adding supplemental functions. Thus, ERC-721 can be described as a protocol that is subject to free peer assessment, which advises how to build NFTs on the Ethereum decentralized network of nodes. A whole lot of tokens are fungible (fungibility implies that every token is exactly the same to and changeable with any other token), ERC-721 tokens have to be all one of a kind. ERC-721 tokens are non-fungible in nature. “Their lack of fungibility connotes that each token has a set of characteristics and standards associated with it which are exclusionary to it” [1]. The “uniqueness of such tokens makes ownership more desirable especially in the case of collectibles and other such highly sought-after tokens” [1].

4.4 *Open Zeppelin*

Open Zeppelin is a library that was created for the purpose of safe contract development. It is built on a solid foundation of community-vetted code. It enables the implementation of standards like ERC20 and ERC721. It boasts of a flexible role-based per missioning scheme. Further, components of solidity, which are used to build custom contracts and complex decentralized systems, are reusable, which enhances its utility. Moreover, top-grade interoperability with the gas station network for systems with no gas fees also contributes to its varying uses. It has been audited by leading security firms and is, thus, perfectly safe.

4.5 *Truffle Suite*

“Truffle is a development environment, testing framework, and asset pipeline for Ethereum, which aims to make the life of an Ethereum developer much easier” [12]. Truffle offers the following advantages to an Ethereum developer:

- Deployment and binary management linked using built-in smart contract compilation.

- Computer robotized testing of contract.

- Custom build processes supported by configurable build pipeline.

- Scriptable deployment and migrations framework.

- Public and private networks deployed using network management.

- Scripts executed within a truffle environment by an external script runner.

4.6 *Web3.js*

Web3.js enables an Ethereum developer to fulfill his/her second responsibility, i.e., creating modules that communicate with the Ethereum decentralized network. Web3.js is basically an amalgamation of non-volatile computer code that permits the developer to perform actions like the transfer of Ether and modify smart contracts. The diagram demonstrates how a client does tête-à-tête with Ethereum.

Web3.js talks to the Ethereum blockchain using a JSON RPC, which stands for the “Remote Procedure Call” protocol. As Ethereum is based on a peer-to-peer network of nodes that needs to save an additional version of important information on the blockchain, Web3.js allows a developer or a user to make requests to an individual Ethereum node with JSON RPC in order to modify or access the data in the network. An appropriate would be the cohesion of jQuery with a JSON API to modify a web server.

4.7 *Metamask*

“MetaMask is a browser plugin that also serves as an Ethereum wallet and is installed like any regular plugin. It allows users to store Ether and other ERC-20 tokens and thus enables them to consummate transactions with any Ethereum address” [13].

4.8 Ganache

Ganache is used for setting up an individual Ethereum decentralized network of nodes for validating solidity contracts. Moreover, it provides more functionalities when compared to other similar software.

5 Design and Implementation Setup

See Figs. 7, 8.

A. Installing the Following Dependencies:

```
{
  ``name``: ``blockchain-game``,
  ``version``: ``0.1.0``,
  ``description``: ``token collection game``,
  ``author``: ``akash and kanisk``,
  ``dependencies``: {
    ``@openzeppelin/contracts``: ``^2.3.0``,
    ``babel-polyfill``: ``6.26.0``,
    ``babel-preset-env``: ``1.7.0``,
    ``babel-preset-es2015``: ``6.24.1``,
```

Fig. 7 Design workflow: <https://medium.com/free-code-camp/how-to-design-a-secure-backend-for-your-decentralized-application-9541b5d8bddb>

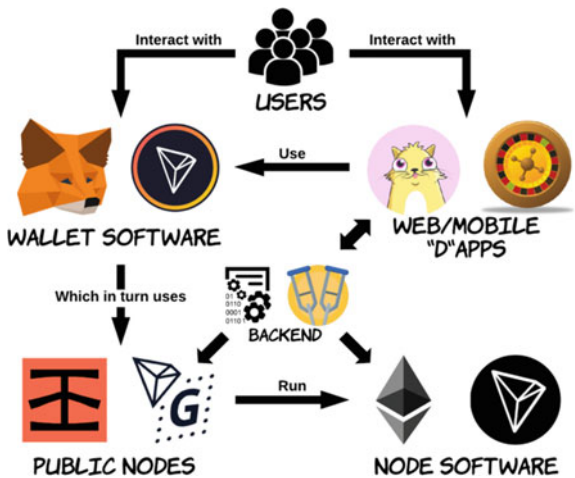


Fig. 8 Setting up environment

```
npm install --g truffle@5.1.39
```

```

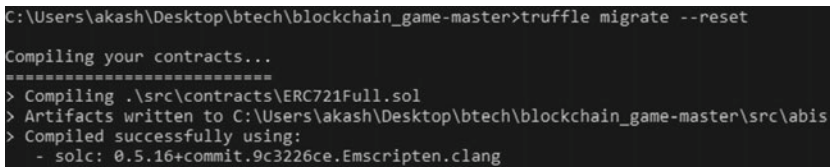
``babel-preset-stage-2``: ``6.24.1``,
``babel-preset-stage-3``: ``6.24.1``,
``babel-register``: ``6.26.0``,
``bootstrap``: ``4.3.1``,
``chai``: ``4.2.0``,
``chai-as-promised``: ``7.1.1``,
``chai-bignumber``: ``3.0.0``,
``react``: ``16.8.4``,
``react-bootstrap``: ``1.0.0-beta.5``,
``react-dom``: ``16.8.4``,
``react-particles-js``: ``^3.4.1``,
``react-scripts``: ``2.1.3``,
``truffle``: ``5.0.5``,
``truffle-flattener``: ``^1.4.2``,
``web3``: ``1.0.0-beta.55``
n}

```

See Figs. 9, 10.

6 Application. Conclusions

See Figs. 11, 12, 13, 14, 15, 16, 17.



```

C:\Users\akash\Desktop\btech\blockchain_game-master>truffle migrate --reset
Compiling your contracts...
=====
> Compiling .\src\contracts\ERC721Full.sol
> Artifacts written to C:\Users\akash\Desktop\btech\blockchain_game-master\src\abis
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang

```

Fig. 9 Compilation of smart contract


```

Starting migrations...
=====
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)

1_initial_migration.js
=====

Replacing 'Migrations'
-----
> transaction hash: 0xcabda9f9cd0af580fbd415ca42290620d536c8a450f0f4e9ade3ea01a6b28816f
> Blocks: 0        Seconds: 0
> contract address: 0x02C116f2d00Fe641C77af4cd4548e8718a90dE9c
> block number:    7
> block timestamp: 1606671479
> account:         0xbB82F68A0eE8942b4552159D83C2dBAF1230EC0E
> balance:         99.93599252
> gas used:        225237 (0x36fd5)
> gas price:       20 gwei
> value sent:      0 ETH
> total cost:      0.00450474 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost:      0.00450474 ETH

2_deploy_contracts.js
=====

Replacing 'MemoryToken'
-----
> transaction hash: 0x22cfaea207211e9d38fd66c6d405c5813f6348c49fe29ef1d66d751f5314cf4a
> Blocks: 0        Seconds: 0
> contract address: 0xc2E7000ae8210dAA2C508316a60DE30c882b264d
> block number:    9
> block timestamp: 1606671480
> account:         0xbB82F68A0eE8942b4552159D83C2dBAF1230EC0E
> balance:         99.89009146
> gas used:        2252690 (0x225f92)
> gas price:       20 gwei
> value sent:      0 ETH
> total cost:      0.0450538 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost:      0.0450538 ETH

Summary
=====
> Total deployments: 2
> Final cost:       0.04955854 ETH

```

Fig. 10 Deploying the smart contract

```

> contract address: 0xc2E7000ae8210dAA2C508316a60DE30c882b264d

```

Fig. 11 Smart contract address

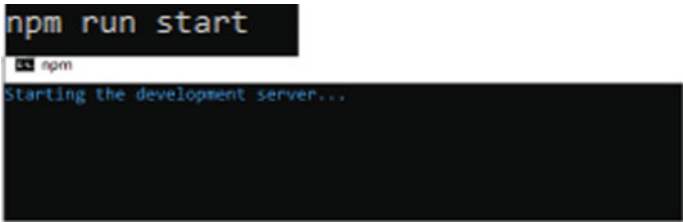


Fig. 12 For starting the server



Fig. 13 Application home page

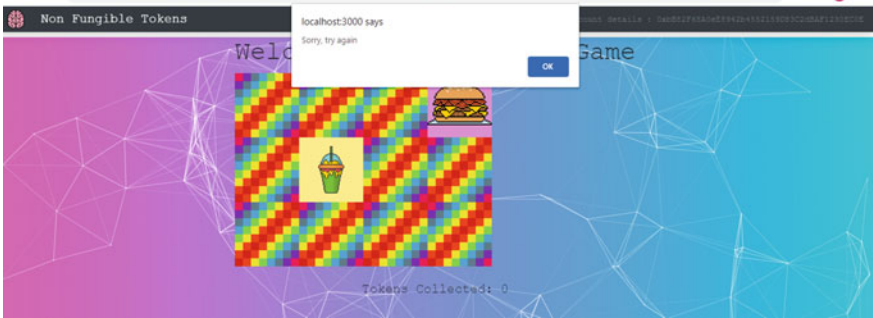


Fig. 14 Flow 1: tiles did not match (unsuccessful attempt)

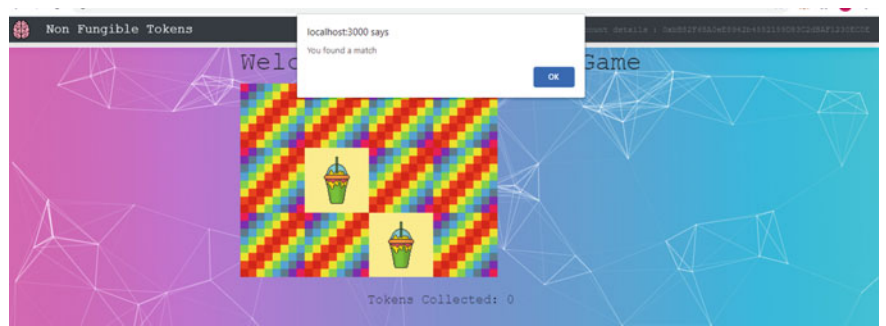


Fig. 15 Flow 2: tiles match (successful attempt)

Fig. 16 Transaction notification by MetaMask on successful matching and collection of tokens initiated

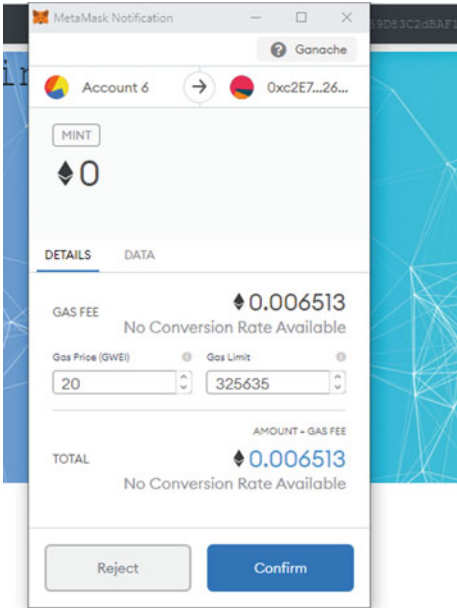


Fig. 17 Token collected



7 Results

The application environment is initially set up by installing the required dependencies. Consequent to the successful installation of the required dependencies, we then proceed to initiate the Ethereum node on the blockchain network using Ganache. Pursuant to the successful initiation of the Ethereum node, we then move on to smart contracts. The custom-tailored smart contract, written in solidity, has to be compiled by using truffle suite. Upon the successful compilation of the smart contract, we then deploy the smart contract. Once the smart contract is successfully deployed, a unique address is provided to the smart contract using which all further transactions can be accounted for and tracked [14]. This completes the back-end setup of our Ethereum decentralized application (Dapp).

The front-end part of the application is initiated by starting the application server, which is based on a React.js framework. It should be noted that a prerequisite for the successful initiation of the application server is that the plugin for MetaMask is installed in the web browser. The absence of which will lead to an unsuccessful connection attempt with the back-end server. On the successful initiation of the application server, we will see an image tile-matching game hosted successfully on localhost.

There are two flows of events that may take place henceforth. The first flow can be labeled as the unsuccessful match event. In this flow, as two similar-looking tiles are not matched by the user, no transaction is initiated by the smart contract as the token IDs do not match. The local host displays an unsuccessful attempt message and directs the user to make another attempt. The second flow can be labeled as a successful match event. In this flow, as two similar-looking tiles are correctly matched by the user, the smart contract initiates the transaction as the token IDs are matched correctly. The local host displays a successful attempt message to the user. A MetaMask window subsequently pops up to inform the user of the gas charge deducted to facilitate the transaction. The transaction details can be tracked using Ganache.

After the completion of the transaction, the token is collected and displayed on the application window. Hence, we have successfully gained a collectible NFT. By gaining this collectible NFT, the user can establish the provenance of the collectible as that collectible now possesses the uniqueness and rareness of the NFT it is linked to.

8 Conclusion and Future Scope

To conclude, we can state that the transaction of non-fungible tokens using deployable smart contracts has immense commercial and academic implications. The ERC-721 standards have given us immense power to accomplish next-generation advancements in fields as varied as art, culture, online gaming, rare collectibles, and more. This whole process hence confirms the non-fungibility of the

tokens created on the ERC-721 standard and their interaction with the smart contract as shown with the help of an Ethereum Dapp.

In the future, this work can be extended to the domains of authentication and human user verification so as to prevent misuse of cyber facilities and public data by automated scripts. These non-fungible tokens can also be used to validate the authenticity and pedigree of high value, rare, vintage collectibles, which could otherwise be faked so as to commit fraud. This will give the power of proof of originality/ownership to the owner or creator in the digital world. Hence, advanced applications can be made along similar lines of thoughts, which would give us a deeper insight into this domain and hence help us in verifying the antecedents of the product or any other item of value.

References

1. Musan, D.I.: NFT . finance leveraging non-fungible tokens. Imperial College London, Department of Computing (2020)
2. Kumar, A., Kumar, S.: A systematic review of the research on disruptive technology—blockchain. In: 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 900–905 (2020). <https://doi.org/10.1109/ICCES48766.2020.9138055>
3. Kumar, A., Kumar, S.: Implementation of decentralized electronic polling system using ethereum blockchain. *Int. J. Adv. Sci. Technol.* **29**(4), 10717–10728. (SCOPUS Indexed) ISSN: 2005–4238 (2020)
4. Uribe, D., Waters, G.A.: Privacy laws, genomic data and non-fungible tokens. *J. Br. Blockchain Assoc.* **3**(2) (2020) [Online]. Available: <https://jbba.scholasticahq.com/article/13164-privacy-laws-genomic-data-and-non-fungible-tokens>
5. Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., Wang, F.Y.: An overview of smart contract: architecture, applications, and future trends. *IEEE Intell. Veh. Symp. Proc.* 2018-June, no. Iv, pp. 108–113 (2018). <https://doi.org/10.1109/IVS.2018.8500488>
6. Chevet, S.: Blockchain technology and non-fungible tokens: reshaping value chains in creative industries. Sylvé CHEVET Under the supervision of Alain BUSSON, pp. 1–73 (2017)
7. Zheng, P., Zheng, Z., Wu, J., Dai, H.: XBlock-ETH: Extracting and exploring blockchain data from ethereum. *IEEE Open Journal of the Computer Society*, pp. 1–1 (2020). <https://doi.org/10.1109/OJCS.2020.2990458>
8. Buterin, V.: A next-generation smart contract and decentralized application platform. *Ethereum*, no. January, pp. 1–36 (2014) [Online]. Available: <http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf>
9. Bal, M., Ner, C. arXiv: NFTracer: A Non-Fungible Token Tracking Proof-of-Concept Using Hyperledger Fabric, pp. 1–9 (2019)
10. Chirtoaca, D., Ellul, J., Azzopardi, G.: A framework for creating deployable smart contracts for non-fungible tokens on the ethereum blockchain. In: 2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), Oxford, United Kingdom, pp. 100–105 (2020). <https://doi.org/10.1109/DAPPS49028.2020.00012>
11. Serada, A., Sihvonen, T., Harviainen, J.T.: CryptoKitties and the New Ludic Economy: How Blockchain Introduces Value, Ownership, and Scarcity in Digital Gaming. *Games Cult.* (2020). <https://doi.org/10.1177/1555412019898305>

12. Anilkumar, V., Joji, J.A., Afzal, A., Sheik, R.: Blockchain simulation and development platforms: survey, issues, and challenges. 2019 International Conference on Intelligent Computing and Control Systems ICCS 2019, no. Iciccs, pp. 935–939 (2019). <https://doi.org/10.1109/ICCS45141.2019.9065421>
13. Dagher, G.G., Marella, P.B., Milojkovic, M., Mohler, J.: Bron covote: secure voting system using ethereum's blockchain. ICISSP 2018—Proceedings of 4th International Conference on Information System Security and Privacy, vol. 2018-Janua, pp. 96–107 (2018). <https://doi.org/10.5220/0006609700960107>
14. Mofokeng, N.E.M., Matima, T.K.: Future tourism trends: Utilizing non-fungible tokens to aid wildlife conservation. *Afr. J. Hosp. Tour. Leis.* **7**(4) (2018)