Scientific Journal of Metaverse and Blockchain Technologies
Volume 1 | Issue 1 | Jul - Dec 2023
DOI: https://doi.org/10.36676/sjmbt.v1i1.10

Integration of IoT and Blockchain for user Authentication

Mandeep Gupta
9.mandeep@gmail.com
https://orcid.org/0009-0005-3542-1408

Abstract: The proliferation of Internet of Things (IoT) devices has ushered in a new era of connectivity, necessitating robust solutions for user authentication to address security and trust challenges. This paper explores an innovative approach to user authentication in IoT environments by leveraging the unique capabilities of Non-Fungible Tokens (NFTs) and 9NM (9NFTMANIA) tokens, with a specific focus on utilizing contract addresses. The proposed system involves the tokenization of user identities through the creation of contract addresses on blockchain networks. Each user is assigned a unique digital identity represented by an NFT or 9NM token, providing a tamper-proof association between the user and their cryptographic keys. Blockchain smart contracts are employed to manage authentication processes, dictating access control policies based on the user's contract address. The research underscores the importance of industry-wide collaboration to develop common standards for user authentication in IoT environments. By offering a comprehensive exploration of user authentication in IoT using contract address-based NFTs and 9NM tokens that are developed on Satoshi core based blockchain, this paper contributes to the advancement of secure and user-centric practices in the rapidly evolving landscape of IoT technology. The proposed framework not only enhances the overall security posture of IoT networks but also lays the foundation for a more transparent and interoperable authentication ecosystem. This paper has discussed the mechanism where web 3.0 based programming is made using Javascript, Python, ASP.NET and PHP for user authentication considering presence of smart contracts in user wallet.

Keyword: IoT, Blockchain, Satoshi core chain, Contract address, NFT, 9NFTMANIA token, User authentication, Javascript, Python, ASP.NET, PHP

[1] INTRODUCTION

The Internet of Things (IoT) refers to the network of interconnected devices, objects, and systems that communicate and share data with each other over the internet. These devices can range from everyday objects such as refrigerators and thermostats to industrial machines and wearable devices. The proliferation of IoT has led to an exponential increase in the volume of data generated by these devices, raising concerns about data security, privacy, and the integrity of the information exchanged. Blockchain technology has emerged as a potential solution to address the security and trust issues associated with IoT. Blockchain is a decentralized and distributed ledger that records transactions across a network of computers in a secure and transparent manner. Each block in the chain contains a timestamped and encrypted record of a transaction, and once added to the chain, it becomes virtually immutable. This decentralized nature of blockchain enhances the security and integrity of data, making it resistant to tampering or unauthorized access. Integrating blockchain with IoT can provide several advantages. One key benefit is enhanced security through cryptographic algorithms and consensus mechanisms, ensuring that the data generated and exchanged between IoT devices remains confidential and trustworthy. Blockchain also enables the creation of smart contracts, self-executing contracts with predefined rules, which can automate and enforce agreements between IoT devices without the need for intermediaries.

Furthermore, blockchain can address the issue of data ownership and control in IoT ecosystems. With blockchain, individuals have greater control over their data and can grant permission for specific devices or applications to access it. This empowers users to manage their privacy preferences and share data selectively, fostering a more transparent and user-centric approach to IoT data management.

However, the integration of blockchain with IoT is not without challenges. The scalability and energy consumption of blockchain networks, especially in the context of IoT devices with limited resources, remain significant concerns. Additionally, standardization and interoperability issues need to be addressed to ensure seamless communication and integration between diverse IoT devices and blockchain platforms.

Published: 07/12/2023

The combination of Internet of Things and blockchain holds great promise for addressing the security, privacy, and trust challenges associated with the massive amounts of data generated by interconnected devices. As the technology continues to evolve, overcoming scalability issues and establishing industry standards will be crucial for unlocking the full potential of this synergistic relationship.

The integration of the Internet of Things (IoT) with blockchain technology has emerged as a powerful combination that addresses various challenges associated with security, privacy, and data integrity in IoT ecosystems. Here's a brief overview of how these two technologies intersect and complement each other:

1. Security Enhancement:

- Decentralized Security: Blockchain provides a decentralized and tamper-resistant ledger, ensuring the integrity of IoT data. Each block in the chain contains a cryptographic hash of the previous block, creating a secure and immutable record of transactions or data exchanges.

2. Data Integrity and Trust:

- Immutable Record: Blockchain's immutability ensures that once data is recorded, it cannot be altered. This feature is crucial for maintaining the integrity of data collected from IoT devices, as it prevents unauthorized tampering and enhances trust in the information generated by these devices.

3. Smart Contracts for Automation:

- Self-Executing Contracts: Smart contracts, which are self-executing contracts with the terms of the agreement directly written into code, can automate and enforce agreements within the IoT network. This automation reduces the need for intermediaries, streamlines processes, and enhances efficiency.

4. Identity and Access Management:

- Secure Identity: Blockchain facilitates secure and decentralized identity management for IoT devices. Each device can have a unique identity stored on the blockchain, reducing the risk of unauthorized access and enhancing overall network security.

5. Supply Chain Transparency:

- Traceability: Blockchain ensures transparency and traceability in the supply chain by recording every transaction or movement of goods. This is particularly beneficial for industries where provenance and traceability are critical, such as food and pharmaceuticals.

6. Data Monetization and Privacy:

- User Control: Blockchain allows users to have greater control over their data. With decentralized identity and permissioned access, users can decide who can access their data, and under what conditions, leading to potential new models for data monetization where users are compensated for sharing their information.

7. Scalability and Performance:

- Challenges: While the combination of IoT and blockchain offers numerous benefits, challenges such as scalability and performance issues need to be addressed. The decentralized nature of blockchain can introduce latency, and efforts are ongoing to develop solutions that ensure efficient and scalable integration.

The convergence of IoT and blockchain holds significant promise, especially in industries where security, transparency, and trust are paramount. As both technologies continue to evolve, their synergistic applications are likely to expand, driving innovation in various sectors.

[2] BLOCKCHAIN

Blockchain is a revolutionary and decentralized technology that serves as a distributed ledger to record transactions across a network of computers. It was originally conceptualized as the underlying technology for the cryptocurrency Bitcoin, introduced in a 2008 whitepaper by an unknown person or group using the pseudonym Satoshi Nakamoto. Since then, blockchain technology has evolved and found applications far beyond its initial use case in cryptocurrency.

Key Concepts:

1. Decentralization: One of the fundamental features of blockchain is its decentralized nature. Instead of relying on a central authority, blockchain distributes the control and validation of transactions across a network of nodes (computers). This decentralization enhances security and resilience.
2. Distributed Ledger: The blockchain itself is a distributed ledger that contains a chain of blocks, each storing a list of transactions. These blocks are linked and secured through cryptographic hashes. Once a block is added to the chain, it is extremely challenging to alter, ensuring the integrity of the transaction history.
3. Consensus Mechanisms: To validate and agree on the state of the blockchain, consensus mechanisms are employed. The most well-known is Proof of Work (PoW), used by Bitcoin, where nodes (miners) solve complex mathematical problems to add a new block. Other consensus mechanisms include Proof of Stake (PoS) and Delegated Proof of Stake (DPoS), each with its own approach to validating transactions.
4. Cryptographic Security: Cryptography plays a crucial role in securing transactions and ensuring the privacy of participants. Public and private keys are used to sign and verify transactions, providing a secure means of authentication.

Applications Beyond Cryptocurrency:

1. Smart Contracts: Smart contracts are self-executing contracts with coded rules. They automatically enforce and execute terms when predefined conditions are met. This feature enables trustless and transparent execution of agreements.
2. Supply Chain Management: Blockchain is applied to supply chain management to enhance transparency and traceability. Each step in the supply chain, from manufacturing to distribution, can be recorded on the blockchain, reducing fraud and ensuring product authenticity.
3. Identity Management: Blockchain can be utilized for secure and decentralized identity management. Users have control over their personal information, and the immutability of the blockchain helps prevent identity theft.
4. Healthcare Records: Storing healthcare records on a blockchain ensures secure and interoperable access to patient data. Patients, healthcare providers, and insurers can have real-time and secure access to medical histories.
5. Financial Services: Blockchain has significant applications in the financial sector, offering faster and more secure cross-border transactions. It also facilitates the creation of digital assets and tokenized securities.

6. Tokenization of Assets: Assets, both physical and digital, can be represented as tokens on the blockchain. This opens up new possibilities for fractional ownership, increased liquidity, and more efficient trading.

Blockchain technology continues to evolve, and its potential applications are expanding across various industries. Its core principles of decentralization, transparency, and security make it a transformative force in reshaping how transactions and data are managed in the digital age.

[3] ROLE OF BLOCK CHAIN IN BUILDING CONTRACT FOR NFT AND CRYPTOTOKEN

Certainly, blockchain technology enables the creation and execution of smart contracts, which, in turn, can facilitate the generation and management of Non-Fungible Tokens (NFTs) and various types of cryptographic tokens, including cryptocurrencies or utility tokens. Here's an overview of how blockchain facilitates the smart contract creation of NFTs and crypto tokens:

1. Smart Contracts: Smart contracts are self-executing contracts with predefined rules and conditions. They run on blockchain networks and automatically execute actions when specific criteria are met. The decentralized and tamper-proof nature of blockchain ensures the trustworthiness and security of these contracts.
2. NFT Creation: NFTs are unique digital assets representing ownership or proof of authenticity of a specific item or content. These can include digital art, collectibles, or other unique digital entities. Smart contracts on blockchain platforms, such as Ethereum, are programmed to create and manage NFTs. The contract specifies the rules for ownership, transfer, and any associated metadata.
3. Token Standards: Various blockchain networks follow token standards that define the rules for creating fungible and non-fungible tokens. For example, the ERC-721 standard is widely used for NFTs, ensuring compatibility across different platforms. Similarly, ERC-20 is a common standard for fungible tokens.

4. Crypto Token Creation: Blockchain facilitates the creation of cryptographic tokens, commonly known as cryptocurrencies or utility tokens, through smart contracts. These tokens can represent a unit of value or serve a specific purpose within a decentralized application (DApp) or ecosystem. The rules governing the token, such as total supply and transferability, are encoded in the smart contract.

5. Ownership and Transferability: Smart contracts manage ownership and transferability of both NFTs and cryptographic tokens. The ownership of an NFT, for instance, is tied to a unique identifier on the blockchain, ensuring verifiable ownership and preventing duplication.

6. Decentralization and Security: The decentralized nature of blockchain ensures that smart contracts and the tokens they create are not controlled by a single entity. This decentralization enhances security, reduces the risk of fraud, and prevents censorship.

7. Interoperability: Blockchain networks that support token standards provide a level of interoperability, allowing tokens to be transferred and used across various applications and platforms that adhere to the same standards. This interoperability contributes to the widespread adoption of tokens created through smart contracts.

Blockchain's ability to execute smart contracts is pivotal in the creation and management of NFTs and cryptographic tokens. It provides a secure, transparent, and decentralized framework for representing and transferring ownership of unique digital assets and tokens within the evolving landscape of blockchain-based applications and ecosystems.

[4] USER AUTHENTICATION IN IOT BASED ON SMART CONTRACT ADDRESS

Authentication of users in the Internet of Things (IoT) based on contract addresses involve leveraging blockchain technology to enhance security and trust within IoT ecosystems. In this context, each user is associated with a unique digital identity represented by a contract address on the blockchain.

1. User Registration and Identity Tokenization: When a user joins an IoT network, their identity is tokenized through the creation of a unique contract address on the blockchain. This process typically involves the generation of cryptographic keys that serve as the user's digital signature, ensuring a secure and tamper-proof association between the user and their contract address.

2. Blockchain Smart Contracts for Authentication: Smart contracts are employed to manage the authentication process. These contracts define the rules and conditions for user access to IoT devices or services. Authentication can be based on cryptographic keys, digital signatures, or other secure methods tied to the user's contract address.

3. Access Control and Permissions: The contract address associated with each user dictates their access privileges within the IoT network. Smart contracts enforce access control policies, ensuring that only authenticated users with valid contract addresses can interact with specific devices or services. This enhances the overall security of the IoT ecosystem.

4. Biometric or Multi-Factor Authentication Integration: To further strengthen user authentication, biometric data or multi-factor authentication methods can be integrated into the process. These additional layers of security can be linked to the user's contract address, providing a more robust and user-friendly authentication experience.

5. Token Revocation and User Management: In case of a compromised or lost device, or if a user's access needs to be revoked, blockchain smart contracts facilitate the seamless revocation or transfer of the associated contract address. This ensures that the user's identity is promptly updated and reflects the current status within the IoT network.

6. Decentralized Trust and Transparency: By distributing user authentication across the blockchain network, decentralized trust is established. Users can be confident that their identity and access rights are secured without reliance on a central authority. The transparency of the blockchain ensures that all authentication-related transactions are visible and auditable.

7. Interoperability and Standards: To promote widespread adoption and compatibility across various IoT platforms and blockchain networks, industry-wide standards for user authentication based on contract addresses need to be established. Interoperability standards contribute to a seamless and consistent user experience across diverse IoT environments.

Leveraging contract addresses on the blockchain for user authentication in the IoT offers a secure, transparent, and decentralized approach. It not only enhances the overall security posture of IoT networks but also provides users with greater control over their identities and access rights. Standardization efforts and collaboration within the industry will play a crucial role in the successful implementation and scalability of this authentication method.

[5] AUTHENTICATION IN IOT BY CONTRACT ADDRESS OF NFTS / 9NFTMANIA TOKEN

Authentication in the Internet of Things (IoT) based on the contract address of NFTs (Non-Fungible Tokens) or a 9NM token involves leveraging blockchain technology to enhance security and establish trust within IoT ecosystems. Here's an overview of how this authentication process could work:

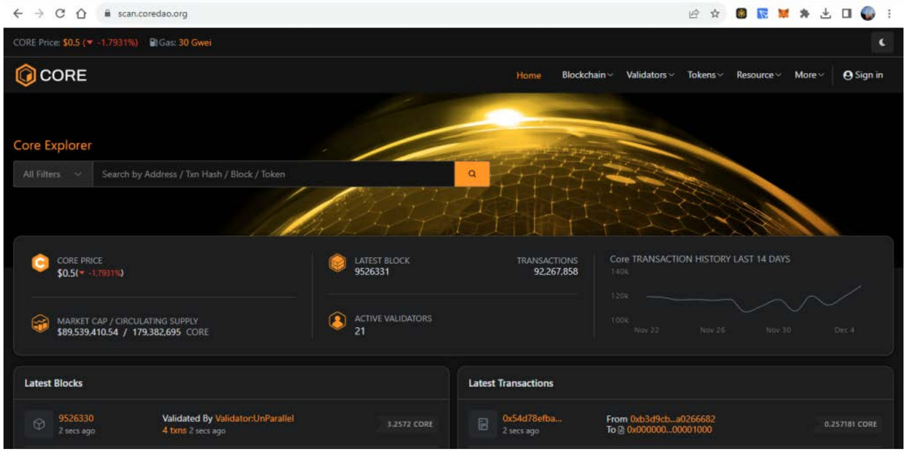
1. Tokenized Identity and Ownership: Each IoT device is assigned a unique digital identity in the form of an NFT or a 9NM token. This token serves as a representation of the device's ownership and authenticity on the blockchain.
2. Blockchain Smart Contracts: Smart contracts are programmable agreements that execute automatically when predefined conditions are met. In this context, smart contracts can be created to manage the authentication and authorization of IoT devices.
3. Device Registration: During the manufacturing or initialization process, IoT devices are registered on the blockchain. This registration involves associating the device's unique identifier or public key with the corresponding NFT or 9NM token contract address.
4. Access Control and Permissions: Smart contracts can define access control rules based on the ownership status reflected in the associated tokens. Only devices with valid and authenticated tokens will be granted permission to communicate or interact within the IoT network.
5. Token Transfer and Revocation: Ownership of IoT devices can be transferred securely through the transfer of NFTs or 9NM tokens. In case of device loss, theft, or decommissioning, token revocation or transfer can be initiated to maintain a secure and up-to-date registry of authorized devices.
6. Decentralized Authentication: The decentralized nature of blockchain ensures that authentication is not reliant on a centralized authority. This reduces the risk of a single point of failure or a central repository that could be compromised.
7. Auditability and Transparency: All transactions related to device authentication, ownership transfers, or access control are recorded on the blockchain. This creates a transparent and auditable trail of events, enhancing the overall security and accountability of the IoT ecosystem.
8. Interoperability and Standards: To ensure widespread adoption and compatibility across various IoT platforms and blockchain networks, there should be efforts to establish interoperability standards for tokenized IoT identities.

While using NFTs or 9NM tokens for IoT authentication offers several advantages, including improved security, transparency, and decentralized control, it's important to consider the scalability of blockchain networks and the potential energy consumption associated with certain consensus mechanisms. Additionally, the adoption of such authentication methods would benefit from industry-wide collaboration and the development of common standards.

[6] TECHNICAL WORK

6.1 ACCESSING CONTRACT ADDRESS

In order to access the contract address of NFT or 9NFTMANIA token that are built on satoshi core, there is need to visit website https://scan.coredao.org.



CORE Price: $0.5 (▾ -1.7931%) Gas: 30 Gwei

CORE PRICE
$0.5(▾ -1.7931%)

LATEST BLOCK
9526331

TRANSACTIONS
92,267,858

MARKET CAP / CIRCULATING SUPPLY
$89,539,410.54 / 179,382,695 CORE

ACTIVE VALIDATORS
21

Latest Blocks

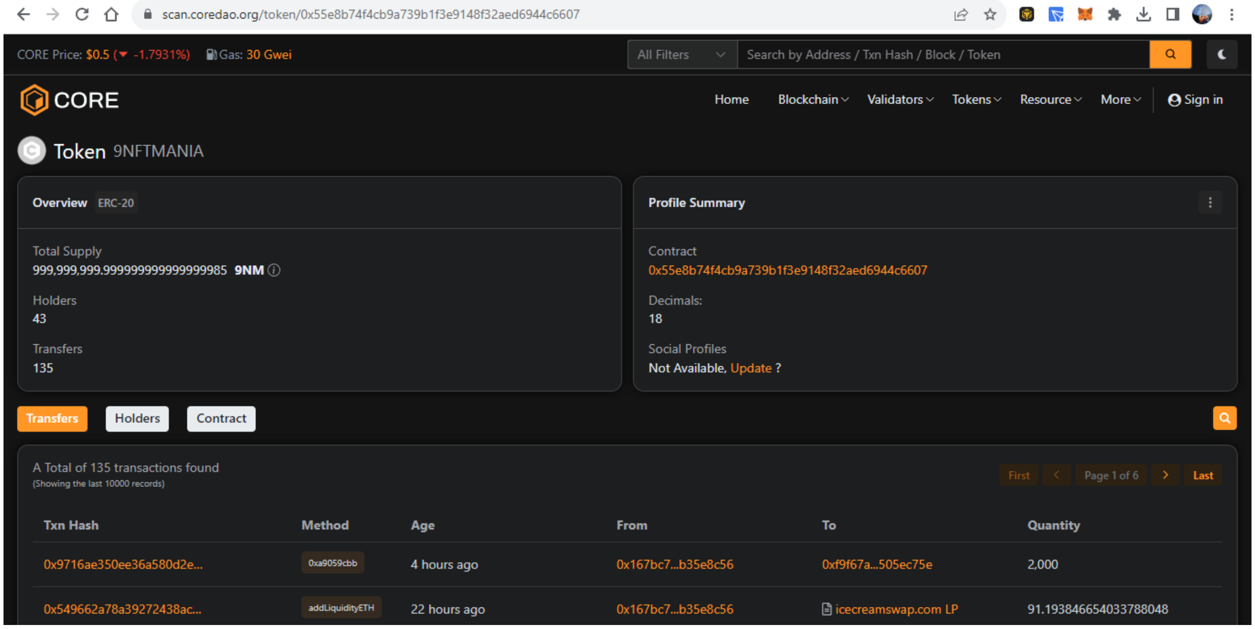
9526330
2 secs ago
Validated By Validator:UnParallel
4 txns 2 secs ago
3.2572 CORE

Latest Transactions

0x54d78efba...
2 secs ago
From 0xb3d9cb...a0266682
To 0x000000...00001000
0.257181 CORE

Fig. 1 View of scan.coredao.org

Then set the 9NFTMANIA in search box to get relevant information. After fetching information pages look like this.


scan.coredao.org/token/0x55e8b74f4cb9a739b1f3e9148f32aed6944c6607
All Filters
Search by Address / Txn Hash / Block / Token
CORE
Home
Blockchain
Validators
Tokens
Resource
More
Sign in
Transfers
Holders
Contract
First
Page 1 of 6
Last

CORE Price: $0.5 (▾ -1.7931%) Gas: 30 Gwei

Token 9NFTMANIA

Overview ERC-20

Total Supply
999,999,999.999999999999999985 9NM

Holders
43

Transfers
135

Profile Summary

Contract
0x55e8b74f4cb9a739b1f3e9148f32aed6944c6607

Decimals:
18

Social Profiles
Not Available, Update ?

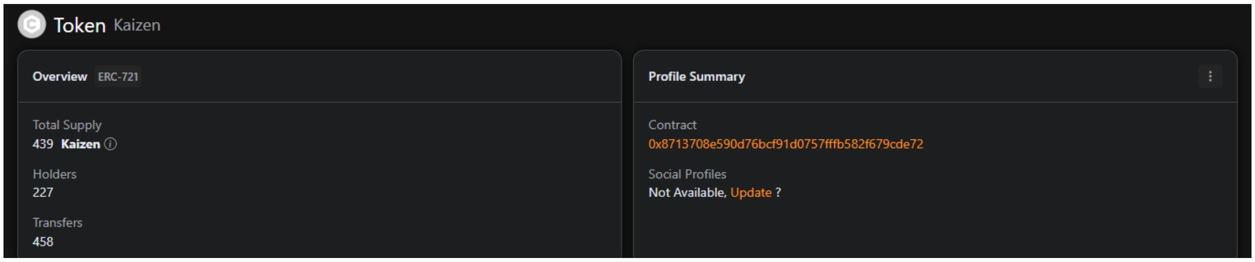
A Total of 135 transactions found
(Showing the last 10000 records)

Txn Hash	Method	Age	From	To	Quantity
0x9716ae350ee36a580d2e...	0xa9059cbb	4 hours ago	0x167bc7...b35e8c56	0xf9f67a...505ec75e	2,000
0x549662a78a39272438ac...	addLiquidityETH	22 hours ago	0x167bc7...b35e8c56	icecreamswap.com LP	91.193846654033788048

Fig. 2 View of 9NFTMANIA

In profile summary there is detail of contract address : 0x55E8B74F4CB9a739b1f3E9148F32aed6944c6607

In same way, contract address of any NFT could be found. Following figures are presenting the contract address of different NFT.



Token Kaizen

Overview ERC-721

Total Supply
439 Kaizen

Holders
227

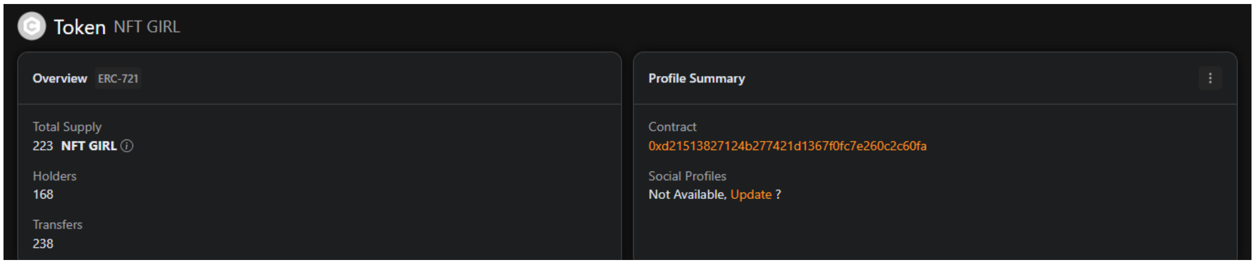
Transfers
458

Profile Summary

Contract
0x8713708e590d76bcf91d0757fffb582f679cde72

Social Profiles
Not Available, Update ?

Fig. 3 Kaizen contract details



Token NFT GIRL

Overview ERC-721

Total Supply
223 NFT GIRL

Holders
168

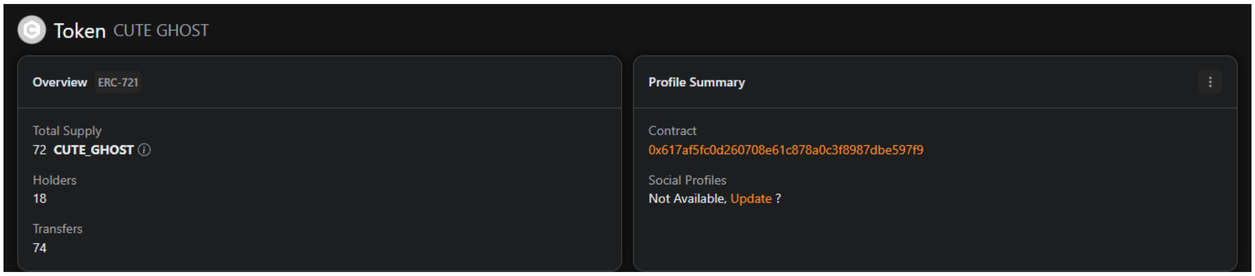
Transfers
238

Profile Summary

Contract
0xd21513827124b277421d1367f0fc7e260c2c60fa

Social Profiles
Not Available, Update ?

Fig. 4 NFT Girl contract details



Token CUTE GHOST

Overview ERC-721

Total Supply
72 CUTE_GHOST

Holders
18

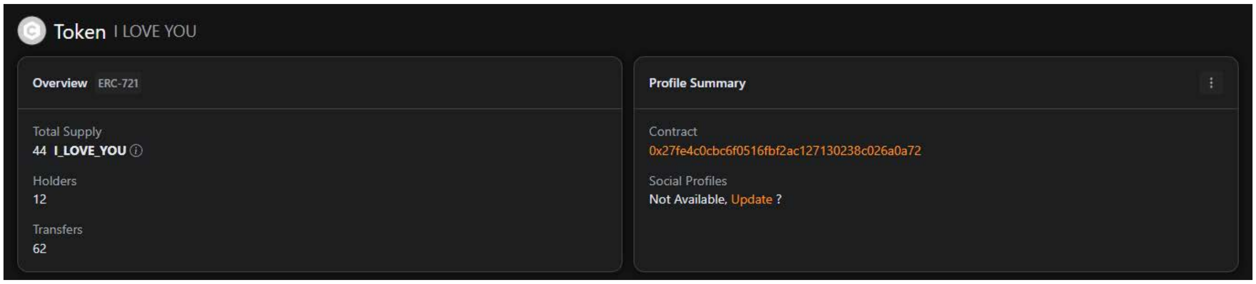
Transfers
74

Profile Summary

Contract
0x617af5fc0d260708e61c878a0c3f8987dbe597f9

Social Profiles
Not Available, Update ?

Fig. 5 Cute Ghost contract details



Token I LOVE YOU

Overview ERC-721

Total Supply
44 I_LOVE_YOU

Holders
12

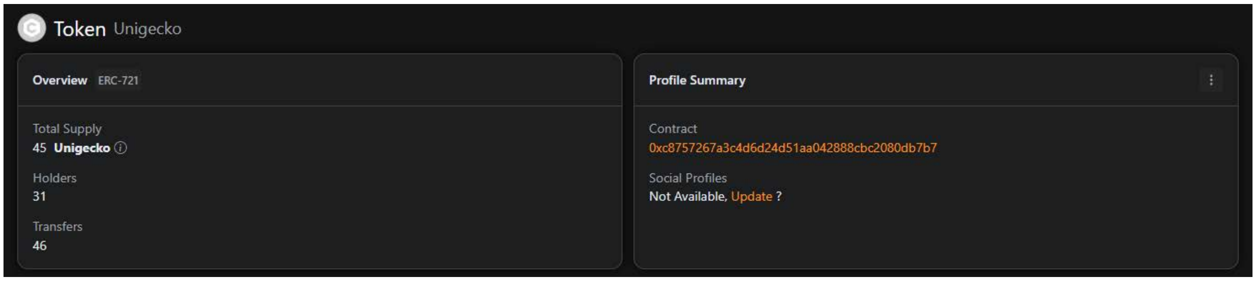
Transfers
62

Profile Summary

Contract
0x27fe4c0cbc6f0516fbf2ac127130238c026a0a72

Social Profiles
Not Available, Update ?

Fig. 6 I LOVE YOU contract details



Token Unigecko

Overview ERC-721

Total Supply
45 Unigecko

Holders
31

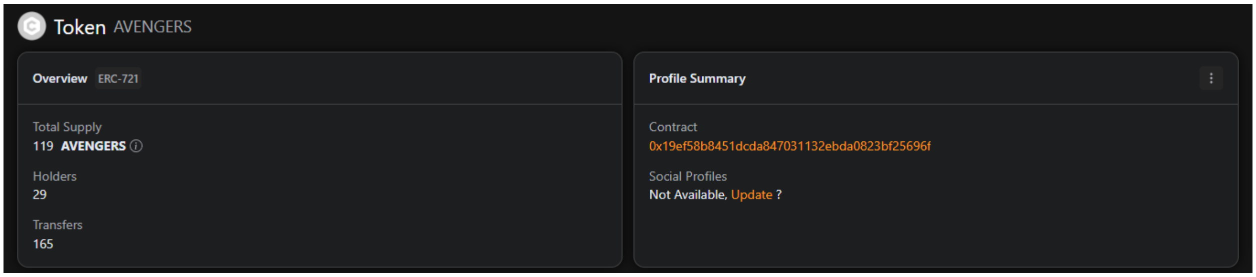
Transfers
46

Profile Summary

Contract
0xc8757267a3c4d6d24d51aa042888cbc2080db7b7

Social Profiles
Not Available, Update ?

Fig. 7 Unigecko contract details



Token AVENGERS

Overview ERC-721

Total Supply
119 AVENGERS

Holders
29

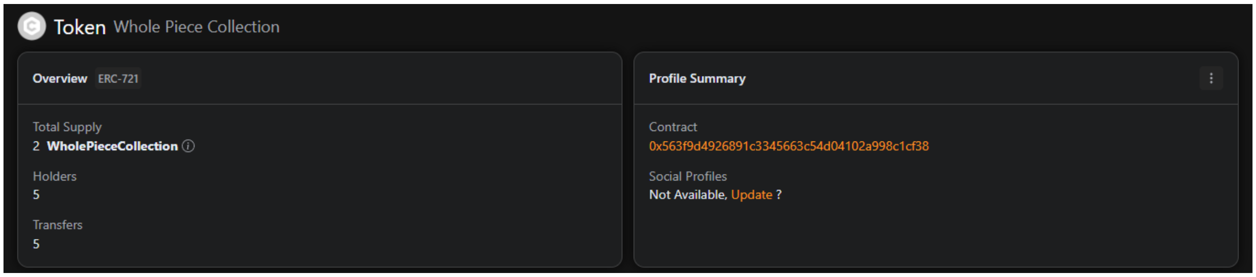
Transfers
165

Profile Summary

Contract
0x19ef58b8451dcda847031132ebda0823bf25696f

Social Profiles
Not Available, Update ?

Fig. 8 Avengers contract details



Token Whole Piece Collection

Overview ERC-721

Total Supply
2 WholePieceCollection

Holders
5

Transfers
5

Profile Summary

Contract
0x563f9d4926891c3345663c54d04102a998c1cf38

Social Profiles
Not Available, Update ?

Fig. 9 Whole Piece collection

9NFTMANIA is ERC20 where as NFTs are ERC 721. The difference between ERC20 and ERC721 is shown below ERC-20 and ERC-721 are two distinct standards for tokens on the Ethereum blockchain, each serving different purposes. Here are the key differences between ERC-20 and ERC-721:

1. Token Type:

- ERC-20: ERC-20 tokens are fungible, meaning each token is identical and interchangeable with every other token of the same type. These tokens represent a unit of value and are commonly used for cryptocurrencies and utility tokens within decentralized applications (DApps).
- ERC-721: ERC-721 tokens are non-fungible tokens (NFTs), and each token is unique. NFTs are often used to represent ownership of digital or physical assets, such as digital art, collectibles, or real estate, where each token has distinct characteristics and cannot be replaced on a one-to-one basis.

2. Use Case:

- ERC-20: ERC-20 tokens are suitable for representing any divisible and interchangeable asset, making them well-suited for applications like cryptocurrencies, utility tokens, and ICOs (Initial Coin Offerings).
- ERC-721: ERC-721 tokens are designed for representing ownership of unique and indivisible assets. They are particularly popular in applications where each token needs to have distinct properties or represent ownership of a specific item.

3. Divisibility:

- ERC-20: ERC-20 tokens are divisible, meaning they can be divided into smaller units. For example, you can send a fraction of an ERC-20 token.
- ERC-721: ERC-721 tokens are indivisible, and each token represents a whole unit. You cannot send a fraction of an ERC-721 token; it is either owned in its entirety or not owned at all.

4. Standard Functions:

- ERC-20: ERC-20 tokens follow a set of standard functions, including those for transferring tokens, checking balances, and approving spending by another address. These functions make ERC-20 tokens easily compatible with various wallets and platforms.
- ERC-721: ERC-721 tokens have functions for transferring tokens, checking ownership, and getting token metadata. Additionally, ERC-721 tokens often implement optional functions for managing unique properties or characteristics of each token.

5. Examples:

- ERC-20: Examples of ERC-20 tokens include popular cryptocurrencies like Ethereum (ETH), Binance Coin (BNB), and utility tokens such as Chainlink (LINK).
- ERC-721: Examples of ERC-721 tokens include digital collectibles like CryptoKitties, digital art pieces, and virtual real estate tokens.

Understanding these differences is crucial when choosing the appropriate token standard based on the specific requirements of a decentralized application or use case. ERC-20 is more suitable for fungible assets, while ERC-721 is ideal for representing ownership of unique and non-fungible assets.

6.2 Writing code to verify the existence of 9NFT token or NFT in User's Metamask wallet to authenticate them

6.2.1 Javascript based implementation

Verifying the existence of 9NFT (Non-NFT Token) or NFT in a user's MetaMask wallet involves interacting with the Ethereum blockchain and reading the user's wallet balances. Here's JavaScript and the web3 library:

```
<!-- Include web3.js library -->
    <script src="https://cdn.jsdelivr.net/npm/web3@1.5.3/dist/web3.min.js"></script>

    <script>
     // Check if Web3 is injected by MetaMask
     if (typeof web3 !== 'undefined') {
      // Use MetaMask's provider
      web3 = new Web3(web3.currentProvider);

      // Check if MetaMask is connected
      web3.eth.net.isListening()
       .then(() => {
        console.log('Connected to MetaMask');

        // Replace 'yourTokenAddress' with the actual contract address of your 9NFT or NFT
        const contractAddress = 'yourTokenAddress';

        // Replace 'yourUserAddress' with the actual user's MetaMask wallet address
        const userAddress = 'yourUserAddress';

        // Check the balance of the user's wallet for the specified token
        web3.eth.getBalance(userAddress, (err, balance) => {
         if (err) {
          console.error('Error retrieving balance:', err);
         } else {
          console.log('ETH Balance:', web3.utils.fromWei(balance, 'ether'), 'ETH');
         }
        });

        // Check the balance of the specified token in the user's wallet
        const tokenContract = new web3.eth.Contract(abi, contractAddress);

        tokenContract.methods.balanceOf(userAddress).call((err, balance) => {
         if (err) {
          console.error('Error retrieving token balance:', err);
         } else {
          console.log('Token Balance:', balance, 'tokens');
         }
        });
       })
       .catch(() => {
        console.error('Not connected to MetaMask');
       });
     } else {
      console.error('MetaMask not found. Please install MetaMask to use this feature.');
     }
    </script>
```

6.2.2 ASP.NET based

To verify the existence of a 9NFT token or NFT in a user's MetaMask wallet and authenticate them using ASP.NET, you can use C# on the server side and interact with the Ethereum blockchain. Below is a basic example using Nethereum, a popular Ethereum library for .NET:

1. Install the Nethereum.Web3 NuGet package:

```
Install-Package Nethereum.Web3
```

2. Use the following C# code in your ASP.NET application

```
using System;
using System.Threading.Tasks;
using Nethereum.Web3;
using Nethereum.Web3.Accounts;

public class NftVerificationService
{
    // Replace 'yourRpcUrl' with the actual RPC endpoint of your Ethereum node
    private const string RpcUrl = "yourRpcUrl";

    // Replace 'yourTokenAddress' with the actual contract address of your 9NFT or NFT
    private const string TokenAddress = "yourTokenAddress";

    public async Task<bool> VerifyNftOwnership(string userAddress)
    {
        try
        {
            var web3 = new Web3(new Account(), RpcUrl);

            // Check the balance of ETH in the user's wallet
            var ethBalance = await web3.Eth.GetBalance.SendRequestAsync(userAddress);
            Console.WriteLine($"ETH Balance: {Web3.Convert.FromWei(ethBalance)} ETH");

            // Check the balance of the specified token in the user's wallet
            var contract = web3.Eth.GetContract(TokenAbi, TokenAddress);
            var tokenBalance = await contract.GetFunction("balanceOf").CallAsync<int>(userAddress);
            Console.WriteLine($"Token Balance: {tokenBalance} tokens");

            // Perform additional authentication logic based on token ownership

            return true; // Authentication successful
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
            return false; // Authentication failed
        }
    }

    // Replace 'yourTokenAbi' with the actual ABI (Application Binary Interface) of your smart contract
    private static string TokenAbi => "yourTokenAbi";
}
```

6.2.3 Python based

To verify the existence of a 9NFT token or NFT in a user's MetaMask wallet and authenticate them using Python, you can use the web3.py library to interact with the Ethereum blockchain.

1. Install the web3.py library:
   pip install web3

2. Use the following Python code:

```
from web3 import Web3

# Replace 'your_rpc_url' with the actual RPC endpoint of your Ethereum node
rpc_url = 'your_rpc_url'

# Replace 'your_token_address' with the actual contract address of your 9NFT or NFT
token_address = 'your_token_address'

# Replace 'your_user_address' with the actual user's MetaMask wallet address
user_address = 'your_user_address'

# Connect to the Ethereum node
web3 = Web3(Web3.HTTPProvider(rpc_url))

if web3.isConnected():
    print("Connected to Ethereum node")

    # Check ETH balance in the user's wallet
    eth_balance = web3.eth.getBalance(user_address)
    print(f"ETH Balance: {web3.fromWei(eth_balance, 'ether')} ETH")

    # Load the contract ABI
    # Replace 'your_token_abi' with the actual ABI (Application Binary Interface) of your smart contract
    token_abi = 'your_token_abi'

    # Create a contract object
    token_contract = web3.eth.contract(address=token_address, abi=token_abi)

    # Check the balance of the specified token in the user's wallet
    token_balance = token_contract.functions.balanceOf(user_address).call()
    print(f"Token Balance: {token_balance} tokens")

    # Perform additional authentication logic based on token ownership
    if token_balance > 0:
        print("Authentication successful")
    else:
        print("Authentication failed")

else:
    print("Not connected to Ethereum node")
```

6.2.4 PHP CODE

To verify the existence of a 9NFT token or NFT in a user's MetaMask wallet and authenticate them using PHP, you can use the web3.php library to interact with the Ethereum blockchain.

Step 1 Install the web3.php library using Composer:

```
composer require web3p/web3.php
```

Step 2 Use the following PHP code:

```
<?php

require 'vendor/autoload.php';

use Web3\Web3;
use Web3\Contract;
use Web3\Utils;

// Replace 'your_rpc_url' with the actual RPC endpoint of your Ethereum node
$rpcUrl = 'your_rpc_url';

// Replace 'your_token_address' with the actual contract address of your 9NFT or NFT
$tokenAddress = 'your_token_address';

// Replace 'your_user_address' with the actual user's MetaMask wallet address
$userAddress = 'your_user_address';

// Connect to the Ethereum node
$web3 = new Web3($rpcUrl);

if ($web3->provider->connected()) {
    echo "Connected to Ethereum node\n";

    // Check ETH balance in the user's wallet
    $ethBalance = $web3->eth->getBalance($userAddress);
    echo "ETH Balance: " . Utils::fromWei($ethBalance, 'ether') . " ETH\n";

    // Load the contract ABI
    // Replace 'your_token_abi' with the actual ABI (Application Binary Interface) of your smart contract
    $tokenAbi = 'your_token_abi';

    // Create a contract object
    $tokenContract = new Contract($web3->provider, $tokenAbi);
    $tokenContract->at($tokenAddress);

    // Check the balance of the specified token in the user's wallet
    $tokenBalance = $tokenContract->call('balanceOf', $userAddress);
    echo "Token Balance: $tokenBalance tokens\n";

    // Perform additional authentication logic based on token ownership
    if ($tokenBalance > 0) {
        echo "Authentication successful\n";
    } else {
        echo "Authentication failed\n";
    }

} else {
    echo "Not connected to Ethereum node\n";
}
```

DOI: https://doi.org/10.36676/sjmbt.v1i1.10

The your_rpc_url in the provided examples represents the RPC (Remote Procedure Call) endpoint of an Ethereum node. An Ethereum node is a computer that participates in the Ethereum network, maintains the Ethereum blockchain, and facilitates communication with the blockchain.

When interacting with the Ethereum blockchain through a programming language like JavaScript, Python, or PHP, you need to connect to an Ethereum node. The RPC URL is the address of the Ethereum node's RPC server, and it allows your code to send requests to and receive responses from the Ethereum node.

[7] SCOPE OF RESEARCH

However, the technical implementation of these function might be complex for beginners but present paper is opting to provide programmers an idea how web3.0 based script could authenticate users considering the presence of particular token or NFT in their metamask wallet. This paper would allow web developer to initiate their premium web services for specific NFT holders. Programmer may have following benefit in future:

1. Web Programmers may develop their own NFT and smart contract and program web 3.0 based web module that would be capable to provide some special service to holders of those NFT or tokens
2. Web Programmers may also make their own web 3.0 based metaverse module where token or NFT holder may have special privileges.
3. Web developers may create commercial websites where NFT or Token holder may get special discount.

References

1. Gupta, M., Gupta, D., & Duggal, A. (2023). NFT Culture: A New Era. Scientific Journal of Metaverse and Blockchain Technologies, 1(1), 57–62. https://doi.org/10.36676/sjmbt.v1i1.08
2. M. Gupta, “Reviewing the Relationship Between Blockchain and NFT With World Famous NFT Market Places”, SJMBT, vol. 1, no. 1, pp. 1–8, Dec. 2023.
3. R. Gupta, M. Gupta, and D. Gupta, “Role of Liquidity Pool in Stabilizing Value of Token”, SJMBT, vol. 1, no. 1, pp. 9–17, Dec. 2023.
4. M. GUPTA and D. Gupta, “Investigating Role of Blockchain in Making your Greetings Valuable”, URR, vol. 10, no. 4, pp. 69–74, Dec. 2023.
5. R. Issalh, A. Gupta, and M. Gupta, “PI NETWORK : A REVOLUTION”, SJMBT, vol. 1, no. 1, pp. 18–27, Dec. 2023.
6. A. Duggal, M. Gupta, and D. Gupta, “SIGNIFICANCE OF NFT AVTAARS IN METAVERSE AND THEIR PROMOTION: CASE STUDY”, SJMBT, vol. 1, no. 1, pp. 28–36, Dec. 2023.
7. M. Gupta, “Say No to Speculation in Crypto market during NFT trades: Technical and Financial Guidelines”, SJMBT, vol. 1, no. 1, pp. 37–42, Dec. 2023.
8. A. Singla, M. Singla, and M. Gupta, “Unpacking the Impact of Bitcoin Halving on the Crypto Market: Benefits and Limitations”, SJMBT, vol. 1, no. 1, pp. 43–50, Dec. 2023.
9. I. Gupta and P. Jain, “EXPECTED IMPACT OF DECENTRALIZATION USING BLOCKCHAIN BASED TECHNOLOGIES”, SJMBT, vol. 1, no. 1, pp. 51–56, Dec. 2023.
10. D. Gupta and S. Gupta, “Exploring world famous NFT Scripts: A Global Discovery”, SJMBT, vol. 1, no. 1, pp. 63–71, Dec. 2023.