UNIVERSITY OF WESTMINSTER

5/13/2024

Blockchain Technologies and Cryptocurrencies

MSc FinTech with Business Analytics

Versha Sandesh

Student Id: w2039582

Module Code: 7FNCE042W

Module Leader: Dr. Hui Gong

Word Count: About 4200, excluding Cover Page, Table of Contents and Figures and References.

TABLE OF CONTENTS

Table of Figures

Q1. BITCOIN

A. SCENARIO: COMPANY'S CEO WITH THREE OTHER BUSINESS PARTNERS

As the CEO of a company with three other business partners (Alice, Bob, and Coo), each holding a 20% stake, the control and management of the company's Bitcoin wallet and keys is a crucial matter that requires careful consideration to ensure transparency, security, and trust among all parties involved.

(I) WHO SHOULD CONTROL THE KEYS?

There are several approaches to distribute the keys among the stakeholders:

1. Each partner holds one key (4 keys total), and a certain number of signatures (e.g., 3 out of 4) is required to authorise transactions. This approach ensures that no single partner can act unilaterally, promoting transparency and accountability.
2. Divide the keys among the partners based on their ownership percentages. For example, the CEO with a 40% stake could hold two keys, while Alice, Bob, and Coo each have one key. Again, a predetermined number of signatures would be required for transactions.
3. Introduce an additional neutral third party (e.g., a trusted advisor or legal entity) to hold one of the keys, reducing the risk of any single partner gaining control over the funds.

Managing a company's Bitcoin wallet involves multiple stakeholders, each acting as a "general" where consensus is vital for transaction authorisation and wallet management. This multi-stakeholder setup of managing the Bitcoin wallet and ensuring consensus among all parties mirrors the challenges the Byzantine Generals' Problem poses.

The Byzantine Generals' Problem serves as a key analogy for the need for consensus in distributed ledger technology (DLT), addressing challenges in achieving agreement within a system where some nodes (or "generals") may be faulty or malicious. It aims to facilitate agreement among loyal generals despite potential interference from traitorous ones who may send conflicting messages.

In blockchain networks like Bitcoin, the 51% attack closely relates to the Byzantine Generals' Problem, where a malicious actor or group gains majority control of hashing power, compromising network security and integrity by manipulating transactions or hindering confirmations.

To address these challenges, specific consensus algorithms must be adopted, allowing nodes to update the ledger securely. The goal is to ensure that even amidst malicious actions or false information from some stakeholders, honest ones can still reach a consensus on the correct course of action (Gong, 2024).

(II) IS THERE A SOLUTION TO THIS PROBLEM?

Byzantine Fault Tolerance (BFT) empowers distributed systems to uphold consensus despite the presence of faulty or malicious nodes, making them resilient against crashes and malicious behaviour. Employing a multi-signature wallet, alongside appropriate governance frameworks and policies, offers an effective solution to mitigate these risks. In such a setup, each partner holds a key, necessitating multiple keys for transaction authorisation, thereby preventing abuse and malicious actions while safeguarding against the 51% attack in blockchain networks (Lamport, et al., n.d.).

Specifically, for the given scenario with four stakeholders (CEO, Alice, Bob, and Coo), one approach could be to create a 3-out-of-4 multi-signature wallet, where any three signatures are required to authorise a transaction. This way, no single stakeholder can unilaterally control the funds, requiring collusion between at least two stakeholders to misuse them.

Satoshi Nakamoto devised the Proof of Work (PoW) mechanism to achieve consensus in the Bitcoin blockchain, addressing the Byzantine Generals' Problem by facilitating decentralised agreement among nodes, even in the face of potential dishonesty. Miners compete to solve intricate mathematical puzzles using computational power in PoW. This process, termed "proof of work," involves hashing ledger data with a nonce (a random number) to find a hash value meeting network protocol criterion.

PoW prevents dominance by any single entity by necessitating computational effort and maintains a fair playing field for miners. This decentralised approach safeguards against manipulation of the ledger, effectively addressing the Byzantine Generals' Problem and upholding Bitcoin blockchain integrity and security.

B. HOW CAN THEY PROVE THEIR IDENTITY?

Cryptographic techniques and algorithmic solutions can be employed to ensure secure communication and decision-making in scenarios where trust among participants is limited or compromised. To prove their identity in the multi-stakeholder Bitcoin wallet scenario, the

stakeholders (CEO, Alice, Bob, and Coo) can employ various methods, depending on the level of security and trust required. Common approaches include:

1. Each stakeholder possesses a unique digital signature using a cryptographic key pair (public and private keys) linked to their identity. Transactions or messages signed with their private key can be authenticated using their corresponding public key, proving their authenticity.
2. Stakeholders can utilise multi-factor authentication methods (MFA), such as biometric authentication (e.g., fingerprint, facial recognition), hardware security keys, or one-time passwords (OTPs) generated by mobile applications or dedicated hardware devices.
3. Third-party identity verification services, like those provided by regulated entities like banks or government agencies, can be employed to verify the stakeholders' identities through know-your-customer (KYC) processes.
4. Trusted third parties like legal firms or notaries can attest to stakeholders' identities and provide signed documents or digital certificates confirming their identities.
5. For high-stakes scenarios, stakeholders can opt for in-person verification processes, verifying identities through physical documentation and personal interactions.

C. IMPLEMENTING THE CASE IN PYTHON

1: Install the 'bitcoin' Python library, which provides various functions and utilities for working with Bitcoin.

```
pip install bitcoin
```

```
Requirement already satisfied: bitcoin in c:\users\sande\anaconda3\lib\site-packages (1.1.42)
Note: you may need to restart the kernel to use updated packages.
```

```
from bitcoin import *
```

2: Import all functions and classes from the bitcoin library, making them available for use in the script.

3: Define the stakeholders and their respective ownership percentages in the company. In this case, the CEO owns 40%, while Alice, Bob, and Coo each own 20%.

```
# Define the stakeholders and their respective ownership percentages
stakeholders = {
    "CEO": 0.4,
    "Alice": 0.2,
    "Bob": 0.2,
    "Coo": 0.2
}
```

4: Set the number of required signatures for authorising transactions. Here, it is set to 3, meaning that at least three stakeholders must sign a transaction to be valid.

```
# Define the required number of signatures for authorizing transactions
required_signatures = 3  # Adjust this value as needed
```

```
# Generate private keys for each stakeholder
ceo_private_key = random_key()
alice_private_key = random_key()
bob_private_key = random_key()
coo_private_key = random_key()
```

5: Generate random private keys for each stakeholder using the random_key() function from the bitcoin library.

```
# Generate public keys from private keys
ceo_public_key = privtopub(ceo_private_key)
alice_public_key = privtopub(alice_private_key)
bob_public_key = privtopub(bob_private_key)
coo_public_key = privtopub(coo_private_key)
```

6: Generate the corresponding public keys from the private keys using the privtopub() function from the bitcoin library.

```
# Generate Bitcoin addresses from public keys
ceo_address = pubtoaddr(ceo_public_key)
alice_address = pubtoaddr(alice_public_key)
bob_address = pubtoaddr(bob_public_key)
coo_address = pubtoaddr(coo_public_key)
```

7: Generate Bitcoin addresses from the public keys using the pubtoaddr() function from the bitcoin library.

8: This class defines a MultisigWallet object. The _init_ method accepts stakeholder public keys, the required number of signatures, and the total number of stakeholders as arguments. It constructs a multi-signature script using the mk_multisig_script() function from the bitcoin library and derives the corresponding wallet address using the scriptaddr() function.

```
# Define MultisigWallet class
class MultisigWallet:
    def __init__(self, stakeholder_keys, required_signatures, num_stakeholders):
        self.stakeholder_keys = stakeholder_keys
        self.required_signatures = required_signatures
        self.num_stakeholders = num_stakeholders
        self.wallet_address = scriptaddr(mk_multisig_script(*stakeholder_keys, required_signatures, num_stakeholders))
```

9: Create an instance of the MultisigWallet class with the stakeholder private keys, the required number of signatures (k = 3), and the total number of stakeholders (n = 4). Print the multi-signature wallet address and the required number of signatures to the console.

```
# Create an instance of the MultisigWallet class
multisig_wallet = MultisigWallet([ceo_private_key, alice_private_key, bob_private_key, coo_private_key], required_signatures, le
```

```
# Print wallet information
print("Multisignature Wallet Address:", multisig_wallet.wallet_address)
print("Required Signatures:", multisig_wallet.required_signatures)

print()
```

```
Multisignature Wallet Address: 3PEKe9zLJUqfUrE5BNfUQAmiEJ1vSXAzZ
Required Signatures: 3
```

10: Retrieve a specific Bitcoin address, taken from Blockchain.com/explorer, and print its transaction history using the history() function from the bitcoin library. This example of a Bitcoin address contains a history of about 126 transactions.

```
valid_address = 'bc1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpquqfle4gjqlszztp'
print(history(valid_address))
```

```
Fetching more transactions... 50
Fetching more transactions... 100

[{'address': 'bc1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpquqfle4gjqlszztp', 'value': 3010, 'output': '01b60141e95cd7a9e371
5e59e5695784d65dba2371af35f60d95ea8bd9b336be:0', 'block_height': 843216, 'spend': 'b8c18698ceb1ca4192f5f6c081dc99b7698ccab048
7640b33fa3a0fbe1e53bb4:0'}, {'address': 'bc1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpquqfle4gjqlszztp', 'value': 3010, 'out
put': 'b638c59494f3b5f0ccb1a2ff734b04b2c7031274205ad1ecff626d9c65225883:0', 'block_height': 843216, 'spend': '331cabddb88ec29
a284bbe93f7c3b4a670d3705651fba6b989900dd52bc49be4:0'}, {'address': 'bc1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpquqfle4gjql
szztp', 'value': 3010, 'output': 'aee36d49cb351adfe49184c421050ca32bb3172be0b8e65b78aded3fcf42ad40:0', 'block_height': 84321
6, 'spend': '62d28f79c583e9612d7ecaa38bac8148f1e1748a3ef2066d02404d02829389e5:0'}, {'address': 'bc1pjp6g8xeyqvzsj4yue585ca0rg
lc25jwehe9l9lgelpquqfle4gjqlszztp', 'value': 3010, 'output': 'ad4b132546f6959195ae57e1c9b3b579884d0373d6709fabddc2625d3847104
d:0', 'block_height': 843216, 'spend': '71c052702f1a23e9b2db10e112b95498e3040fe9df1d743221b5113c90c9c330:0'}, {'address': 'bc
1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpquqfle4gjqlszztp', 'value': 3010, 'output': '4701194b0be5e4c7e9115d6240bd2e28541b
c536533697f7b2683e72aa7a190b:0', 'block_height': 843216, 'spend': 'db980af94281f0dcda4aa328caec391ec0fd0fcf9cc2f05f5d61ce386e
408c00:0'}, {'address': 'bc1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpquqfle4gjqlszztp', 'value': 3010, 'output': '31d911bc0
670eaf86ea74700a4311f3919c14cabde149f39b7b42b186988743e:0', 'block_height': 843147, 'spend': 'cc1e7c6beaa8b6ccaf76e4e2115038f
6afcc4a93c08c61979873d3fa3810d7ba:0'}, {'address': 'bc1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpquqfle4gjqlszztp', 'value':
2856, 'output': 'a4730abd372b609a05577f520366a34186e7781599c9feed0ee651bbc2d640b1:0', 'block_height': 843110, 'spend': '524bc
f80fb9ecd6891861bbd226511f8edd6a4f39d41a0456cd47c805f79de9e:0'}, {'address': 'bc1pjp6g8xeyqvzsj4yue585ca0rglc25jwehe9l9lgelpq
```

11: This code defines the function prove_identity(), simulating stakeholders proving their identity by signing a message with their private keys. While actual private keys would be required for this, a basic string is returned as the "signature" for demonstration. The prove_identity() function is called with "CEO" as the stakeholder name, and the resulting "signature" is printed to the console.

```
# Demonstrating how stakeholders can prove their identities (without actual private keys)
def prove_identity(stakeholder_name):
    message = "I am {}.".format(stakeholder_name)
    # In a real-world scenario, stakeholders would sign the message with their private keys
    # For demonstration purposes, we'll simulate the signature generation
    signature = "Signature for {}".format(stakeholder_name)
    return signature

# Example: CEO proving identity
ceo_signature = prove_identity("CEO")
print("CEO Signature:", ceo_signature)
```

```
CEO Signature: Signature for CEO
```

These Python codes illustrate the sequential creation of a multi-signature Bitcoin wallet, enabling multiple stakeholders to manage the funds. It generates private and public keys along with Bitcoin addresses for each stakeholder. Subsequently, it creates a multi-signature wallet address necessitating a designated number of signatures (here, 3 out of 4) for transaction authorisation. Additionally, the code encompasses a function that simulates stakeholders' proving their identities through message signing with their private keys. Access the Python code file for this Bitcoin wallet creation at:

https://github.com/vershasandesh/Blockchain/blob/main/Blockchain%20-%20Python%20Code%20for%20Bitcoin%20Wallet.ipynb

D. CENTRALISED, DECENTRALISED AND DISTRIBUTED NETWORKS

Centralised, decentralised, and distributed networks represent different approaches to organising and managing information sharing and resources. Each network type offers unique advantages and challenges, and the choice between centralised, decentralised, and distributed systems depends on factors such as the desired level of control, resilience, scalability, and ease of development (Fortify Institute, 2023).

1. **Centralised Network:**

   In a centralised network, a single central network owner controls decision-making and resource allocation. The central network owner serves as the sole point of contact for information sharing, which can lead to efficiency but poses significant risks.

2. **Decentralised Network:**

   In a decentralised network, multiple central owners hold copies of the resources, distributing decision-making and control among them. Decentralisation mitigates the risk of a single point of failure in centralised networks, as information can still be accessed from other nodes if one fails.

3. **Distributed Network:**

   The distributed network takes decentralisation to the extreme, altogether avoiding centralisation and ensuring equal access to resources for all participants. Distributed networks prioritise accessibility and inclusivity, with every participant accessing resources and decision-making.

Figure 1: Centralised, Decentralised and Distributed Network Architectures

Source: (quantalysus, 2018)

Pros and Cons	**Centralised Network**	**Decentralised Network**	**Distributed network**
Points of Failure /Maintenance	-Easy to Maintain -Single point of failure makes them vulnerable to disruptions.	-Have more points of failure but offer greater resilience	-Most challenging to maintain -Offer remarkable resilience against failures
Fault Tolerance /Stability	-Can be highly unable as any failure in central authority can cause chaos.	-Relatively stable than centralised – failures in individual nodes do not disrupt the entire network	-Highly stable
Scalability /Max Population	-Low scalability, limiting capacity to accommodate a large population of users	-Moderate scalability	-Infinite scalability -Suitable for accommodating large population
Ease of Development /Creation	-Quickly developed using established frameworks	-Requires resource-sharing and communication protocols	-Requires addressing complex issues of resource-sharing and communication -Allows for tremendous evolution once the basic infrastructure is in place.

Q2. ETHEREUM

A. THE DIFFERENCE BETWEEN POW AND POS

With blockchains like Ethereum, transaction validation occurs in a decentralised manner. Previously, Ethereum, along with many other cryptocurrencies, relied on the proof of work (PoW) consensus mechanism. Miners employ computer hardware processing power to solve intricate mathematical puzzles and validate new transactions. The miner who successfully solves the puzzle first adds a new transaction to the blockchain ledger and receives cryptocurrency as a reward. However, the PoW process's high energy consumption raises concerns about its environmental impact (bake, 2023).

Proof of stake (PoS) represents a departure from PoW. In PoS, transaction validators stake cryptocurrency as collateral to gain the right to validate transactions. Validators are chosen to

propose a block based on the amount of cryptocurrency they hold and the duration of ownership. Other validators then attest to having seen the proposed block. Once enough attestations are gathered, the block can be added to the blockchain. Validators are rewarded for successfully proposing a block, a process referred to as "forging" or "minting."

The primary advantage of PoS lies in its significantly enhanced energy efficiency compared to PoW. PoS reduces the environmental footprint associated with blockchain validation by decoupling energy-intensive computational processing from the consensus algorithm. Additionally, PoS eliminates the need for extensive computing power to secure the blockchain, making it more accessible and potentially fostering greater decentralisation (Gong, n.d.).

The transition from Ethereum 1.0 (PoW) to Ethereum 2.0 (PoS), also known as Serenity, is termed "The Merge." It aims to improve the Ethereum network's speed, efficiency, and scalability while mitigating its environmental impact. PoS systems like Ethereum 2.0 also address scalability challenges through features like sharding, enabling parallel transaction processing across multiple shard chains, a hurdle PoW systems had faced. Ethereum estimates by switching from PoW to PoS after the Merge, its CO2 emissions fell by 99.992% (Skrill, n.d.).

B. DECENTRALISED FINANCE (DEFI)

PROS OF SWAPPING IN DEFI

1. **Decentralisation:**
   DeFi swapping operates without intermediaries or central authorities, reducing counterparty risks and giving users more control over their assets (wackerow, 2024).
2. **Accessibility:**
   DeFi swapping platforms are globally accessible to anyone with an internet connection and a compatible wallet.
3. **Transparency:**
   Transactions on DeFi swapping platforms are recorded on a public blockchain, providing transparent and auditable trading histories.
4. **Liquidity:**
   DeFi swapping protocols often leverage liquidity pools, where users contribute their assets to facilitate trades, enabling faster and more efficient trading.
5. **Programmability:**
   Built on smart contract platforms, DeFi protocols allow for automated trading processes and complex strategies.

6. **Composability:**

   DeFi protocols can be easily integrated with other DeFi protocols, enabling complex financial transactions.

7. **Permissionless:**

   DeFi platforms do not require approval from any central authority.

CONS OF SWAPPING IN DEFI

1. **Smart Contract Risks:**

   DeFi swapping platforms rely on smart contracts to execute trades. Vulnerabilities or bugs in these smart contracts can lead to security breaches, resulting in the loss of funds for users.

2. **Volatility:**

   DeFi markets can be highly volatile, with asset prices subject to rapid fluctuations.

3. **Complexity:**

   DeFi protocols can be complex and challenging for users unfamiliar with blockchain technology and decentralised applications.

4. **Liquidity Challenges:**

   Some DeFi platforms may face liquidity issues, resulting in slippage or failed transactions.

5. **Regulatory Uncertainty:**

   DeFi operates outside traditional regulatory frameworks, raising concerns about investor protection and compliance with anti-money laundering (AML) and Know Your Customer (KYC) regulations.

6. **Limited User Support:**

   DeFi platforms may lack robust customer support mechanisms compared to traditional finance.

7. **Market Fragmentation:**

   The DeFi ecosystem consists of numerous decentralised exchanges (DEXs) and swapping protocols, leading to fragmentation of liquidity and trading volumes, which can result in price discrepancies and inefficiencies across different platforms.

PERFORMANCE COMPARISON: DEFI VS. TRADFI SWAPPING

Unlike traditional finance (TradFi), decentralised finance (DeFi) swapping presents distinct advantages and challenges across several key dimensions. DeFi swapping offers faster and more

accessible trading experiences than TradFi by eliminating intermediaries and reducing settlement delays. However, DeFi operates within a regulatory grey area, granting users greater autonomy and exposing them to potential regulatory risks.

While traditional finance benefits from deep liquidity pools supported by institutional investors, DeFi platforms rely on user-provided liquidity, which may vary and be susceptible to impermanent loss. Security challenges differ, with traditional financial institutions investing heavily in security measures, whereas DeFi faces risks associated with smart contracts and the absence of centralised oversight.

User experience also varies, with traditional finance platforms often offering user-friendly interfaces, comprehensive customer support, and familiar trading tools. In contrast, DeFi platforms may have steeper learning curves and less intuitive interfaces.

Considering these factors, users can make informed decisions when choosing between DeFi and TradFi swapping platforms based on their specific needs and preferences. Additionally, as the DeFi ecosystem continues to evolve, the performance and user experience of DeFi swapping platforms may improve further.

C. LAYER 2 SOLUTIONS

Layer 2 solutions in the blockchain and cryptocurrency ecosystem refer to a set of technologies and protocols designed to address scalability and throughput limitations of the underlying layer 1 blockchain networks, such as Bitcoin and Ethereum. These solutions aim to improve transaction speed, reduce fees, and increase overall network capacity by processing transactions off the main blockchain while still inheriting the security guarantees of the layer 1 network.

A Layer-2 solution refers to infrastructure built on top of an existing blockchain that can execute transactions off-chain. The core idea behind Layer 2 solutions is to move a significant portion of the computational workload and transaction processing off the main blockchain, which can become congested and slow during periods of high activity. By moving these transactions to a separate layer, the main blockchain is relieved from the burden of processing every single transaction, allowing for higher throughput and lower fees (Sankrit, 2023).

The recent Dencun upgrade, an amalgamation of two separate upgrades — Deneb and Cancun-tackles Ethereum's consensus and execution layers in a single upgrade. This upgrade, which rolled out in March 2024, is set to significantly slash the transaction fees of layer 2 solutions and

boost the scalability of Ethereum. Dencun unveils proto-danksharding, benefiting layer 2 networks such as Polygon, Arbitrum, and Optimism (Gong, n.d.).

Proto-danksharding is a novel technique that removes the limitations of the present on-chain data storage system, opening a vastly more effective data management system. Rollups benefit from scalable data storage via data blobs designed for managing large data volumes outside Ethereum, reducing network congestion, and optimising performance with lower gas prices.

Layer 2 solutions on Ethereum address the blockchain trilemma of decentralisation, scalability, and security. Typically, achieving all three simultaneously is challenging, but solutions like Rollups handle transactions off the mainchain, increasing throughput and reducing costs without compromising network decentralisation or security.

Rollups, grouped transactions submitted to the main Ethereum chain, alleviate network congestion, and expedite transactions. Optimistic Rollups assume transaction validity by default, resorting to computations only during challenges, while ZK-Rollups use zero-knowledge proofs for transaction verification without revealing content. ZK-Rollups offer faster speeds and lower fees but entail a more complex setup prone to centralisation, while Optimistic Rollups are easier to implement but may face withdrawal delays (Dwyer, 2023).

EXAMPLE OF LAYER 2

One prominent example of a layer 2 solution is Polygon (formerly known as Matic Network), a layer 2 scaling solution for the Ethereum network. Polygon combines different scaling techniques, including Plasma and Optimistic Rollups, to increase transaction throughput and reduce fees. Plasma is a technique that creates child chains, which are separate blockchains anchored to the main Ethereum blockchain, allowing for parallel transaction processing. On the other hand, Optimistic Rollups execute transactions off-chain and periodically submit compressed transaction data to the main Ethereum blockchain, reducing the computational load on the main network.

D. NFT COLLECTION

Nonfungible tokens (NFTs) are unique and not interchangeable with one another. An NFT possesses a unique identifier with additional parameters that allow it to store certain information. That unique identifier is what makes a token non-fungible (Hor, et al., 2022).

The ERC721 proposal is a standard for non-fungible tokens, also known as "deeds", which intend to reflect the 'ownership of property", in this case, a digital item (Gong, 2024). The NFTs track the ownership of a unique thing via the unique identifier, which, in the case of the ERC721 standard on the Ethereum blockchain, is achieved by a 256-bit identifier.

Figure 2: The Non-Fungible Token (NFT) Landscape

Non-Fungible Tokens (NFTs)
Non-interchangeable
Unique - Tradeable on marketplaces - Used in games, collectibles, art pieces, intellectual property
Non-divisible - Cannot be divided into parts - Generally illiquid and is difficult to sell instantly

Source: (Hor, et al., 2022)

Therefore, NFTs are blockchain-based, programmable deeds of ownership of an asset. This digital deed gives its holder the exclusive ability to use, sell and transfer the asset's ownership rights, as dictated by their private key signature (Cointelegraph Research, 2021).

The detailed stepwise process of creating NFTs on the Sepolia Network using an ERC721 Contract follows.

NFT CREATION PROCESS

1. **Upload images to IPFS:**

The first step is to upload our images to the InterPlanetary File System (IPFS), which will generate a CID (Content Identifier) for each image file.

For this, a folder is created with the name 'NFTVS_IMAGE' that contains 3 image files, including a coffee jar, a pizza box, and a collection of souvenirs intended to be converted to NFTs. That folder is subsequently uploaded on the 'Pinata' website, a popular IPFS Service provider.

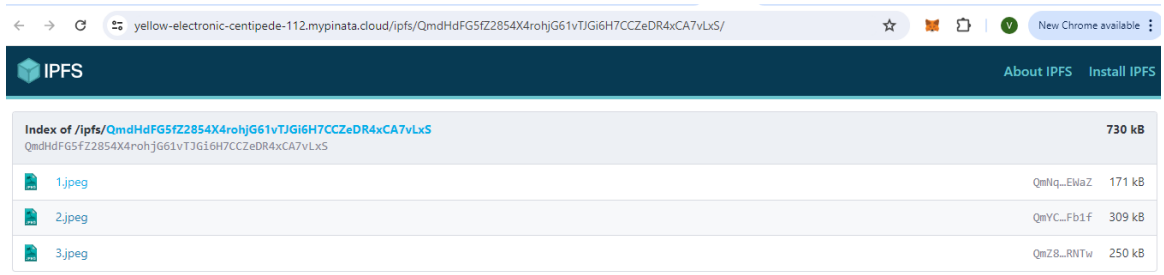
Pinata

Pinata is the first multimedia hub for NFT creators, builders, and artists in the decentralized world of Web 3, which gives users the ability to upload, manage, and share their content whenever, wherever and, with whomever they want (Cointelegraph Research, 2021).

After successful upload, an IPFS-associated CID is created, which is the image URI (Uniform Resource Identifier) pointing to the location of the actual image file associated with the NFT.as shown below:

Figure 3: CID for NFTVS_IMAGE Folder

QmdHdFG5fZ2854X4rohjG61vTJGi6H7CCZeDR4xCA7vLxS



2. Create JSON Metadata files for NFTs:

Create a JSON metadata file for each image that includes information about the image, such as title, description, image URI (IPFS CID), and any other relevant attributes for each NFT.

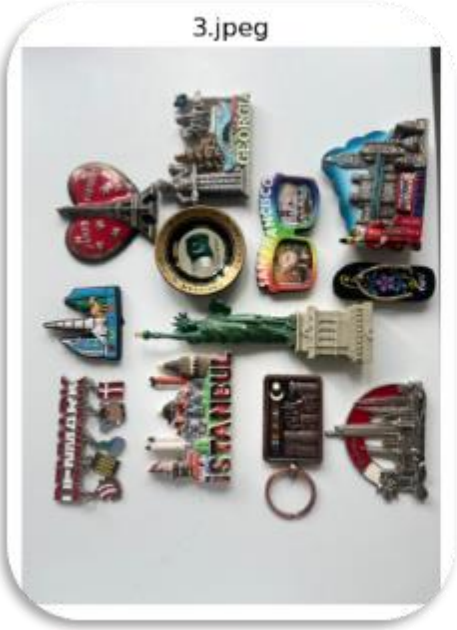
Figure 4: JSON Metadata for Image 1


```
"name": "NFT #1",
"description": "Costa Coffee",
"image": "ipfs://QmdHdFG5fZ2854X4rohjG61vTJGi6H7CCZeDR4xCA7vLxS/1.jpeg",
"attributes": [
    {
        "trait_type": "Keywords",
        "value": [
            "Coffee Break"
        ]
    },
    {
        "trait_type": "Width",
        "value": 1200
    },
    {
        "trait_type": "Height",
        "value": 1600
    },
    {
        "trait_type": "File Size",
        "value": 170853
    },
    {
        "trait_type": "Color Space",
        "value": "RGB"
    },
    {
        "trait_type": "Compression",
        "value": "JPEG"
    },
    {
        "trait_type": "Creation Date",
        "value": "2024-05-10"
    },
    {
        "trait_type": "Author",
        "value": "Versha"
    }
]
```

Figure 5: JSON Metadata for Image 2


```
"name": "NFT #2",
"description": "Pizza",
"image": "ipfs://QmdHdFG5fZ2854X4rohjG61vTJG16H7CCZeDR4xCA7vLx5/2.jpeg",
"attributes": [
    {
        "trait_type": "Keywords",
        "value": [
            "Feeling Hungry"
        ]
    },
    {
        "trait_type": "Width",
        "value": 1200
    },
    {
        "trait_type": "Height",
        "value": 1600
    },
    {
        "trait_type": "File Size",
        "value": 308763
    },
    {
        "trait_type": "Color Space",
        "value": "RGB"
    },
    {
        "trait_type": "Compression",
        "value": "JPEG"
    },
    {
        "trait_type": "Creation Date",
        "value": "2024-05-10"
    },
    {
        "trait_type": "Author",
        "value": "Versha"
    }
]
```


Figure 6: JSON Metadata for Image 3




```
"name": "NFT #3",
"description": "Souvenirs",
"image": "ipfs://QmdHdFG5fZ2854X4rohjG61vTJG16H7CCZeDR4xCA7vLx5/3.jpeg",
"attributes": [
    {
        "trait_type": "Keywords",
        "value": [
            "Let's Travel"
        ]
    },
    {
        "trait_type": "Width",
        "value": 1200
    },
    {
        "trait_type": "Height",
        "value": 1600
    },
    {
        "trait_type": "File Size",
        "value": 250343
    },
    {
        "trait_type": "Color Space",
        "value": "RGB"
    },
    {
        "trait_type": "Compression",
        "value": "JPEG"
    },
    {
        "trait_type": "Creation Date",
        "value": "2024-05-10"
    },
    {
        "trait_type": "Author",
        "value": "Versha"
    }
]
```


For this report, the JSON files for each image have been created using Python and stored in a new folder named 'NFTVS_JSON'. Each JSON file contains an image URI generated through Pinata in Step 1. The Python code for the creation of the JSON folder can be found on the below link:

https://github.com/vershasandesh/Blockchain/blob/main/NFT_VS.ipynb

Ensure that the JSON files conform to the OpenSea metadata standard. OpenSea is a generalised marketplace set up to facilitate the trading of NFTs (opensea, 2024).

3. Upload the JSON files to IPFS:

Next, upload the 'NFTSVS_JSON' folder containing JSON metadata files to IPFS using Pinata. After uploading, a CID for the directory that contains the JSON files is received, which works for setting the 'baseURI' to the IPFS gateway URL. The base URI would usually point to the location where your JSON files containing metadata are hosted.

Figure 7: CID for NFTVS_JSON Folder

QmfBZU1T2yBtD4GDDH4hx5qBF7pGSYHiR4N369DWbrcQqs

Base URI

ipfs://QmfBZU1T2yBtD4GDDH4hx5qBF7pGSYHiR4N369DWbrcQqs/

4. Deploy the NFT smart contract to the Sepolia Network:

The Remix IDE tool is used here to compile and deploy the ERC721 contract to the Sepolia test network. It is essential to have some Sepolia ETH in the Metamask account to pay for gas fees.

For compilation, the ERC721.sol contract from OpenZeppelin is used as a base and extended with the required custom functionality as below:

(i) Pragma: This statement specifies the version of the Solidity compiler to be used.

(ii) Imports: The contract imports two libraries from the OpenZeppelin contracts: 'ERC721Enumerable' and 'Ownable'. These libraries provide functionalities for ERC721 tokens (NFTs) and ownership control.

(iii) Contract Declaration: The NFTVS contract inherits from ERC721Enumerable and Ownable. It means that this contract will have ERC721 token functionality and ownership controls.

(iv) State Variables:

- baseURI: Base URI for token metadata.
- baseExtension: File extension for token metadata.
- cost: Cost in Ether to mint one token, set as 0.01 ether.
- maxSupply: Maximum supply of tokens that can be minted, set as 300.
- maxMintAmount: The maximum number of tokens that can be minted at once is set as 3.
- paused: Boolean variable indicating whether minting is paused.
- whitelisted: Mapping is used to keep track of whitelisted addresses; users on the whitelist only need to pay the gas fee, not the cost.

(v) Constructor: Initializes the contract with a name, symbol, and initial base URI. Additionally, it mints 3 tokens to the contract deployer.

(vi) Internal and Public Functions:

- _baseURI(): Internal function to return the base URI.
- mint(): Public function to mint tokens. Requires a payment in Ether.
- walletOfOwner(): Public function to get the tokens owned by a specific address.
- tokenURI(): Public function to return the token URI for a given token ID.

(vii) OnlyOwner Functions:

- setCost(): Update the cost to mint a token.
- setMaxMintAmount(): Update the maximum number of tokens that can be minted at once.
- setBaseURI(): Update the base URI for token metadata.
- setBaseExtension(): Update the file extension for token metadata.
- pause(): Pause or resume minting.
- whitelistUser(): Add an address to the whitelist.
- removeWhitelistUser(): Remove an address from the whitelist.
- withdraw(): Withdraw the contract balance, distributing 2.5% to another address and the remaining to the owner.

In this way, the solidity code defines a smart contract named NFTVS that represents a non-fungible token (NFT) collection is compiled.

Next, during deployment, the following parameters are passed:

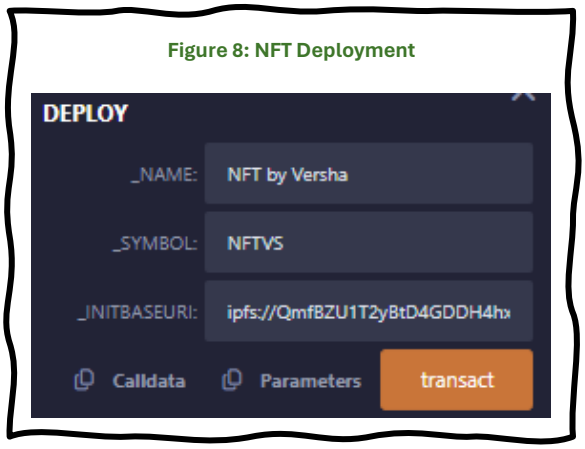
(i) _name: The name of the NFT collection, mentioned as “NFT by Versha”.

(ii) _symbol: The symbol for the NFT collection, set as “NFTVS”.

(iii) _initBaseURI: The IPFS gateway URL followed by the CID of the directory containing the metadata files, as generated in step 3.

"ipfs://QmfBZU1T2yBtD4GDDH4hx5qBF7pGSYHiR4N369DWbrcQqs//"

Figure 8: NFT Deployment



5. NFT Minting:

As part of the minting process, the metadata and associated files are deployed onto the IPFS, which provides us with a secure, transparent, decentralised, and public way to host asset metadata. This means that the NFT's properties will be permanently locked and stored using the decentralised file storage system and cannot be edited or removed. It will be required to pay the gas/transaction fees.

Once the minting transaction is confirmed and recorded on the blockchain, the NFTs are officially minted, and ownership is established. The NFTs are now publicly available and can be transferred to other addresses through subsequent transactions.

6. Verification on OpenSea:

Once the NFTs are minted, verify their presence on the smart contract directly or on an NFT marketplace like OpenSea that pulls the smart contract data. Select the MetaMask icon to be prompted to connect your wallet to OpenSea. Observe that the NFTs were displaying correctly, with metadata (name, description, image, and attributes) fetched from the IPFS-hosted JSON files. If it is intended to sell or trade the NFTs, they can be listed on NFT marketplaces where collectors and enthusiasts can discover and purchase them.

NFT Collection address on OpenSea Marketplace:
https://testnets.opensea.io/collection/nft-by-versha-3
Smart contract Address:
0x8154d55476cF84E1641766a4e4A3c110213852a9

PROPERTIES OF NFT

The minted contract now represents Versha's unique collection of digital assets, where each NFT represents a piece of Versha's favourite things. Among them, you'll find a delightful Costa Coffee jar, a mouthwatering pizza, and a captivating collection of souvenirs from around the world.

These NFTs are more than just digital art – they come with special coupons attached. With Versha's NFT collection, the joy of ownership extends beyond the digital realm, offering tangible rewards and memorable experiences to its lucky owners.

The following are the properties and associated market values for each NFT.

I need a Coffee Break!!!

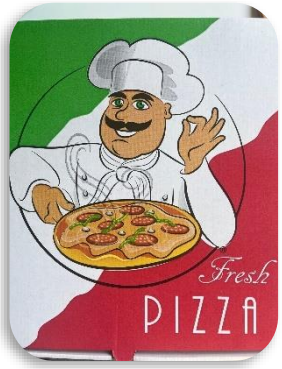
This NFT does not only give you ownership of this digital item, but also benefit you with a weekly treat of free coffee at any Costa's coffee shop around the world.

Collection Name	**NFT by Versha**
Symbol	NFTVS
NFT Name	NFT #1
Description	Costa Coffee
Image URI	ipfs://QmdHdFG5fZ2854X4rohjG61vTJGi6H7CCZeDR4xCA7vLxS/1.jpeg
Attributes	1. Keywords: "Coffee Break" 2. Width: 1200 (The width of the image in Pixels) 3. Height: 1600 (The height of the image in Pixels) 4. File size: 170853 (The size of the image file in bytes) 5. Color Space: RGB (The colour space used in the image) 6. Compression: JPEG (The compression format of the image) 7. Creation Date: 2024-05-10 (The data when the image was captured) 8. Author: Versha (The name of the author or compiler of the image)
Contract Address	0x8154d55476cF84E1641766a4e4A3c110213852a9
Collection Address	https://testnets.opensea.io/collection/nft-by-versha-3
Network	Sepolia test network.
Blockchain	Ethereum blockchain
Token Standard	ERC721 Standard

Token ID	1
Starting Price	0.3 ETH ≈ $873.13 Total
Coupon Attached	A free coffee of choice at any Costa coffee shop once a week.

Feeling Hungry?

This NFT does not only give you ownership of this digital asset but also benefits you with free food delivery service on up to 5 orders each month through Deliveroo, Uber Eats, or Just Eat.



Collection Name	**NFT by Versha**
Symbol	NFTVS
NFT Name	NFT #2
Description	Pizza
Image URI	ipfs://QmdHdFG5fZ2854X4rohjG61vTJGi6H7CCZeDR4xCA7vLxS/2.jpeg
Attributes	1. Keywords: “Feeling Hungry” 2. Width: 1200 (The width of the image in Pixels) 3. Height: 1600 (The height of the image in Pixels) 4. File size: 308763 (The size of the image file in bytes) 5. Color Space: RGB (The colour space used in the image) 6. Compression: JPEG (The compression format of the image) 7. Creation Date: 2024-05-10 (The data when the image was captured) 8. Author: Versha (The name of the author or compiler of the image)
Contract Address	0x8154d55476cF84E1641766a4e4A3c110213852a9
Collection Address	https://testnets.opensea.io/collection/nft-by-versha-3
Network	Sepolia test network.
Blockchain	Ethereum blockchain
Token Standard	ERC721 Standard
Token ID	2
Starting Price	0.51421 ETH ≈ $1,500.00 Total
Coupon Attached	Free delivery coupons for up to 5 orders each month (Deliveroo, Uber Eats, Just Eat)

Let's Travel?

This NFT does not only give you ownership of this digital asset but also benefits you with a generous 30% discount on your hotel stay at destinations featured in the souvenir image, redeemable once a year.

Destinations included: Denmark, Dubai (UAE), Paris (France), Istanbul (Turkey), Pakistan, Georgia, San Francisco, London (UK), Bali (Indonesia), Bahrain, Azerbaijan, New York (USA).

Collection Name	**NFT by Versha**
Symbol	NFTVS
NFT Name	NFT #3
Description	Souvenirs
Image URI	ipfs://QmdHdFG5fZ2854X4rohjG61vTJGi6H7CCZeDR4xCA7vLxS/3.jpeg
Attributes	1. Keywords: "Let's Travel" 2. Width: 1200 (The width of the image in Pixels) 3. Height: 1600 (The height of the image in Pixels) 4. File size: 250343 (The size of the image file in bytes) 5. Color Space: RGB (The colour space used in the image) 6. Compression: JPEG (The compression format of the image) 7. Creation Date: 2024-05-10 (The data when the image was captured) 8. Author: Versha (The name of the author or compiler of the image)
Contract Address	0x8154d55476cF84E1641766a4e4A3c110213852a9
Collection Address	https://testnets.opensea.io/collection/nft-by-versha-3
Network	Sepolia test network.
Blockchain	Ethereum blockchain
Token Standard	ERC721 Standard
Token ID	3
Starting Price	4 ETH ≈ $11,641.00 Total
Coupon Attached	30% off on hotel stays at 12 listed beautiful destinations.

Note: Once the offer price for any NFT is accepted, then, the ownership of the token will be transferred from the creator to another party through a subsequent transaction, and the address of the buyer will be recorded in the whitelist, allowing them to mint tokens without paying the minting fee, and access the premium features and exclusive benefits.

REFERENCES

bake, 2023. *Proof-of-Work vs. Proof-of-Stake: Why did Ethereum Switch to Proof-of-Stake?*. [Online]
Available at: https://blog.bake.io/why-did-ethereum-switch-to-proof-of-stake/
[Accessed 12 May 2024].

Cointelegraph Research, 2021. *Nonfungible Tokens: A New Frontier,* s.l.: s.n.

Dwyer, K., 2023. *What Are Cryptocurrency Layer 2 Scaling Solutions?*. [Online]
Available at: https://coinmarketcap.com/academy/article/what-are-cryptocurrency-layer-2-scaling-solutions
[Accessed 12 May 2024].

Fortify Institute, 2023. *Centralised, Decentralised and Distributed networks - What's the deal?*. [Online]
Available at: https://www.linkedin.com/pulse/centralised-decentralised-distributed-networks-whats/
[Accessed 11 May 2024].

Gong, H., 2024. *Blockchain*. s.l.:University of Westminster - Week 1 Lecture Notes.

Gong, H., 2024. *Metaverse and NFT - Lecture Notes*. s.l.:University of Westminster.

Gong, H., n.d. *Ethereum Basics - Week 4 Lecture Notes*. s.l.:University of Westminster.

Gong, H., n.d. *Layer 2 - Week 11 Lecture Notes*. s.l.:University of Westminster.

Hor, B. et al., 2022. *How to NFT,* s.l.: CoinGecko.

Lamport, L., Shostak, R. & Pease, M., n.d. *The Byzantine Generals Problem,* s.l.: SRI International.

opensea, 2024. *Metadata Standards*. [Online]
Available at: https://docs.opensea.io/docs/metadata-standards
[Accessed 11 May 2024].

quantalysus, 2018. *Choosing between Centralized, Decentralized, and Distributed Networks*. [Online]
Available at: https://steemit.com/cryptocurrency/@quantalysus/choosing-between-centralized-decentralized-and-distributed-networks
[Accessed 11 May 2024].

Sankrit, K., 2023. *What are Layer-2 Solutions? A guide to L-2 Blockchains*. [Online]
Available at: https://www.moonpay.com/en-gb/learn/blockchain/what-are-layer-2-solutions
[Accessed 12 May 2024].

Skrill, n.d. *The difference between Proof-of-Work and Proof-of-Stake*. [Online]
Available at: https://www.skrill.com/en/crypto/the-skrill-crypto-academy/advanced/the-difference-between-proof-of-work-and-proof-of-stake/#:~:text=PoW%20uses%20a%20combination%20of,CO2%20emissions%20fell%20by%2099.992%25.
[Accessed 12 May 2024].

wackerow, 2024. *PROOF-OF-STAKE VS PROOF-OF-WORK*. [Online]
Available at: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/pos-vs-pow/
[Accessed 12 May 2024].