BlockChain Project report

Abstract

Introduction

what is blockchain and NFTs?

Blockchain

Blockchain is a shared, immutable, and decentralized chain of nodes designed for recording transactions between users in a secure and transparent manner. Each block in the chain is linked to the previous one in a sequential order, forming an unbroken chain. These blocks are recorded immutably across a peer-to-peer network, ensuring consistency and trust among participants without the need for a central authority.

The process of **mining** serves as the primary mechanism for validating new blocks within the Blockchain. During mining, participants compete to solve complex mathematical problems, which, once resolved, confirm the validity of the transactions contained in a block. Every participant in the chain maintains an encrypted, synchronized record of all transactions waiting for validation, enhancing the transparency and security of the system.

Once a block is validated, it is permanently added to the chain, making it **virtually impossible to alter or delete** its contents. This immutability is a cornerstone of Blockchain technology, ensuring the integrity and reliability of the ledger. The decentralized nature of the network further strengthens its resistance to tampering, as any attempt to modify a transaction would require consensus across the majority of nodes, an infeasible task in a well-distributed Blockchain.

Non-Fungible Tokens (NFTs)

Non-Fungible Tokens (NFTs) are unique, immutable, and indivisible digital assets recorded on a blockchain. Unlike cryptocurrencies such as Bitcoin or Ethereum, which are fungible and can be exchanged on a one-to-one basis, NFTs represent ownership of distinct items or pieces of content. Each NFT has a unique identifier and metadata that distinguish it from any other token, ensuring its uniqueness and scarcity.

NFTs are typically associated with digital art, music, videos, collectibles, and virtual real estate, but their applications extend far beyond. The blockchain serves as a transparent ledger for recording the ownership and provenance of each NFT, making it possible to trace its history with complete confidence.

The process of **minting** is used to create NFTs, where digital files are converted into blockchain-based tokens. Once minted, an NFT is permanently recorded on the blockchain, making it impossible to duplicate or alter its record. Ownership of an NFT is secured through cryptographic keys, and the rights to transfer or sell it lie exclusively with the token's holder.

By leveraging blockchain technology, NFTs enable creators to monetize their work directly while ensuring authenticity and proof of ownership. Additionally, smart contracts embedded in NFTs can automate royalties, allowing artists to receive a share of proceeds each time their work is resold.

NFTs represent a significant shift in how digital ownership is perceived, offering a secure, transparent, and decentralized way to trade unique digital assets while fostering new opportunities for creators and collectors in the digital economy.

Our problem

Under the 2030 Moroccan digital strategy, administrative services are being modernized to enhance efficiency and reduce reliance on traditional paper-based operations. One such service is the verification of scholar certifications, particularly diplomas, which are currently validated manually by companies during the hiring process for fresh graduates. This method is time-consuming and prone to inefficiencies. By adopting Blockchain and NFT technologies, these certifications can be assigned a unique digital identity, enabling instant and secure verification through a decentralized network.

Blockchain provides a tamper-proof and transparent infrastructure for managing digital certifications. Each diploma can be minted as a Non-Fungible Token (NFT), a unique and immutable record stored on the blockchain. This ensures that certifications are easily accessible and verifiable, eliminating the need for intermediaries.

Using Blockchain offers several key benefits. Its decentralized nature ensures that no single entity controls the verification process, while its immutability prevents alterations or fraud. Additionally, the efficiency of blockchain-based systems reduces processing time, allowing companies to validate certifications quickly and accurately. Graduates retain ownership of their digital certifications and can share them securely, providing greater control over their credentials.

By leveraging Blockchain and NFTs, Morocco’s digital strategy not only enhances the reliability and transparency of scholar certification processes but also sets a foundation for broader applications of these technologies in administrative services.

Technical analysis of the problem

app architecture

(ndiro shema hna)

- Blockchain structure (institut - entreprise - student)
  we suggest a blockchain structure with three peers: Institut for validation of the certif, entreprise and student as verifiers.
- diploma validation process: the verification is done by checking if the student has all of his semesters as valid and approved by the university director.

Solutions and why?

Implementation

Backend (smart contracts)

certification

the certification smart contract represent the NFT version of the diploma issued by the student.
the contract specify several information about the diploma holder such as:
- `uid`: Unique identifier for the recipient.
- `candidate_name`: Name of the certificate holder.
- `speciality`: Name of the specialty of candidate.
- `inst_name`: Issuing institute.
- `ipfs_hash`: IPFS link to the certificate's metadata or details.

- (explain IPFS)
- The contract inherits from the `ERC721` standard, making certificates unique, transferable, and interoperable with NFT-compatible wallets and platforms. (explain ERC721)
- The `mintCertificate` function mints a new NFT for the caller (`msg.sender`).
- It stores the certificate details in a mapping, associating them with the token ID.
- Emits a `CertificateMinted` event upon successful creation.
- The `getCertificate` function allows anyone to retrieve the details of a certificate using its token ID.
- Ensures that only existing certificates can be queried by validating the token ID.

migrations

The `Migrations` contract, commonly used in Truffle-based Ethereum projects, helps manage and track the deployment process of smart contracts during migrations. It stores the deployer's address as the `owner` and a `last_completed_migration` variable to record the most recent migration step. The `restricted` modifier ensures that only the owner can execute certain functions. The `setCompleted` function updates the migration progress by setting the latest completed step, while the `upgrade` function allows transitioning to a new `Migrations` contract by transferring the current migration state. This contract ensures that migrations are executed in order, avoids re-execution of completed steps, and maintains continuity in the migration process, all while restricting access to authorized users. It is a utility contract designed to streamline deployment workflows rather than directly implementing application logic.

frontend

the User Interface part of our project is no less important than the backend and Smart Contracts side. We designed the Frontend to be simple to use for both the institution and the verifier, with functionalities such as User authentication, certificate issuance and verification.
Our app interface will be divided into two portals: one for the institute and other for the verifier. the institute portal will be exploited by the institution to upload the certificate to the blockchain as a NFT once the student verifies the conditions for obtaining the Diploma. While the verifier portal act as a gate for whom want to verify the validity of the diploma by uploading the given version of the certificate or the certificate ID.
The Frontend constitute by several pages to mention:
(picture for each point)

- `institute.py` : represent the UI for the institution portal.
- `verifier.py` : represent the UI for the verification side.
- `login.py` : for login (for institution)

as well as `app.py` which represent the main page of our page. The page allows users to choose between two roles—**Institute** or **Verifier**—before proceeding to the login screen. This layout ensures clear role-based access within the application.
(picture of app)
NB: The full code is attached in appendix.

IPFS

jo

Conclusion

bibliography