Tokenization

Tokenization

- Tokenization is a Process of representing real-world assets as digital tokens on a blockchain
- Types of Assets: Real estate, stocks, commodities, art, intellectual property, etc.
- Tokens can be stored, moved, and recorded on blockchain.

Process of Tokenization

- Asset Identification: Selection of asset to be tokenized
- Legal Compliance: Compliance with regulations governing asset tokenization
- Token Design: Determining token characteristics (e.g., supply, divisibility)
- Smart Contract Development: Creating smart contracts to govern token issuance, transfer, and management

A tokenization system needs to prove that transactions are executed correctly to prevent double spending. This means it ensures that when someone spends a token once, they can't spend it again elsewhere. It's like making sure a coin can't be used to buy two things at once. This proof keeps transactions safe from being used more than once.

Security Requirement

Tokenization system should provide a proof of correct execution of transactions i.e, protection from double spending attacks.

Security Model

Trust in the token issuer: This means trusting that the issuer has enough collateral to back up each token they create, ensuring a 1:1 ratio. It's like making sure there's enough money in the bank to cover every dollar bill they issue.

Each asset management requires two types of trust.

- Trust to a token issuer (proof of 1:1 ratio for collateral)
- Trust to a transaction processing system (protection from double spending attacks).

An issuer is an entity or organization responsible for creating and distributing tokens within a tokenization system. They are the ones who generate the tokens and typically back them up with some form of collateral or value. In the context of cryptocurrencies or digital assets, the issuer may be a company, a financial institution, or a decentralized organization responsible for creating and managing the tokens on a blockchain network.

Advantages of Tokenization in Blockchain

- Fractional Ownership: Enables fractional ownership of high-value assets
- Accessibility: Allows global access to investment opportunities
- Liquidity: Facilitates trading and liquidity for traditionally illiquid assets
- Transparency: Transparent ownership records on the blockchain
- Security: Immutable and tamper-proof transactions
- Lower management cost
- Eliminate middle man
- Quick and cheaper transactions

Liquidity refers to the ease with which an asset can be bought or sold in the market without significantly affecting its price. When we say that tokenization facilitates trading and liquidity for traditionally illiquid assets, it means that by representing these assets as tokens on a digital platform, they become easier to buy and sell. This is because digital tokens can be traded on online exchanges, allowing a wider pool of investors to participate in the market. As a result, assets that were previously difficult to sell or buy, such as real estate or art, become more liquid and accessible to a broader range of investors.

Use Cases of Tokenization

- Real Estate: Fractional ownership of properties
- Art and Collectibles: Tokenizing ownership of artworks and collectibles as NFTs
- Securities: Issuance of digital securities representing ownership in companies
- Commodities: Tokenization of commodities such as gold, oil, etc.
- Intellectual Property: Representing ownership of patents, copyrights, etc.

Ex:Tokenization of real world assets

Assets	Tokenized by
gold and other commodities	DigiX, Platform for tokenizing physical assets such as gold and other commodities
gold	DigiX Gold (DGX),Tokenized gold backed by physical gold bullion
Real Estate	RealIT
Bonds	Society Generate
Equity	Spice VC
Paintings	Andy warhol

Tokenization Process

Acquisition of Physical Gold: Procurement and storage of gold bullion in secure vaults

Asset Verification: Independent audits and verification of gold reserves

Token Issuance: Creation of DGX tokens equivalent to the stored gold

Redemption: Ability to redeem DGX tokens for physical gold

Get Gold: First, we buy and store physical gold in safe vaults.
Check Gold: We have independent experts check regularly to make sure the amount of gold we say we have is correct.
Make Tokens: We create digital tokens called DGX. Each token represents a certain amount of the gold we have.
Trade Tokens: People can buy and sell these DGX tokens like they do with any digital currency.
Get Gold Back: If someone wants, they can exchange their DGX tokens for real gold whenever they like.

Advantages of DigiX

Stability: Backed by physical gold, providing stability and hedge against market volatility

Accessibility: Enables fractional ownership of gold, making it accessible to a wider audience

Liquidity: Facilitates trading and liquidity through digital exchanges and platforms

Transparency: Transparent audit trail of gold reserves on the blockchain

Use Cases of DigiX

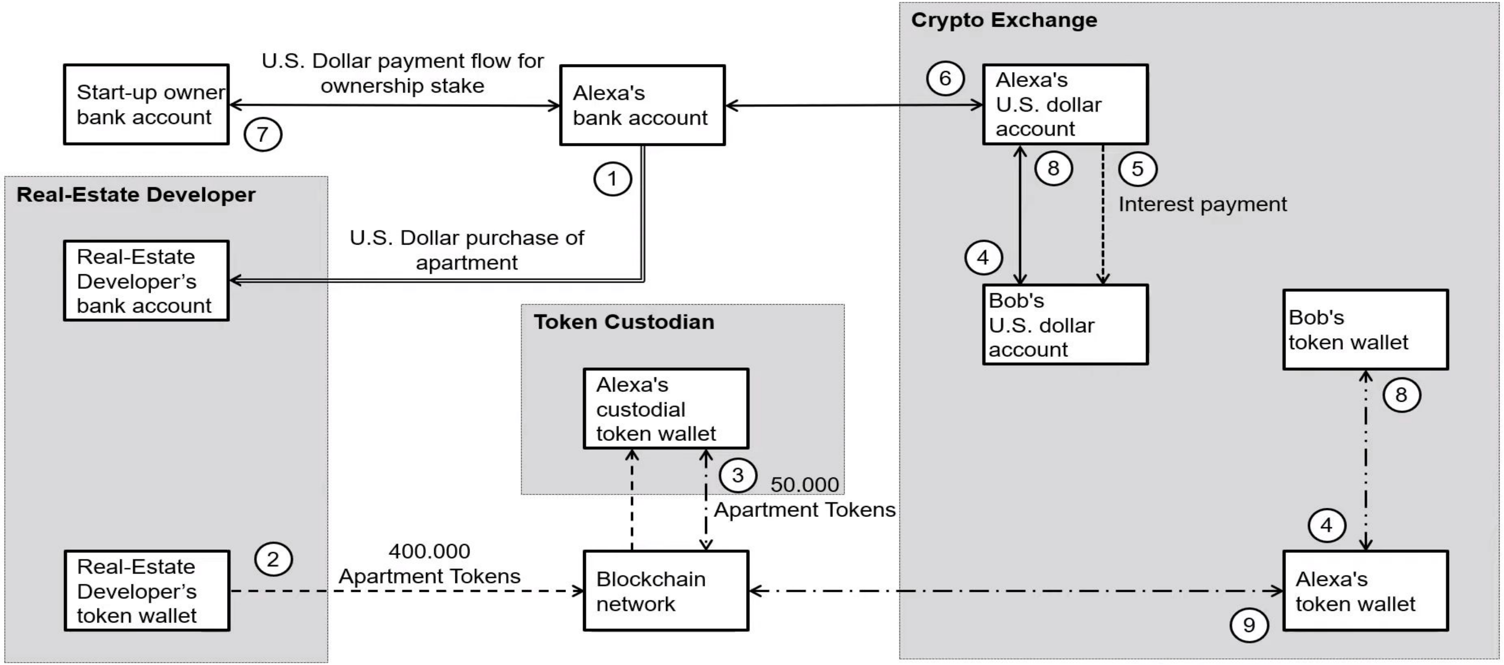
Investment: Diversification of investment portfolios with gold-backed tokens

Payments: Use of DGX tokens for transactions and remittances

Wealth Preservation: Hedging against inflation and economic uncertainties

Collateral: Utilization of DGX as collateral for loans and financial products

Case Studies:Real Estate: Fractional ownership of properties


Crypto Exchange
U.S. Dollar payment flow for ownership stake
Start-up owner bank account
7
Alexa's bank account
6
Alexa's U.S. dollar account
1
8
5
Interest payment
Real-Estate Developer
U.S. Dollar purchase of apartment
Real-Estate Developer's bank account
4
Bob's U.S. dollar account
Bob's token wallet
Token Custodian
Alexa's custodial token wallet
8
3
50.000 Apartment Tokens
2
400.000 Apartment Tokens
4
Real-Estate Developer's token wallet
Blockchain network
Alexa's token wallet
9

Fungible Tokens:

- Refers to a type of digital asset that is interchangeable with another asset of the same type
- Fungible tokens are often used to represent assets such as currency, commodities, or other financial instruments on a blockchain (Bitcoin (BTC),Ethereum (ETH),USD Coin (USDC), Ethereum's ERC-20
- key characteristics and examples of fungible tokens:
  - Interchangeability
  - Indivisibility
  - Standardization

Fungible tokens are like identical coins or bills: one is as good as another. They're digital assets that can be swapped one-for-one with another of the same type. Think of them as representing things like money or shares in a company on a blockchain. Some key points:
Interchangeability: Each token is the same as any other of its kind, making them easily exchangeable.
Indivisibility: They can't be divided into smaller parts; you can only trade them as whole units.
Standardization: Fungible tokens follow set standards, making them compatible across various platforms and wallets.
Examples include cryptocurrencies like Bitcoin (BTC), Ethereum (ETH), stablecoins like USD Coin (USDC), and tokens following Ethereum's ERC-20 standard.

- Use Cases:

Fungible tokens have various use cases, including **digital currencies, stablecoins, loyalty points, and tokenized assets such as gold or real estate**. They can be traded on decentralized exchanges, used as a medium of exchange for goods and services, or utilized within decentralized finance (DeFi) applications for lending, borrowing, and liquidity provision.

ERC-20 is the **technical standard** for fungible tokens created using the Ethereum blockchain.

ERC-20 allows developers **to create smart-contract-enabled tokens** that can be used with other products and services. These tokens are a representation of an asset, right, ownership, access, cryptocurrency, or anything else that is not unique in and of itself but can be transferred.

ERC-20 Contents

ERC-20 is a list of functions and events that must be implemented into a token for it to be considered ERC-20 compliant. These functions (called methods in the ERC) describe what must be included in the smart-contract-enabled token, while events describe an action. The functions a token must have are:

- **TotalSupply**: The total number of tokens that will ever be issued
- **BalanceOf**: The account balance of a token owner's account
- **Transfer**: Automatically executes transfers of a specified number of tokens to a specified address for transactions using the token
- **TransferFrom**: Automatically executes transfers of a specified number of tokens from a specified address using the token
- **Approve**: Allows a spender to withdraw a set number of tokens from a specified account, up to a specific amount
- **Allowance**: Returns a set number of tokens from a spender to the owner

The events that must be included in the token are:

- **Transfer**: An event triggered when a transfer is successful
- **Approval**: A log of an approved event (an event)

The following functions are optional and are not required to be included, but they enhance the token's usability:

- Token's name (optional)
- Its symbol (optional)
- Decimal points to use (optional)

TotalSupply: Indicates the total number of tokens that will ever exist in circulation. It provides transparency regarding the token's maximum supply.
BalanceOf: Shows the number of tokens held by a specific account, providing visibility into individual token ownership.
Transfer: Automatically moves a specified number of tokens from one account to another, enablin transactions using the token.

TransferFrom: Allows automated transfers of tokens from one account to another, but with the approval of the token owner. This function is commonly used in scenarios where tokens are held i

custody by a smart contract.
Approve: Grants permission to another account (spender) to withdraw a specified number of toker from the owner's account, up to a certain limit. It facilitates token spending by approved third partie
Allowance: Returns the amount of tokens approved by the owner for withdrawal by a specific spender. This function provides transparency regarding the spending limits granted by the token owner.

Non Fungible Tokens:

- Non-fungible tokens (NFTs) are a type of digital asset that represent ownership or proof of authenticity of a unique item or piece of content using blockchain technology
- Created, bought, sold, and traded on various blockchain platforms, etherium popular for NFT creation.
- unique characteristics and ownership history, providing transparency and authenticity to the digital asset.

Use Cases of NFTs

- Digital Art: Examples of NFTs representing digital artworks and their high-profile sales
- Collectibles: NFTs for virtual collectibles, trading cards, and rare items
- Gaming: Integration of NFTs in gaming for unique in-game assets and items
- Music and Media: NFTs for music albums, video clips, and other digital media.
- ERC721 is a standard for representing ownership of non-fungible tokens, that is, where each token is unique.
-

- ERC721 provides a standardized way to represent ownership of unique digital assets on the Ethereum blockchain. It has inspired the development of various applications and platforms for buying, selling, and trading NFTs.
- ERC-721 tokens are implemented using smart contracts on the Ethereum blockchain. These smart contracts define the rules for creating, transferring, and managing the tokens.
- ERC-721 tokens can store metadata that provides additional information about the asset they represent. This metadata can include details such as the creator of the asset, its characteristics, and its history.

Standard Functions:

balanceOf(address _owner) returns (uint256):

- Returns the number of tokens owned by a specific address `_owner`.

ownerOf(uint256 _tokenId) returns (address):

- Returns the address that owns the token with ID `_tokenId`.

approve(address _to, uint256 _tokenId):

- Allows an address `_to` to transfer the ownership of a token `_tokenId` owned by the caller.

transfer(address _to, uint256 _tokenId):

- Transfers the ownership of a token `_tokenId` to another address `_to`. This function is used when the caller owns the token.

transferFrom(address _from, address _to, uint256 _tokenId):

- Transfers the ownership of a token `_tokenId` from address `_from` to address `_to`. This function is used when the caller has been approved to transfer the token on behalf of `_from`.

safeTransferFrom(address _from, address _to, uint256 _tokenId):

- Similar to `transferFrom`, but additionally checks if `_to` is a smart contract that implements the `onERC721Received` interface to prevent accidental loss of tokens.

Standard Events:

Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId):

- Triggered when ownership of a token is transferred from address `_from` to address `_to`.

Approval(address indexed _owner, address indexed _approved, uint256 indexed _tokenId):

- Triggered when an address `_owner` approves another address `_approved` to transfer a token `_tokenId` on their behalf.

ApprovalForAll(address indexed _owner, address indexed _operator, bool _approved):

- Triggered when an address `_owner` grants or revokes approval for another address `_operator` to manage all of their tokens.

These functions and events provide the basic infrastructure for managing ERC-721 tokens within a smart contract.
Developers can extend these functionalities to create more complex interactions and implement additional features as needed for their specific use cases.

Challenges and Considerations

- Technological hurdles: need to scale to handle large volume of data.
- Ecosystem is still in early stage of development.
- Demand for experienced and custodian partners is not met.
- Legal and Regulatory Compliance: Compliance with securities laws, KYC/AML regulations, that need some form of centralization.
- Asset Valuation: Determining the value of tokenized assets
- Interoperability: Ensuring compatibility between different blockchain platforms and token standards
- Security Concerns: Risks of hacking, smart contract vulnerabilities

Future Trends

- Increased Adoption: Growing acceptance of tokenization as a means of asset representation
- Interoperability Solutions: Development of interoperability protocols for seamless asset transfer across different blockchains
- Regulatory Clarity: Continued efforts towards regulatory clarity and framework development
- Integration with DeFi: Integration of tokenized assets with decentralized finance (DeFi) protocols

Types of Tokens

- Tokens are classified according to the function and purpose
- Tokens serve various functions within blockchain ecosystems
- Different types of tokens catering to different use cases.

Types of tokens

utility tokens give access to the specialized products or services the company is working on

1. **Transactional**: Tokens designed to be used as a payment method. Bitcoin is the most well-known of these.
2. **Utility:** XRP and ETH are two examples of utility tokens. They serve specific functions on their respective blockchains.
3. **Security tokens:** Tokens representing ownership of an asset, such as a stock that has been tokenized (value transferred to the blockchain). MS Token is an example of a securitized token. If you can find one of these for sale, you can gain partial ownership of the Millenium Sapphire.

A security token is a peripheral device used to gain access to an electronically restricted resource. The token is used in addition to, or in place of, a password.

4. **Governance:** These tokens represent voting or other rights on a blockchain, such as Uniswap.
5. **Non-Fungible Tokens (NFTs)**
6. **Stablecoins**
7. **Equity Token:**
8. **Wrapped Tokens**
9. **Tokenization Platforms:** These tokens support applications built to use a blockchain, such as Solana.

Transactional Tokens

Transactional tokens are used to transact—they serve as units of account and are exchanged for goods and services. These tokens often function like traditional currencies, but in some cases, provide additional benefits.

Transactional tokens are like digital coins or tokens that people can use to buy things or trade with others. They work similarly to regular money, but they often have some special features.

For example, think about Bitcoin or Dai. These are types of transactional tokens. With them, you can buy things online without needing a bank or a payment service like PayPal. They're like money that you can use on the internet.

But there are other transactional tokens that aren't exactly like money. Some businesses use them to keep track of things they buy and sell. They use a special technology called blockchain to make sure nobody can change the records. It's like having a super secure digital notebook where everything is written down, and once it's written, it can't be changed. This helps businesses keep track of their stuff and make sure everything is fair and honest.

For example, with decentralized cryptocurrencies, such as Bitcoin and Dai, it is possible for users to execute transactions without a traditional intermediary or central authority, such as a bank or payment gateway. In addition to its function as a currency, Dai offers transactional performance to other networks. For example, POA Network created xDai, a Dai-like transactional token that lives on a sidechain, allowing for fast, inexpensive transactions.

Not all transactional tokens are currencies. Global supply chains and other industries utilize transactional tokens to apply the immutable nature of the blockchain and the flexibility of smart contracts to their operations.

Utility Tokens

https://www.gemini.com/cryptopedia/what-is-basic-attention-token-and-how-does-it-work Basic attention token on brave browser

https://trustmachines.co/glossary/utility-token/

https://www.ledger.com/academy/glossary/utility-token

https://www.coinsmart.com/articles/what-are-utility-tokens/

https://www.blockchain-council.org/blockchain/security-tokens-vs-utility-tokens-a-concise-guide/

Governance token

Tokenized equity

https://eqvista.com/cap-table/tokenized-equity/#:~:text=Example%20of%20tokenized%20equity&text=Quadrant%20Token%20was%20released%2C%20and,tokens%20at%20%241.25%20per%20token.