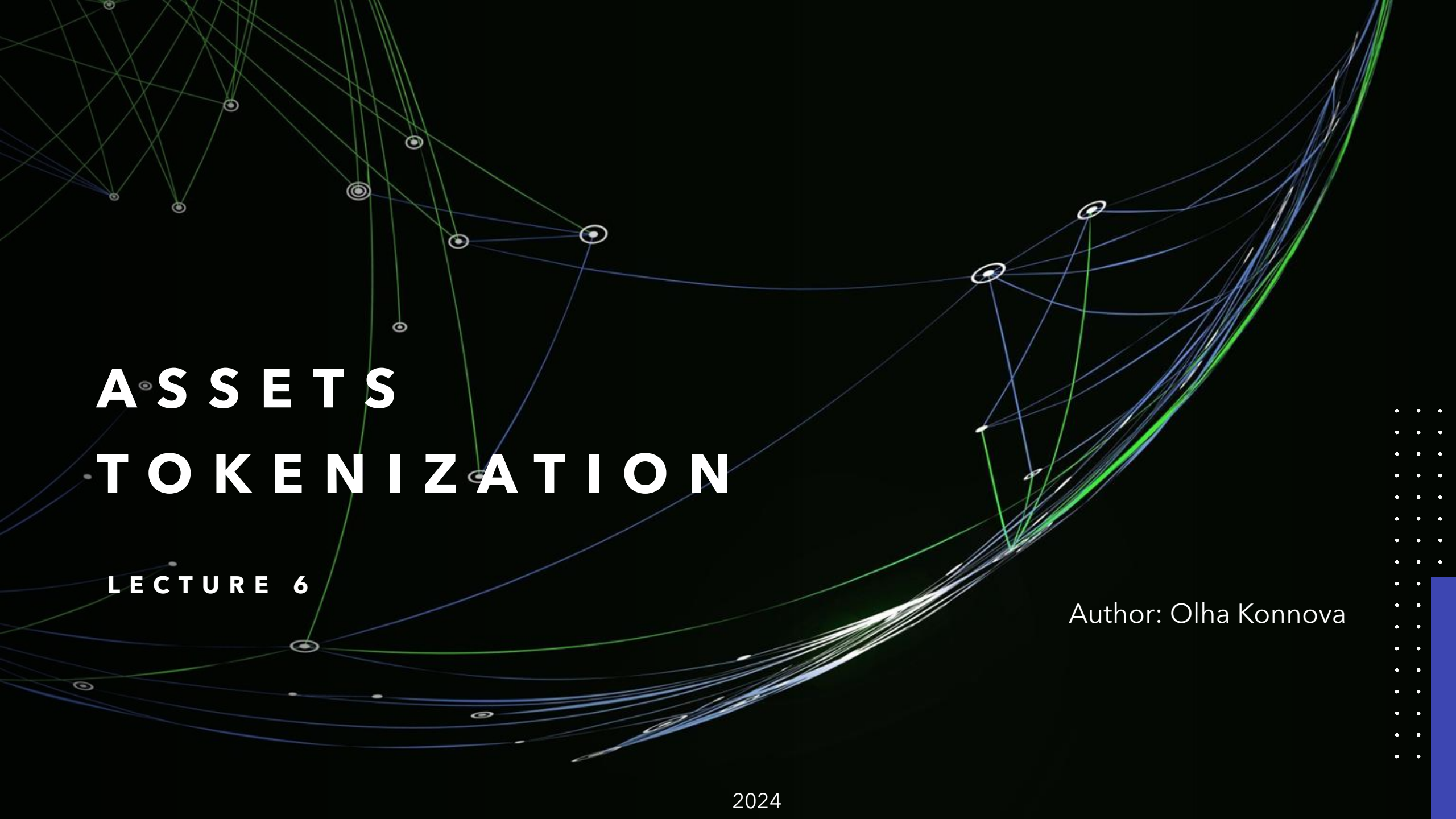


ASSETS TOKENIZATION

LECTURE 6

Author: Olha Konnova

2024

PLAN

1) Concept of tokenization
2) Token. Classification of tokens
3) Fungible (FT) and Non-fungible (NFT) tokens
4) Algorand Standard Asset
5) Creating your own ASA using SDK and Dappflow
6) Pera Wallet. Creating an account

TOKENIZATION

Tokenization is the process of transforming rights to an asset into digital tokens that exist on the blockchain. It involves creating a digital representation of physical or digital assets on the blockchain, where each token is a unique digital identifier of the asset.

Virtually any asset – whether physical, like real estate, or intangible, like company shares – can theoretically be tokenized.

THE CONCEPT OF "TOKEN"

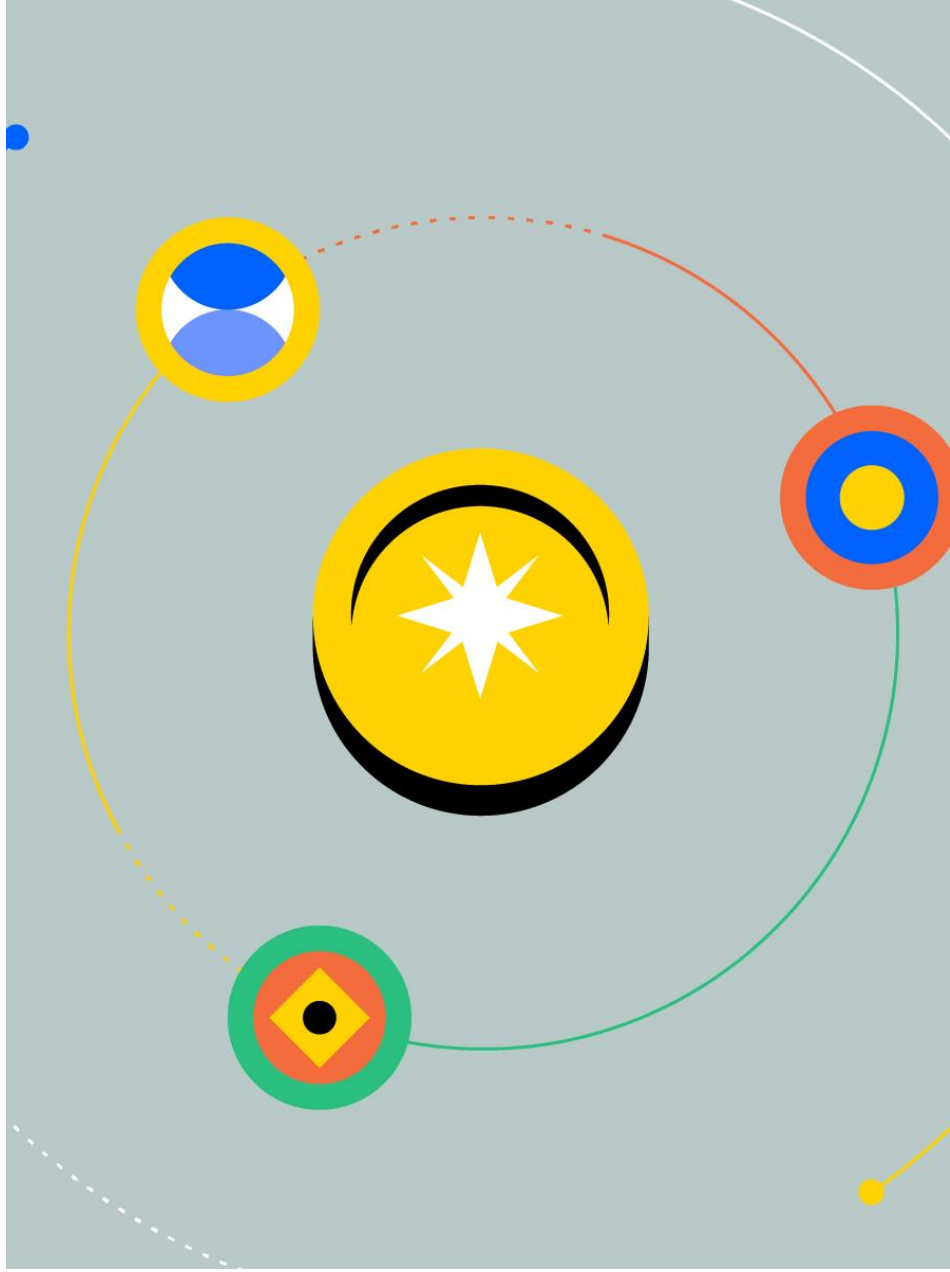
The term "token" originates from the Old English word "tacen," meaning sign or symbol. It is often used to describe special-purpose tokens issued privately and lacking intrinsic value, such as arcade or transit tokens.

TOKEN

A **token** is a digital representation of ownership or access rights to an asset on the blockchain.

Tokens can represent various types of assets, including:

- **Digital assets:** cryptocurrencies, intellectual property, gaming items
- **Physical assets:** real estate, cars, artwork
- **Financial instruments:** stocks, bonds
- **Services:** access to cloud storage, streaming services



KEY MILESTONES IN TOKEN DEVELOPMENT

- **Bitcoin Emergence (2009):** the first practical use of blockchain as a digital currency. The term "token" was not yet in use. Bitcoin was used as a digital currency or "coin".
- **Ethereum and Smart Contracts (2015):** Ethereum enabled developers to create their own tokens using the ERC-20 standard, which defined a set of rules for issuing tokens.
- **ICO Boom (2017–2018):** the period of intensive use of Initial Coin Offerings (ICO), when startups raised funds by issuing and selling tokens. This led to the popularization of the concept of "token" and its use in a wide range of projects.
- **Rise of DeFi and NFTs (2020–today)**: demonstrated how tokens provide access to financial services and unique digital assets.

TOKEN CLASSIFICATION: UTILITY TOKENS

- **Utility Tokens:** provide holders with access to a specific product or service in the blockchain ecosystem. They are typically used for internal transactions within the platform, such as payment for services or access to special features.

Example:

- **Ether (ETH)** on the Ethereum platform, which is used to pay gas for the execution of smart contracts and to incentivize validators to ensure network security.
- **BNB (Binance Coin)** on the Binance platform, which is used to pay commissions on the exchange.

TOKEN CLASSIFICATION: SECURITY TOKENS & ASSET-BACKED TOKENS

- Security Tokens: a digital representation of traditional securities such as stocks, bonds, or real estate, and are subject to the same financial rules as their physical counterparts. These tokens derive their value from the underlying asset and can offer a variety of benefits, including dividends, profit-sharing rights, or voting rights.

  Example:

  tZERO (TZROP) representing equity in tZERO Group.

- Asset-Backed Tokens are backed by real-world assets like real estate or gold, offering fractional ownership. These tokens allow investors to gain partial ownership of assets without the need to physically store them.

TOKEN CLASSIFICATION: PAYMENT & GOVERNANCE TOKENS

- Payment Tokens: Also called cryptocurrencies or digital currencies, they are intended to be used as a medium of exchange, similar to traditional currencies. They are used to pay for goods and services and are often created to enable fast and secure transactions.

  Example: Bitcoin (BTC): the most well-known cryptocurrency used as a medium of exchange.

- Governance Tokens: give holders the right to participate in the management of a blockchain platform or project. Holders of such tokens can vote on protocol changes, new features, or the use of funds.

  Example: Uniswap (UNI): a token that gives holders a voice in the management of the decentralized Uniswap exchange.

TOKEN CLASSIFICATION: STABLECOINS

Another application of tokens is **stablecoins**, which solve the problem of cryptocurrency volatility.

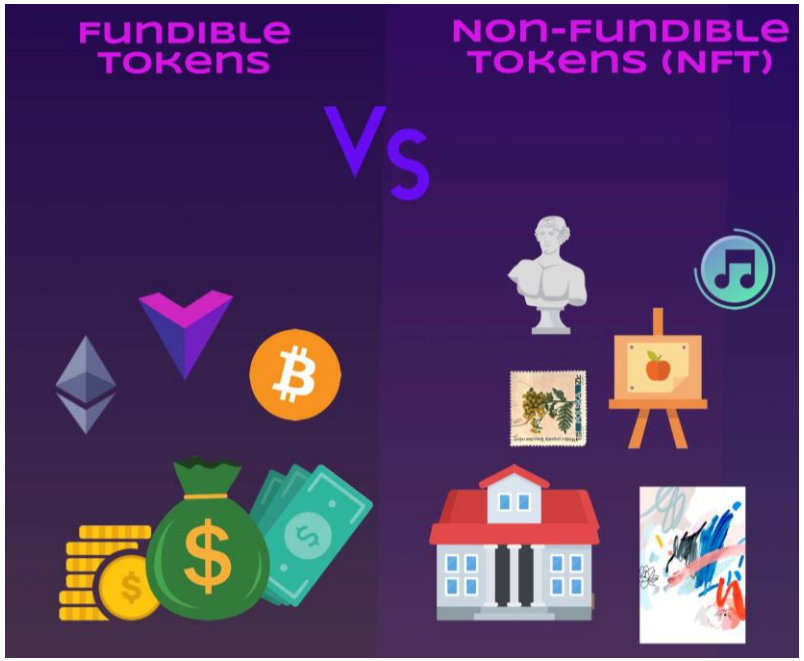
> "Stablecoins are cryptocurrencies whose value is pegged, or tied, to that of another currency, commodity, or financial instrument"
>
> - Adam Hayes, Investopedia

Stablecoins can be divided into several categories depending on what they are backed by. These are: **fiat-backed, crypto-backed, commodity-backed, and algorithmic stablecoins**.

Examples: Tether (USDT), USD Coin (USDC), Dai (DAI)

Source: https://www.investopedia.com/terms/s/stablecoin.asp

NFT VS FT



- **NFTs (Non Fungible Tokens)** are unique cryptographic assets that have their **own identifier** and cannot be reproduced. NFTs may look similar, but they can be distinguished by their unique identifier and are not interchangeable. NFTs can represent digital and real assets such as photos, art, gaming items, tickets, etc.
- **Fungible tokens** are similar to NFTs, but the big difference is that these assets are fungible and have the **same identifier**. Users cannot distinguish between these assets that come from the same set because the image, metadata, and identifier are the same. FTs can represent interchangeable assets such as in-game potions, currency, points, etc.

Source: https://support.nft.io/hc/en-gb/articles/14111783884305-What-is-a-Non-Fungible-Token-NFT-and-Fungible-Token-FT

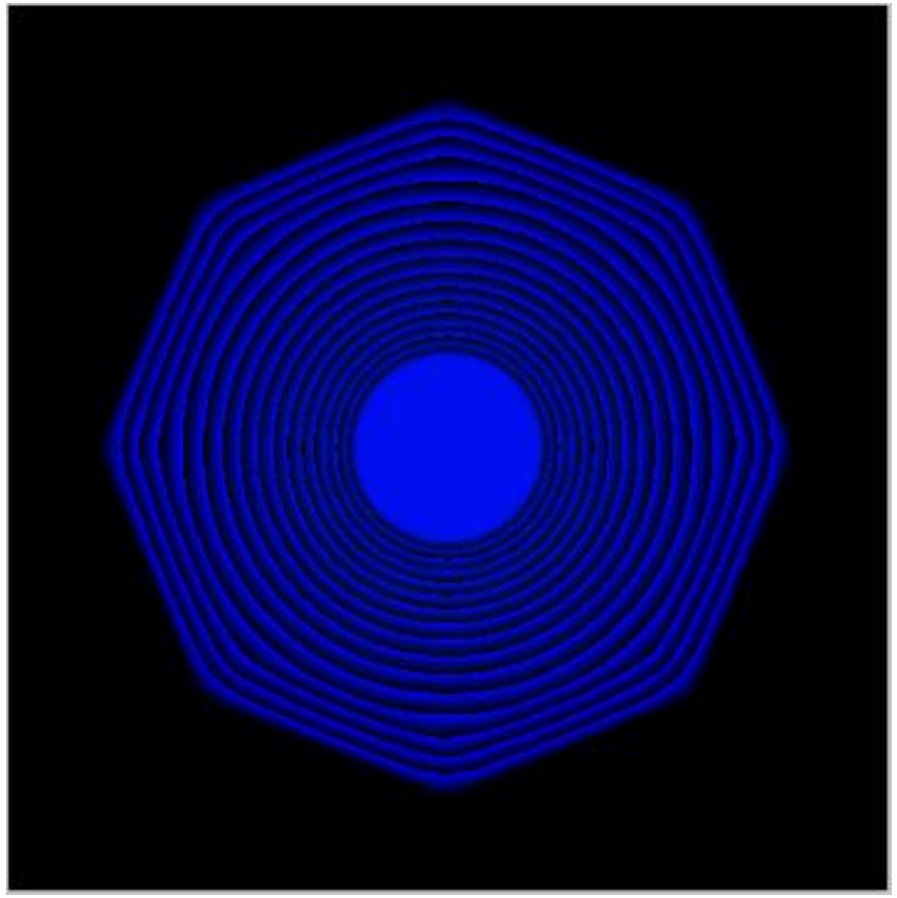
FUNGIBLE VS NON-FUNGIBLE

- Fungibility is an economic term that describes the interchangeability of certain goods. Such items can be easily exchanged for each other because their value is not tied to their uniqueness.

  For example, a dollar bill is also equal to any other dollar bill.

- Non-fungibility is about making items unique or different from each other. In NFT, each token has unique properties and is not worth the same as other similar tokens.

  For example, if you take a dollar bill and have it signed by a famous artist, it will become unique, unlike all other dollar bills, and possibly worth more than its face value.



First NFT – Quantum

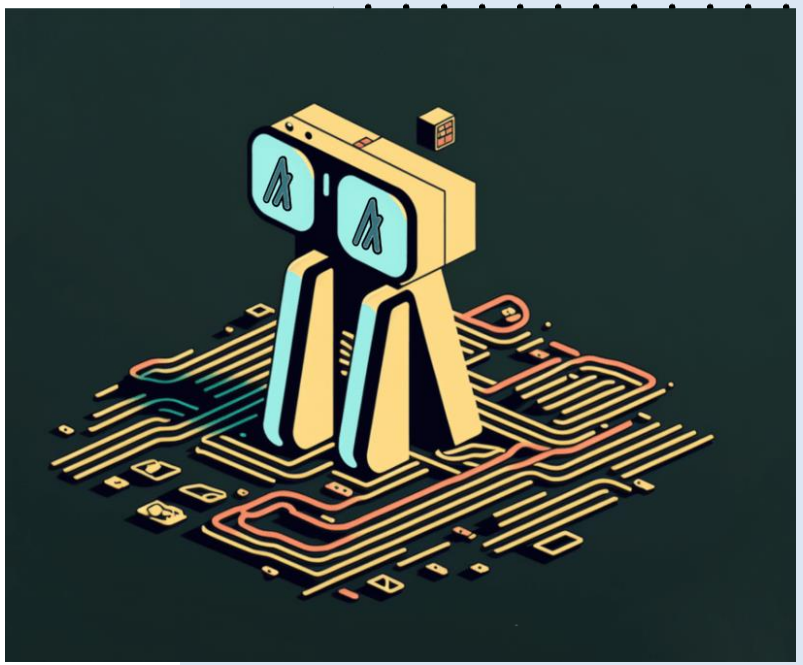
THE DIFFERENCE BETWEEN TOKENS AND CRYPTOCURRENCY

- **Platform:** Cryptocurrencies are the native asset of a particular blockchain protocol, while tokens are created by platforms built on top of those blockchains. For example, the Ethereum blockchain's native token is Ether (ETH). But there are many other tokens that utilize the Ethereum blockchain, including DAI, LINK, COMP, and CryptoKitties, among others.
- **Purpose:** Cryptocurrencies are primarily used as a means of exchanging or storing value, while tokens can have a wide range of uses, including utilities and asset representation.
- **Technical Implementation:** Cryptocurrencies function as independent currencies, while tokens function as part of a larger ecosystem, using smart contracts to provide their functions.

ALGORAND STANDARD ASSETS

In the context of Algorand, an asset is a digital representation of a valuable item that can be transferred and stored on the Algorand blockchain. ASAs are digital assets in a chain built on the Algorand blockchain.

These assets can represent anything from stablecoins and cryptocurrencies to tokenized real assets such as real estate or art.



KEY ASA CHARACTERISTICS

- **Versatility:** ASAs allow you to create a wide range of assets, including:
  - Fungible tokens
  - Game points
  - Loyalty and reward points
  - Stablecoins
  - Tokenized securities
  - NFTs, etc.
- **Easy to create:** ASAs do not require writing smart contracts! ASAs can be minted and customized using simple transactions (SDK/CLI).
- **Customizability:** users can configure various asset parameters such as issue quantity, transferability, and other attributes.

01

For each asset created by or belonging to an account, the minimum balance of that account is increased by 0.1 Algo (100,000 microAlgo).

02

If any transaction is made that violates the minimum balance requirements, the transaction will fail.

03

Before a new asset can be transferred to a particular account, the recipient must opt-in to receive the asset.

04

The number of assets that can be created or used by one account is unlimited.

ASA IMMUTABLE PARAMETERS

These eight parameters can be specified only when creating an asset:

Parameter	Description	Mandatory
Creator	Specifies the account that creates the asset.	Yes
AssetName	Extended asset name. Helps identify the asset to users.	No, but recommended
UnitName	Short name of the asset unit. Used to indicate the asset unit (e.g. "USD", "BTC").	No, but recommended
Total	The total number of units of the asset to be issued.	Yes
Decimals	Specifies how many decimal places an asset unit can have.	Yes
DefaultFrozen	Provides control over whether new assets will be automatically frozen until further approval.	Yes
URL	Indicates additional information or a website related to the asset.	No
MetaDataHash	Asset metadata hash.	No

Source:https://developer.algorand.org/docs/get-details/asa/#asset-parameters

MODIFIABLE ASSET PARAMETERS¶

Algorand Standard Assets (ASA) have several parameters that **can be changed** after the asset is created. These parameters correspond to addresses that are authorized to perform certain functions for the asset. While these addresses must be specified when the asset is created, they can also be changed later.

Parameter	**Description**
Manager Address	The only account that can authorize transactions to modify or destroy an asset.
Reserve Address	A reserve address stores unissued (unmoved) units of an asset. This means that all assets that have not yet been transferred to other users are stored in the reserve account.
Freeze Address	The account that has the authority to freeze or unfreeze assets for a specific account. If empty, freezing is not allowed.
Clawback Address	An account that allows assets to be transferred from and to any asset owner (if they have opted-in to the asset). A Clawback Address is typically used to reclaim assets from users.

Source:https://developer.algorand.org/docs/get-details/asa/#mutable-asset-parameters

ASSET CREATION

There are 3 ways to create your own asset on the Algorand blockchain:

- Via Algorand SDK
- Via “goal” CLI
- Via Dappflow

CREATING ASA

Using the SDK to create your own ASA

Step 1: creating a transaction in which all the asset parameters are defined.

Can create a transaction: any account with sufficient Algo balance

```
const suggestedParams = await algodClient.getTransactionParams().do();
const txn = algosdk.makeAssetCreateTxnWithSuggestedParamsFromObject({
  from: creator.addr,
  suggestedParams,
  defaultFrozen: false,
  unitName: 'rug',
  assetName: 'Really Useful Gift',
  manager: creator.addr,
  reserve: creator.addr,
  freeze: creator.addr,
  clawback: creator.addr,
  assetURL: 'http://path/to/my/asset/details',
  total: 1000,
  decimals: 0,
});
```


Source: https://developer.algorand.org/docs/get-details/asa/

CREATING ASA

Using the SDK to create your own ASA

Step 2: signing the transaction and sending it to the network

```
const signedTxn = txn.signTxn(creator.privateKey);
await algodClient.sendRawTransaction(signedTxn).do();
const result = await algosdk.waitForConfirmation(
  algodClient,
  txn.txID().toString(),
  3
);

const assetIndex = result['asset-index'];
console.log(`Asset ID created: ${assetIndex}`);
```


Source: https://developer.algorand.org/docs/get-details/asa/

CREATING NFT

01

NFTs are created using an ASA creation transaction, just like fungible tokens.

02

There is no need to create a special smart contract for NFTs.

03

You need to set the total number of units you want to create for this asset to 1 and the number of decimal places to 0. This ensures that you create exactly one unit of your ASA and cannot split the newly created asset.

TOKEN STANDARDS FOR NFTS ON ALGORAND

ARC-3 (or ARC-0003) is the primary token standard for creating NFTs on Algorand. It defines how metadata is stored off-chain and associates it with the asset via a URL. It provides compatibility with NFT marketplaces on Algorand.

- **Metadata Storage:** Metadata such as name, description, and attributes are stored off-chain, typically on IPFS or other decentralized or centralized servers.
- **Identification:** ARC-3 tokens can be identified by the asset URL ending in #arc3.
- **Mass Minting:** ARC-3 supports mass minting, useful for handling large amounts of metadata such as videos or other large formats.
- **Marketplace Support:** ARC-3 is supported by all Algorand NFT marketplaces.
- **Cost:** Creating and owning an ARC-3 ASA requires a lock of approximately 0.1 Algo for each creator and owner.

Source: https://algorand.co/learn/arc-token-standards-explained-for-nft-creators

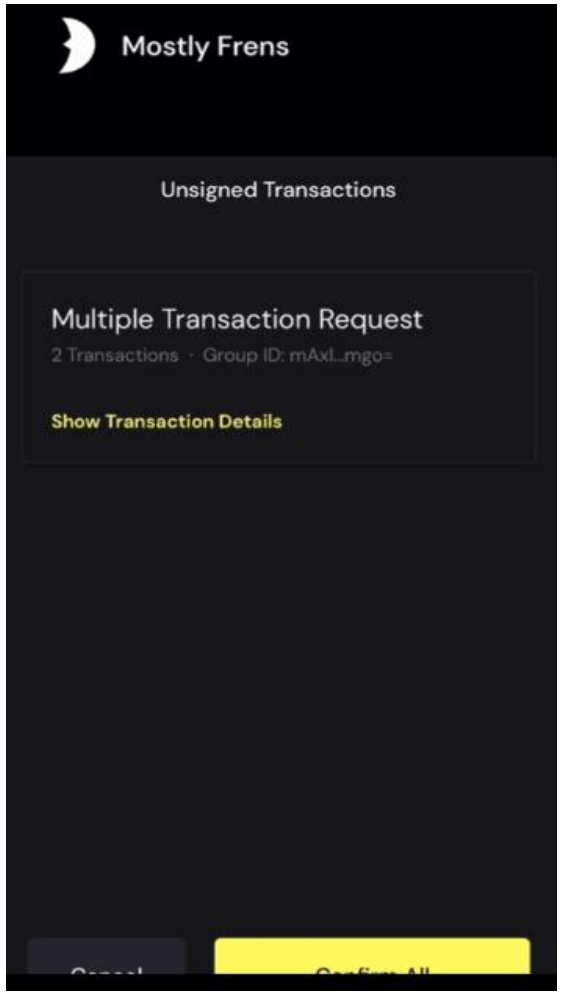
TOKEN STANDARDS FOR NFTS ON ALGORAND

ARC-19: upgradable or dynamic NFTs

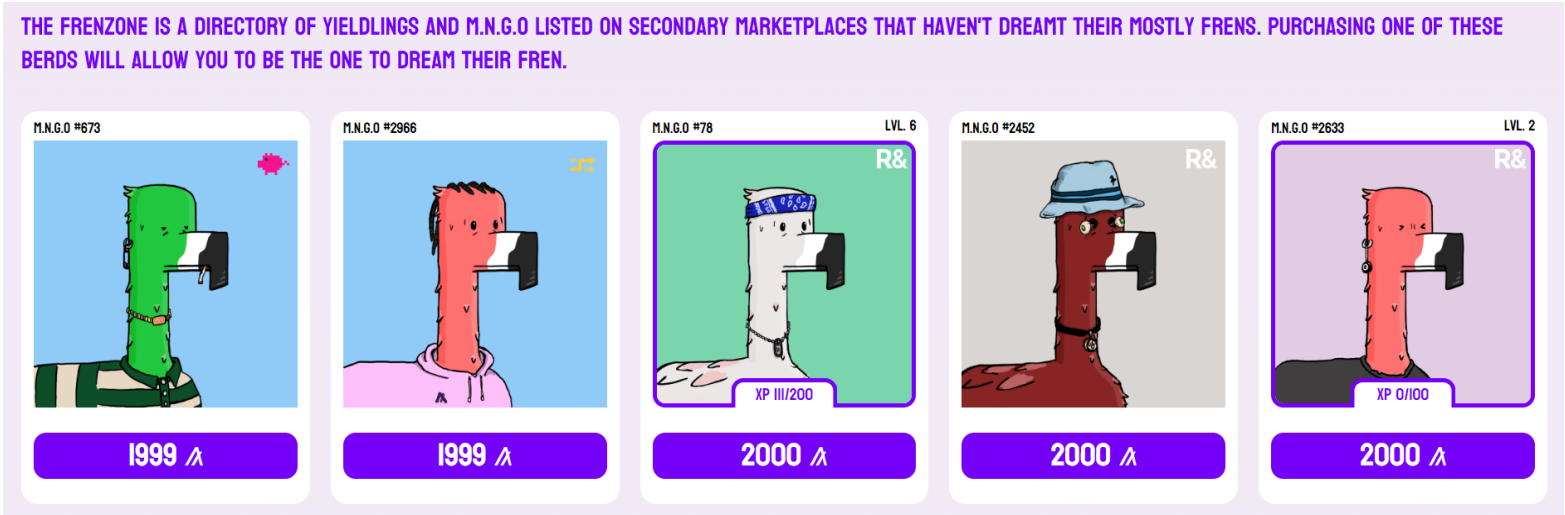
- **Functionality:** ARC-19 allows you to update metadata or digital media associated with an NFT after creation, providing dynamic and interactive NFTs.
- **Use cases:** Suitable for NFTs that need to evolve over time or require manual updates.
- **Tools:** Tools like AlgoKit and Evil Tools can make it easier to create and manage ARC-19 NFTs.

Джерело: https://algorand.co/learn/arc-token-standards-explained-for-nft-creators

ARC-19 EXAMPLE



Mostly Frens is an example of ARC-19. It is a dynamically changing collection of PFPs where Frens evolve as tokens are sent to them.



Source: 9 Dynamic NFT Collections

TOKEN STANDARDS FOR NFTS ON ALGORAND

ARC-72 is a standard for creating NFTs using smart contracts on Algorand.

- **NFT Smart Contract:** allows creation of NFTs that leverage smart contract capabilities, providing more sophisticated and customizable features than traditional ASAs.
- **No Opt-in Transaction Required:** ARC-72 eliminates the need for creators and collectors to opt-in to receive ASA tokens (opt-in transaction).
- **Programming Knowledge:** developers need programming skills to deploy smart contract-based NFTs.
- **Similarity to ERC-721:** works similarly to Ethereum's ERC-721 standard.
- **Cost-Effectiveness:** ARC-72 requires only the creator to lock in 0.1 Algo, not the owner. In contrast, ASA requires both the creator and the owner of the NFT to have a minimum balance of 0.1 Algo.

Source: https://algorand.co/learn/arc-token-standards-explained-for-nft-creators

TOKEN STANDARDS FOR NFTS ON ALGORAND

ARC-18 is the standard for paying royalties on NFTs. It should be used in combination with ARC-3 and ARC-19.

- ARC-18 allows creators to receive royalty payments every time their NFT is sold.
- ARC-18 means that creators don't have to rely on marketplaces to facilitate royalty payments.
- The creator specifies the royalty rate during the initial minting of the NFT.

Source: https://algorand.co/learn/arc-token-standards-explained-for-nft-creators

HOW DO I CHOOSE THE RIGHT ARC?

Requirements	**ARC token standard**
I want the image and metadata to always remain the same.	ARC-3
I want to program my NFT to change over time.	ARC-19
I want to update the images and metadata in the future.	ARC-3 and ARC-19
I want the images and metadata to always remain the same. I also want to receive royalty payments every time my NFTs are traded.	ARC-3 and ARC-18

Source: https://algorand.co/learn/arc-token-standards-explained-for-nft-creators

ARC (Algorand Request for Comments) is a set of standards and guidelines designed to facilitate the development of applications and assets on the Algorand blockchain. It is presented in the form of a design document that provides information to the Algorand community or describes a new Algorand feature, its processes and environment. ARCs must be referenced when creating fungible and non-fungible tokens, developing applications, conducting transactions, and generally anything that requires ecosystem consensus to work.

NFTs can represent any unique digital or physical item. This representation is usually done by assigning files to the NFT that represent it. However, it is important to note that these files are not stored on the blockchain. This would have economic and technical implications, as blockchains are not designed to store large files.

Instead, developers store files representing NFTs using cloud storage solutions.

IPFS

The InterPlanetary File System, or IPFS, is a distributed system for downloading, storing and accessing websites, applications, data and files.

What is IPFS?

- **Decentralised file system**: IPFS uses P2P (peer-to-peer) technology to store and transfer files between users without the need for centralised servers.
- **Content addressing**: files in IPFS are addressed using their hashes (encrypted values), which makes them unique and immutable. This means that each file or version of a file has a unique identifier based on its content. Thus, all pieces of content in the IPFS ecosystem have a unique hash that serves as a content identifier (CID).

WHY DO I NEED IPFS WHEN CREATING AN NFT?

- **Decentralised storage:** many NFTs refer to multimedia content (images, video, audio, etc.). Storing these files on centralised servers makes them vulnerable to loss or modification. Using IPFS provides long-term data storage without centralised dependency.
- **Data immutability**: thanks to content addressing, files on IPFS cannot be changed after they are uploaded. This ensures trust in the authenticity and uniqueness of the NFT.
- **Availability and resilience**: files on IPFS can be duplicated and stored on multiple network nodes, ensuring high availability and resilience to data loss.

Currently, IPFS is the main protocol for storing digital media, which is represented by many NFTs. In short, an NFT is on the blockchain and points to IPFS media files in its metadata.

HOW DOES IT WORK?

Each file that is added to IPFS is assigned a unique address based on a hash of the file's contents. This address is called the Content Identifier (CID), and it combines the file hash and the unique identifier of the hashing algorithm used into a single string.

As a result, if you upload the same file again, it will always receive the same IPFS/CID hash. This means that even if the third party you used to upload your tokens no longer supports your files and has "deleted" them, you can re-upload the files yourself and NFT will now extract the media files correctly.

EXAMPLE

1. Create a JSON metadata file: create a JSON file containing the metadata for your NFT. This file must have a specific format. Here is an example of the format (ARC-3 JSON Metadata file):

```
{
    name: 'NFT::ARC3::IPFS::1@arc3',
    description: 'This is a Scenario1 NFT created with metadata JSON in ARC3 compliance and using
IPFS via Pinata API',
    image: 'ipfs://QmQp7k5JrwQJ2XLfmvEDzmwsTgKaLxzNUNh5eZWmXEAHSe',
    image_integrity: 'sha256-48DEQpj8HBSa...',
    image_mimetype: 'image/png',
    external_url: 'https://github.com/emg110/arc3ipfs',
    properties: {
        file_url: 'arc3-asa',
        file_url_integrity: 'sha256-48DEQpj8HBSa...',
        file_url_mimetype: 'image/png'
    }
}
```

2. Upload the JSON metadata file to IPFS (for example, Pinata). After uploading, you will receive a CID for the JSON file.

3. Use the CID of the JSON metadata file as the URL in the NFT minting process.

ipfs://YOUR_METADATA_CID

*The resource URL (au) indicates the URI of the JSON metadata file. If you use IPFS, you should use only the standard IPFS URI (ipfs://...), not the gateway format (https://ipfs.io/ipfs/...). For reasons of web security standards, http URIs should not be used. If the asset name does not end in @arc3, the asset URL must end in #arc3.

MODIFICATION OF AN ASSET¶

Creates a transaction:

account - Manager Address for this asset

```
const manager = accounts[1];

const configTxn = algosdk.makeAssetConfigTxnWithSuggestedParamsFromObject({
  from: creator.addr,
  manager: manager.addr,
  freeze: manager.addr,
  clawback: manager.addr,
  reserve: undefined,
  suggestedParams,
  assetIndex,
  // don't throw error if freeze, clawback, or manager are empty
  strictEmptyAddressChecking: false,
});

const signedConfigTxn = configTxn.signTxn(creator.privateKey);
await algodClient.sendRawTransaction(signedConfigTxn).do();
const configResult = await algosdk.waitForConfirmation(
  algodClient,
  txn.txID().toString(),
  3
);
console.log(`Result confirmed in round: ${configResult['confirmed-round']}`);
```


Source: https://developer.algorand.org/docs/get-details/asa/

APPROVAL TO RECEIVE AN ASSET

An opt-in transaction is simply a transfer of assets with a value of 0 to and from the receiving account. The minimum balance for this account is increased by 100,000 microAlgos.

Creates the transaction: the account that agrees to receive the asset

```
// opt-in is simply a 0 amount transfer of the asset to oneself
const optInTxn = algosdk.makeAssetTransferTxnWithSuggestedParamsFromObject({
  from: receiver.addr,
  to: receiver.addr,
  suggestedParams,
  assetIndex,
  amount: 0,
});

const signedOptInTxn = optInTxn.signTxn(receiver.privateKey);
await algodClient.sendRawTransaction(signedOptInTxn).do();
await algosdk.waitForConfirmation(algodClient, optInTxn.txID().toString(), 3);
```


Source: https://developer.algorand.org/docs/get-details/asa/

ASSET TRANSFER

Assets can be transferred between accounts that have agreed to receive the assets.

Creates a transaction: an account that stores the asset to be transferred.

```
const xferTxn = algosdk.makeAssetTransferTxnWithSuggestedParamsFromObject({
  from: creator.addr,
  to: receiver.addr,
  suggestedParams,
  assetIndex,
  amount: 1,
});

const signedXferTxn = xferTxn.signTxn(creator.privateKey);
await algodClient.sendRawTransaction(signedXferTxn).do();
await algosdk.waitForConfirmation(algodClient, xferTxn.txID().toString(), 3);
```


Source: https://developer.algorand.org/docs/get-details/asa/

OBTAINING INFORMATION ABOUT AN ASSET¶

Get asset configuration information from the network using

- SDK
- 'goal'
- blockchain explorer

OBTAINING INFORMATION ABOUT THE ASSET¶

You can get information about an asset using the Algorand JS SDK as follows:

```
const assetInfo = await algodClient.getAssetByID(assetIndex).do();
console.log(`Asset Name: ${assetInfo.params.name}`);
console.log(`Asset Params: ${assetInfo.params}`);
```

Result:

```
Asset Name: Test Algorand Standart Asset
Asset Params: {
  "clawback": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4O54",
  "creator": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4O54",
  "decimals": 0,
  "default-frozen": false,
  "freeze": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4O54",
  "manager": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4O54",
  "name": "Test Algorand Standart Asset",
  "name-b64": "VGVzdCBBbGdvcmFuZCBTdGFuZGFydCBBc3NldA==",
  "reserve": "FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4O54",
  "total": 1000,
  "unit-name": "TASA",
  "unit-name-b64": "VEFTQQ==",
  "url": "http://path/to/my/asset/details",
  "url-b64": "aHR0cDovL3BhdGgvdG8vbXkvYXNzZXQvZGV0YWlscw=="
}
```

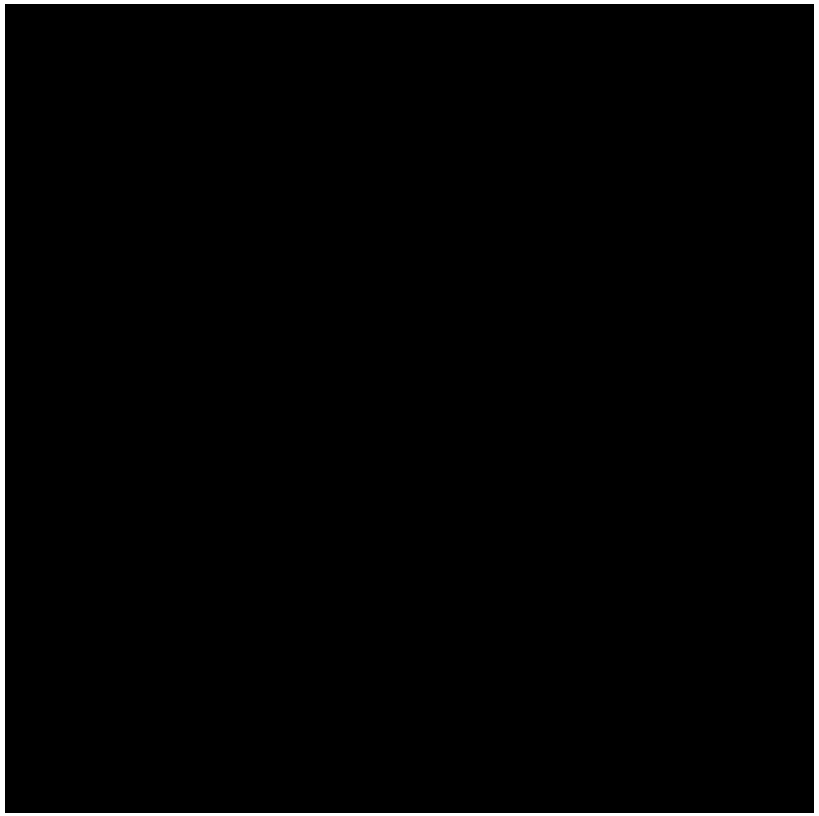
BLOCKCHAIN EXPLORER

Blockchain explorer is a web-based tool that allows users to browse and search the public blockchain ledger. It provides detailed information about blocks, transactions, accounts and assets on the Algorand blockchain.

AlgoExplorer

Pera Algorand Explorer

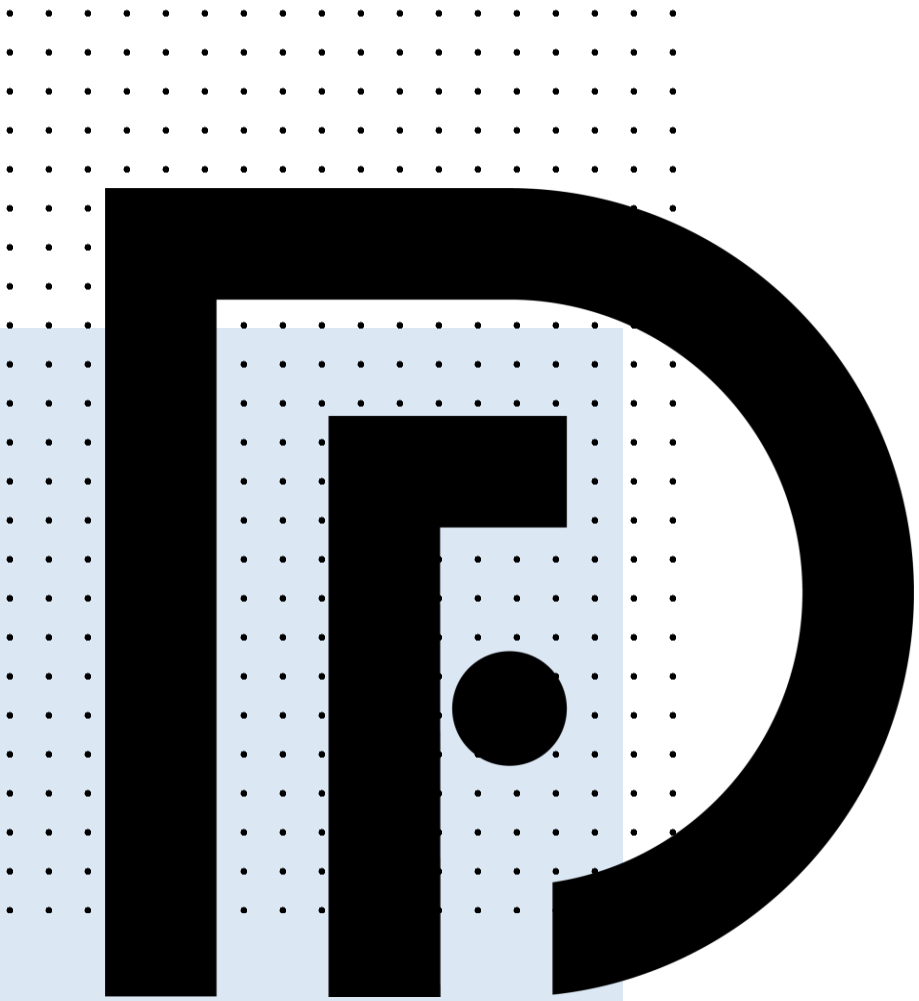
Allo.info



CREATING AN ASSET. **DAPPFLOW**

Dappflow is a web-based user interface that allows you to visualise accounts, transactions, assets, and applications in the Algorand network, as well as deploy and invoke smart contracts.

Dappflow also supports the creation of Algorand Standard Assets



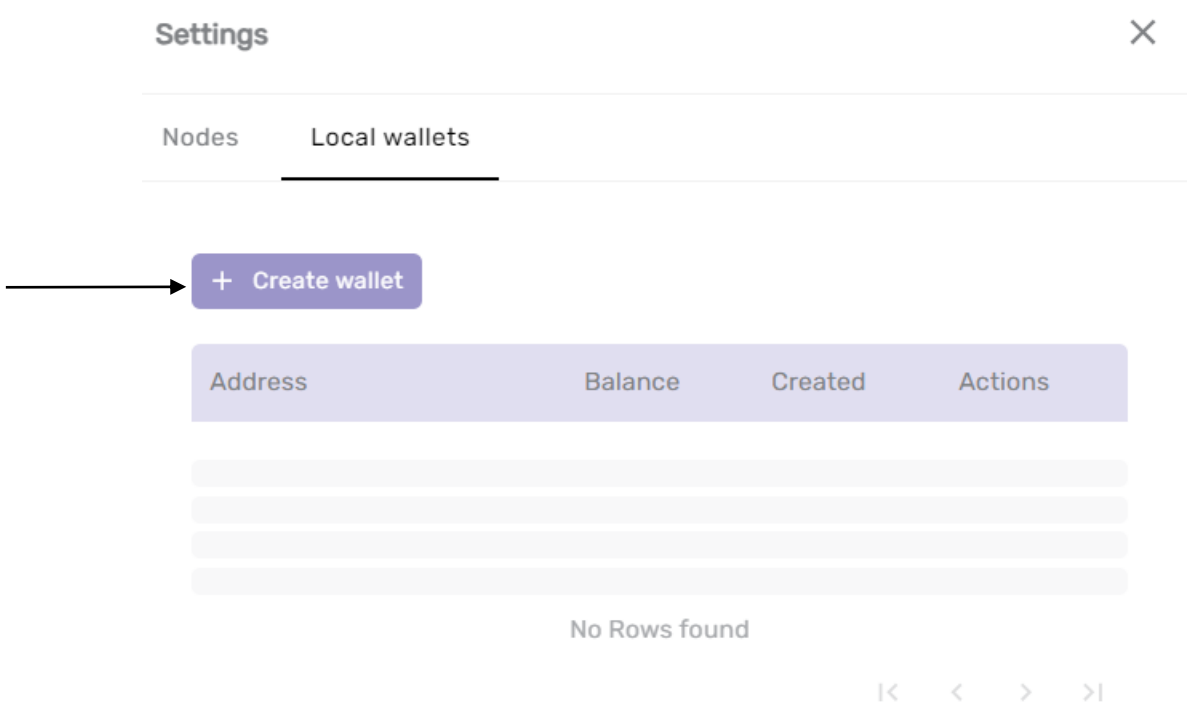
CREATING AN ASSET. DAPPFLOW

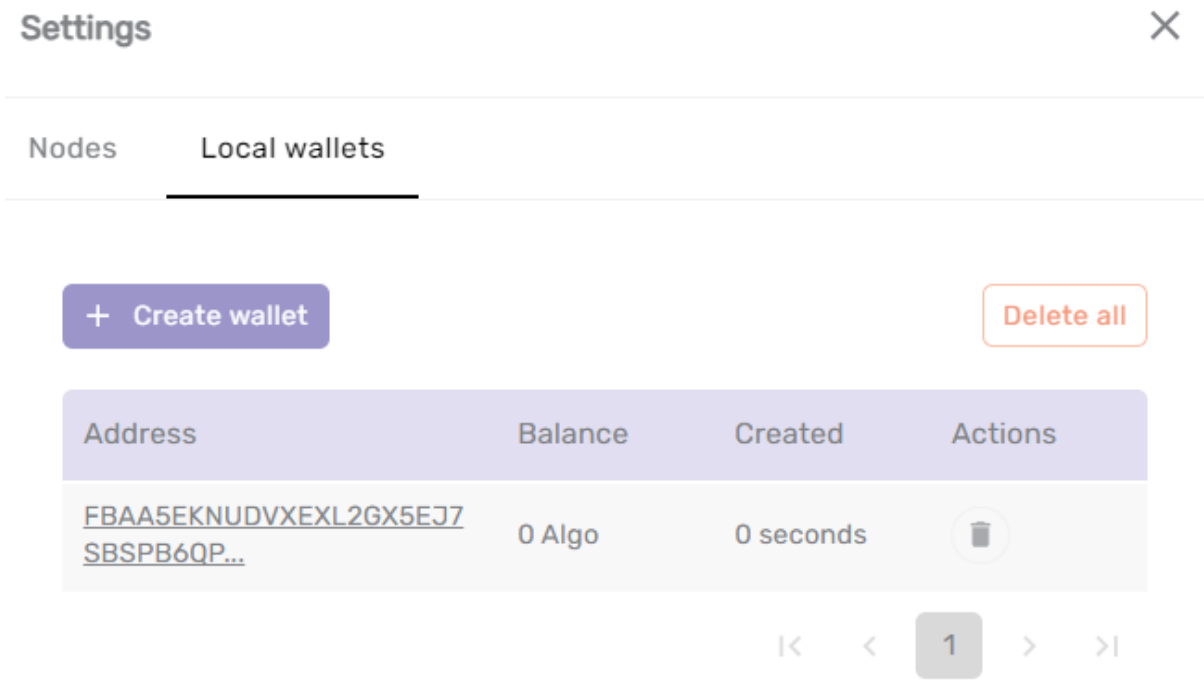
Dappflow is a web-based user interface that allows you to visualise accounts, transactions, assets, and applications in the Algorand network, as well as deploy and invoke smart contracts.

Dappflow also supports the creation of Algorand Standard Assets

CREATION OF AN ASSET. DAPPFLOW

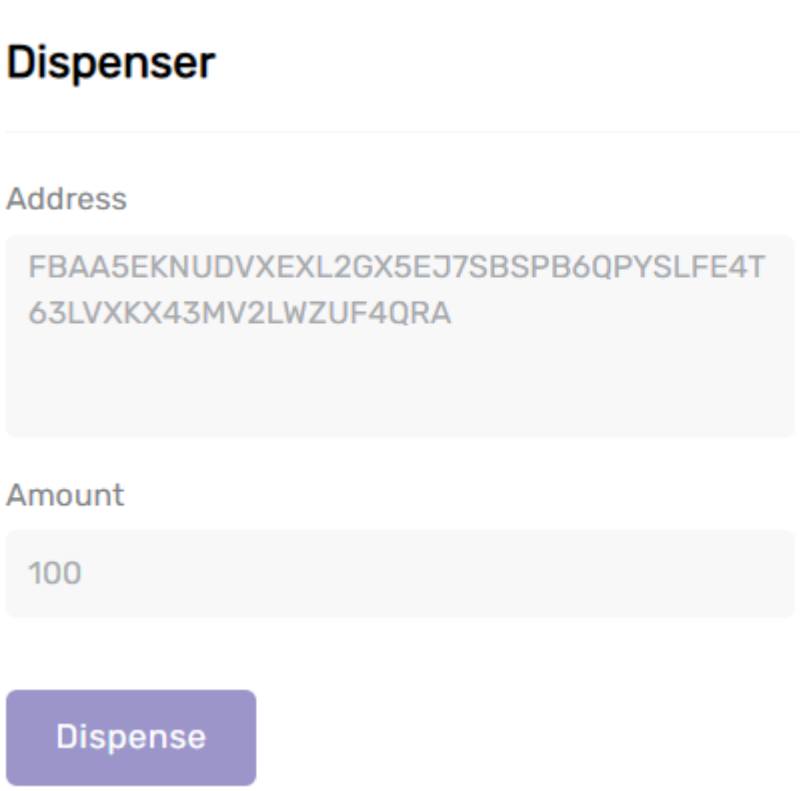
STEP 1: Create a test wallet

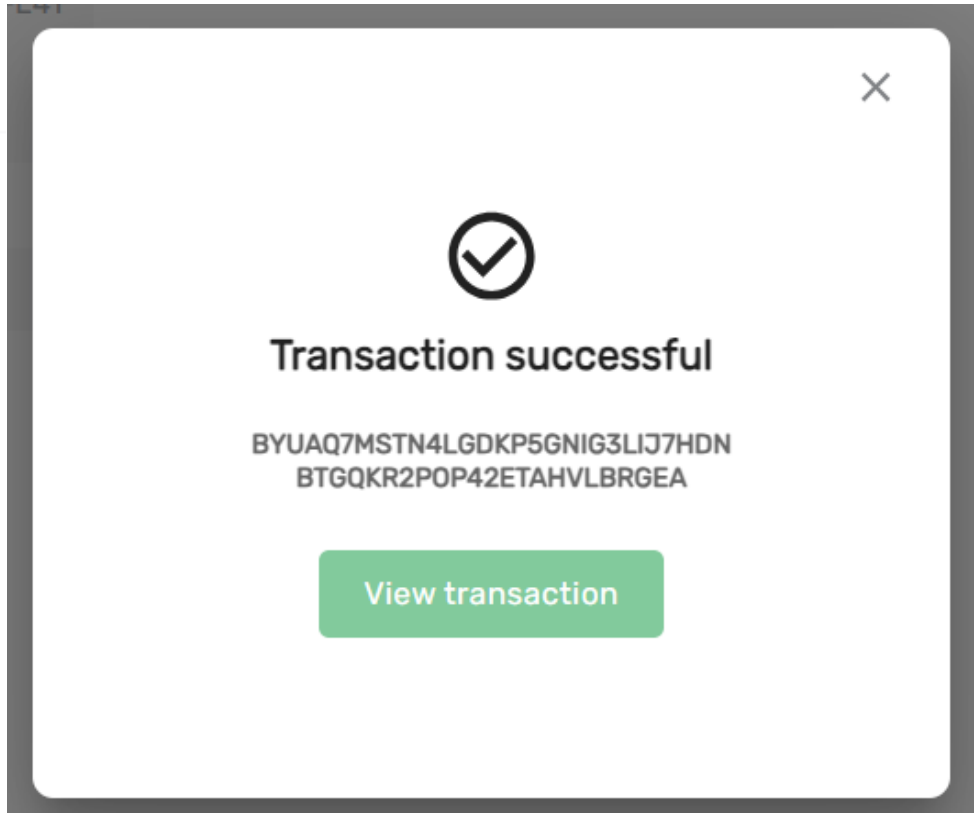




CREATION OF AN ASSET. DAPPFLOW

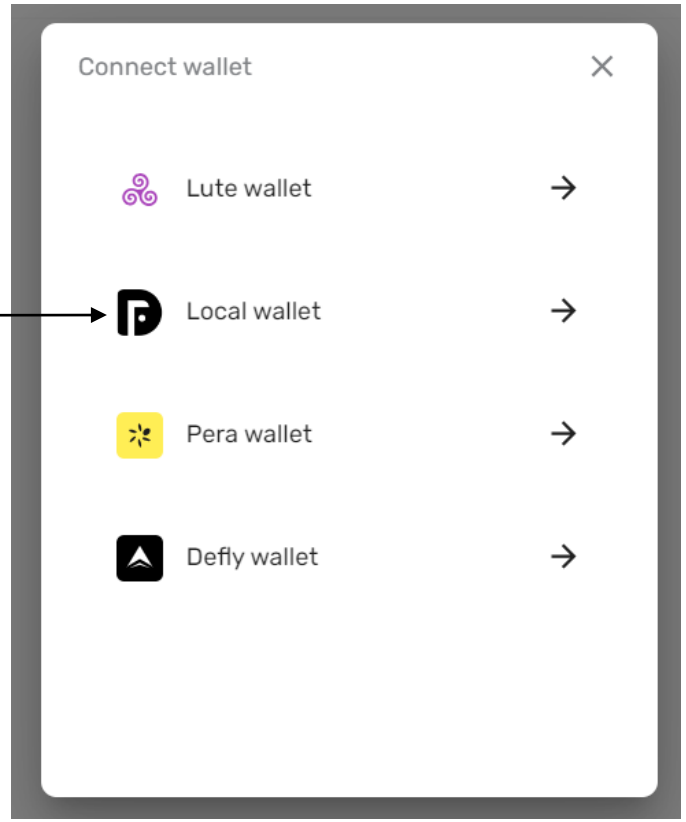
STEP 2: Top up your test wallet via Dispenser

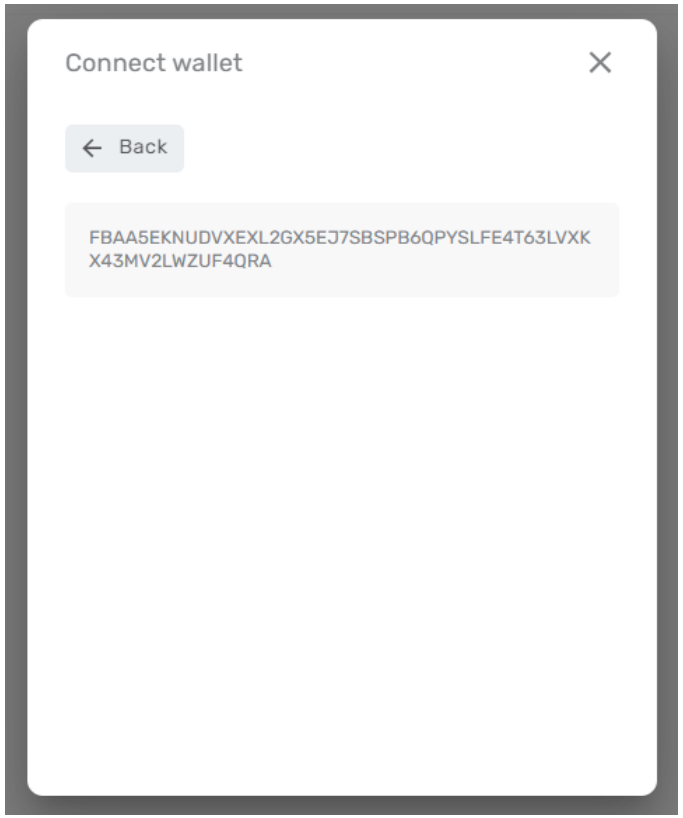




CREATION OF AN ASSET. DAPPFLOW

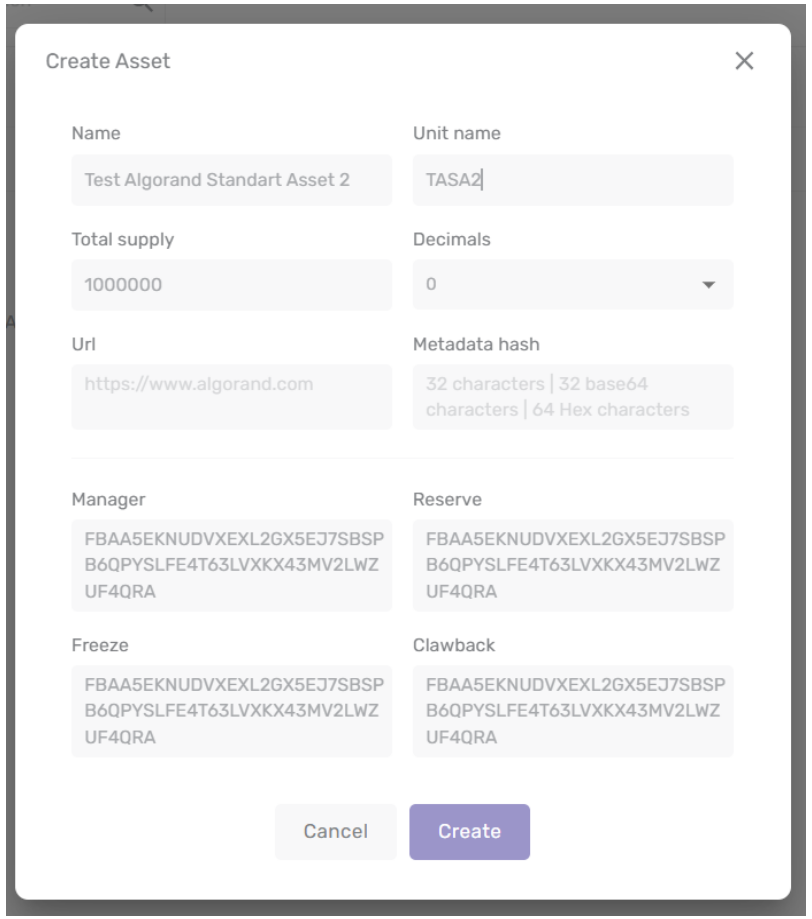
STEP 3: Go to Asset manager and add the created wallet

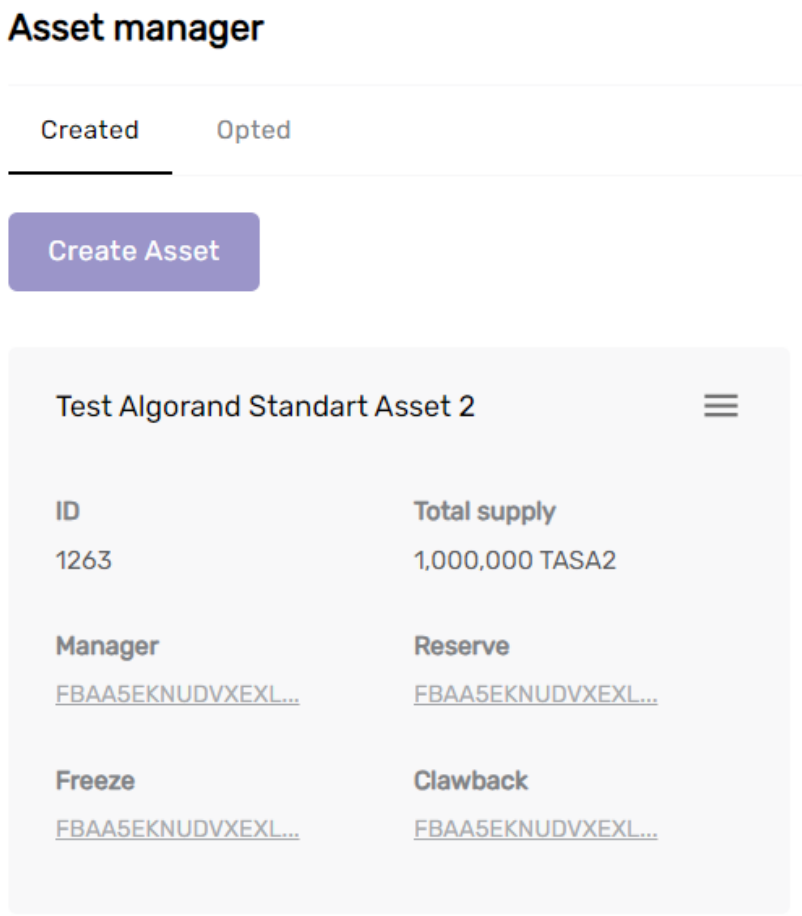




CREATION OF AN ASSET. DAPPFLOW

STEP 4: Define the parameters of the new asset





Information about the asset presented in Dappflow

TA

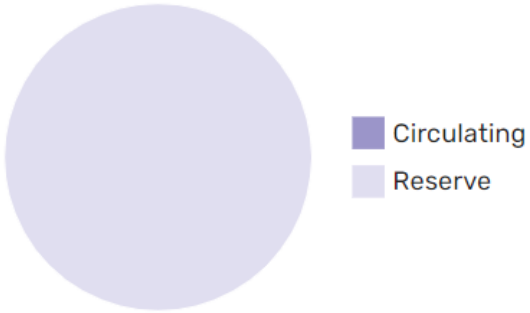
Asset #1263

Test Algorand Standart Asset 2

✓ Freeze ✓ Clawback

Raw data

Unit name	TASA2
Creator	FBAA5EKNUDVXEXL2GX5E...
Manager	FBAA5EKNUDVXEXL2GX5E...
Reserve	FBAA5EKNUDVXEXL2GX5E...
Freeze	FBAA5EKNUDVXEXL2GX5E...
Clawback	FBAA5EKNUDVXEXL2GX5E...
URL	--Empty--
Default frozen	No
Creation txn	7HVCF3JD3YAR7X5DKFDP...
Creation block	#58529 3 weeks, 18 hours ago



Total supply	1 Million
Circulating supply	0
Reserve supply	1 Million
Decimals	0

Transactions | Opted accounts

Transaction ID	Block	Age	From	To	Amount

No Rows found

PERA WALLET

Pera Wallet is a non-custodial wallet for buying, sending, and storing cryptocurrency on the Algorand blockchain.

It has mobile and web versions.

Pera Wallet is a container for your Algorand accounts

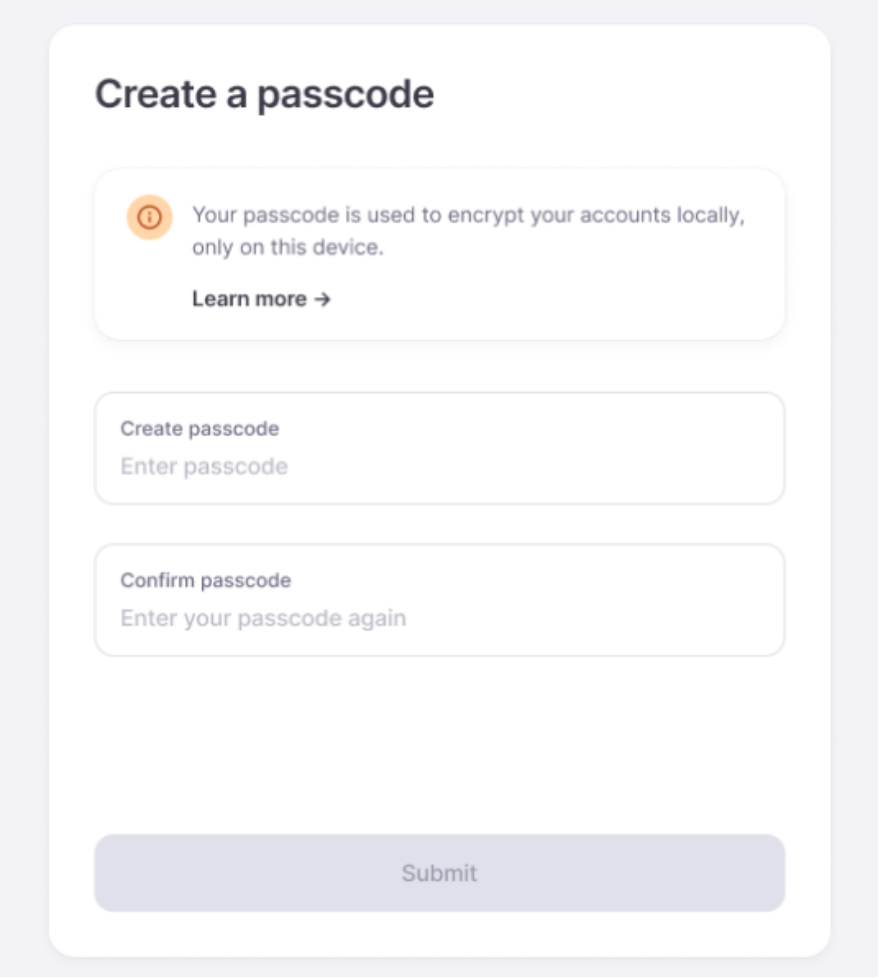
Pera Wallet allows you to create and manage Algorand accounts, but the accounts exist on the blockchain, not "in Pera"

Pera Wallet is a non-custodial wallet. Any accounts created using Pera remain the responsibility of the users and Pera will not be able to assist if access to the accounts is lost.

CREATE AN ACCOUNT IN PERA WALLET

STEP 1: create a password

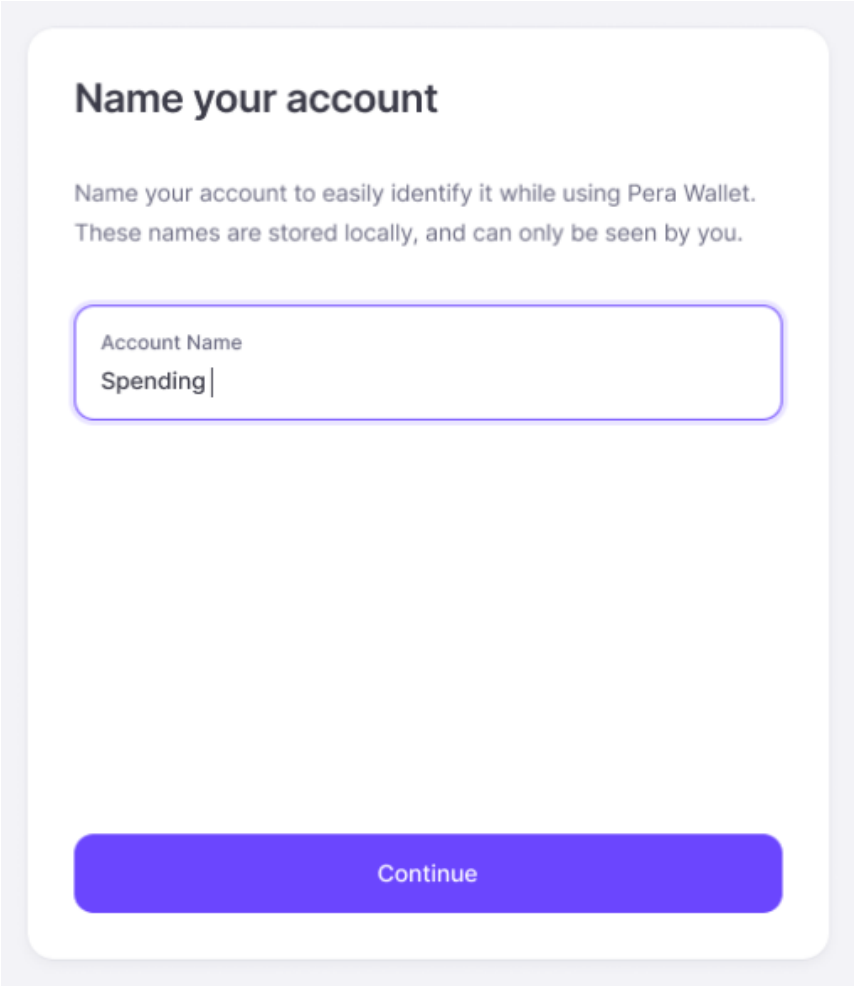
The Pera Web Wallet password is used to unlock the wallet ONLY in this browser and on this device where it was created. It will not work in incognito mode or on another computer.



CREATE AN ACCOUNT IN PERA WALLET

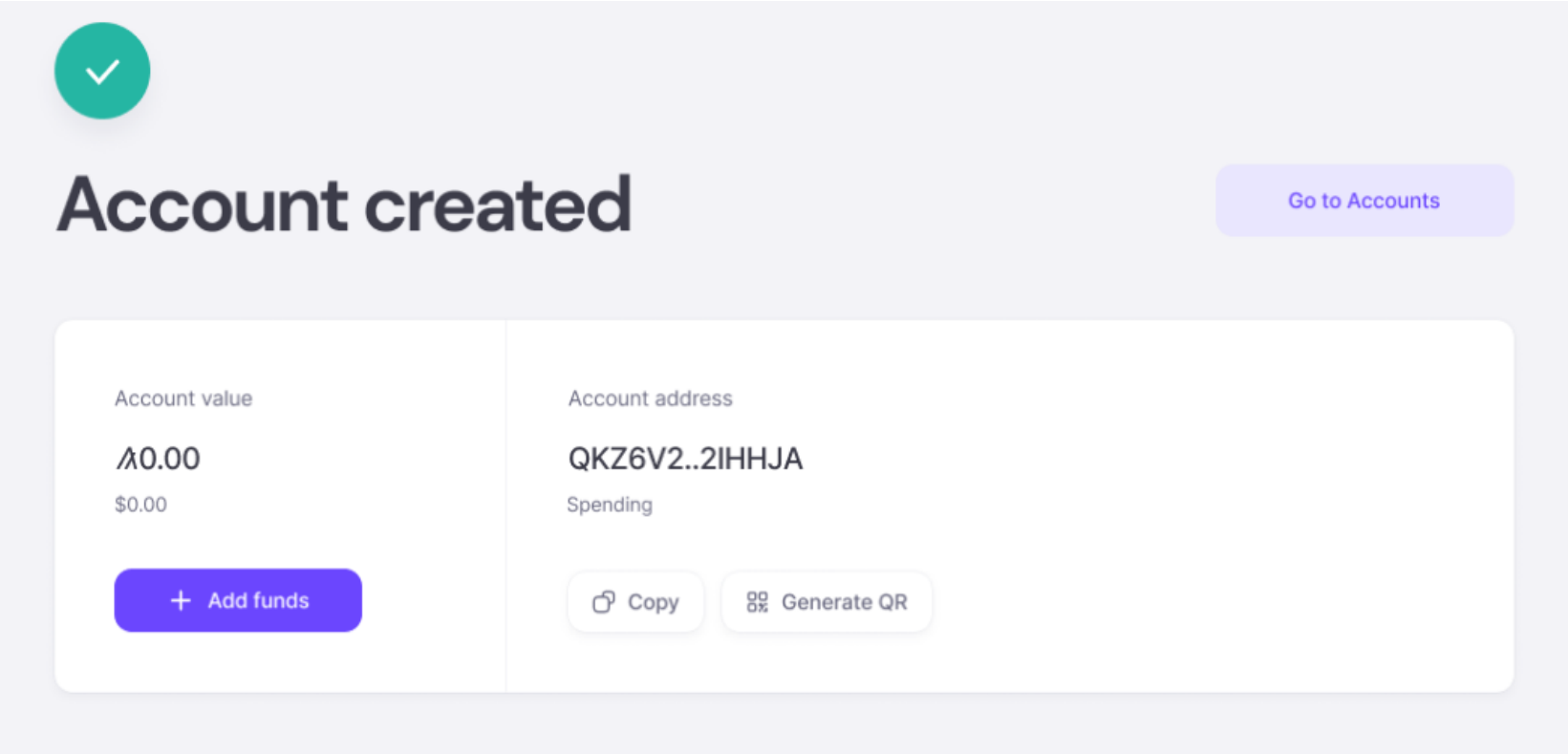
STEP 2: Name your account

This name only applies to your web wallet - it will never be used on a blockchain or any other wallet or website.



CREATE AN ACCOUNT IN PERA WALLET

STEP 3: Ready-made account



Each account has its own recovery passphrase (mnemonic). They should be kept separate, and each one provides access to only one Algorand account.

ADDITIONAL MATERIALS

- Intro to Assets (ASA): an introduction to assets on the Algorand blockchain (video).
- Building Solutions Using ASAs: Learn about the ASA lifecycle and see how to create fungible tokens and non-fungible tokens using the SDK, AlgoDesk.io, and non-fungible domains (NFDs) (video).
- The 4 A's of Algorand: information about accounts, assets, atomic transactions, and applications in Algorand (video).
- How to mint an NFT