Lesson 12

Today's Topics

- Anchor review
- NFTs introduction
- Inscriptions
- DeFi continued

> ### Anchor Quick Start
>
> Assumes node / npm / yarn installed
>
> Rust
>
> ```
> curl --proto '=https' --tlsv1.2 -sSf
> https://sh.rustup.rs | sh
> ```
>
> Solana CLI
>
> ```
> sh -c "$(curl -sSfL
> https://release.solana.com/v1.18.9/install)
> "
> ```
>
> Create wallet
>
> ```
> mkdir ~/my-solana-wallet
> solana-keygen new --outfile ~/my-solana-
> wallet/my-keypair.json
> ```
>
> Install Anchor

```
cargo install --git
https://github.com/project-serum/anchor avm
--locked --force
```

```
avm install latest
avm use latest
```

```
anchor --version
```

To set up a project

```
anchor init <project-name>
```

Build the project with

```
anchor build
```

Remember you will need to change the declare_id , to do this you can run

```
anchor keys list
```

then add the key to the declare_id! macro in your lib.rs
and re run

```
anchor build
```

For testing run

`anchor test`

The testing will be done on a local cluster, once you are happy you can deploy to devnet.

Change the `provider.cluster` variable in `Anchor.toml` to `devnet`.

```
[provider]
cluster = "devnet"
wallet =
"/home/extropy/.config/solana/id.json"
```

then run

`anchor deploy`

Useful Account macro attributes

Full list here

Common useful ones

- `#[account(mut)]` - Used to tell Solana you will change the data in the account and it will need to serialise this back.
- `#[account(signer)]` verifies that an account is also the signer.

- `#[acccount(has_one=target)]` this uses a constraint to ensure that a field within the accout matches what we have in the account structure.
- `#[account(owner=target)]` Similar to the previous example but ensures that owner matches the target.

Anchor Program Registry

See Docs

Search by name or address ⌘K

The Anchor Program Registry allows you to publish the source code of your programs.

Process

1. You need to create an account at apr.dev
2. Create an token, from the account section on the top right
3. Connect your anchor session
   `anchor login <token>`
4. Prepare the program details
   You need to configure your anchor.toml file
   It will look something like this

```
anchor_version = "0.25.0"

[workspace]
members = ["programs/multisig"]


[provider]
cluster = "mainnet"
wallet = "~/.config/solana/id.json"


[programs.mainnet]
multisig =
"A9HAbnCwoD6f2NkZobKFf6buJoN9gUVVvX5PoUnDH
S6u"


[programs.localnet]
multisig =
"A9HAbnCwoD6f2NkZobKFf6buJoN9gUVVvX5PoUnDH
S6u"
```

5. To publish the details to the program registry run

```
anchor publish <program name>
```

Solana Mobile Stack

This was launched in June this year alongside Saga, an Android mobile phone with features tightly integrated with Solana

The Solana Mobile Stack (SMS) provides a new set of libraries for wallets and apps, allowing developers to create rich mobile experiences on Solana, the world's most performant blockchain, and is built to run alongside Android. The SDK provides libraries and programming interfaces for Android apps and secure key private

storage, simplifying the developer experience to build and extend dApps functionality for Solana.

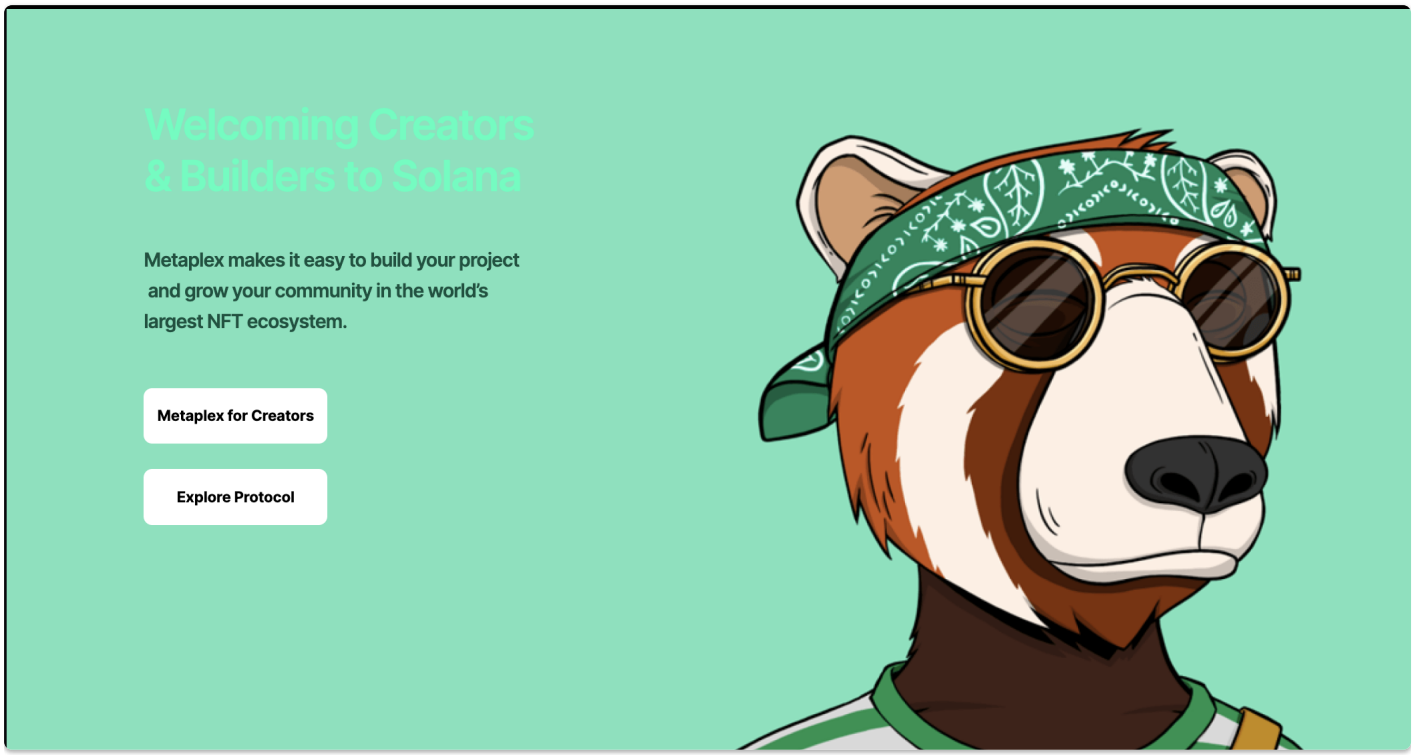
The Solana dApp Store will provide a distribution channel for apps that want to establish direct relationships with their customers, allowing them to transact without platform fees. The Solana dApp Store will co-exist with Google Play on devices, providing access to web2 and web3 apps.

To sync your phantom and file system wallet

export your 12 word phrase from phantom

```
solana-keygen recover --force 'prompt://?key=0/0' -o <keypair.json>
```

NFTs



NFT Overview

NFTs are Non Fungible Tokens
They can be used for many things, but commonly they are used to tokenise or establish ownership of digital assets.

Although the standard was established in 2018, their popularity increased greatly in 2020 / 2021 and they became a mainstay of DeFi.

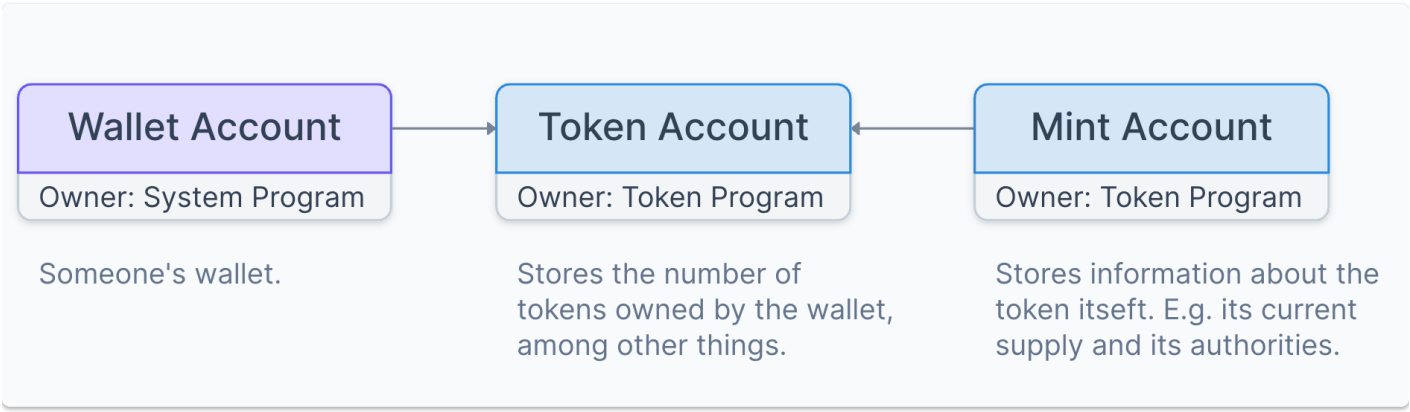
Non fungible means that each token is unique, unlike SOL for example for any one SOL is equivalent to any other.
An NFT therefore will have a means to identify it, we can then find who owns a particular NFT.

When used with digital assets, such as images the NFT will include a link to the asset. Typically the assets are stored in decentralised storage systems such as Arweave, IPFS, NFT.Storage etc. The details of the location of the asset are often referred to as the metadata of the NFT.

More recently there has become a need for semi fungible tokens, these are SPL tokens with a supply greater than 1 but which has typical NFT attributes such as an image.

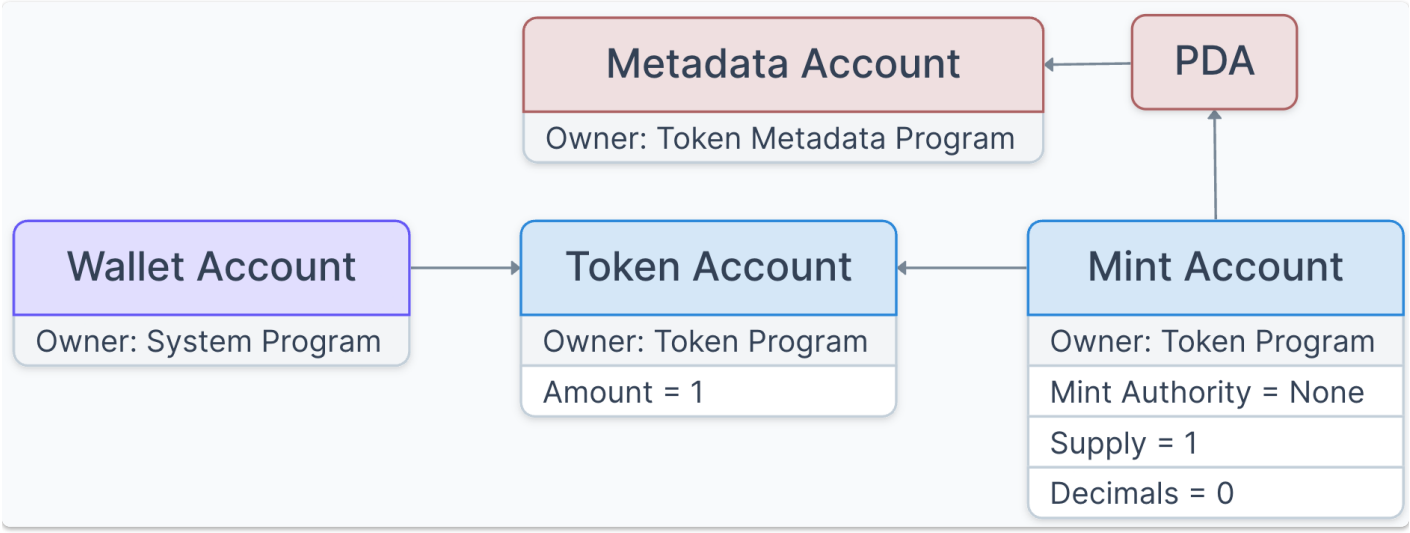
NFTs on Solana

See Documentation

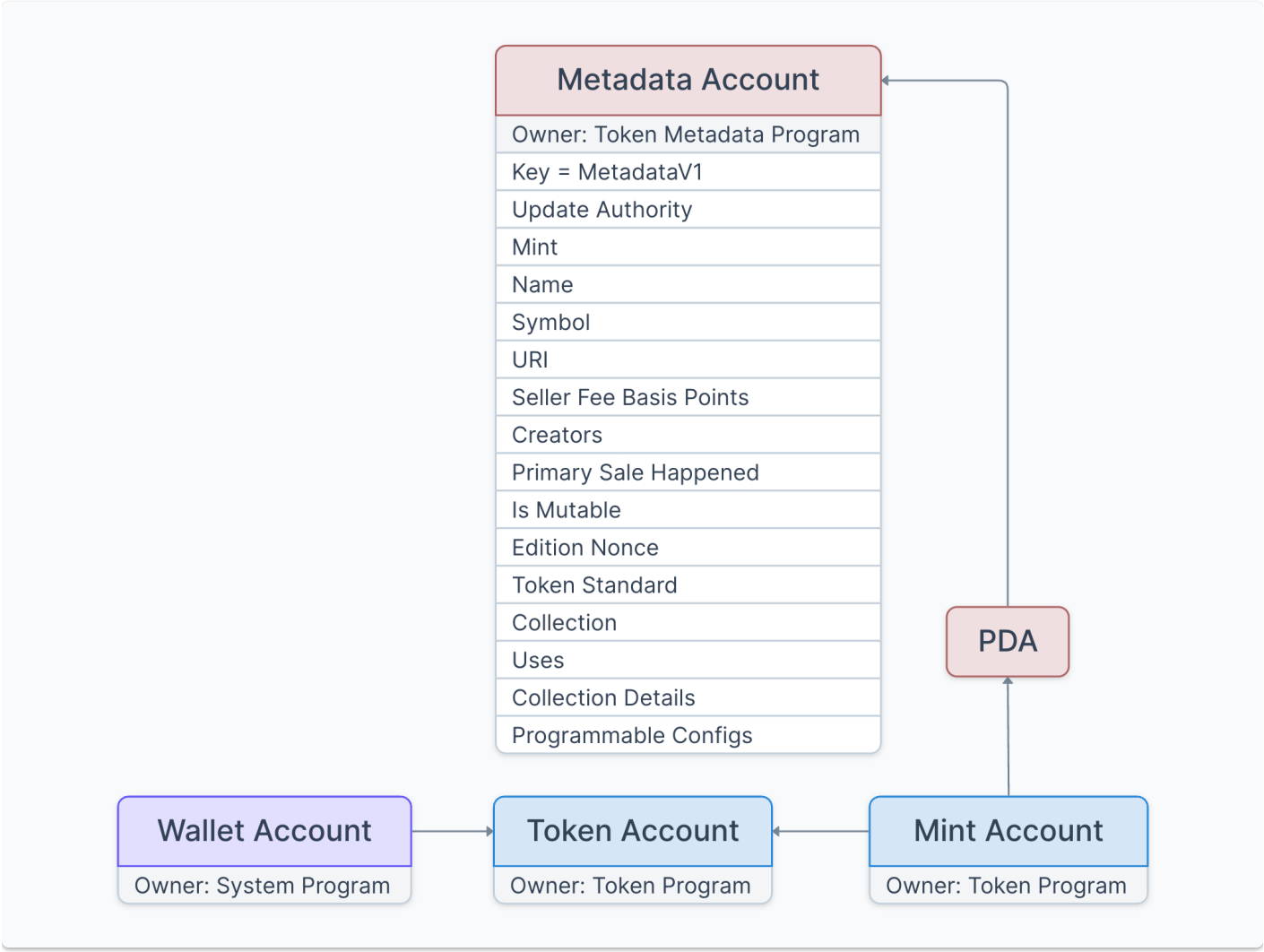


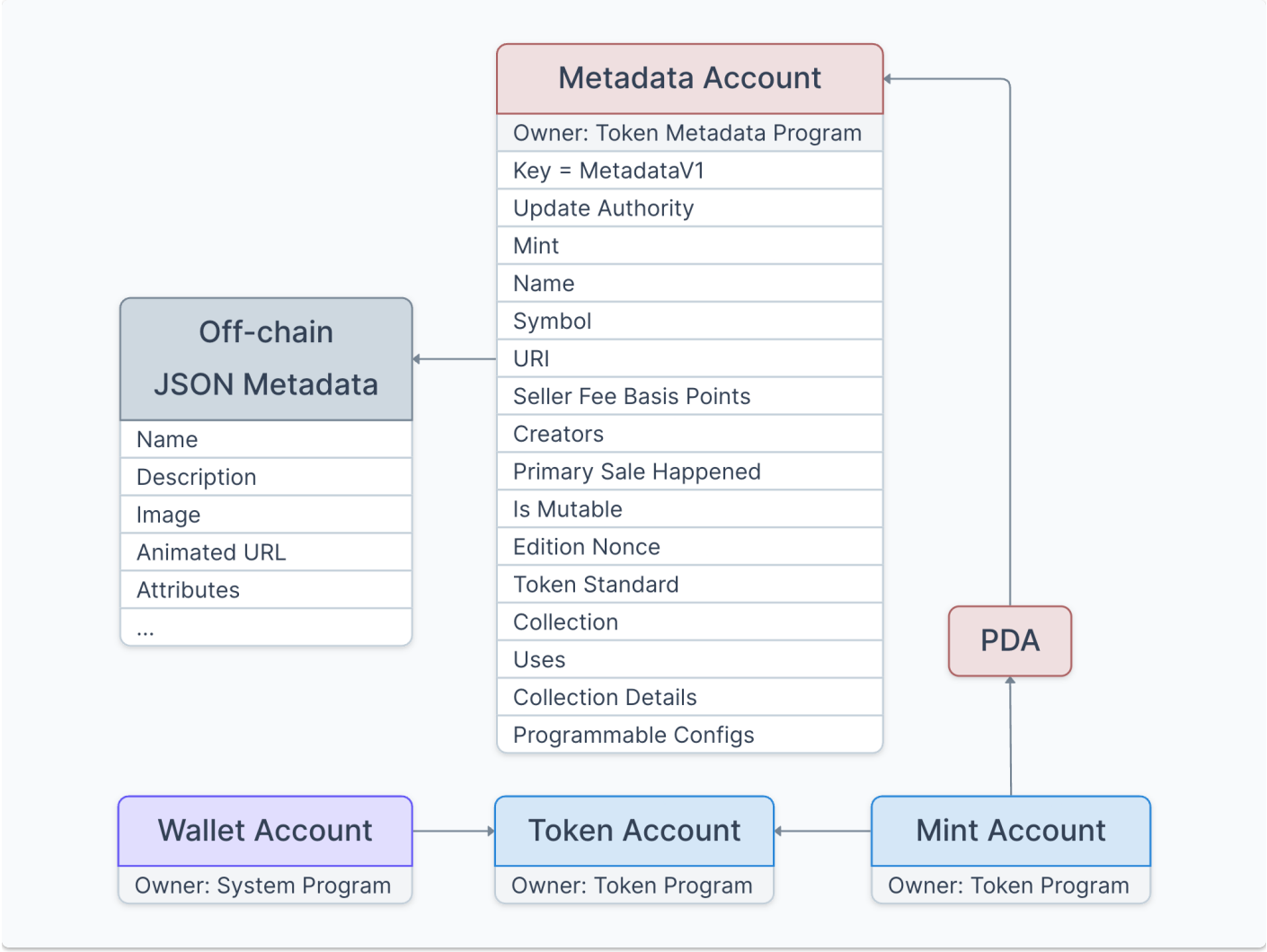
More precisely, NFTs in Solana are Mint Accounts with the following characteristics:

- It has a supply of 1, meaning only one token is in circulation.
- It has zero decimals, meaning there cannot be such a thing as 0.5 tokens.
- It has no mint authority, meaning no one can ever mint additional tokens.



Token Metadata




Metadata Account
Owner: Token Metadata Program
Key = MetadataV1
Update Authority
Mint
Name
Symbol
URI
Seller Fee Basis Points
Creators
Primary Sale Happened
Is Mutable
Edition Nonce
Token Standard
Collection
Uses
Collection Details
Programmable Configs
Off-chain JSON Metadata
Name
Description
Image
Animated URL
Attributes
...
PDA
Wallet Account
Owner: System Program
Token Account
Owner: Token Program
Mint Account
Owner: Token Program

Metaplex overview

Metaplex is a framework for NFT launches, sales, auctions and partitioning that includes on-chain programs as well as client libraries. Some of the programs are not stable while others do not have a ready client.

Metaplex has 3 main projects

- Token Metadata - the NFT standard for Solana
- Candy Machine v3™ - a Profile Picture (PFP) focused tool that works like the gumball-style candy machines of old
- Auction House - a decentralized sales protocol for NFT marketplaces

Metaplex NFTs off-chain data can be stored on either Arweave, IPFS or AWS and multiple file formats are supported.
Wallets and websites that support Metaplex will be able to display all or some of these formats.

File Type	Extension
Audio	.mp3, .flac, .wav
Img	.png, .jpg, .gif
Movie	.mp4, .mov, .webm
VR	.gib
HTML	.html

Collections can be derived from existing collections.

Reasons to use Metaplex

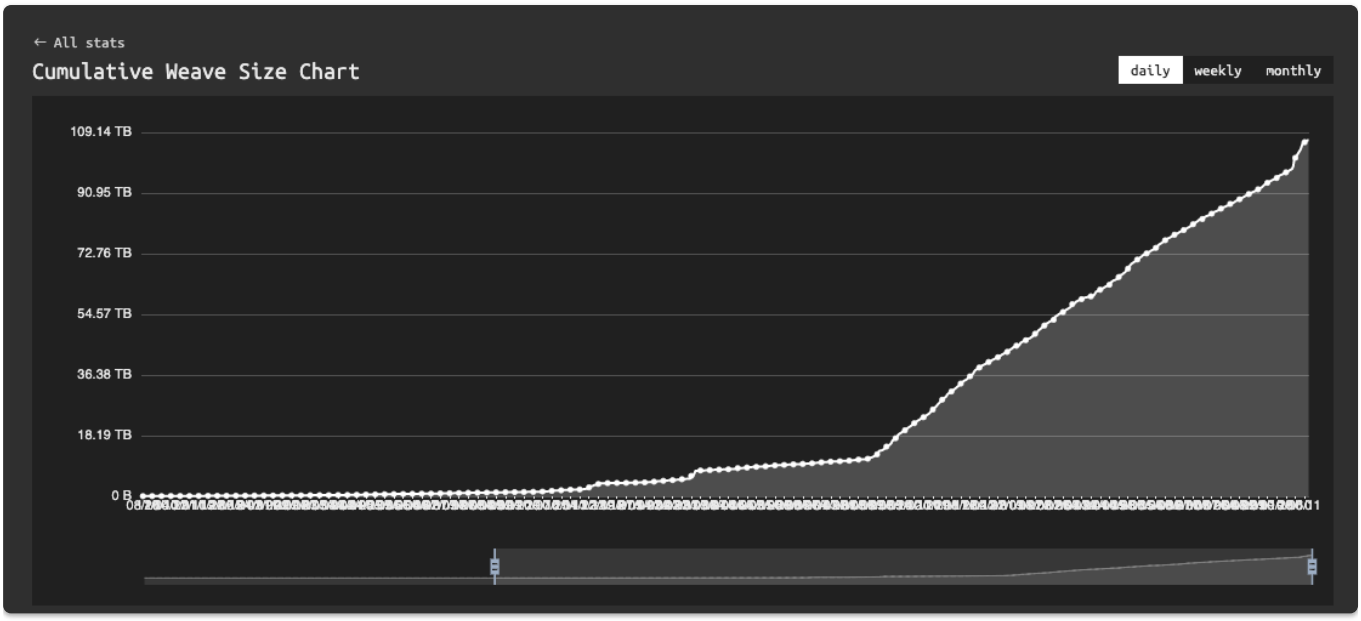
A creator that is agnostic to the platform that will host NFTs, may consider Metaplex/Solana due to cheap gas costs.
The price of a single NFT mint varies due to the size of off-chain data and SOL price but will usually be under $3.

Arweave

Features

- Decentralised storage built on top of the Arweave protocol
- Permanent and resilient
- Proof of Access mechanism (similar to Filecoin's Proof of Space and Proof of Replication)
- Incentivises quick retrieval
- Data replicated and distributed across the network

Currently adding approx 6 TB per month



Storage details

- Mechanism to distribute tokens to sustainably incentivise perpetual storage or arbitrary quantities of data.
- Based on a logarithmically decreasing $/GB-h.
- Storage media assumed to take 434 yrs to reach max theoretical data density limit of 1.53x10^67 bits/cm^3 (currently 1.66x10^12)
- Storage medium reliability increases.
- Incentive for cheap storage grows as humanity's demand for data is growing.
- Predictions for future safekeeping of data based on ultra-low cost of storage medium.
- Arweave expected to be 'nested' inside future storage systems.

Chain Architecture

- Block constructed from previous block, recall block and transactions
- Recall bloc
  - A deterministic but unpredictable choice of block from the "weave's" history
  - Prevents data from being lost if it is rarely accessed
  - Lesser known blocks increase miner fees

Metaplex architecture

Metaplex

A top level manager that knows the existence of other programs and interacts with them via CPI to accomplish necessary logic.

Token Metadata

This program allows decoration of SPL tokens with PDAs that store additional on-chain data and in them a pointer to the off-chain data.

Token data is split between:

- on-chain data
- off-chain data

On-chain data holds parameters such as:

- mint authority
- creator
- royalty splits
- link to off-chain metadata

The Non Fungible Standard from Metaplex.

See Docs

Off-chain data stores a pair of media file and associated JSON per NFT.

This JSON has anything that describes the individual parameters of this NFT.

Standard

Field	Type	Description
name	string	Name of the asset.
symbol	string	Symbol of the asset.
description	string	Description of the asset.
image	string	URI pointing to the asset's logo.
animation_url	string	URI pointing to the asset's animation.
external_url	string	URI pointing to an external URL defining the asset — e.g. the game's main site.
attributes	array	Array of attributes defining the characteristics of the asset.
trait_type	string	The type of attribute.
value	string	The value for that attribute.

For example

```
{
  "name": "SolanaArtProject #1",
  "description": "Generative art on
Solana.",
  "image":
"https://arweave.net/26YdhY_eAzv26YdhY1uu9
uiA3nmDZYwP8MwZAultcE?ext=jpeg",
  "animation_url":
"https://arweave.net/ZAultcE_eAzv26YdhY1uu
9uiA3nmDZYwP8MwuiA3nm?ext=glb",
  "external_url": "https://example.com",
```

```
  "attributes": [
    {
      "trait_type": "trait1",
      "value": "value1"
    },
    {
      "trait_type": "trait2",
      "value": "value2"
    }
  ],
  "properties": {
    "files": [
      {
        "uri":
"https://www.arweave.net/abcd5678?
ext=png",
        "type": "image/png"
      },
      {
        "uri":
"https://watch.videodelivery.net/9876jkl",
        "type": "unknown",
        "cdn": true
      },
```

```
      {
        "uri":
"https://www.arweave.net/efgh1234?
ext=mp4",
        "type": "video/mp4"
      }
    ],
    "category": "video",

    // @deprecated
    // Do not use - may be removed in a
future release.
    // Use on-chain data instead.
    "collection": {
      "name": "Solflare X NFT",
      "family": "Solflare"
    },

    // @deprecated
    // Do not use - may be removed in a
future release.
    // Use on-chain data instead.
    "creators": [
      {
```

```
        "address":
"xEtQ9Fpv62qdc1GYfpNReMasVTe9YW5bHJwfVKqo72u",
        "share": 100
      }
    ]
  }
}
```

Auction

This program supports the running of an auction, currently, only English auctions are supported. It has mechanisms for collecting bids (that can be any SPL token), cancelling bids, stating the winner and withdrawing funds.

The Auction PDA specifies parameters such as start and end time, authority or minimum bid price.

```
#[repr(C)]
#[derive(Clone, BorshSerialize,
BorshDeserialize, PartialEq, Debug)]
pub struct AuctionData {
        pub authority: Pubkey,
        pub token_mint: Pubkey,
        pub last_bid:
Option<UnixTimestamp>,
        pub ended_at:
Option<UnixTimestamp>,
        pub end_auction_gap:
Option<UnixTimestamp>,
        pub price_floor: PriceFloor,
```

```
    pub bid_state: BidState,
}
```

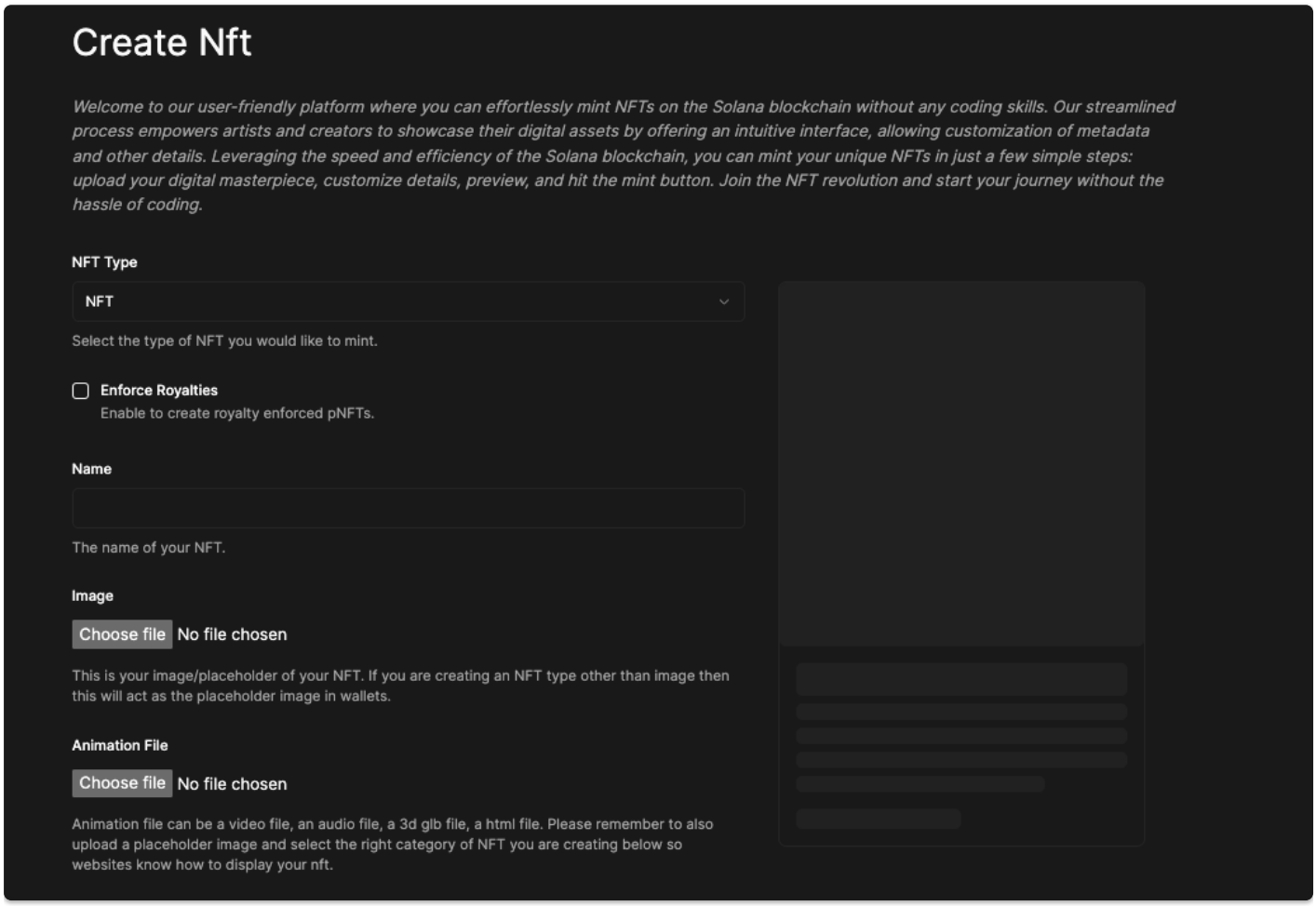
Token Vault

This program holds tokens being auctioned by the Auction program and allows for fractionalisation of an NFT. Partitioned NFT with a share like structure could be used so that multiple entities can own a single expensive asset.

Further tools

[sol tools]

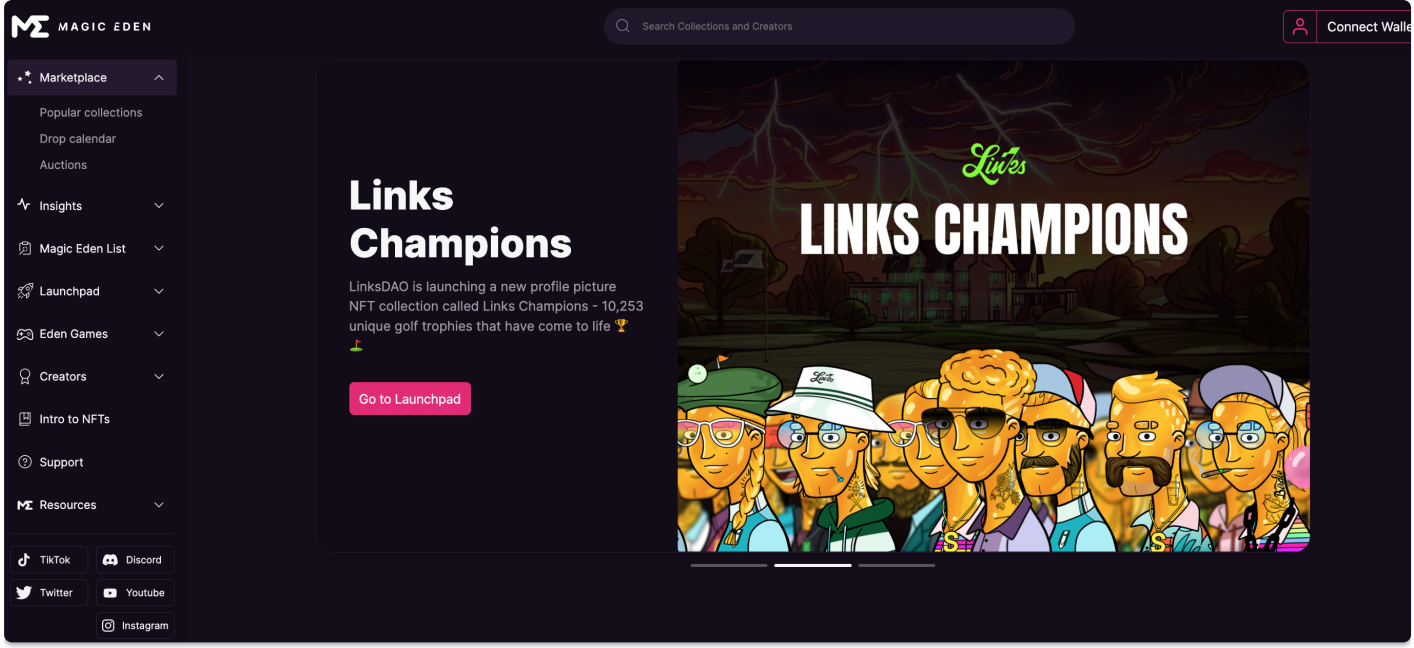
This site helps ou to create for example NFTs



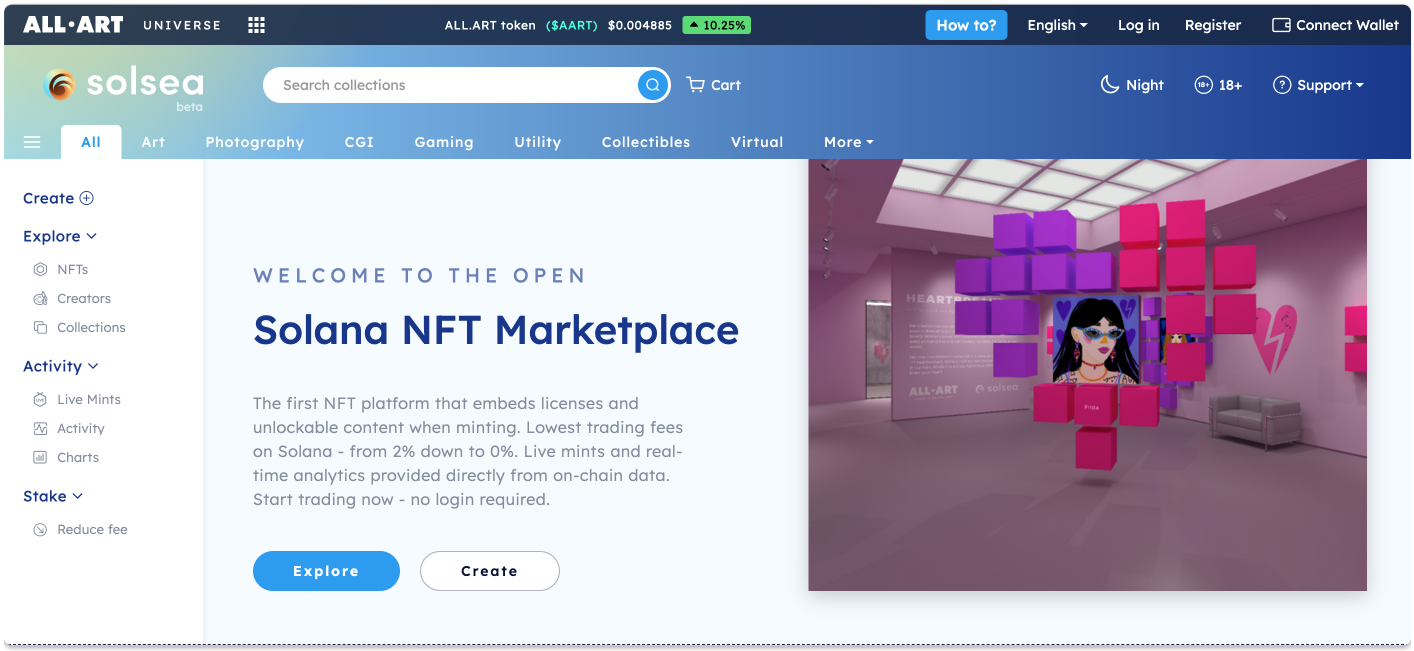
(NFT) Marketplaces

- Digital assets
- Can be fractionalised
- Can be used as collateral

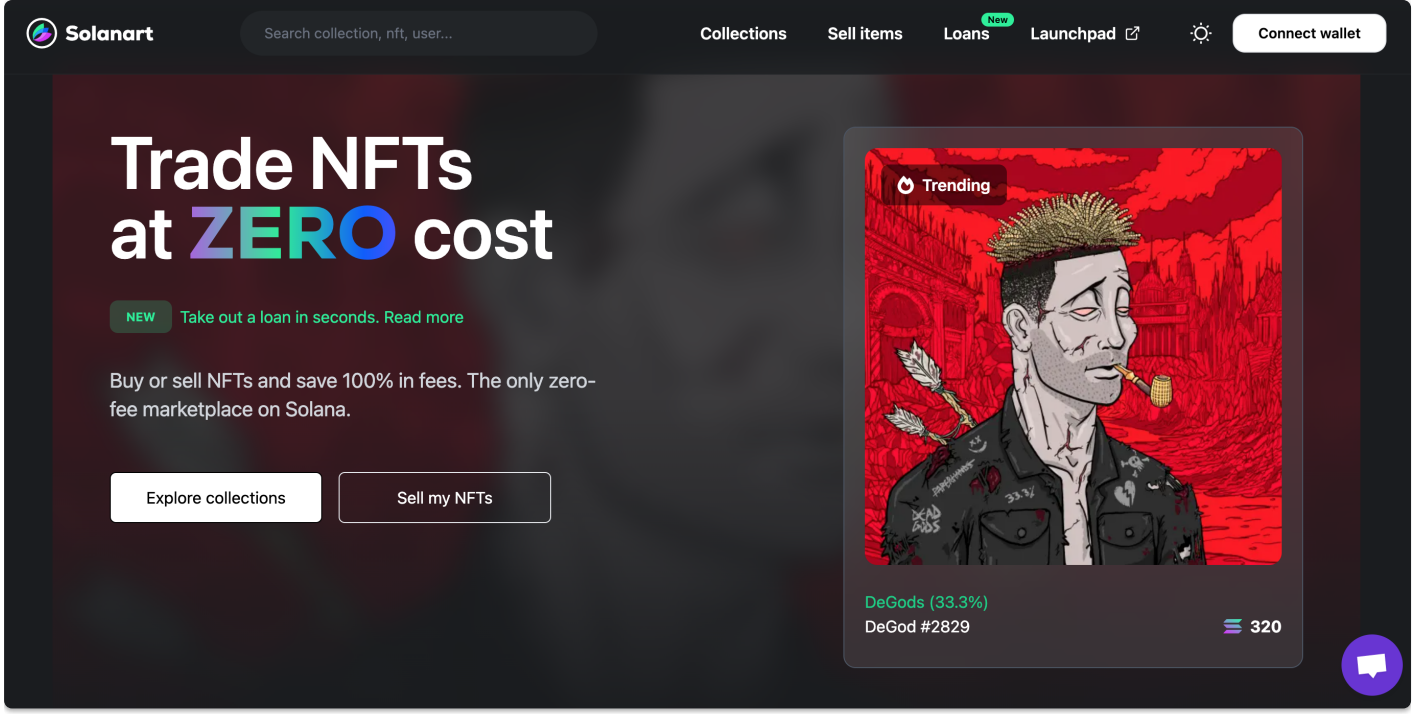
Magic Eden



Solsea



Solanart


Solanart
Search collection, nft, user...
Collections
Sell items
Loans
New
Launchpad
Connect wallet
Trade NFTs
at ZERO cost
NEW
Take out a loan in seconds. Read more
Buy or sell NFTs and save 100% in fees. The only zero-fee marketplace on Solana.
Explore collections
Sell my NFTs
Trending
DeGods (33.3%)
DeGod #2829
320

Candy machine

Candy machine has recently been upgraded.
See Docs
Features

- Accept payments in SOL, NFTs or any Solana token.
- Restrict your launch via start/end dates, mint limits, third party signers, etc.
- Protect your launch against bots via configurable bot taxes and gatekeepers like Captchas.
- Restrict minting to specific NFT/Token holders or to a curated list of wallets.
- Create multiple minting groups with different sets of rules.
- Reveal your NFTs after the launch whilst allowing your users to verify that information.

Candy Machine process flow

1. Create the machine and set up configuration.
2. Add the items we are minting
3. Mint the NFTs

4. Clean up

Candy Guards

There is the ability to add extra features to your candy machine by the use of 'guards'
Example guards

- **Sol Payment**: This guard ensures the minting wallet has to pay a configured amount of SOL to a configured destination wallet.
- **Start Date**: This guard ensures minting can only start after the configured time.
- **Mint Limit**: This guard ensures each wallet cannot mint more than a configured amount.

Sugar

Sugar is a CLI to be used with metaplex, and a replacement for the javascript CLI.

Installation

Mac / Linux / WSL

```
bash <(curl -sSf
https://sugar.metaplex.com/install.sh)
```

```
✅ Installation successful: type 'sugar' to start using it.
gitpod /workspace/SolanaBootcampOctober (main) $ sugar
sugar-cli 1.1.0
Command line tool for creating and managing Metaplex Candy Machines.

USAGE:
    sugar [OPTIONS] <SUBCOMMAND>

OPTIONS:
    -h, --help                    Print help information
    -l, --log-level <LOG_LEVEL>   Log level: trace, debug, info, warn, error, off
    -V, --version                 Print version information

SUBCOMMANDS:
    bundlr            Interact with the bundlr network
    collection        Manage the collection on the candy machine
    create-config     Interactive process to create the config file
    deploy            Deploy cache items into candy machine config on-chain
    freeze            Commands for the Candy Machine Freeze feature
    hash              Generate hash of cache file for hidden settings
    help              Print this message or the help of the given subcommand(s)
    launch            Create a candy machine deployment from assets
    mint              Mint one NFT from candy machine
    reveal            Reveal the NFTs from a hidden settings candy machine
    show              Show the on-chain config of an existing candy machine
    sign              Sign one or all NFTs from candy machine
    thaw              Thaw a NFT or all NFTs in a candy machine
    unfreeze-funds    Unlock treasury funds after freeze is turned off or expires
    update            Update the candy machine config on-chain
    upload            Upload assets to storage and creates the cache config
    validate          Validate JSON metadata files
    verify            Verify uploaded data
    withdraw          Withdraw funds from candy machine account closing it
gitpod /workspace/SolanaBootcampOctober (main) $ sugar -V
sugar-cli 1.1.0
gitpod /workspace/SolanaBootcampOctober (main) $ ›
```

Sugar will then use these settings by default if you don't specify them as CLI options, allowing commands to be much simpler.

Prepare Metaplex

Install the solana CLI if not already installed

Generate a new keypair

```
solana-keygen new --outfile ~/.config/solana/devnet.json
```

Set the config

```
solana config set --keypair
~/.config/solana/devnet.json
```

```
solana config set --url
https://metaplex.devnet.rpcpool.com/
```

Check the config

```
solana config get
```

Sugar Quickstart

1. Create an assets directory
   `mkdir assets`
2. In this directory add the assets and asset json files. You can find an example in the repo

```
{
    "name": "Studious Crab #1",
    "symbol": "CRAB",
    "description": "The Studious Crabs are smart and productive crabs.",
    "image": "0.png",
    "attributes": [
        {
            "trait_type": "accessory",
            "value": "lamp"
        },
        {
            "trait_type": "chair",
            "value": "red"
        },
        {
            "trait_type": "books",
```

```
            "value": "blue"
        }
    ],
    "properties": {
        "files": [
            {
                "uri": "0.png",
                "type": "image/png"
            }
        ]
    }
}
```

From the project directory run

`sugar create-config`

This will take you through a number of choices, you can read more about the choices here
At the end it will save a config file which will look like

```
{
  "price": 1.0,
  "number": 10,
  "gatekeeper": null,
```

```
  "creators": [
    {
      "address":
"PanbgtcTiZ2PveV96t2FHSffiLHXXjMuhvoabUUKK
m8",
      "share": 100
    }
  ],
  "solTreasuryAccount":
"PanbgtcTiZ2PveV96t2FHSffiLHXXjMuhvoabUUKK
m8",
  "splTokenAccount": null,
  "splToken": null,
  "goLiveDate": "11 Aug 2022 18:19:16
+0000",
  "endSettings": null,
  "whitelistMintSettings": null,
  "hiddenSettings": null,
  "freezeTime": null,
  "uploadMethod": "bundlr",
  "retainAuthority": true,
  "isMutable": true,
  "symbol": "TEST",
  "sellerFeeBasisPoints": 500,
```

```
  "awsS3Bucket": null,
  "nftStorageAuthToken": null,
  "shdwStorageAccount": null
}
```

Once the config is saved you can upload the files

`sugar upload`

You can then create and deploy a candy machine with

`sugar deploy`

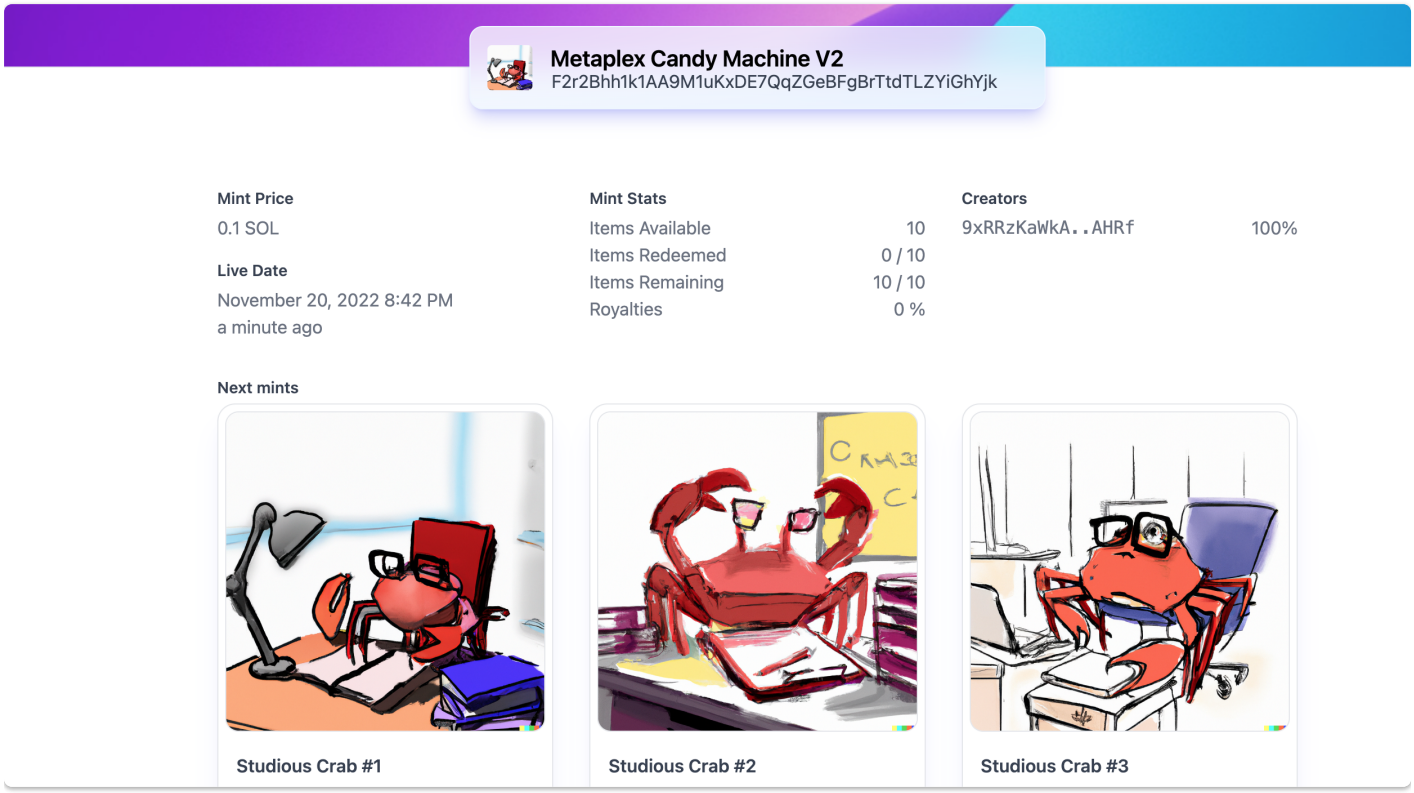
Verify that it has all worked with

`sugar verify`

you can get the details of the candy machine with

`sugar show`

You will get a URL to show your candy machine



Minting NFTs

Now that we have our candy machine deployed we can use it to mint NFTs

```
sugar mint
```

will mint an NFT to your wallet.

Creating a UI for your Candy Machine

Clone the directory into your project folder

`git clone https://github.com/metaplex-foundation/candy-machine-ui`

Get your Candy Machine ID, either from

`sugar show`

or from the cache.json file

Add the ID to the `.env.example` file and save the file as

`.env`

From within the candy-machine-ui directory run

`yarn install && yarn start`

A Javascript SDK is available, Swift and Kotlin SDKs are being planned.

Rust crates are also available

Using the javascript SDK

Setup

You need to setup a connection

```
import { Metaplex } from "@metaplex-
foundation/js";
import { Connection, clusterApiUrl } from
"@solana/web3.js";

const connection = new
Connection(clusterApiUrl("mainnet-beta"));
```

You can then create a Metaplex instance

```
const wallet = Keypair.generate();

const metaplex = Metaplex.make(connection)
    .use(keypairIdentity(wallet))
    .use(bundlrStorage());
```

The metaplex instance then gives the nft module

```
const nftClient = metaplex.nfts();
```

which has useful methods

- `findByMint`
- `findAllByMintList`
- `load`
- `findAllByOwner`
- `findAllByCreator`
- `uploadMetadata`
- `create`
- `update`
- `printNewEdition`
- `use`

See the full API here

Example for uploading metadata and creating an NFT.
The ownership and authority information would be taken from the user set up earlier.

```
const { uri } = await metaplex
  .nfts()
  .uploadMetadata({
    name: "My off-chain name",
    description: "My off-chain
description",
    image: "https://arweave.net/123",
  };

const { nft } = await metaplex
  .nfts()
  .create({
    uri,
    name: 'My on-chain NFT',
    sellerFeeBasisPoints: 250, // 2.5%
  };
```

Other useful programs

Gumdrop

See Docs

This helps with creating token and NFT airdrops.

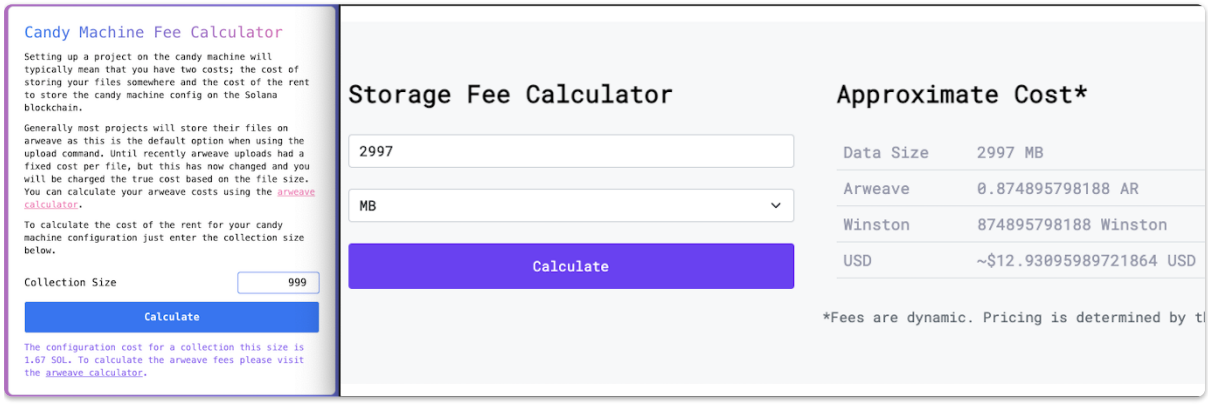
Other considerations

Costs

A collection with 999 items and average off-chain data ~3Mb will cost 1.67 SOL and 0.87 AR.

The total price tag for the deployment of this collection at today's Sol and AR valuations would be ~$30, but during the peak price of crypto same collection would cost in excess of $350.

Fair to say to anyone considering launching their own collection now is a good time.



Marketing and brand management

For an NFT launch to be successful it requires a strategy that includes selecting a target audience, brand creation, marketing or potential celebrity endorsement. Even after the launch continuous community management via a social platform such as discord is likely needed to make sure people do not lose interest especially if the creator is to receive royalties from each resale for perpetuity.

Permanence

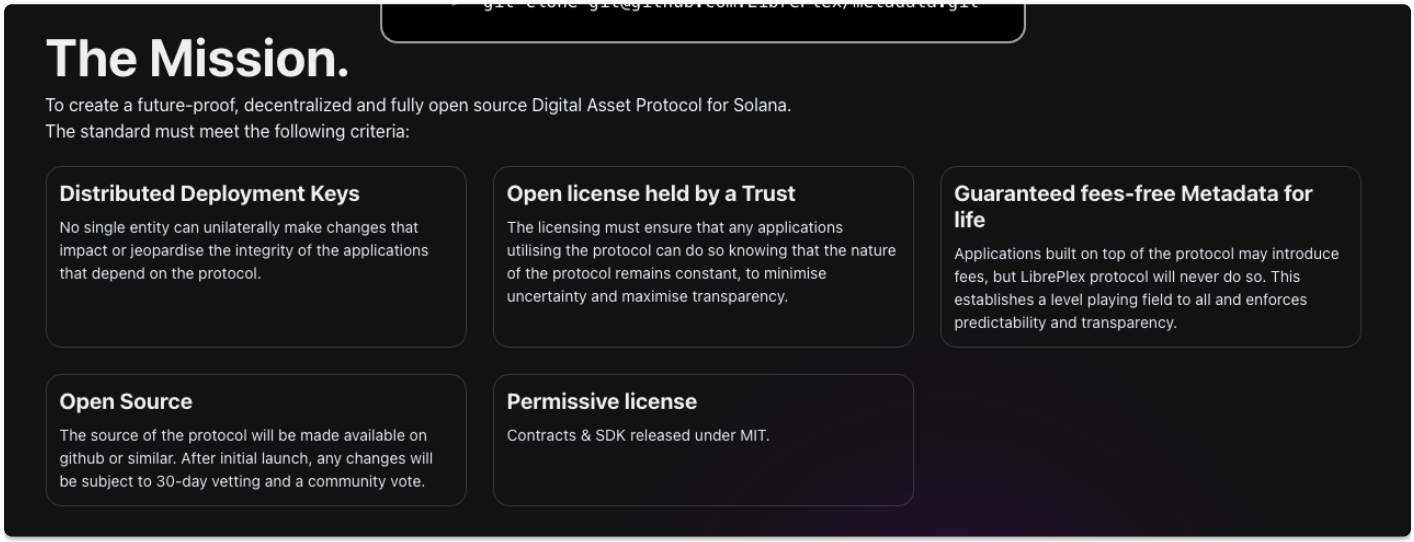
It is important to state whether NFT is fixed or whether it can be modified. Generally, NFTs are thought of as immutable and owners might object to suddenly realising their content has been changed. This can be enforced with `is_mutable` field being set to `false`.

On the flip side, should NFT be intentionally upgradable, for example, to reflect loalty points or game character attributes it would require `is_mutable` field being set to `true`. As Arweave data is immutable, new off-chain data deployment will be required which will incur some fees as well as updating of the Arweave URI.

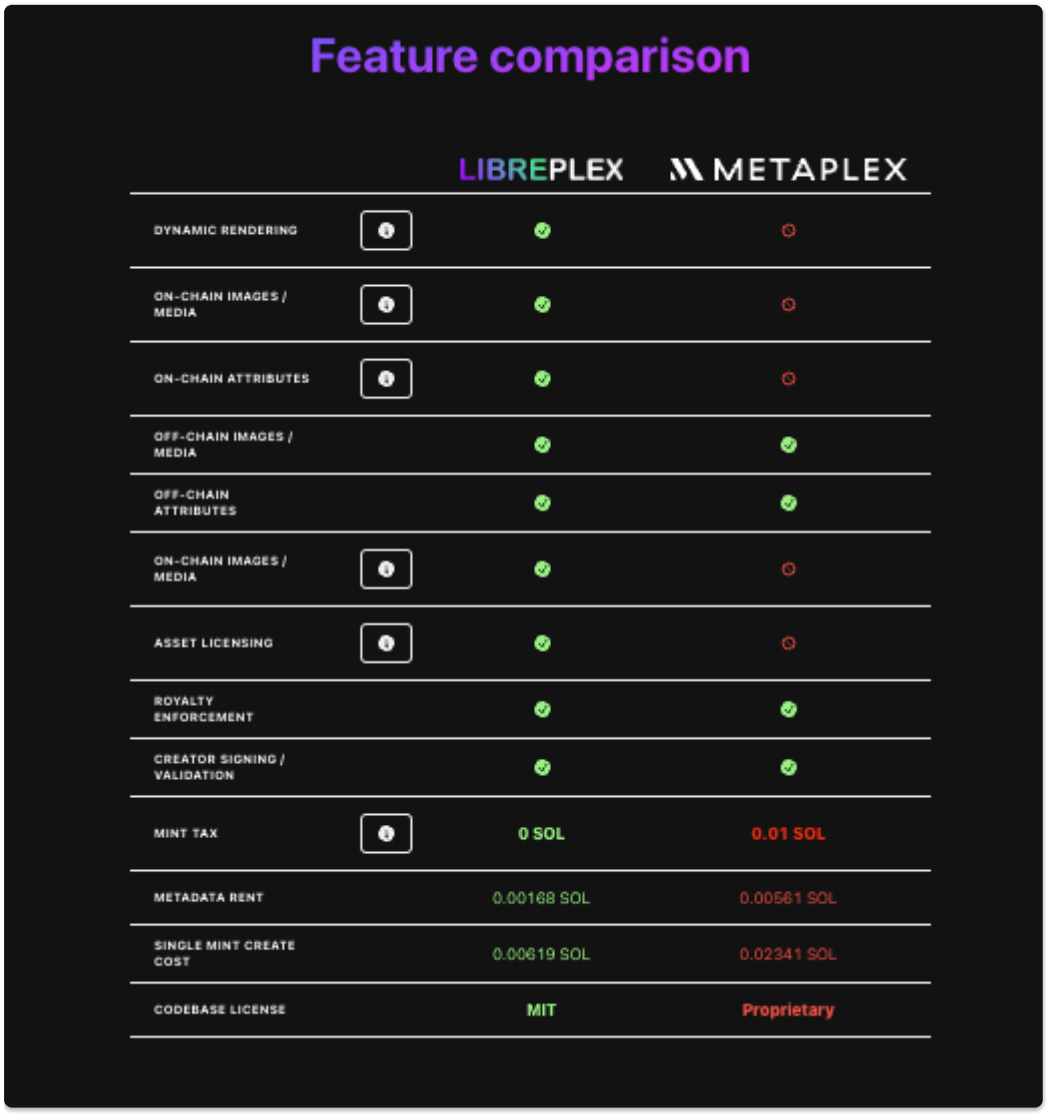
> Either approach is valid as long as it is communicated to the users.

Introduction

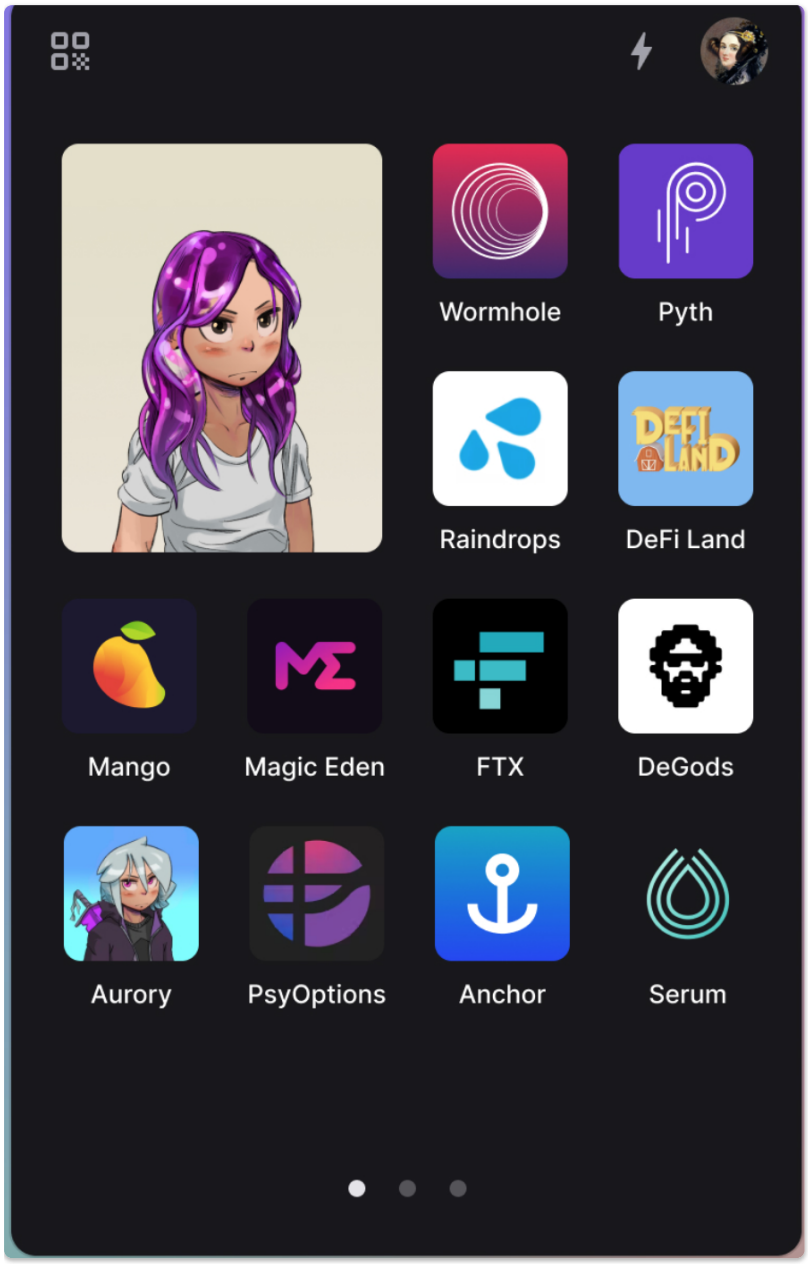
https://www.libreplex.io/



An alternative to Metaplex



xNFTs



An extension to the NFT concept are xNFTs, the 'x' stands for executable.
"tokenised code, representing ownership rights over its execution"

Coral have built a multichain wallet backpack to hold xNFTs in addition to your other assets.

The xNFT standard enables any dapp built on Solana to create an xNFT and be executed natively inside the backpack wallet.

Example use is with Mango Markets, a Backpack user can access their trading account and manage their positions on Mango Market, from the Mango Market xNFT inside the Backpack wallet.

Inscriptions Introduction

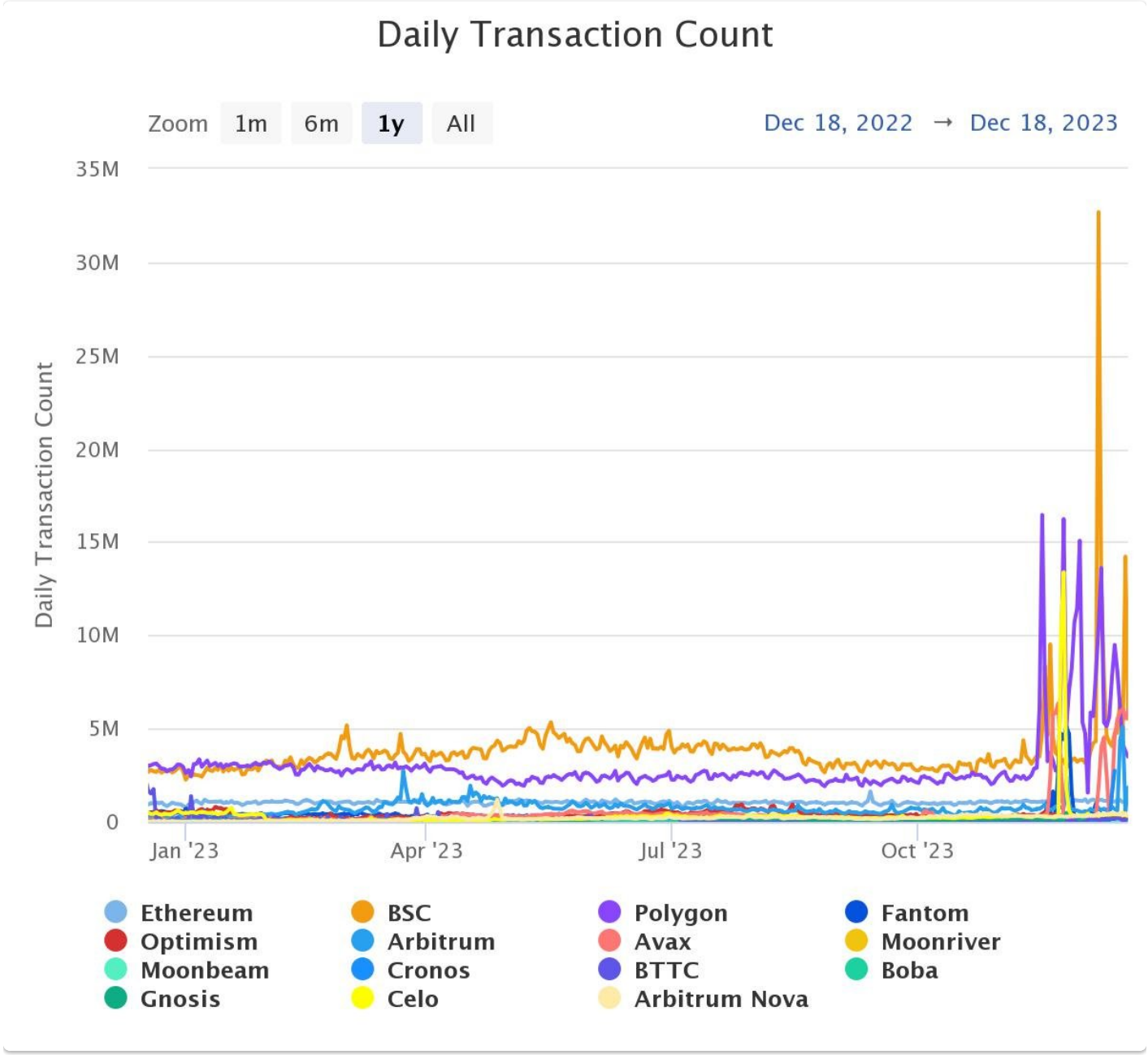
Inscriptions are a means to add digital assets on the blockchain. Traditionally the pattern used has been to store a link to a decentralised file system containing the asset.
The mechanism was first seen on Bitcoin, where assets were added to satoshis, the mechanism is slightly different on non UTXO chains, where the asset data is added to a transaction.

For example in an Ethereum transaction this looks like

Ether Price:	$2,177.38 / ETH
Gas Limit & Usage by Txn:	22,120 | 22,120 (100%)
Gas Fees:	Base: 48.135243525 Gwei | Max: 65.913521807 Gwei | Max Priority: 0.1 Gwei
Burnt & Txn Savings Fees:	Burnt: 0.001064751586773 ETH ($2.36) Txn Savings: 0.00039104351559784 ETH ($0.87)
Other Attributes:	Txn Type: 2 (EIP-1559) Nonce: 185 Position In Block: 97
Input Data:	`data:,{"p":"erc-20","op":"mint","tick":"frog","id":"597","amt":"1000"}`

The volume of inscriptions quickly became significant, here is a chart from Etherescan.

[Post]



> Where transaction data is used he mechanism relies on third parties to index the data, and apply the rules you would see in a smart contract.

Inscriptions on Solana

Inscriptions volumes on Solana have been high, on Dec 16th 2023 287,000 inscriptions were minted.

Support for inscriptions - SPL-20 standard

This was introduced on November 16th, it draws inspiration from the BRC20 standard on Bitcoin. it is subtly different to the mechanisms used on other chains.
From explantion

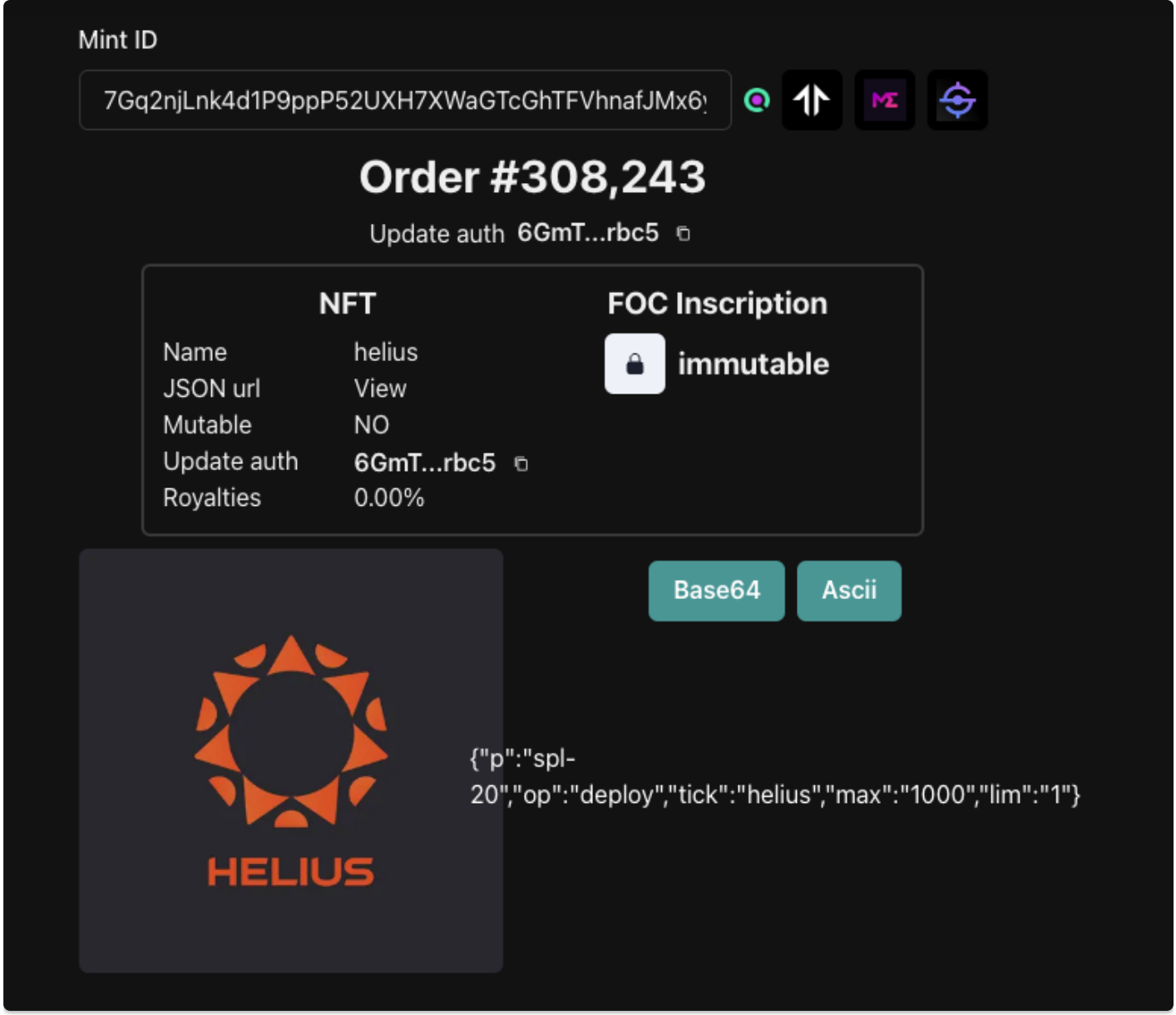
An inscription is a PDA associated with an NFT. It's similar to the token metadata account, except it stores arbitrary binary data in an array. Due to transaction size limits, the array might need to be populated over multiple transactions.

Note that you can inscribing an NFT does not make it an SPL-20 token. They are different things. SPL-20 uses Solana tokens, plus inscriptions, to mimic Bitcoin's BRC-20. It's defined via JSONs. The following JSON defines

a helius token with a supply of 1000:

Note the "op" is "deploy". For Solana folks, this is akin to your mint account. When you mint SPL-20 tokens you produce a new JSON. In this example, we mint one "helius" token:

With SPL-20, each operation you take is an NFT mint. The JSON is then inscribed into the NFT. Similar to ordinals, every single operation (mint) is associated with an order number. Here's an example for the "helius" token deployment:



Example token SOLS

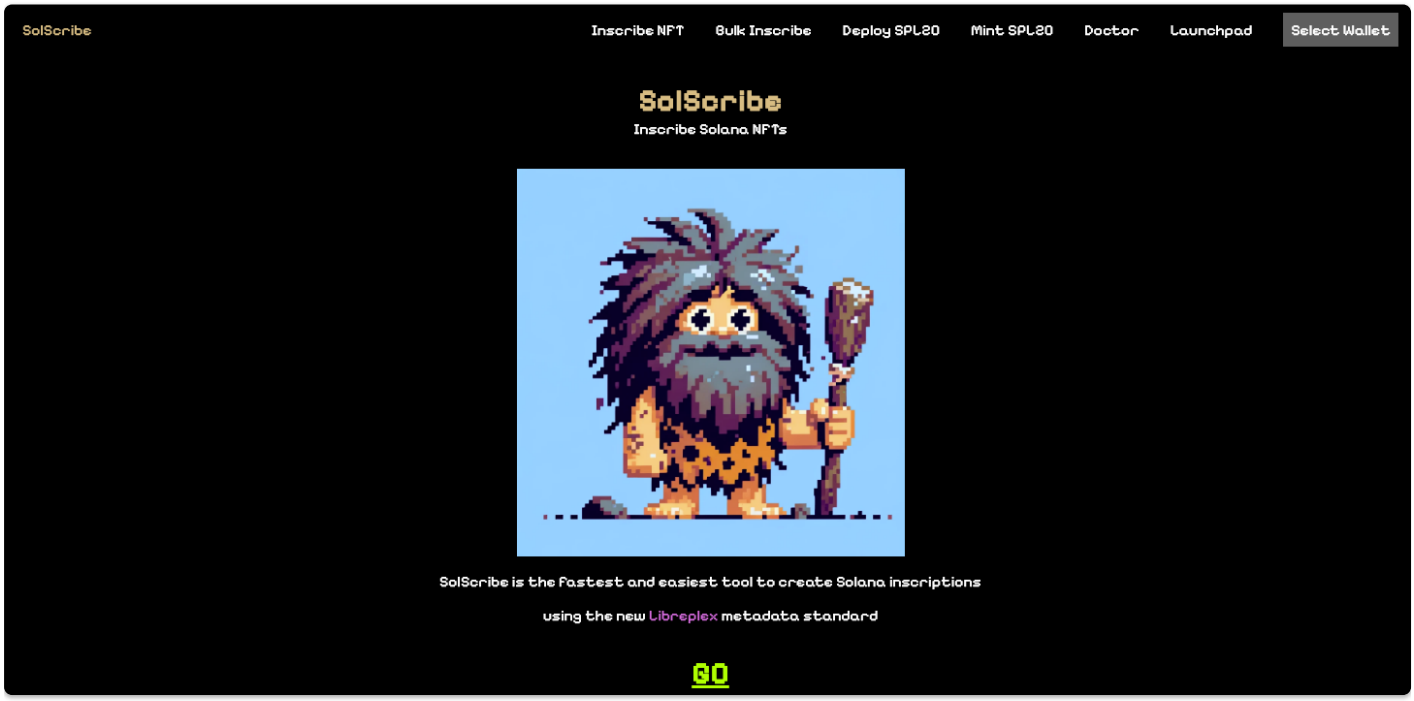
From [bitget article](#) [Sols (SOLS)](#) is a token released under the SPL-20 standard. Taking inspiration from Bitcoin's BRC-20 standard, Solana Program Library-20 (SPL-20) inscriptions are unique Solana addresses that contain images or metadata stored directly on the [blockchain](#). This token standard streamlines token creation and functionality within the network. SPL-20

enhances interoperability, simplifying token development and fostering a robust DeFi ecosystem.

Tools

Solscribe

See Solscribe



Using Metaplex

See MPL Inscriptions

See Repo

Metaplex inscriptions have two operating modes:

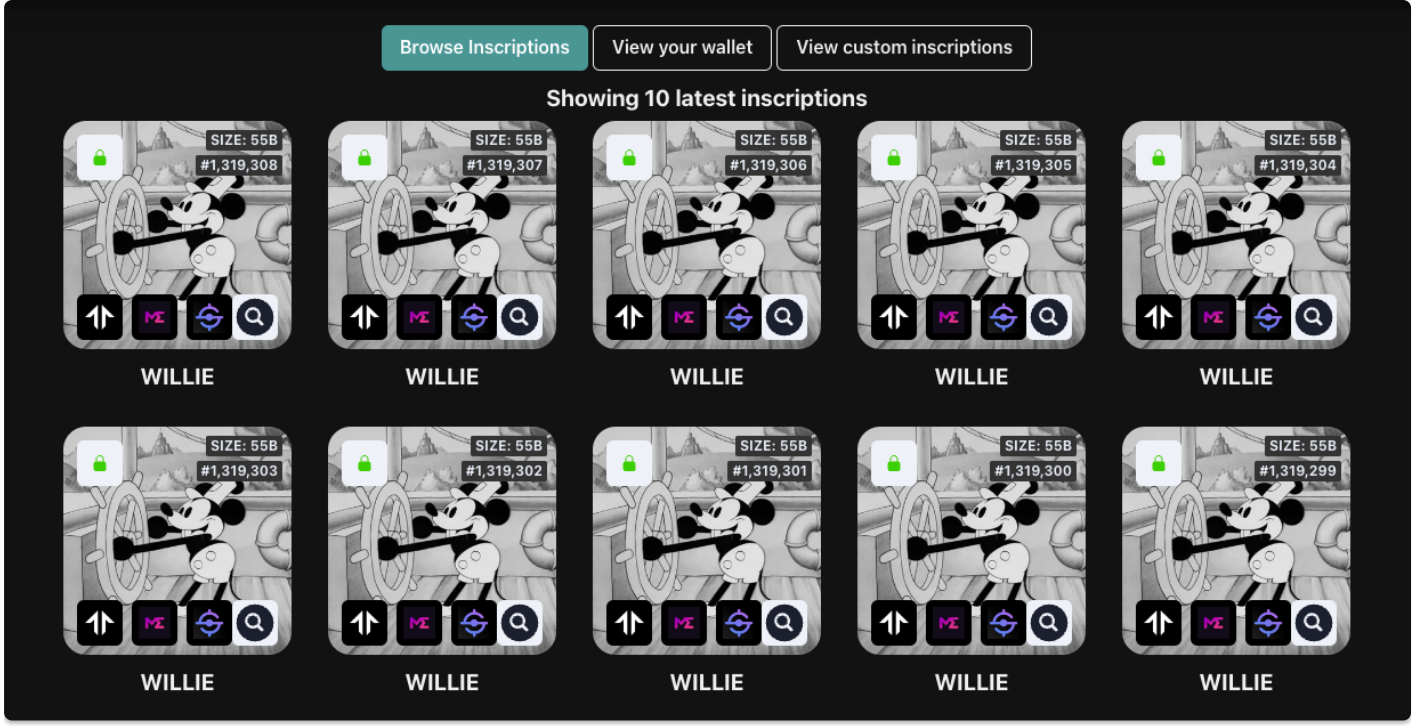
- Metadata inscribing
- Direct data storage

In both modes binary data of any format can be directly written to the chain through the Metaplex Inscription program. In addition, the Metaplex SDKs provide direct support for inscribing schemas commonly used for NFTs (i.e. JSON and Image formats).

The Metadata inscribing method creates a PDA attached to a mint account, the same way Metadata is attached to a token mint. The JSON and image data of the NFT can then be written directly to the chain in the PDA. This method provides a backup of the NFT data in the event that current data storage providers should ever go down and means the asset is “fully on Solana.”

Using LibrePlex

See Site



Steps to inscribe

1. Mint your NFT using a preferred tool.
2. Head over to [LibrePlex] and select "Inscriptions."
3. Click "View your wallet".
4. Browse and select the NFT you want to inscribe. You need to hold the update authority for the NFT you wish to inscribe.
5. Follow the three-step process: Initialize, Upload (compressed image in webp format is recommended), Resize, and finally, Inscribe.
6. You have the option to make the NFT immutable, find the details magnifying

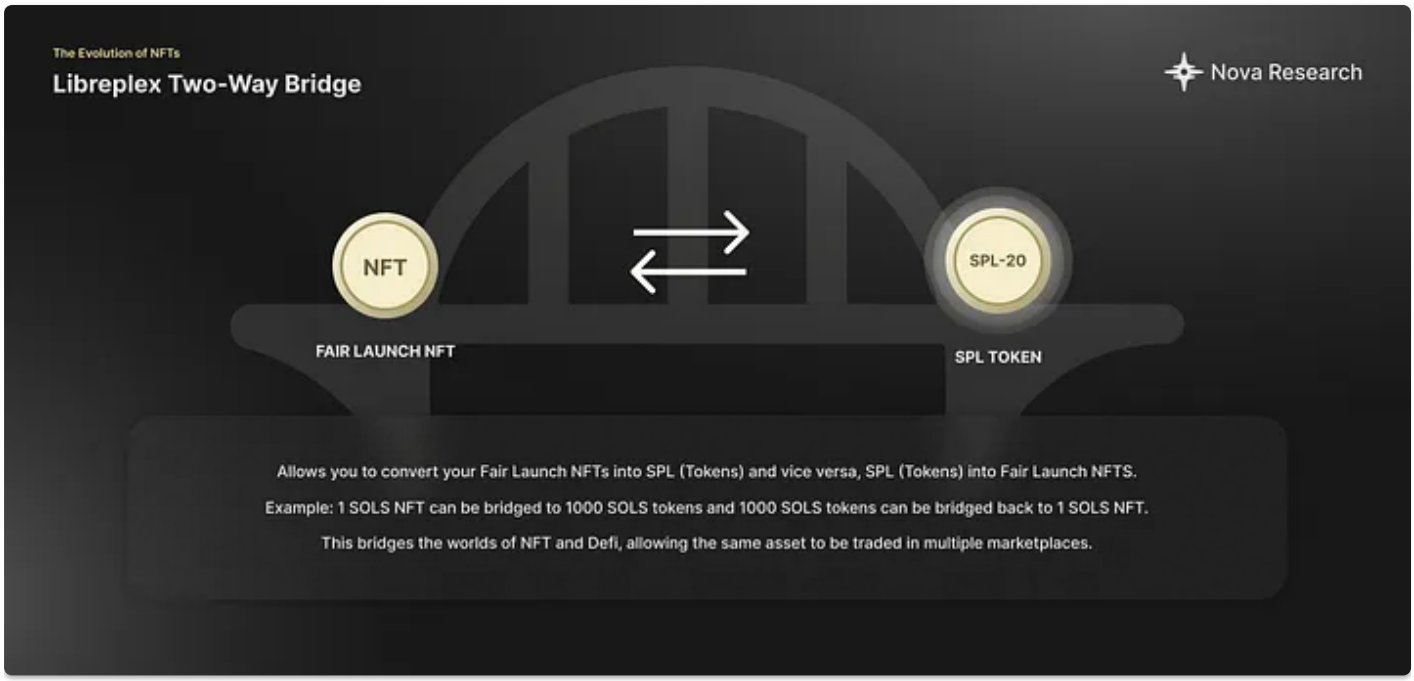
glass opening the individual inscription and select the 'Make Immutable' button.

Costs

Size	Estimated cost
10kb	0.07 SOL
50kb	0.35 SOL
100kb	0.7 SOL
150kb	1.05 SOL
200kb	1.4 SOL
1mb	7 SOL
10mb	70 SOL

2 way bridge

From Nova article



This two-way bridge allows for the conversion of SPL20 coins into regular SPL tokens and vice

versa.

Side effects

From article

An analysis of Token Terminal data reveals a 77% increase in monthly fees on the Solana network for December.

With Solana burning 50% of its fee revenue, this surge in fees could potentially exert deflationary pressure on its native token, SOL.

The token has recently seen a 23.18% growth over 30 days. As network activity directly influences SOL's value, stakeholders are likely monitoring inscription growth closely.

Solana resources

Magic Eden Article

Nova Research article

DeFi on Solana

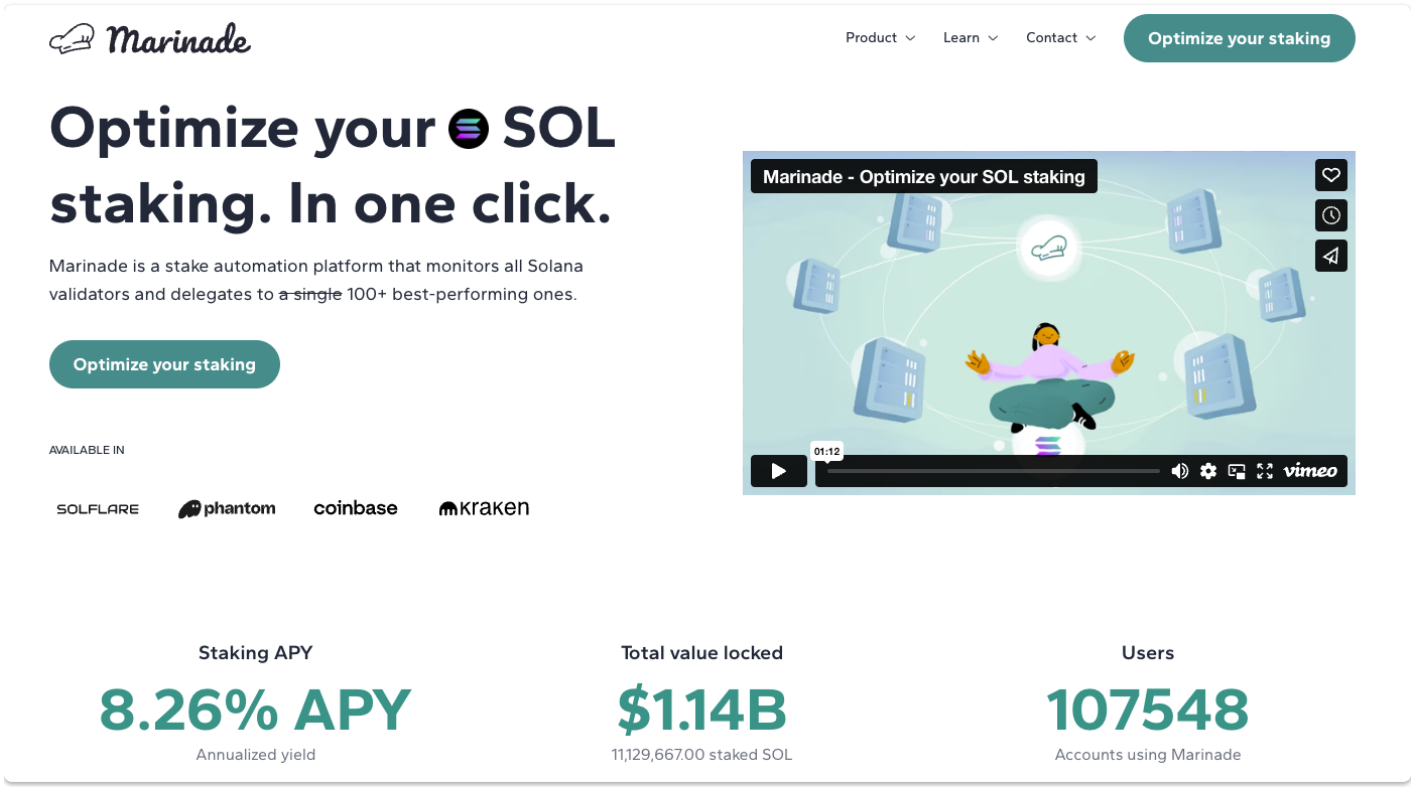
See [Ecosystem]

From DefiLLama

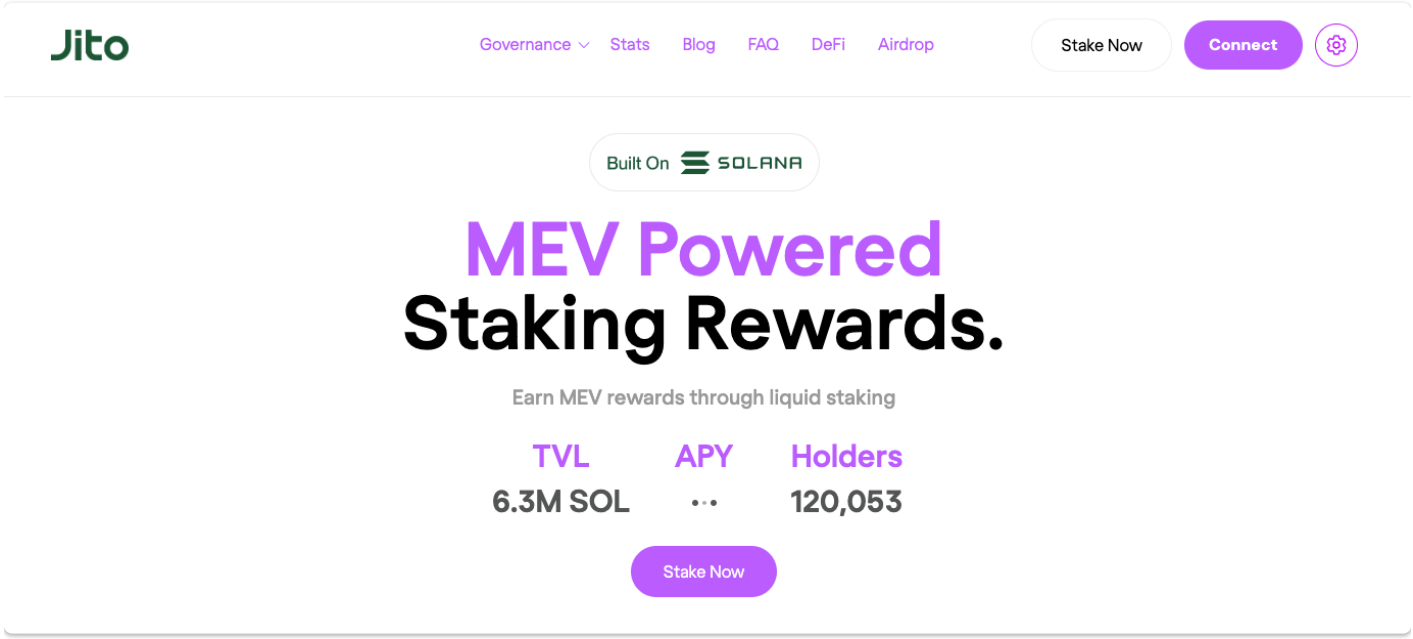
					TVL
Name	Category	TVL	1d Change	7d Change	1m Change
1 Marinade Finance 1 chain		$1.133b	-6.89%	+21.58%	+97.95%
2 Jito 1 chain	Liquid Staking	$661.8m	-7.07%	+17.02%	+67.83%
3 marginfi 1 chain		$377.7m	-6.61%	+15.25%	+161%
4 Kamino 1 chain		$226.62m	-3.67%	+19.93%	+571%
5 BlazeStake 1 chain	Liquid Staking	$206.42m	-6.01%	+25.44%	+420%
6 Solend 1 chain	Lending	$201.39m	-5.09%	+19.56%	+50.32%
7 Orca 1 chain	Dexes	$191.51m	-3.26%	+6.44%	+129%
8 Raydium 1 chain	Dexes	$122.4m	-6.55%	+21.25%	+110%
9 Drift 1 chain	Derivatives	$118.23m	-4.94%	+25.57%	+186%
10 JPool 1 chain	Liquid Staking	$68.14m	-7.13%	+23.35%	+78.49%
11 Meteora 1 chain		$48.85m	-6.17%	+15.05%	+155%
12 01 1 chain	Derivatives	$36.34m	-4.08%	+19.06%	+56.93%
13 Lido 5 chains	Liquid Staking	$34.05m	-7.98%	-53.28%	-39.79%
14 SPL Governance 1 chain	Services	$32.98m	-0.24%	+0.50%	+1.13%
15 Francium 1 chain	Yield	$21.16m	-4.61%	+8.58%	+61.52%

Major Projects

Marinade Finance

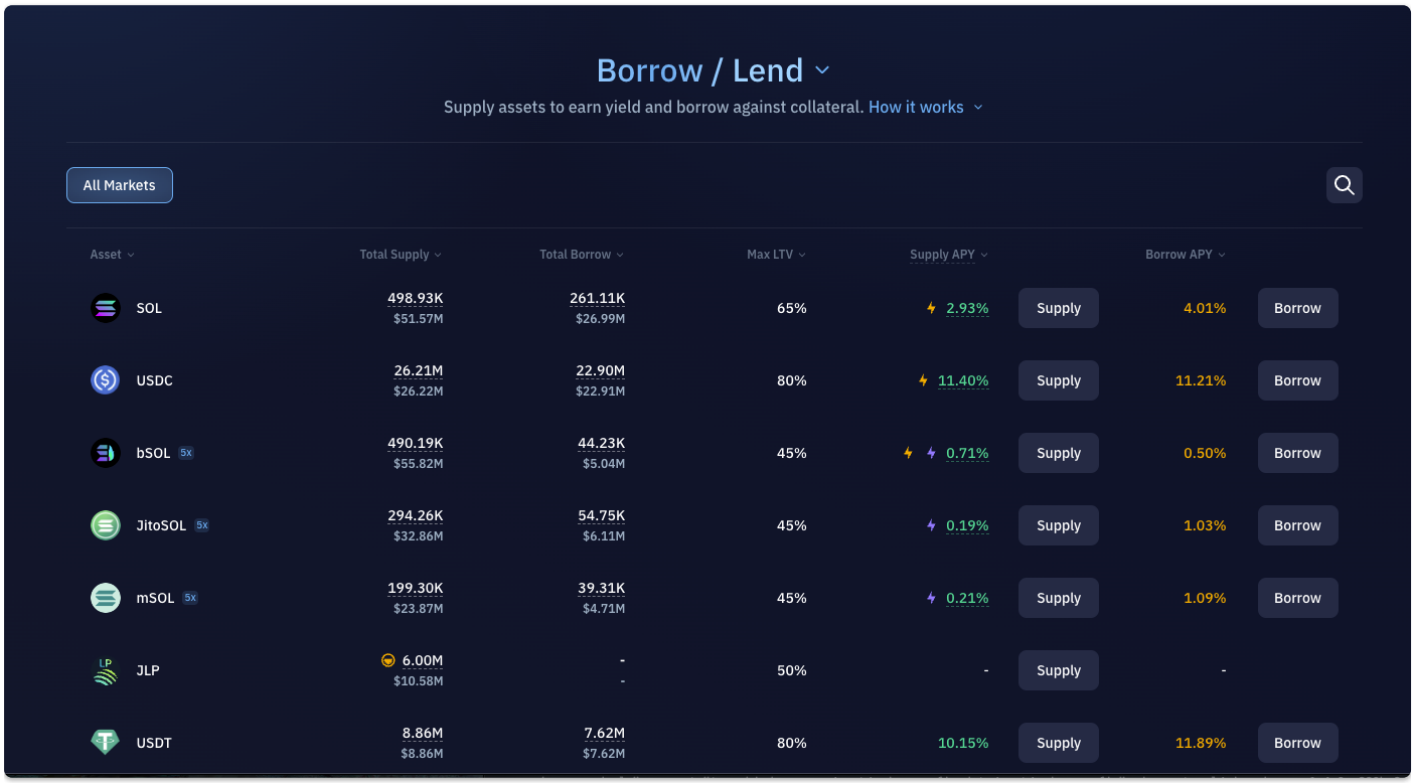


Jito

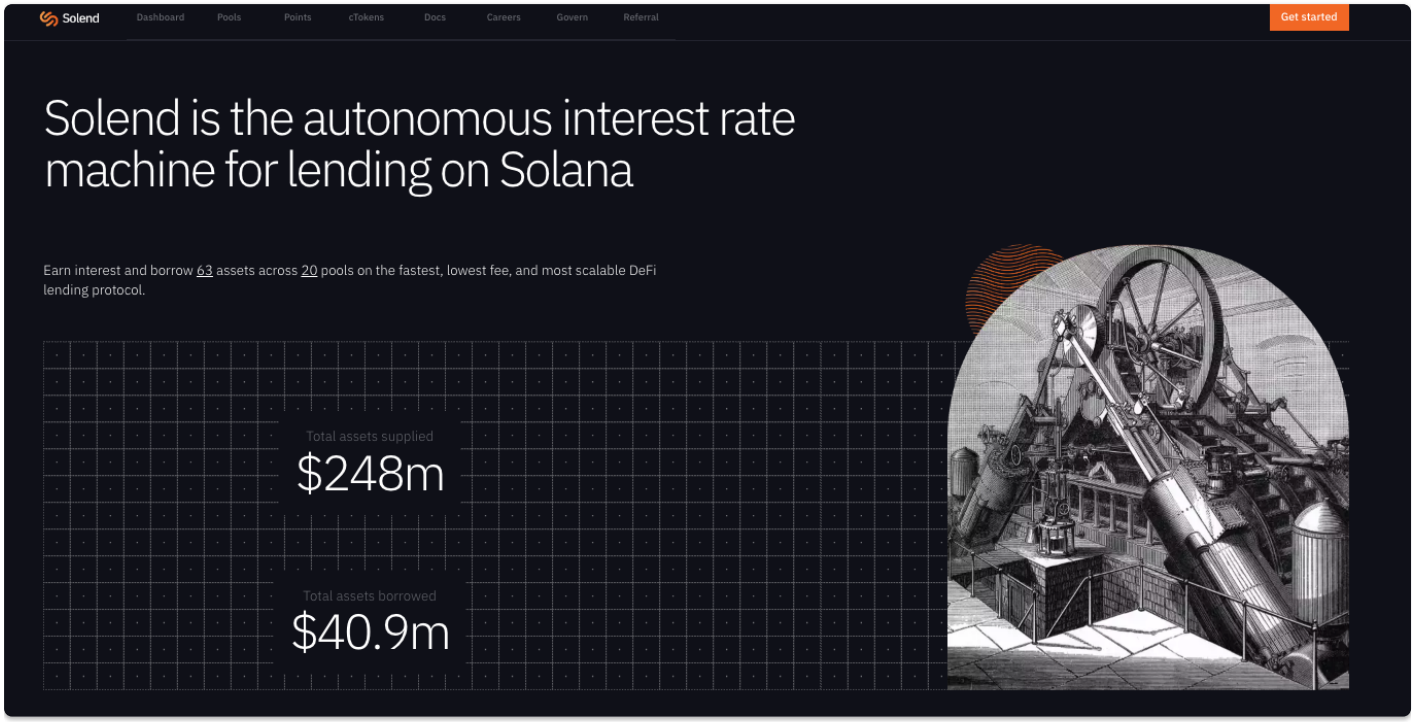


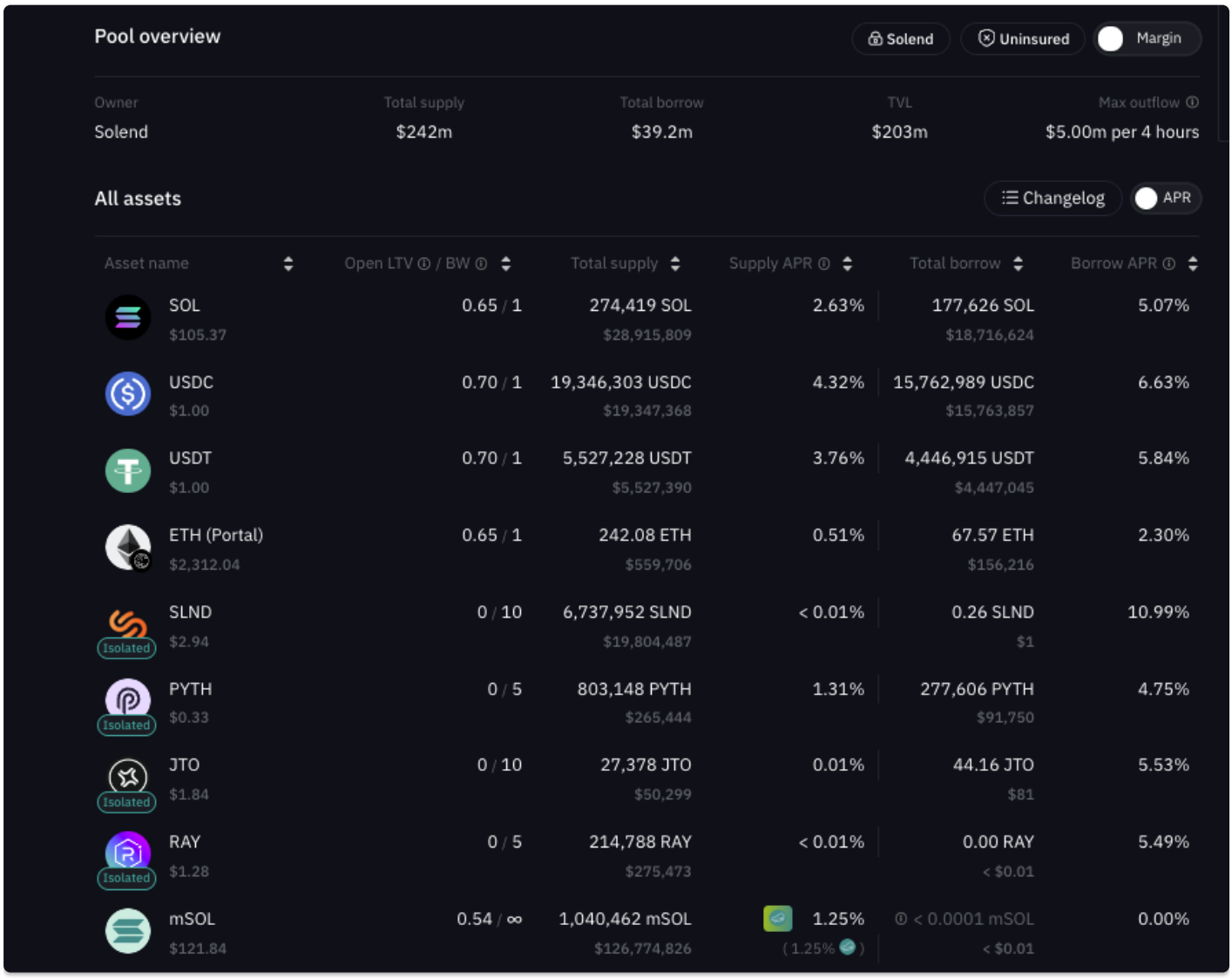
Kamino

Link

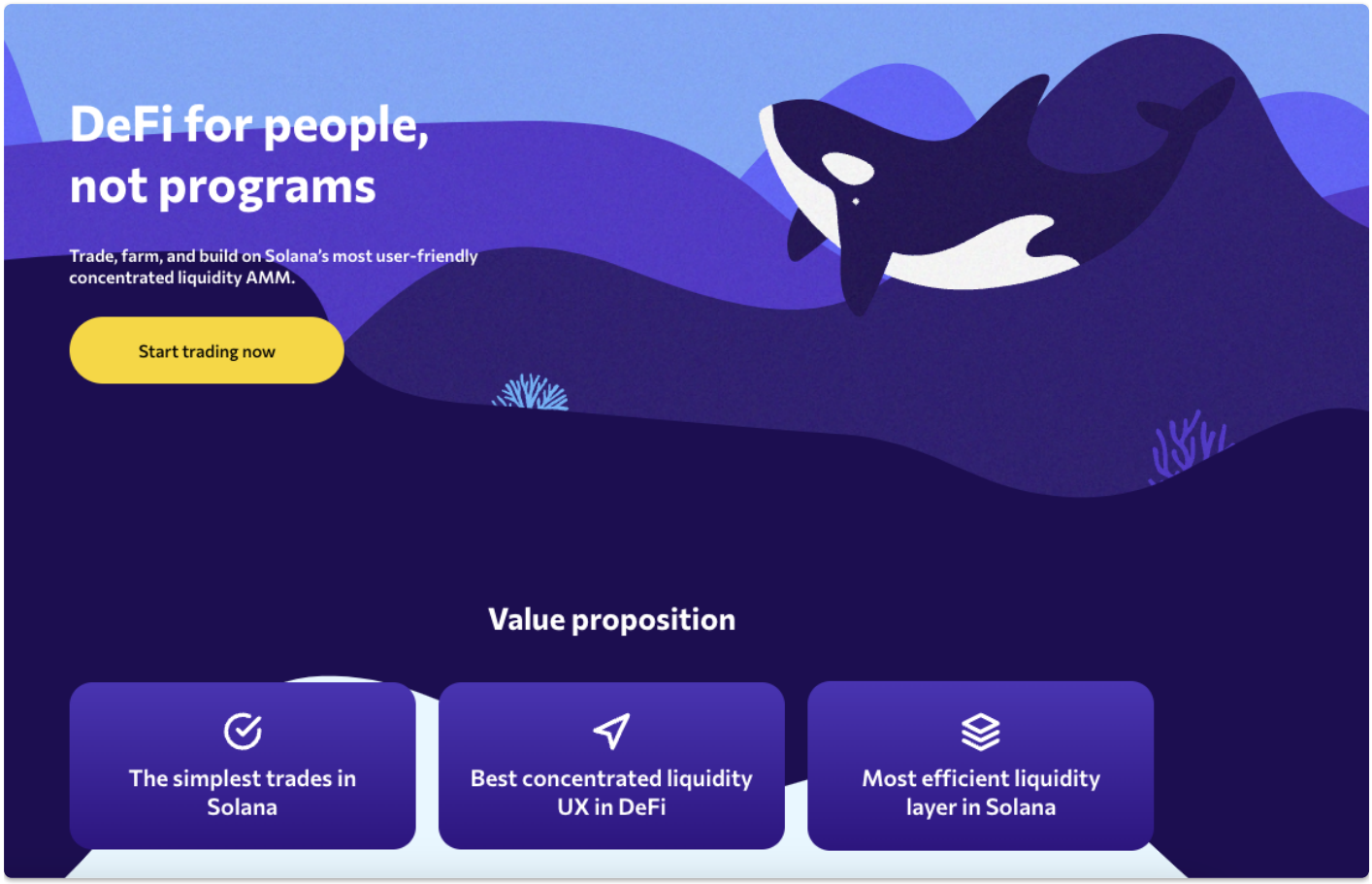

Borrow / Lend
Supply assets to earn yield and borrow against collateral. How it works
All Markets
Asset
Total Supply
Total Borrow
Max LTV
Supply APY
Borrow APY
SOL 498.93K $51.57M 261.11K $26.99M 65% 2.93% Supply 4.01% Borrow
USDC 26.21M $26.22M 22.90M $22.91M 80% 11.40% Supply 11.21% Borrow
bSOL 5x 490.19K $55.82M 44.23K $5.04M 45% 0.71% Supply 0.50% Borrow
JitoSOL 5x 294.26K $32.86M 54.75K $6.11M 45% 0.19% Supply 1.03% Borrow
mSOL 5x 199.30K $23.87M 39.31K $4.71M 45% 0.21% Supply 1.09% Borrow
JLP 6.00M $10.58M - - 50% - Supply -
USDT 8.86M $8.86M 7.62M $7.62M 80% 10.15% Supply 11.89% Borrow

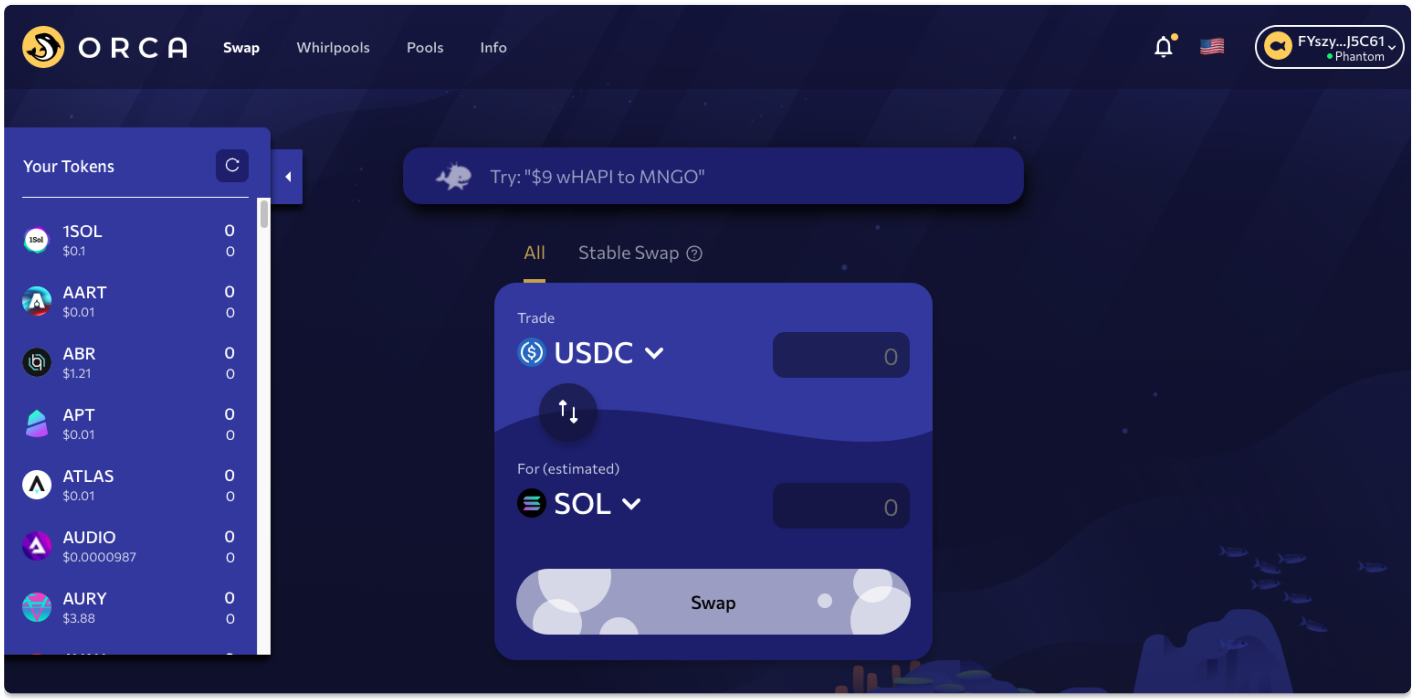
Solend

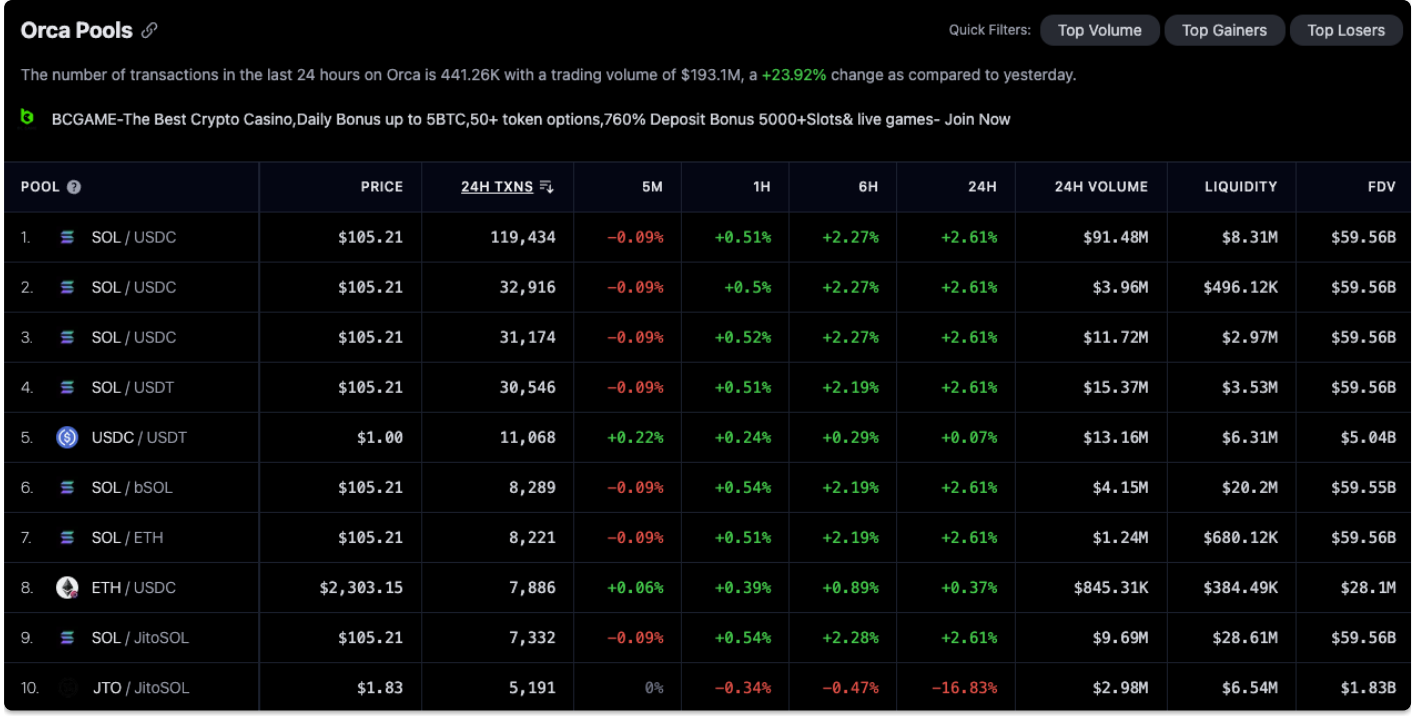

Solend
Dashboard
Pools
Points
cTokens
Docs
Careers
Govern
Referral
Get started
Solend is the autonomous interest rate machine for lending on Solana
Earn interest and borrow 63 assets across 20 pools on the fastest, lowest fee, and most scalable DeFi lending protocol.
Total assets supplied
$248m
Total assets borrowed
$40.9m


Pool overview
Solend
Uninsured
Margin
Owner
Solend
Total supply
$242m
Total borrow
$39.2m
TVL
$203m
Max outflow
$5.00m per 4 hours
All assets
Changelog
APR
Asset name
Open LTV / BW
Total supply
Supply APR
Total borrow
Borrow APR
SOL $105.37 | 0.65 / 1 | 274,419 SOL $28,915,809 | 2.63% | 177,626 SOL $18,716,624 | 5.07%
USDC $1.00 | 0.70 / 1 | 19,346,303 USDC $19,347,368 | 4.32% | 15,762,989 USDC $15,763,857 | 6.63%
USDT $1.00 | 0.70 / 1 | 5,527,228 USDT $5,527,390 | 3.76% | 4,446,915 USDT $4,447,045 | 5.84%
ETH (Portal) $2,312.04 | 0.65 / 1 | 242.08 ETH $559,706 | 0.51% | 67.57 ETH $156,216 | 2.30%
SLND Isolated $2.94 | 0 / 10 | 6,737,952 SLND $19,804,487 | < 0.01% | 0.26 SLND $1 | 10.99%
PYTH Isolated $0.33 | 0 / 5 | 803,148 PYTH $265,444 | 1.31% | 277,606 PYTH $91,750 | 4.75%
JTO Isolated $1.84 | 0 / 10 | 27,378 JTO $50,299 | 0.01% | 44.16 JTO $81 | 5.53%
RAY Isolated $1.28 | 0 / 5 | 214,788 RAY $275,473 | < 0.01% | 0.00 RAY < $0.01 | 5.49%
mSOL $121.84 | 0.54 / ∞ | 1,040,462 mSOL $126,774,826 | 1.25% (1.25%) | < 0.0001 mSOL < $0.01 | 0.00%

Orca


DeFi for people, not programs
Trade, farm, and build on Solana's most user-friendly concentrated liquidity AMM.
Start trading now
Value proposition
The simplest trades in Solana
Best concentrated liquidity UX in DeFi
Most efficient liquidity layer in Solana


ORCA
Swap
Whirlpools
Pools
Info
FYszy...J5C61
Phantom
Your Tokens
Try: "$9 wHAPI to MNGO"
All
Stable Swap
Trade
USDC
0
For (estimated)
SOL
0
Swap
1SOL $0.1
AART $0.01
ABR $1.21
APT $0.01
ATLAS $0.01
AUDIO $0.0000987
AURY $3.88



Orca Pools

Quick Filters: Top Volume | Top Gainers | Top Losers

The number of transactions in the last 24 hours on Orca is 441.26K with a trading volume of $193.1M, a +23.92% change as compared to yesterday.

	POOL	PRICE	24H TXNS	5M	1H	6H	24H	24H VOLUME	LIQUIDITY	FDV
1.	SOL / USDC	$105.21	119,434	-0.09%	+0.51%	+2.27%	+2.61%	$91.48M	$8.31M	$59.56B
2.	SOL / USDC	$105.21	32,916	-0.09%	+0.5%	+2.27%	+2.61%	$3.96M	$496.12K	$59.56B
3.	SOL / USDC	$105.21	31,174	-0.09%	+0.52%	+2.27%	+2.61%	$11.72M	$2.97M	$59.56B
4.	SOL / USDT	$105.21	30,546	-0.09%	+0.51%	+2.19%	+2.61%	$15.37M	$3.53M	$59.56B
5.	USDC / USDT	$1.00	11,068	+0.22%	+0.24%	+0.29%	+0.07%	$13.16M	$6.31M	$5.04B
6.	SOL / bSOL	$105.21	8,289	-0.09%	+0.54%	+2.19%	+2.61%	$4.15M	$20.2M	$59.55B
7.	SOL / ETH	$105.21	8,221	-0.09%	+0.51%	+2.19%	+2.61%	$1.24M	$680.12K	$59.56B
8.	ETH / USDC	$2,303.15	7,886	+0.06%	+0.39%	+0.89%	+0.37%	$845.31K	$384.49K	$28.1M
9.	SOL / JitoSOL	$105.21	7,332	-0.09%	+0.54%	+2.28%	+2.61%	$9.69M	$28.61M	$59.56B
10.	JTO / JitoSOL	$1.83	5,191	0%	-0.34%	-0.47%	-16.83%	$2.98M	$6.54M	$1.83B

Useful guides to getting started in DeFi : Part 1 , Part 2

AMM based Dex

Features

1. Fair Price Indicator

   Checks

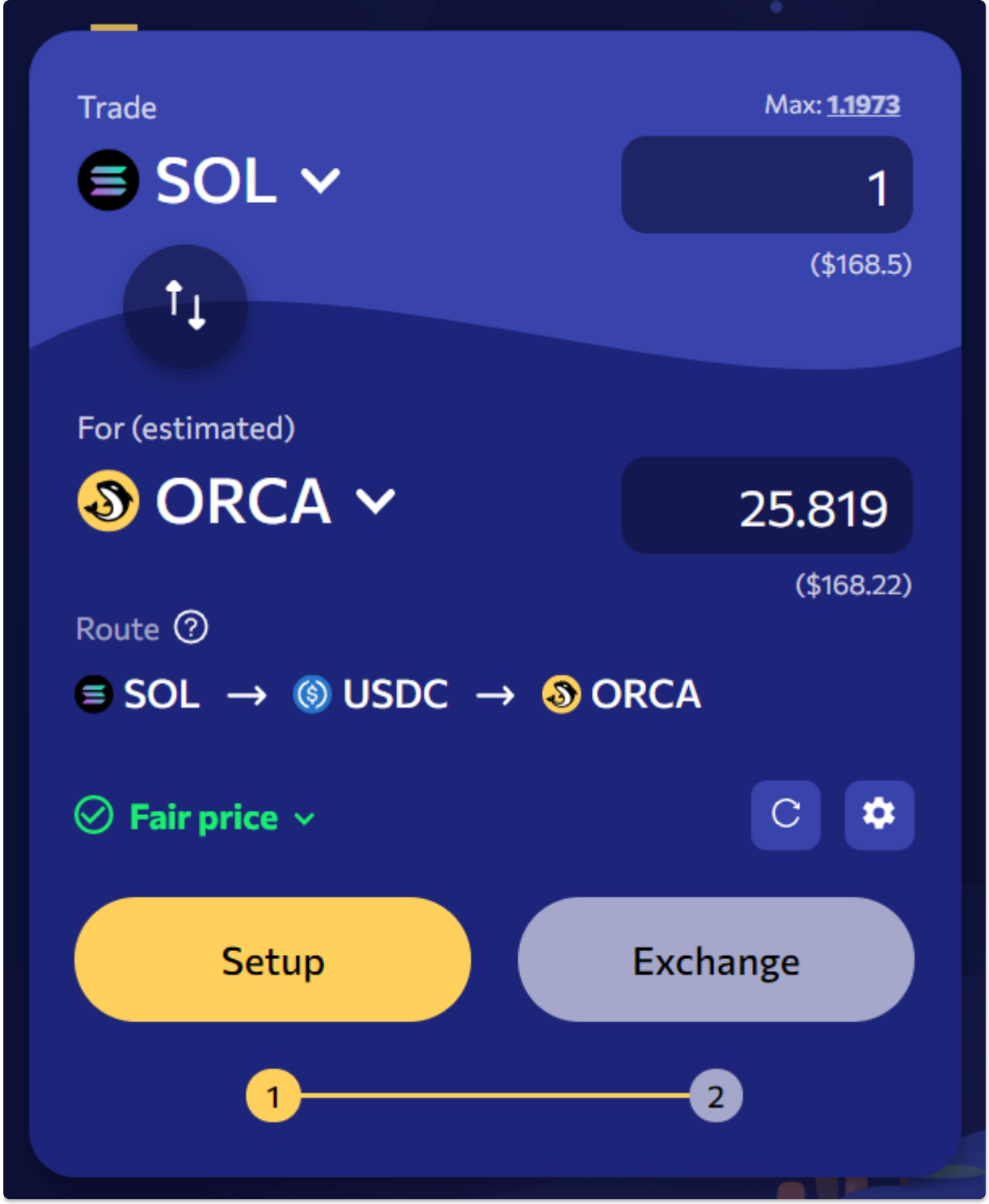
   1. The rate is within 1% of the listed market rate on CoinGecko
   2. Slippage is less than or equal to your tolerance setting (defaults to 1%)

2. Token Balances

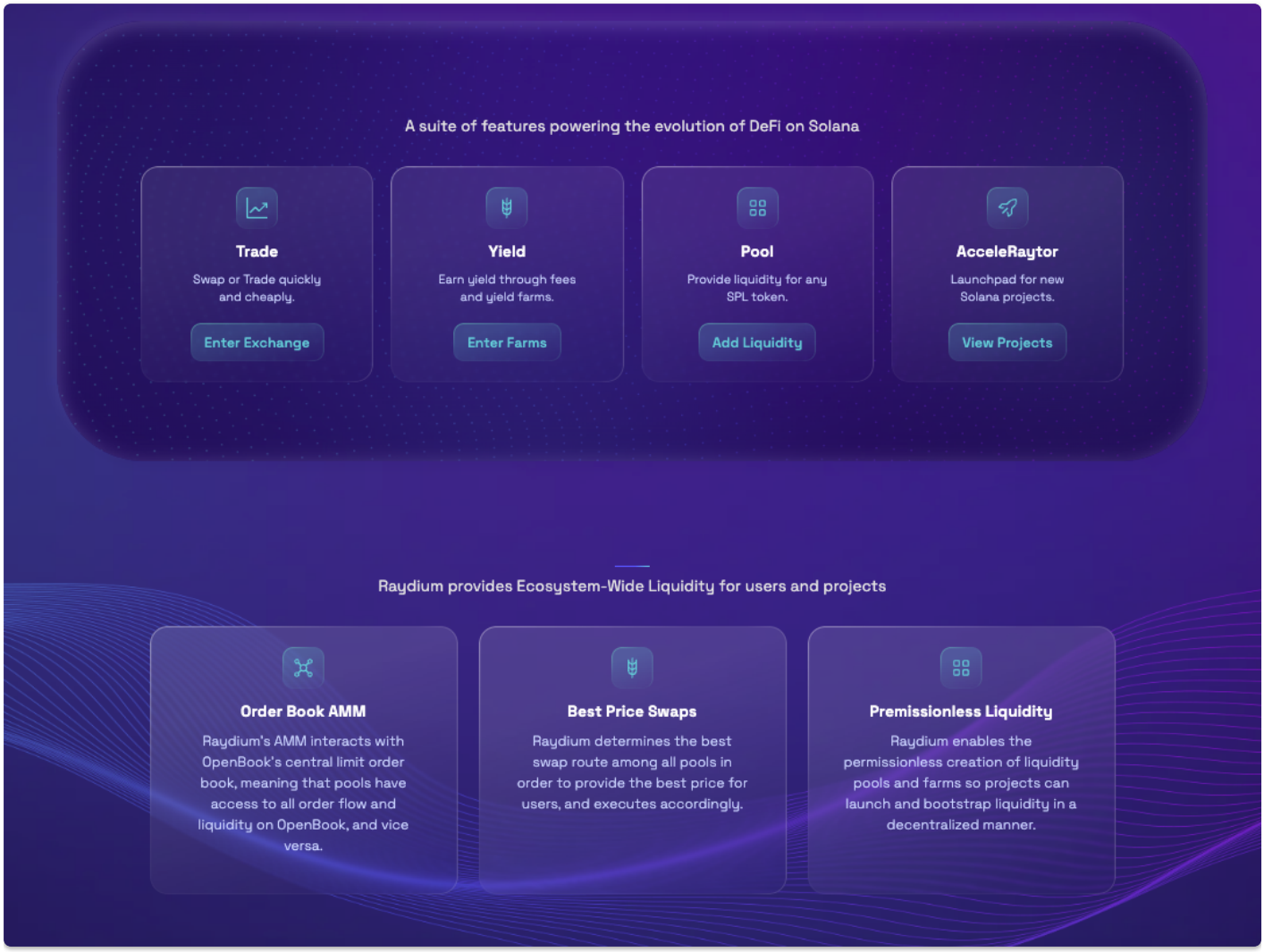
   Essentially a wallet displaying balances

3. Route Indication

   An indication of the route used if there is more than one swap involved


Trade
Max: 1.1973
SOL
1
($168.5)
For (estimated)
ORCA
25.819
($168.22)
Route
SOL → USDC → ORCA
Fair price
Setup
Exchange
1
2

Raydium



See Docs

From Docs

"Raydium offers a few key advantages:

1. Faster and cheaper:
   Raydium leverages the efficiency of the Solana blockchain to achieve transactions magnitudes faster than Ethereum and gas fees which are a fraction of the cost.
2. A central order book for ecosystem-wide liquidity: Raydium provides on-chain

liquidity to a central limit order book, meaning that Raydium allows access to 3rd party order flow and liquidity on the order book.

3. Trading interface:
   For traders who want to be able to view TradingView charts, set limit orders and have more control over their trading."

Stablecoins on Solana

From DeFI Llama

	Name	Chains	% Off Peg	1m % Off Peg	Price	1d Change	7d Change	1m Change	Market Cap
1	USD Coin (USDC)	+53	0%	+0.12%	$1.00	-14.83%	-19.28%	-38.24%	$1.28b
2	Tether (USDT)	+55	-0.12%	-0.20%	$1.00	-0.06%	-55.57%	-55.85%	$799.31m
3	Parrot USD (PAI)		-0.49%	-1.78%	$1.00	0%	+0.36%	-9.56%	$17.09m
4	UXD Stablecoin (UXD)		-0.03%	-0.05%	$1.00	-0.09%	-0.17%	+6.73%	$16.01m
5	Hedge USD (USH)		-0.01%	-0.88%	$1.00	+1.98%	+31.50%	-64.14%	$3,421,554
6	USDH (USDH)		-0.18%	-0.45%	$1.00	+0.03%	-0.30%	-76.00%	$2,074,585
7	Ratio Stable Coin (USDR)		-0.78%	-1.22%	$0.99	-0.06%	-10.33%	-36.81%	$1,941,241
8	Dai (DAI)	+34	+0.01%	+0.36%	$1.00	-0.07%	-7.30%	-14.61%	$1,402,833
9	Binance USD (BUSD)	+27	+0.03%	+0.35%	$1.00	+2.09%	-22.28%	-88.44%	$472,618
10	TerraClassicUSD (UST...	+12	-97.95%	-98.06%	$0.02	-0.36%	-9.72%	-46.86%	$213,637

Parrot

The Parrot Protocol is a DeFi network built on Solana that will include the stablecoin PAI, a non-custodial lending market, and a margin trading vAMM. These are all use cases designed

to solve one single problem: making value locked in DeFi systems accessible.

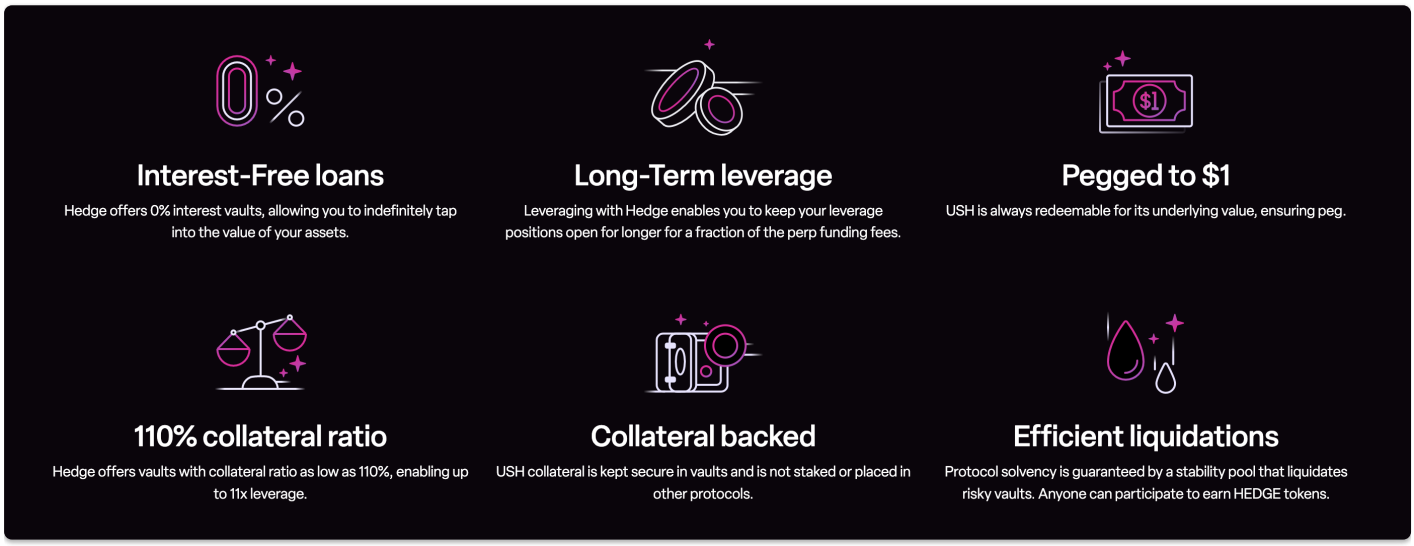
The Parrot Protocol is setting out to make value locked in LP tokens accessible, by creating a liquidity & lending network collateralised by these LP tokens. T

- Create the PAI stablecoin, backed by LP tokens as collaterals. This creates a common unit of account to make it easy for holders of different types of LP tokens to transact with each other.
- Create the Parrot Lending market, taking LP tokens as collaterals. This allows LP holders to access their locked value by borrowing against lender liquidity.
- Create a margin trading product (virtual AMM) using PAI as the common unit of account. This allows the Parrot community to collect fees.

Hedge USD

See Docs

See White paper



Hedge is a protocol that enables the minting of USH, a stablecoin soft-pegged to the US dollar. USH is minted on flexible terms and gives users instant access to the USH ecosystem.

Hedge allows users to take out 0% interest loans for a one-time fee by depositing collateral for USH. Users are incentivised to keep their collateral-to-debt ratio above 110% to avoid liquidation. When an under collateralized vault is liquidated, users who have deposited USH in a stability pool are returned discounted collateral as a reward. USH is always

redeemable for its underlying value, but a fee ensures the protocol is impacted infrequently.

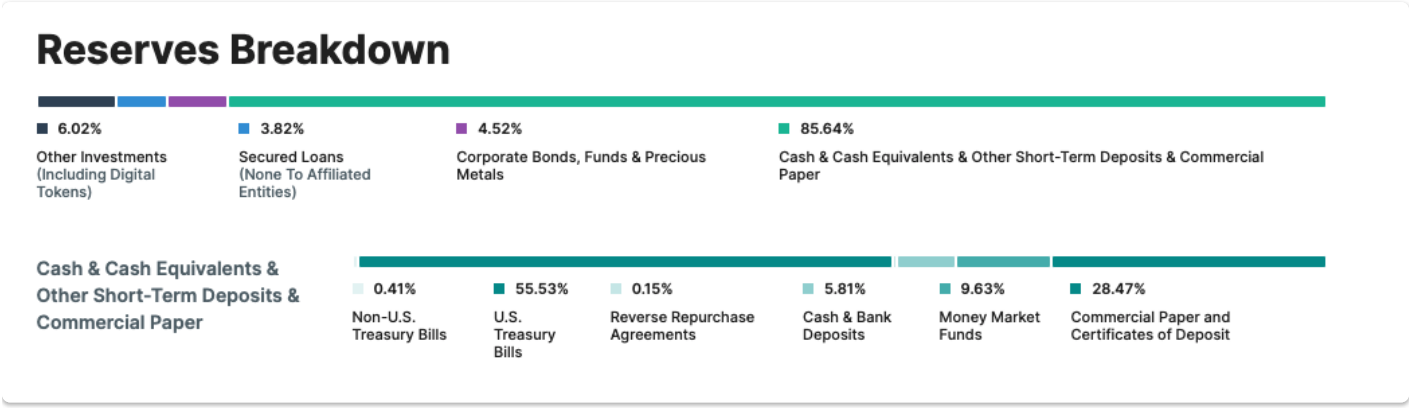
USH Stablecoin

The ceiling price of USH is $1.10, as users can always deposit the equivalent of $110 of collateral to mint $100 worth of USH. If USH is trading above $1.10, a profit could be made by selling USH on the market.

To establish a floor price for USH, there are two different mechanisms. If a user has a loan and finds USH to be below $1, they can repay their loan at a lower price by buying more USH, in turn stabilising the price of USH. Alternatively, they can redeem their USH.

The USH token mint is 9iLH8T7zoWhY7sBmj1WK9ENbWdS1nL8n9wAxaeRitTa6.

Tether



UXD Protocol

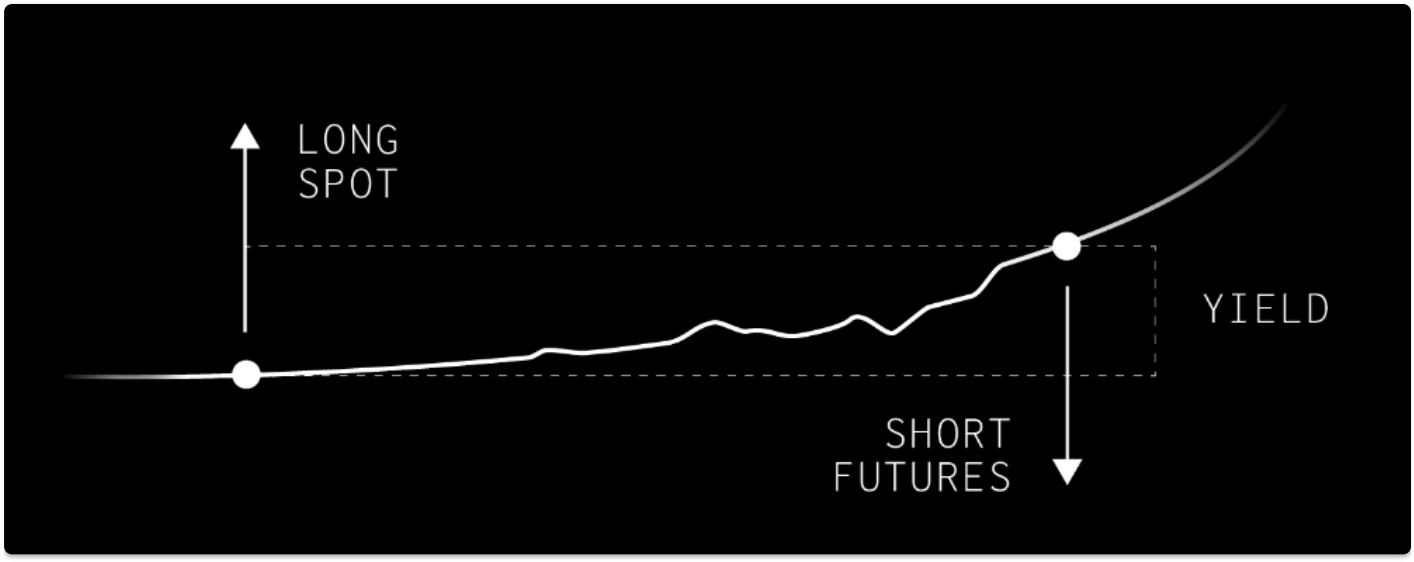
See UXD

Coindesk Article

Algorithmic stablecoin backed 100% by a delta neutral position using derivatives.

Launched in January 2022

Aims to have a yield of about 10%



Connected to Mango markets DEX