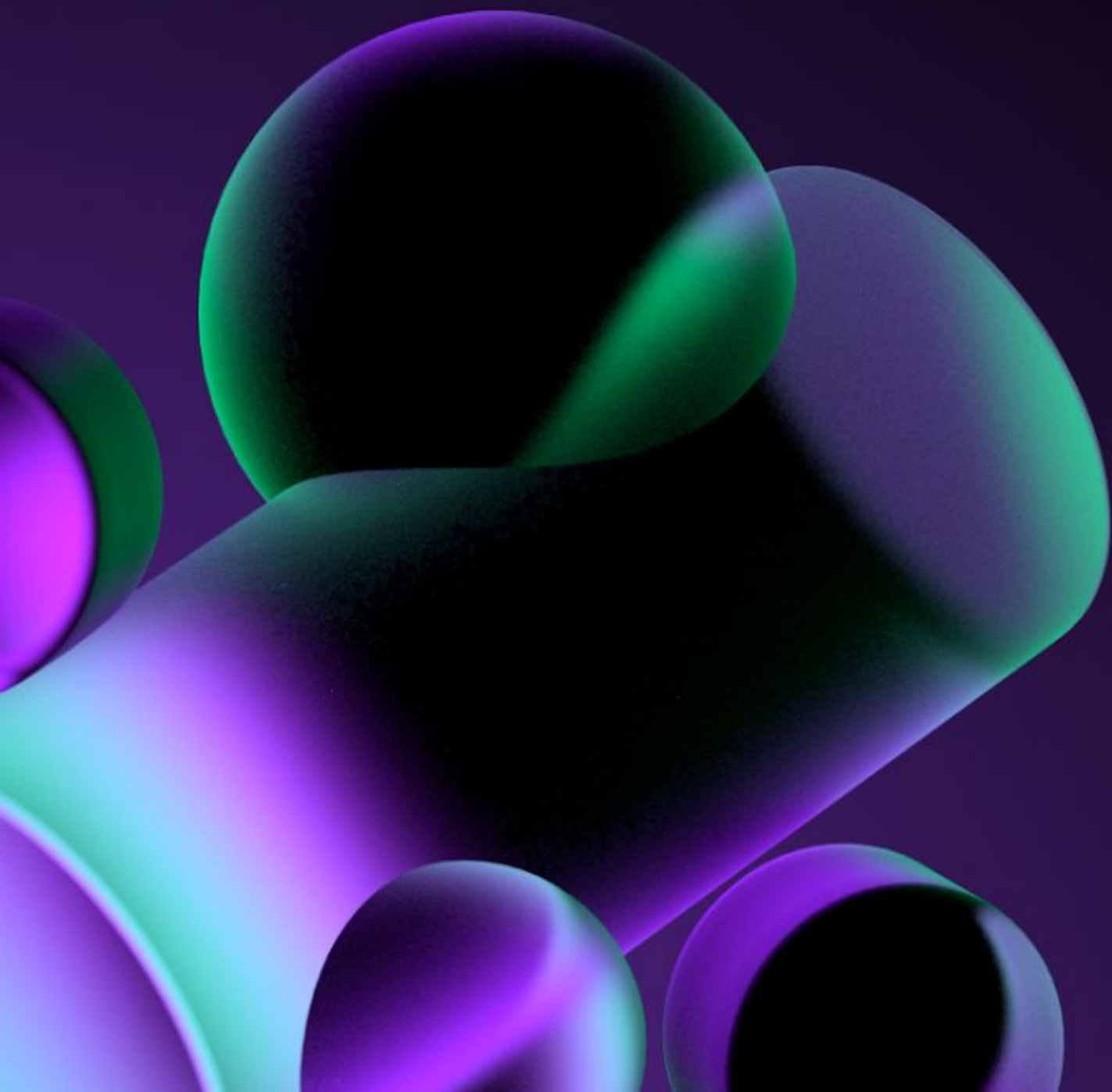


School of Solana

LECTURE 4

Solana Programming model 2/2

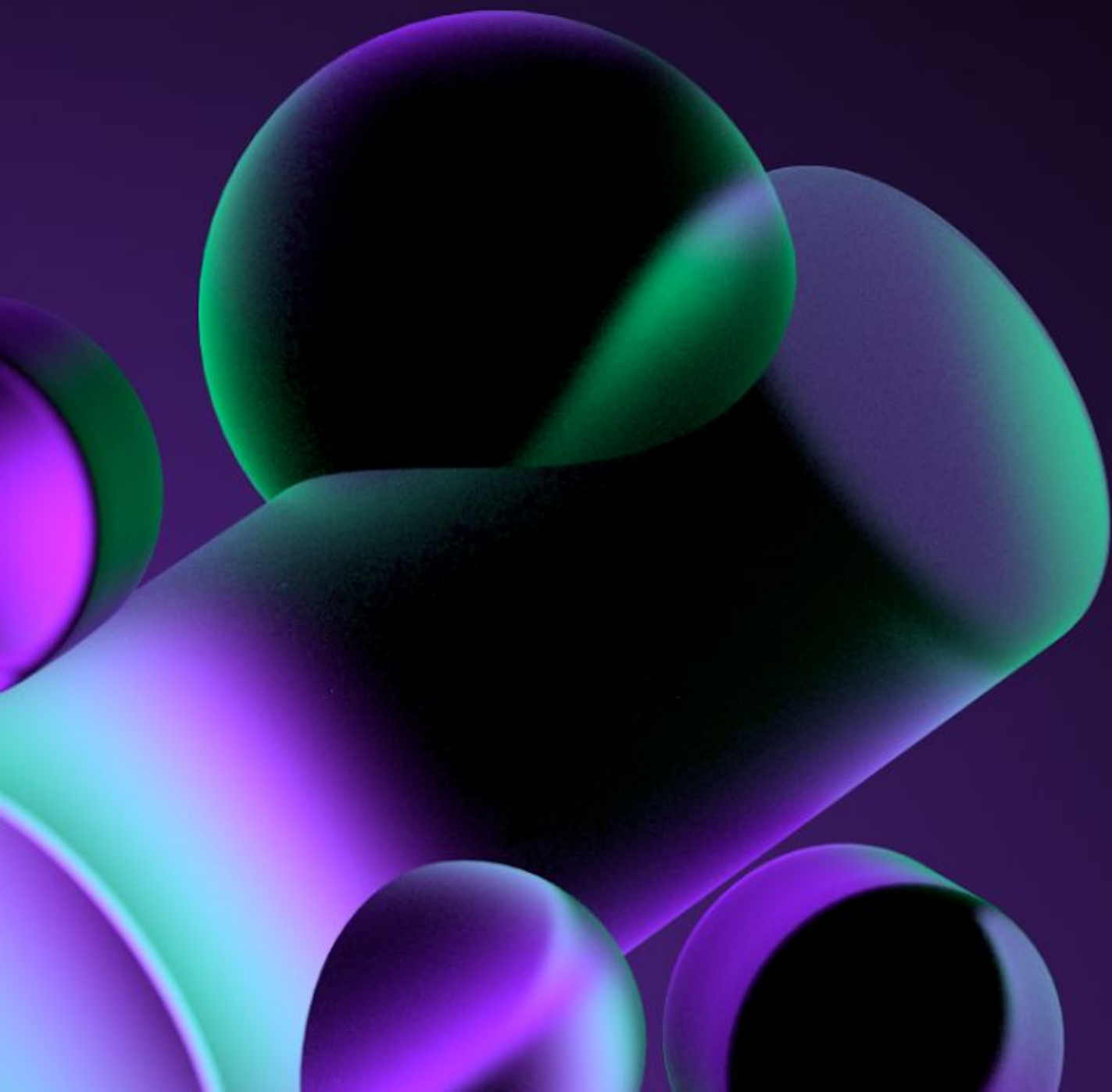
About this lecture



About this lecture

- Solana Programming model 2
 - Recap
 - Program Derived Addresses (PDAs)
 - SPL Tokens
- Hands on Example
 - PDA

Recap



Accounts

- Solana **accounts**:
 - Program accounts
 - Data accounts
- Program accounts **do not store state!**
- Only a data account's **owner** can **modify its data** and **subtract lamports**.
- To prevent an account from being deleted, **you must pay rent**.

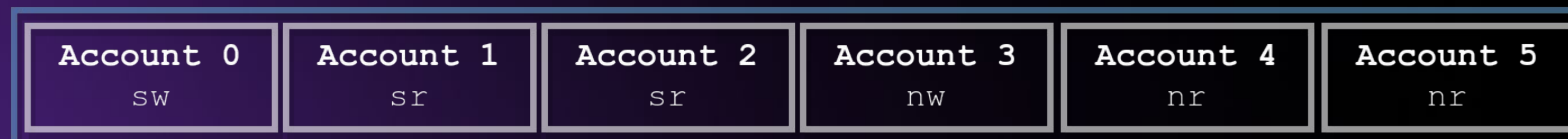
Transactions

- The basic operational unit on Solana is an **instruction**.
- One or more instructions can be bundled into a **transaction**.
- Instructions in one transaction are processed **in order** and **atomically**.
- You must **forward-declare every account** you intend to read from or write to.
 - be aware of potential **SECURITY RISKS**

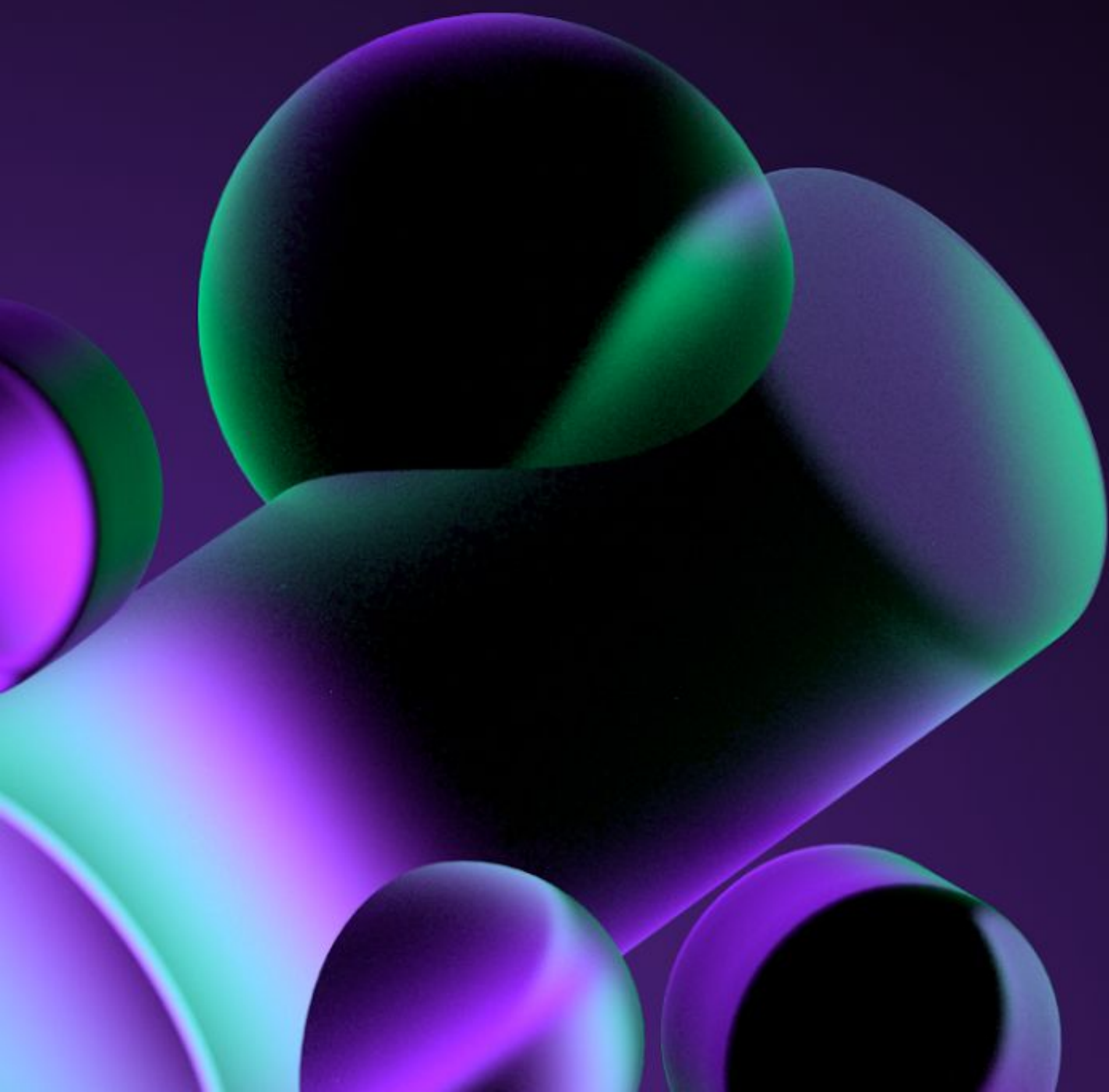
Instruction Format



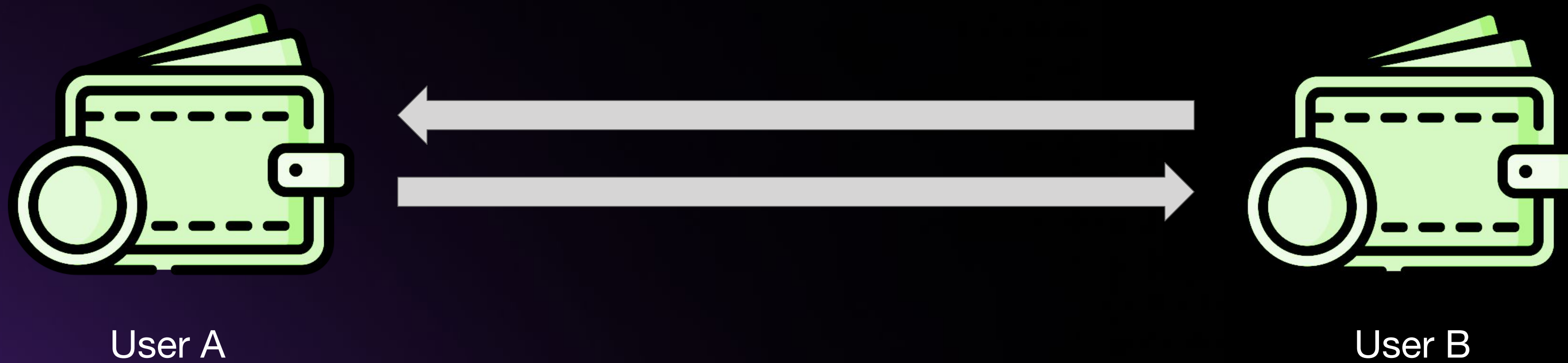
s - signed
 n - not signed
 w - writable
 r - read only



PDAs Basics



PDA's Motivation



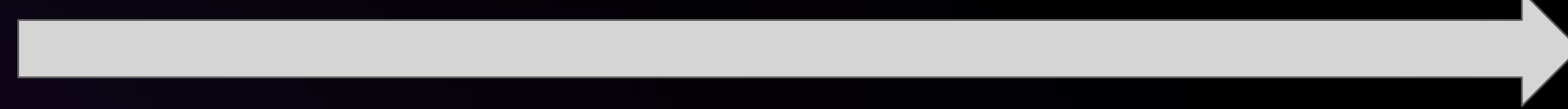
PDA's Motivation



PDA's Motivation



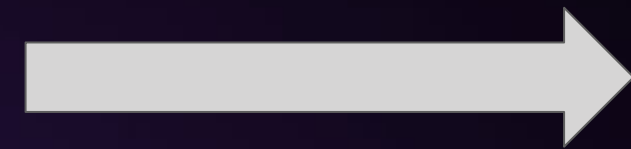
User A



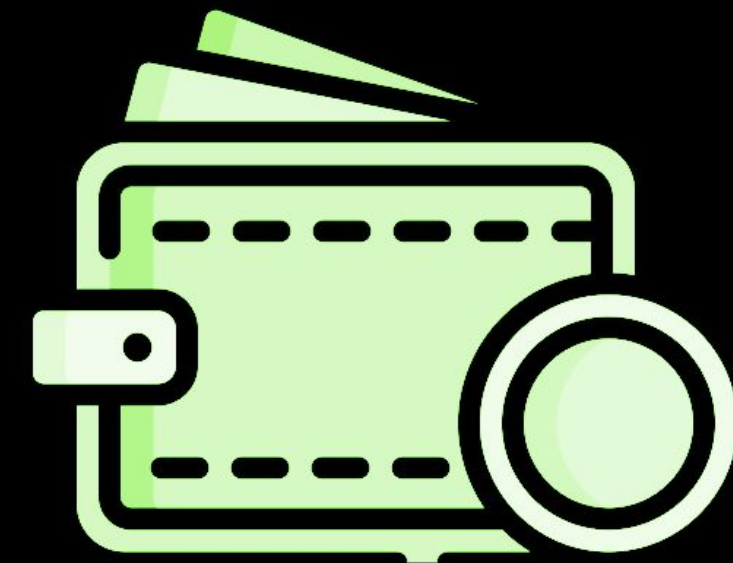
Program Account

PDA's Motivation

User requested our
program to
withdraw funds
stored in the
campaign.

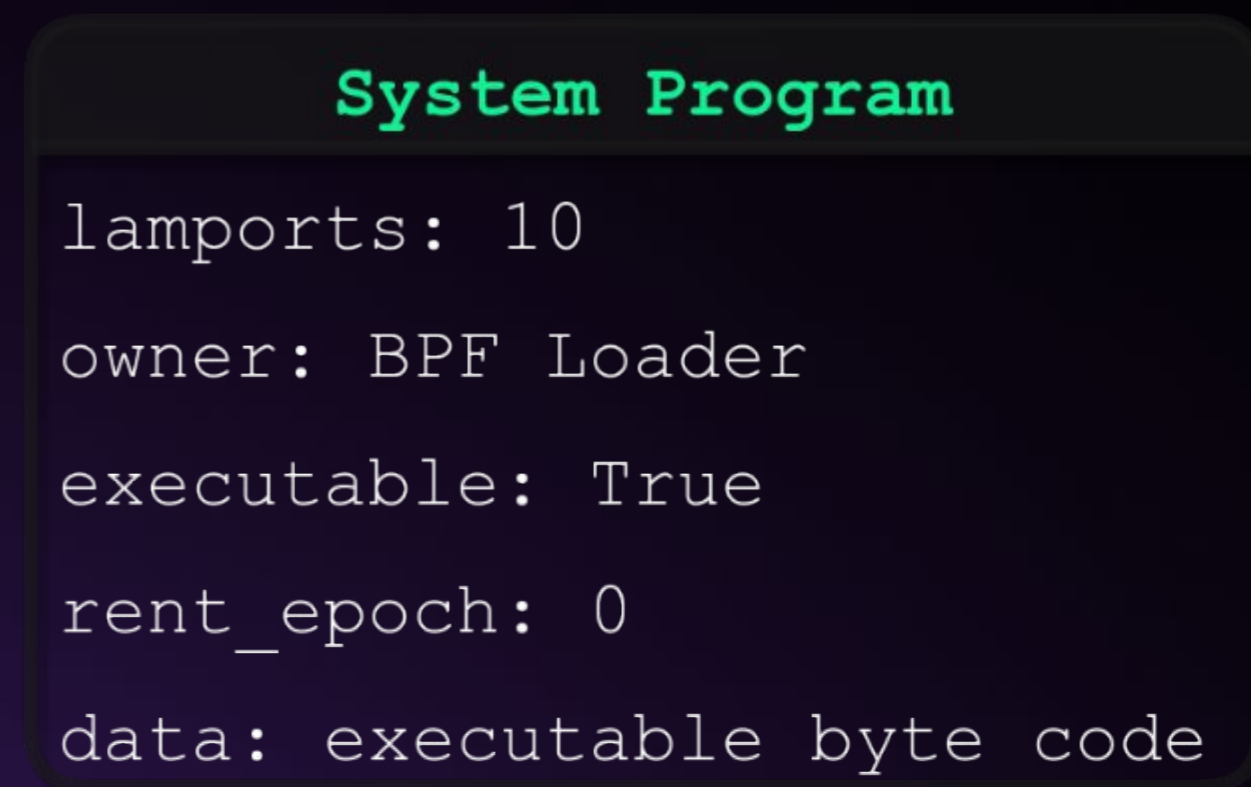


Program Account



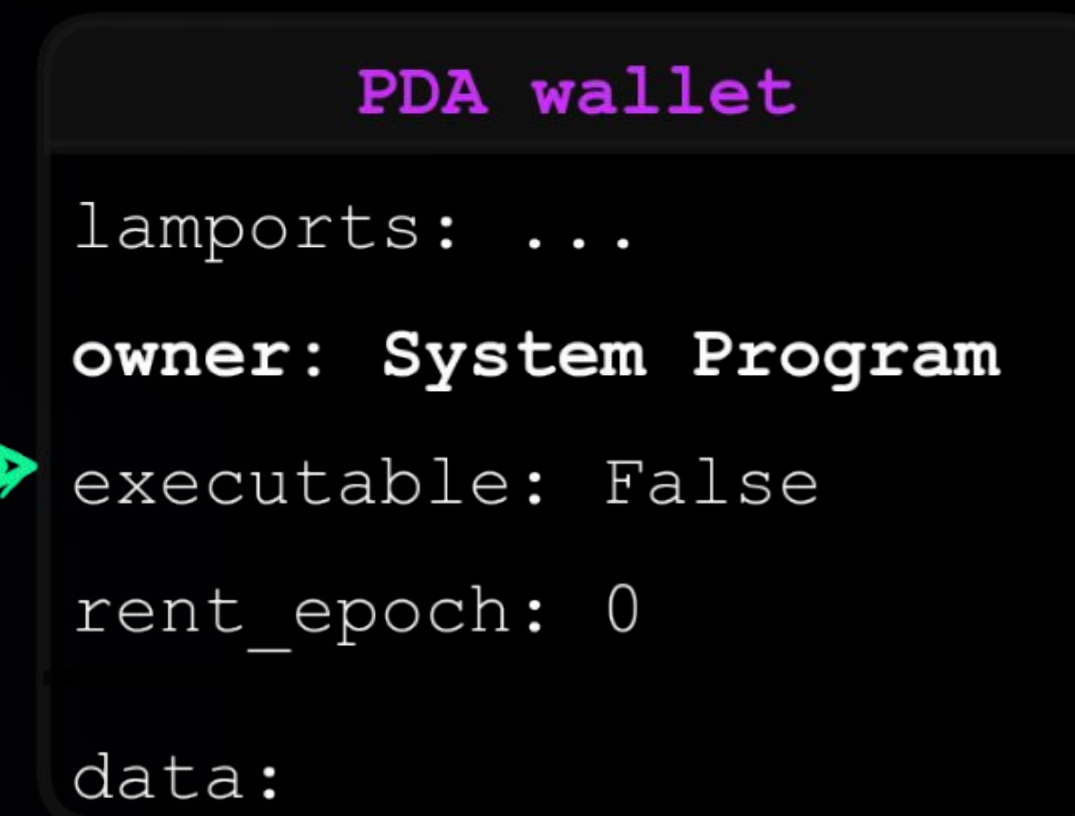
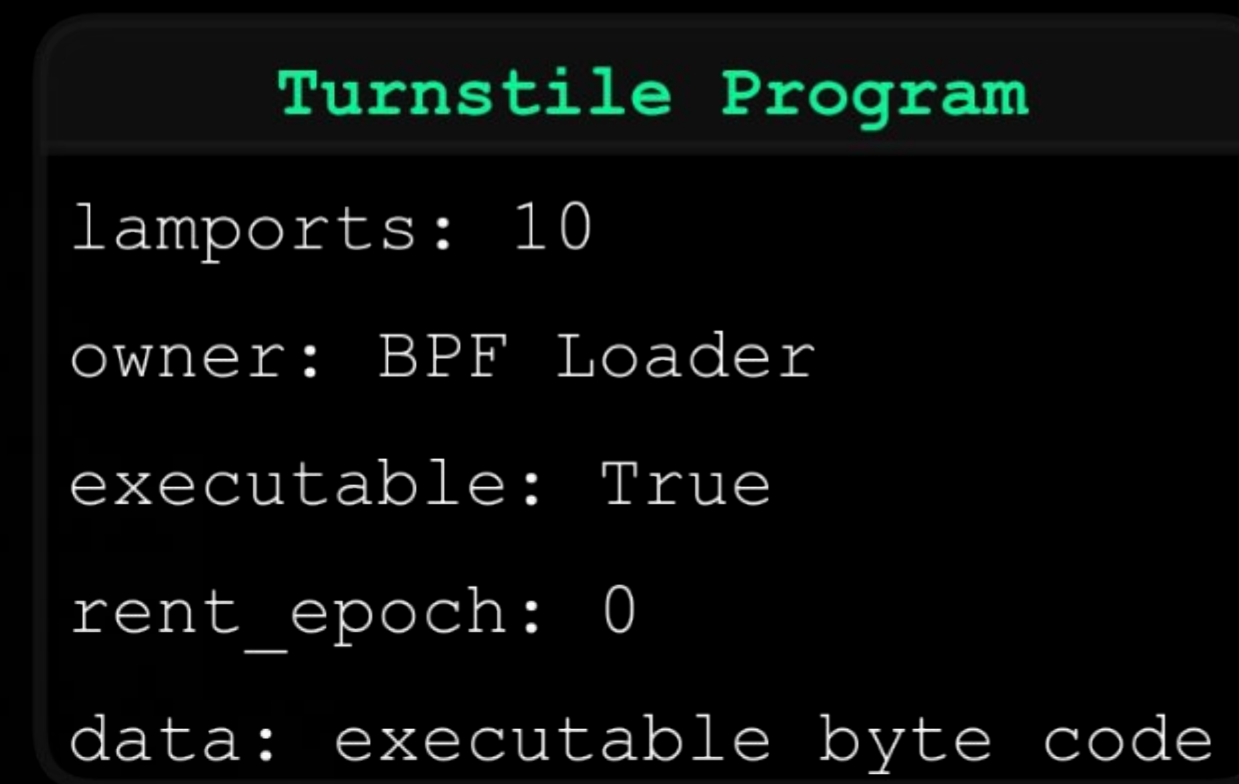
User A

PDA (invoke_signed)



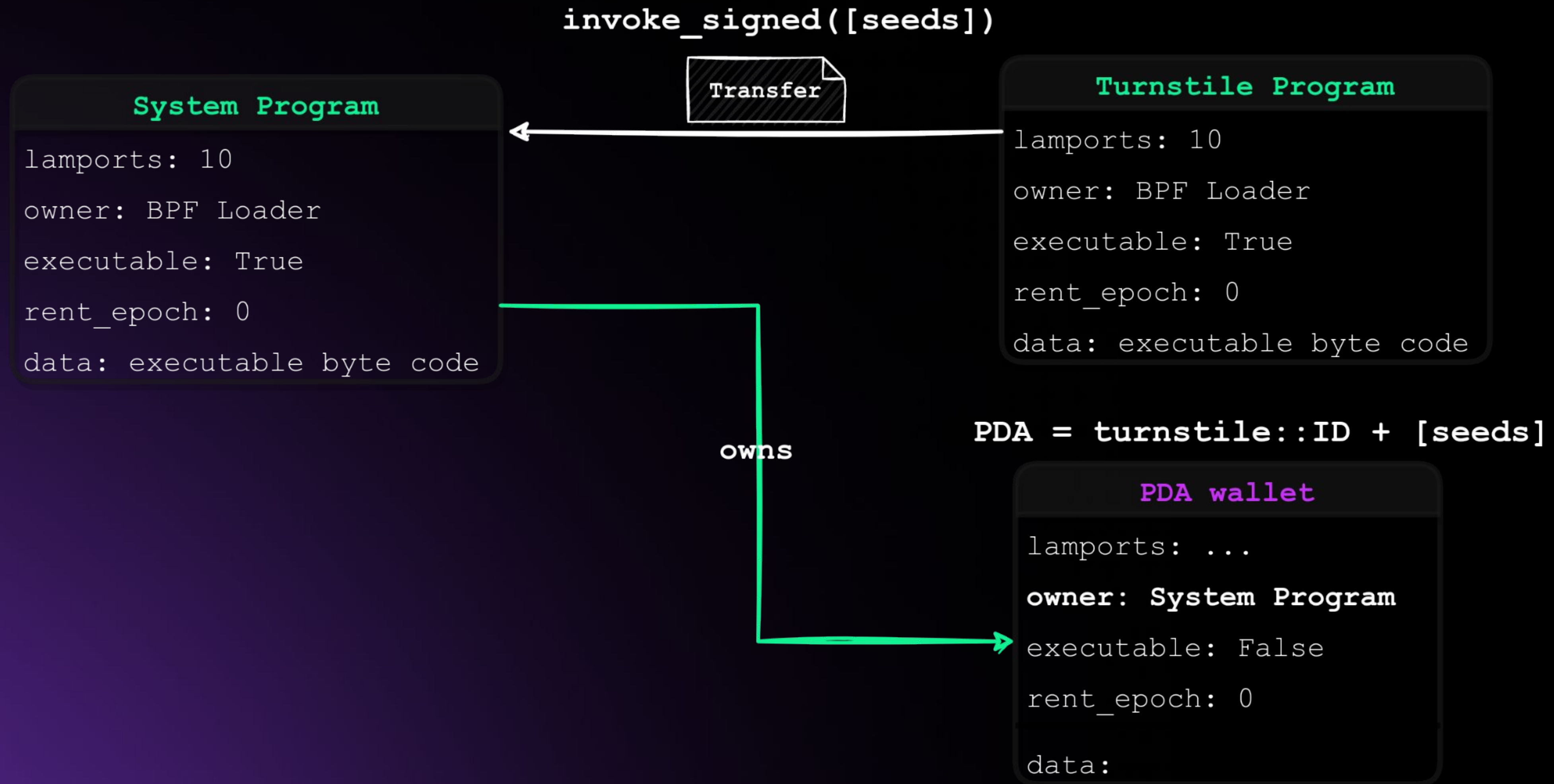
owns

PDA = turnstile::ID + [seeds]



- PDA does not have a PK, so we have a different signing mechanism.
- PDA address is derived from our program address + some seed.
- Signing instructions is then executed using a invoke_signed instruction.

PDA (invoke_signed)



PDA

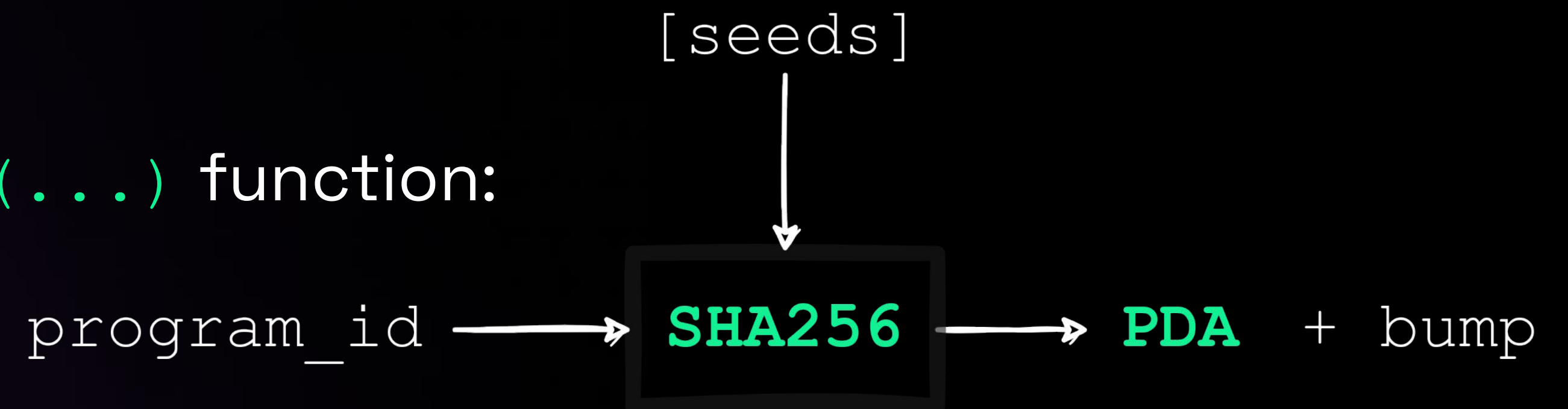
- Programs want to be able to sign accounts for cross-program invocation.

- PDA = program derived address

- PDA account = an account whose address is a PDA.

- How does a PDA get derived?

- `Pubkey::find_program_address(...)` function:



- PDA is bumped off the Ed25519 elliptic curve. Hence, there is no private key.

Abstract 3D geometric shapes, including spheres and cylinders, in shades of blue and purple, positioned in the bottom-left corner of the slide.

Solana Program Library

The advantage of the program/account model is that you can have **one generic program** that operates on **various data**.

Solana Program Library (SPL)

The screenshot displays the GitHub repository for the Solana Program Library (SPL). The repository is maintained by **joncinque** and has 36 branches and 64 tags. The commit history shows updates for token-2022 (#2984) and other recent commits.

Folder	Commit Message	Time Ago
.github	Run CI on version updates (#2981)	4 days ago
.travis	Switch to Github Actions for CI (#768)	16 months ago
associated-token-account/program	Bump solana to v1.9.9 (#2902)	11 days ago
binary-option	Bump solana to v1.9.9 (#2902)	11 days ago
binary-oracle-pair	Bump solana to v1.9.9 (#2902)	11 days ago
ci	fuzz: Fix fuzz build (#2980)	4 days ago
docs	Update Token TS Docs (#2938)	13 days ago
examples	Bump solana to v1.9.9 (#2902)	11 days ago
farms	Bump solana to v1.9.9 (#2902)	11 days ago
feature-proposal	Bump solana to v1.9.9 (#2902)	11 days ago
governance	Bump solana to v1.9.9 (#2902)	11 days ago
libraries/math	Bump solana to v1.9.9 (#2902)	11 days ago
memo	Bump solana to v1.9.9 (#2902)	11 days ago
name-service	Bump rust versions (#2978)	4 days ago
record/program	Bump solana to v1.9.9 (#2902)	11 days ago
shared-memory	Bump solana to v1.9.9 (#2902)	11 days ago
stake-pool	Bump solana to v1.9.9 (#2902)	11 days ago
stateless-asks	Bump solana to v1.9.9 (#2902)	11 days ago
token-lending	Bump solana to v1.9.9 (#2902)	11 days ago
token-swap	Bump solana to v1.9.9 (#2902)	11 days ago
token	token-2022: minor rename pod types with descriptive types (#2983)	3 days ago
utils	Bump solana to v1.9.9 (#2902)	11 days ago

About
A collection of Solana-maintained on-chain programs
solana.com
Readme
View license
1.5k stars
72 watching
863 forks

Releases 40
SPL Stake Pool and CLI - v0.6.4 (Latest)
19 days ago
[+ 39 releases](#)

Packages
No packages published

Used by 908
 + 900

Contributors 94
 + 83 contributors

<https://github.com/solana-labs/solana-program-library>

Solana Tokens

ERC20 token = a token whose smart contract uses the ERC20 standard



- Fungible tokens use the **ERC20 standard**.
- **Smart contract** template to create a new token.
- To create a new token, you **deploy** ERC20 smart contract on chain.

SPL token = a token created with the Solana token program



- There no such thing as a standard.
- On Solana, there is a **single token program** (spl-token-program).
- We already talked about how to interact with SPL token program via Solana CLI.

Solana Tokens

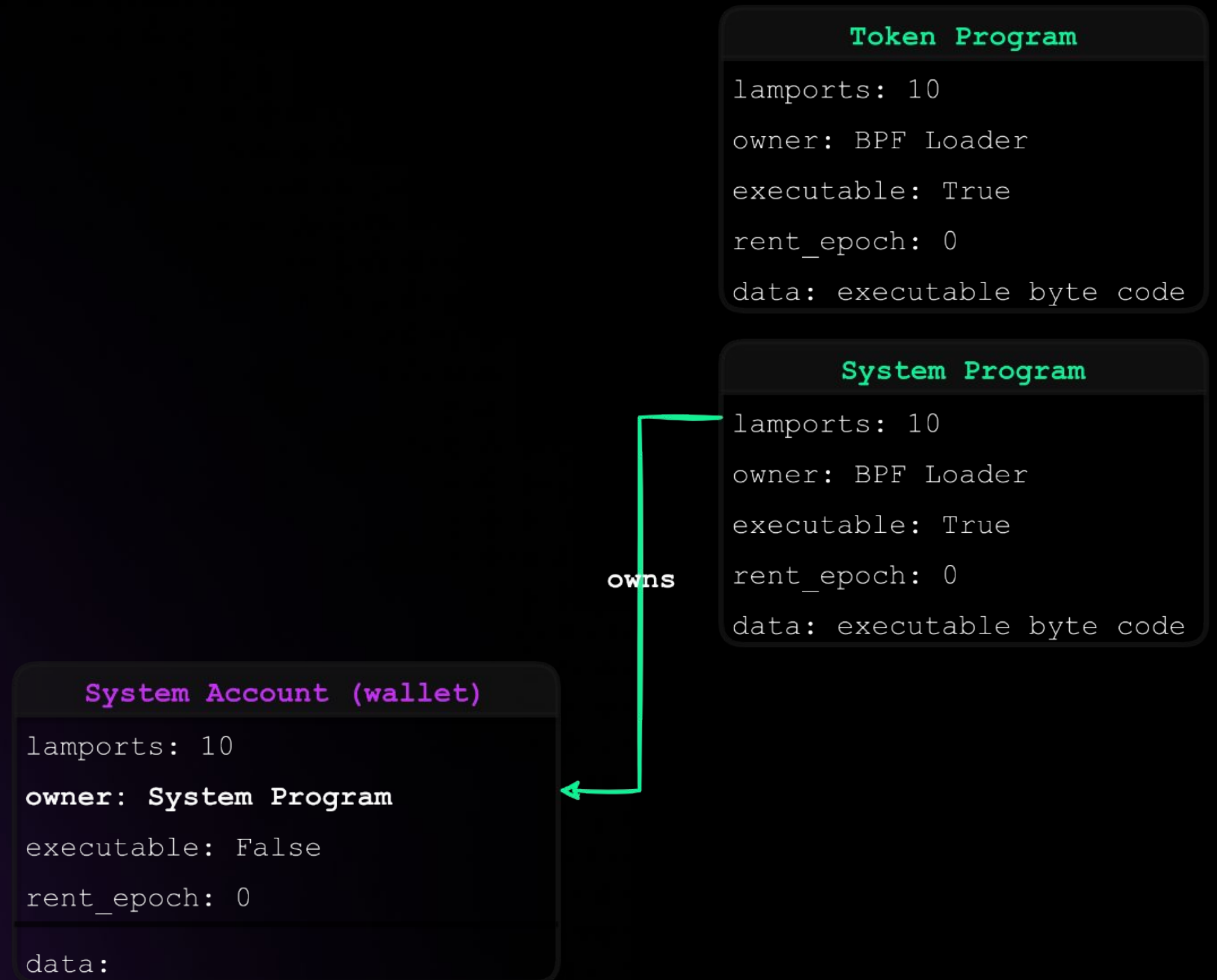
Token Program

```
lamports: 10  
owner: BPF Loader  
executable: True  
rent_epoch: 0  
data: executable byte code
```

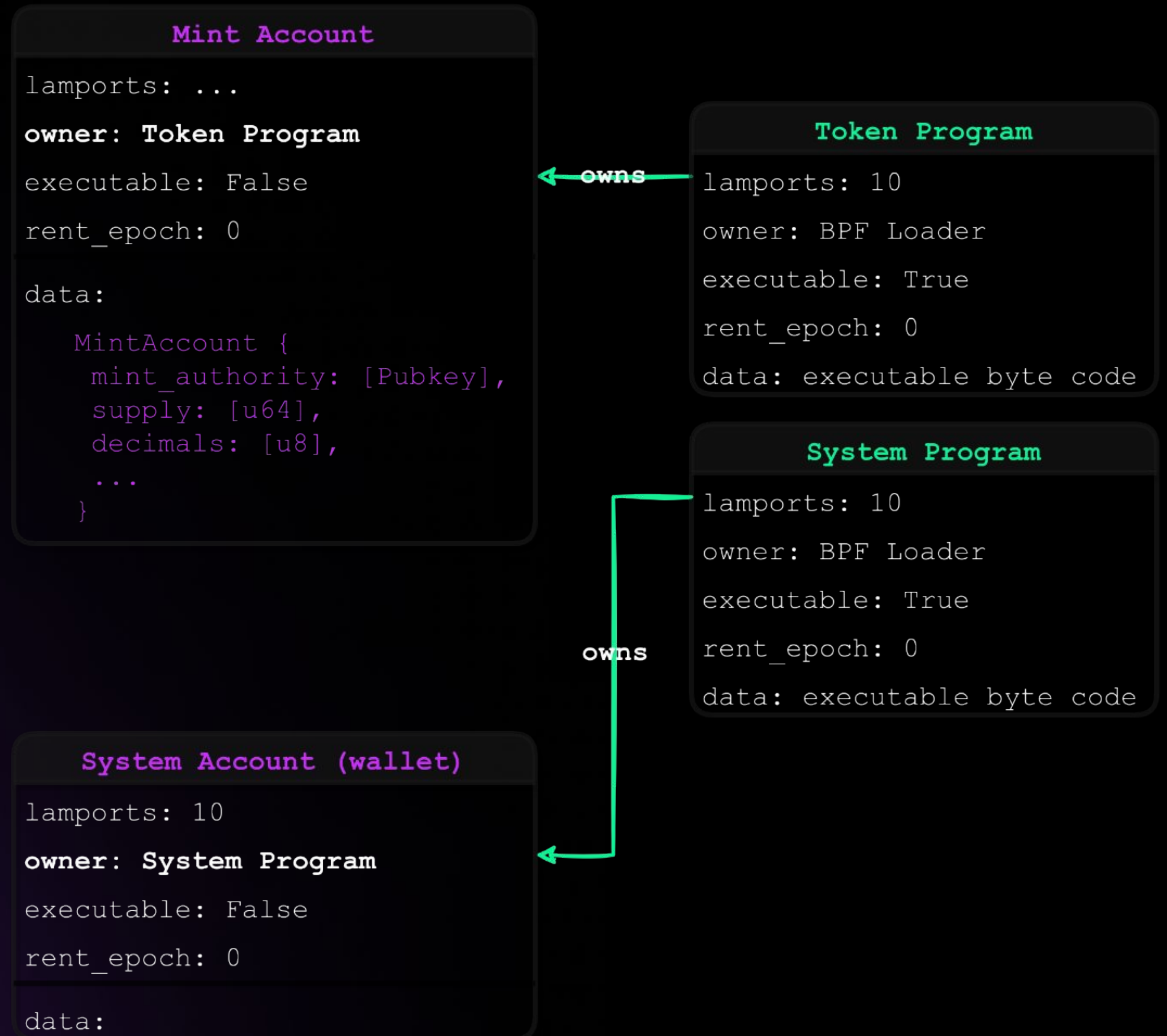
System Program

```
lamports: 10  
owner: BPF Loader  
executable: True  
rent_epoch: 0  
data: executable byte code
```

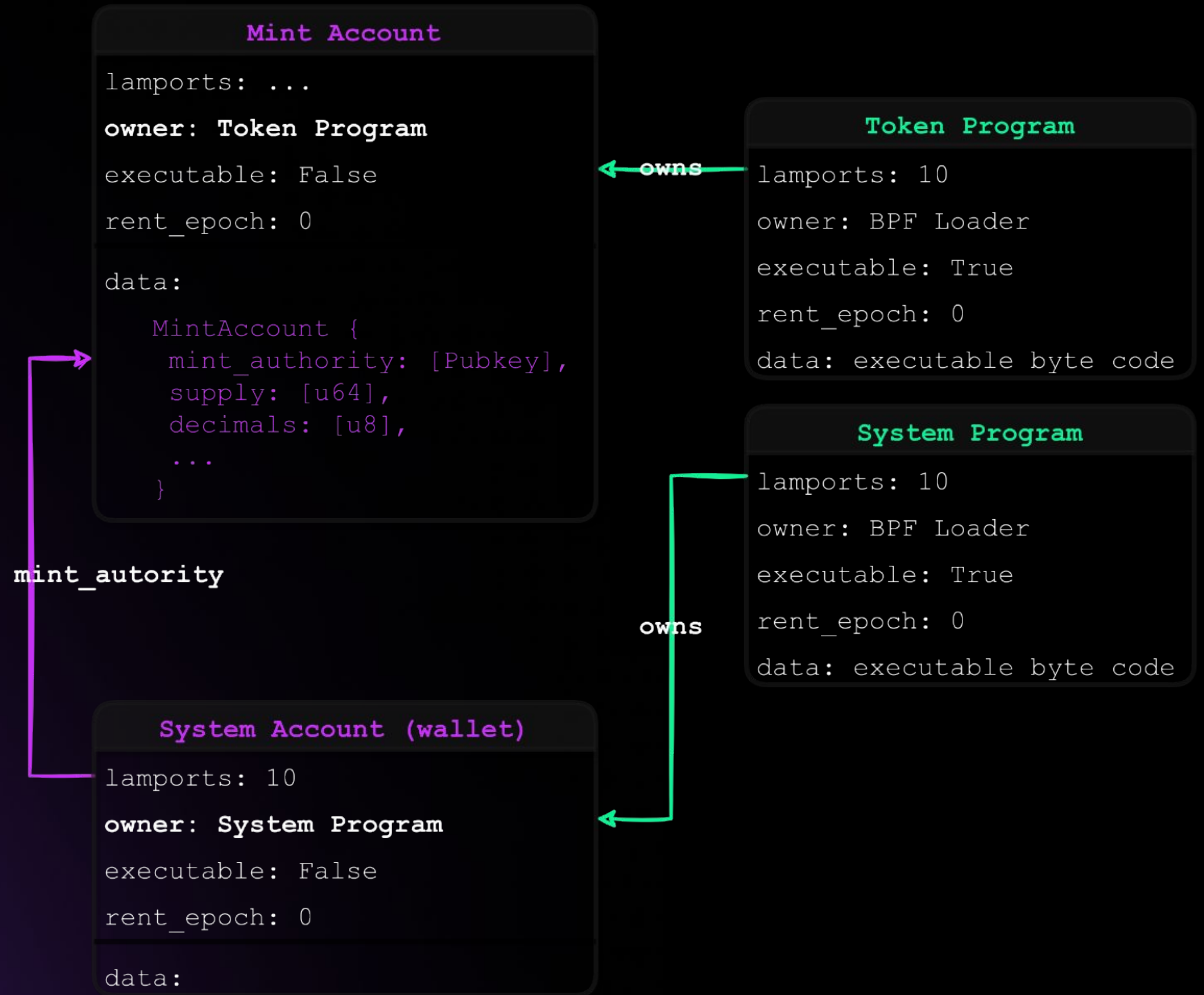
Solana Tokens



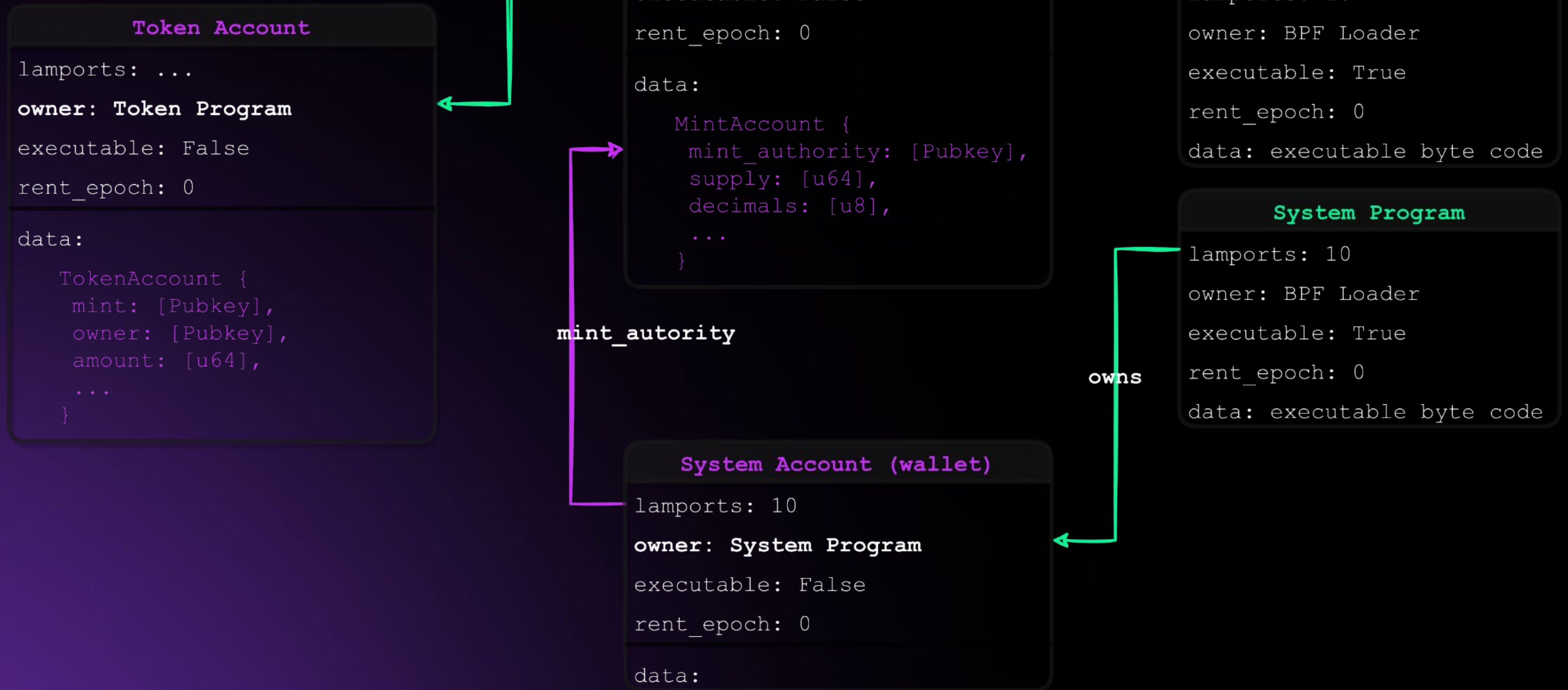
Solana Tokens



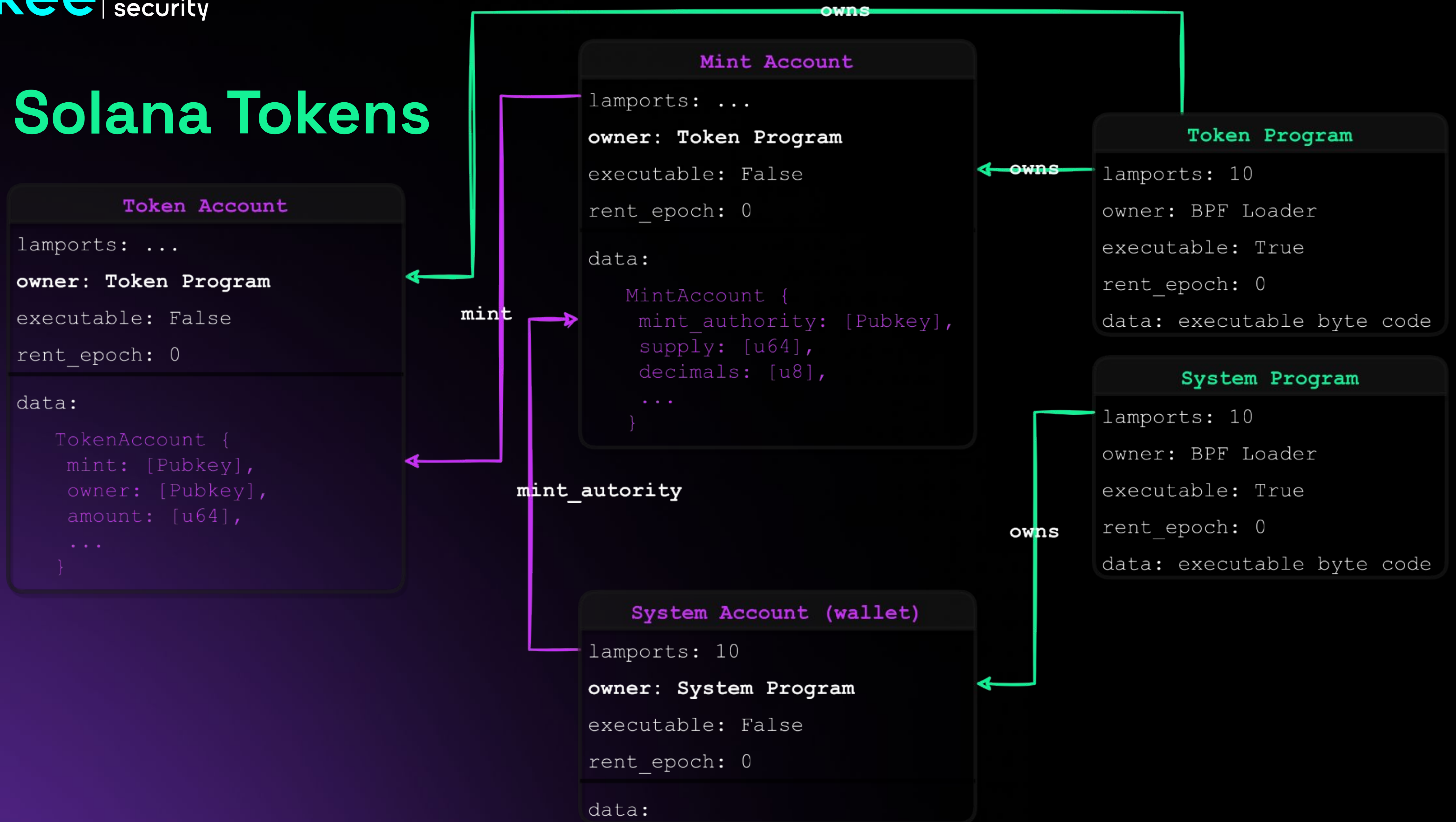
Solana Tokens



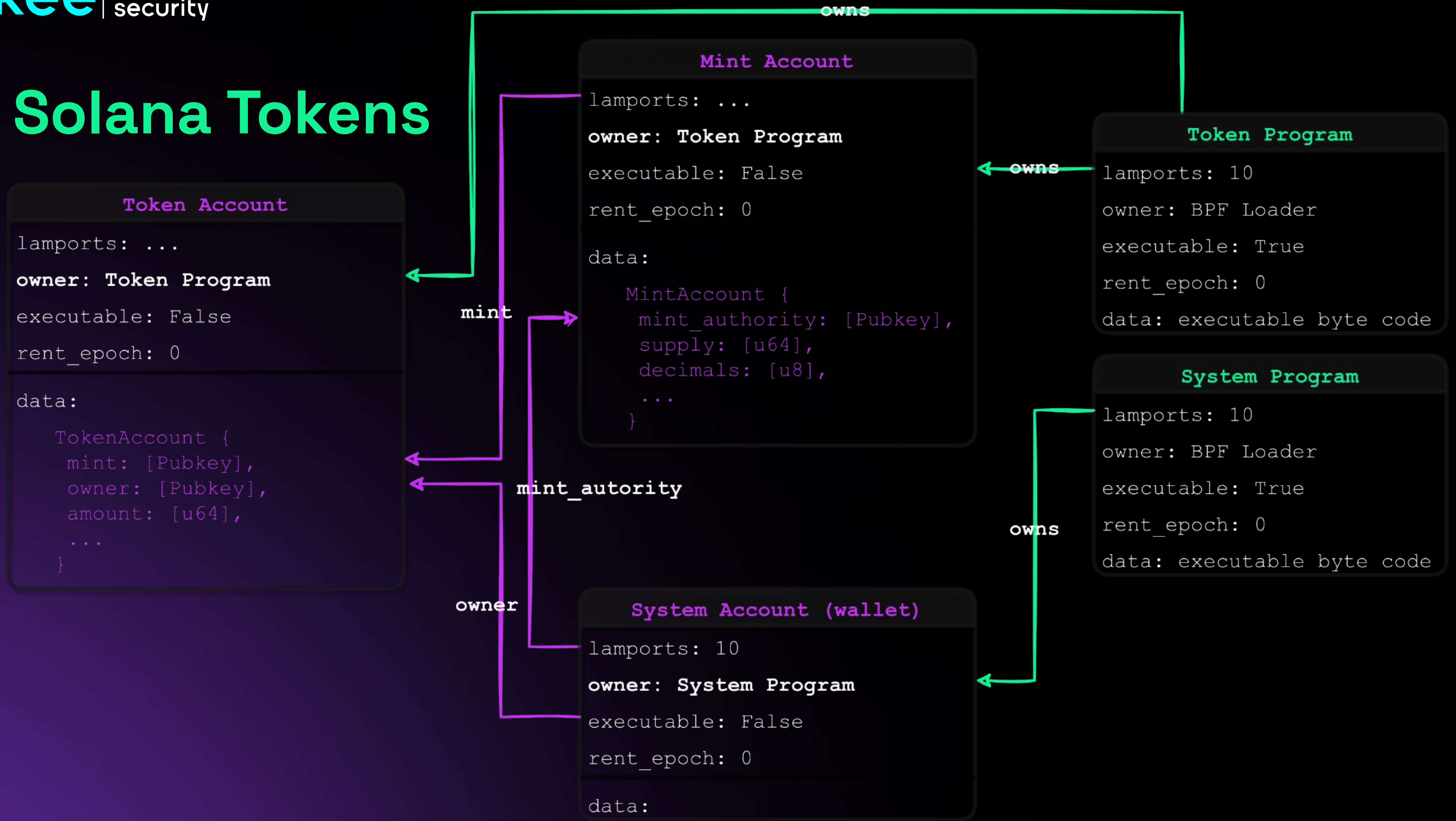
Solana Tokens



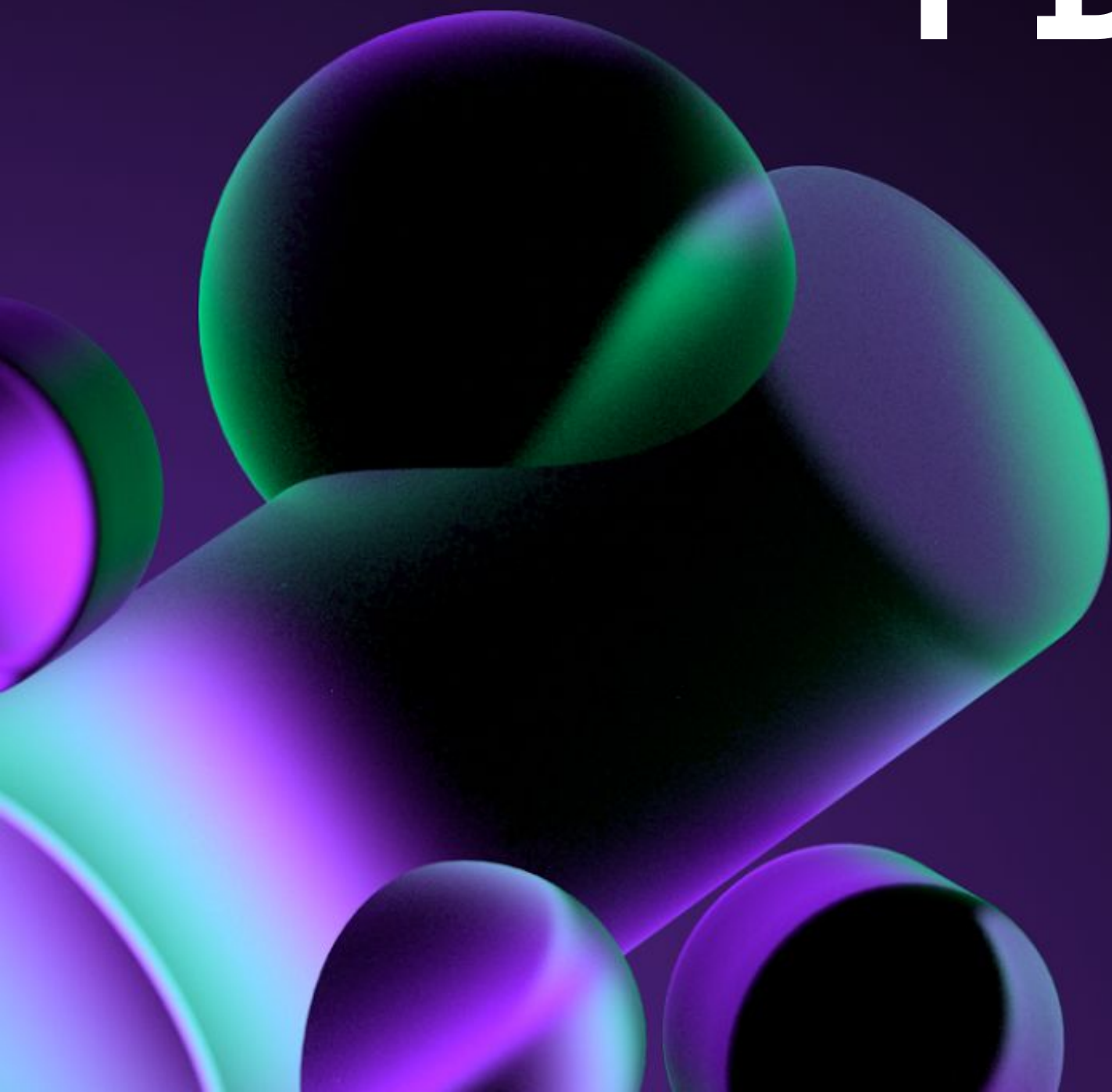
Solana Tokens



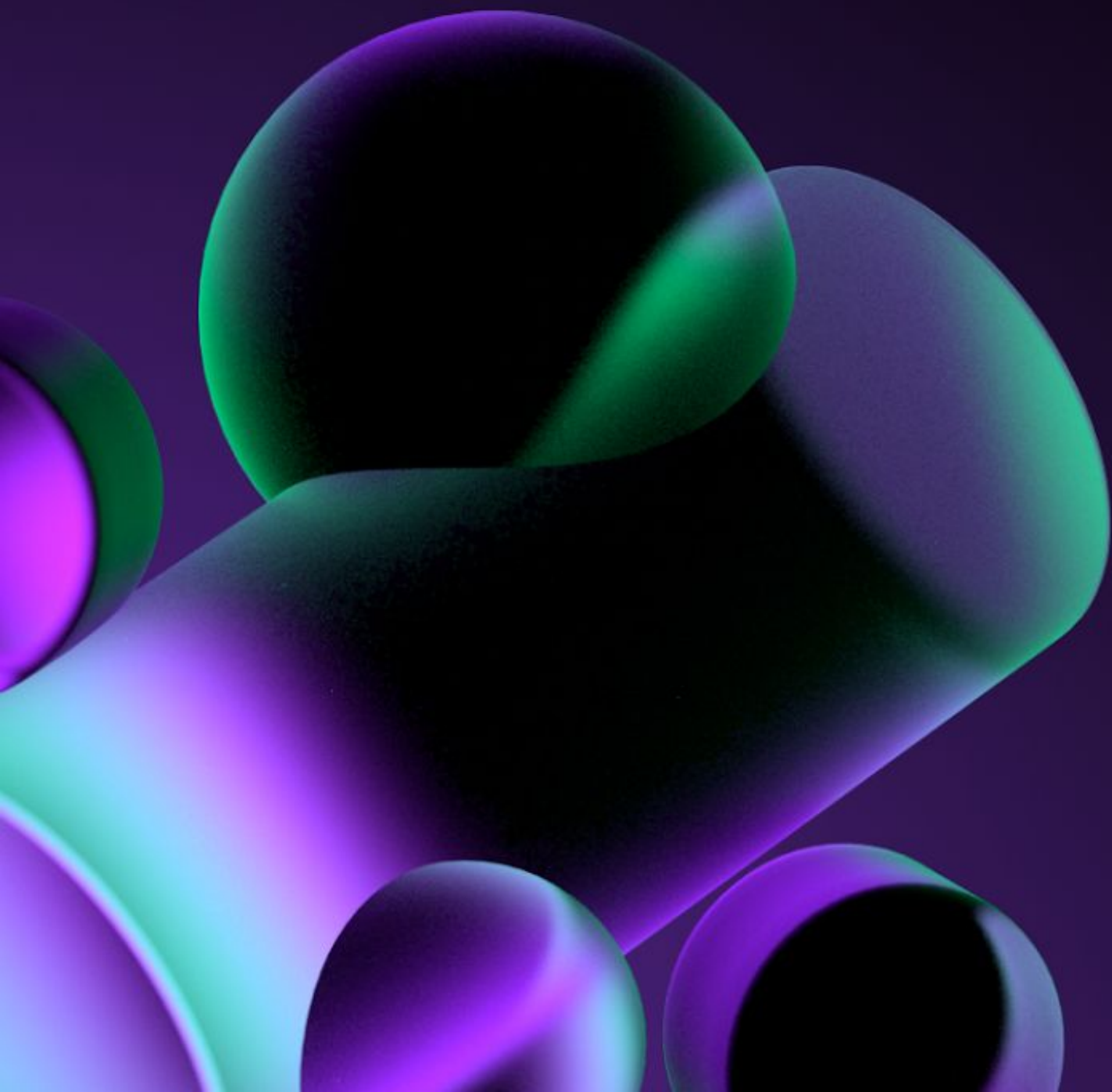
Solana Tokens



PDA Hands on example



Task 4





Thank you

See you next time!