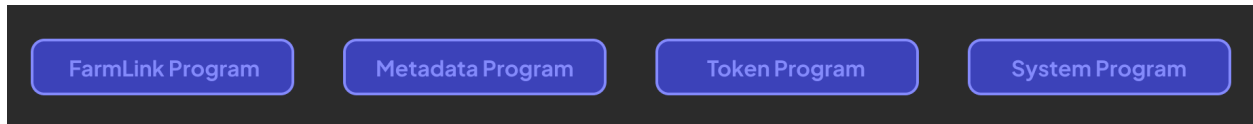


FarmLink Architecture Diagram

Program Boxes



FarmLink Program

The FarmLink Program will be the program responsible for managing the overall operations of the FarmLink marketplace. This includes creating and maintaining listings, handling user interactions (e.g., purchases, cancellations), and interacting with other programs for tasks such as token transfers and metadata updates. It acts as the orchestrator of all processes.

Metadata Program

The Metadata Program will be the program responsible for managing the metadata of assets, particularly tokens. This includes reading, updating, or verifying the metadata associated with a product being listed on the marketplace. It ensures that the assets retain their descriptive attributes (e.g., name, image, collection) during and after transactions.

Token Program

The Token Program will be the program responsible for managing token transfers and ownership. It handles the minting, transferring, and burning of fungible and non-fungible tokens. This program ensures the secure movement of tokens between accounts (e.g., Farmer ATA to Vault Account) during marketplace transactions.

System Program

The System Program will be the program responsible for creating new accounts and PDAs, and for managing lamports (Solana's native token) in the context of rent and fees. It is fundamental to initializing and maintaining the accounts needed for marketplace operations.

PDA Boxes



FarmLink PDA

The FarmLink PDA will be a program-derived address representing a marketplace. It will be used to store global configuration data for the marketplace, such as platform settings, commission rates, or system-wide parameters.

Treasury PDA

The Treasury PDA will be a program-derived address used to manage the marketplace's treasury. It will store funds collected from marketplace fees, and only the Marketplace Program will have the authority to withdraw or distribute these funds.

Metadata PDA

The Metadata PDA will be a program-derived address linked to an individual product or asset. It will store metadata information such as the product name, description, and any associated attributes that define the item within the marketplace.

Rewards PDA

The Rewards PDA will be a program-derived address used to track and manage user rewards or incentives. It may be used to store data on reward points or bonuses for users participating in the marketplace (e.g., frequent buyers or sellers).

Product PDA

The Product PDA will be a program-derived address representing an individual product listing on the marketplace. It will store data specific to the listing, such as the price, owner (Maker), status (active or sold), and any associated metadata or conditions for sale.

Account Details

1. User-Related Accounts

Farmer Account(maker)

- Represents the user listing an item on the FarmLink marketplace.
- Not directly managed by the program but required as a "signer" to authorize operations.

Consumer Account(taker)

- Represents the user purchasing a listed agricultural product.
- It is only a "signer" to authorize fund or token transfers.

Farmer Associated Token Account(maker ATA)

- Holds the farmer's tokens being listed.
- Managed by the Token Program.

Consumer Associated Token Account(consumer ATA)

- Receives the token acquired by the consumer after the transaction.
- Managed by the Token Program.

2. Program-Related Accounts

Product Account

- Stores information about a listed product (like name, description, category, unit, total amount, etc).
- Created and managed by the FarmLink Program.

Vault Account

- Temporary account used to hold the token listed by the farmer.
- Managed by the Token Program and "owned" by the Vault PDA.

3. PDA-Related Accounts

FarmLink PDA Account

- Represents global settings for the platform, such as commission fees and general parameters.
- Only the FarmLink Program can modify or access it.

Treasury PDA Account

- Manages funds collected as marketplace fees.
- Contain lamports or tokens accumulated from completed transactions.

Metadata PDA Account

- Stores detailed information about the product (like name, description, category, unit, total amount, etc).
- Associated with the Metadata Program.

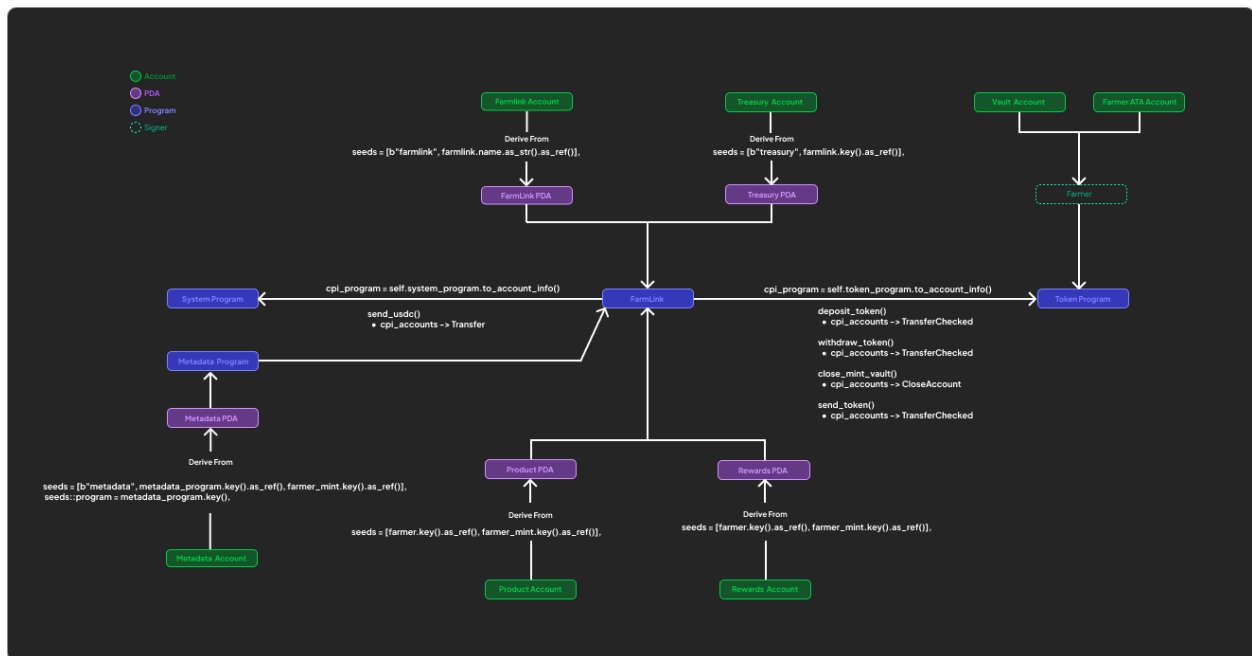
Rewards PDA Account

- Used to track rewards or incentives for users.
- May store data such as scores or reward balances.

Product PDA Account

Represents an individual listing on the marketplace.

Stores specific listing data such as price, status, and connection to the Product Account.



Instructions

+ `initialize(ctx: Context<>, name: String, fee: u16)`

Responsible to initialize marketplace with the name and fee

+ `create_product(ctx: Context<>, price: u64)`

Responsible to create a new agricultural product and list in the marketplace

+ `purchase_product(ctx: Context<>)`

Responsible to buy an agricultural product and remove from marketplace

+ deliver_product(ctx: Context<>)

Responsible to pay the farmer when the consumer receives the product

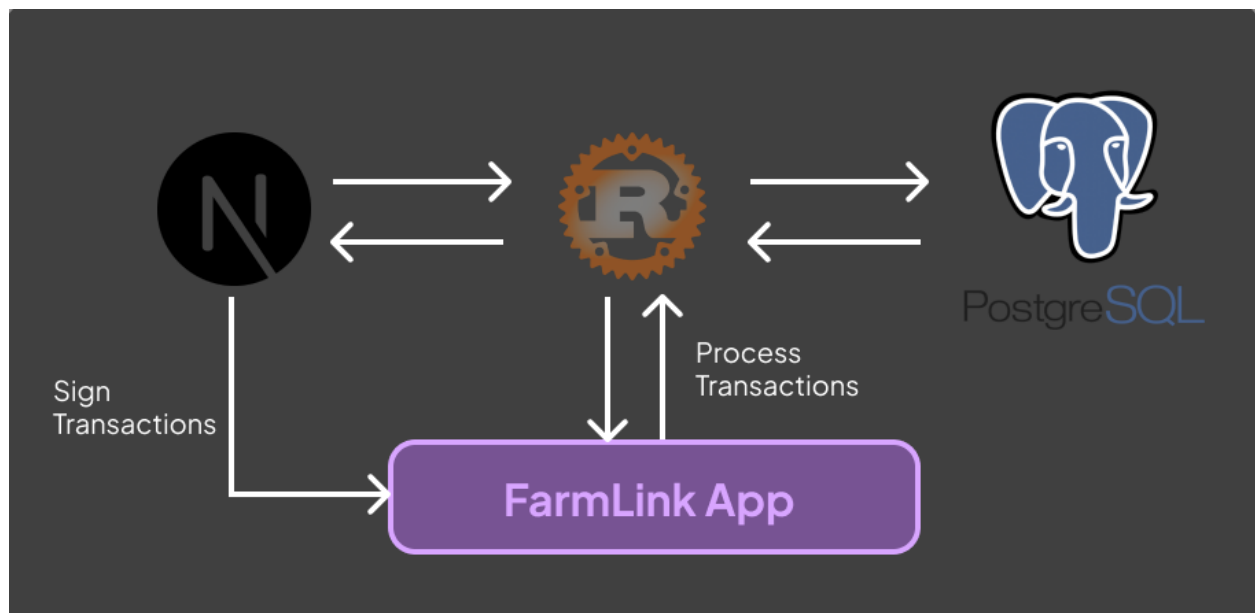
+ refund_consumer(ctx: Context<>)

Responsible to pay back when the consumer didn't receive the product

+ refund_farmer(ctx: Context<>)

Responsible to pay back the fees spent by the farmer

External Dependencies



Next.js(frontend)

- Where users will sign transactions
- Interacts with FarmLink, that actually processes transactions
- Interacts with Axum

Rust/Axum(backend)

- Handle the requests
- Receive data and store data in database

PostgreSQL(database)

- Stores user data to display
- Stores transaction data to display the history
- Stores products to display

Docker

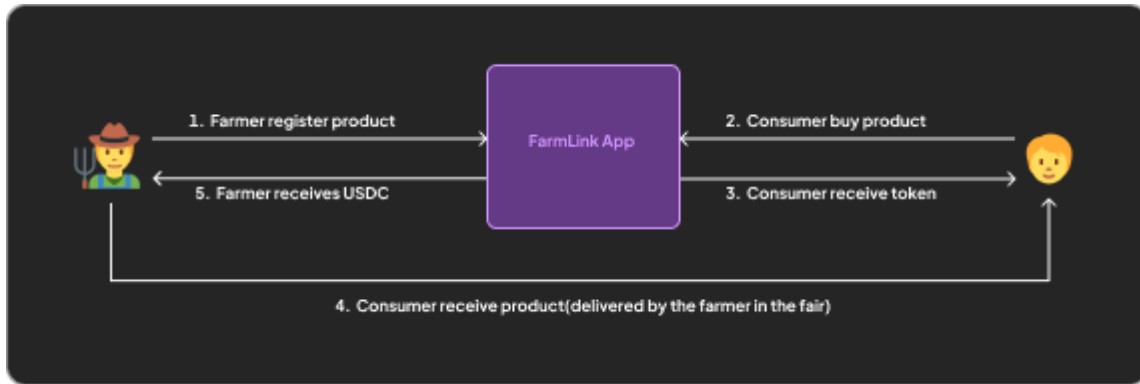
- Will containerize and run all this 3 elements in a modular way

Future plans

- Deploy frontend in Vercel
- Deploy backend in AWS EC2
- Deploy database in AWS Aurora

Flow Diagram

Overall Flow

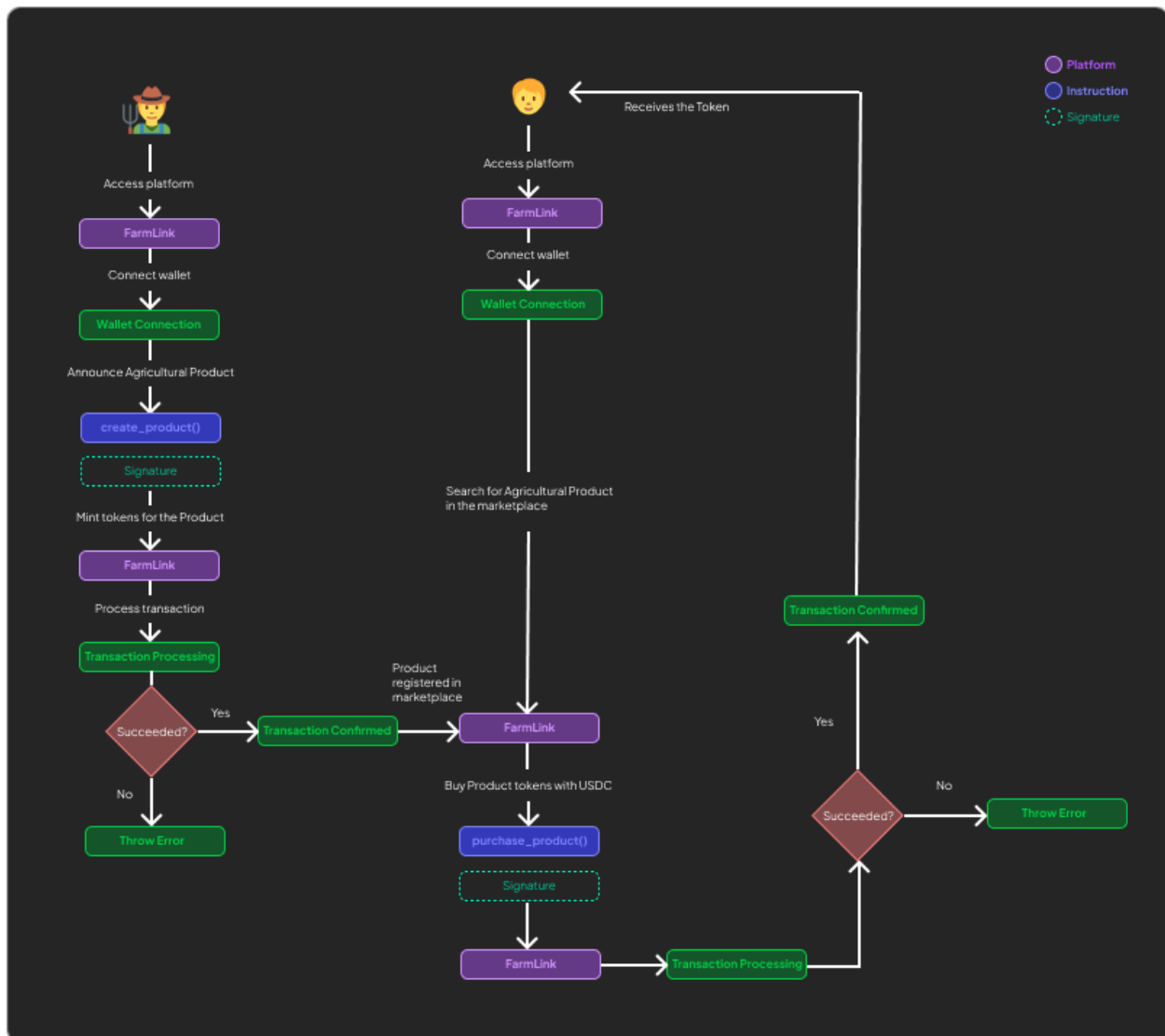


1. The farmer should register an agricultural product
 - Fungible token creation
 - Storage in the Vault
2. The consumer should buy an agricultural product
 - USDC storage in the Vault
3. Consumer should receive token from the product bought
 - Fungible token go to consumer's wallet
4. The farmer should deliver the product
5. The farmer should receive the payment
 - USDC go to farmer's wallet
 - Token get burn from consumer's wallet

Create Product(Farmer) and Buy Product(Consumer)

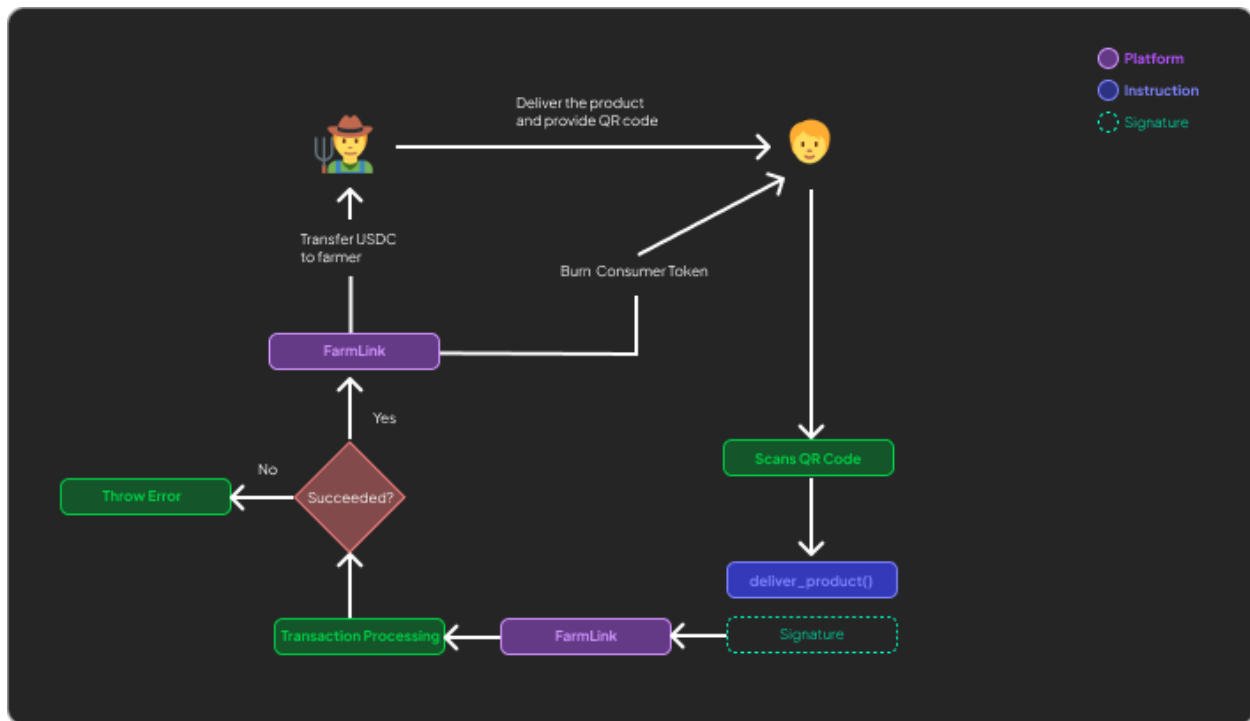
In this flow, the farmer will access the platform, and fill the fields in the frontend that will represent the token metadata of the product to be minted, by doing that the product becomes available in the marketplace.

After that, the consumer access the platform, search for the product and click to buy it, by doing that he will sign the transaction and receive the tokens minted by farmer(that was in the vault) and send the money into the vault.



Deliver Product

In this flow, the farmer and the consumer will meet in a fair, the farmer will deliver the product and provides a QR Code to the consumer, by scanning that, the money that was stored in the vault goes to the farmer's wallet.



Refund

In this flow, the farmer do not show up in the fair, by doing that, the consumer will sign a transaction to receive the funds back and burn the tokens in the wallet.

If the consumer does not show up, the platform will pay back the fee used to the product creation by using the consumer money in the vault. Consumer's tokens is burned and the money goes back into consumer's wallet.

