ANASTASIA LABS

Project Design Specification

Project Number	1000013
Project manager	Philip DiSarro
Date Started	February 24, 2024
Date Completed	...

Contents

Payment Subscription Smart Contract

1. Overview

This Payment Subscription Smart Contract is developed using Aiken to facilitate automated recurring payments between Subscribers and Merchants on the Cardano blockchain. This smart contract enables users to set up, manage, and cancel subscriptions directly from their wallets.

2. Architecture

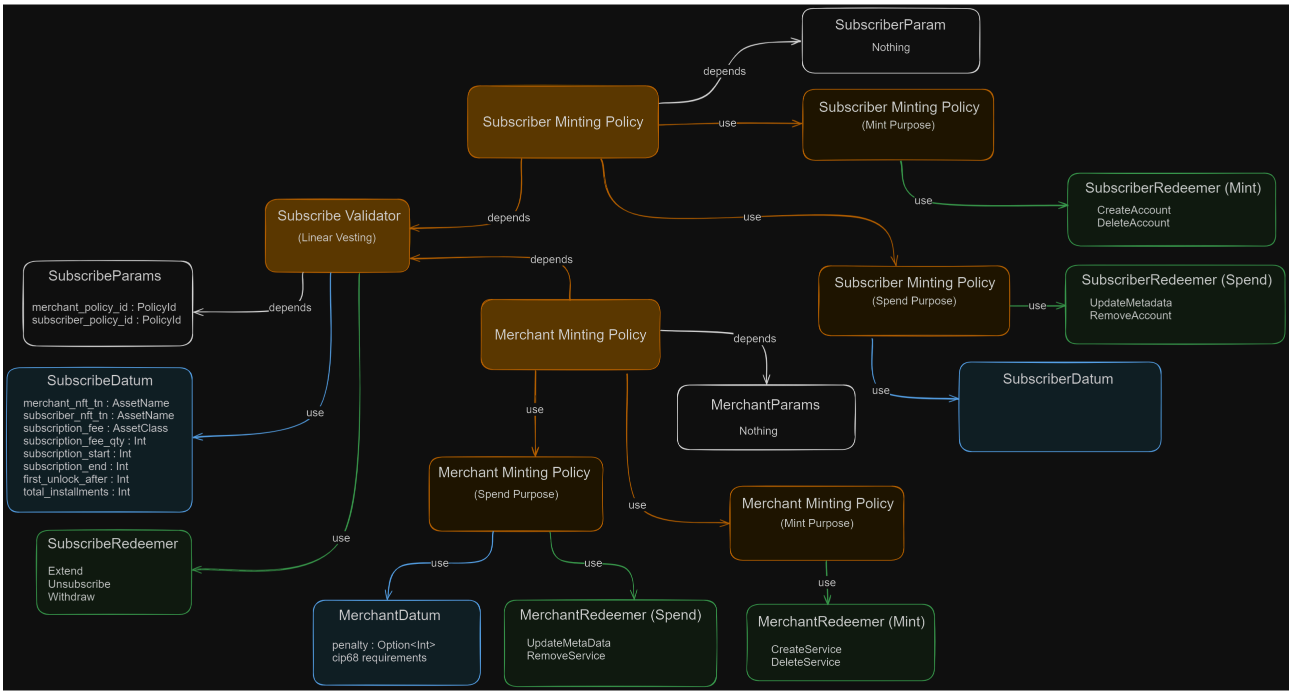

Figure 1: Contract Architecture

There are three contracts in this subscription system.

- **Merchant Contract:** A multi-validator responsible for creating an initial service by minting a single CIP-68 compliant MerchantNFT and sending it to the merchant while sending the reference NFT to the Subscribe Contract. It also updates the metadata for the merchant and deletes the service by burning the MerchantNFT.
- **Subscriber Contract:** A multi-validator responsible for creating the initial subscription to a service by minting a SubscriberNFT and sending it to the user, while sending the reference NFT to the Subscribe

Contract. It also updating the metadata for the subscriber and deletes the user account by burning a SubscriberNFT.

- **Subscribe Contract:** Responsible for holding the prepaid subscription fees for a service, renewing a subscription to a service, unsubscribing from a service and withdrawing subscription fees.

3. Specification

3.1. System Actors

- **Merchant:** An entity who interacts with the Merchant Contract in order to create a service and receives subscription payments for the respective service or services.
- **Subscriber:** An entity who interacts with the Subscriber Contract in order to create an account and deposit prepaid subscription fees to the Subscribe Contract.

3.2. Tokens

- **Merchant NFT:** Can only be minted by a merchant when creating a subscription service and burned when merchant removes their service/services from the system. Datum is updated when a subscription is paid or fees are withdrawn from Subscribe Contract.
  - Policy Id: Merchant Minting Policy
  - TokenName: Defined in Merchant Minting Policy parameters with Merchant OutputReference
- **Subscriber NFT:** Can only be minted when a subscription fee is paid to Subscribe Contract and burned when subscriber exits the system. Datum is updated when fees are deposited and withdrawn from Subscribe Contract.
  - Policy Id: Subscriber Minting Policy
  - TokenName: Defined in Subscriber Minting Policy parameters with Subscriber OutputReference

3.3. Smart Contract

3.3.1. Subscribe Validator

Subscribe validator is responsible for holding subscription fees and validating subscriptions.

3.3.1.1. Parameters

- `merchant_policy_id` : Hash of the PolicyId

- `subscriber_policy_id` : Hash of the PolicyId

3.3.1.2. Datum

- **`merchant_nft_tn`:** Merchant's token name encoding UTxO to be consumed when minting the NFT.
- **`subscriber_nft_tn`:** Subscriber's token name encoding UTxO to be consumed when minting the NFT.
- **`subscription_fee`:** AssetClass type for the subscription fee.
- **`subscription_fee_qty`:** Amount of the subscription fee.
- **`subscription_start`:** Start of the subscription.
- **`subscription_end`:** Expiry time of the subscription.
- **`total_installments`:** The number of periodic intervals over which to release subscription fees.

3.3.1.3. Redeemer

- Extend
- Unsubscribe
- Withdraw

3.3.1.4. Validation

- **Extend:** The redeemer will allow anyone with a subscriberNFT to spend Subscribe UTxO to unlock subscription fees to merchant address beyond the previous `subscription_end` date.
  - validate that there is a single UTxO in the transaction input and contains a single subscriberNFT token.
  - validate that the transaction must execute after `subscription_end`.
  - validate the correct `subscription_fee` Asset and quantity is sent to the merchant address.
- **Unsubscribe:** The redeemer will allow anyone with a subscriberNFT to spend Subscribe UTxO to unlock funds back to their address.
  - validate that there is a single UTxO in the transaction input and contains a single subscriberNFT token.
  - validate that transaction withdraws subscription fee to the subscriber address.

- **Withdraw:** The redeemer will allow anyone with a subscriberNFT or merchantNFT to spend Subscribe UTxO to unlock funds to the merchant address.
  - validate that there is a single UTxO in the transaction input and contains a single merchantNFT token.
  - validate that transaction withdraws subscription fee to the merchant address.
  - validate that transaction withdraws penalties to the merchant address.

3.3.2. Merchant Minting Policy

Merchant Minting Policy is responsible for registering a service creating, updating and removing a service for a merchant.

3.3.2.1. Parameter

Nothing

3.3.2.2. Minting Purpose

3.3.2.2.1. Redeemer

- CreateService
- RemoveAccount

3.3.2.2.1.1. Validation

- **CreateService:** The redeemer allows creating of a new subscription sevice by minting only one unique Token.
  - validate that out_ref must be present in the Transaction Inputs
  - validate that the redeemer only mints a single CIP68 compliant merchant Token
- **RemoveAccount:**
  - validate that the redeemer only burns a single CIP68 compliant merchant NFT Token

3.3.2.3. Spend Purpose

3.3.2.3.1. Datum

- penalty_fee
- penalty_fee_qty
- cip-68 requirements

3.3.2.3.2. Redeemer

- UpdateMetaData
- RemoveService

3.3.2.3.2.1. Validation

- **UpdateMetaData:** The redeemer allows for updating the metadata attached to the UTxO sitting at the script address.
    - validate that there is a single UTxO in the transaction input and contains a single merchantNFT token.
    - updates the metadata of the Reference NFT token and sends the token back to the Subscribe Contract.
- **RemoveService:** The redeemer allows the removal of a service by a merchant from the subscription system.
    - validate payment signature is equal to merchant payment signature.
    - validate that there is a single UTxO in the transaction input and contains a single merchentNFT token.
    - validate that unlocked funds are sent to the merchant address.
    - Removes all the Reference NFT tokens to another external address.

3.3.3. Subscriber Minting Policy

3.3.3.1. Parameter

Nothing

3.3.3.2. Minting Purpose

3.3.3.2.1. Redeemer

- CreateAccount

- DeleteAccount

3.3.3.2.1.1. Validation

- **CreateAccount:** The redeemer allows creating of a new subscription service account by minting only one unique Token.
  - validate that out_ref must be present in the Transaction Inputs
  - validate that the redeemer only mints a single CIP68 compliant SubscriberNFT Token
- **DeleteAccount:**
  - validate that the redeemer only burns a single CIP68 compliant SubscriberNFT Token

3.3.4. Spend Purpose

3.3.4.1.1. Datum

- cip-68 requirements

3.3.4.1.2. Redeemer

- UpdateMetaData
- RemoveAccount

3.3.4.1.2.1. Validation

- **UpdateMetaData:** The redeemer allows for updating the metadata attached to the UTxO sitting at the script address.
  - validate that there is a single UTxO in the transaction input and contains a single SubscriberNFT Token.
  - updates the metadata of the Reference NFT token and sends the token back to the address entity executing it.
- **RemoveAccount:** The redeemer allows the removal of an account by a subscriber from the subscription system.
  - validate payment signature is equal to subscriber payment signature.

- validate that there is a single UTxO in the transaction input and contains a single SubscriberNFT Token.
- validate that unlocked funds are sent to the subscriber address
- validate that penalty is calculated and fees are sent to the merchant address
- Removes all the Reference NFT tokens to another external address.