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

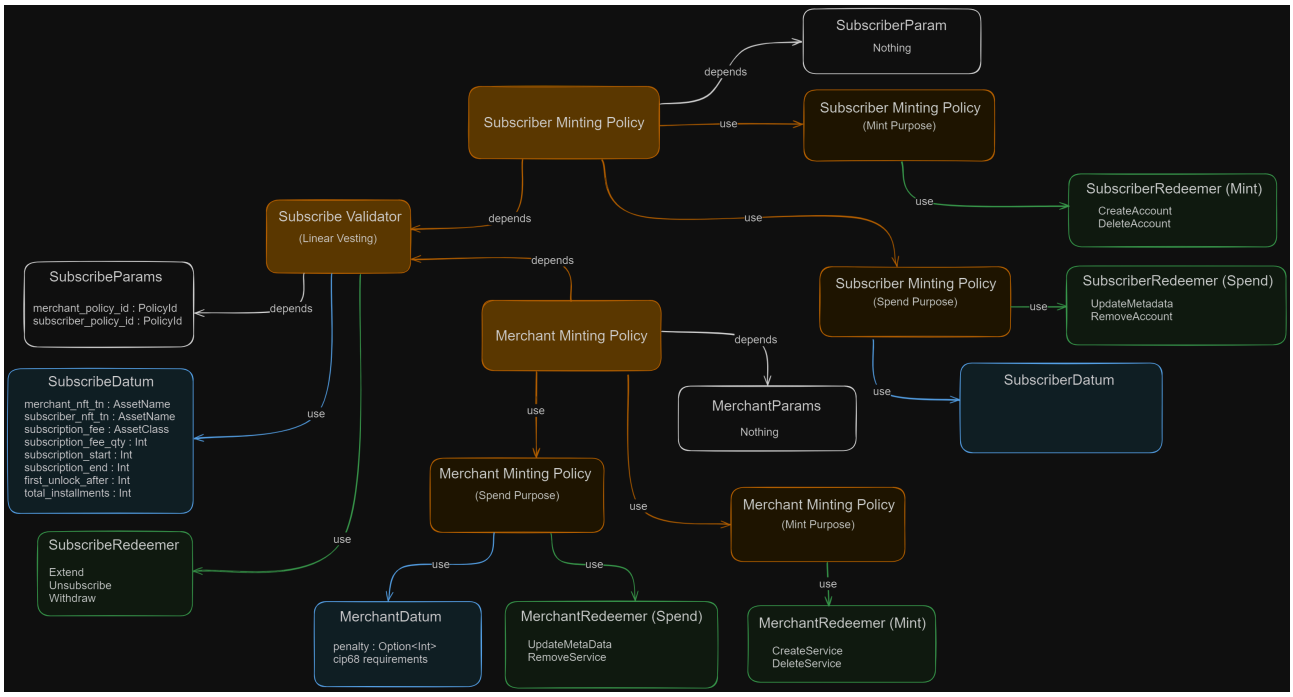

Figure 1: Contract Architecture

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 spending end point. 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 spending

endpoint. 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. This could also be a multi-validator to authenticate the UTxO.

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 the merchant withdraws from the Subscribe Contract.
  - TokenName: Defined in Merchant Minting Policy parameters with the hash of the Merchant Minting Policy 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.
  - TokenName: Defined in Subscriber Minting Policy parameters with hash of the Subscriber Minting Policy 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

This is a Sum type datum where one represents the main datum and the other one represents a penalty datum.

3.3.1.2.1. Main 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.

3.3.1.2.2. Penalty datum

- **`merchant_nft_tn`:** Merchant's token name encoding UTxO to be consumed when minting the NFT.

3.3.1.3. Redeemer

- Extend
- Unsubscribe
- Withdraw

3.3.1.4. Validation

- **Extend:** The redeemer will allow anyone to increase the subscription funds.
  - validate that the value of the UTxO is increased as long as the Datum is updated with the Merchant Token Name.

- **Unsubscribe:** The redeemer will allow anyone with a subscriberNFT to spend Subscribe UTxO to unlock funds back to their address.
  - validate the subscriberNFT is being spent.
  - validate that the penalty UTxO is being produced with the merchants Token Name.
- **Withdraw:** The redeemer will allow anyone with a merchantNFT to withdraw funds from the Subscribe contract
  - validate merchantNFT is being spent
  - validate whether the transaction contains a penalty datum or a normal datum.

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`: AssetClass type for the amount of fees to be deducted when subscriber cancels the subscription.
- `penalty_fee_qty`: Amount of the penalty fees.
- 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 merchantNFT is being spent.
  - updates the metadata of the Reference NFT token and sends the token to the spending end point
- **RemoveService:** The redeemer allows the removal of a service by a merchant from the subscription system.
  - validate merchentNFT is being spent.
  - 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 SubscriberNFT is being spent.
  - updates the metadata of the Reference NFT token and sends the token to the spending end point.
- **RemoveAccount:** The redeemer allows the removal of an account by a subscriber from the subscription system.
  - validate that SubscriberNFT is being spent.
  - validate that unlocked funds are sent back to the subscriber address
  - validate that penalty is calculated accurately and fees are in the penalty UTxO
  - Removes all the Reference NFT tokens to athe spending endpoint.