ANASTASIA LABS

# Upgradable Multi-Signature Contract

# Project Design Specification

# Contents

## 1. Overview

This document outlines the design specification for an upgradable multisignature (multisig) smart contract implemented using Aiken for the Cardano blockchain. The contract enables authorized members to execute asset transactions within predefined thresholds, demonstrates secure spending of assets, allows seamless adjustment of signer thresholds, and supports dynamic addition or removal of signers for enduring usability and adaptability.

## 2. Architecture

There are one main contracts in this multi-sig system.

1. **Multisig Validator**

   This is the core contract responsible for handling the logic for:

   - Managing authorized signers
   - Validating and executing transactions
   - Handling contract upgrades
   - Verifying signatures and authenticating transactions

# 3. Specification

## 3.1. System Actors

- **Signers**

  Entities who are authorized to sign transactions, participate in the management of the multi-sig wallet, and approve changes to the signer list or thresholds.

- **Initiator**

  The entity who creates the initial multi-sig wallet setup, defining the initial set of signers and thresholds.

## 3.2. Tokens

- None

## 3.3. Smart Contracts

### 3.3.1. Multi-sig validator

The Multi-sig Contract is the primary contract responsible for managing the list of authorized signers, validating transactions, and ensuring the proper execution of multi-sig operations.

#### 3.3.1.1. Parameters

- None

#### 3.3.1.2. Spend Purpose

The contract uses the Spend purpose, allowing it to manage and spend funds locked in its address.

#### 3.3.1.3. Datum

The datum structure holds the current state of the multisig arrangement:

- **`signers`:** List of public key hashes of authorized signers.
- **`threshold`:** Minimum number of required signatures.
- **`funds`:** AssetClass of the funds to be withdrawn.
- **`funds_qty`:** The total amount of funds controlled by the contract
- **`spending_limit`:** Max Amount of funds to be withdrawn per transaction.

#### 3.3.1.4. Redeemer

The contract supports two types of operations, represented by the redeemer:

- Sign: For executing fund transfers
- Update: For modifying the multisig configuration

#### 3.3.1.5. Validation

1. **Sign**

The redeemer allows a majority of the authorized signers to collectively approve and execute transactions using the funds controlled by the multi-signature contract

- Verifies that the required number of authorized signers have signed the transaction
- Ensures the transfer amount does not exceed the spending limit
- Checks that the total value is preserved across inputs and outputs
- Ensure the output datum matches the input datum (no changes to the multisig configuration)

2. **Update**

The redeemer enables the modification of the multi-signature arrangement itself.

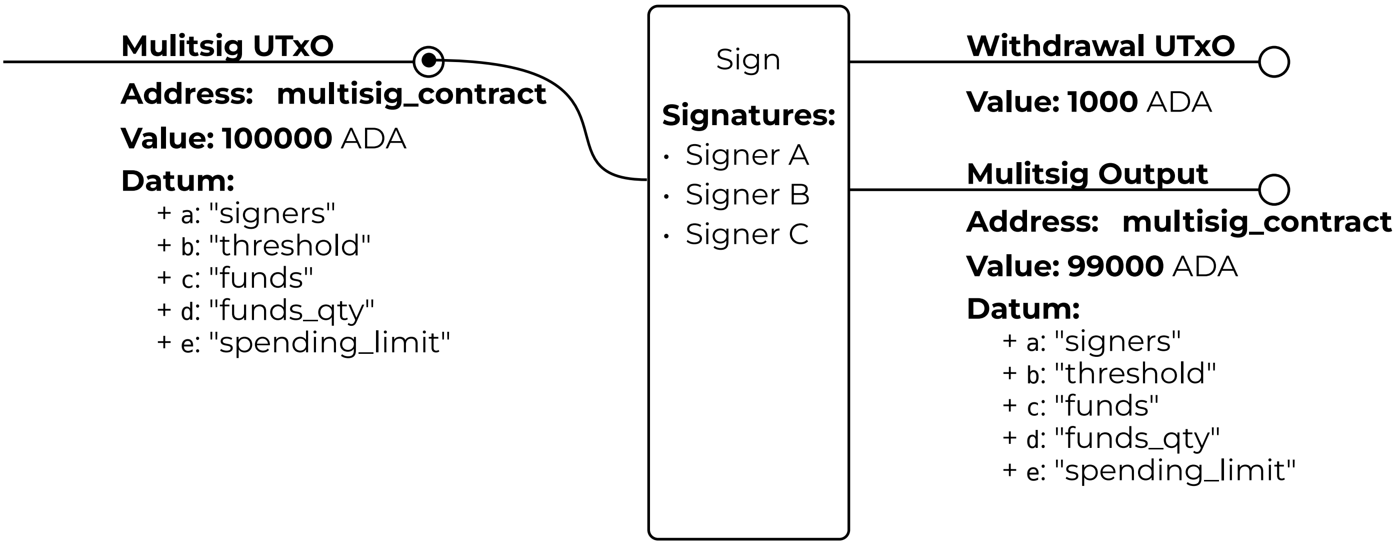
- Verifies that the required number of authorized signers have signed the transaction
- Enforce bounds on new signers list and threshold:
  - New signer count must be greater than 0
  - New threshold must be greater than 0 and less than or equal to the new signer count
  - New funds quantity must be greater than or equal to 0
  - New spending limit must be greater than or equal to 0 and less than or equal to the new funds quantity
- Ensure there are no duplicate keys in the new list of signers
- Verify that input value equals output value (no spending during update)
- Verify that the total value is preserved (input value equals output value, no funds are spent during update)
- Ensure the new configuration is stored in the output datum
- Allows addition or removal of only one signer at a time

# 4. Transactions

This section outlines the various transactions involved in the Upgradable Multi-Signature Contract on the Cardano blockchain.

### 4.1.1. Spend :: Sign

This action ensures that the number of signers meets or exceeds the specified threshold and The datum of the Multisig remains the same.



**Note**: Sign UTxO Diagram

#### 4.1.1.1. Inputs

1. **Multisig Validator UTxO**
   - Address: Multisig validator script address
   - Datum:
     - `signers`
     - `threshold`
     - `funds`
     - `funds_qty`
     - `spending_limit`

.

- Value:
  - ADA + Any tokens

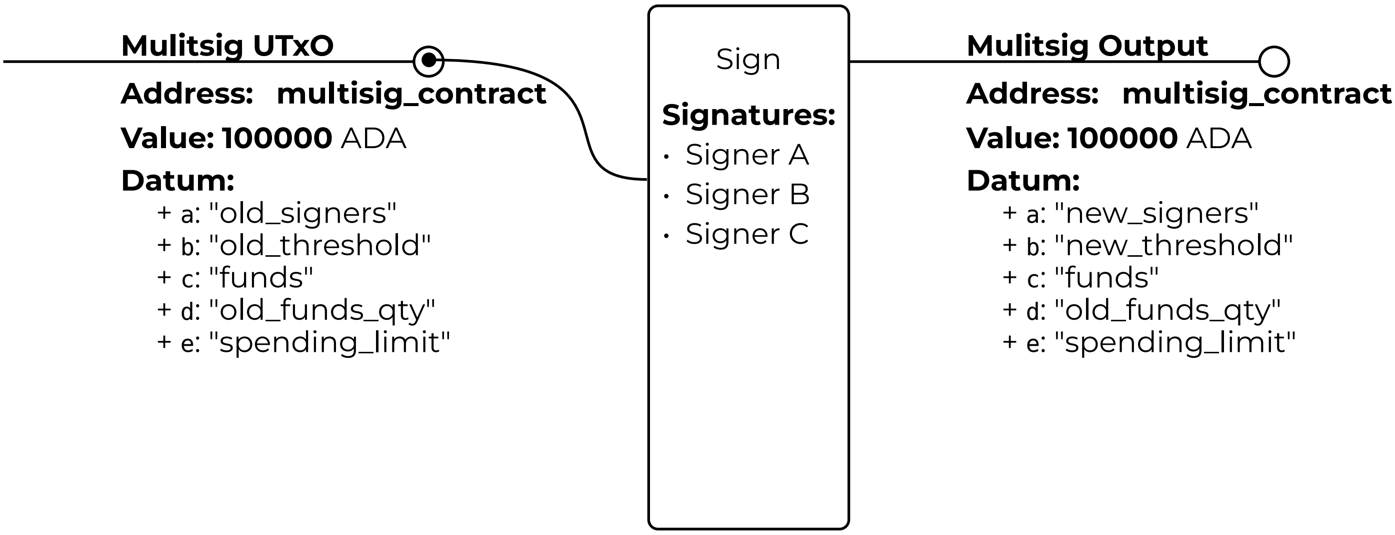
  - Locked Value

#### 4.1.1.2. Outputs

1. **Recipient Wallet UTxO**
   - Address: Recipient wallet address
     - Transferred ADA/tokens
2. **Multisig Validator UTxO:**
   - Address: Multisig validator script address
   - Datum:
     - `signers`
     - `threshold`
     - `funds`
     - `funds_qty`
     - `spending_limit`
   - Value:
     - Remaining ADA + Remaining tokens

#### 4.1.2. Spend :: Update

Allows for the addition or removal of members from the Multisig arrangement, and updates the required signers threshold.



**Note**: Update UTxO Diagram

##### 4.1.2.1. Inputs

1. **Multisig Validator UTxO**
   - Address: Multisig validator script address
   - Datum:
     - `old_signers`
     - `old_threshold`
     - `funds`
     - `old_funds_qty`
     - `old_spending_limit`
   - Value:
     - X ADA + Any tokens

##### 4.1.2.2. Outputs

1. **Multisig Wallet UTxO**
   - Address: Merchant wallet address

- Datum:
  - `new_signers`
  - `new_threshold`
  - `funds`
  - `new_funds_qty`
  - `new_spending_limit`
- Value:
  - X ADA + Any tokens (unchanged)