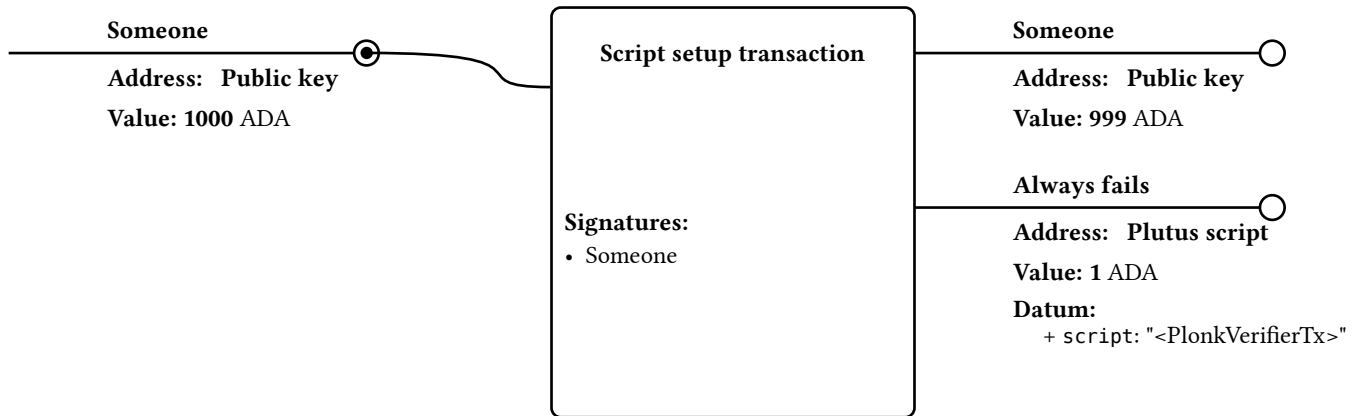# End-to-end test for the PlonkVerifierTx
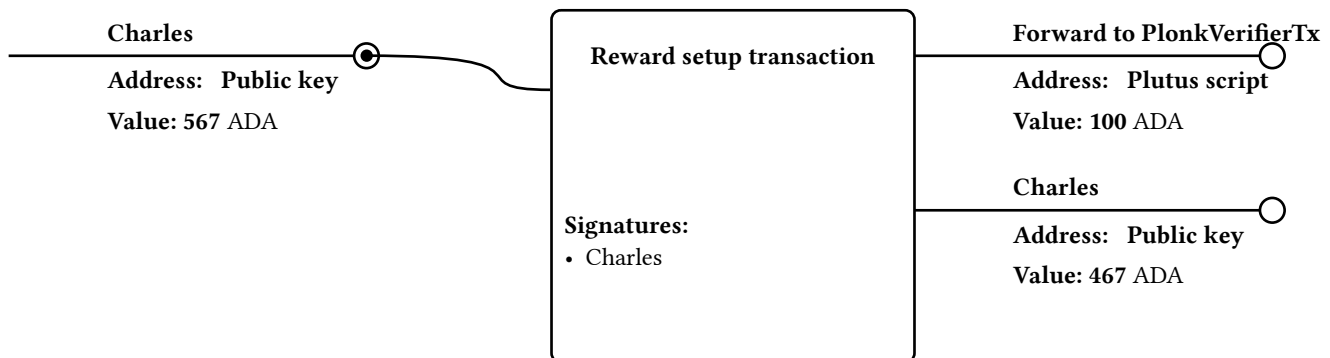
A ZK-KYC scenario: Charles sets up a reward in ada that can be claimed by proving (using the Plonk `verify` algorithm) the completion of a KYC process.

First, we perform a setup transaction that posts the PlonkVerifierTx script on-chain.

Someone

Address: Public key

Value: 1000 ADA

Script setup transaction

Signatures:
• Someone

Someone

Address: Public key

Value: 999 ADA

Always fails

Address: Plutus script

Value: 1 ADA

Datum:
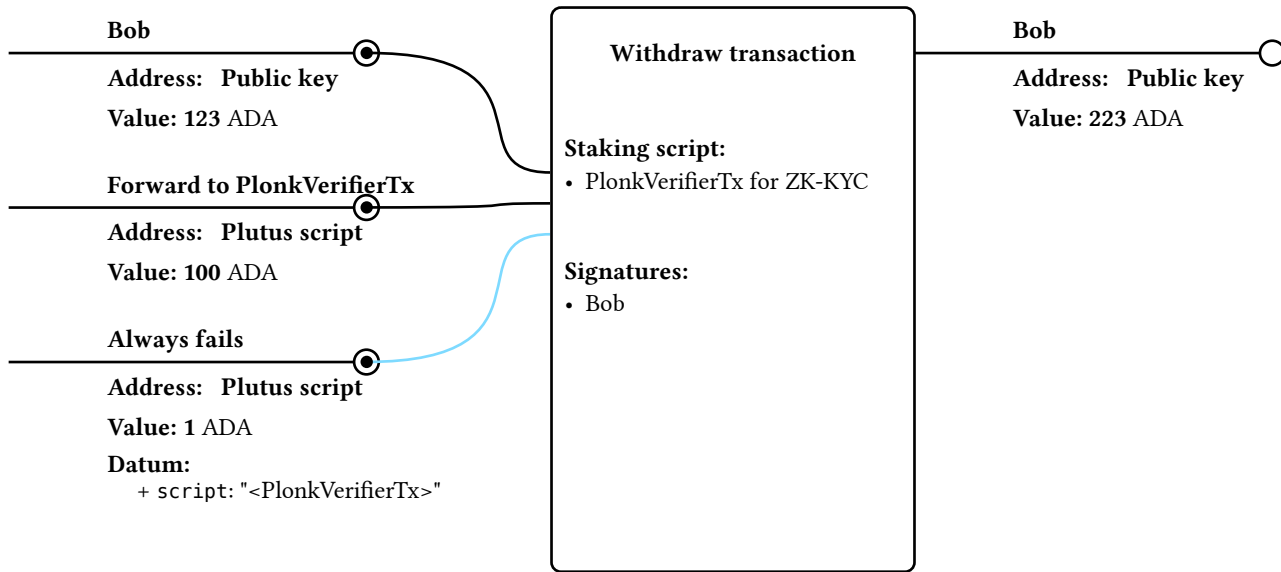+ `script`: "\<PlonkVerifierTx\>"

Note: Someone posts the PlonkVerifierTx script on-chain.

In the second transaction, Charles sets up a reward for doing the ZK-KYC process. He sends 100 ada to a Plutus script address. The script unlocks the funds if and only if the respective PlonkVerifierTx is executed successfully.

Charles

Address: Public key

Value: 567 ADA

Reward setup transaction

Signatures:
• Charles

Forward to PlonkVerifierTx

Address: Plutus script

Value: 100 ADA

Charles

Address: Public key

Value: 467 ADA

Note: Charles sets up a reward for proving completion of a ZK-KYC process.

Bob proves KYC process completion and claims the reward.

Bob

Address: Public key

Value: 123 ADA

Forward to PlonkVerifierTx

Address: Plutus script

Value: 100 ADA

Always fails

Address: Plutus script

Value: 1 ADA

Datum:
  + `script: "<PlonkVerifierTx>"`

Withdraw transaction

Staking script:
• PlonkVerifierTx for ZK-KYC

Signatures:
• Bob

Bob

Address: Public key

Value: 223 ADA

Note: Bob proves completion of the ZK-KYC process and claims the reward.