Zero-Knowledge Proof Trustless P2P Fiat-to-Crypto On-Ramp for Cardano: a progress report

ZkFold

March 17, 2025

Trustless P2P on-ramp is a smart contract app that connects cryptocurrency sellers and buyers. A seller deposits funds into the smart contract and ask for a fiat payment. Once the payment is delivered, the buyer can generate a cryptographic proof and unlock the funds from the smart contract.

1 Smart contract

The *onRamp* smart contract is parametrized by:

```
data OnRampParams = OnRampParams
  { feeAddress      :: Address
  , feeValue        :: Value
  , fiatPubKeyBytes :: BuiltinByteString
  }
```

providing the fee address and value, as well as the public key of the fiat witness certifying the trade's fiat payment.

Each sell offer is sent to a UTxO containing the datum:

```
data OnRampDatum = OnRampDatum
    { paymentInfoHash   :: BuiltinByteString
    , sellPriceUsd      :: Integer
    , valueSold         :: Value
    , sellerPubKeyBytes :: BuiltinByteString
    , buyerPubKeyHash   :: Maybe PubKeyHash
    , timelock          :: Maybe POSIXTime
    }
```

providing the hash of the payment information, the sell price (in USD), the value offered, the seller's, a place holder for the buyer's pub-key hash, and place holder for a time lock serving as a deadline after which the sell order can be closed if no fiat payment was received. Since at the moment of posting the sell offer the buyer is unknown, initially buyer's public key field is set to `Nothing`.

During the P2P trade sequence, there are are three actions that the *onRamp* smart contract deals with:

```
data OnRampRedeemer
  = Update BuiltinByteString  -- Signed updated datum by seller
  -- ^ Update the bid with the buyer's public key hash and the timelock.
  | Claim BuiltinByteString
  -- ^ Buyer claims the value.
  | Cancel
  -- ^ Seller cancels the value selling.
```

1. The `Update` redeemer provides the seller's signature certifying the updated datum with agreed upon buyer's pub-key hash, as well as a deadline for sending out the fiat payment. The UTxO with the sell offer is updated with this datum.
2. The `Claim` redeemer provides the fiat witness' signature certifying fiat payment.
3. The `Cancel` redeemer allows the seller to cancel the operation after the deadline is reached.

The smart contract performs following checks (depending on the redeemer):

1.1 Update

- The new UTxO with the updated datum contains the original value and does not leave the *onRamp* script.
- The payment info, the sell price, and the seller's public key on the datum have not been altered.
- A buyer's pub-key-hash has been provided.
- A deadline time-lock in the future has been provided.
- The updated datum has been signed by the seller.

1.2 Claim

- The value is sent to the address corresponding to the buyer's pub-key hash registered on the datum of the UTxO being consumed.
- The value sent to buyer corresponds to the whole value of the consumed UTxO.
- Cryptographic proof of fiat payment is provided.
- The correct fee is sent ot the fee address.

In the current version, cryptographic proof of fiat payment is represented by the fact that the fiat witness has signed the payment info (hash of). This can be replaced with a ZK proof of the fiat payment.

1.3 Cancel

- The time-lock deadline for the UTxO (sell offer) being canceled is in the past.
- The value is sent to the original seller's address.

2 Off-chain code (backend)

The main logic of the off-chain code is written in Haskell and resides on the *backends* directory. Executables are named after the subdirectories within.

It is assumed that the secret and public keys of the fee address wallet, the seller's wallets, and the buyer wallet, have been set-up.

Module `src/ZkFold/Cardano/P2P/Example.hs` contains our working example for a function mocking the payment information simply as the hash of a string (wallet's name). It also contains a working example of a "choice value" function needed to have a notion of "best sell-offer".

2.1 `p2p-init-transaction`

Takes two arguments: a) name of wallet receiving fees, and b) fee address. For simplicity, it is assumed that the public key of the fiat witness corresponds to the fees wallet.

Executable serializes the parameterized plutus smart contract. It also writes the cbor for the Cancel redeemer.

2.2 p2p-add-seller

Takes four arguments: a) name of seller's wallet, b) sell price, c) lovelace sold, and d) deadline. For simplicity, it is assumed that the seller's wallet name identifies the file containing the public key as in `name.vkey`.

Executable constructs and writes the initial datum for the sell order.

2.3 p2p-choose-offer

Takes as argument the string representation of the UTxO's at the *onRamp* script. This string can be generated with `cardano-cli` with:

```
onRampUtxos=$(cardano-cli conway query utxo --address $onRampAddr --testnet-magic $mN --out-file /dev/stdout |
                jq -c 'to_entries')
```

(Here shell variables `onRampAddr` and `mN` store the script's address and the chain's "magic number", respectively.)

Executable parses `onRampUTxos` and chooses a "best" sell-offer.

2.4 p2p-buy-order

Takes four arguments: a) name of buyer's wallet, b) buyer's address, c) name of seller's wallet, and d) string representation of seller's UTxO.

Executable signs the updated datum, containing the buyer's pub-key hash, with the seller's private key. It also generates and writes the updated datum and "Update" redeemer in cbor format.

3 p2p-claim-transaction

Takes two arguments: a) name of fiat witness' wallet and b) name of seller's wallet.

Executable signs payment information (associated with the seller's name) with the private key of the fiat witness. It then generates and writes the corresponding Claim redeemer in cbor format.

4 End-to-end testing

An end-to-end trading scenario is tested with shell scripts described in what follows. These shell scripts put together the backend Haskell executables described in the previous section and build and submit transactions using the `cardano-cli` utility.

We have provided a testing scenario involving five wallets:

1. Fiat administrator (*Alice*)
2. Sellers (*Barbara*, *Bob*, *Brandon*)
3. Buyer (*Charlie*)

4.0.1 Setup first wallet (Alice)

Setup Alice's wallet with enough funds to further fund the other participating wallets.

```
e2e-test$ ./p2p/local-ini.sh
Create Alice...
Done.  Put some funds in Alice's address: addr_test1vr9jluna269fnputlnefc48kame4g9j6uy04638tm8h57lq4njtyh
Now funding Alice...
Estimated transaction fee: Coin 170693
Transaction successfully submitted.
Waiting to see initial funding tx onchain...

Transaction Id: ae3b19f2ed828918fe9d13434db6794432348fb09a51f4d6bb78c49c0fcbe80f

Alice's wallet:
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
ae3b19f2ed828918fe9d13434db6794432348fb09a51f4d6bb78c49c0fcbe80f     0        1000000000 lovelace + TxOutDatumNone
```

Initialization

Serialize the plutus script for *onRamp* validator and fund the remaining wallets.

```
e2e-test$ ./p2p/01-p2p-init-transaction.sh

Initialization...

Creating 'onRamp' script...
Resolving dependencies...

Done serializing plutus script.

Initializing sellers...

Initializing buyer...

Funding wallets...
Estimated transaction fee: Coin 175577
Transaction successfully submitted.
Waiting to see initial funding tx onchain...

Transaction Id: f474860fbfbfae1e867ebd593c852f1883f0b1951fee28de3bce55652ea398dd

Initialization completed.
```

After initialization, each seller (Barbara, Bob and Brandon) has 50 Ada, and buyer (Charlie) has 10 Ada.

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/barbara.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
f474860fbfbfae1e867ebd593c852f1883f0b1951fee28de3bce55652ea398dd     0        50000000 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/bob.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
f474860fbfbfae1e867ebd593c852f1883f0b1951fee28de3bce55652ea398dd     1        50000000 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/brandon.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
f474860fbfbfae1e867ebd593c852f1883f0b1951fee28de3bce55652ea398dd     2        50000000 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/charlie.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
f474860fbfbfae1e867ebd593c852f1883f0b1951fee28de3bce55652ea398dd     3        10000000 lovelace + TxOutDatumNone
e2e-test$
```

Sending sell offers

Now each seller (Barbara, Bob and Brandon) posts a sell offer at the *onRamp* script. In our example, each is selling around (but not exactly) 30 Ada, for approximately (but not exactly) 30 USD; exact numbers are chosen at random.

```
e2e-test$ ./p2p/02-p2p-add-sellers.sh
Adding sellers...

Barbara sells 30001079 lovelace...

Wrote barbaraSellDatum.cbor

Sending Barbara's sell offer UTxO...
Estimated transaction fee: Coin 175269
Transaction successfully submitted.

Bob sells 30010444 lovelace...

Wrote bobSellDatum.cbor

Sending Bob's sell offer UTxO...
Estimated transaction fee: Coin 175269
Transaction successfully submitted.

Brandon sells 30014411 lovelace...

Wrote brandonSellDatum.cbor
```

```
Sending Brandon's sell offer UTxO...
Estimated transaction fee: Coin 175269
Transaction successfully submitted.
```

After Tx is submitted, we can see the sell orders at the *onRamp* script:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/onRamp.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
2ec2d263c61b5160e749d387d46c564156438e8f518188bfd827b87c6ee67da0     0        30001079 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \164\149#\203\156N\"u\172\235\&9\246\ACKZx\165\165\247\244\
\144a9\169/\173;\DC3\178\228/\152\190\CAN\RS\161@\161@\SUB\SOH\201\199\183X \162\170LsN\228FE\156\199\DC3\SO_~\186\150\
\177tn\204B\224VV*\136\234Py\ENQp7\216z\128\216y\159\SUBg\215\134\SUB\255\255" (ScriptDataConstructor 0 [ScriptData\
Bytes "\164\149#\203\156N\"u\172\235\&9\246\ACKZx\165\165\247\244\144a9\169/\173;\DC3\178\228/\152\190",ScriptData\
Number 30,ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30001079)])],Script\
DataBytes "\162\170LsN\228FE\156\199\DC3\SO_~\186\150\177tn\204B\224VV*\136\234Py\ENQp7",ScriptDataConstructor 1 [],\
ScriptDataConstructor 0 [ScriptDataNumber 1742177818]]))
312cc6150db0cbd2b7c8473f293f914a846fc347e805495379a13859a52e7ee3     0        30010444 lovelace + TxOutDatumInline\
BabbageEraOnwardsConway (HashableScriptData "\216y\159X \135h=\168\&7\DC3v\145\ETB\SO\SUB\170\&9\STX\160\DEL\225c\
\156\199\t\244`+\SO\ESCr\241\151s\244\205\CAN%\161@\161@\SUB\SOH\201\236LX \153G\245\NAK`\205}O\ESC\226\b\172\219R,\
\172\230ED\176\FS\239O)g\aC\a\155\245>\230\216z\128\216y\159\SUBg\215\134\ESC\255\255" (ScriptDataConstructor 0 [Script\
DataBytes "\135h=\168\&7\DC3v\145\ETB\SO\SUB\170\&9\STX\160\DEL\225c\156\199\t\244`+\SO\ESCr\241\151s\244\205",Script\
DataNumber 37,ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30010444)])],\
ScriptDataBytes "\153G\245\NAK`\205}O\ESC\226\b\172\219R,\172\230ED\176\FS\239O)g\aC\a\155\245>\230",ScriptData\
Constructor 1 [],ScriptDataConstructor 0 [ScriptDataNumber 1742177819]]))
9f10f5faffa37eb60f781f98efdff56122eaf77c0bc700185136f91951ae1292     0        30014411 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \DLE\146\245Td\DLE\164D'\249tb\225\219\141G\138\153M$%\RS:\
\246c\169C\242\251\141I\243\CAN#\161@\161@\SUB\SOH\201\251\203X \182\149?\243\214\134\222\140\240\170\DC1gH\245\149\
\205\155\226\201\250h`\218\243Ef\238\237j\155a\218\216z\128\216y\159\SUBg\215\134\ESC\255\255" (ScriptDataConstructor\
 0 [ScriptDataBytes "\DLE\146\245Td\DLE\164D'\249tb\225\219\141G\138\153M$%\RS:\246c\169C\242\251\141I\243",ScriptData\
Number 35,ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30014411)])],Script\
DataBytes "\182\149?\243\214\134\222\140\240\170\DC1gH\245\149\205\155\226\201\250h`\218\243Ef\238\237j\155a\218",\
ScriptDataConstructor 1 [],ScriptDataConstructor 0 [ScriptDataNumber 1742177819]]))
```

We can verify that balance on each of the seller's wallets has diminished by the amount sent to the script (around 30 Ada) plus Tx fees:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/barbara.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
2ec2d263c61b5160e749d387d46c564156438e8f518188bfd827b87c6ee67da0     1        19823652 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/bob.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
312cc6150db0cbd2b7c8473f293f914a846fc347e805495379a13859a52e7ee3     1        19814287 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/brandon.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
9f10f5faffa37eb60f781f98efdff56122eaf77c0bc700185136f91951ae1292     1        19810320 lovelace + TxOutDatumNone
e2e-test$
```

Buyer selects a sell offer and posts a buy order

Buyer queries the UTxOs at the *onRamp* script for available sell offers. Then selects the best offer according to some criterion, for example the ratio value/price. Buyer communicates his intent to seller off-chain and seller signs the buy order. Datum of corresponding UTxO is updated accordingly:

```
e2e-test$ ./p2p/03-p2p-buy-order.sh

Choosing best sell offer...

Selected TxOutRef: 2ec2d263c61b5160e749d387d46c564156438e8f518188bfd827b87c6ee67da0#0
Selected seller: Barbara

Generating buyer's datum and redeemer...

Seller signs datum updated with buyer's pub-key-hash...

Verifies signature:
True

Signature: c23b97ea34d8bc944cf8b5ed1f512bcb46ce9091c38da571d4a63cdadbdf1feeadc44884efdef71c56e463ac039a0ae7eff620d\
c933feb8bdfa4d180a2c7df0a

Wrote charlieBoughtDatum.cbor
Wrote barbaraSoldRedeemer.cbor

Buy-order transaction...
Estimated transaction fee: Coin 473192
Transaction successfully submitted.
Waiting to see buy order tx onchain...

Transaction Id: 6927e58e9c453b36503d7fd8367a1e115d56b9ffd2fa4fab2027e4bbc4f8e6d4
```

Claim funds

After making fiat deposit, buyer wants to claim the corresponding funds in Ada. A fiat admin certifies the Tx by signing the corresponding payment info (hash of).

```
e2e-test$ ./p2p/04-p2p-claim-transaction

Fiat admin signs hash of fiat payment info...

Verifies signature:
True

Signature: 37d68710095c87244e03784598b80a4c055a7ccd8205a68309be3c7f0d4f27594a251a96c352582932bc7074c8f90366b8372d6\
41755738b987cca54f52ff409

Claim transaction...
Estimated transaction fee: Coin 462487
Transaction successfully submitted.
```

```
Waiting to see claim tx onchain...

Transaction Id: 198b93e7f741975f399274ca07a243d0abff27f94abb5d87e078d9c1cbe66a8a
```

So now the bouth assets are in Charlie's wallet:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/charlie.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
198b93e7f741975f399274ca07a243d0abff27f94abb5d87e078d9c1cbe66a8a     0        30001079 lovelace + TxOutDatumNone
198b93e7f741975f399274ca07a243d0abff27f94abb5d87e078d9c1cbe66a8a     2        6064321 lovelace + TxOutDatumNone
```

and there remain only two sell offers at the *onRamp* script:

```
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/onRamp.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
312cc6150db0cbd2b7c8473f293f914a846fc347e805495379a13859a52e7ee3     0        30010444 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \135h=\168\&7\DC3v\145\ETB\SO\SUB\170\&9\STX\160\DEL\225c\
 \156\199\t\244‘+\SO\ESCr\241\151s\244\205\CAN%\161@\161@\SUB\SOH\201\236LX \153G\245\NAK‘\205}O\ESC\226\b\172\219R,\
\\172\230ED\176\FS\239O)g\aC\a\155\245>\230\216z\128\216y\159\SUBg\215\134\ESC\255\255" (ScriptDataConstructor 0 [Script\
DataBytes "\135h=\168\&7\DC3v\145\ETB\SO\SUB\170\&9\STX\160\DEL\225c\156\199\t\244‘+\SO\ESCr\241\151s\244\205",Script\
DataNumber 37,ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30010444)])],\
ScriptDataBytes "\153G\245\NAK‘\205}O\ESC\226\b\172\219R,\172\230ED\176\FS\239O)g\aC\a\155\245>\230",ScriptData
Constructor\ 1 [],ScriptDataConstructor 0 [ScriptDataNumber 1742177819]]))
9f10f5faffa37eb60f781f98efdff56122eaf77c0bc700185136f91951ae1292     0        30014411 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \DLE\146\245Td\DLE\164D’\249tb\225\219\141G\138\153M$%\RS:\
 \246c\169C\242\251\141I\243\CAN#\161@\161@\SUB\SOH\201\251\203X \182\149?\243\214\134\222\140\240\170\DC1gH\245\149\
\205\155\226\201\250h‘\218\243Ef\238\237j\155a\218\216z\128\216y\159\SUBg\215\134\ESC\255\255" (ScriptDataConstructor\
 0 [ScriptDataBytes "\DLE\146\245Td\DLE\164D’\249tb\225\219\141G\138\153M$%\RS:\246c\169C\242\251\141I\243",ScriptData\
Number 35,ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30014411)])],Script\
DataBytes "\182\149?\243\214\134\222\140\240\170\DC1gH\245\149\205\155\226\201\250h‘\218\243Ef\238\237j\155a\218",\
ScriptDataConstructor 1 [],ScriptDataConstructor 0 [ScriptDataNumber 1742177819]]))
```

After the set deadline expiration (in our example, one minute after posting sell offers), seller's can reclaim the unexecuted orders:

```
e2e-test$ ./p2p/05-p2p-cancel-transaction.sh
Cancel sell orders...

Barbara has sold or closed sell offer; nothing to cancel.

Canceling Bob’s order...
Estimated transaction fee: Coin 451192
Transaction successfully submitted.
Bob’s order has ben canceled.

Canceling Brandon’s order...
Estimated transaction fee: Coin 451192
Transaction successfully submitted.
Brandon’s order has ben canceled.
```

So now there remain no UTxOs at the onRamp script,

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/onRamp.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
e2e-test$
```

and the balances at the seller's wallets have been updated accordingly:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/barbara.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
2ec2d263c61b5160e749d387d46c564156438e8f518188bfd827b87c6ee67da0     1        19823652 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/bob.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
2f4c0ced001b5c72ae4c9964540354f3d9f88df0119e08661ebb0cbc8874daa1     0        30010444 lovelace + TxOutDatumNone
2f4c0ced001b5c72ae4c9964540354f3d9f88df0119e08661ebb0cbc8874daa1     1        19363095 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/brandon.addr) --testnet-magic 42
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
6a140f47bfe7820196681e7c92615c28b98afde64f798c96c02614512a6fbc19     0        30014411 lovelace + TxOutDatumNone
6a140f47bfe7820196681e7c92615c28b98afde64f798c96c02614512a6fbc19     1        19359128 lovelace + TxOutDatumNone
```

As expected, one of the seller's has sold around 30 Ada and the other two have recovered almost the original 50 Ada (the difference due to spent fees).