

ATOMIC SWAPS

PRIVATE, PEER-TO-PEER, SCRIPTLESS

ON BITCOIN

WITH RGB 0.11.1

Gideon Nweze
FOUNDER

BitMask / DIBA Team · Technical paper · bitmask-core v0.11.0-beta.10

***Abstract.** A purely peer-to-peer exchange of unique digital assets on bitcoin would allow a seller and a buyer to settle directly with each other without going through an exchange. Pre-signed offers provide part of the solution, but the main benefits are lost if a trusted third party is still required to hold funds or assets while the trade completes.*

We propose a solution to the settlement problem using client-side validation and a single Bitcoin transaction. The seller signs an offer that spends the output holding the asset. A buyer merges that offer with a payment, commits the transfer in a tapret output and broadcasts. The same transaction that pays the seller reassigns the asset. The system itself requires minimal structure: an order book for discovery and a relay for proof packages, neither of which holds funds or assets. The seller can leave after publishing and rejoin to settle, and the asset history stays off the chain, with its parties.

1. Introduction

Commerce in unique assets such as art, titles, tickets and certificates still relies on platforms that hold the asset, the cash, or both while a trade completes. In the words of the Bitcoin whitepaper, “What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.” [1] We apply that principle to the sale of a unique digital asset (UDA, the RGB21 schema) for BTC, as implemented in bitmask-core infrastructure [2].

2. Offers

“We define an electronic coin as a chain of digital signatures.” [1] We define an offer as one signature given in advance. The seller builds a partial transaction (PSBT v2 [4]) that spends the output holding the asset, adds a host output that returns the input value and will carry the commitment, and adds a price output paying a fresh address. The input is signed with `SIGHASH_NONE | ANYONECANPAY` so that no one needs to be online to complete it. A proof package, the shard, and a blinded seal travel with the offer.

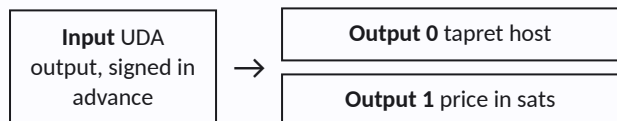


Fig. 1. The offer: the seller's partial transaction.

3. Seals and Commitment

An RGB asset is assigned to a single-use seal, which is a Bitcoin output [3]. Spending the output closes the seal, and the spending transaction commits to the state transition with a tapret commitment (Method::TapretFirst, [5]). The only on-chain trace is a hash hidden in a taproot output, which looks like any other spend. The order of events is given by the Bitcoin chain, not by the parties.

4. Client-Side Validation

The history of an asset is not on the chain. It travels with the asset in a consignment, a chain of transition bundles each anchored in a transaction. The buyer validates the contract against an esplora resolver and ingests the seller's anchors before paying. Each party validates the consignment it receives. Validation costs only the data a party already holds.

5. Network

The steps to run the swap are as follows:

- 1) The seller signs an offer and publishes it.
- 2) A buyer fetches the offer, validates its contract and history.
- 3) The buyer funds a bid: BTC inputs, change and an RGB invoice.
- 4) The buyer merges the PSBTs and checks the price output is kept.
- 5) The seller's host output is found by script binding; the tapret commitment is built.
- 6) The buyer posts the consignment to the relay, then broadcasts.
- 7) The match is recorded; the seller verifies and registers the commitment.

“Nodes can leave and rejoin the network at will” [1]. Here the seller leaves after step 1 and rejoins at step 7.

6. Incentive

The buyer funds the price and a fixed chain fee; other fee modes are refused. The seller receives the price in sats and the spent output's value back.

7. Other Assets on the Same Output

A seller's output may carry allocations of other contracts. For each one the offer includes an extra shard, and the merge writes blank transitions that reassign the allocation to the seller's host output. If any extra is RGB20 the host uses the RGB20 keychain, so those assets stay spendable by the seller. Selling one asset does not strand the others.

8. Verification

“It is possible to verify payments without running a full network node.” [1] Likewise a party needs only an esplora endpoint and its own stock: UTXOs and witnesses are prefetched, contracts are validated, and the transaction is broadcast through the same endpoint.

9. Combining and Splitting Value

“Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change.” [1] The merge keeps that shape. Inputs are ordered buyer's first, seller's first, then the rest. Outputs are the seller's host and price outputs, then the buyer's outputs. The host output's script is recorded in the offer, and the merge fails if it is missing.

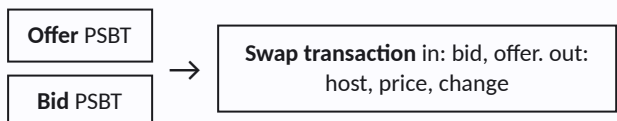


Fig. 2. Two partial transactions merge into one.

10. Privacy

Privacy can still be maintained “by breaking the flow of information in another place: by keeping public keys anonymous.” [1] Here the break is between chain and state. Asset data stays off chain, the recipient is named by a blinded seal, and fresh derivations are used for each offer and bid. The chain shows a transaction with a taproot output and no asset data. The shared order book is a separate place that keeps trade records.

11. Conclusion

“We have proposed a system for electronic transactions without relying on trust.” [1] This paper proposes the same for unique assets. Offers are signed once, merged once and settled in one Bitcoin transaction, with asset state kept private.

References

- [1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
- [2] DIBA Team, bitmask-core v0.11.0-beta.10, src/rgb.rs and src/rgb/swap.rs.
- [3] RGB protocol 0.11.1: rgb-ops, rgb-consensus, rgb-invoicing, rgb-psbt-utils.
- [4] BIP-174, Partially Signed Bitcoin Transaction Format; BIP-370, PSBT Version 2.
- [5] BIP-341, Taproot; BIP-86, Key Derivation for Single Key P2TR Outputs.